НОВА УКРАЇНСЬКА ШКОЛА

Джон Ендрю Біос

8

ІНФОРМАТИКА

ЦИФРОВІ ПІДЛІТКИ

Містить цифровий контент

© Binary Logic та © Видавництво «Формула». Усі права застережені. Жодна частина цієї публікації не може бути відтворена, збережена в пошукових системах або передана в будь-якій формі будь-якими способами без попереднього дозволу видавців. За жодних обставин жодна частина цього матеріалу не може бути скопійована для подальшого або для масового використання.

Гарячі клавіші в Excel

Enter ↵	Ввести надруковані дані в комірку.
Esc	Скасувати введення надрукованих даних.
Ctrl + B	Заповнити та перейти вниз.
Ctrl + K	Заповнити та перейти вправо.
=	Почати запис формули.
Esc	Скасувати результати правки комірки або рядки формул.
F2	Перейти в режим редагування вмісту комірки.
Ctrl + Enter ↵	Заповнити виділені комірки однаковими даними.
Alt + =	Виконати функцію **Автосума**.
Ctrl + I	Виконати або видалити форматування курсивом.
Ctrl + U	Виконати або видалити форматування підкресленням.
Ctrl + B	Виконати або видалити форматування жирним шрифтом.
Ctrl + Z	Скасувати останню дію.
Ctrl + Space	Вибрати стовпець.
Shift + Space	Вибрати рядок.

Корисні модулі в Python

Кросплатформений графічний інтерфейс для програм	**tkinter**
Обробка email-повідомлень	**email**
Модулі для роботи зі стиснутими даними	**gzip, zlib**
Взаємодія з операційною системою	**os**
Усе, що стосується математики	**math**
Генератор випадкових чисел	**random**
Модуль для створення ігор	**pygame**
Надає доступ до основного механізму відтворення звуку, що підтримується платформами Windows	**winsound**
Системні параметри та функції	**sys**
Моніторинг подій виконання	**sys.monitoring**
Надає можливість доступу й парсингу даних із вікіпедії	**wikipedia**
Надає функції та класи для операцій, пов'язаних із часом	**time**

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Основні теги HTML та CSS

`<!DOCTYPE>`	Визначає тип документа.
`<html> ... </html>`	Початок і кінець документа. Вказує програмі перегляду сторінок, що це HTML-документ.
`<head> ... </head>`	Заголовок. Визначає текст, що не відображається в тілі документа. Розміщує тег назви документа й теги для пошукових систем.
`<title> ... </title>`	Вказує назву документа, що відображається в заголовку вікна.
`<body> ... </body>`	Тіло документа. Визначає видиму частину документа.
`<h1> ... </h1>`	Створює найбільший заголовок.
`<h6> ... </h6>`	Створює найменший заголовок.
`<p> ... </p>`	Визначає новий абзац.
` `	Вставляє розрив рядка.
`<!--...-->`	Визначає коментар.
`<b> ... </b>`	Виділення жирним шрифтом.
`<i> ... </i>`	Виділення курсивом.
`<u> ... </u>`	Виділення підкресленням.
`<ol> ... </ol>`	Створює нумерований список.
`<ul> ... </ul>`	Створює ненумерований список.
`<li> ... </li>`	Створює елемент списку.
`<a href="url"></a>`	Створює гіперпосилання на іншу вебсторінку, інтернет-ресурс або файл.
`<img src="url">`	Додає зображення в HTML-документ.

Бережіть себе

1. Сидіть прямо, спина повинна спиратися на спинку стільця, ноги на підлозі.
2. Тримайте екран на відстані 50–70 см від очей.
3. Робіть перерви кожні 20–25 хвилин, щоб дати відпочити очам та розім'ятися.
4. Тримайте робоче місце чистим.
5. Використовуйте надійні паролі та часто змінюйте їх. Не передавайте свої паролі іншим.
6. Не відкривайте невідомі сайти та не розповідайте особисту інформацію.
7. Не користуйтеся смартфоном або планшетом під час заряджання.

© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

НОВА УКРАЇНСЬКА ШКОЛА

Джон Ендрю Біос

ІНФОРМАТИКА 8

Підручник для 8 класу закладів загальної середньої освіти

Рекомендовано Міністерством освіти і науки України

ЦИФРОВІ ПІДЛІТКИ

Київ
Видавництво «Лінгвіст»
2025

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

УДК 004*кл8(075.3)
Б 65

Рекомендовано Міністерством освіти і науки України
(наказ Міністерства освіти і науки України від 21.02.2025 № 347)

Видано за рахунок державних коштів. Продаж заборонено

Рецензенти:

Григорій Громко, учитель інформатики вищої категорії Нечаївського ліцею ім. Ю. І. Яновського Кіровоградської області, вчитель-методист;

Ірина Сокол, кандидатка педагогічних наук, доцентка, учителька інформатики вищої категорії, вчителька-методистка Комунального закладу «Запорізька спеціалізована школа-інтернат II–III ступенів «Січовий колегіум» Запорізької обласної ради.

Біос Джон Ендрю

Б 65 Інформатика: підруч. для 8 кл. закл. загальн. середн. освіти / Джон Ендрю Біос. — К. : Видавництво «Лінгвіст», 2025. — 208 с. : іл.

ISBN 978-617-8146-47-4

Підручник призначений для вивчення інформатики в 8 класі, розроблений згідно з Державним стандартом базової середньої освіти. Зміст підручника має практичне спрямування та реалізує компетентнісний підхід у навчанні. Розвиває наскрізні вміння та компетенції.

УДК 004*кл8(075.3)

Підручник створено відповідно до модельної навчальної програми «Інформатика. 7–9 класи» для закладів загальної середньої освіти (автори: О. В. Пасічник, Л. З. Козак, А. В. Ворожбит).

Електронний додаток до підручника можна знайти за посиланням:
https://lms.e-school.net.ua/courses/join/b69a1c3c/

ISBN 978-617-8146-47-4

Усі права застережено. All rights reserved
Copyright © 2023 by Binary Logic
Published by arrangement with Binary Logic

Підручник охороняється авторським правом.
Виключне майнове право на використання твору в Україні належить ТОВ «Видавництво "Лінгвіст"». Використання твору, а також його фрагментів без дозволу правовласника та видавця заборонено

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

ЗМІСТ

Розділ 1. Основи інформатики 6

Тема 1. Кодування даних 7
Тема 2. Архітектура комп'ютера 11
Тема 3. Операційні системи 16
Тема 4. Збереження даних 25
Тема 5. Захист цифрових пристроїв та інформації 30
Тема 6. Сучасні технології 35
Тема 7. Технології та навколишнє середовище 42

Розділ 2. Працюємо із числами 46

Тема 1. Збір даних 47
Тема 2. Додаткові параметри форматування 52
Тема 3. Прості обчислення 60
Тема 4. Складні розрахунки 64
Тема 5. Логічні функції 70
Тема 6. Посилання 80
Тема 7. Діаграми та графіки 84
Тема 8. Дослідження та гіпотези 90

Розділ 3. Програмування 94

Тема 1. Що таке програма 95
Тема 2. Змінні та команди 99
Тема 3. Введення та виведення даних 104
Тема 4. Умови та розгалуження 107
Тема 5. Функції 110

Розділ 4. Проєктування графічних інтерфейсів 114

Тема 1. Графічний інтерфейс продукта. UI/UX-дизайн 115
Тема 2. Векторний графічний редактор 119
Тема 3. Графічні примітиви 124
Тема 4. Типографіка. Ефекти та фільтри 131
Тема 5. Композиція та макет 137
Тема 6. Прототип. Документація інформаційного продукту 141

Розділ 5. Сайт і його верстка 144

Тема 1. Створення вебсторінки 145
Тема 2. HTML та CSS 152
Тема 3. Додаємо контент 157
Тема 4. Вебформи 160
Тема 5. Хостинг та SEO 170
Тема 6. Координація роботи команди. Система контролю версій 182

Розділ 6. Дизайн-мислення 191

Тема 1. Бренд і фірмовий стиль 192
Тема 2. Елементи стилю та брендбук 194
Тема 3. Технології генерування ідей 201
Тема 4. Дизайн-мислення 204

© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

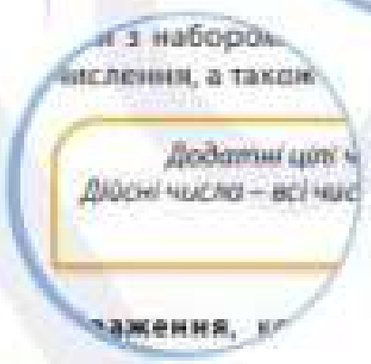
Привіт!

Разом ми реалізуємо місію – зрозуміти ЦИФРОВИЙ світ і встигнути за його швидкими змінами! Вмикайте свій комп'ютер та вирушайте за мною!

У підручнику ви натрапите на такі позначки

Розкриває елемент інтерфейсу та функціональні можливості інструментів.

Вказує на елемент або інструмент, який треба обрати чи натиснути на нього.



Звертає увагу на важливі поняття і термінологію.

ІСТОРІЯ – містить факти з історії розвитку ІКТ.

ПІДКАЗКА – акцентує увагу на альтернативних варіантах виконання поставлених завдань і додаткових інструкціях.

ДОВІДКА – розкриває додаткові можливості.

БЕРЕЖІТЬ СЕБЕ – нагадує про хвилинки відпочинку.

Відвідайте цифрові ресурси та знайдіть ті, які допоможуть вам працювати з підручником. Повторюйте те, чого навчилися, у будь-який час та в будь-якому місці. Використовуйте для цього зручний для вас цифровий пристрій.

https://formula.education/online-ict-8-klas/

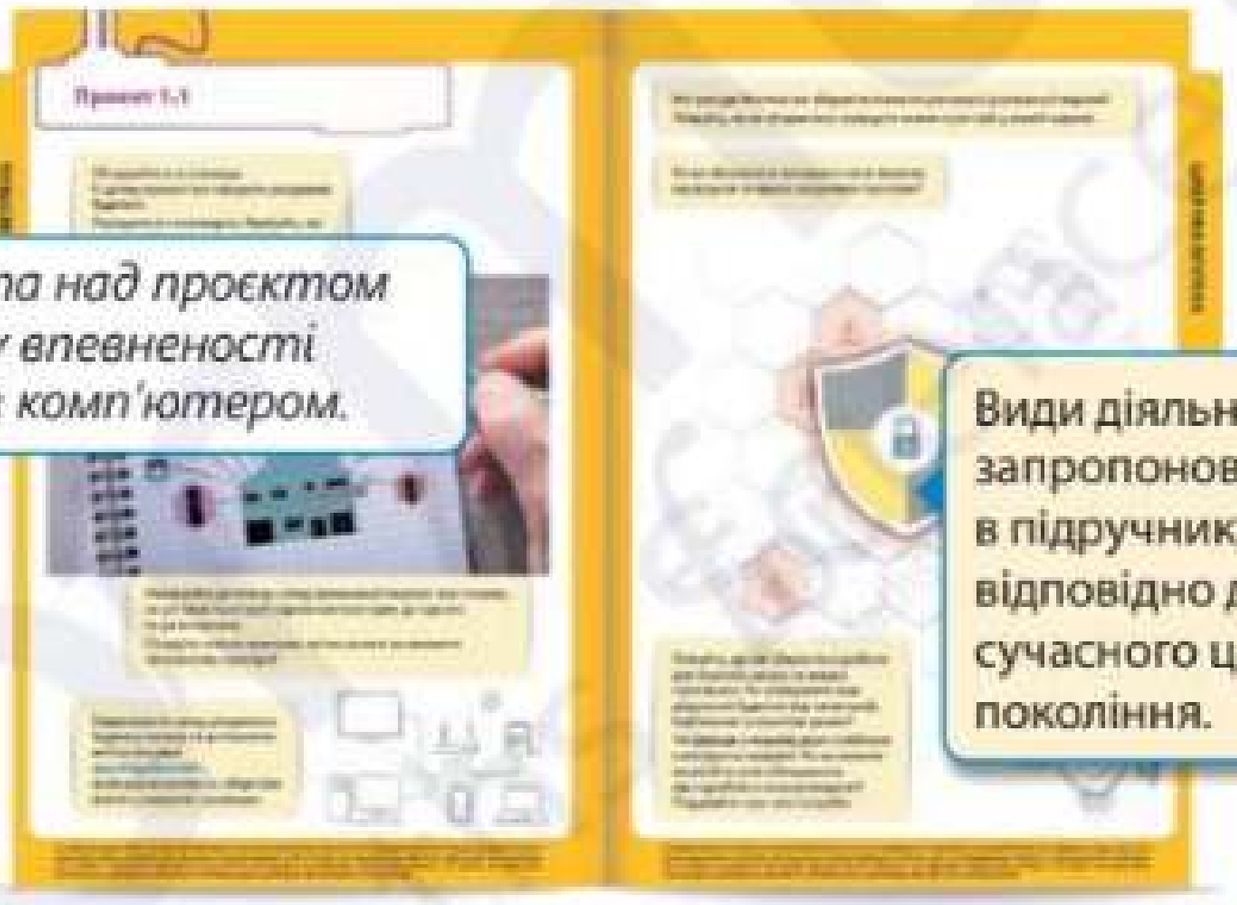
© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Познайомимось!

Матеріал підручника містить актуальні програмні продукти.

Спіральна структура підручника дає змогу повторювати матеріал у класі щоразу на більш високому рівні.

Чіткі пояснення та ілюстрації. Сучасні приклади.



Командна робота над проєктом сприяє розвитку впевненості під час роботи з комп'ютером.

Види діяльності, запропоновані в підручнику, підібрано відповідно до потреб сучасного цифрового покоління.

Сформульовано цілі навчання та функціональні вміння.

Розділ **Інші платформи** відкриває для здобувачів і здобувачок освіти доступні альтернативи програм та їхніх інструментів.

Нова лексика на перехресних темах.

© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

1. Основи інформатики

У наш час технології розвиваються швидкими темпами й кардинально змінюють життя. У цьому розділі ви дізнаєтесь, як працює комп'ютер і кодуються дані. Обговоримо його архітектуру та принципи зберігання й опрацювання даних. Дізнаєтеся про позитивний і негативний вплив технологій на навколишнє середовище та способи запобігання негативним проявам.

У цьому розділі ви навчитеся:

- розуміти двійкову та шістнадцяткову системи числення;
- переводити числа з однієї системи числення в іншу;
- мати уяву про архітектуру комп'ютера;
- розуміти, навіщо потрібна операційна система;
- вмикати мережевий брандмауер;
- архівувати дані для економії дискового простору;
- належним чином користуватися цифровими пристроями;
- дбати про навколишнє середовище;
- використовувати технології для особистої та суспільної користі.

У цьому розділі ви дізнаєтеся:

- який вигляд мають дані в комп'ютерній системі;
- як комп'ютер зберігає та опрацьовує дані;
- як комп'ютери керують пам'яттю, процесами та файлами;
- чому конфіденційність в інтернеті є дуже важливою;
- як управляти дисковим простором;
- як комп'ютери допомагають ухвалювати рішення та будувати гіпотези;
- як технології впливають на навколишнє середовище;
- як використовувати технології на благо.

Програмне забезпечення:

- Microsoft Windows.
- ChromeOS;
- Linux;
- MacOS;
- Android;
- iOS.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 1
Кодування даних

У наш час натрапити на сучасний цифровий комп'ютер можна будь-де: вдома, у школі, в офісах компаній тощо. Постійними нашими супутниками є смартфони, які носимо із собою в кишені, а також потужні ноутбуки, на яких можна працювати навіть у дорозі.

Десяткова система – двійкова система числення

Оскільки комп'ютер працює за допомогою електроенергії, то всі його внутрішні компоненти «розуміють» лише два стани: **струм є** або **струму немає**. Комп'ютери – машини, які працюють на основі двійкової системи числення. Це своєрідна мова, яка використовується всередині комп'ютера. У ній для запису чисел застосовуються лише два символи: **0** (струму немає) та **1** (струм є).

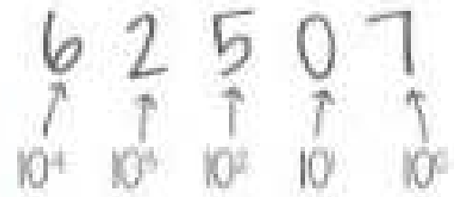
Використовуючи певну послідовність 0 та 1, можна передати всі інші числа. У десятковій системі числення будь-яке число можна записати з використанням лише цифр: 0; 1; 2; 3; 4; 5; 6; 7; 8; 9. Значення числа залежить від позиції цифр у числі. Розряд є «робочим місцем» цифри в числі. Починаючи з крайнього правого розряду, значення цифр збільшується в 10 разів із кожною позицією вліво. Кожна одиниця вищого розряду містить 10 одиниць нижчого розряду, суміжного з ним.

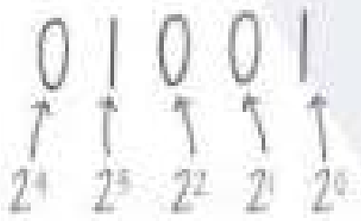
Цифри – це «цеглинки» для побудови чисел.

Число – математичне поняття для вираження кількості.

Передаємо число 131 у десятковій системі

Цифри	1	3	1	
Розряд	$10^2 = 100$	$10^1 = 10$	$10^0 = 1$	
	1×100 +	3×10 +	1×1 =	131_{10}





Такий самий принцип використовується і у двійковій системі числення. Різниця полягає в тому, що тут кожна цифра може приймати одне з двох значень 0 чи 1, а позиційне значення дорівнює числу 2 в певному степені (один, два, чотири, вісім тощо).

Число 131 у двійковій системі має вигляд 10000011

Цифри	1	0	0	0	0	0	1	1	
Розряд	$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$	
	1×128 +	0×64 +	0×32 +	0×16 +	0×8 +	0×4 +	1×2 +	1×1 +	131

Зверніть увагу, що значення місця крайньої правої цифри в будь-якій системі дорівнює 1.

Будь-яке число (крім нуля) в нульовому степені дорівнює одиниці. Отже, $10^0 = 2^0 = 1$.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Шістнадцяткова система числення

Із часом комп'ютери ставали потужнішими, тому вони могли опрацьовувати дедалі більше бітів інформації. Щоб скоротити дуже довгі двійкові рядки, у комп'ютерах почали використовувати шістнадцяткову систему числення. У **шістнадцятковій системі числення** кожне число записується за допомогою 16 символів. Звичайно, у цьому випадку слід використовувати додаткові символи, які відповідатимуть значенням 10, 11, 12, 13, 14 та 15. Отже, літера A позначає число 10, B – 11, C – 12 тощо.

16 цифр шістнадцяткової системи

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

У шістнадцятковій системі кожне позиційне значення дорівнює числу 16 у певному степені, який збільшуються на 1 в міру переміщення вліво вздовж числа. Очевидно, позиційне значення крайньої цифри справа дорівнює 1.

Переведемо число 1C8A в десяткову систему

Цифри	1	C	8	A	
Розряд	$16^3 = 4096$	$16^2 = 256$	$16^1 = 16$	$16^0 = 1$	
	1 × 4096 (=4096) +	12 × 256 (=3072) +	8 × 16 (=128) +	10 × 1 (=10) =	7306

Переведення чисел з одної системи числення в іншу

За допомогою **Windows Калькулятора** можна дуже легко перевести число з однієї системи в іншу.

Щоб перевести число в шістнадцяткову систему:

- відкрийте **Калькулятор**;
- натисніть **Відкрити навігацію** (1) і виберіть режим **Програмний** (2);
- оберіть систему числення (3);
- введіть число в обраній системі числення (4);
- на екрані з'явиться число у відповідних системах числення (5).

Оскільки шістнадцяткова система містить певні літери, то комп'ютерні програмісти люблять створювати та використовувати у своїх програмах «магічні цифри», які утворюють слова.

Наприклад, C0FFEE використовується як «магічне число» в різних експериментальних проєктах та програмуванні мікроконтролерів, а Apple використовує в iOS 8BADF00D («з'їв погану їжу») для позначення помилки програми.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Комп'ютерне подання даних

Комп'ютер опрацьовує різні дані. Оскільки комп'ютери можуть працювати лише з двійковою системою числення, то нам необхідно з'ясувати, як дані різних типів, як-от числа, текст, зображення та відео, перетворити у двійкові числа.

Нам уже відомо, як десяткові додатні цілі числа перевести у двійкові числа. Ми маємо справу з дійсними числами. Існують системи подання, які дають можливість перевести будь-яке число в рядок із нулями (0) та одиницями (1).

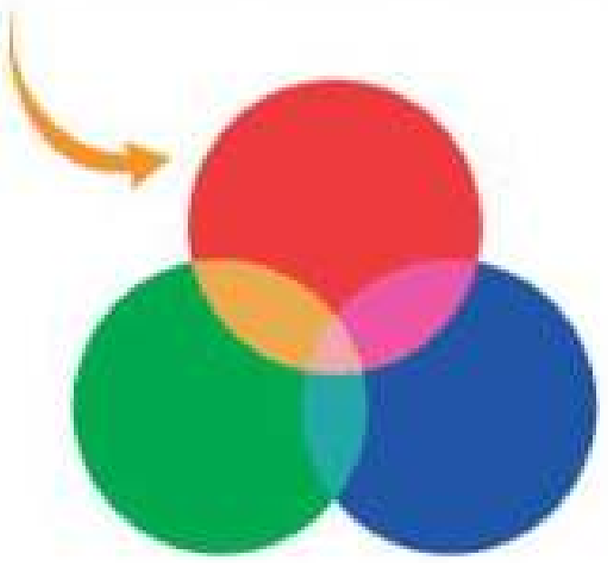
Подання тексту на комп'ютерах відбувається за допомогою набору символів. **Набір символів** – таблиця із символами та двійковими кодами, які використовуються для представлення цих символів. Одним із найбільш використовуваних наборів символів є **набір символів ASCII**, який показано нижче. ASCII (англ. American Standard Code for Information Interchange) означає **Американський стандартний код для інформаційного обміну**.

Перейдіть за посиланням formula.education/wp-content/uploads/2024/12/informatyka-8-1r-tablyczya-internet-pidtrymka-2.pdf та ознайомтеся з набором символів ACSII, дізнайтеся, як вони кодуються в різних системах числення, а також що ховається за символами, які не друкуються.

Додатні цілі числа – це числа, більші ніж нуль: 1, 2, 3, 4…
Дійсні числа – всі числа, які існують.
Від'ємні числа – це дійсні числа, які є меншими ніж нуль.

Щоб зберегти зображення, колір кожного пікселя потрібно перевести в код, зрозумілий комп'ютеру. Найчастіше для цього використовується модель RGB, у якій кожен колір представлений значенням трьох основних кольорів – червоного, зеленого та синього. Для кожного пікселя зберігається три значення, по одному для кожного кольору. Ці значення мають по 8 біт кожний та розміщені в діапазоні від 0 до 255, що дозволяє відтворювати широку палітру кольорів – від повністю чорного до найрізноманітніших відтінків кольорів. Отже, зображення – це код, який передає значення інтенсивності трьох кольорів пікселів, із яких складається це зображення.

Відображення кольору на моніторі	Інтенсивність кольорів монітора R	G	B
Білий	255	255	255
Червоний	255	0	0
Зелений	0	255	0
Синій	0	0	255
Блакитний	0	255	255
Пурпуровий	255	0	255
Жовтий	255	255	0
Чорний	0	0	0



ПІДКАЗКА

У комп'ютерах мінімальна одиниця інформації – біт, який може бути 0 або 1. Слово «біт» (англ. bit) – це скорочення від binary digit (англ. двійковий знак).

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Розглянемо, як кодуються растрові та векторні зображення.

Кольорові пікселі складають растрове зображення. Кожен піксель має свій колір і місце розташування на зображенні.

Комп'ютер зберігає растрове зображення як упорядковану прямокутну таблицю (матрицю) числових значень, де кожна комірка (піксель) містить закодовану інформацію про колір. Якщо зображення має роздільну здатність 1024 × 768 пікселів, то комп'ютер буде зберігати 1024 · 768 = 786 432 числа для його представлення.

На відміну від растрових, векторні зображення складаються з математичних формул, які описують лінії, криві та форми. Вони не мають пікселів і можуть бути збільшеними чи зменшеними без втрати якості.

Комп'ютер зберігає векторне зображення як набір математичних формул і команд для побудови різних фігур. Коло може бути задане рівнянням та координатами центра, а лінія – двома точками. Ці формули займають набагато менше місця в пам'яті комп'ютера, ніж растрове зображення.

Відео – найскладніший для представлення тип даних, але загалом його можна уявити як серію зображень, збережених у двійковому форматі, які відтворюються одне за одним. Ці зображення зазвичай стискаються для того, щоб заощадити місце на диску та обробити їх якомога швидше.

Практичне завдання

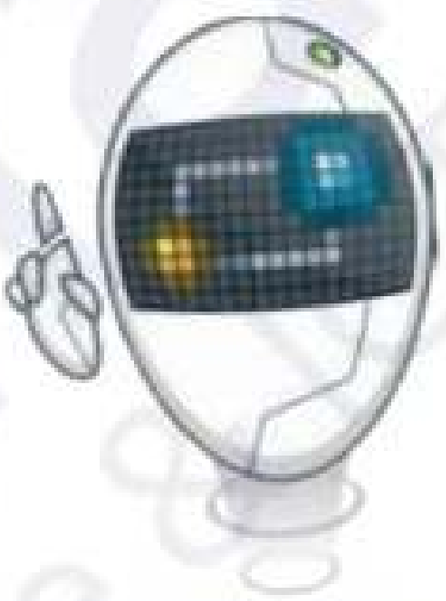
Завантажте файл-заготовку зі сторінки інтернет-підтримки. Скористайтеся підказками та переведіть числа в різні системи числення. Закодуйте колір у системі RGB.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 2

Архітектура комп'ютера

Усі комп'ютери виконують чотири основні дії: введення, опрацювання, зберігання та виведення даних. Щоб комп'ютери були корисними, їм потрібно «сказати», що робити з даними, тобто надати команди. Ці команди потрібно десь зберігати, щоб комп'ютер мав до них доступ. Оскільки комп'ютери – двійкові машини, то й команди мають бути написані двійковими послідовностями. Отже, команди й дані можуть зберігатися в одному й тому самому місці, а процесор послідовно вибирає з пам'яті команди й дані, опрацьовує їх. Іншою важливою особливістю комп'ютерів є те, що елементи, які опрацьовують інформацію, відокремлені від елементів, які зберігають інформацію.



Ці базові принципи формують **архітектуру фон Неймана**, на якій побудовані всі сучасні комп'ютери. У такій архітектурі можна виокремити деякі компоненти, які працюють разом для того, щоб комп'ютер міг функціонувати.

ІСТОРІЯ

Джон фон Нейман описав однойменну архітектуру комп'ютера разом з іншими інженерами під час розробки ENIAC у 1945 році. Це відомий американський математик угорського походження, який зробив великий внесок у різні галузі, як-от математика, фізика та комп'ютерні науки.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Цикл вибірка – декодування – виконання

Попри появу нових пристроїв та зміну вигляду комп'ютера, загальний принцип його роботи не змінився. Розглянемо, як відбувається виконання команд та опрацювання даних. Цей процес відомий як цикл: вибірка – декодування – виконання. Пам'ятайте, що й дані, й команди зберігаються в оперативній пам'яті.

Етапи циклу вибірка – декодування – виконання показані на схемі.

ДОВІДКА

Тип пам'яті **ROM** (англ. Read Only Memory – пам'ять тільки для читання) є ще одним типом пам'яті. Забезпечує довільний доступ до даних, але енергонезалежна. Команди, які зберігаються в ній, є постійними, їх не можна стерти чи перезаписати, їх можна лише прочитати. Саме тому ROM використовується для зберігання команд, які потрібні для запуску комп'ютера. Ці команди називають мікропрограмою, або вбудованим програмним забезпеченням чи «прошивкою».

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Оперативна пам'ять

Один із типів оперативної пам'яті, який використовують усі комп'ютери, називається **RAM** (англ. Random Access Memory – пам'ять із довільним доступом).

Це означає, що кожен байт даних у пам'яті доступний і його можна змінити. RAM повинна бути дуже швидкою, щоб забезпечувати швидкий доступ до даних і команд, необхідних CPU. RAM потребує електроживлення, тобто її вміст зберігається лише доти, доки комп'ютер увімкнений.

Зовнішня пам'ять

Оперативна пам'ять тимчасова та має обмежений обсяг. Отже, нам потрібен інший тип накопичувача, де можна безпечно зберігати дані, коли вони більше не опрацьовуються або коли комп'ютер вимкнений. Існують й енергонезалежні модулі пам'яті, які можуть зберігати свій вміст без постійного живлення, як-от флеш-пам'ять на USB-накопичувачах. На них дані зберігаються після того, як ви від'єднуєте пристрій від комп'ютера, але доступ потребує трохи більше часу. Такі типи пристроїв називають зовнішніми пристроями зберігання даних, найвідоміший із них – жорсткий диск (HDD або SSD).

До настільного комп'ютера може бути підключено один або декілька жорстких дисків. Вони працюють за допомогою голівки зчитування / запису, яка рухається по магнітному диску, що обертається, зчитуючи або зберігаючи дані. Поверхня кожного магнітного диска логічно поділена на доріжки та сектори. **Доріжка** – це концентричне коло на поверхні диска. Кожна доріжка розділена на сектори. Кожен **сектор** містить **блок** даних у вигляді безперервної послідовності бітів.

Голівка зчитування / запису диска розташована на кронштейні, який переміщується з однієї доріжки на іншу. Щоб зчитати або записати дані, жорсткому диску потрібна команда введення / виведення, яка вказує доріжку та сектор. Коли голівка зчитування / запису розміщується над відповідною доріжкою, вона чекає, поки під нею розташується відповідний сектор. Тоді відбувається доступ до відповідного блоку даних. Процес зчитування та запису даних на жорсткому диску визначає чотири показники його ефективності.

Різні програми мають різні вимоги. Системі баз даних потрібен швидкий час доступу, оскільки вона постійно читає та записує тисячі записів, розташованих по всьому диску. Коли відтворюється HD-відео, потрібно, щоб жорсткий диск забезпечував високу швидкість передачі даних, оскільки кожна секунда відео містить багато даних.

	Час пошуку – час, за який голівка зчитування / запису позиціонується над указаною доріжкою.
	Час затримки – це час пошуку потрібного сектора.
	Час доступу – час, необхідний для початку зчитування блоку. Це сума часу пошуку й затримки.
	Швидкість передачі – це швидкість, з якою дані передаються з диска в оперативну пам'ять.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Сучасні жорсткі диски (HDD), що їх можна знайти в наших комп'ютерах, складаються з декількох магнітних дисків, які називаються **пластинами**. Вони розташовані одна на одній, кожна має власну голівку для зчитування / запису, яка прикріплена до валу (шпинделя), що обертається. Всі доріжки, які вишикувалися одна під одною, називають циліндром. Щоб знайти певні дані на жорсткому диску, у команді до диска потрібно вказати номер пластини, номер циліндра та сектор.

Твердотільний накопичувач (SSD) – пристрій на основі мікросхем пам'яті та контролера керування ними. У ньому немає рухомих механічних частин, як у HHD.

Пристрої введення

Для взаємодії користувача з комп'ютером існують пристрої введення. Ці пристрої дозволяють користувачеві передавати дані та команди до комп'ютера. Без них було б неможливо керувати роботою комп'ютера чи надавати йому вхідну інформацію для обробки.

Клавіатура використовується для введення тексту, символів і команд.

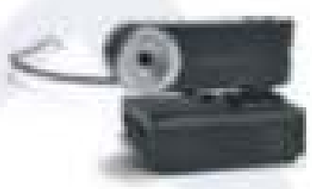
Миша дозволяє переміщати курсор, вибирати об'єкти й виконувати дії на екрані.

Сканер перетворює зображення чи текст у цифровий формат.

Вебкамера створює відео та фотографії.

Мікрофон записує звук або використовується для голосового керування пристроєм.

Графічний планшет дозволяє вводити зображення за допомогою спеціального пера – стилуса.



Усі компоненти всередині корпусу жорсткого диска ретельно закріплені та захищені від зовнішнього середовища.

Маленька частинка пилу з повітря або навіть легкий дотик до голівок може вивести жорсткий диск із ладу.

Пристрої виведення

Такі пристрої дозволяють комп'ютеру передавати результати своєї роботи в зручній для людини формі.

Монітор відображає текст, графіку й відео.

Принтер друкує текст і зображення на папері.

Проєктор демонструє зображення з комп'ютера на екран або стіну.

Навушники чи **колонки** відтворюють звук із комп'ютера.

3D-принтер створює фізичні тривимірні об'єкти на основі цифрових моделей.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Існують і комбіновані пристрої введення / виведення.

Сенсорний екран може приймати вхідні дані від дотиків і відображати вихідну інформацію.

Гарнітура (навушники з мікрофоном) дозволяє вводити голосові команди чи диктувати текст, прослуховувати аудіо.

Ці пристрої підключаються до комп'ютера через різні інтерфейси, як-от USB, HDMI, Bluetooth тощо. Вони дозволяють комп'ютеру отримувати дані від користувача та видавати результати обробки в зручній формі.

Стилус – допоміжний інструмент, використовується для взаємодії із сенсорними екранами пристроїв, як-от планшети, смартфони, графічні планшети, деякі моноблоки.

Драйвери

Пристрої введення / виведення можна підключити до комп'ютера за допомогою різних роз'ємів і кабелів або за допомогою бездротового з'єднання. В комп'ютері має бути бездротовий приймач. Роз'єми на звуковій карті використовують для підключення колонок, навушників і мікрофона.

Найчастіше під час першого підключення нового обладнання комп'ютер автоматично налаштовує його. Але іноді цього не відбувається. Щоб нове обладнання працювало коректно, треба встановити відповідне ПЗ – драйвер. **Драйвери** – це програми, які уможливлюють зв'язок і коректну взаємодію операційної системи з підключеними пристроями та обладнанням. Вони містять команди для операційної системи, зокрема як правильно взаємодіяти з тим чи тим пристроєм, і забезпечують повний функціонал підключеного обладнання. Без належного драйвера пристрій просто не працюватиме.

Правильно налаштовані драйвери забезпечують найкращу продуктивність пристроїв та сприяють уникненню конфліктів з іншими компонентами системи.

Виробники регулярно випускають оновлені версії драйверів із новими можливостями та виправленнями помилок.

Драйвери – невід'ємна частина комп'ютерної системи, яка забезпечує повноцінну роботу підключеного обладнання та дозволяє використовувати весь його функціонал.

Під'єднання різноманітних пристроїв до комп'ютера дозволяє налаштувати ПК під власні потреби та максимально пристосувати обладнання для вашого зручного використання.

Багато пристроїв постачаються з власними утилітами або програмами для налаштування різноманітних параметрів і функцій. Наприклад, для миші можна налаштувати чутливість, прискорення, програмувати додаткові кнопки чи налаштувати кнопки для шульги.

Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки. Опишіть, як дані циркулюють у комп'ютері, як працюють команди в ПК.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 3

Операційні системи

У попередній темі ви ознайомилися з внутрішніми компонентами та пристроями комп'ютера. Усі ці частини утворюють **апаратну частину** комп'ютера. ПЗ складається з команд, які ми даємо комп'ютеру, щоб він виконував певні операції. Набір команд, призначених для виконання певного завдання, називається **програмою**.

Сучасне ПЗ поділяється на дві категорії: прикладне та системне.

- До **прикладного програмного забезпечення** належать усі програми, які призначені для виконання завдань у реальному світі, допомагаючи користувачеві комп'ютера. Більшість програм, які ви використовуєте на своєму комп'ютері, як-от текстові редактори, браузери, ігри чи медіаплеєри, – це прикладне ПЗ.
- **Системне програмне забезпечення** керує самою комп'ютерною системою. Воно надає інструменти й середовище, у якому можна створювати та запускати прикладне ПЗ. Системне ПЗ взаємодіє безпосередньо з апаратним забезпеченням і надає більше функціональних можливостей, ніж саме апаратне забезпечення.

Операційна система *комп'ютера (ОС) – це основа його системного ПЗ. Вона керує ресурсами комп'ютера, як-от пам'ять і пристрої введення / виведення, надає ПЗ доступ до системних ресурсів і забезпечує взаємодію користувача з комп'ютерною системою.*

Утиліта *– це програма, призначена для налаштування, обслуговування та оптимізації комп'ютерної системи.*

Апаратне забезпечення	Системне програмне забезпечення	Прикладне програмне забезпечення
Центральний процесор Диск Принтер Миша	Операційна система Драйвери Утиліти	Браузери Бази даних Текстові процесори Електронні таблиці Комп'ютерні ігри
	Windows	W

ІСТОРІЯ

Назва Windows на початку розробки операційної системи розглядалася як тимчасова, адже остаточну версію ОС планували назвати Spectrum. Але залишили Windows, бо це відображало концепцію ОС з графічним інтерфейсом користувача, яка надає вікна у світ програм і файлів.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Коли запускається програма, вона завантажується в оперативну пам'ять, а її команди обробляються процесором одна за одною.

Усі комп'ютери підтримують **багатозадачність**, яка полягає в тому, що в оперативній пам'яті одночасно зберігається декілька програм. Ці програми конкурують за доступ до центрального процесора, щоб бути виконаними. Отже, завданням операційної системи є управління пам'яттю, щоб відстежувати, які програми є в пам'яті та де вони розташовані.

Операційна система також **управляє процесами**. **Процес** – програма, яка виконується в цей момент. Оскільки в центральному процесорі багато активних процесів хочуть виконати свої сценарії одночасно, операційна система повинна відстежувати хід виконання кожного з них і ретельно управляти ними, щоб вони продовжували роботу з того місця, на якому зупинилися, по черзі використовуючи процесор.

ІСТОРІЯ

Майже всі відомі операційні системи, що використовуються сьогодні, засновані на **UNIX**, операційній системі, яка вперше була представлена в 1969 році.

Управління пам'яттю

Операційна система повинна:

- відстежувати, у якій ділянці пам'яті розташована програма та як програма її використовує;
- перетворювати логічні адреси програм у фактичні адреси пам'яті.

Операційна система використовує оперативну пам'ять як безперервний простір для зберігання даних. Його поділено на групи бітів, що містять команди чи дані. Кожна із цих частин має бути однозначно ідентифікована, тому їй присвоюється **адреса**. Адреси – це цілі числа, що починаються з 0. Відповідно 0 є першою адресою пам'яті.

Проблема полягає в тому, що програма не знає заздалегідь, які адреси в пам'яті будуть їй виділені. Тому звертається до своїх команд і даних, використовуючи відносні адреси, які називаються **логічними адресами**. Завдання операційної системи – зіставлення логічних адрес програми з відповідними фізичними адресами. Вони – фактичні адреси оперативної пам'яті, після того як програма буде завантажена в пам'ять. Цей процес називається **прив'язкою адрес**.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Управління процесами

Операційна система повинна також контролювати, як процеси використовують ресурси процесора. Кожен процес, отримуючи та втрачаючи контроль над ЦП, проходить певний цикл, під час якого перебуває в різних станах: вхід у систему, готовність, виконання, очікування необхідного ресурсу чи завершення.

Створюється **новий процес**. Це відбувається під час запуску програми. Наприклад, коли ви відкриваєте браузер, щоб увійти в інтернет, або відкриваєте Word для роботи з текстом.

Процес у стані **виконання** опрацьовується ЦП. Саме тоді одна або декілька команд процесу проходять цикл **вибірка – декодування – виконання**. Але триває це недовго, тому ЦП чекають й інші процеси.

Завершення – це процес, який уже виконаний і більше не є активним. Операційній системі більше не потрібно зберігати інформацію щодо цього процесу, і вона очищає пам'ять, яку більше не використовує.

Процес перебуває в стані **готовності**, коли для його виконання все готове, йому потрібен лише ЦП. Інші процеси можуть бути не готові до виконання, оскільки вони очікують на певну подію, наприклад, їм може знадобитися отримати дані із зовнішнього накопичувача, перш ніж продовжити роботу. Такі процеси не належать до черги готовності.

Процес у стані **очікування** чекає на ресурси (окрім ЦП). Так, процес може очікувати на отримання необхідних даних із жорсткого диска й завантаження їх у пам'ять або на введення даних користувачем за допомогою пристрою введення, наприклад він надрукує текст на клавіатурі.

ПІДКАЗКА

Зверніть увагу, що одночасно багато процесів можуть перебувати в стані готовності або в стані очікування, але в стані виконання може перебувати тільки один процес. Черги готовності та очікування існують для керування порядком виконання процесів, що дозволяє операційній системі ефективно розподіляти ресурси.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Файлова система

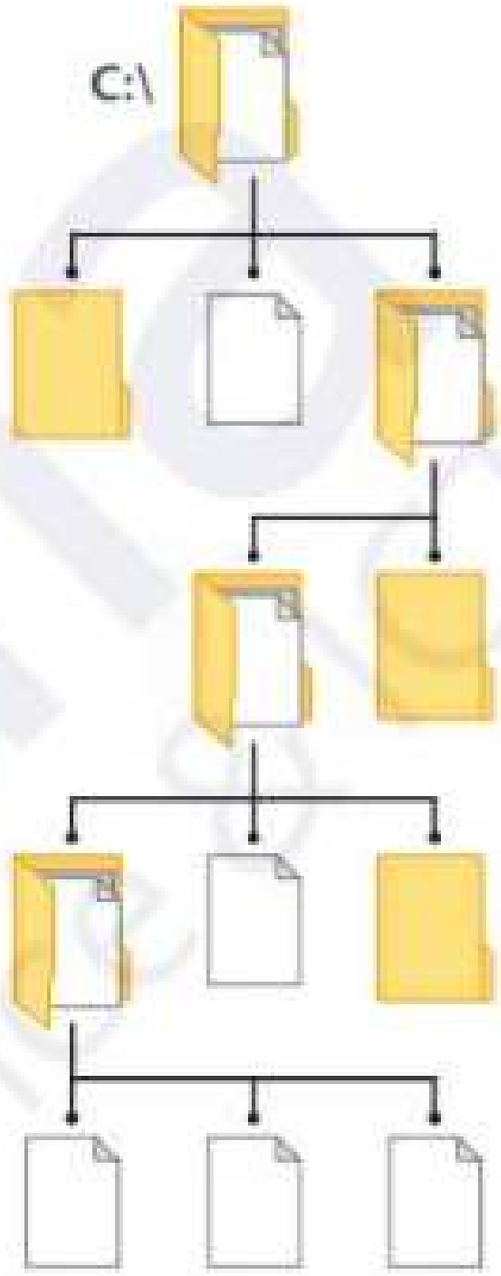
Впорядкування зовнішньої пам'яті, як-от жорстких дисків, також є однією з найважливіших функцій операційної системи. Пам'ятайте, що накопичувачі енергонезалежні, саме там зберігаються всі наші програми й дані, коли комп'ютер вимкнено.

Інформація на жорсткому диску структурована й зберігається у файлах. **Файл** – це впорядкований набір даних, який має назву. Він є основною організаційною одиницею жорсткого диска. Файл може містити програму або дані різних типів. Наприклад, і файл браузера, і цифрове зображення – файли на жорсткому диску.

Файлова система – це логічний спосіб подання даних у вигляді набору файлів, який надає операційна система користувачу. Файлова система часто організована групуванням файлів у каталоги чи теки. **Каталог / тека** – це іменований набір файлів.

Тека може містити іншу теку. Каталог, що містить інший такий перелік, називають батьківським, а той, що розміщений усередині, – підкаталогом. Ви можете створити стільки вкладених тек, скільки вам потрібно для впорядкування даних. Звичайно, одна тека може містити декілька підкаталогів, а кожен із них може містити власні підкаталоги, створюючи **ієрархічну структуру**. Файлову систему можна розглядати як дерево каталогів, яке показує теки та файли в інших каталогах.

Кореневий каталог – *тека найвищого рівня.*



Види файлової системи

FAT (File Allocation Table)

Найстаріша файлова система. Розроблена компанією Microsoft, використовувалася в ранніх версіях Windows. Існує декілька версій FAT: FAT12, FAT16, FAT32, exFAT. Число в абревіатурі вказує розмір елемента таблиці в бітах. Одиницею розподіленої пам'яті є кластер.

NTFS (New Technology File System)

Сучасна файлова система Windows. Забезпечує кращу безпеку та ефективність.

ext4 (Fourth Extended File System)

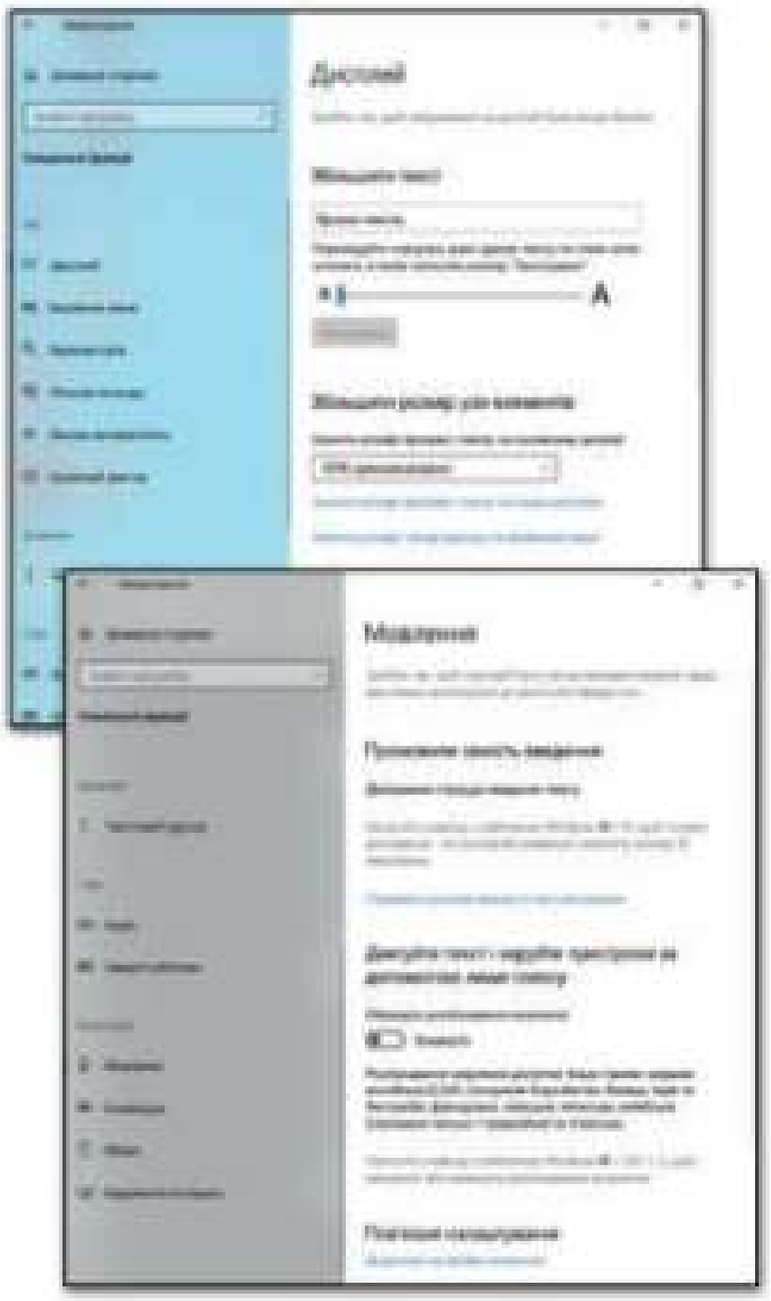
Одна з найпопулярніших файлових систем для Linux.

HFS+ (Hierarchical File System Plus)

Файлова система для MacOS.

Файлова система є невід'ємною частиною операційної системи та відповідає за ефективне зберігання й організацію даних на цифрових пристроях.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.



Налаштування пристроїв

Операційні системи мають вбудовані засоби для базової конфігурації пристроїв. Тут можна регулювати роздільну здатність екрана, частоту оновлення, гучність звуку тощо.

Для окремих завдань існують додаткові програми, що розширюють можливості пристроїв. Наприклад, для ігрової миші або клавіатури є програми, які дозволяють налаштовувати комбінації клавіш, змінювати підсвічування та створювати профілі.

Правильна конфігурація дозволить пристосувати пристрої для максимальної продуктивності, зручності та задоволення ваших індивідуальних потреб.

Наприклад, у **Налаштуваннях** операційної системи на вкладці **Спеціальні функції** можна налаштувати комп'ютер для людей з особливими потребами, які мають вади зору чи слуху. За допомогою таких налаштувань комп'ютером можна управляти не тільки голосом, а й поглядом.

Оновлення ПЗ

Оновлення ПЗ – процес внесення змін до наявної версії програми. Оновлення зазвичай покращують продуктивність або функціональність, виправляють помилки, додають нові можливості, посилюють захист.

Оновлення ПЗ

Безпека. Оновлення часто містять виправлення вразливостей безпеки, щоб захистити користувачів від зловмисного ПЗ та хакерських атак.

Усунення помилок. Розробники виправляють виявлені помилки або баги в програмах для забезпечення стабільної роботи.

Нові функції. Оновлення можуть додавати нові функції, інструменти чи можливості, які роблять програму більш корисною та зручною.

Підтримка. Оновлення забезпечують сумісність програми з новими ОС, апаратним забезпеченням або іншими програмами.

Автоматичні оновлення. Багато програм можуть автоматично перевіряти наявність оновлень та встановлювати їх після схвалення користувача.

Ручне оновлення. В деяких випадках необхідно завантажувати та встановлювати оновлення вручну з вебсайту розробника.

Пакетні оновлення. ОС та деякі програми можуть надавати регулярні пакети оновлень, що містять виправлення та покращення.

Регулярне оновлення ПЗ важливе для забезпечення стабільності, безпеки та доступу до нових функцій. Проте завжди ретельно перевіряйте джерело оновлень та їх вплив на систему перед встановленням.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Облікові записи користувачів

Щоб додатково убезпечити комп'ютер і дані, варто створити окремі облікові записи для адміністратора вашого пристрою та кожного з користувачів. Це дуже важливо, якщо комп'ютер використовують кілька людей.

Крім захисту від несанкційованого встановлення ПЗ, окремий запис дозволить налаштувати ресурси для кожного користувача, власний вигляд робочого столу, профілі для роботи з різними пристроями, доступ тільки до особистих файлів і даних.

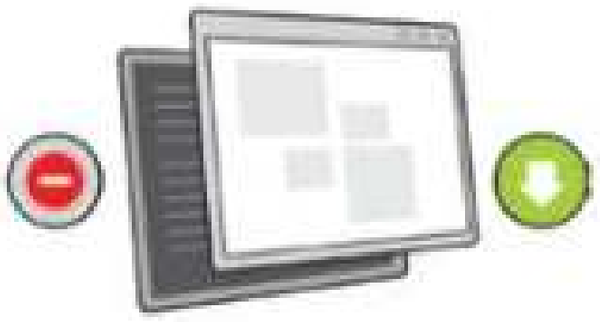
Належне управління обліковими записами користувачів є важливим для забезпечення конфіденційності, організації доступу в комп'ютерних системах, а також безпеки.

Ліцензія на програмне забезпечення

Ліцензія на ПЗ – термін, що описує юридичні права, які користувачі мають на використання, редагування та розповсюдження ПЗ. Ці права визначаються розробником оригінального ПЗ. Користувач має законне право використовувати копію ПЗ. Ліцензія призначена лише для використання конкретної копії ПЗ, а не самого ПЗ. Всі типи ліцензій захищені законом про авторське право.

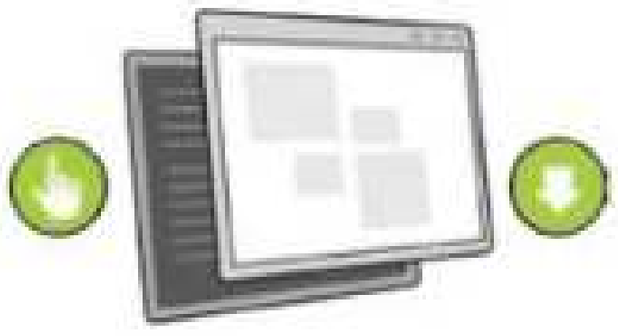
Існує дві основні категорії ліцензій на ПЗ: вільні та пропрієтарні ліцензії на ПЗ. Різниця між категоріями ліцензій – у правах кінцевого користувача. Вільна ліцензія означає, що ПЗ є безоплатне. Існує три підкатегорії вільного ПЗ: безоплатне ПЗ, умовно-безкоштовне та ПЗ з відкритим вихідним кодом.

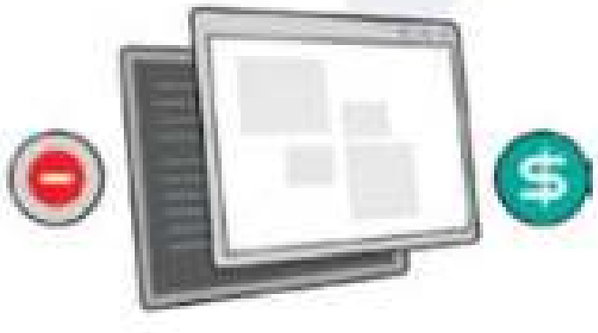
Ліцензія на безоплатне ПЗ дозволяє користувачам завантажувати й використовувати його безплатно. Вихідний код змінювати заборонено.



Ліцензія умовно-безкоштовного ПЗ дозволяє користуватися програмою безплатно протягом короткого періоду часу, а коли пробний період закінчується, то, щоб продовжити використання, треба її купити. У цьому випадку вихідний код змінювати не дозволяється.

Ліцензія з відкритим вихідним кодом є не тільки вільним ПЗ, але й робить вихідний код доступним для всіх, хто хоче вивчати, копіювати, змінювати й поширювати його. ПЗ з відкритим кодом здебільшого розробляється спільно та публічно.





Пропрієтарне ПЗ належить виключно компанії, яка його розробила. Для того щоб користуватися таким ПЗ, потрібно придбати ліцензію. Зверніть увагу, що ви купуєте не ПЗ, а дозвіл на його використання. Копіювання, розповсюдження, модифікація та перебудова ПЗ заборонені.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Ліцензійний ключ

Коли купуєте ПЗ, як-от Windows, Office, ігри тощо, то для активації своєї покупки ви отримуєте унікальну послідовність символів електронним листом або SMS, на скретч-карті чи USB-накопичувачі. Ліцензійний ключ не тільки захищає права розробників ПЗ, а й надає можливість отримання повного доступу до усіх функцій придбаної програми.

Ліцензійний ключ – унікальна послідовність символів (букв і/або чисел), яка використовується для активації чи розблокування ПЗ.

Види ліцензійних ключів

Апаратний ключ – невеликий пристрій, що під'єднується до комп'ютера через USB або паралельний порт. Програма перевіряє наявність цього ключа для роботи.

Код продукту – унікальний набір символів (букв та цифр), який вводиться під час активації програми.

Онлайн-активація – процес активації програми через інтернет на серверах розробника.

Ключі на основі обладнання – ліцензії, прив'язані до певних характеристик комп'ютера, наприклад серійного номера материнської плати.

Ключі оренди – придбання ліцензії на визначений термін користування програмою.

Випадкові ключі – генеруються випадковим чином для кожної встановленої копії програми.

Підписка – потрібно періодично поновлювати / оплачувати ліцензію для продовження використання.

Початкові ключі – тимчасова ліцензія для оцінки програми перед придбанням.

Функції ліцензійного ключа	
Активація програми	Після встановлення програми потрібно ввести ліцензійний ключ, щоб повністю активувати її функції та можливості.
Підтвердження ліцензії	Ключ підтверджує права на використання певної програми.
Уникнення піратства	Розробники ПЗ використовують ліцензійні ключі, щоб запобігти нелегальному його поширенню та використанню.
Відстеження користувачів	Кожен ключ унікальний, тому розробник може відстежити, кому він був виданий.
Різні функції	Деякі ліцензійні ключі розблоковують додаткові функції або видаляють обмеження в програмі.
Строк дії	Інколи ключі діють протягом певного часу, після чого потрібно придбати нову ліцензію.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Автозавантаження

Автозавантаження чи **автозапуск** – функція ОС, яка дозволяє певним програмам запускатися автоматично під час завантаження комп'ютера чи іншого цифрового пристрою.

Особливості автозавантаження

Допоміжні програми
Користувачі можуть додавати до автозапуску різноманітні допоміжні програми на свій розсуд, як-от месенджери (WhatsApp, Viber тощо), програми для синхронізації даних (OneDrive, Google Drive тощо) та інші програми за потреби.

Економія часу
Програми, які потрібні користувачеві відразу після запуску ОС, запускаються автоматично, без втручання користувача.

Системні програми
Деякі системні утиліти, антивірусні програми, оновлення та інші часто додаються до автозавантаження виробниками ОС або комп'ютерів.

Зловмисне ПЗ
Віруси та шкідливі програми можуть потрапляти до автозавантаження без відома користувача.

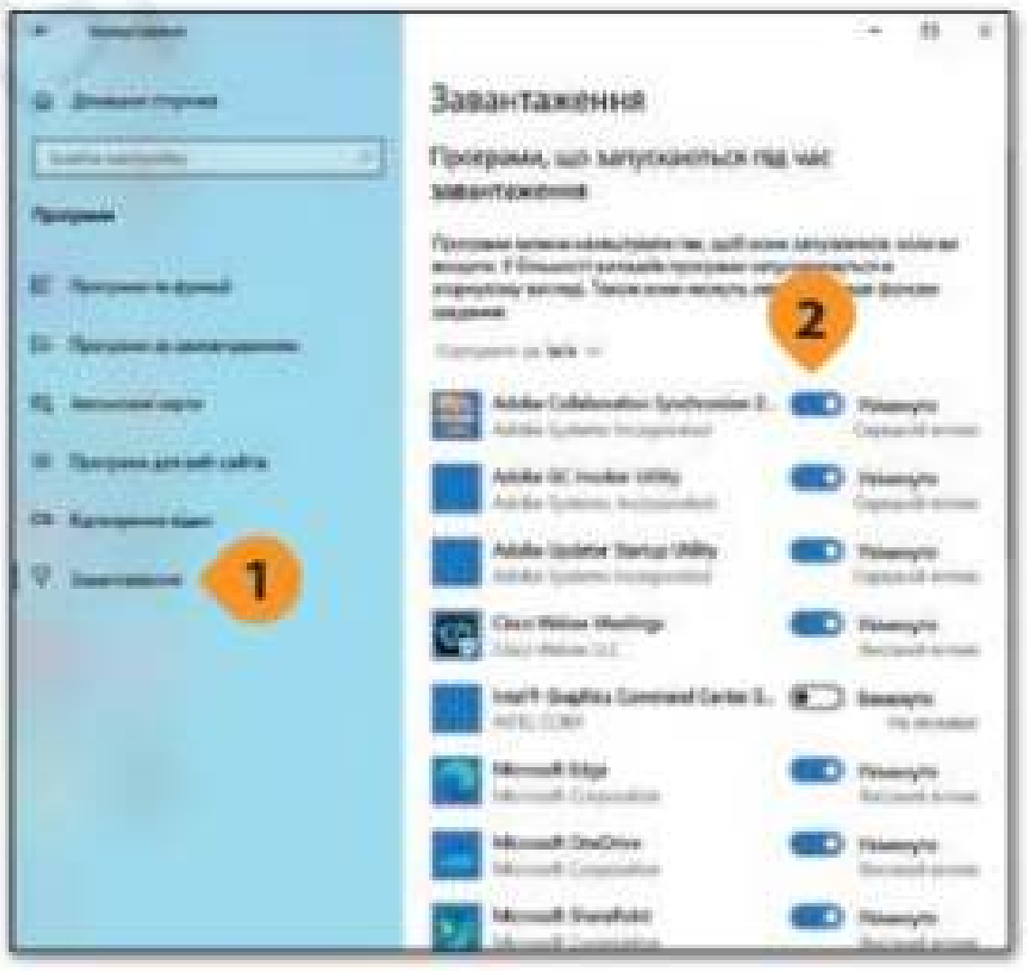
Керування автозавантаженням
Сучасні ОС мають вбудовані інструменти для перегляду та керування списком програм автозапуску.

Оптимізація продуктивності
Надмірна кількість програм в автозапуску може сповільнювати завантаження системи, тому потрібно обережно керувати цим списком.

Розумне використання автозавантаження може заощадити час і зусилля користувачів, але потрібно стежити, щоб зайві програми не перевантажували систему.

Щоб налаштувати Автозавантаження:

- натисніть **Пуск**;
- клацніть **Налаштування**;
- виберіть **Програми**;
- натисніть **Завантаження** (1);
- у вікні, що відкриється, передивіться перелік програм (2) та вимкніть ті, які вам непотрібні, увімкніть потрібні для вас.



ПІДКАЗКА

Перед оновленням системи обов'язково зробіть резервне копіювання. Це допоможе повернути систему в попередній стан, якщо щось піде не так.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Сумісність програмного забезпечення

Коли ви шукаєте програму, яку плануєте завантажити, на сайті майже завжди пропонують завантажити її для різних ОС та цифрових пристроїв. Це означає, що програма може нормально працювати на різних пристроях та ОС.

Сумісність ПЗ – здатність програми коректно працювати на різних апаратних платформах, ОС або у взаємодії з іншими програмами та пристроями.

Апаратна сумісність – програма має бути сумісною з різними типами процесорів, відеокарт, пристроїв введення / виведення тощо. Це забезпечує її роботу на різних комп'ютерах.

Сумісність з ОС – програма повинна правильно функціонувати під управлінням різних ОС, наприклад Windows, macOS, Linux. Версії ОС також можуть впливати на сумісність.

Сумісність форматів даних – програми мають підтримувати стандартні формати файлів (документів, зображень, відео тощо) для обміну даними між собою.

Програмна сумісність – при взаємодії різних програм вони мають дотримуватися певних протоколів та інтерфейсів для успішної інтеграції та уникнення конфліктів.

Зворотна сумісність – нові версії програм повинні зберігати сумісність із попередніми версіями, щоб користувачі могли оновлювати ПЗ.

Вебсумісність – вебзастосунки та сайти мають бути сумісними з різними веббраузерами та їхніми версіями.

Забезпечення сумісності є важливим завданням для розробників ПЗ, адже це дозволяє користувачам безперешкодно працювати з різними комбінаціями апаратного та програмного забезпечення.

Користувач має можливість обрати інсталяційний файл відповідно до свого пристрою та ОС.

Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки. Опишіть, як працює ОС, за що відповідає, яка файлова система у вас на ПК.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 4

Збереження даних

У процесі користування цифровими пристроями ви робите фото, пишете повідомлення, нотатки, створюєте різноманітні файли, відправляєте їх друзям та подругам. Ваші дані зберігаються на жорсткому диску чи карті пам'яті, або ви одразу відправляєте їх у хмарне сховище. Незалежно від того, де ви зберігаєте дані, питання їх збереження завжди актуальне.

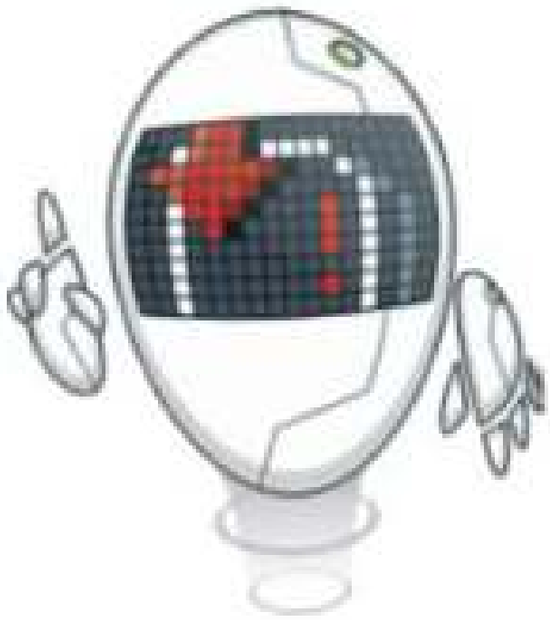
Збереження даних – це процес забезпечення цілісності інформації в комп'ютерних системах. Належне збереження даних є критично важливим для безпеки, цілісності та доступності інформації в персональних комп'ютерах, корпоративних системах і вебсервісах.

Пристрої зберігання даних

Жорсткі диски (HDD) та твердотільні накопичувачі (SSD)	USB-накопичувачі	Хмарні сховища даних

Диски – основні пристрої зберігання даних у комп'ютерах. Існують різні типи дисків.

Вибір типу диска залежить від потреб користувача: ємності зберігання, швидкості доступу, мобільності, надійності й вартості. HDD і SSD – найбільш поширені жорсткі диски для зберігання операційних систем, програм і персональних даних на сучасних комп'ютерах.



Типи дисків

Жорсткі диски
(HDD – Hard Disk Drive):

- для запису даних використовують магнітні пластини;
- мають високу ємність зберігання (зазвичай від 500 ГБ до кількох ТБ);
- широко використовуються та відносно недорогі;
- механічна конструкція робить їх більш уразливими до пошкоджень від ударів чи вібрацій.

Твердотільні накопичувачі
(SSD – Solid State Drive):

- використовують флеш-пам'ять для зберігання даних;
- не мають рухомих частин, що робить їх більш надійними та енергоефективними;
- забезпечують високу швидкість читання / запису даних;
- зазвичай дорожчі за HDD із такою самою ємністю.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Підготовка дисків до роботи

Новий диск для зберігання даних потрібно підготувати до використання.

Підготовка дисків до роботи:

1 Форматування диска:

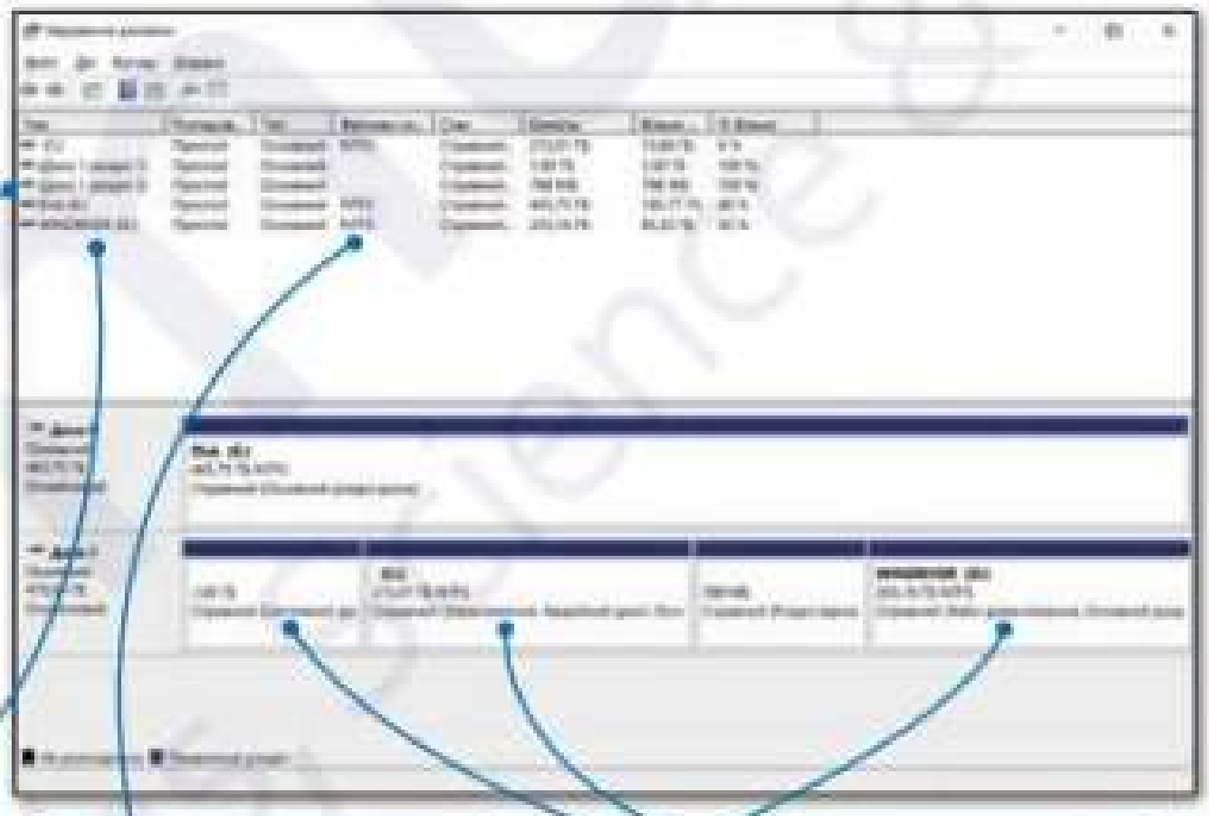
відкрийте **Файловий провідник** > оберіть **диск D:** або **флеш-накопичувач** > клацніть правою кнопкою миші > у вигульковому меню натисніть **Форматувати**.

Створює відповідну файлову систему (NTFS, FAT32, exFAT, ext4 тощо) для організації даних на диску

Очищає весь вміст диска під час форматування

2 Створення розділів:

на панелі **Пуск** натисніть праву кнопку миші > у вигульковому меню оберіть **Керування дисками**.



Розбиття диска на логічні розділи для різних цілей використання

Диски та розділи отримують літерні позначення (C:, D:, E: тощо), що полегшує ідентифікацію та доступ до пристроїв зберігання

Кожен розділ може мати власну файлову систему

На одному фізичному диску можна створити кілька розділів

3 Перевірка диска на наявність помилок:

- використання вбудованих утиліт для сканування диска на наявність фізичних дефектів або помилок файлової системи;
- виявлені помилки можна спробувати виправити.

4 Налаштування параметрів диска:

- визначення обсягу кластера (розміру найменшої одиниці зберігання);
- вибір можливості стиснення даних або шифрування для підвищення безпеки.

5 Створення томів (дисків) із декількох фізичних дисків (необов'язково):

- технологія RAID дозволяє об'єднати кілька дисків в один логічний том;
- підвищує ефективність, надійність або ємність зберігання даних.

Правильна підготовка дисків забезпечує їх оптимальне використання, підвищує продуктивність, допомагає уникнути помилок і проблем із файловими системами.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Фізичні та логічні пошкодження дисків

Диски, як і будь-яка інша апаратна частина комп'ютера, можуть пошкоджуватися. Є два основних типи пошкоджень дисків.

Фізичні (апаратні) пошкодження	• механічні пошкодження через удари, падіння або сильну вібрацію (для традиційних жорстких дисків); • пошкодження головок запису / зчитування або поверхні магнітних пластин; • збої в електроніці або схемах управління диском; • зношування або старіння компонентів диска із часом; • фізичні пошкодження можуть призвести до повної втрати даних і виходу з ладу диска диска.
Логічні (програмні) пошкодження	• пошкодження файлової системи або таблиць розміщення файлів; • порушення цілісності даних через збої ПЗ або ОС; • неправильне відключення диска без виконання пункту безпечного вилучення пристрою; • вірусні або шкідливі програми, що змінюють чи пошкоджують дані; • логічні пошкодження можуть призвести до втрати доступу до даних або неможливості запуску системи.

Використовуйте джерело безперебійного живлення (ДБЖ) для захисту дисків від стрибків напруги.

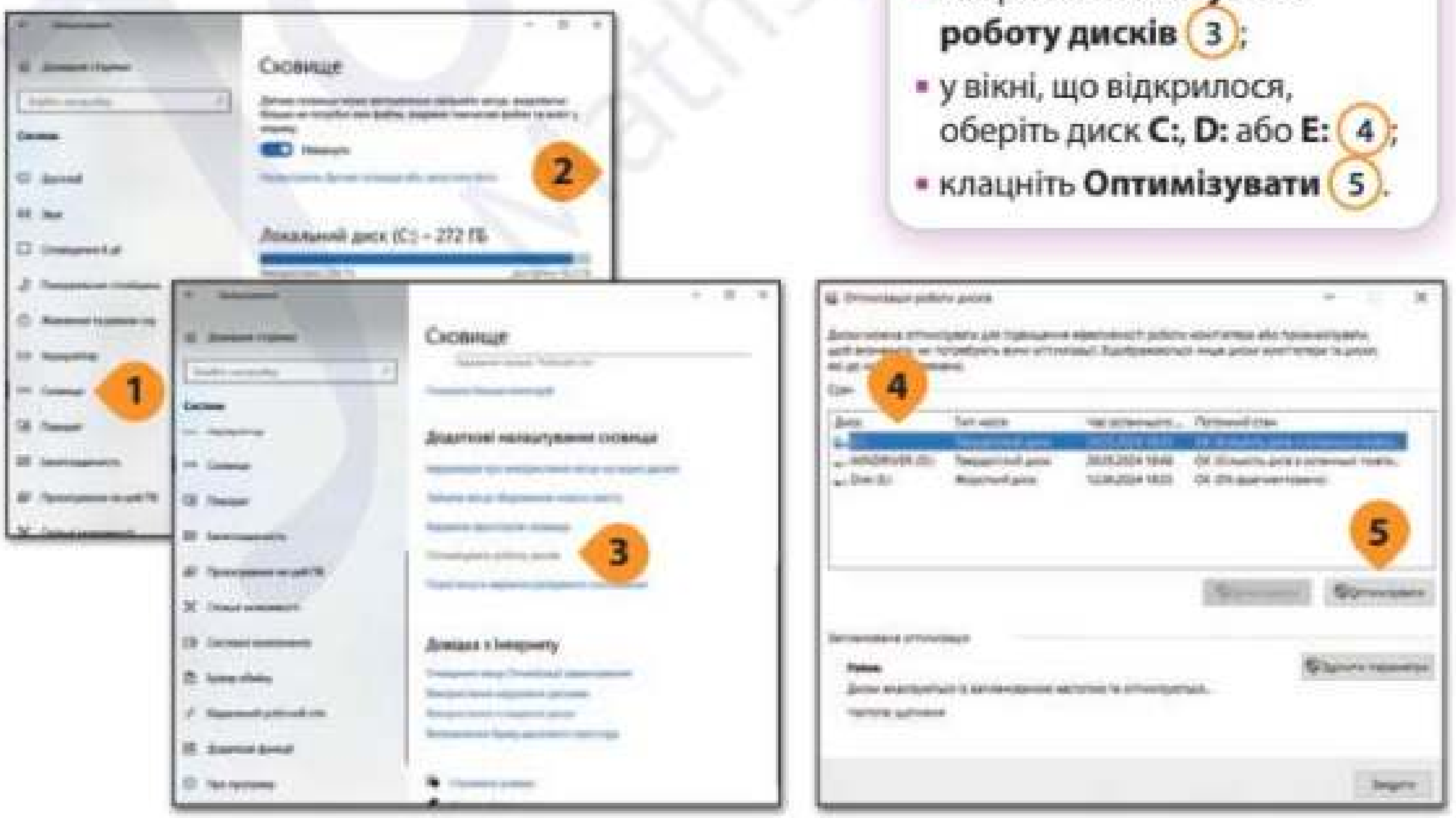
Не записуйте нові дані на диск, якщо необхідно відновити старі файли.

Якщо диск почав видавати дивні звуки, краще відразу його вимкнути й звернутися до фахівців.

Своєчасне технічне обслуговування та обережне поводження з дисками допомагають мінімізувати ризик їх пошкодження.

Щоб оптимізувати роботу дисків:

- натисніть **Пуск**;
- клацніть **Налаштування**;
- оберіть вкладку **Система**;
- натисніть **Сховище** (1);
- прокрутить сторінку вниз (2);
- оберіть **Оптимізувати роботу дисків** (3);
- у вікні, що відкрилося, оберіть диск **C:**, **D:** або **E:** (4);
- клацніть **Оптимізувати** (5).

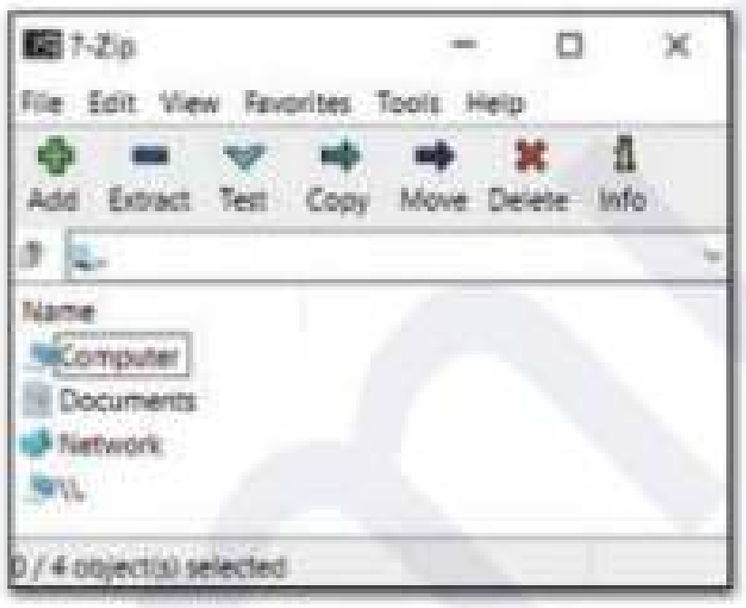


© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Резервне копіювання (або бекап) – важливий процес для збереження даних і захисту від їх втрати. Мета – створити резервні копії файлів, папок, системних налаштувань чи дисків для можливості відновлення в разі збоїв, пошкоджень, вірусних атак або випадкового видалення. У разі виникнення проблем можна спробувати відновити дані за допомогою спеціалізованих утиліт або звернутися до сервісних служб відновлення даних.

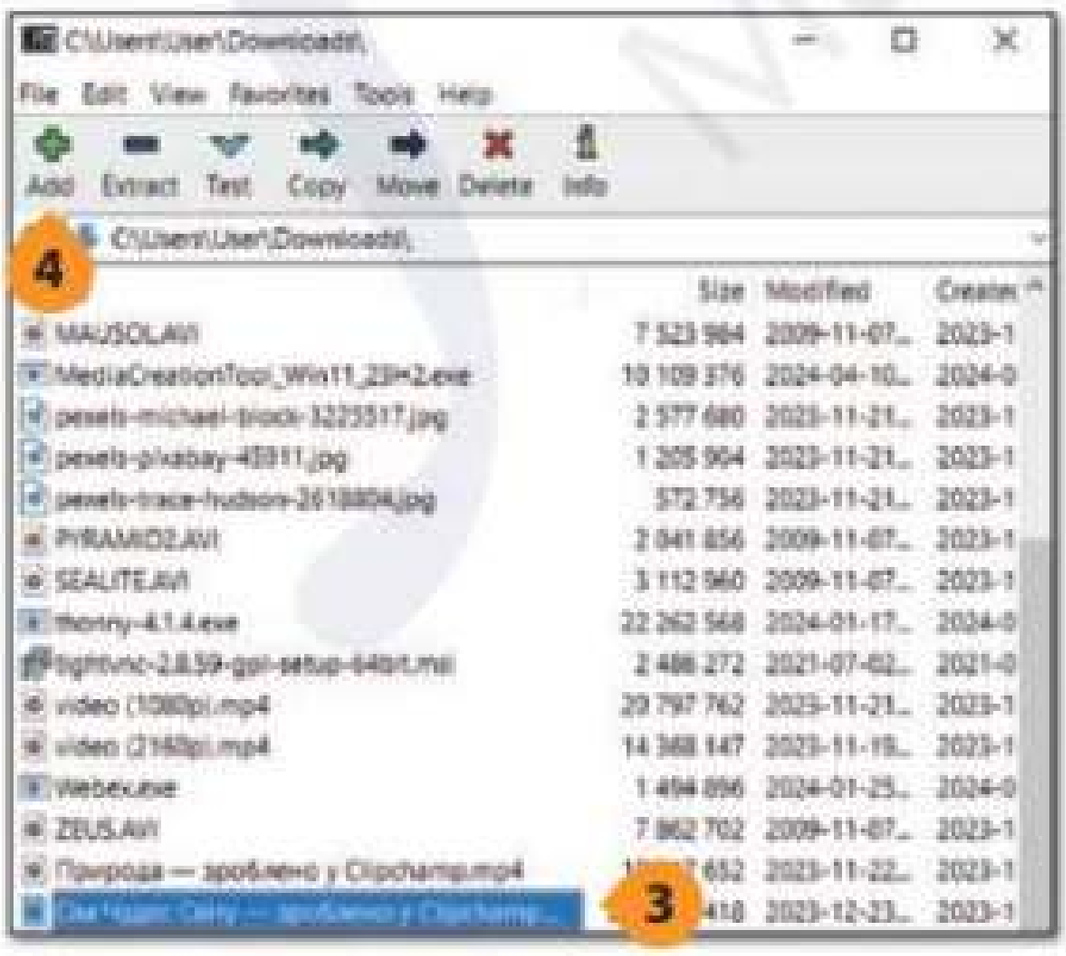
Архівування

Крім своєчасного резервного копіювання, зберегти дані можна завдяки архівуванню. Цей процес допомагає зменшити розмір файлів і заощадити місце для зберігання, а також оптимізувати передачу даних. Для стиснення файлів можна скористатися архіватором, як-от WinRAR, 7-Zip, Universal Extractor, WinZip тощо.



Щоб створити архів:

- у меню **Пуск** відкрийте теку **7-Zip** (1);
- клацніть **7-Zip File Manager** (2);
- оберіть місце збереження файлів, які плануєте заархівувати;
- виділіть файл, який будете архівувати (3);
- натисніть Add (4), щоб файл додати в архів;
- за бажанням змініть назву архіву (5) та місце зберігання (6);
- клацніть **OK** (7).



© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Стиснення даних часто використовується разом з архівуванням, що дозволяє об'єднати кілька файлів в один архівний файл для зручності передачі та зберігання.

Архівні формати, як-от ZIP, RAR, зазвичай включають функції стиснення для зменшення загального розміру архіву.

Існують різні алгоритми стиснення, зокрема DEFLATE (використовується в ZIP і gzip), LZMA (використовується в 7z), Brotli та інші. Вибір алгоритму залежить від типу даних, необхідного рівня стиснення та швидкості обробки. Розрізняють алгоритми стиснення без втрат, як-от ZIP, GZIP, LZW, та стиснення з втратами, наприклад JPEG для зображень, MP3 для аудіо.

Для текстових даних частіше використовується алгоритм Хаффмана, а для мультимедійних даних – алгоритми на основі перетворень.

Для подолання цих проблем організації даних користувачі можуть впроваджувати різні стратегії, як-от стиснення даних, архівування, хмарні рішення для зберігання, резервне копіювання тощо.

Алгоритм Д. Хаффмана та алгоритм Шеннона-Фано передбачають перекодування даних. Обидва ці алгоритми використовуються для стиснення даних без втрат і базуються на принципах теорії інформації. Символи, що частіше трапляються в повідомленні, перекодовуються меншою кількістю символів, а які рідше – більшою.

Алгоритм Хаффмана будує дерево «знизу вгору», починаючи з найменш частих символів, тоді як алгоритм Шеннона-Фано працює «зверху вниз», розділяючи символи на групи.

У стиснутому таким способом файлі міститься таблиця кодування та додаткові службові відомості. Алгоритм Хаффмана зазвичай дає кращі результати стиснення, тому він частіше використовується на практиці.

Алгоритм RLE (англ. Run-Length encoding – кодування повторів або довжин серій) використовується частіше для кодування графічних даних. Цей метод передбачає заміну послідовності однотипних даних її скороченим описом.

Стиснення даних є важливим інструментом ефективного зберігання та передачі даних, дозволяючи економити місце для зберігання та пропускну здатність мережі. Але важливо враховувати компроміс між ступенем стиснення, швидкістю обробки та можливою втратою даних для конкретного типу даних і вимог до застосунку.

Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 5

Захист цифрових пристроїв та інформації

Інтернет надає користувачам можливість доступу до невичерпного джерела інформації. Цей обсяг інформації зберігається на мільйонах комп'ютерів по всьому світу та постійно поповнюється всіма користувачами, які вирішили завантажити дані. Невпорядкованість інтернету (брак чіткої структури та контролю) створює для користувачів багато онлайн-небезпек.

Шкідливе ПЗ – це загальна категорія програм, метою яких є порушення роботи комп'ютера, збір конфіденційної інформації або отримання доступу до приватних комп'ютерних систем.

Віруси

Найпоширенішою та найнебезпечнішою загрозою для комп'ютера є **вірус**. Це програма, здатна до саморозмноження, що виконує несанкціоновані дії. Його мета – перешкоджати нормальній роботі комп'ютера. Вірус може поширюватися через комп'ютерну мережу, заражені пристрої зберігання даних, електронну пошту або через відвідування зараженого вебсайту чи реклами. Він може видаляти чи пошкоджувати файли, змінювати або вимикати основні функції, шифрувати дані тощо. Вірус зазвичай ховається в EXE-файлах, і, коли ви запускаєте їх, він починає поширюватися, копіюючи сам себе.

Коли мандруєте в інтернеті, пам'ятайте:

1	Інтернет – спільнота людей, яка приховує такі самі небезпеки, як і кожна спільнота.
2	Інформація, представлена в інтернеті, не завжди є правдивою.
3	Уникайте обміну особистою інформацією (ім'я, адреса, номер телефону, фотографія, паролі, номер банківської картки, електронна пошта тощо), особливо коли вона стосується інших людей.

Сучасні віруси намагаються знайти «лазівку» в мережі, щоб заразити її. Їх можуть використовувати для руйнування інфраструктури, наприклад дамб та електростанцій, або для пошкодження обладнання в лікарнях. Віруси створюють люди з глибокими знаннями в галузі комп'ютерного програмування та мережевих технологій. Вони можуть блокувати доступ до даних користувачів та вимагати оплату за їх розблокування чи розшифрування.

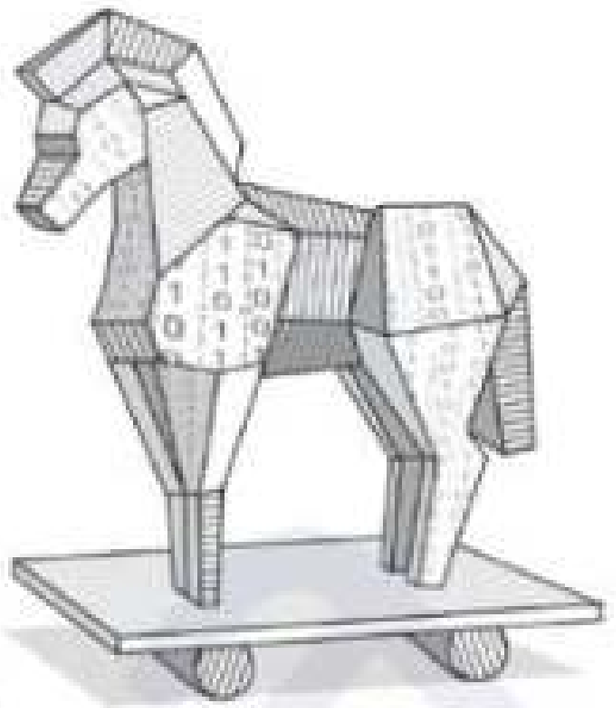
ДОВІДКА

DoS (Denial of Service) – це атака на сервер, сайт або мережу, яка перевантажує систему великою кількістю запитів. Унаслідок цього ресурс стає недоступним для користувачів.

DDoS (Distributed Denial of Service) – це розподілена DoS-атака, коли напад здійснюється з великої кількості пристроїв одночасно (як-от ботнетів). Таку атаку складніше заблокувати, оскільки вона йде з різних IP-адрес.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Троян



Свою назву троян запозичив із давньогрецького міфу про оманливого дерев'яного коня, який став причиною падіння міста Троя. Він маскується під звичайні програми, а після потрапляння в комп'ютер працює у фоновому режимі, створює проблеми з безпекою та надає доступ до системи третім особам. Це може буди як простий моніторинг активності комп'ютера, так і повний його контроль. Троянські програми призначені для крадіжки інформації з комп'ютера. Багато комп'ютерів, заражених троянами, можуть стати частиною ботнету. **Ботнет** (поєднання слів «робот» і «мережа») – це мережа комп'ютерів, які використовуються без відома їхніх власників для розсилки спаму, крадіжки даних або для здійснення DoS- чи DDoS-атаки. Троянські програми здебільшого діють непомітно – користувачі не повинні знати, що комп'ютер заражений, і тому вони не будуть навмисно сповільнювати роботу комп'ютера або порушувати її. Зазвичай трояни можна знайти в піратських програмах та іграх.

Хробаки

Хробак – це вірус, дуже агресивний у способі свого поширення. Він не чекатиме, поки користувач надішле заражений файл, а проникає до списку розсилки й розсилає свою копію всім контактам, маскуючись під звичайний електронний лист. Інші типи хробаків шукають проблеми з безпекою в системах, що працюють під управлінням ОС, і впливають на них, не чекаючи, поки користувач запустить файл. Хробак може завдати шкоди комп'ютеру різними способами: від уповільнення його роботи до повного знищення файлів. Крім зараження файлів, хробаки призначені для зараження систем. Щоб захистити систему від хробаків, не відкривайте підозрілі електронні листи та постійно оновлюйте ПЗ.

Шпигунське програмне забезпечення

Шпигунські програми збирають інформацію про користувачів без їхнього відома. Вони приховані від користувача, тож їх дуже важко виявити. Існує дві категорії таких програм: програми, які можуть копіювати паролі або номери банківських карток і зазвичай поставляються разом із троянами, та моніторингові програми. Такий тип моніторингових програм відстежує вподобання користувачів і продає дані рекламним компаніям. Такі програми зазвичай потрапляють на комп'ютери разом зі встановленням безплатних програм, іноді з'являючись у вигляді панелі інструментів.

Рекламне ПЗ

Рекламне ПЗ показує рекламу без дозволу користувача. Ця реклама може бути у вигляді вигулькового вікна в інтерфейсі програми. Зазвичай рекламне ПЗ супроводжується шпигунськими програмами.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

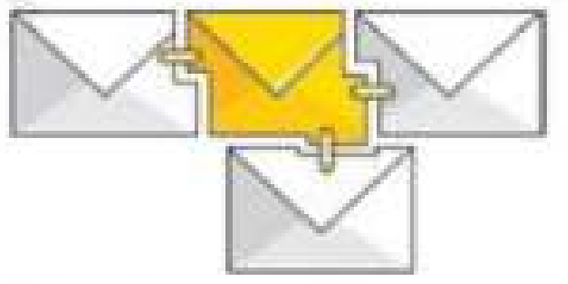
Програми-вимагачі

Програми-вимагачі – тип шкідливого ПЗ, яке тримає систему в «заручниках», вимагаючи від користувача заплатити, щоб користуватися нею далі. Програми-вимагачі можуть з'являтися у вигляді вигулькових вікон, коли користувач працює в інтернеті, інший тип може змусити браузер вимкнутись. Такі програми запускаються в ОС та не дозволяють користуватися комп'ютером, якщо користувач не заплатить. Деякі програми-вимагачі мають такий вигляд, ніби вони походять від наявної організації, та вимагають сплатити фальшивий штраф. Найновіший і найнебезпечніший тип програм-здирників шифрує всі файли та дані й вимагає гроші за ключ до розшифрування.

Фішинг

Фішингові повідомлення намагаються виманити особисту інформацію, як-от імена користувачів, паролі та номери банківських карток. Їх надсилають електронною поштою чи через месенджери та мають на меті обдурити користувачів, щоб викрасти їхні дані. Перехід за посиланням перенаправляє користувача на фальшивий вебсайт, який має вигляд справжнього, і просить користувача надати особисту інформацію. Таким чином хакери отримують доступ до облікових записів, акаунтів у соціальних мережах або банківських рахунків користувача.

Листи щастя



Листи щастя – це електронні листи, які намагаються переконати користувача скопіювати його та надіслати іншим одержувачам. Вміст – сумна історія чи щасливий текст, який потрібно переслати. Його мета – зібрати адреси електронної пошти, які можуть бути використані для розсилки спаму.

Перехоплення трафіку

Коли користуєтеся безкоштовним Wi-Fi чи відкритою Wi-Fi-мережею, будьте обережними. В таких мережах дуже часто можна стикнутися з перехопленням сигналів або прослуховуванням бездротових мереж. Завдяки таким діям зловмисники можуть несанкційовано переглядати ваше листування в месенджерах чи поштою, якщо дані незашифровані, то можуть отримати логіни та паролі, дані банківських карток або просто відслідковувати активність вашого пристрою.

БЕРЕЖІТЬ СЕБЕ

Правильна постава за робочим столом, вправи для очей, обмежена тривалість роботи за комп'ютером – запорука вашого здоров'я! «Mens sana in corpore sano», як говорили давньоримські та грецькі філософи, – здоровий дух у здоровому тілі.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Захист від вірусів

Щоб захистити свій комп'ютер, **ОБОВ'ЯЗКОВО** потрібно мати антивірусну програму. Її можна купити або завантажити безоплатно. Антивірусна програма постійно перевіряє наявність шкідливих програм. Щодня з'являється нове шкідливе ПЗ, тому у вас повинна бути завжди актуальна версія. Налаштуйте автоматичне оновлення антивірусної програми. Складіть розклад, коли буде проводитися сканування вашого пристрою, щоб убезпечити його та ваші дані.

Поради, щоб захистити себе

1	Не розголошуйте логіни та паролі користувачів.
2	Не відкривайте листи від невідомих відправників із дивними темами чи без них.
3	Створюйте надійні паролі та часто змінюйте їх.
4	Не використовуйте один і той самий пароль для всіх своїх акаунтів.
5	Не відповідайте на електронні листи та не відвідуйте підозрілі вебсайти, які запитують ваш пароль, дані платіжних карток тощо.
6	Постійно оновлюйте свій антивірус.
7	Не переходьте за гіперпосиланнями з невідомих електронних листів або вебсторінок.

Брандмауер

Брандмауер, також відомий як мережевий екран або файрвол, – система безпеки мережі, яка контролює вхідний і вихідний мережевий трафік на основі встановлених правил безпеки, аналізуючи пакети даних і визначаючи, чи слід їх пропускати, чи ні. Брандмауер працює на вашому комп'ютері у вигляді програми чи вбудованої функції в мережеві апаратні пристрої, як-от маршрутизатори, або у вигляді окремого пристрою. Брандмауер створює бар'єр безпеки, який відокремлює та захищає окремий комп'ютер або локальну мережу від загроз із боку інтернету.

Розглянемо еволюцію брандмауерів, щоб краще уявляти функціональність і можливості сучасних брандмауерів.

Перше покоління

Перші брандмауери працювали як **пакетні фільтри**, перевіряючи кожен окремий пакет, який хотів увійти або вийти з локальної мережі. Брандмауер фільтрує пакети на основі TCP/IP інформації, яку вони містять, і вирішує, чи пройде пакет, звіряє з набором правил, налаштованих власником мережі, що перебуває під захистом. Наприклад, брандмауер можна налаштувати так, щоб він пропускав лише пакети певного протоколу та блокував усі інші або дозволяв пакети, що надходять із певного сервера.

Друге покоління

Брандмауери другого покоління виконують роботу своїх попередників першого покоління, але затримують пакети до тих пір, поки не буде достатньо інформації, щоб зробити висновок про їхній стан. Бранмауер, відомий як Stateful Packet Inspection (Перевірка стану пакетів), реєструє всі з'єднання, що проходять через нього, та визначає, чи є пакет початком нового з'єднання, частиною наявного з'єднання, чи не є частиною жодного з'єднання.

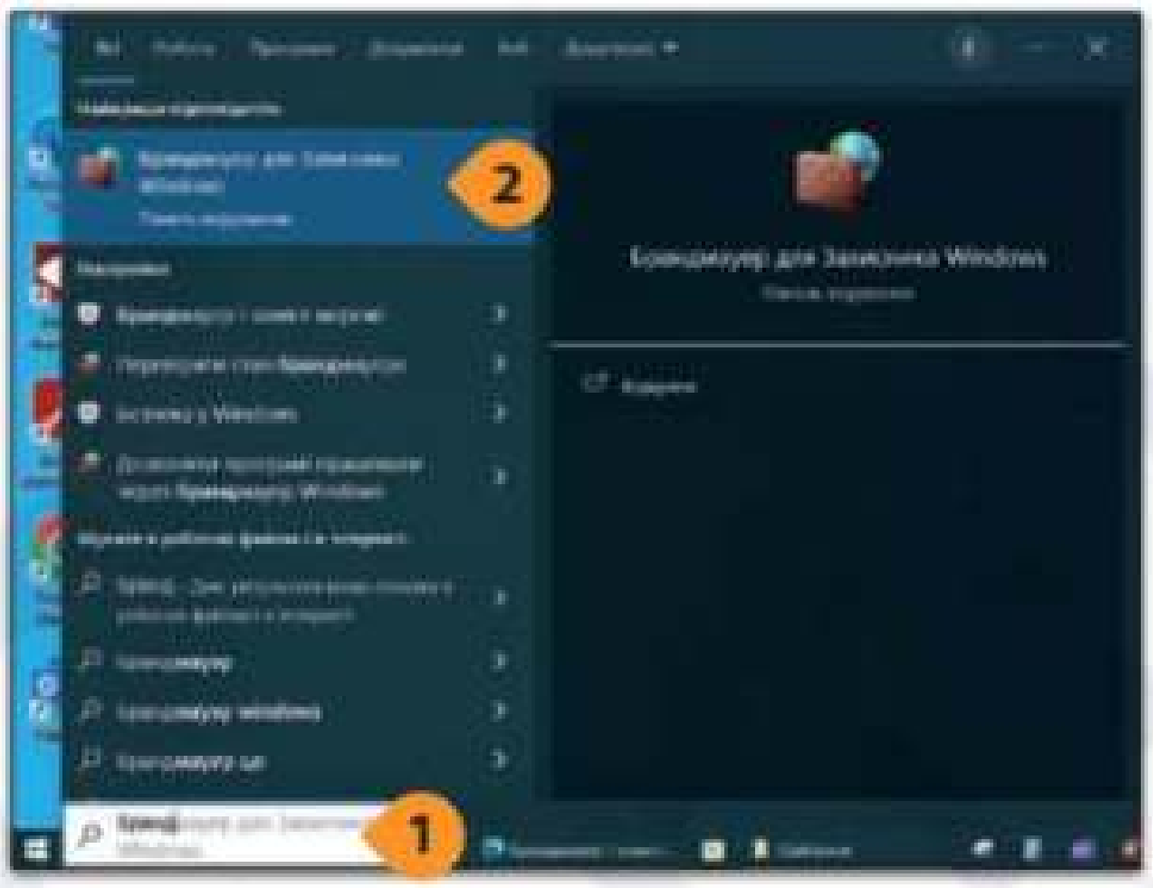
© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Третє покоління

Новітні брандмауери, які називаються брандмауерами прикладного рівня, здатні перевіряти трафік, фільтруючи високорівневі протоколи, як-от FTP, DNS і HTTP, та блокувати будь-які небажані протоколи або виявляти, чи відбувається зловживання протоколом у будь-який шкідливий спосіб. Вони можуть «розуміти» використання пакетів даних і відкидати все, що здається дивним.

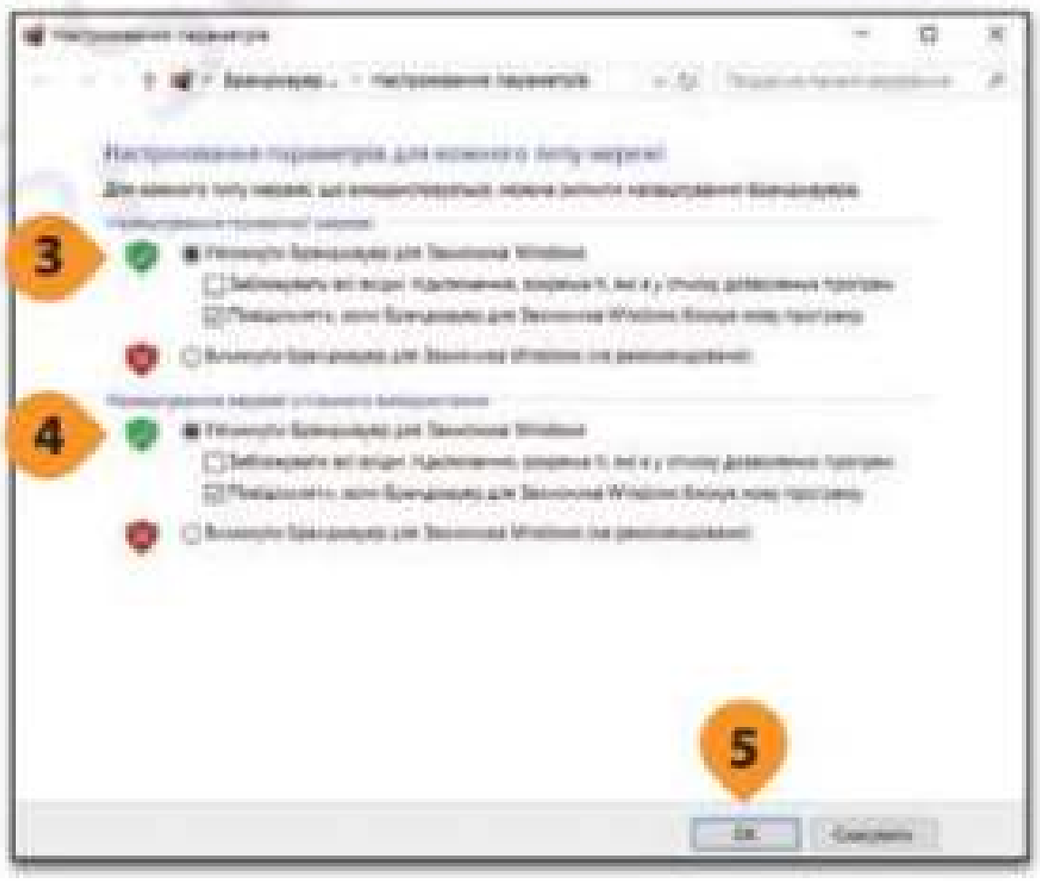
Перевірте свій брандмауер

Microsoft Windows постачається з програмним брандмауером. Перевірте, чи він увімкнений, або увімкніть його.



Щоб увімкнути брандмауер:

- клацніть у полі **Пошук** (1) та почніть друкувати **Брандмауер**;
- оберіть **Брандмауер для Захисника Windows** (2);
- якщо все позначено зеленим кольором, брандмауер увімкнено;
- якщо брандмауер вимкнено, натисніть **Увімкнення** або **Вимкнення Брандмауера для Захисника Windows** (3);
- увімкніть **Брандмауер для Захисника Windows** (4) для всіх мереж і натисніть **ОК** (5).



Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 6

Сучасні технології

Технології вже змінили наш спосіб життя. Сьогодні нові технології з'являються безперервно. Очікується, що вони що вони й надалі будуть докорінно змінювати ділове та соціальне середовище.

Деякі технології змінюють способи нашої взаємодії з інформацією та навколишнім середовищем.

Віртуальна реальність (VR) – симуляція реального або уявного середовища за допомогою комп'ютера. Застосування: відеоігри, навчання, віртуальні екскурсії.

Змішана реальність (MR) – гібрид цифрового та реального контенту. Користувач бачить реальний світ із додатковими цифровими об'єктами. Застосування: навігація, освіта, розваги, реклама.

Доповнена реальність (AR) – накладання комп'ютерного контенту, іноді 3D, на реальний світ, але цей контент не пов'язаний із ним безпосередньо. Користувач може взаємодіяти як із реальними, так і з віртуальними об'єктами. Застосування: промисловий дизайн, медицина, архітектура.

Технологія HoloLens дає можливість керувати марсоходом Curiosity через HoloLens. Гарнітура надає можливість співробітникам NASA встановлювати віртуальні прапорці на місцевості та працювати так, ніби вони самі перебувають на планеті.

ІСТОРІЯ

У 1965 році Айвен Сазерленд придумав концепцію під назвою Ultimate Display. За допомогою наголовного дисплея (HMD), підключеного до комп'ютера, користувачі могли бачити віртуальний світ.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Хмарні обчислення та портативність

Персональні цифрові пристрої, ноутбуки, планшети та смартфони стають дедалі потужнішими. Для всіх цих пристроїв, і підключених до інтернету, і для тих, що працюють автономно, потрібен бездоганний та зручний користувацький досвід. Основна увага приділяється вже не платформам і застосункам, а даним та інформації, до яких можна отримати доступ різними способами.

Подумайте про кількість пристроїв, якими ми користуємося щодня, про файлові сховища, як-от Dropbox, Microsoft OneDrive, Apple iCloud і Google Drive. Згідно з оцінками IoT Analytics, до 2030 року налічуватиметься понад 30 мільярдів підключених пристроїв, причому багато людей використовуватимуть кілька пристроїв протягом дня.

Як хмарні обчислення змінили IT-середовище

- Підвищена безпека. Хмарні технології є більш надійними, ніж традиційні системи.
- Безперервне резервне копіювання незалежно від локального сховища, що значно полегшує відновлення після аварійних ситуацій.
- Існують рішення для застосунків, які зберігають локальну автономну копію даних і синхронізують їх, коли це необхідно.
- Можливість працювати з різних пристроїв і платформ кінцевих користувачів.

Інтернет речей (IoT)

Уявіть, що всі предмети та пристрої навколо вас з'єднані між собою через інтернет. Наприклад, автомобілі, які спілкуються один з одним, обмінюючись інформацією та уникаючи заторів і аварій.

Інтернет речей (IoT) – мережа фізичних пристроїв, транспортних засобів, побутової техніки та інших об'єктів з електронікою, ПЗ, датчиками та засобами зв'язку, які дозволяють цим об'єктам з'єднуватися та обмінюватися даними через наявну інфраструктуру інтернету. IoT дозволяє нам контролювати об'єкти або керувати ними дистанційно, створюючи можливості для більш прямої інтеграції фізичного світу з комп'ютерними системами. ПЗ може бути розроблене для величезного спектру кросплатформених IoT-пристроїв – від маленького комп'ютера Raspberry Pi до великих промислових машин.

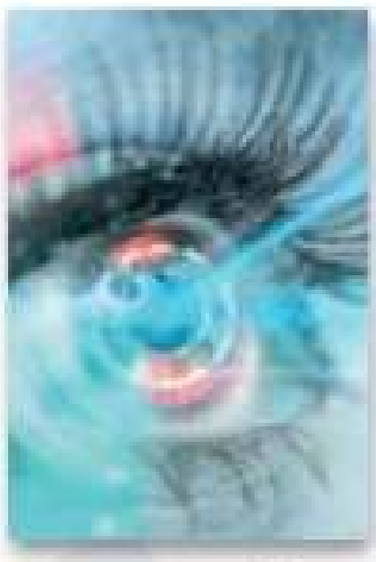
© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Розвиток інтернету речей висвітлює проблему безпеки таких пристроїв. Хакери мають можливість легко зламати розумні пристрої, наприклад камери, датчики чи смартколонки.

Однією з функцій Google Glass є покрокові вказівки. Віртуальні вказівники в поєднанні з назвою вулиць дуже корисні для навігації.

Проблеми безпеки в таких пристроїв схожі на ті, що є у звичайних комп'ютерів та серверів. Але захист, який працює на комп'ютерах (наприклад, антивіруси та брандмауери), не завжди підходить для портативних смартпристроїв, тому що вони мають слабше апаратне забезпечення та ПЗ.

Інтернет речей потребує нових способів захисту та нових правил для виробників, які змусять їх робити пристрої більш захищеними.



Біометрія

Нові паспорти використовують біометричні дані для автентифікації своїх власників. Біометричні дані – унікальні вимірювані фізичні характеристики, які використовуються для підтвердження особи. Такими ознаками є, зокрема, колір очей, зріст і відбитки пальців.

Якщо у вас є біометричний паспорт, можна скористатися автоматизованими пропускними пунктами ePassport замість того, щоб ваш паспорт вручну перевіряв офіцер. Це прискорює в'їзд до країни.

Сучасні застосунки використовують біометричні дані найчастіше як вторинну форму автентифікації. Але в деяких випадках момент, коли біометричні системи повністю замінять PIN-коди, вже настав.

Наприклад, для розблокування комп'ютера або смартфона можна використовувати відбиток пальця або обличчя (Windows Hello, Apple Face ID) чи голос. В установах із високим рівнем безпеки, як-от центри обробки даних або дослідницькі центри, використовують сканери очей, що сканують райдужну оболонку або сітківку ока. Ці технології використовують математичні методи розпізнавання образів на зображеннях одного або обох очей, чиї складні візерунки є унікальними та стабільними.

RFID-мітка *(англ. Radio Frequency Identification) – електронний пристрій, який використовує радіочастотне електромагнітне поле для зберігання та передавання інформації. Застосовується Застосовується у платіжних системах (безконтактні картки), роздрібній торгівлі (захист від крадіжок), паспортах і посвідченнях особи, а також для контролю доступу, ідентифікації тварин.*

ІСТОРІЯ

Термін «Інтернет речей» придумав наприкінці 1990-х років бізнесмен Кевін Ештон. Він один із засновників Центру автоматичної ідентифікації в Массачусетському технологічному інституті, був частиною команди, яка розробила спосіб підключення об'єктів до інтернету за допомогою RFID-мітки. Підприємець стверджує, що вперше використав фразу «Інтернет речей» у презентації, яку він зробив у 1999 році.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Проблеми сучасних технологій

Існують проблеми, пов'язані з конфіденційністю та безпекою.

Біометрична інформація є конфіденційною персональною інформацією, завжди повинна збиратися з відома та згоди особи, якій вона належить.

Систему, що містить конфіденційну інформацію, відносно легко зламати. Це може призвести до крадіжки особистих даних. Отримавши такі дані, зловмисник може видавати себе за іншу особу.

Навіть якщо ця інформація не викрадена, постають питання: хто її збирає, зберігає та контролює? Як ця інформація використовується правоохоронними органами чи іншими державними установами? Чи не порушуються при цьому права людини?

Використання біометричних технологій у правоохоронних та безпекових цілях вимагає впровадження відповідної правової бази, яка б гарантувала, що ці процедури ідентифікації застосовуються з дотриманням та заохоченням прав людини. Персональна інформація повинна залишатися приватною та не підлягати зловживанню, а збирати її слід лише в разі потреби та в тому обсязі, який є необхідним і законним. Доступ до цієї інформації повинен бути регламентований, а обмін нею повинен обмежуватися обміном, необхідним для функціонування системи ідентифікації.

Портативні цифрові пристрої

Розумні годинники та фітнес-трекери – два основних типи пристроїв, які завжди можна носити на тілі та під'єднувати до інтернету або до комп'ютерів поруч із вами.

Пристрої для навігації чи охорони здоров'я корисні в певних сферах застосування. Одяг, прикраси та окуляри, як-от Google Glass, – наступна хвиля пристроїв, які поєднують технологію зі стилем та розширеною функціональністю.

Портативні пристрої здатні, залежно від розміру екрана, показувати повідомлення та сповіщення, інформацію про погоду, останні новини, допомагати робити нотатки чи збирати дані про показники здоров'я, здійснювати навігацію за допомогою онлайн-карт. Вони можуть працювати як автономні пристрої та синхронізуватися з комп'ютером / смартфоном або через Bluetooth чи WiFi.

Недоліки портативних пристроїв

- Проблеми з конфіденційністю під час зйомки особистих моментів людей, які вас оточують.
- При віддаленому доступі до цих пристроїв може бути порушена конфіденційність, а ваші дані можуть бути викрадені.
- Ви завжди перебуваєте в хмарі електромагнітних хвиль. Теоретично вони нешкідливі, однак тільки тривале дослідження покаже, чи так це насправді.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Мобільні платежі

Мобільні платежі – транзакції, що здійснюються за допомогою смартфонів або портативних пристроїв, як-от смартгодинники, браслети чи каблучки. Наприклад, компанія VISA розробила браслет, який можна використовувати як безконтактну банківську картку. Існують різноманітні браслети та каблучки для швидких, зручних і стильних безконтактних транзакцій. Усі ці пристрої використовують технологію NFC (ближній радіозв'язок), яка дозволяє двом пристроям, що розташовані на відстані не більше 10 см один від одного, обмінюватися даними, наприклад платіжною інформацією. Ви торкаєтеся пристроєм до POS-термінала, який зчитує вашу особисту інформацію, вбудовану в пристрій, і передає дані в застосунок, ваш банк або мережу кредитних карток.

Високошвидкісний мобільний зв'язок

Четверте та п'яте покоління технологій (4G та 5G) змінили галузь розваг, бізнесу та медицини.

LTE (англ. Long-Term Evolution) – стандарт, який є частиною технологій 4G і зараз широко використовується. Він може підтримувати з'єднання на пристроях, які рухаються зі швидкістю 350–500 км/год. Теоретично максимальна швидкість завантаження становить 300 Мбіт/с, а швидкість передачі даних – 80 Мбіт/с. У 2018 році хороше з'єднання могло завантажувати дані зі швидкістю 100 Мбіт/с і передати їх зі швидкістю 45 Мбіт/с. Очікується, що LTE-Advanced досягне пікової швидкості передачі даних до 1 Гбіт/с.

Стандарт 5G надає швидкість понад 1 Гбіт/с і зменшує час затримки, яка ускладнює роботу застосунків у реальному часі через мобільні мережі.

Зберігання даних

Місця для зберігання даних ніколи не буде достатньо. На теперішній час диск на 4 ТБ може бути маленьким, а крихітна карта MicroSD на 400 ГБ може легко загубитися. Еволюція комп'ютерних накопичувачів демонструє прогрес у мініатюризації та збільшенні ємності носіїв інформації – від громіздкого дискового накопичувача IBM 350 1956 року, що важив тонну й мав мізерні 3,75 МБ пам'яті, до сучасних компактних SSD-накопичувачів ємністю 15 ТБ і розміром всього 15 мм. Типова 42-дюймова шафа центру обробки даних сьогодні може вмістити близько 10 ПБ даних, тобто 10 мільярдів гігабайт. Ці цифри зростають із кожним роком.

Розробляються та постійно розвиваються нові технології: окрім магнітного та твердотільного, розвивають оптичне та голографічне зберігання. Досліджуються революційні методи зберігання даних на молекулах білків або ДНК, які обіцяють величезний стрибок в обсягах пам'яті.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Проблеми зі зберіганням даних

1	Вартість за гігабайт.
2	Швидкість доступу.
3	Довговічність даних.
4	Енергоспоживання. Процесори та накопичувачі споживають найбільше енергії на сервері.

Квантові обчислення

Квантові обчислення використовують дивну здатність елементарних частинок існувати в більш ніж одному стані одночасно.

Завдяки тому, як поводяться ці частинки, квантові комп'ютери можуть виконувати операції швидше і при цьому використовувати менше енергії, ніж традиційні комп'ютери. У класичних обчисленнях біт – одиниця інформації. Квантові обчислення використовують «кубіти», квантові біти.

Щоб зрозуміти різницю, уявіть собі сферу. Біт, який може існувати лише у двох станах – **1** або **0** – може бути на будь-якому з двох полюсів сфери, але кубіт може існувати в будь-якій точці сфери. Розвиваючи квантові обчислення там, де традиційні закони фізики більше не застосовуються, створюють процесори, які значно швидші (в мільйон і більше разів) за ті, які використовуються сьогодні, та зможуть обробляти напрочуд величезні обсяги даних.

Квантові комп'ютери працюють за допомогою кубітів, які є еквівалентом бітів, але замість того щоб просто мати положення «увімкнено» або «вимкнено», як у наших двійкових версіях, квантовий біт (кубіт) може бути або вимкненим, або увімкненим, або і тим, і іншим (так звана суперпозиція).

Непросто передбачити, як квантові обчислення змінять наш світ. Це абсолютна нова галузь фізики, яка надасть можливість з'явитися рішенням та застосуванням, про які ми ще навіть не думали. Квантові обчислення вплинуть на кожну галузь та людство.

Гнучкі дисплеї

Гнучкі дисплеї – це неймовірні екрани, які можна згинати, скручувати, розтягувати, при цьому екран не зламається і не буде спотворювати зображення. Такий дисплей складається з OLED-матриці, а замість скла як підкладку використовують спеціальні пластики. Вони не тільки тонкі й легкі, але й не б'ються при падінні.

Такі дисплеї можна використовувати не тільки в смартфонах чи комп'ютерах, але й під час створення розумного одягу, в автомобілях, у медицині.

ДОВІДКА

OLED (Organic Light-Emitting Diode) – це технологія виготовлення дисплеїв, де кожен піксель самостійно випромінює світло без потреби в підсвічуванні.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Нові технології дисплеїв

Сьогодні індустрія дисплеїв розвивається швидкими темпами. Протягом кількох років ви бачили монітори, які використовують різні типи технологій, як-от LCD, плазма, LED або OLED. Сьогодні деякі телевізори та монітори комп'ютерів мають дисплеї з роздільною здатністю 4K (3840 × 2160 пікселів зі співвідношенням сторін 16 : 9), а інші навіть підтримують дисплеї з роздільною здатністю 5K (5120 × 2880 пікселів).

Дуже скоро повсякденністю стануть телевізори з роздільною здатністю 8K 7680 × 4320 пікселів. Дисплей із роздільною здатністю 8K – це як чотири дисплеї з роздільною здатністю 4K (2 × 2) або 16 дисплеїв із роздільною здатністю Full HD. Це наступний стандарт для цифрового телебачення та цифрового кінематографу, тому треба готуватися до збільшення обсягів зберігання даних.

З такими технологіями відтворення зображення є певні проблеми, і їх слід узяти до уваги.

Всі монітори з високою роздільною здатністю гарні та відображають реалістичне зображення, але контент із надвисокою роздільною здатністю недоступний.

Існують вимоги до зберігання даних, які набагато вищі, ніж Full HD, а це означає, що традиційні носії інформації не мають достатньої ємності для розповсюдження такого типу контенту.

Потокове відео з роздільною здатністю 4K або вище означає, що потрібна пропускна здатність інтернету, яка може підтримувати вчетверо більший розмір, ніж Full HD, медіа-сервери з боку провайдера, які можуть підтримувати мільйони абонентів на дуже високих швидкостях.

Щоб дивитися відео в надвисокій якості, потрібні дуже великі екрани. По-справжньому відчути різницю у 8K можна лише на телевізорі від 80 дюймів. А для великих екранів потрібні великі приміщення.

ІСТОРІЯ

90 годин зимових Олімпійських ігор 2018 транслювалися у форматі 8K.

Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 7

Технології та навколишнє середовище

Куди подіти старі цифрові пристрої

Приємно мати великий екран із вищою роздільною здатністю для більш якісного зображення. Були часи, коли телевізор можна було використовувати більше десяти років, а комп'ютер – роками. Сьогодні через зміни в технологіях і споживчих тенденціях навряд чи знайдеться пристрій, який буде використовуватися більш ніж два-три роки. Виробники також не сприяють ефективному функціонуванню пристроїв, оскільки припиняють їхню підтримку або роблять ремонт складнішим, незручним чи дуже дорогим. Здебільшого економічно вигідніше купити новий пристрій на заміну попередньому. Уявіть, що більшість телевізорів і моніторів, які ми замінюємо, не мають проблем із функціональністю. Нам просто хочеться найкращу та найновішу технологію.

Що ж відбувається з усіма попередніми моделями? Куди діваються всі ці пристрої? Як щодо принтерів, комп'ютерів, планшетів і смартфонів? Половина викинутих пристроїв все ще працює. Щоб захистити навколишнє середовище, всі пристрої, які більше не використовуються, повинні бути належним чином утилізовані. Цифрові пристрої складаються з електронних компонентів, які містять деякі токсичні речовини, тому їх не можна викидати. Наприклад, ми з дитинства знаємо, що батарейки – небезпечні та підлягають переробці. Одна літій-іонна батарея зі смартфона може забруднити до 60 000 літрів води. Але це лише верхівка айсберга. Чи правильно ми утилізуємо наші пристрої?

У 2016 році електронні відходи, утворені в усьому світі, досягли ваги, еквівалентної дев'яти великим пірамідам Гізи або 4 500 Ейфелевим вежам.

Вплив технологій на навколишнє середовище

Зростання обсягів споживання цифрових пристроїв має два основні негативні екологічні наслідки: по-перше, воно значно збільшує видобуток дуже цінної та дефіцитної сировини, необхідної для виробництва електронних пристроїв, а по-друге, викинуті пристрої утворюють величезну кількість електронних відходів, так званих **e-waste**.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Електронні відходи

Цифрові пристрої містять токсичні матеріали, зокрема свинець, нікель і ртуть. Вони забруднюють ґрунт, повітря й воду. Трава, дерева, сільськогосподарські культури, а також тварини й люди є жертвами забруднення. Важкі метали, що містять пристрої, наприклад миш'як, берилій та кадмій, із часом накопичуються в навколишньому середовищі та організмі людини.

За даними ООН, у 2017 році було незаконно захоронено до 50 мільйонів тонн електронних відходів, переважно комп'ютерів і смартфонів. Агентство з охорони навколишнього середовища США підрахувало, що лише ~ 27 % електронних відходів переробляється. Решта потрапляє на звалищаі становить 70 % усіх токсичних відходів.

Переробка відходів на енергію

За допомогою різних процесів можна отримати енергію у вигляді тепла чи електроенергії через перетворення відходів, що не підлягають вторинній переробці.

Приблизно 150 мільйонів смартфонів щороку викидають лише в США.

Унаслідок переробки 42 смартфонів вивільняється (виділяється) така кількість енергії, якої вистачить для живлення середнього домогосподарства протягом цілого року! Це дозволить зекономити енергію, яка буде вироблена з вичерпних ресурсів, зокрема з вугілля.

Зменшення електронних відходів

Електронні відходи можна зменшити завдяки ремонту, перепродажу чи переробки. Ремонт є складним, але він також допоможе зменшити безробіття, якщо навчити людей необхідних ремонтних навичок та організувати належні ремонтні потужності. З іншого боку, переробка повинна здійснюватися належним чином. Було виявлено, що електроніка, призначена для переробки, потрапляє на звалища або неофіційні «переробні» підприємства в Гані, Індії чи Китаї, де вона розпадається в навколишньому середовищі, забруднюючи його токсичними речовинами. Вся ця електроніка неналежним чином демонтується та спалюється, утворюючи токсичні викиди, шкідливі для населення місцевих громад.

Принципи поводження з електронними відходами

1	Зменшуйте споживання електронного та електричного обладнання.
2	Повторно використовуйте обладнання доти, доки це практично можливо. В іншому випадку продайте, обміняйте його на нове (trade-in) або пожертвуйте.
3	Відправляйте на переробку обладнання, де компоненти відновлюють та використовують для виробництва нових продуктів.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

В Україні торговельні мережі пропонують споживачам здати на переробку акумулятори чи батарейки безпосередньо в їхніх магазинах. А деякі магазини з продажу смартфонів пропонують вам обміняти старий телефон на новий та отримати знижку.

Стрімкий розвиток технологій розширює сферу цифрового споживання та виробництва. Як наслідок, збільшується кількість матеріалів, необхідних для виробництва цифрових пристроїв, будівництва заводів і загалом для задоволення потреб споживача. Втім, учені створили кілька нових технологій, які дозволяють зменшити кількість необхідних матеріалів.

3D-друк

Виробництво на замовлення стало можливим завдяки розвитку таких технологій, як 3D-друк та лазерне різання. Легко створити прототип, а потім за допомогою 3D-принтера надрукувати його в обсязі невеликого замовлення. Це призводить до зменшення складських запасів, відходів, енергоспоживання, зниження витрат і використання точної кількості матеріалів для створення товарів.

Малі масштаби

Мистецькі, інженерні та освітні проєкти мають різні сфери застосування. Команда технічної підтримки може створити запасну частину для машини, яка не може бути доступною в таких місцях, як віддалені райони, наприклад, в Антарктиді, в морі або навіть у космосі. У медицині можна побачити застосування в персоналізованих імплантатах і протезах, а також у створенні штучних тканин через нанесення шарів живих клітин на гелеву основу (біодрук).

У серпні 2017 року в Європейському Союзі було створено першу будівлю, надруковану на 3D-принтері.

Великі масштаби

3D-будівельний друк (3DCP) використовується для будівництва цілих будівель із бетону, пінопласту чи полімеру. Перевагами є менші трудовитрати, швидше будівництво, підвищена точність і менші відходи матеріалів. У Львові вже побудували школу за допомогою такого принтера.

ДОВІДКА

Блейд-сервер (англ. Blade server) – компактний та енергоефективний серверний комп'ютер із модульною конструкцією, який розміщується в спеціальній стійці або шасі.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Технології енергозбереження

Досягнення в технологіях виробництва процесорів і зберігання даних, наприклад SSD, привели до створення енергоефективних блейд-серверів, які спільно використовують апаратні ресурси залежно від конкретних потреб. ПЗ оптимізовано таким чином, щоб отримувати доступ до кожного апаратного компонента лише за необхідності, а системи, що не працюють, споживають мінімальну кількість енергії, доки не будуть потрібні.

Центри обробки даних зараз будують поруч із річками, де є велика кількість води для охолодження, або в дуже холодному середовищі, переважно в північних штатах США, Канади, на півночі Великої Британії та в Скандинавії, де низькі температури навколишнього середовища використовують для охолодження серверів. Раніше охолодження вимагало стільки ж енергії, скільки й живлення цих серверів.

Наприкінці 2016 року компанія Google заявила, що використовує 2,5 мільйона серверів на більш ніж 100 сайтах. Для їхньої роботи компанія уклала угоди на 2,6 ГВт відновлюваної енергії.

Великі компанії, як-от Facebook, Apple та Microsoft, мають центри обробки даних із мільйонами серверів. Microsoft навіть розглядає можливість будівництва підводного дата-центру Project Natick. Дно океану має відносно сталу температуру води та не піддається впливу штормів і течій.

Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

https://formula.education/online-ict-8-klas/

Інші платформи

Проєкт

Підсумки

Глосарій

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

2. Працюємо із числами

У цьому розділі ви дізнаєтеся, що таке електронна таблиця, як збирати дані та впорядковувати їх у рядках, стовпцях та комірках, як ці дані можна використовувати для обчислення та отримання інформації. Навчитеся представляти інформацію різними типами діаграм та форматувати дані так, щоб їх було легше сприймати.
Ви також дізнаєтеся багато цікавого про моделювання та симуляцію.

У цьому розділі ви навчитеся:

- працювати з рядками, стовпцями та комірками;
- виконувати обчислення;
- використовувати формули;
- використовувати відносні та абсолютні посилання;
- розуміти повідомлення про помилки та виправляти їх;
- застосовувати умовне форматування до комірок;
- створювати діаграми різних типів;
- друкувати створені матеріали.

У цьому розділі ви дізнаєтеся:

- як упорядкувати інформацію;
- як виконати обчислення;
- як працювати з логічними функціями;
- як уникнути помилок у розрахунках;
- як підкреслити інформацію за допомогою форматування;
- як створити діаграму для представлення інформації;
- як подавати інформацію в діаграмах;
- як зберігати та друкувати аркуші;
- як використовувати моделювання та симуляції в повсякденному житті.

Програмне забезпечення:

- Microsoft Excel;
- Apple Numbers;
- Google Таблиці;
- LibreOffice Calc.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 1

Збір даних

Ви точно знаєте, що таке дані? **Дані** – це необроблені факти та спостереження. **Інформація** – це дані про певний предмет чи явище або їх сукупність, які були перетворені в придатний для сприйняття та опрацювання вигляд.

Розгляньмо різні типи даних:

- Листя на деревах зелене.
- Автомобіль має чотири колеса.
- $2 + 2 = 4$
- $P = m \cdot g$

Щоб дані стали інформацією, їх спочатку збирають, а потім упорядковують за певними критеріями. Наприклад, школа збирає відомості про своїх учнів та створює впорядковані теки з даними про них. До появи комп'ютерів ці дані зберігалися в паперовому вигляді. Сьогодні шкільні дані зберігаються в електронних базах даних із картками усіх учнів. Кожна картка виглядає так само, як у минулому, але тепер відображається на моніторі комп'ютера.

Збір даних є основою для багатьох галузей та процесів ухвалення рішень. Він вимагає ретельного планування, вибору відповідних методів та інструментів, а також дотримання етичних норм та законодавчих вимог.

Зібрати дані можна різними способами. Поміркуйте, які це способи.

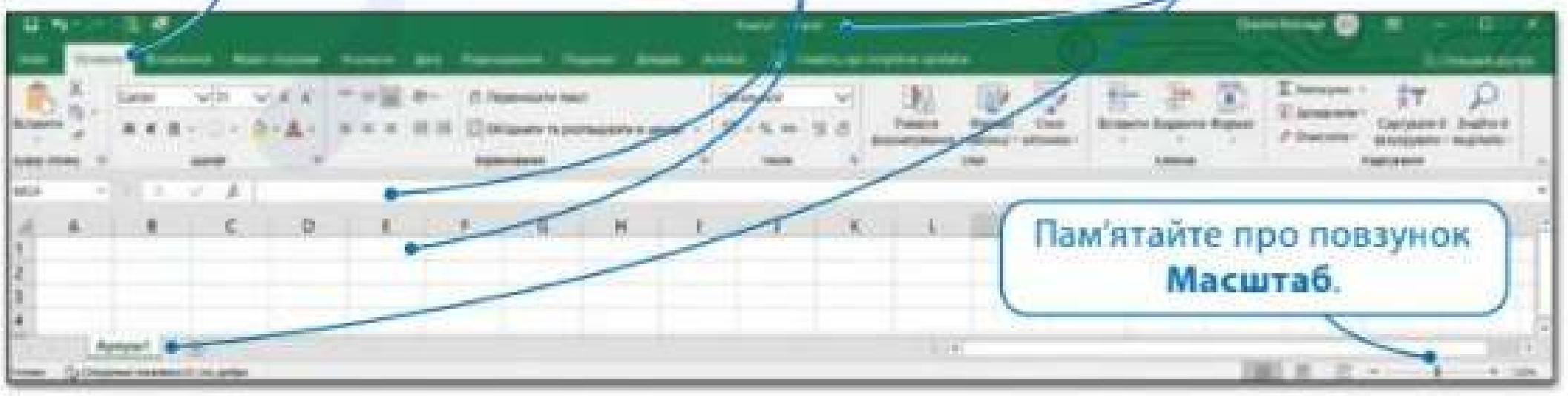
Опитування, анкетування, аналітика вебсайтів, мобільні застосунки для збору даних, збір даних на вебсайтах, ваші дії в інтернет-магазинах, аналіз текстових даних, візуалізація даних тощо – ось основні способи збору даних.

Збір даних – це процес внесення даних у комп'ютерну систему. Впорядкуйте отримані дані, щоб вони стали придатними для аналізу користувачем. Проаналізувавши їх, він зробить висновки, отже, отримає інформацію. Скористайтеся **Microsoft Excel**. Це програма для роботи з електронними таблицями, де основний документ для роботи та зберігання даних – **аркуш**. Робочих аркушів може бути багато, їх зберігають у файлі, який називається **робочою книгою**.

Усі програми електронних таблиць мають схожий інтерфейс користувача: вкладки, кнопки та елементи керування розташовані всюди однаково.

Вводити формули можна у **Рядку формул**. Це можна робити і в комірках.

Назва файлу Microsoft Excel за замовчуванням – **Книга1**. У документі можна додавати робочі аркуші.



© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

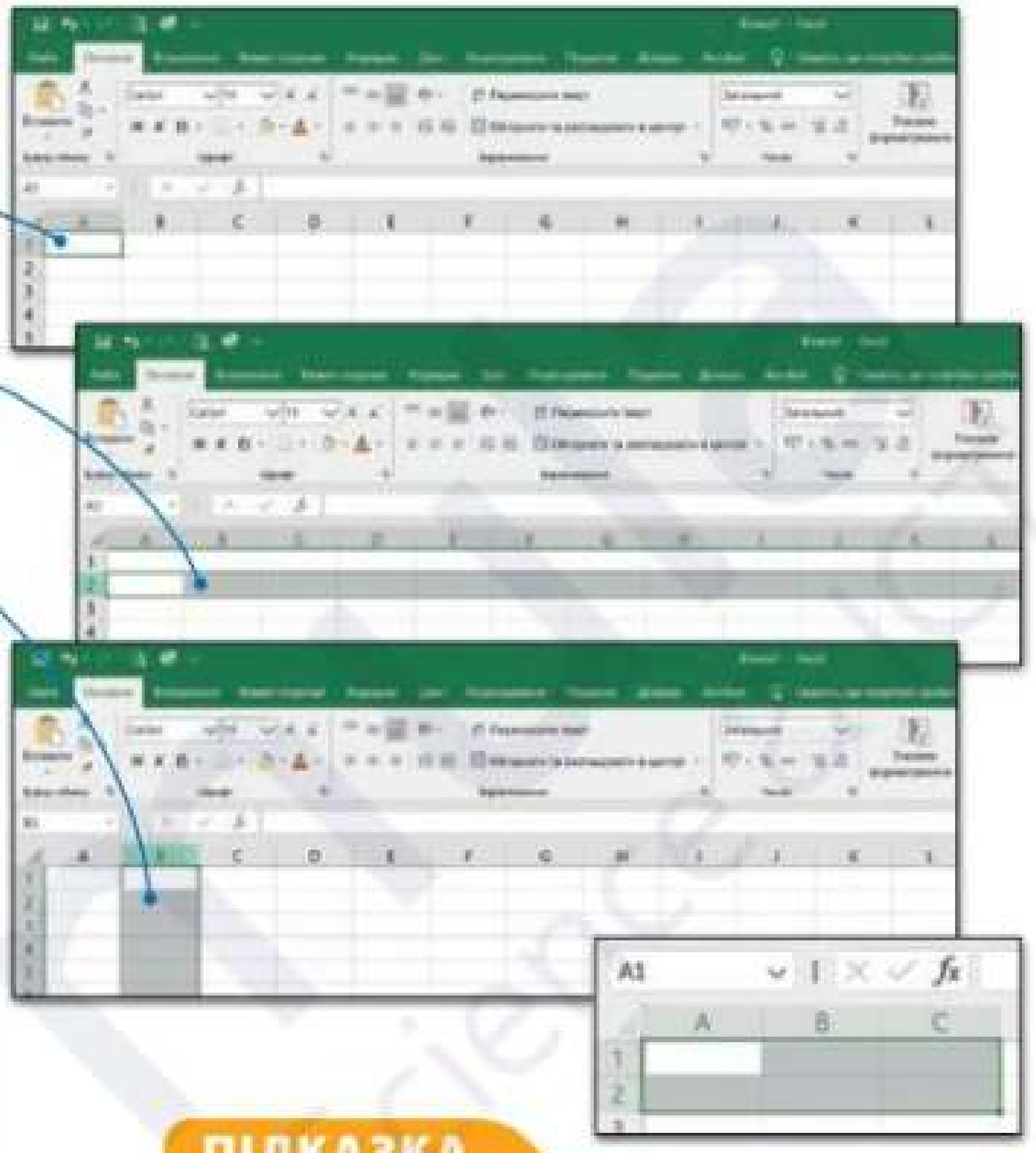
Рядки та стовпці

Електронна таблиця – сітка з комірками, які утворюються на перетині рядків і стовпців. Введіть дані в ці комірки. Відкрийте нову електронну таблицю та розгляньте її детальніше.

У **комірки** вносять дані: букви, тексти, числа, символи, формули тощо.

Кожен **рядок** пронумеровано з лівого боку. 1, 2, … – номери рядків.

Кожен **стовпець** позначено великою англійською літерою вгорі. A, B, … – назви стовпців.



Вибраний діапазон комірок зазвичай пишеться ***A1:C2****, що означає від комірки A1 до комірки C2. Окремі вибрані комірки записуються через кому* ***A1,B2,C3****, що означає комірки A1 і B2 і C3.*

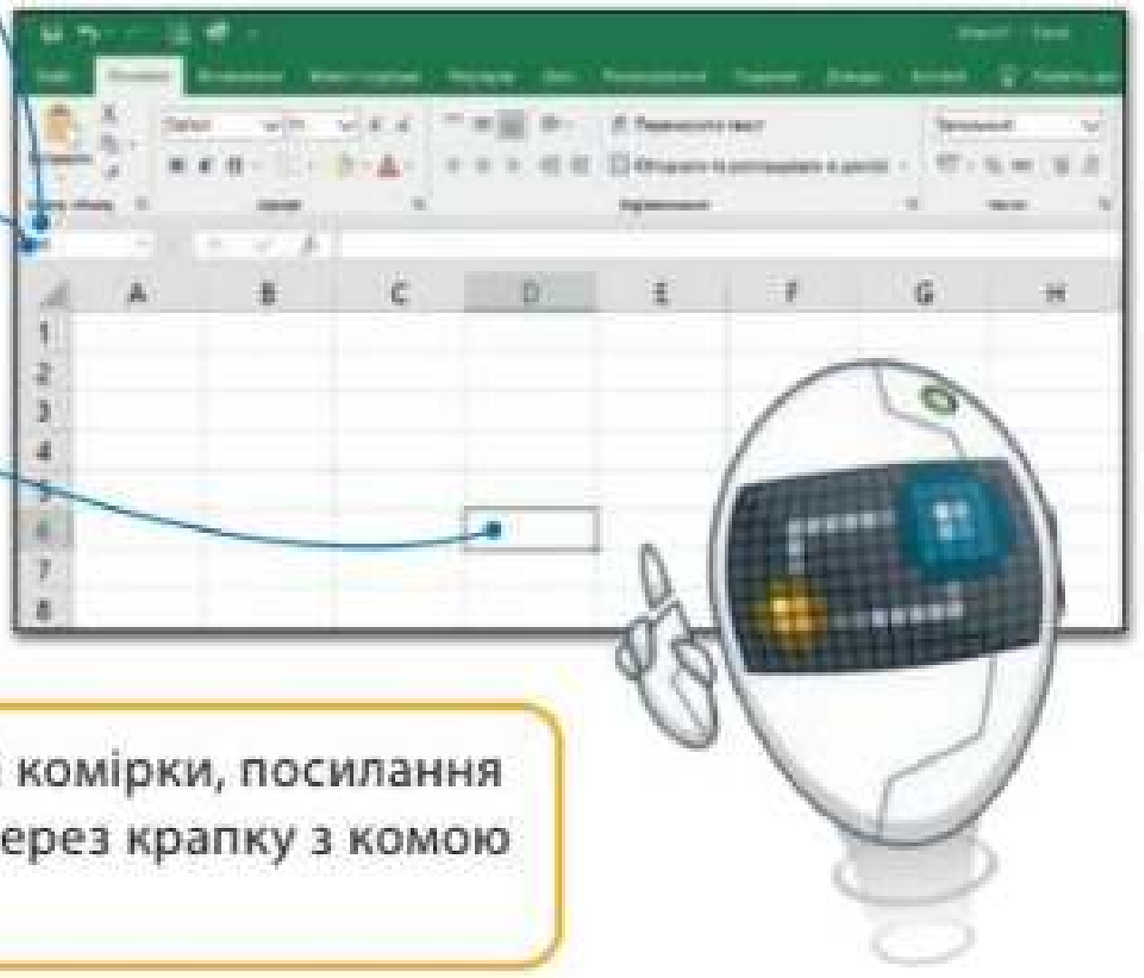
ПІДКАЗКА

Ви коли-небудь грали в шахи або морський бій? Комірки електронної таблиці називають за таким самим принципом, як і клітинки шахової дошки!

Кожна комірка має унікальне ім'я, яке складається з літери стовпця та номера рядка, наприклад **D6**.

Ім'я активної комірки можна побачити у верхньому лівому куті електронної таблиці в полі **Ім'я**.

Якщо клацнути на комірку, вона стає активною: межі стають товстішими та змінюють колір.



ПІДКАЗКА

Щоб швидко знайти та вибрати певні комірки, посилання на комірки записуються в полі **Ім'я** через крапку з комою між ними, наприклад **A1;B2;C3**.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

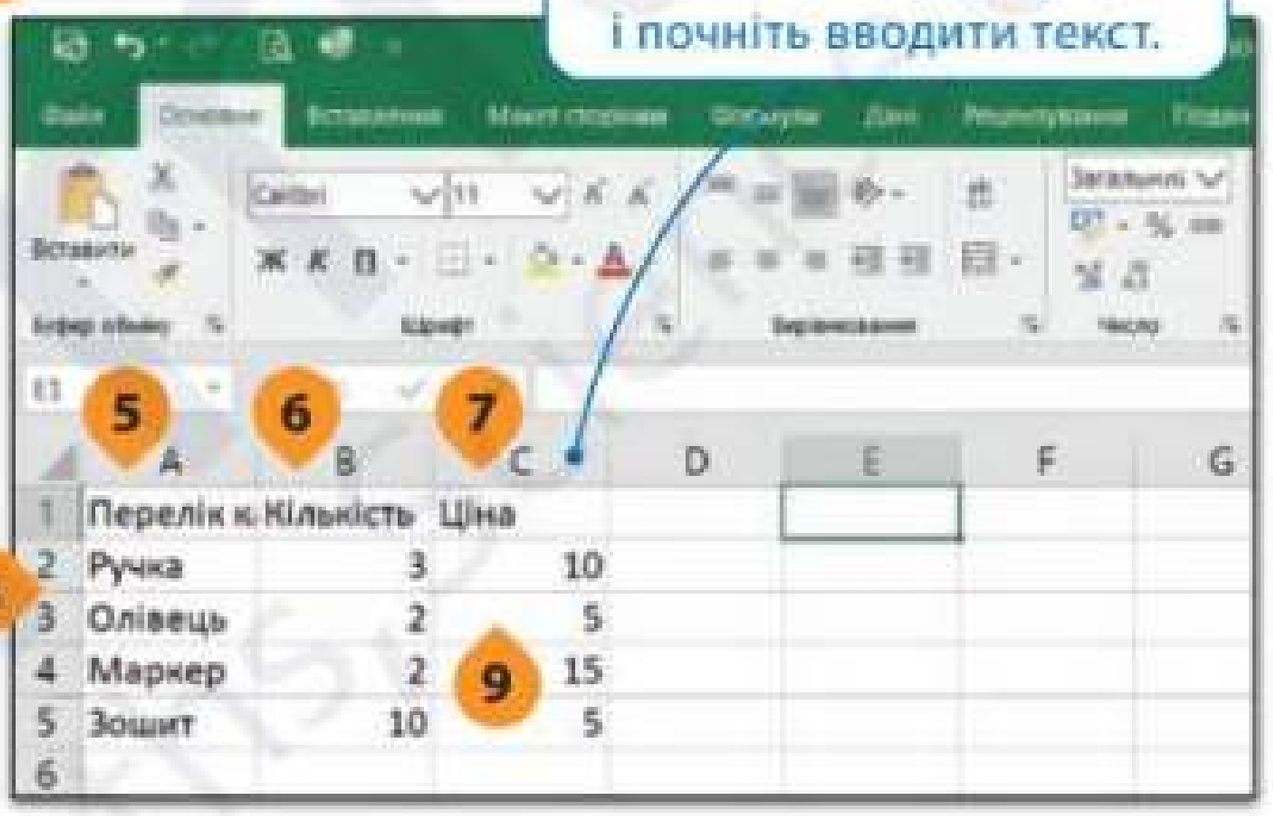
Вводимо числа й текст

Вводити числа й текст в електронну таблицю дуже легко. Просто виділіть комірку, введіть дані та переходьте до іншої комірки чи натисніть Enter ↵. **Microsoft Excel** розрізняє комірки із числами та комірки з текстом. Якщо ви вводите в комірку і текст, і числа, програма сприймає все як текст.

Щоб створити таблицю:

- натисніть кнопку **Пуск** (1);
- прокрутіть бічну панель (2);
- клацніть **Microsoft Excel** (3);
- оберіть **Пуста книга** (4);
- перейдіть до комірки **A1** та надрукуйте **Перелік канцтоварів** (5);
- натисніть →, щоб перейти до комірки **B1**, та введіть **Кількість** (6);
- клацніть →, щоб перейти до **C1**, та введіть **Ціна** (7);
- перейдіть до комірки **A2** та надрукуйте **Ручка**, клацніть **A3** і введіть **Олівець** (8);
- у комірці **B2** введіть **3**, у **B3** – **2**, у **C2** – **10**, а в **C3** – **5** (9).

Якщо в комірку потрібно додати текст, двічі клацніть на неї, або один раз клацніть на **Рядок формул**, і почніть вводити текст.



Типи числових даних

У Microsoft Excel існує кілька типів числових даних.

Ці типи числових даних дають змогу виконувати різноманітні розрахунки та аналіз.

За замовчуванням будь-який текст вирівнюється за лівим краєм, а всі числа – за правим.

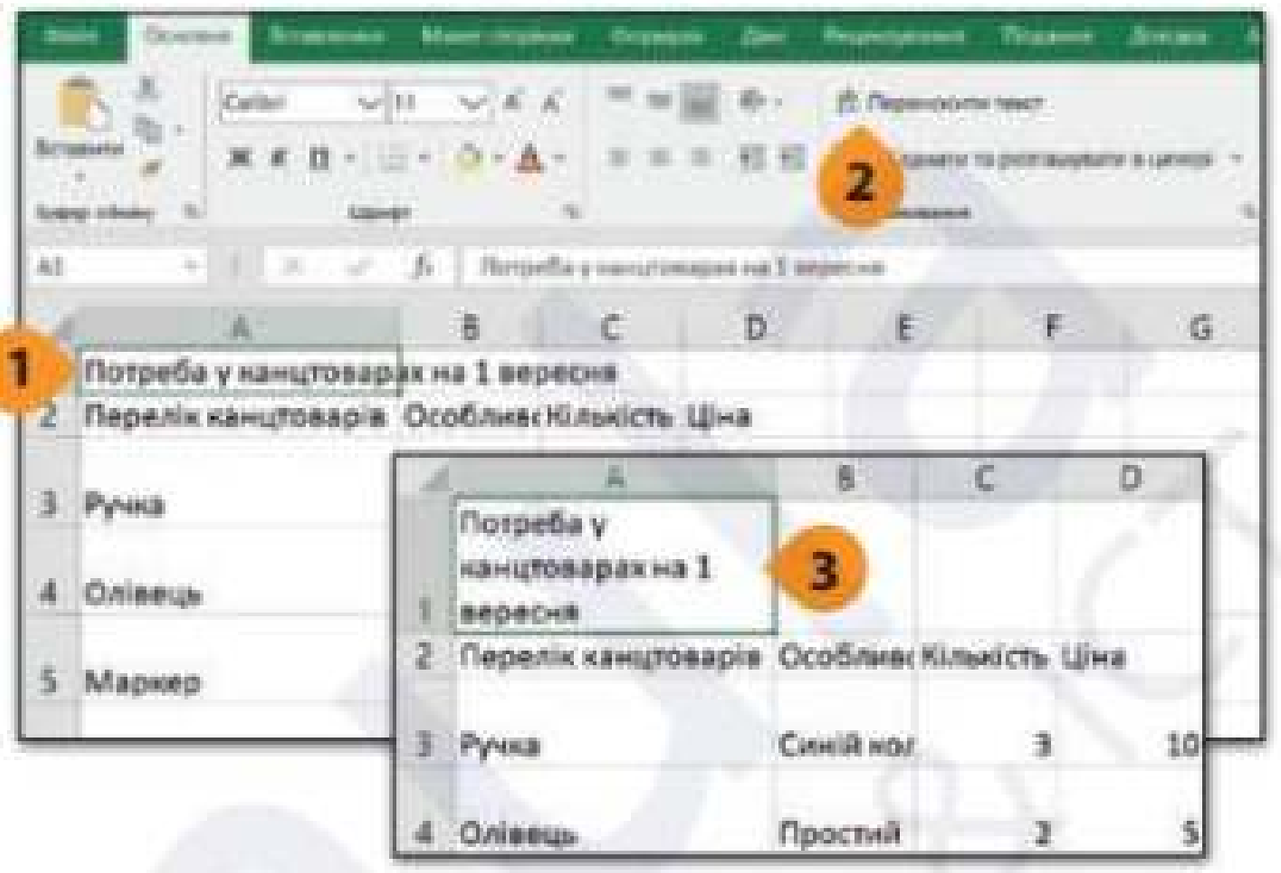
© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Переносимо текст

Якщо треба ввести довгий текст, але не хочете робити стовпець ширшим, можна комірку зробити багаторядковою. Текст буде перенесено на наступні рядки, поки він не відобразиться повністю. Перенесений текст автоматично регулюватиме висоту рядка, у якому розташована комірка.

Щоб перенести текст:

- клацніть комірку **A1** (1);
- на вкладці **Основне** в розділі **Вирівнювання** натисніть **Переносити текст** (2);
- текст буде автоматично перенесено так, щоб він повністю вміщався в комірці (3).

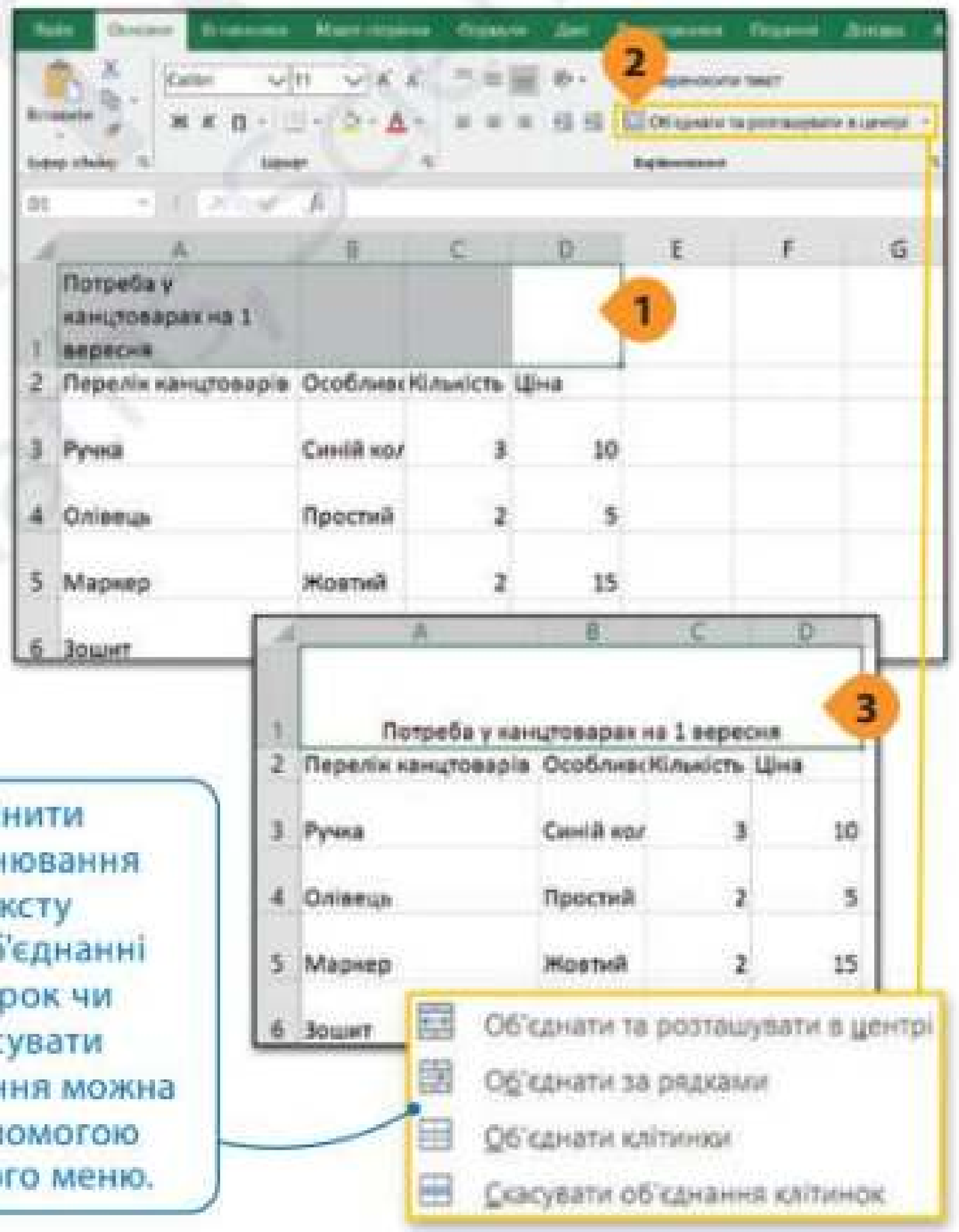


Об'єднуємо комірки

Гарний заголовок краще презентує ваші дані. Щоб створити заголовок над стовпцями, можна об'єднати декілька комірок, і в такий спосіб зробити довгу комірку.

Щоб об'єднати комірки:

- виділіть комірки від **A1** до **D1** (1);
- на вкладці **Основне** в розділі **Вирівнювання** натисніть **Об'єднати та розташувати в центрі** (2);
- Microsoft Excel автоматично об'єднає виділений діапазон комірок та розмістить текст у центрі (3).



Змінити вирівнювання тексту при об'єднанні комірок чи скасувати об'єднання можна за допомогою спадного меню.

© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Просте форматування

Іноді потрібно трохи змінити таблицю, щоб легше було сприймати дані чи привернути до них увагу, особливо якщо це стосується числових даних. Ось чому треба форматувати дані. Панель інструментів форматування дуже схожа на панель у **Microsoft Word**.

Пам'ятайте! Спочатку виділить дані, а потім форматуйте їх.

Вирівнює текст у комірках за верхнім краєм, посередині або за нижнім краєм.

Змінює тип даних.

Збільшує кількість десяткових знаків після коми.

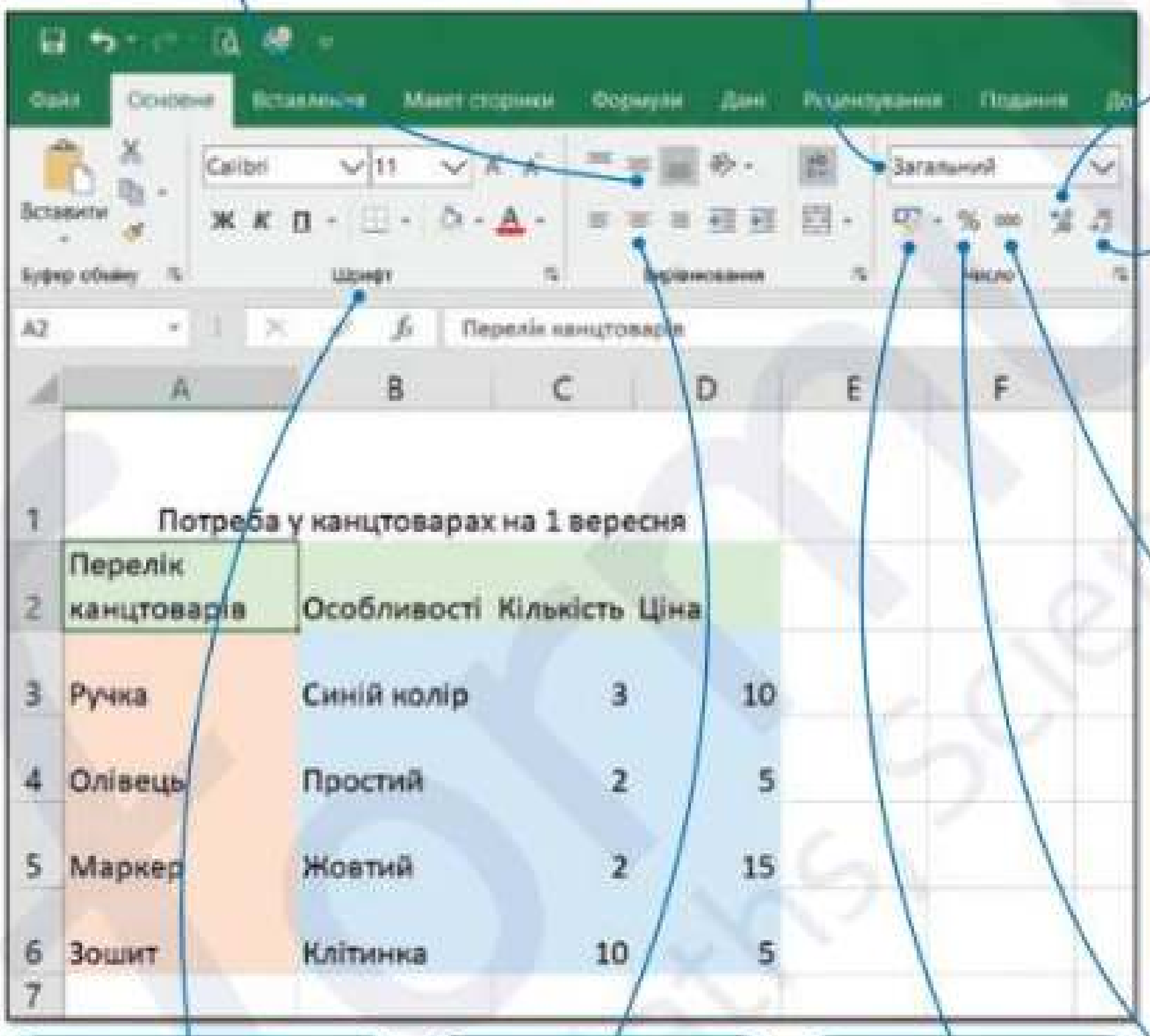
40 40,000

Зменшує кількість десяткових знаків.

40 40,0

Формат із роздільниками змінює відображення даних у комірці, додавши пропуски між розрядами числа. У **Microsoft Excel** роздільник налаштовується в регіональних параметрах **Панелі керування**.

40000 40 000,00



Розділ **Шрифт** використовуйте для форматування таблиці так само, як у Microsoft Word.

Вирівнює текст у комірці за лівим краєм, по центру або за правим краєм.

Відображає вміст комірки у форматі грошової одиниці.

40 40,00 ₴

Перетворює вміст комірки у відсотки.

40 4000%

Практичне завдання

Перейдіть за посиланням на сторінку інтернет-підтримки та завантажте файл-заготовку. Доповніть таблицю інформацією про країни світу. Тепер вам не треба постійно заглядати в атлас або підручник географії. Інформація буде під рукою.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 2

Додаткові параметри форматування

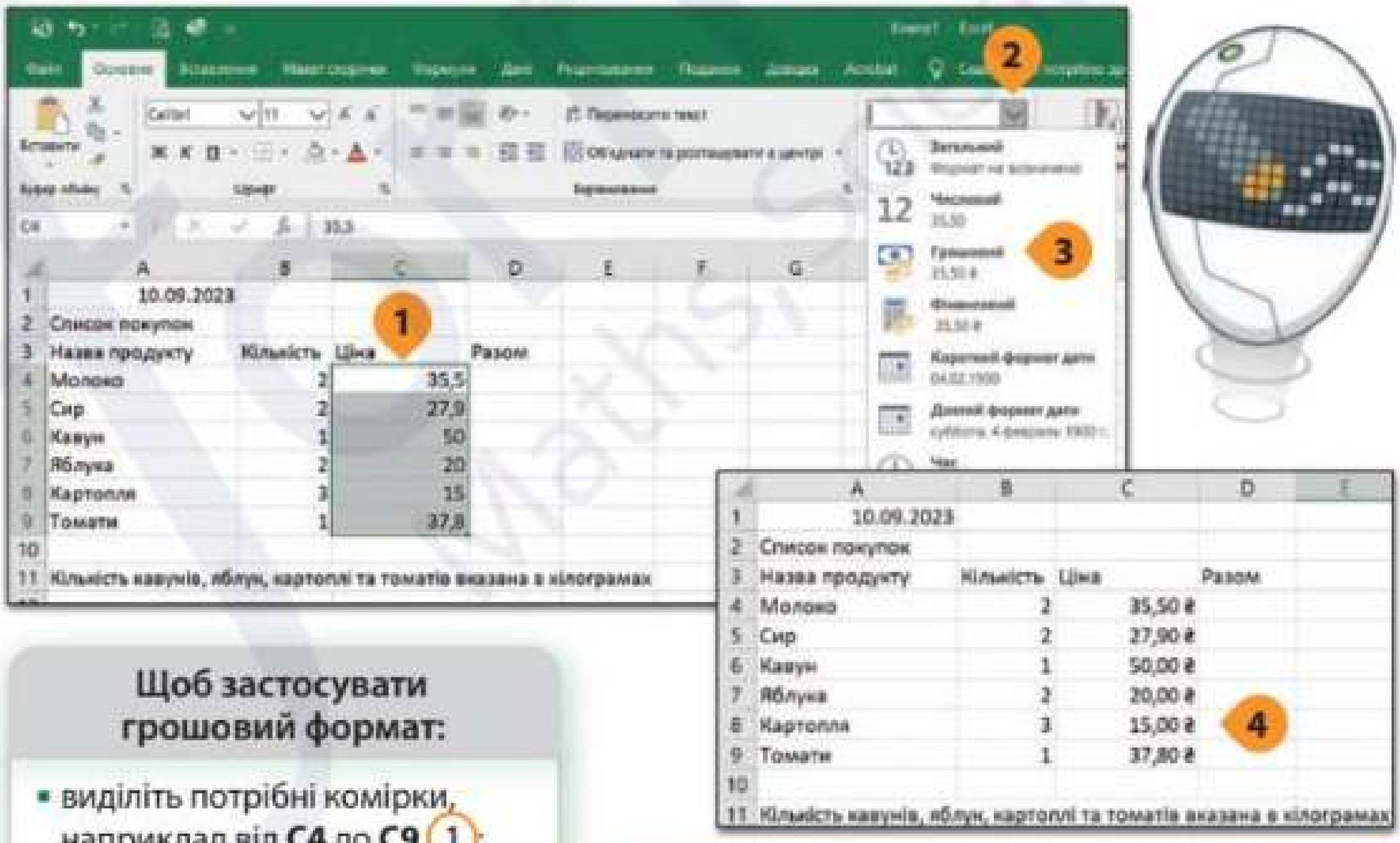
Ви вже знаєте, як створювати невелику таблицю та форматувати її, щоб вона мала належний вигляд. Застосуйте додаткові параметри форматування. Яка різниця між числовим та грошовим форматом? Чи можна в комірку вставити дату й час?

Складіть список покупок, які потрібно придбати в супермаркеті, вкажіть кількість та ціну кожного товару. Застосуйте додаткові параметри форматування.

	A	B	C	D	E
1	10.09.2023				
2	Список покупок				
3	Назва продукту	Кількість	Ціна	Разом	
4	Молоко	2	35,5		
5	Сир	2	27,9		
6	Кавун	1	50		
7	Яблука	2	20		
8	Картопля	3	15		
9	Томати	1	37,8		

Грошовий формат

Часто **Microsoft Excel** використовується саме для впорядкування даних, пов'язаних із фінансами. У таких випадках змініть формат **Числовий** на **Грошовий**.



Щоб застосувати грошовий формат:

- виділіть потрібні комірки, наприклад від **C4** до **C9** (1);
- на вкладці **Основне** в розділі **Число** в списку, що відкривається, **Числовий формат** (2) виберіть **Грошовий** (3);
- формат комірок змінено (4).

*Оберіть **Числовий**, **Грошовий**, **Фінансовий**, **Відсотковий формати**. У **Фінансовому** форматі вміст комірки вирівнюється за грошовою одиницею та десятковою комою.*

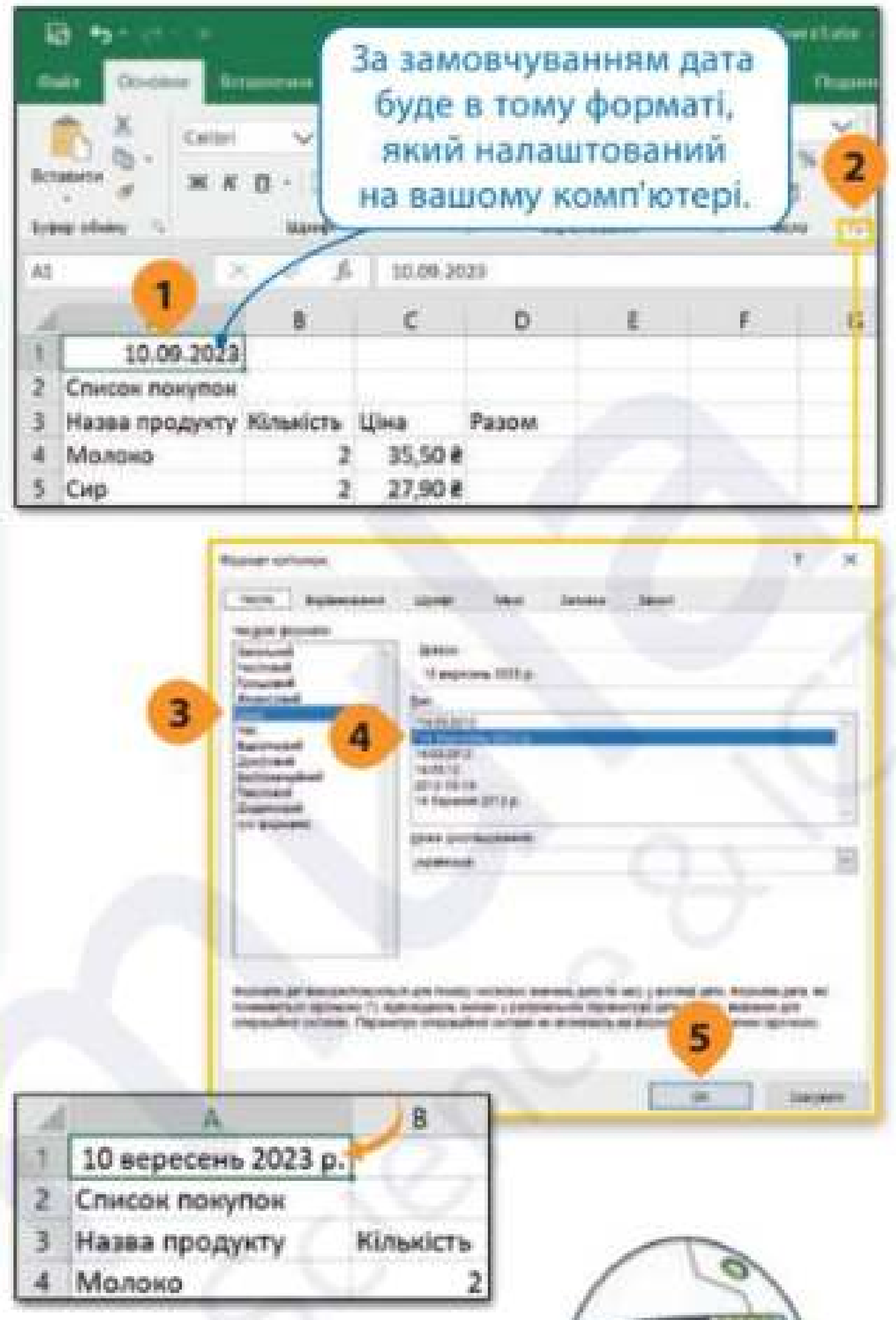
© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Дата

У **Microsoft Excel** часто використовуються дати. Програма пропонує різноманітні формати дат.

Щоб застосувати формат Дата:

- виділіть комірку, де стоїть дата, яку потрібно форматувати, наприклад **A1** (1);
- на вкладці **Основне** в розділі **Число** натисніть **Параметри форматування** (2);
- у вікні **Формат клітинок** клацніть **Дата** (3);
- у списку **Тип** (4) оберіть формат, що подобається, натисніть **ОК** (5).



В різних країнах використовують різний формат дати.
В деяких спочатку подають місяць, потім день.

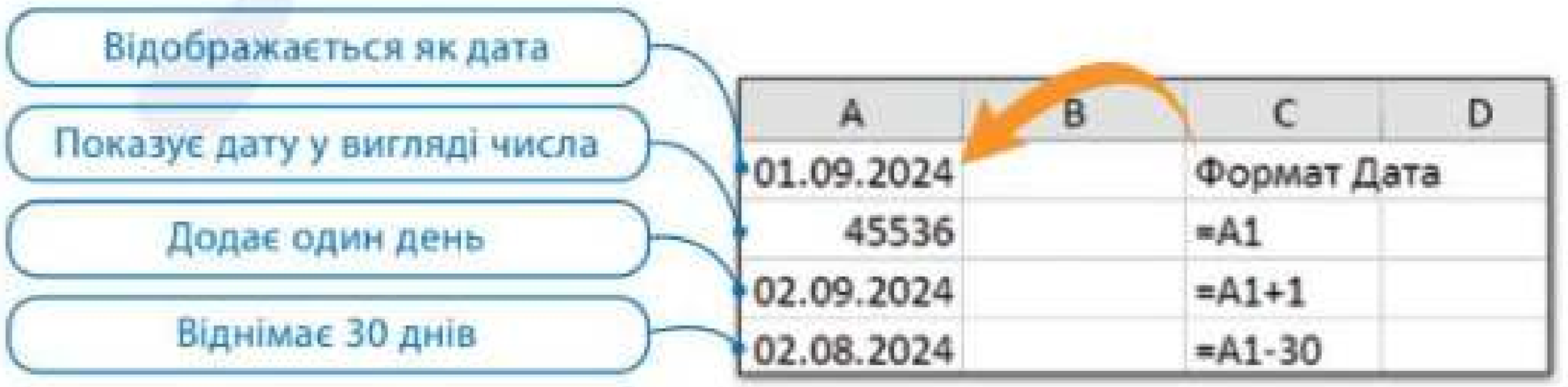
Дата й час в Microsoft Excel належать до числових форматів даних.

Програма використовує систему дат 1900, де відлік починається з 1 січня 1900 року.

Числове представлення дозволяє легко виконувати обчислення з датами, наприклад знаходити різницю між двома датами, додавати дні до дати тощо.

Числові формати дат і часу можна форматувати за допомогою вбудованих числових форматів у Microsoft Excel, таких як **Дата**, **Час**, **Дата і час** тощо.

Дати й час у програмі представлені як числа для зручності обчислень і форматування. Хоча відображаються вони в зрозумілому форматі, у програмі вони зберігаються як числові значення.



© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Час

Функція **Час** є корисним інструментом під час створення розкладів, обліку робочого часу чи інших завданнях, пов'язаних із часом.

Щоб застосувати формат Час:

- виділіть комірку, яку потрібно форматувати (1);
- на вкладці **Основне** в розділі **Число** клацніть стрілку (2);
- у списку, що відкривається, оберіть **Час** (3);
- при внесенні часу використовуйте дві крапки для розділення годин, хвилин і секунд (4).

Об'єднуємо комірки, переносимо текст і налаштовуємо вирівнювання

З попередньої теми вам відомо, як форматувати дані. Усі ці функції доступні в одному вікні.

Щоб об'єднати комірки, перенести текст та вирівняти його:

- виділіть комірки, які потрібно об'єднати, наприклад від **A2** до **D2** (1);
- на вкладці **Основне** в розділі **Число** натисніть **Параметри форматування** (2);
- у вікні **Формат клітинок** клацніть вкладку **Вирівнювання** (3);
- поставте позначки біля **перенос по словах** (4) і біля **об'єднання клітинок** (5);
- у списку, що відкривається, **по горизонталі** виберіть **по центру** (6), у спадному списку **по вертикалі** виберіть **по центру** (7);
- натисніть **ОК** (8).

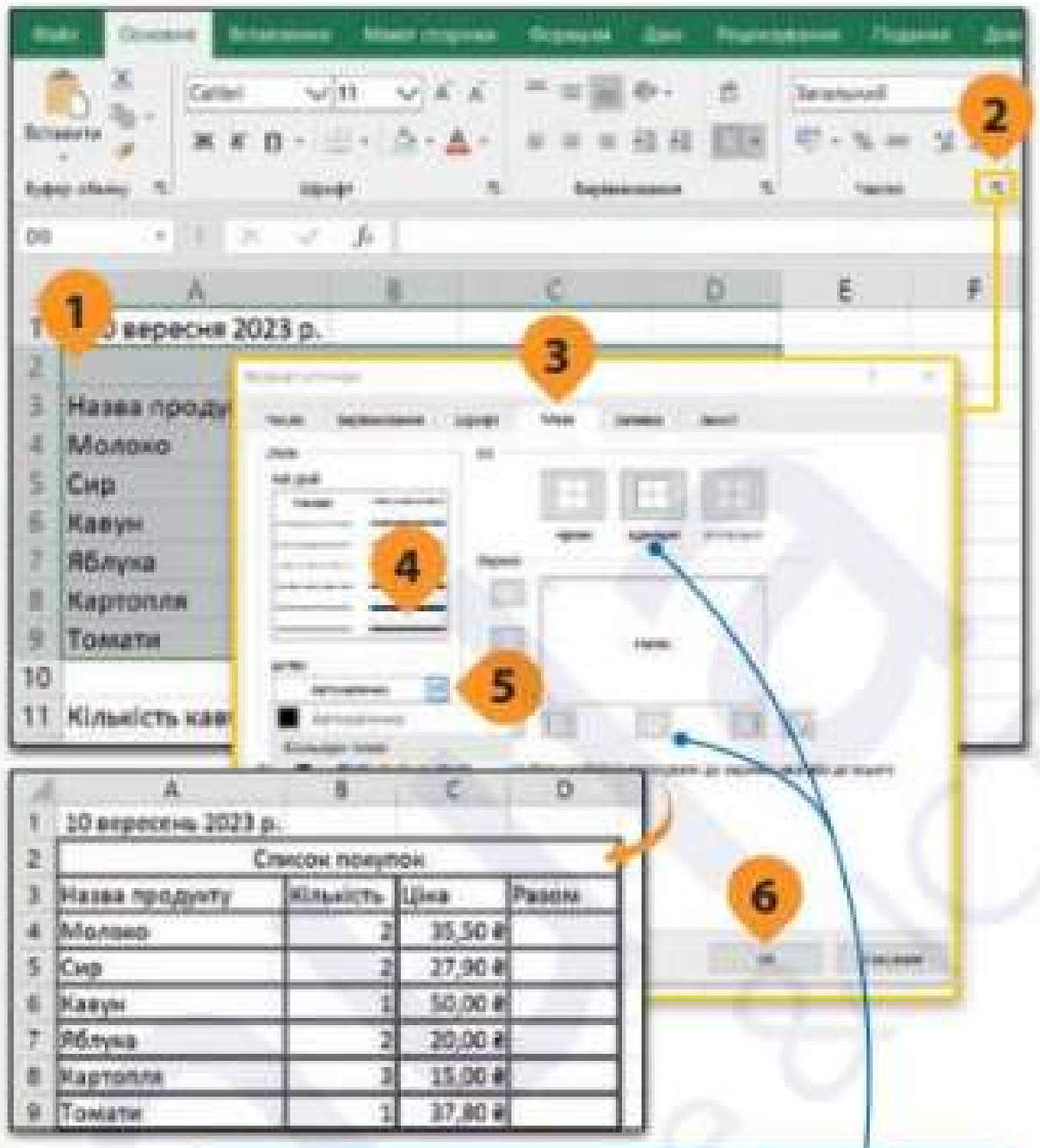
© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Межі та заливка

Форматування таблиці охоплює додавання меж і кольорів. Використовуйте межі та заливку, щоб розділити дані на групи чи зробити відмінності та подібності більш помітними.

Щоб застосувати межі:

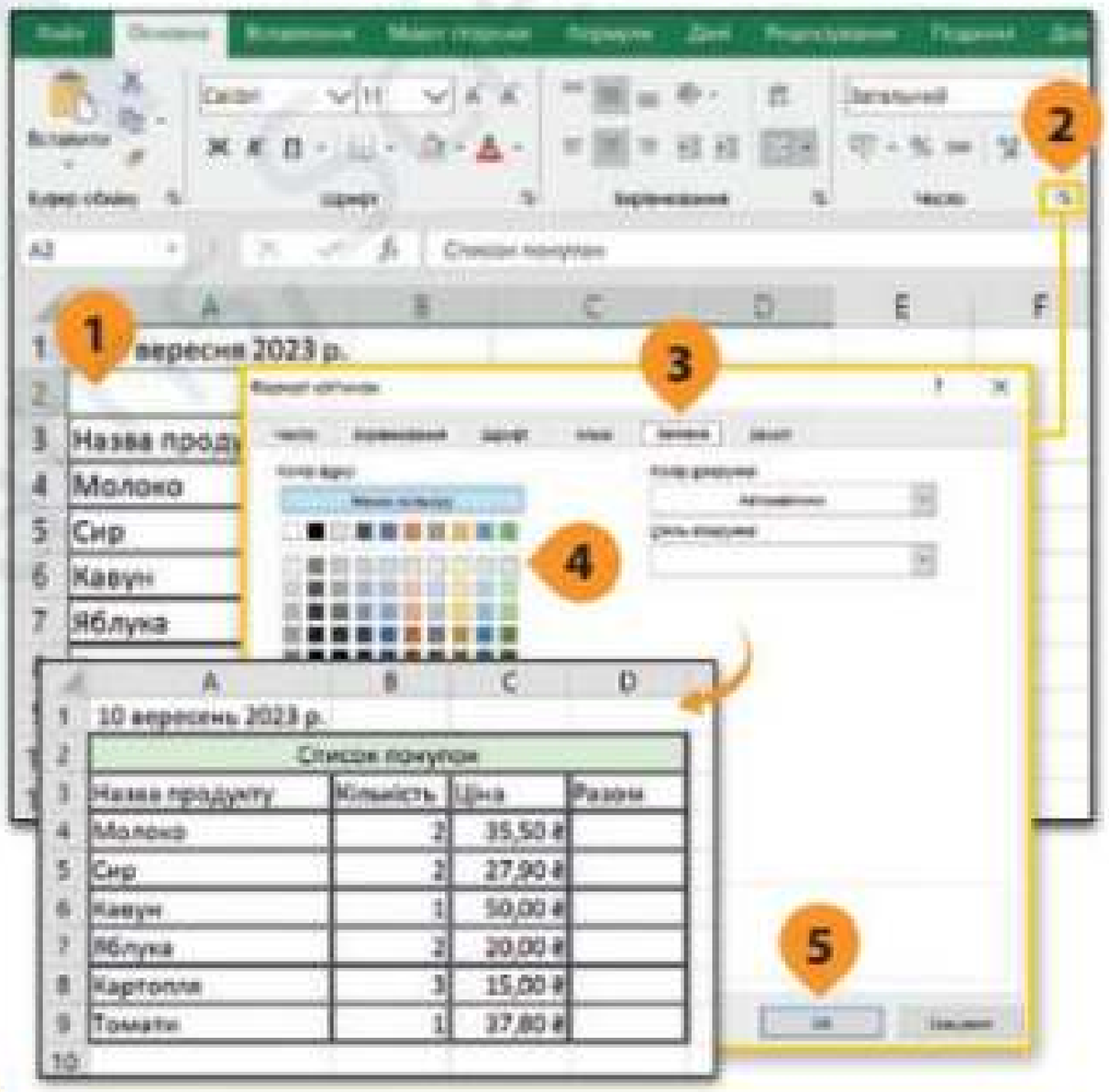
- виділіть комірки, наприклад від **A2** до **D9** (1);
- на вкладці **Основне** в розділі **Число** натисніть **Параметри форматування** (2);
- у вікні **Формат клітинок** клацніть вкладку **Межі** (3);
- у списку **тип лінії** (4) та **колір** (5) оберіть тип лінії та колір, які подобаються, натисніть **ОК** (6).



Пам'ятайте, що до таблиць можна застосувати **зовнішні**, **внутрішні** чи **окремі** межі для кожної сторони.

Щоб застосувати кольори:

- виділіть таблицю або комірку, яку будете форматувати, наприклад заголовок у комірці **A2** (1);
- на вкладці **Основне** в розділі **Число** натисніть **Параметри форматування** (2);
- у вікні **Формат клітинок** клацніть вкладку **Заливка** (3);
- виберіть потрібний колір (4), натисніть **ОК** (5).



ІСТОРІЯ

Поява **Lotus 1-2-3** в листопаді 1982 року не тільки полегшила роботу з електронними таблицями. В ній вперше було додано можливість створення діаграм та основні функціональні можливості баз даних.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Вставляємо зображення

Як і в Microsoft Word, у Microsoft Excel можна вставити зображення, щоб проілюструвати інформацію в таблиці. За допомогою інструментів у розділі **Ілюстрації** можна вставити зображення, розмістити його в потрібному місці та відформатувати.

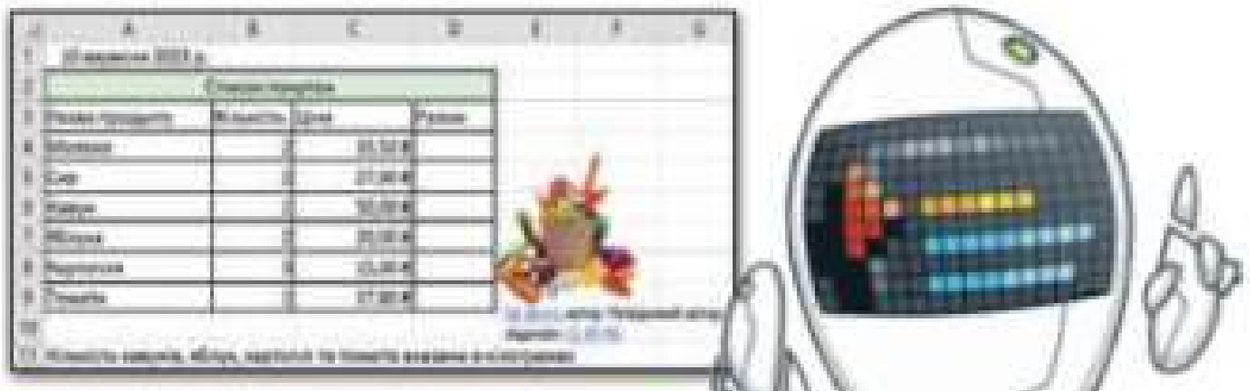
Щоб вставити зображення:

- на вкладці **Вставлення** натисніть **Ілюстрації** (1), клацніть **Зображення** (2) та оберіть **Зображення з Інтернету** (3);
- у вікні **Зображення з Інтернету** в пошуковому рядку введіть ключове слово або фразу й натисніть **Вставити** (4);
- оберіть зображення (5) та клацніть **Вставити** (6);
- обране зображення буде додано на робочий аркуш (7).

Щоб форматувати зображення:

- клацніть правою кнопкою миші на зображення та натисніть **Формат об'єкта** (1);
- панель **Формат зображення** містить безліч налаштувань для форматування зображення (2);
- наприклад, у розділі **Розмір** можна налаштувати розмір зображення.

Експериментуйте, щоб отримати бажаний результат (3).



Програма містить багато налаштувань. Чому б не скористатися їхніми перевагами? Експериментуйте!

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Перевірка даних

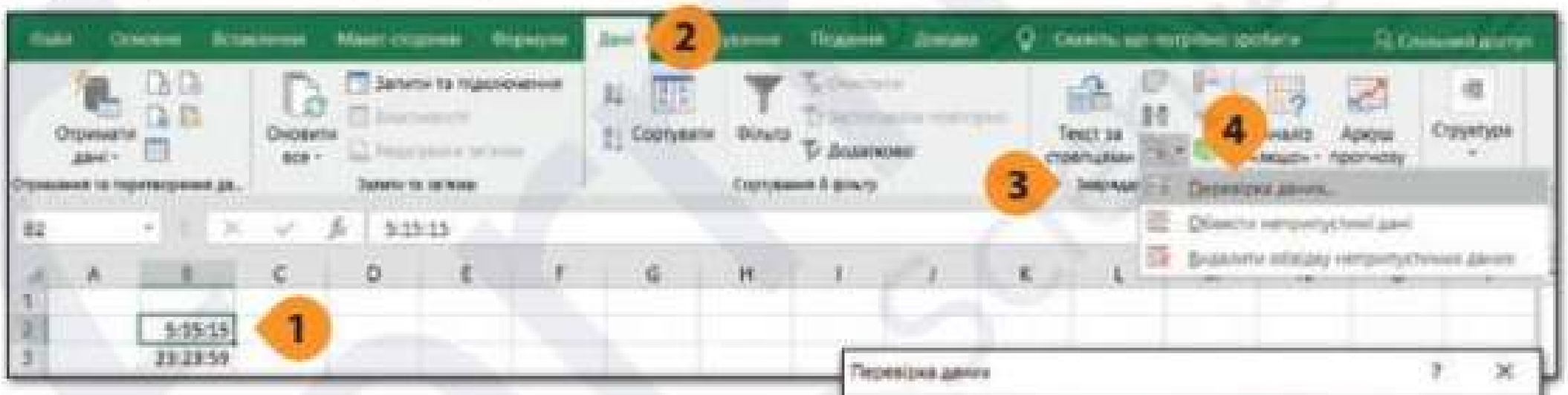
Перевірка, або **валідація даних**, – процес, який забезпечує та перевіряє якість, точність, повноту й достовірність даних у програмах і сервісах. Це будь-яка діяльність, метою якої є перевірка того, що введені дані походять із заданого набору допустимих значень і відповідають встановленим правилам.

Використовуйте перевірку даних для таких цілей:

- обмеження даних заздалегідь визначеними елементами в списку;
- обмеження чисел певним діапазоном;
- обмеження часу певним часовим проміжком;
- обмеження кількості текстових символів;
- перевірка даних на основі формул або значень в інших книгах / аркушах / клітинках.

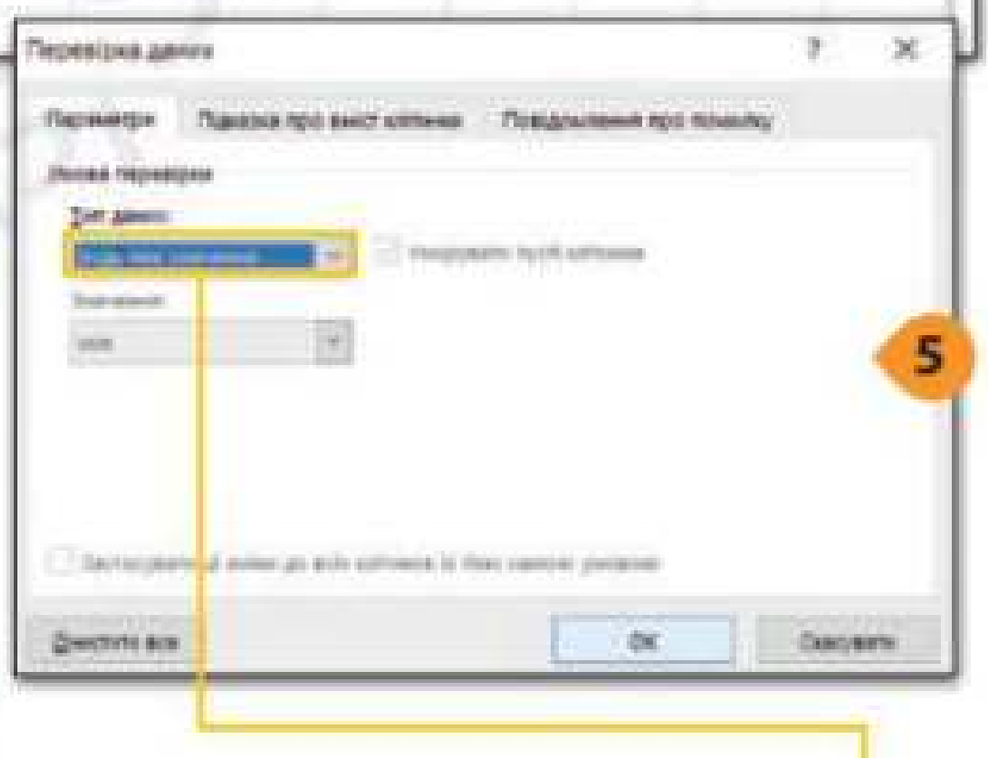
Налаштування перевірки даних

За допомогою перевірки даних можна контролювати тип даних або їхнє значення, які вводяться в клітинку. Таким чином можна запобігти введенню користувачами недостовірних даних.



Щоб перевірити дані:

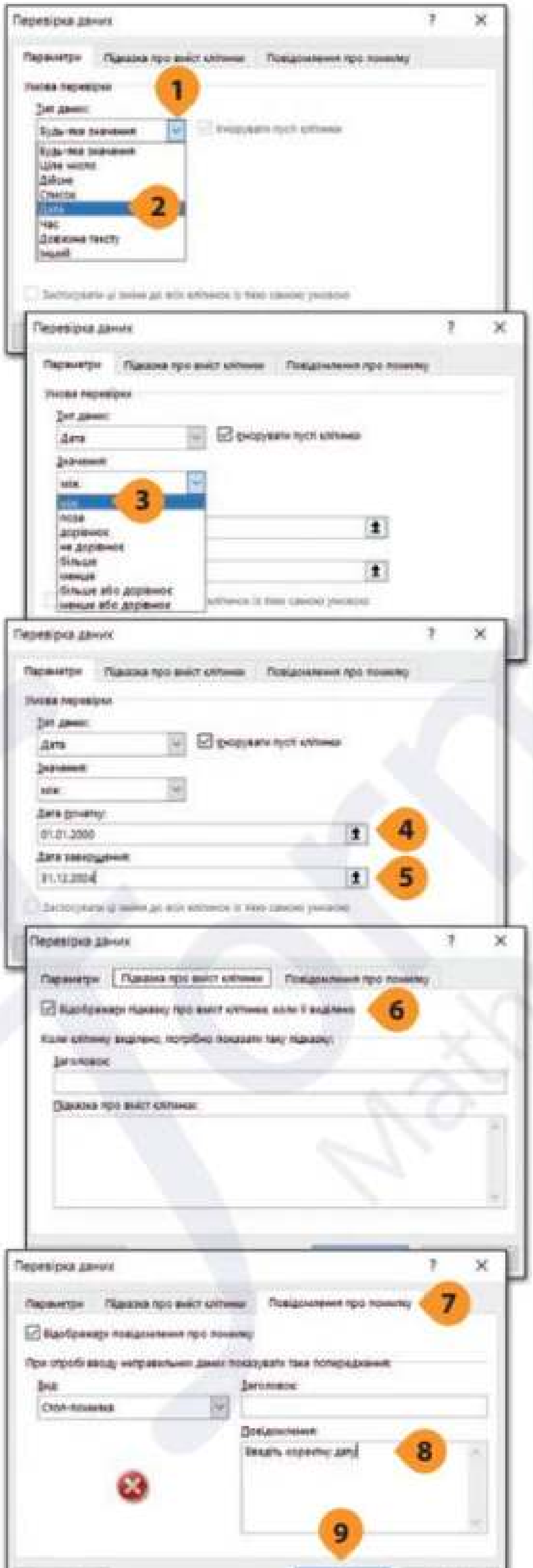
- оберіть клітинку (1), де треба це зробити;
- на вкладці **Дані** (2) в розділі **Знаряддя даних** (3) клацніть **Перевірка даних** (4);
- відкриється вікно **Перевірка даних** (5).



Оберіть потрібний параметр у спадному меню:

- ціле число, щоб обмежити клітинку прийняттям лише цілих чисел;
- дійсне – обмежує кількість десяткових чисел у клітинці;
- список – щоб вибрати дані з розкривного списку;
- дата – щоб клітинка приймала лише дату;
- час – щоб клітинка приймала лише час;
- довжина тексту – для обмеження довжини тексту;
- інший – для настроюваної формули.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.



Щоб додати критерії перевірки даних:

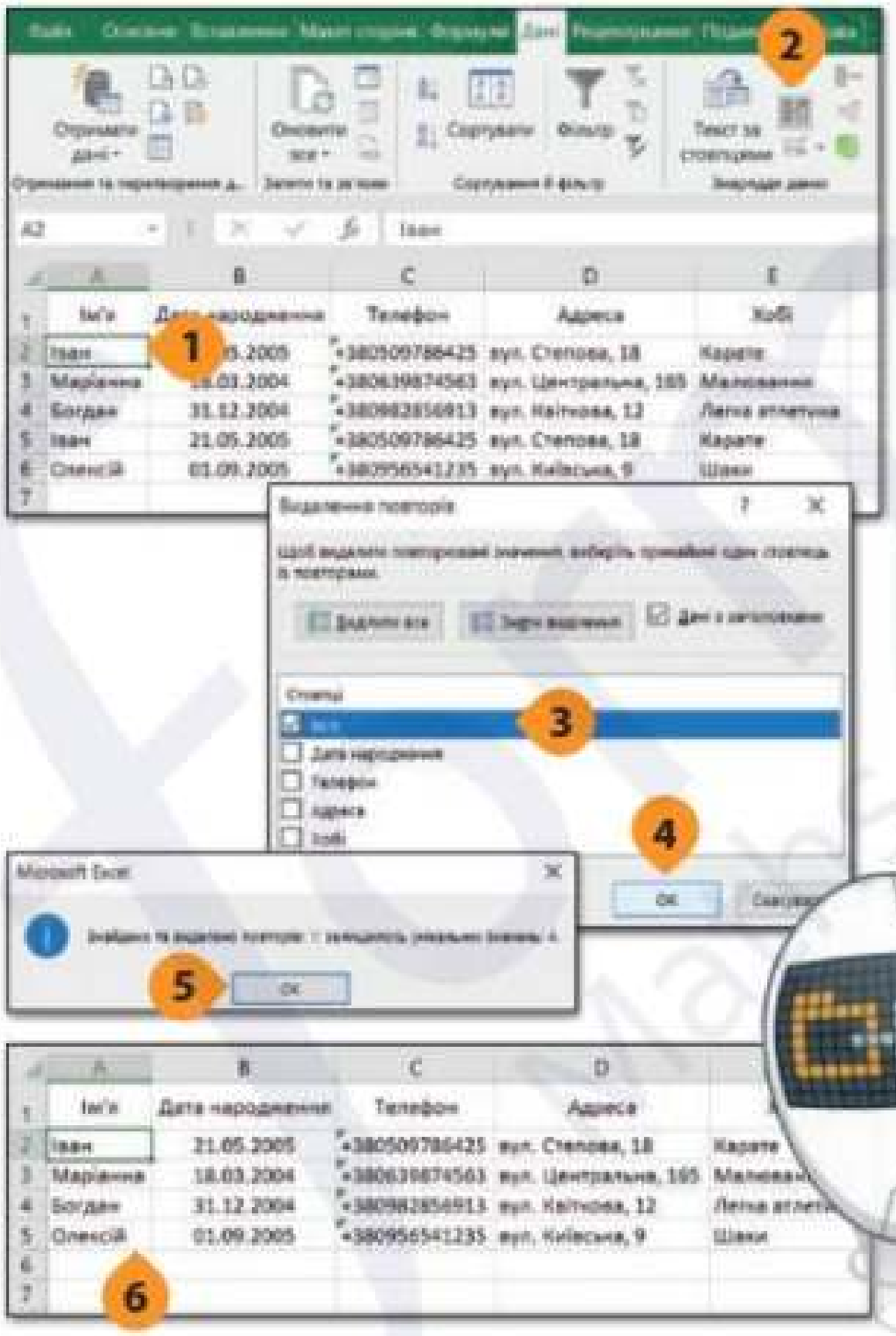
- у вікні **Перевірка даних** на вкладці **Параметри** клацніть по стрілці у вікні **Тип даних** (1) та оберіть **Дата** (2);
- у списку, що відкривається, **Значення** оберіть **між** (3) і введіть **Дата початку** (4) та **Дата завершення** (5) бажаного діапазону дат;
- на вкладці **Підказка про вміст клітинки** переконайтеся, що встановлено прапорець **Відображати підказку про вміст клітинки, коли її виділено** (6) та надрукуйте підказку про вміст клітинки;
- на вкладці **Повідомлення про помилку** (7) введіть повідомлення в текстовому полі **Повідомлення** (8), яке відображатиметься, коли будуть введені неправильні дані;
- натисніть кнопку **ОК** (9);
- у вибрану клітинку введіть дату. Якщо з'явиться повідомлення про помилку, то введена дата неправильна (10). Клацніть **Повторити** (11) та введіть правильну дату.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Видалення повторів

Під час роботи з електронними таблицями можна випадково скопіювати рядки або створити електронну таблицю, що складається з кількох різних таблиць. Це може призвести до появи повторюваних рядків, які потрібно видалити. Excel дає можливість швидко та легко видаляти дублікати.

Коли видаляються повторювані значення, це впливає лише на значення в діапазоні клітинок або таблиці. Будь-які інші значення за межами діапазону клітинок або таблиці не змінюються і не переміщуються.



Щоб видалити повторювані значення з таблиці з даними:

- виділіть діапазон клітинок або будь-яку клітинку в таблиці, з якої потрібно видалити дублікати (1);
- на вкладці **Дані** в групі **Знаряддя даних** клацніть **Видалити повтори** (2);
- у вікні **Видалення повторів** виберіть стовпчик, із якого треба видалити дублікати (3), та клацніть **ОК** (4);
- з'явиться вікно з повідомленням про вилучені дублікати та унікальність даних, що залишилися. Натисніть **ОК** (5);
- таблиця або діапазон клітинок тепер містить унікальні значення даних (6).

*Якщо видалити повтори за допомогою функції **Видалити дублікати**, дані видаляються без можливості автоматичного відновлення. Тому перед видаленням дублікатів збережіть копію вихідних даних на іншому аркуші чи файлі.*

Практичне завдання

Перейдіть за посиланням на сторінку інтернет-підтримки та завантажте файли-заготовки, відформатуйте таблицю, як запропоновано в інструкції, доповніть її зображеннями.

© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 3
Прості обчислення

Настав час спробувати, як виконати прості обчислення. У **Microsoft Excel** можна додавати, віднімати, множити та ділити числа.

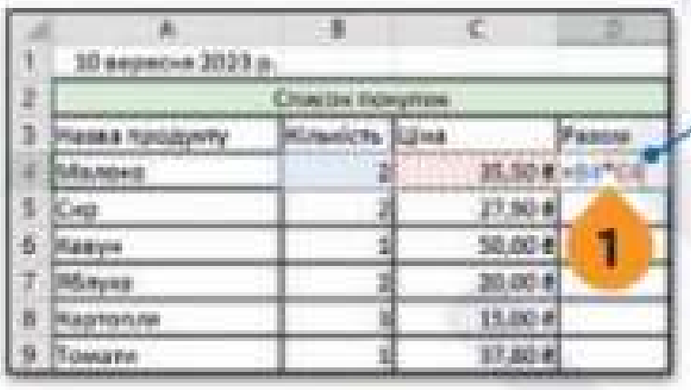
Знайдіть та відкрийте електронну таблицю, яку створили під час вивчення попередньої теми. Як дізнатися, скільки потрібно заплатити за молоко?

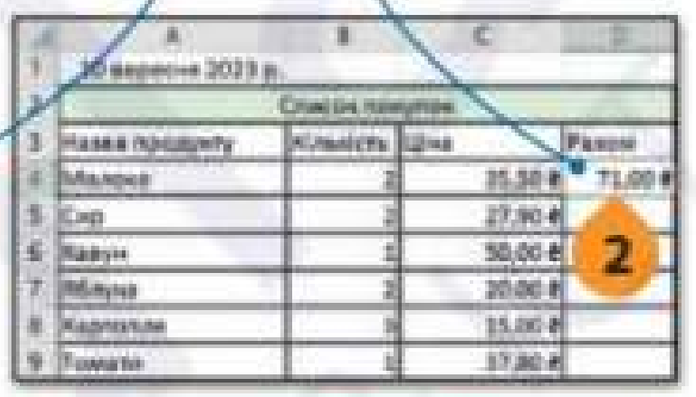
Щоб помножити числа:

- виберіть комірку **D4**, введіть **=B4*C4** та натисніть Enter ↵ (1);
- у результаті отримаємо вартість молока (2).

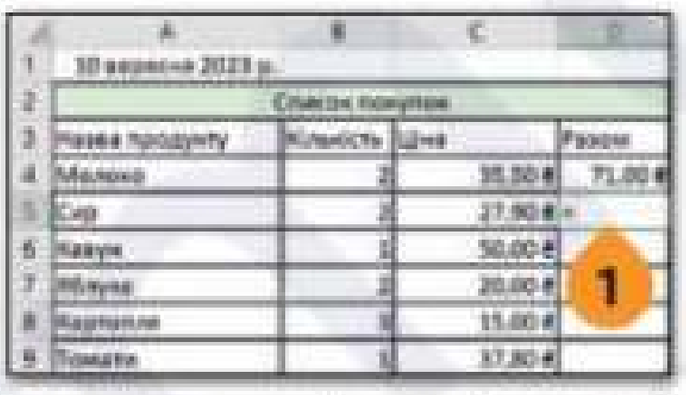
Зробіть те саме із Сиром, але замість того, щоб вводити назви комірок, натискайте на них:

- натисніть комірку **D5** та введіть = (1);
- клацніть комірку **B5** (2), введіть * та натисніть комірку **C5** (3);
- клацніть Enter ↵, щоб отримати результат (4).

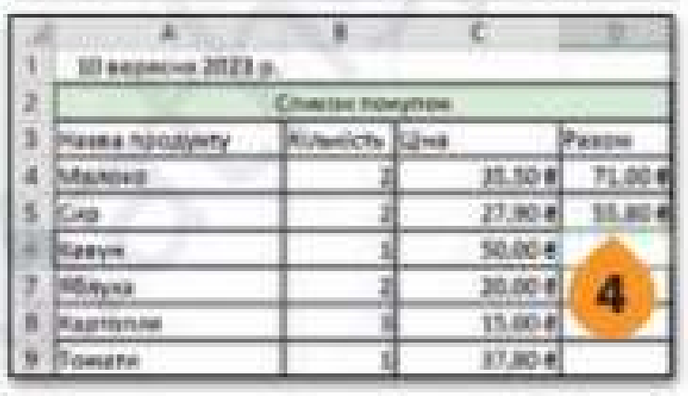




Результат множення **71**, він з'явиться в комірці **D4**. Обчислення, яке ви ввели **=B4*C4** є **формулою**.



Будь-яка формула в електронних таблицях завжди починається знаком (=).



Тепер змініть числа в комірках **B4** *та* **C5**. *Зверніть увагу на те, як змінилися значення в комірках. Електронна таблиця цікава тим, що, якщо змінюєте числа в певних комірках, результати в комірках з формулами також автоматично змінюються!*

Основні розрахунки та їхні символи в

=B5*C5	=B5/C5	=B5+C5	=B5–C5

ПІДКАЗКА

Пам'ятайте, що спочатку у формулі виконується дія множення та ділення, а тоді – додавання та віднімання. Порядок дій у формулі (так само як у математиці) регулюється дужками.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Автозаповнення

Якщо необхідно обчислити суму будь-якого набору даних, наприклад вартість решти продуктів, не потрібно знову виконувати ті самі дії. Є швидший спосіб обчислити суми кількох наборів даних, він називається **Автозаповнення**.

Щоб використати Автозаповнення:

- натисніть комірку **D5**. У ній уже обчислена вартість другого товару зі списку (1);
- у правому нижньому куті меж комірки є маленький квадрат. Він називається **маркером заповнення**. Наведіть на нього вказівник миші, який змінить вигляд на знак + (2);
- утримуйте ліву кнопку миші й тягніть вказівник миші до комірки **D9** (3);
- усі суми будуть обчислені (4).

	A	B	C	D
1	10 вересня 2023 р.			
2	Список покупок			
3	Назва продукту	Кількість	Ціна	Разом
4	Молоко	2	35,50 ₴	71,00 ₴
5	Сир	2	27,90 ₴	55,80 ₴
6	Кавун	1	50,00 ₴	
7	Яблука	2	20,00 ₴	
8	Картопля	3	15,00 ₴	
9	Томати	1	37,80 ₴	

1

2

3

4

Формули з функціями

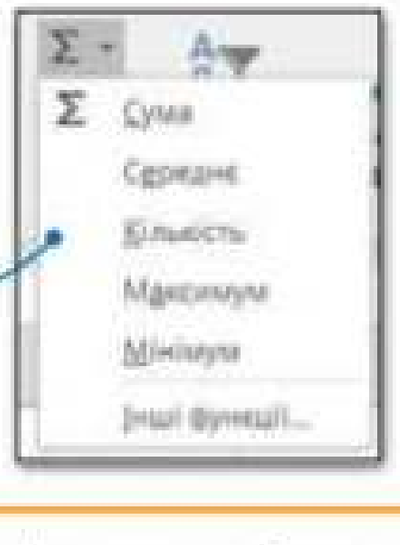
В електронних таблицях дуже зручно виконувати складні обчислення швидко й точно. Наприклад, якщо ви хочете обчислити середню вартість придбаних товарів, слід використати таку формулу:

=(D4+D5+D6+D7+D8+D9)/6.

Складно, чи не так? Щоб спростити обчислення, **Microsoft Excel** пропонує безліч готових формул, які називають функціями, вони виконають замість вас всю роботу.

Найчастіше використовуються такі функції:
Сума – обчислює суму значень вибраних чисел. **Середнє** – обчислює середнє арифметичне вибраних комірок. **Максимум** – визначає найбільше значення серед вибраних чисел. **Мінімум** – визначає найменше значення серед вибраних чисел.

*Пам'ятайте, що інструмент **Автозаповнення** копіює не лише формулу, але й формат комірки.*



ПІДКАЗКА

Зверніть увагу, що у формулах Microsoft Excel можуть бути лише латинські букви. Тому, якщо вводите формулу, використовуючи літери українського алфавіту, програма може не розпізнати їх, а результат матиме такий вигляд: **#NAME?**

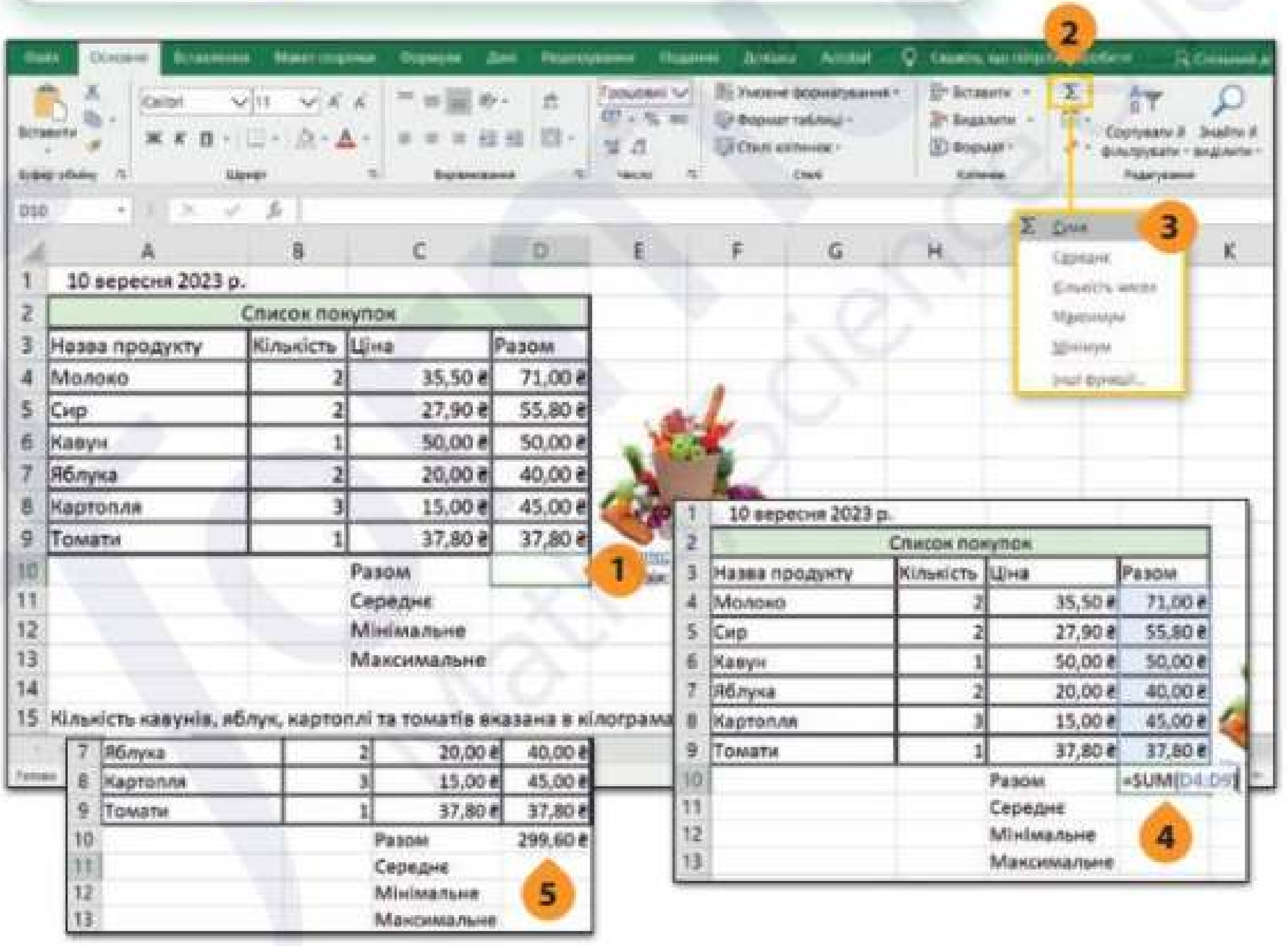
© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Сума

Сума – результат додавання значень діапазону комірок. Якщо потрібно визначити суму значень великої кількості комірок, застосуйте цю функцію, замість того щоб по черзі додавати числа з кожної комірки.

Щоб обчислити суму:

- натисніть комірку, у якій має відображатися сума, наприклад комірка **D10** (1);
- на вкладці **Основне** в розділі **Редагування** натисніть на стрілку біля **Σ** (2);
- клацніть **Сума** (3);
- оберіть комірки, які треба додати, наприклад від **D4** до **D9** (4);
- натисніть Enter. Результат обчислення з'явиться в комірці **D10** (5). Це легко, чи не так?



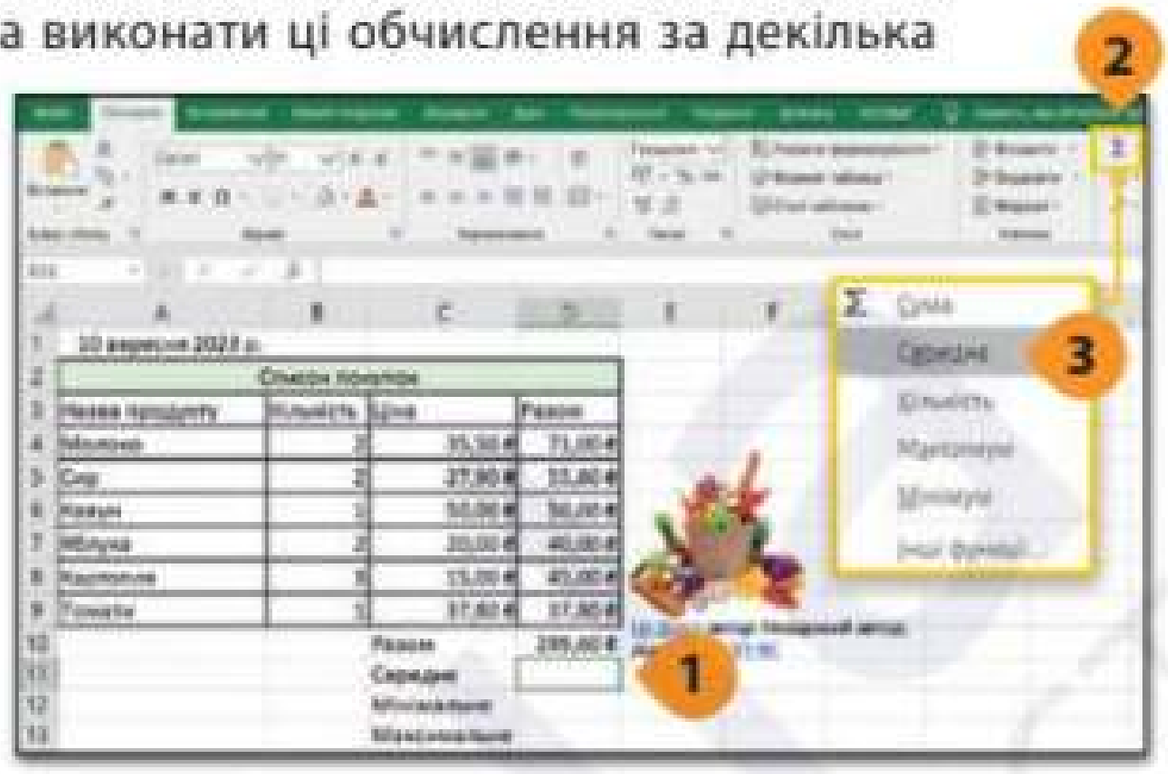
ПІДКАЗКА

Замість того щоб вибирати функції або значення за допомогою миші, їх можна вводити за допомогою клавіатури. Наприклад, якщо потрібно обчислити суму в наведеному вище прикладі, можна ввести **=SUM(D4:D9)** і натиснути Enter. Так само можна робити з іншими функціями. Просто замініть **SUM** на **AVERAGE**, **MIN** або **MAX**.

© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Середнє арифметичне

Формула **Середнє** виводить середнє арифметичне значень вибраного діапазону комірок. Обчислити середнє арифметичне вручну не так легко, але за допомогою цієї формули можна виконати ці обчислення за декілька клацань.



Щоб обчислити Середнє:

- натисніть комірку, у якій буде відображатися середнє арифметичне, наприклад комірку **D11** (1);
- на вкладці **Основне** в розділі **Редагування** натисніть на стрілку біля **Σ** (2);
- клацніть **Середнє** (3);
- виберіть комірки, де буде обчислене середнє значення, наприклад від **D4** до **D9** (4);
- натисніть Enter ↵ (5).

Зверніть увагу, багато функцій мають подібну структуру. Вивчіть одну, і легше буде зрозуміти інші, але кожну функцію потрібно розглядати окремо.

Максимум та Мінімум

Функції можна використовувати для порівняння чисел. Функція **Мінімум** дає можливість знайти найменше значення серед вибраних чисел, а **Максимум** – найбільше. Ці функції працюють так само, як **Сума** або **Середнє**.

Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та проведіть дослідження, а також обчисліть середню температуру у вказаних країнах.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 4
Складні розрахунки

Ви вже знаєте, як робити прості обчислення за допомогою Microsoft Excel. А як щодо складного алгебраїчного виразу? Зробимо складні речі простими.

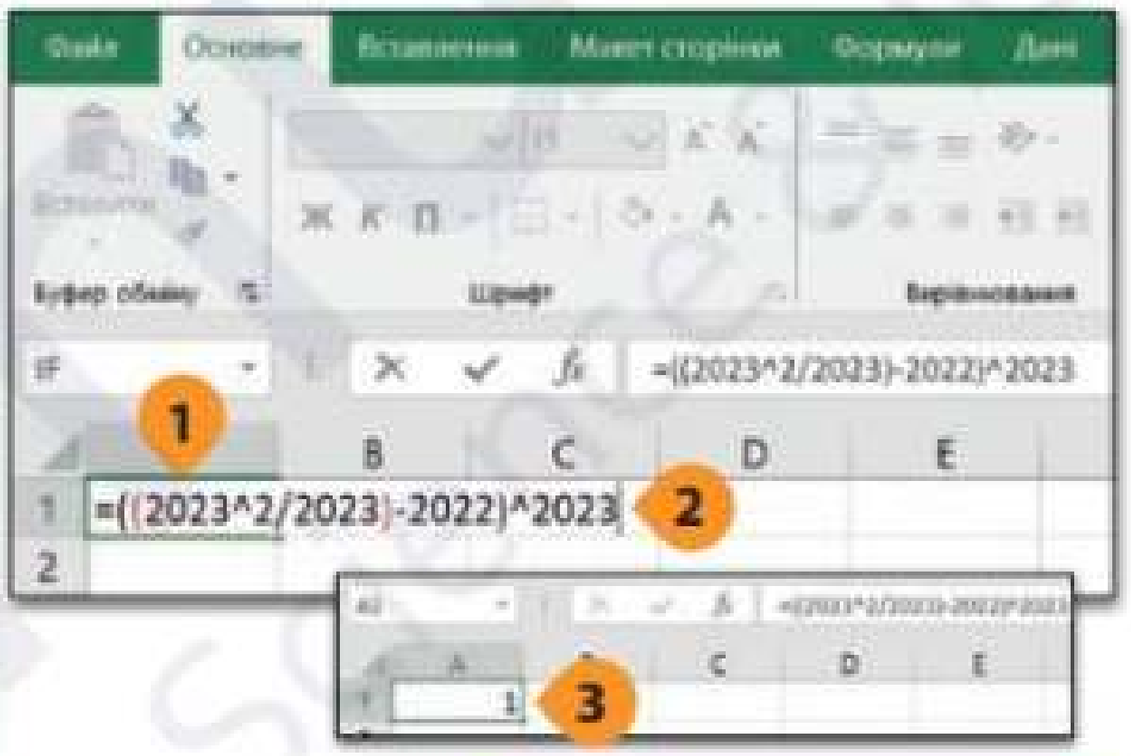
Порядок дій

Як ви обчислюєте такі вирази на уроці алгебри?

$$\left(\frac{2023^2}{2023} - 2022\right)^{2023}$$

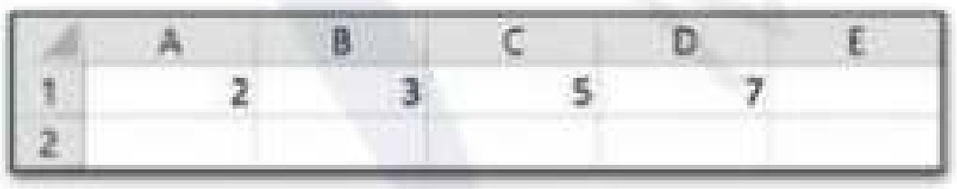
Щоб обчислити такий вираз в Excel:

- на робочому аркуші натисніть комірку **A1** (1);
- щоб почати обчислення, введіть **=**;
- тут же введіть математичну формулу **((2023^2/2023)–2022)^2023** (2);
- натисніть **Ctrl** + **Enter**, щоб обчислити формулу та залишитися в тій самій комірці (3).



Спробуємо інший спосіб! Цього разу напишемо формулу, яка буде містити посилання на комірку. Таким чином, ви отримаєте результат, який зміниться, якщо дані в комірках, на які посилаються, також зміняться.

Введіть наступні числа на новому робочому аркуші.



Для обчислення виразу ((C1^A1)–(D1^B1))+((B1/A1)*A1)^2:

- натисніть на комірку **A3** і введіть **=((C1^A1)–(D1^B1))+((B1/A1)*A1)^2** (1);
- натисніть **Ctrl** + **Enter**, щоб обчислити формулу та залишитися в тій самій комірці (2).

Будь-які обчислення або формули в Microsoft Excel завжди починаються зі знака =.

Коли ви натискаєте на комірку, у **Рядку формул** відображаються всі адреси комірок, на які стоять посилання в обраній комірці.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Працюємо з відсотками

Робота з відсотками може викликати плутанину. Але, якщо зрозуміти послідовність дій, все стане на свої місця.

Створіть наведену таблицю:

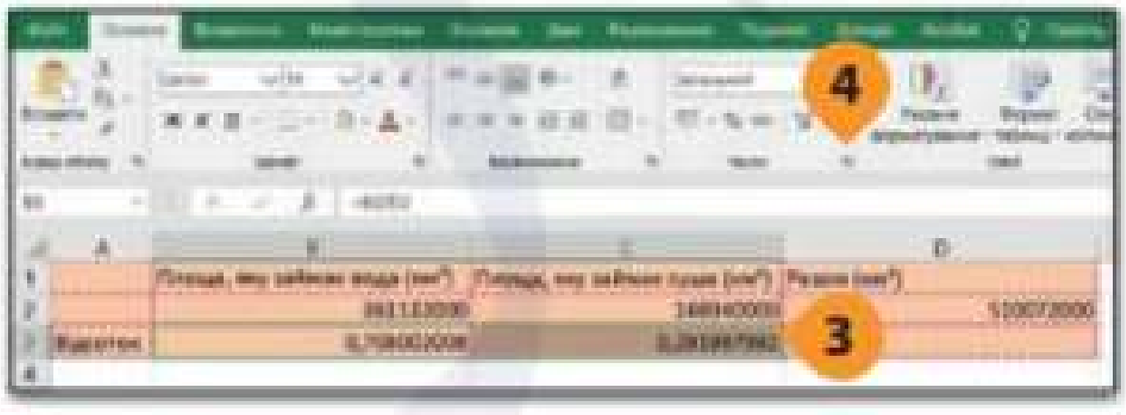
	A	B	C	D
1		Площа, яку займає вода (км²)	Площа, яку займає суша (км²)	Разом (км²)
2		361132000	148940000	510072000
3	Відсоток			
4				

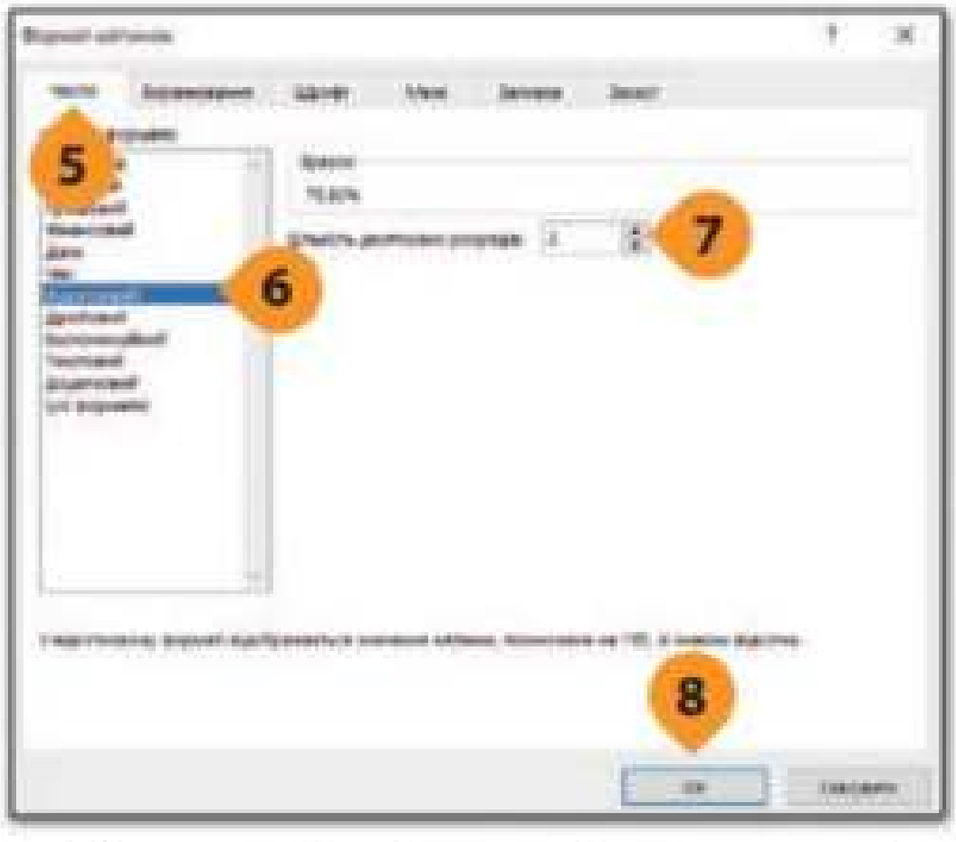
Щоб перевести число у відсотки:

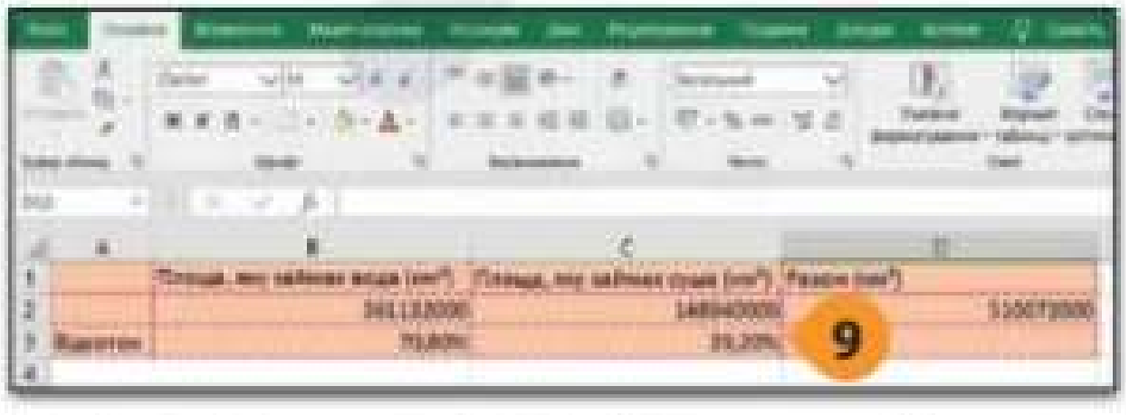
- натисніть комірку **B3** і введіть **=B2/D2** та натисніть Enter ↵ (1);
- натисніть комірку **C3** і введіть **=C2/D2** й натисніть Enter ↵ (2);
- виділіть комірки, що містять значення, які потрібно перетворити, в поданому випадку **B3** та **C3** (3);
- на вкладці **Основне** в розділі **Число** натисніть кнопку **Додаткових параметрів** (4);
- у вікні **Формат клітинок** натисніть вкладку **Число** (5);
- у списку **Числові формати** виберіть **Відсотковий** (6);
- введіть число в текстовому полі **Кількість десяткових розрядів**, наприклад **2** (7);
- натисніть **ОК** (8);
- числа в комірках **B3** та **C3** тепер відображаються у відсотках (9).

*Ви можете змінити відображене значення з десяткового числа на відсоток, застосувавши **Відсотковий формат**. Microsoft Excel множить число в комірці на 100 й відображає результат зі знаком відсотка.*

Ви також можете застосувати формат відсотків, натиснувши кнопку % в розділі **Число** на вкладці **Основне**.





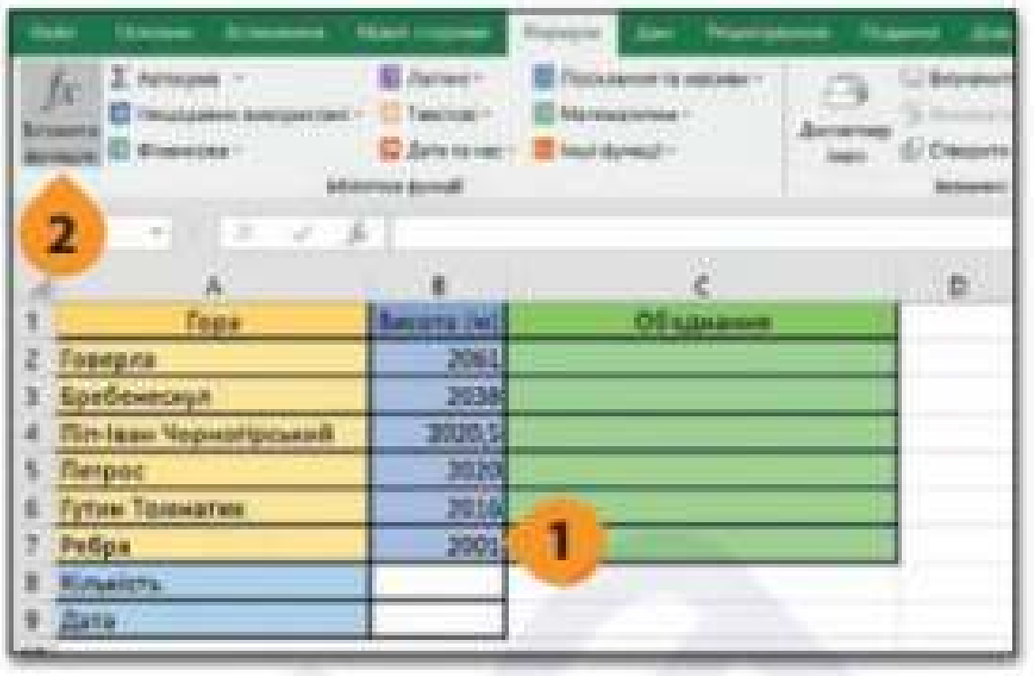


© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Працюємо з текстом і датою

Як відомо, **Microsoft Excel** може допомогти розрахувати та проаналізувати інформацію за допомогою різноманітних функцій. Вони працюють із різними типами даних (числа, текст, дата й час, посилання), що дозволяє виконувати різноманітні обчислення та аналіз.

Ознайомимося з деякими за допомогою висоти вершин Українських Карпат.



Функція COUNT

Ця функція підраховує кількість комірок, що містять числа.

Щоб підрахувати кількість комірок із числами:

- натисніть на комірку, де ви хочете побачити результат, наприклад комірку **B7** (1);
- на вкладці **Формули** в розділі **Бібліотека функцій** натисніть **Вставити функцію** (2);
- у вікні **Вставлення функції** у спадному списку **Категорія** виберіть **Статистичні** (3);
- у полі **Виберіть функцію** знайдіть **COUNT** (4) і натисніть **ОК** (5);
- у вікні **Аргументи функцій** у текстовому полі **Значення1** введіть **A1:B7** (6). Це діапазон комірок, де буде підраховано, скільки з них містять числа;
- натисніть **ОК** (7);
- кількість комірок, що містять цифри, відображатиметься в комірці **B8** (8).

Функція COUNTA

Ця функція обчислює кількість непустих клітинок у вказаному діапазоні.

	A	B
1	Ім'я	Дата народження
2	Іван	21.05.2005
3	Маріанна	
4	Богдан	31.12.2004
5	Іван	21.05.2005
6	Олексій	01.09.2005
7		
8		=COUNTA(B2:B6)

Дата народження
21.05.2005
31.12.2004
21.05.2005
01.09.2005
4

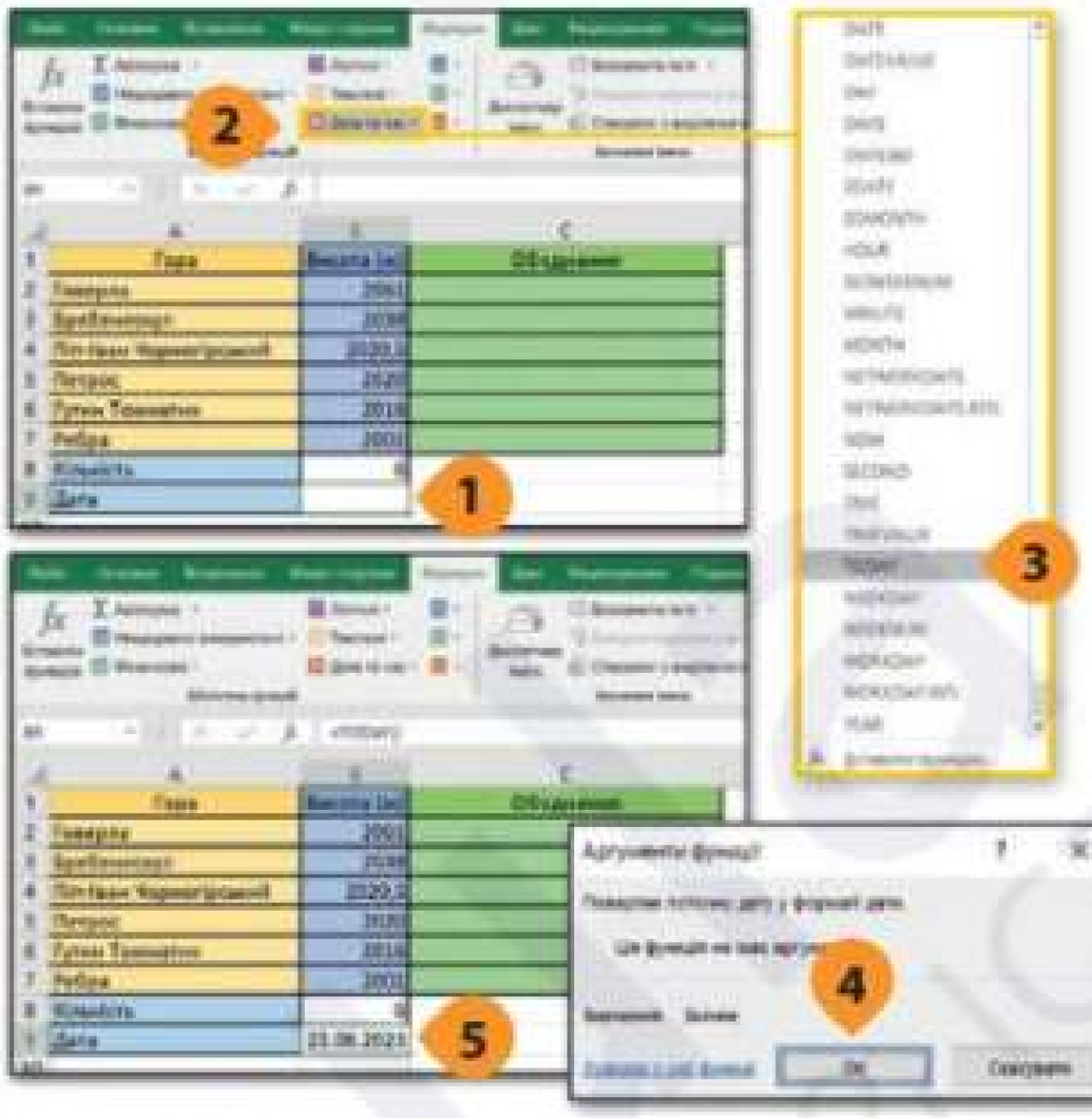
© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Функція TODAY

Часто в документах потрібно, щоб на аркуші відображалася поточна дата. Скористайтеся функцією **TODAY**.

Щоб вказати поточну дату:

- натисніть на комірку, де ви хочете побачити результат, у цьому прикладі комірку **B9** (1);
- на вкладці **Формули** в розділі **Бібліотека функцій** натисніть **Дата й час** (2);
- у списку оберіть **TODAY** (3);
- у вікні **Аргументи функцій**
- натисніть **ОК** (4). У комірці відображатиметься поточна дата, встановлена на комп'ютері (5).

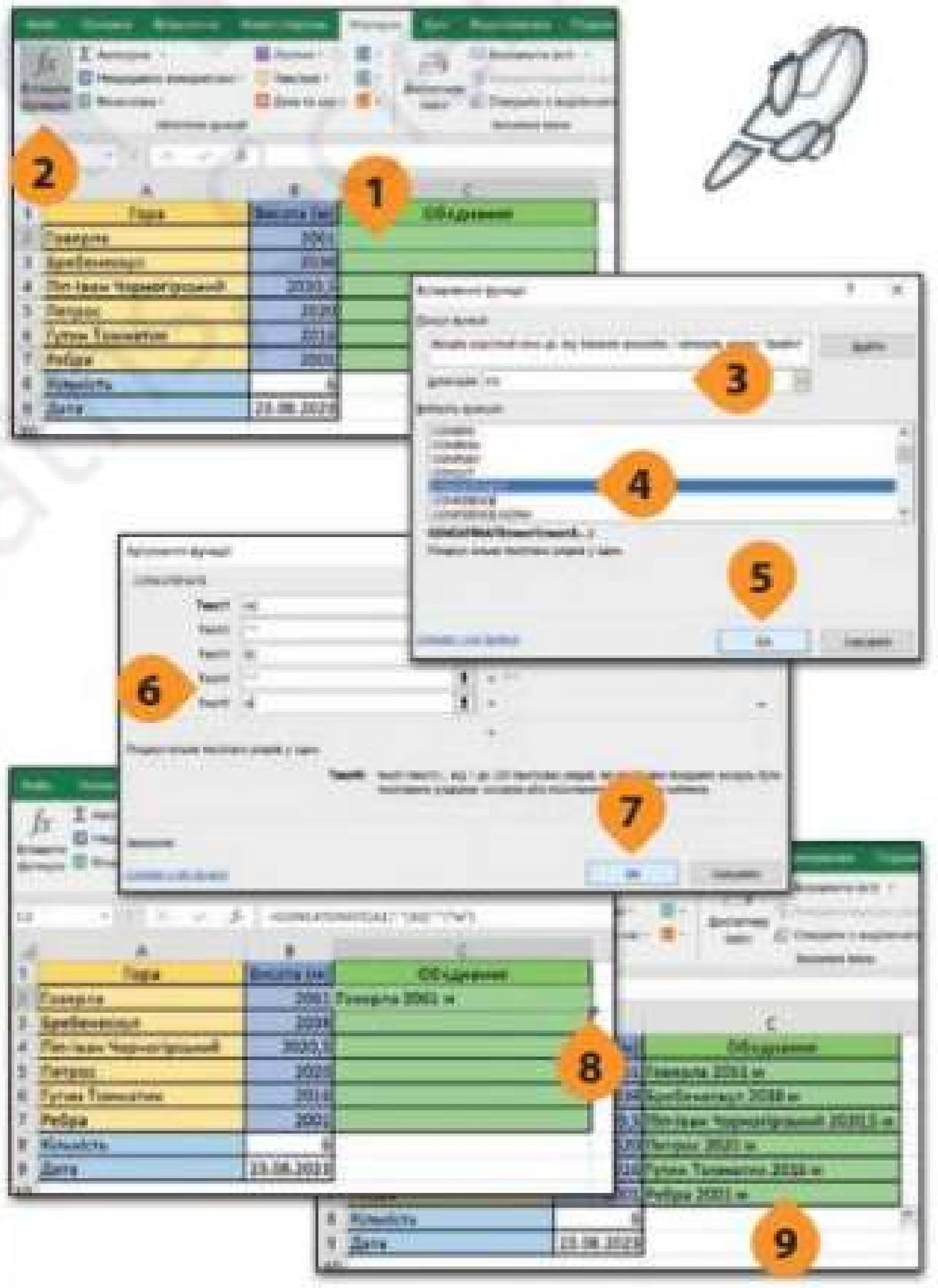


Функція CONCATENATE

Іноді треба об'єднати текстовий вміст комірок, наприклад ім'я та прізвище. Можна скористатися функцією **CONCATENATE**.

Щоб об'єднати вміст комірок:

- натисніть на комірку **C2** (1);
- на вкладці **Формули** в розділі **Бібліотека функцій** натисніть **Вставити функцію** (2);
- у вікні **Вставлення функції** в спадному списку **Категорія** натисніть кнопку **Усі** (3);
- оберіть **CONCATENATE** (4);
- натисніть **ОК** (5);
- у вікні **Аргументи функцій** у текстовому полі **Текст1** введіть **A2**, у текстовому полі **Текст2** введіть **" "**, а в текстовому полі **Текст3** введіть **B2**, у текстовому полі **Текст4** введіть **" "**, у текстовому полі **Текст5** введіть **м** (6);
- натисніть **ОК** (7);
- за допомогою **маркера заповнення** (8) скопіюйте функцію в інші комірки (9).



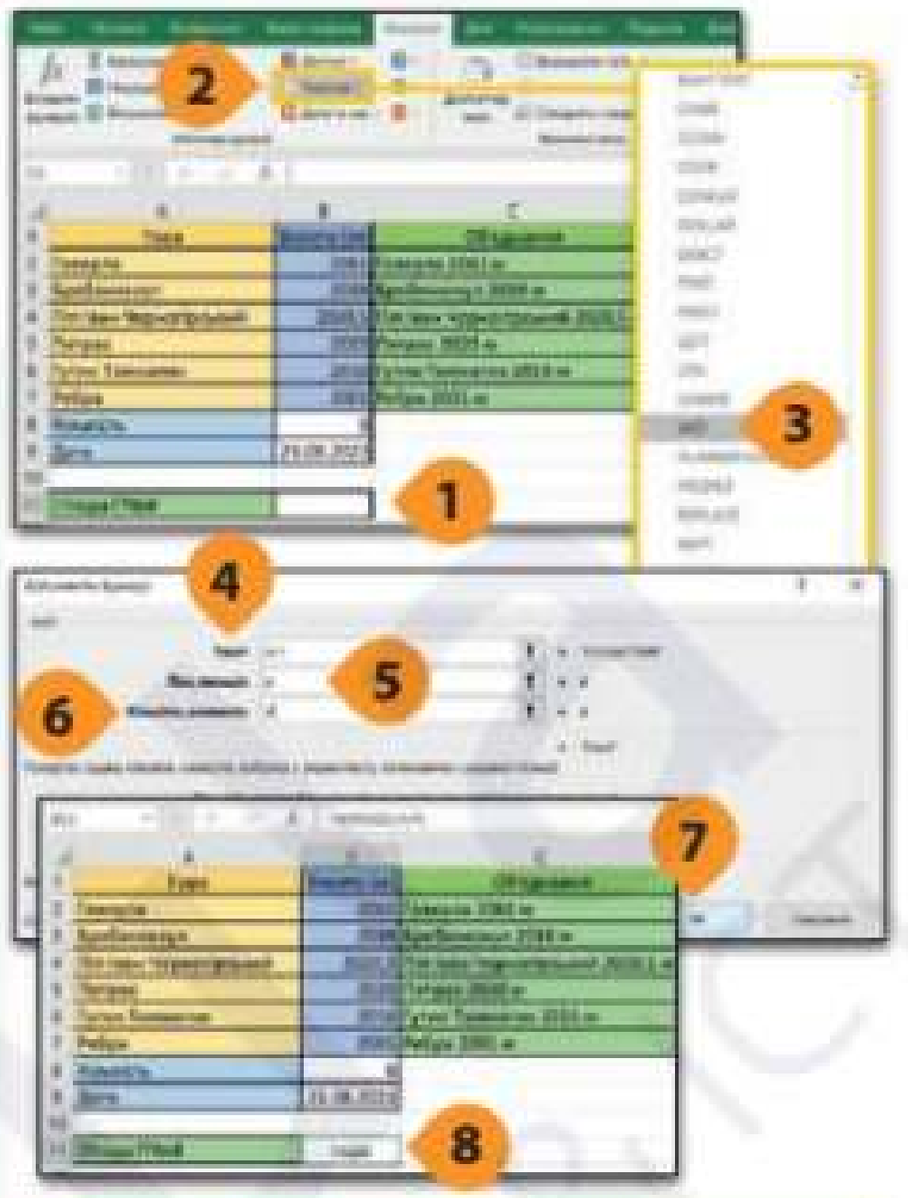
© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Функції LEFT, RIGHT, MID

Якщо вам потрібна лише частина текстового рядка (підрядок), використовуйте функції **LEFT**, **RIGHT** та **MID**.

Щоб виокремити лише частину рядка:

- натисніть на комірку **B11** (1);
- на вкладці **Формули** в розділі **Бібліотека функцій** натисніть **Текстові** (2);
- у списку оберіть **MID** (3);
- у вікні **Аргументи функцій** у полі **Текст** введіть **A11** (4). Це адреса комірки, з якої ви збираєтеся виокремити символи;
- у текстовому полі **Поч_позиція** введіть **4** (5). Це позиція першого символу, який потрібно виокремити;
- у текстовому полі **Кількість_символів** введіть **4** (6), що вказує кількість символів, які будуть виокремленні;
- натисніть **ОК** (7);
- результат функції відображається в комірці **B11** (8).



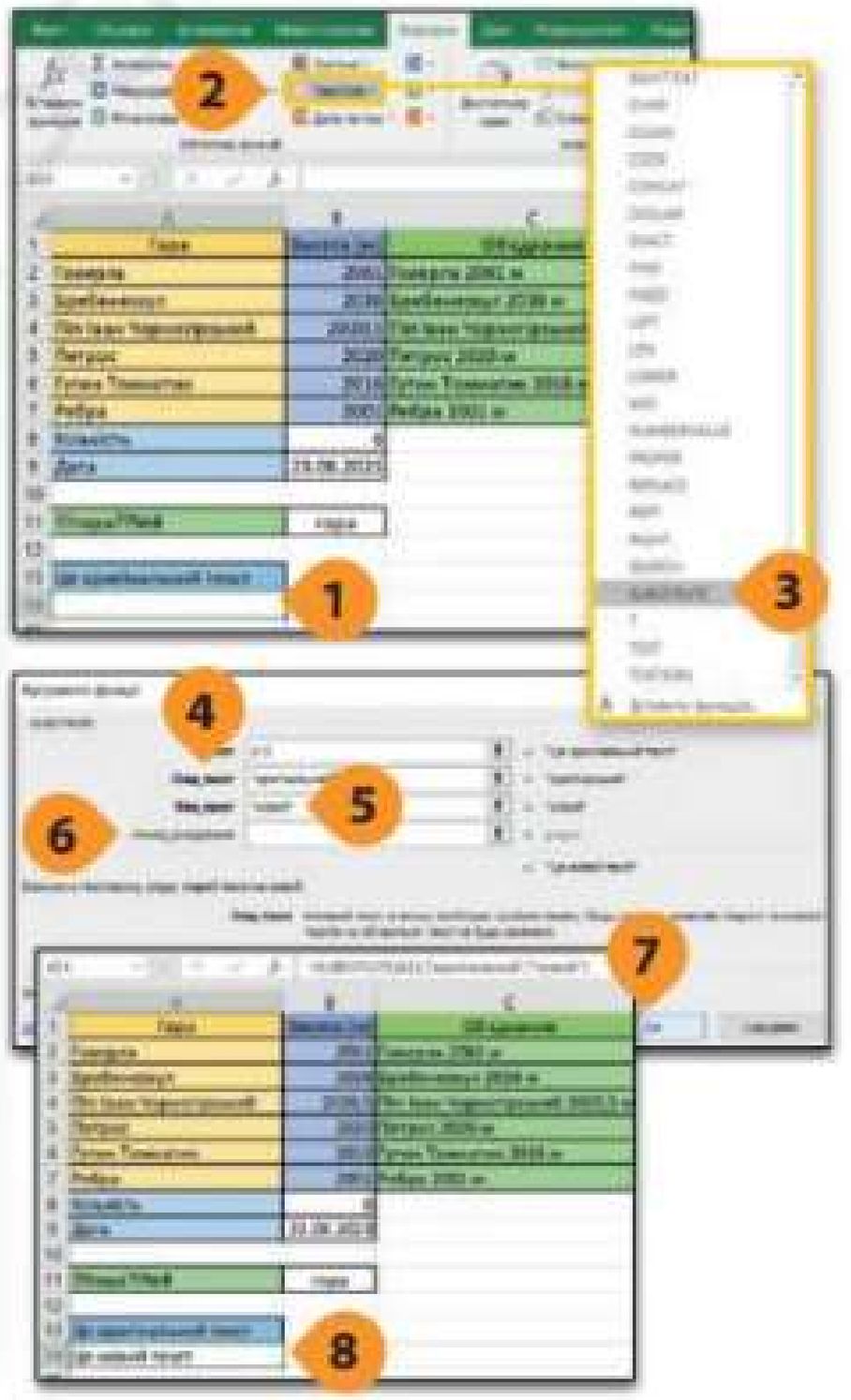
*Аналогічно можна скористатися функціями **LEFT** та **RIGHT** для вилучення крайніх лівих або крайніх правих символів вказаного текстового рядка.*

Функція SUBSTITUTE

Якщо ви хочете замінити частину тексту в комірці, скористайтеся функцією **SUBSTITUTE**.

Щоб замінити частину текстового рядка:

- натисніть на комірку **A14** (1);
- на вкладці **Формули** в розділі **Бібліотека функцій** натисніть **Текстові** (2);
- у спадному списку оберіть **SUBSTITUTE** (3);
- у вікні **Аргументи функцій** у полі **Текст** введіть **A13** (4). Це комірка, яка містить частину тексту, яку ви збираєтеся замінити.
- у полі **Стар_текст** введіть **оригінальний** (5). Це слово, яке буде замінене;
- у полі **Нов_текст** введіть **новий** (6). Це слово, яке буде вставлене;
- натисніть **ОК** (7);
- результат функції відобразиться в комірці **A14** (8).



© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Обчислюємо піднесення до степеня

Функція **Степінь** визначає результат піднесення числа до степеня.

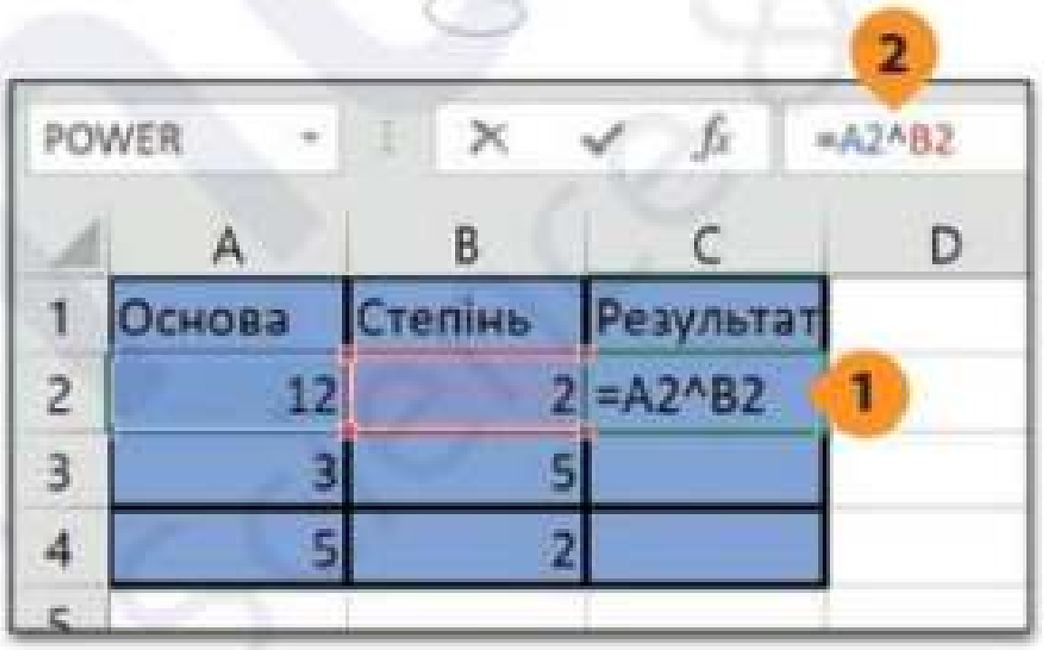
Створіть таку таблицю:

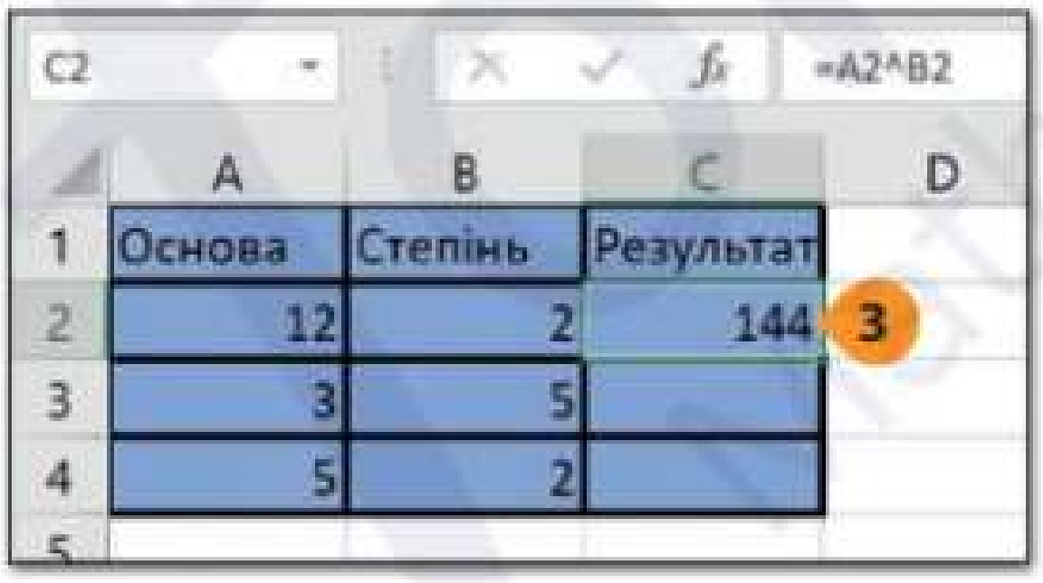
	A	B	C
1	Основа	Степінь	Результат
2	12	2	
3	3	5	
4	5	2	
5			

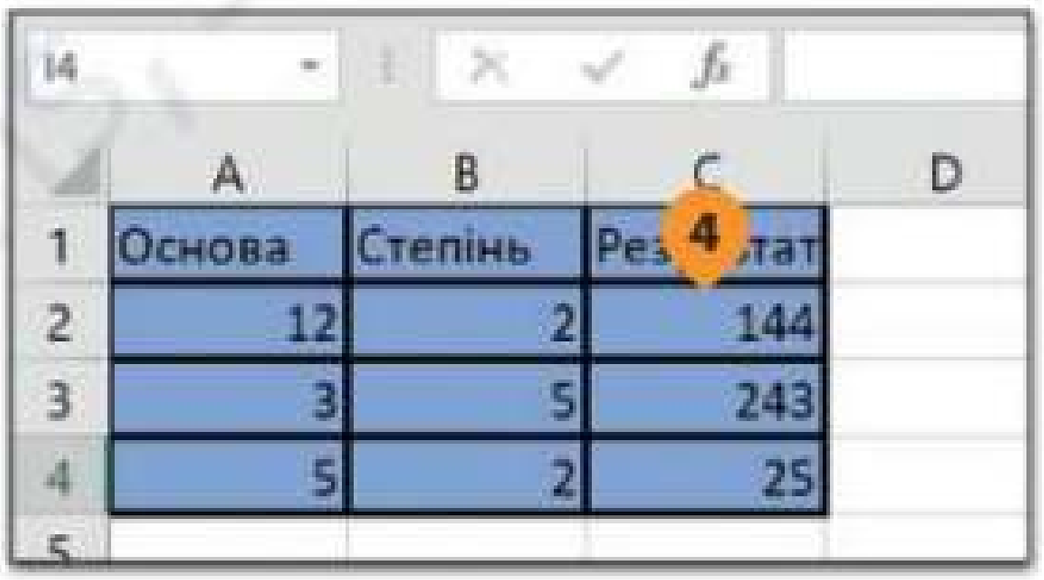
*Замість символу ^ ви можете використовувати функцію піднесення до степеня **POWER(x;y)**.*

Для обчислення степеня:

- натисніть комірку **C2** (1);
- у **Рядку формул** введіть **=A2^B2** (2);
- натисніть Ctrl + Enter ↵, щоб отримати результат та залишитися в тій самій комірці (3);
- повторіть попередні кроки для комірок **C3** та **C4** (4) або скористайтеся **Автозаповненням**.







Практичне завдання

Перейдіть за посиланням на сторінку інтернет-підтримки. Вам запропонували спланувати паркову площу. Розрахуйте, які з наведених елементів паркової зони ви розмістите, якщо вам відома площа парку та бюджет. Зробіть розрахунки.

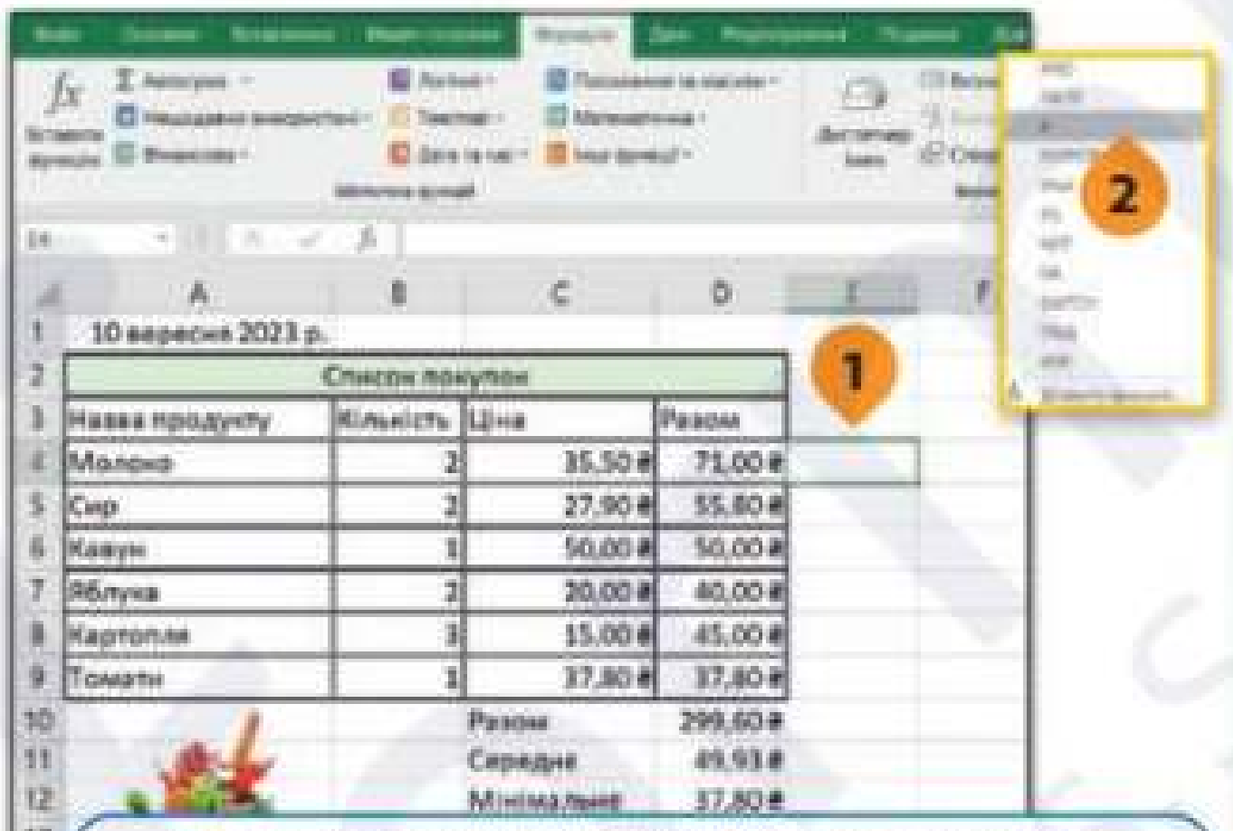
© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 5
Логічні функції

Microsoft Excel має різноманітні функції не лише для чисел, але й для логічних висновків. Однією з них є функція **IF**. Вона перевіряє, чи виконується умова. Якщо умова виконується, то подає одне значення, якщо ні – інше. Коли кажуть «значення», це може бути що завгодно: число, текст або навіть інша функція.

Повернемося до попереднього прикладу та подивимося, які товари вважають дорогими, а які ні. Припустимо, все, що коштує більше ніж 50,00 гривень, є дорогим.

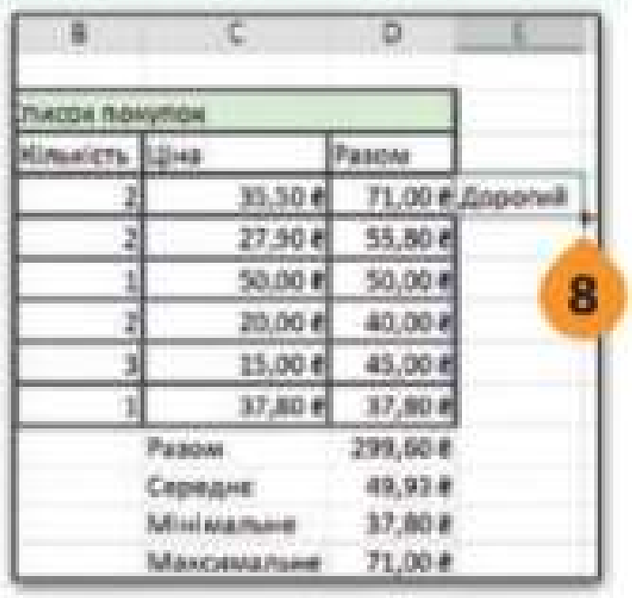
Обчислюємо за допомогою функції IF



Введіть результат, який з'являтиметься, якщо умова буде виконуватися. Стверджуємо, що в такому випадку продукт є дорогим, тому введіть **"Дорогий"**.

Введіть умову. Наприклад, «якщо ціна продукту вища ніж 50». Ціна вказана в комірці **D4**, тому умова буде така: **D4>50**.

Введіть результат, який з'являтиметься, якщо умова не буде виконуватися. Тому надрукуйте **"Дешевий"**.



Щоб вставити функцію IF:

- виберіть комірку, у якій буде відображатися результат, наприклад **E4** (1);
- на вкладці **Формули** в розділі **Бібліотека функцій** натисніть стрілку біля **Логічні** та оберіть **IF** (2);
- у вікні **Аргументи функції** (3) в текстовому полі **Лог_вираз** введіть **D4>50** (4);
- у текстовому полі **Значення_якщо_істина** введіть **"Дорогий"** (5);
- у текстовому полі **Значення_якщо_хибність** введіть **"Дешевий"** (6);
- клацніть **ОК** (7). Бачите результат?
- скористайтеся інструментом **Автозаповнення**, щоб скопіювати формулу (8).

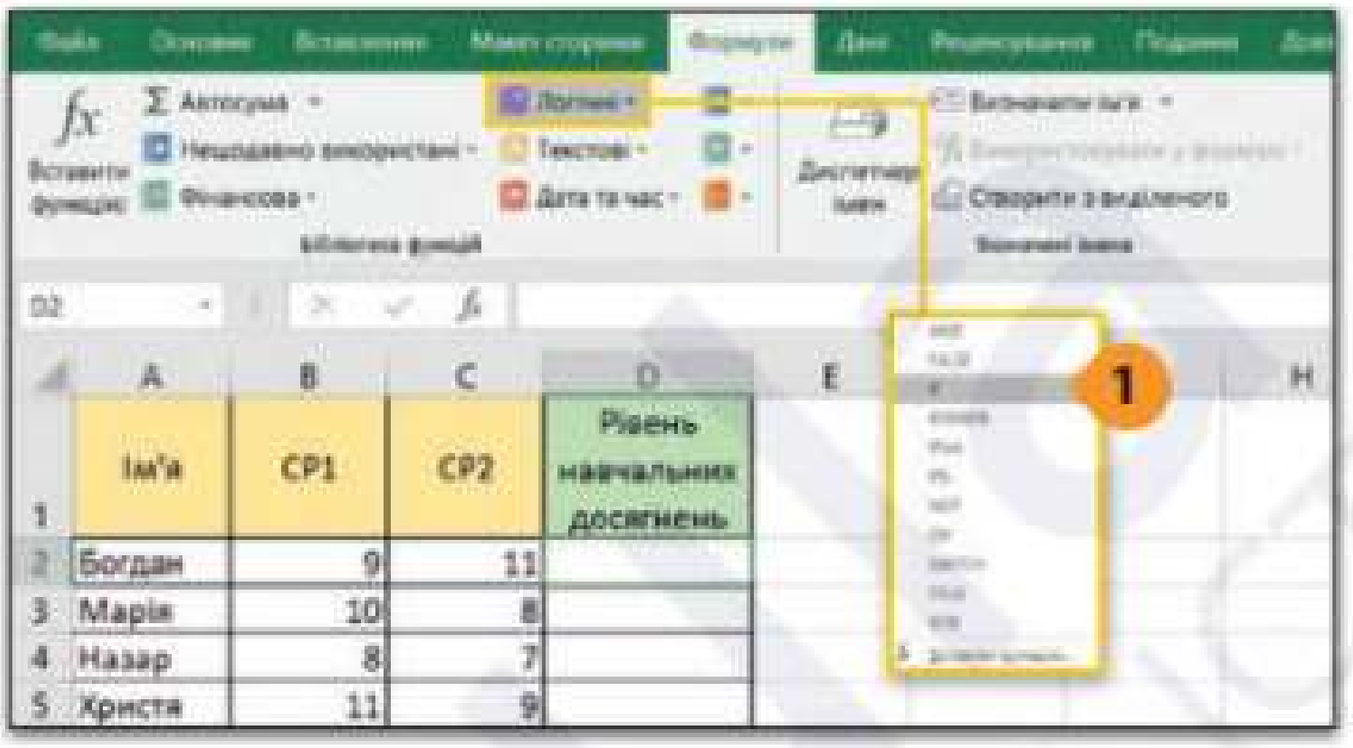
© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Функція **IF** може давати результати будь-якого типу. Окрім тексту, це можуть бути формули. Розглянемо приклад.

Є список учнів та учениць з оцінками за дві роботи. Необхідно оцінити їхній рівень навчальних досягнень. Якщо хтось набрав середній бал за 2 роботи понад 9, то він / вона отримує **"Високий"**, якщо нижче – **"Достатній"**.

Щоб уставити функцію IF із формулами:

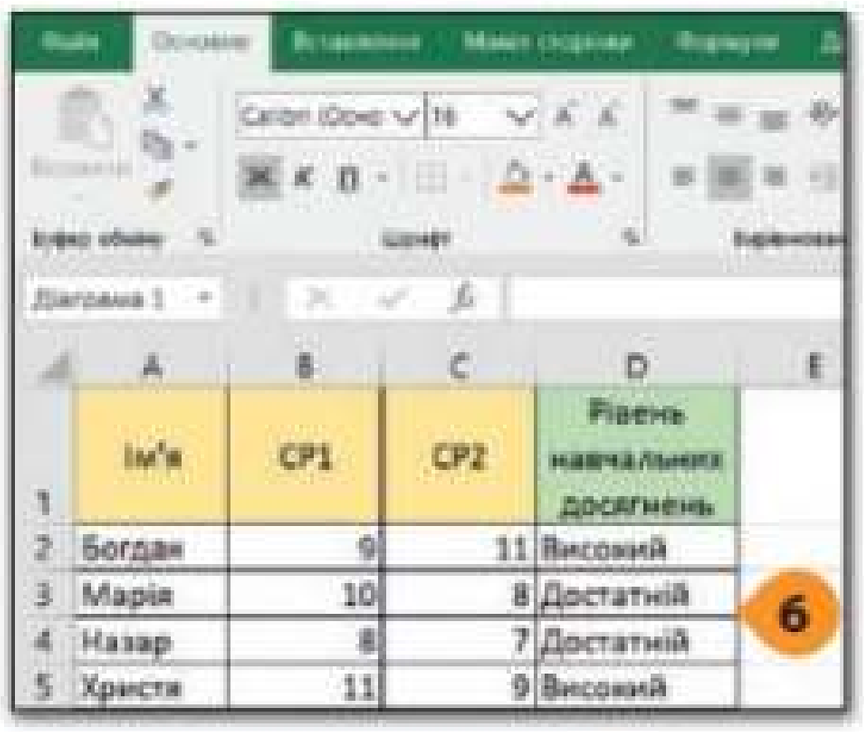
- вставте функцію **IF** (1);
- у текстовому полі **Лог_вираз** введіть **(B2+C2)/2>9** (2);
- у текстовому полі **Значення_якщо_істина** введіть **"Високий"** (3);
- у текстовому полі **Значення_якщо_хибність** введіть **"Достатній"** (4);
- натисніть **ОК** (5);
- скористайтесь інструментом **Автозаповнення** (6).



Введіть результат, який з'являтиметься, якщо умова буде виконана. У цьому випадку учень / учениця отримує **"Високий рівень"** навчальних досягнень.

Введіть умову. У цьому прикладі умова звучить так: **«Якщо середній бал за дві роботи понад 9»**. Отже, потрібно додати результати за **СР1** та **СР2** та поділити на **2**, щоб перевірити, середній бал. Для цього слід ввести **(B2+C2)/2>9**.

Навіть якщо ви не знаєте призначення функції, ви завжди можете прочитати її опис, де дуже просто описано, що обрана функція робить.



ІСТОРІЯ

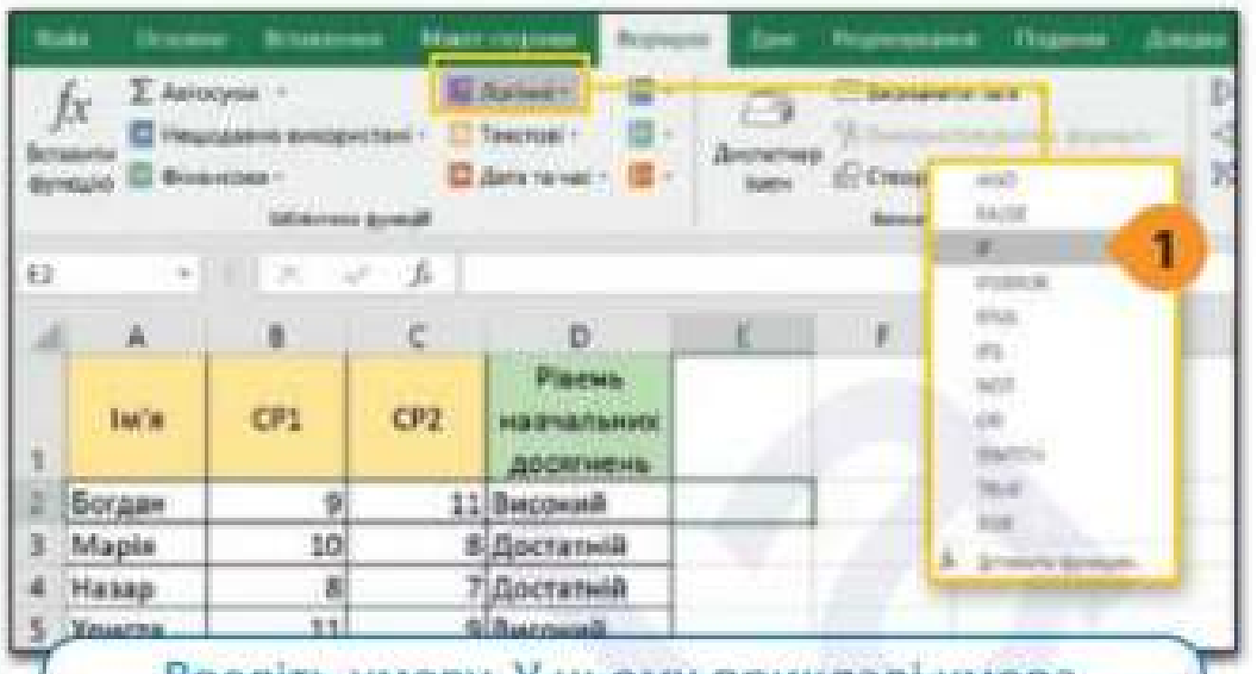
Чи знали ви, що в математиці функція – це така залежність змінної **y** від змінної **x**, при якій кожному значенню **x** відповідає єдине значення змінної **y**.

© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Робимо наступні кроки. Середній бал учня/учениці буде відображатися, якщо він/вона отримали рівень знань «Високий»; якщо ні – буде відображатися повідомлення «Достатній».

Щоб застосувати формулу з текстовим типом даних:

- виберіть комірку, у якій буде відображатися результат, наприклад **E2**, і вставте функцію **IF** (1);
- у текстовому полі **Лог_вираз** введіть умову. У цьому прикладі вона звучить: «Якщо рівень високий». Тому введіть **D2="Високий"** (2);
- у текстовому полі **Значення_якщо_істина** введіть **(B2+C2)/2** (3);
- у текстовому полі **Значення_якщо_хибність** введіть **"Достатній"** (4);
- натисніть **ОК** (5);
- скористайтесь інструментом **Автозаповнення** (6).



Введіть умову. У цьому прикладі умова звучить так: «Якщо рівень високий». Тому введіть **D2="Високий"**.

Введіть результат, який з'являтиметься, якщо умова не буде виконана. У такому випадку з'являється наведене повідомлення.

Введіть результат, який з'являтиметься, якщо умова буде виконана. Потрібно показати середній бал учня. Для цього слід обчислити середнє арифметичне за дві роботи.

ПІДКАЗКА

Залежно від регіональних налаштувань (наприклад, в україномовній програмі) функції Microsoft Excel можуть записуватися з крапкою з комою між аргументами функцій, наприклад:
=IF(лог_вираз; значення_якщо_істина;значення_якщо_хибність).

Практичне завдання

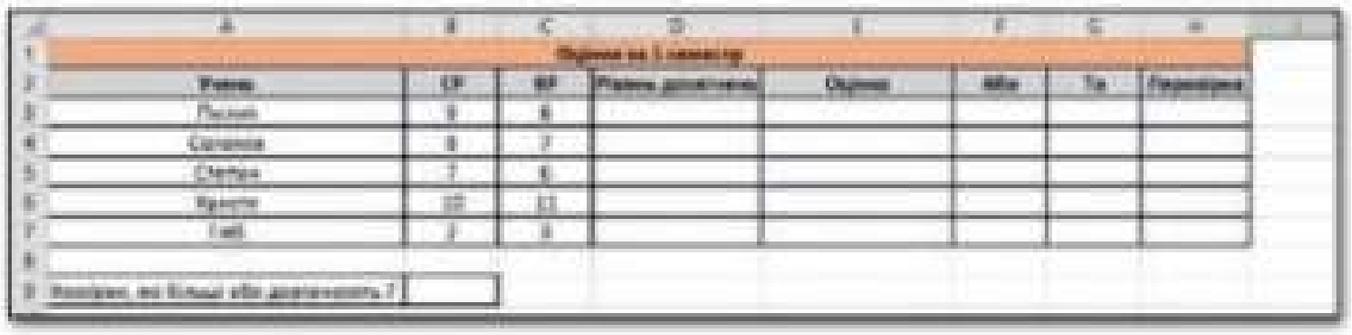
Перейдіть за посиланням на сторінку інтернет-підтримки та завантажте файл-заготовку. Визначіть тип клімату в наведених країнах.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Вкладені функції IF

Ви вже знаєте, як користуватися функціями. Зробімо щось складніше. Пам'ятаєте функцію **IF**? Подивимося, як використовувати її для отримання інших результатів.

Створіть наведену таблицю.



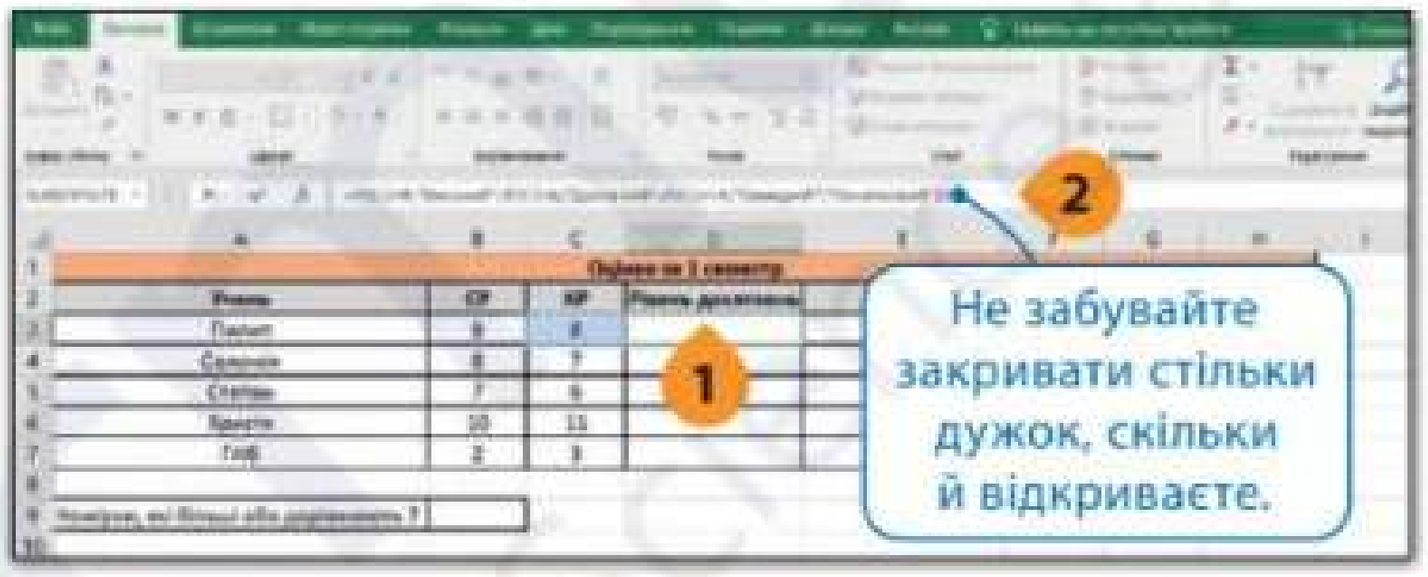
	A	B	C	D
1	Знання			
2		Раунд 1	Раунд 2	Медаль
3	Команда 1	5	6	Бронза
4	Команда 2	9	9	Золото
5	Команда 3	7	8	Срібло
6	Команда 4	4	5	Немає медалі

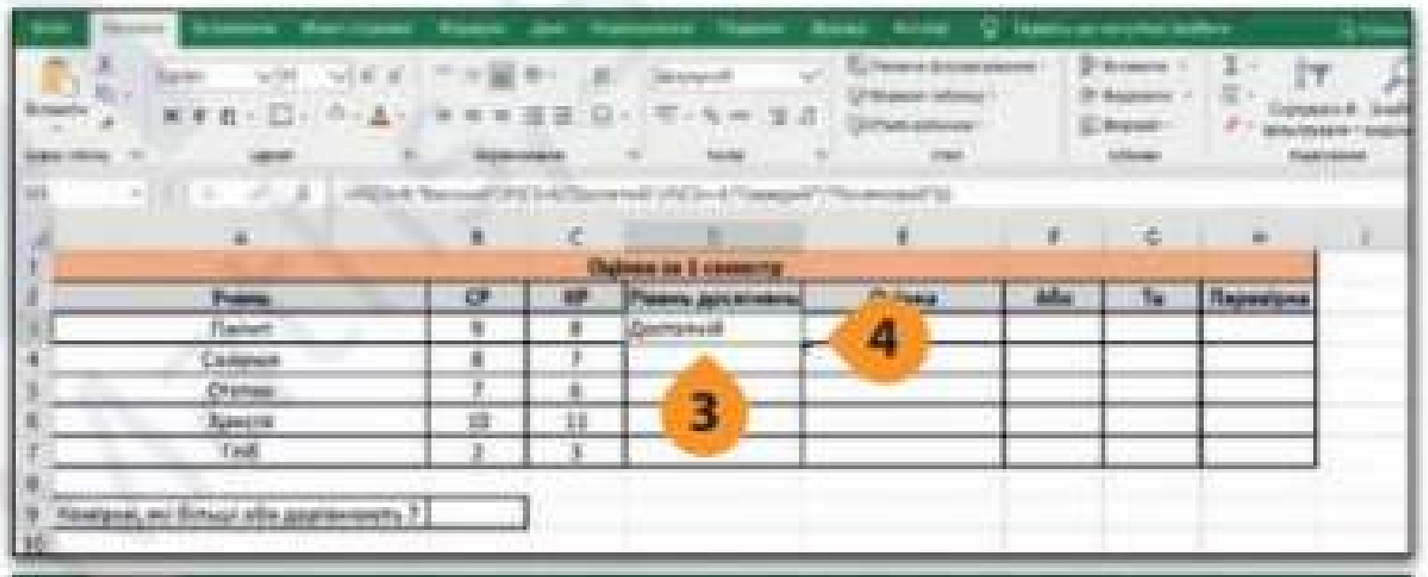
Уявіть себе вчителем/вчителькою. Вам треба оцінити досягнення учнів та учениць.

Якщо оцінка вища 9, тоді він/вона отримує рівень "Високий", інакше (відповідає **else if**), якщо оцінка більша ніж 6, то він або вона отримає "Достатній" рівень, інакше (відповідає **else if**), якщо оцінка понад 3, то він/вона отримає "Середній". Якщо дорівнює 3 або менша (відповідає **else**), то учень/учениця отримує "Початковий" рівень.

Щоб виокремити лише частину рядка:

- клацніть на комірку **D3** (1);
- у **Рядку формул** введіть **=IF(C3>9;"Високий";IF(C3>6;"Достатній";IF(C3>=4;"Середній";"Початковий")))** (2);
- натисніть Enter (3), щоб обчислити формулу та отримати результат;
- ще раз натисніть на комірку **D3** та, потягнувши **маркер заповнення** (4), заповніть стовпчик за допомогою **Автозаповнення** (5).





*Починаючи з версії **Microsoft Excel 2007**, усі наступні версії дозволяють підтримувати до 64 вкладень функцій **IF**.*

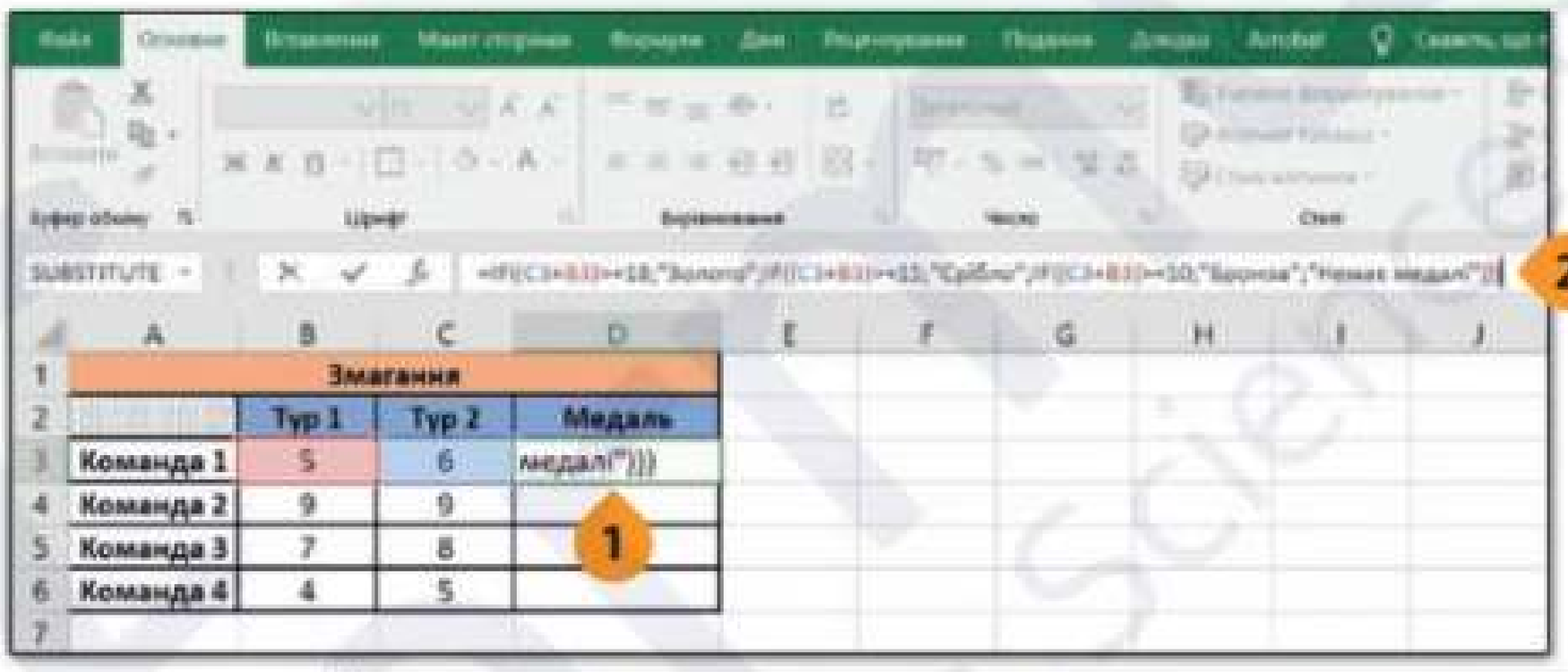
© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

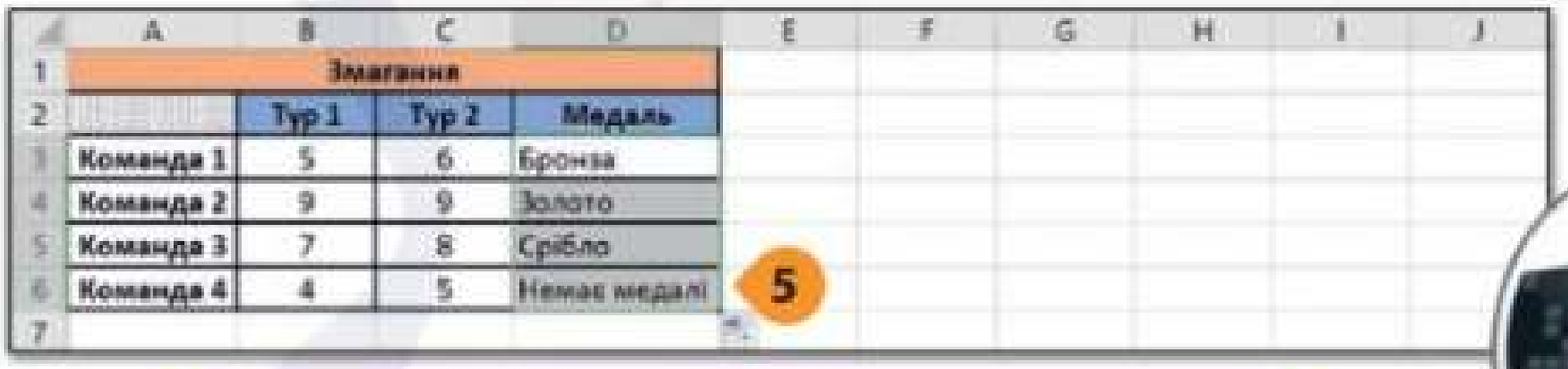
Трохи ускладнимо функцію **IF** із вкладеними умовами, додамо ще кілька **IF** у комірку **D3** другої таблиці.

Якщо загальна кількість перемог дорівнює 18 або більша ніж 18, тоді команда отримає золоту медаль, коли (відповідає **else if**) значення 15 та більше, тоді команда отримає срібну медаль, коли (відповідає **else if**) значення дорівнює 10 або більше ніж 10, тоді команда отримає бронзову медаль, а якщо (відповідає **else**) менше ніж 10, команда не отримає медаль.

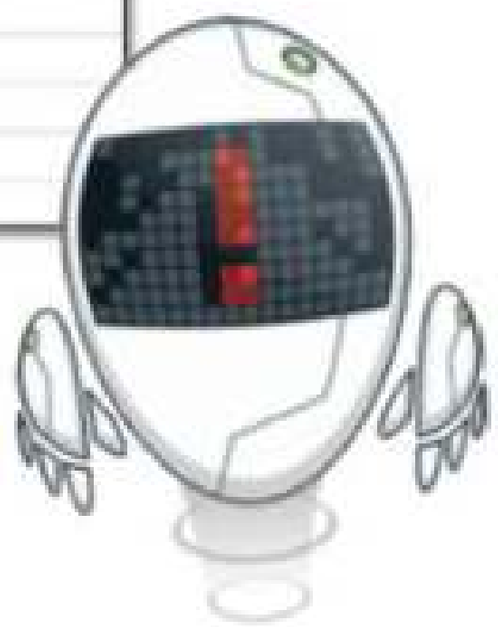
Щоб просту функцію IF перетворити на вкладену:

- клацніть на комірку **D3** (1);
- у **Рядку формул** введіть **=IF((C3+B3)>=18;"Золото";IF((C3+B3)>=15;"Срібло";IF((C3+B3)>=10;"Бронза";"Немає медалі")))** (2);
- натисніть **Ctrl** + **Enter** (3);
- використайте **маркер заповнення** (4), щоб заповнити стовпчик за допомогою **Автозаповнення** (5).





*Майте на увазі, що в полях **Значення_якщо_...** можна ввести одне слово чи невелику фразу або робити складні обчислення й навіть використовувати інші функції.*



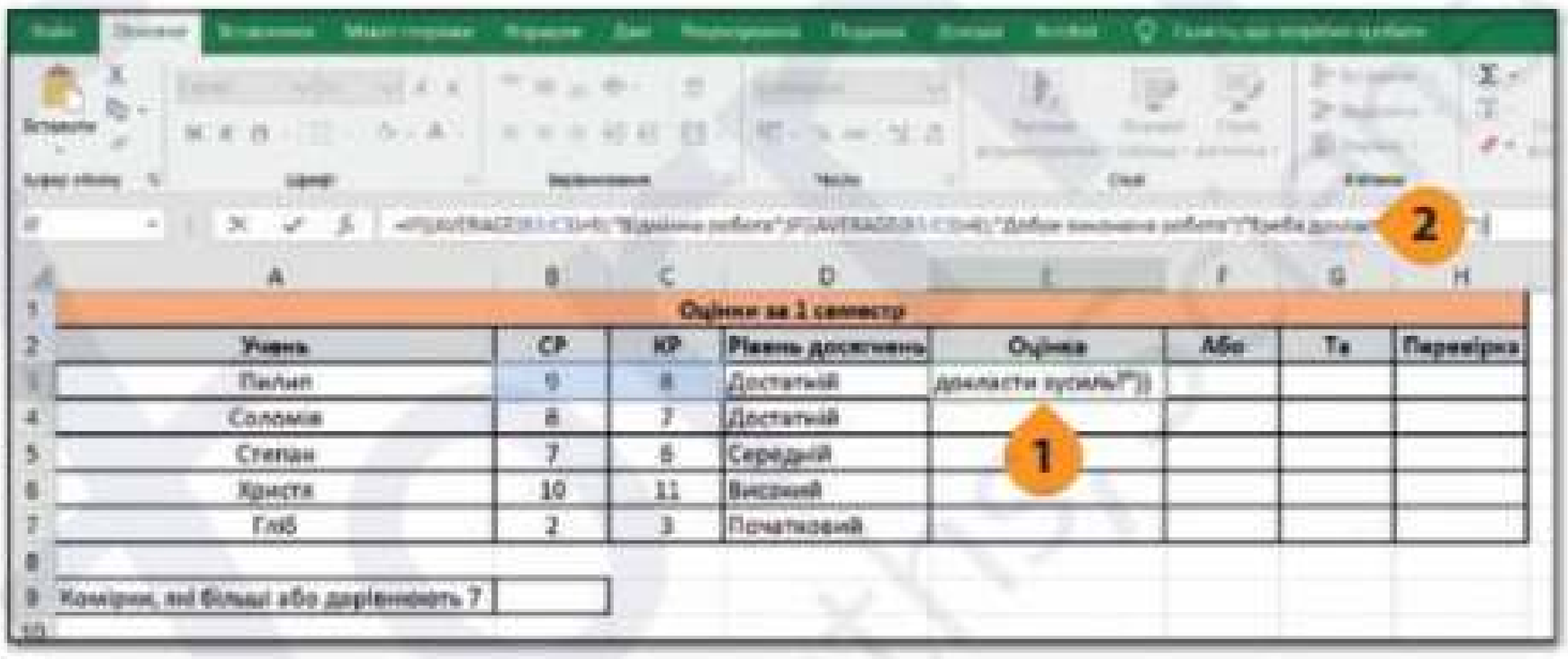
© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Повернемося до таблиці з оцінками. Нарешті додамо вкладену функцію **IF**, яка використовує формулу **AVERAGE**, до комірки **E3**, і буде відображатися наведене нижче.

Якщо середній бал вищий 9, тоді учень/учениця отримає опис "Відмінна робота", коли (відповідає **else if**) середній бал вищий 6, тоді він/вона отримає опис "Добре виконана робота", інакше (відповідає else) він/вона отримає опис "Треба докласти зусиль!".

Щоб поєднати декілька вкладених функцій IF із функцією AVERAGE:

- клацніть на комірку **E3** (1);
- у **Рядку формул** введіть **=IF((AVERAGE(B3:C3)>9);"Відмінна робота"; IF((AVERAGE(B3:C3)>6);"Добре виконана робота";"Треба докласти зусиль!"))** (2);
- натисніть Ctrl + Enter, щоб обчислити функцію і залишитися в тій самій комірці (3);
- використайте **маркер заповнення** (4), щоб заповнити стовпчик за допомогою **Автозаповнення** (5).



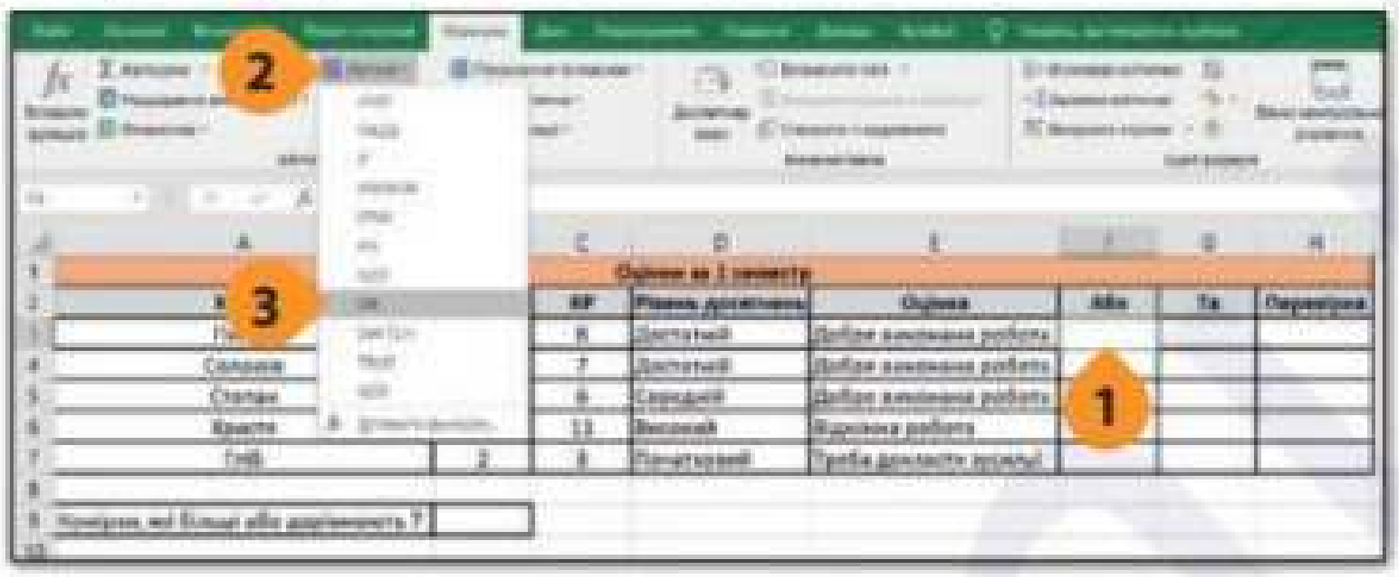
© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

У **Microsoft Excel** є й інші логічні функції, зокрема **OR (АБО)** та **AND (І)**. Результатом цих функцій можуть бути як **TRUE (істинно)**, так і **FALSE (хибно)**.

При вивченні Scratch ви вже мали можливість познайомитися і мали можливість використати такі оператори.

Функція OR

Ви можете використовувати функцію **OR**, щоб перевірити, чи виконується якась з умов. Якщо є хоч одна умова, яка виконується, **Microsoft Excel** відображає значення **TRUE**, інакше відображається **FALSE**. Наприклад, у таблиці **Оцінок за 1 семестр** потрібно перевірити, чи оцінки за самостійну **АБО** контрольну роботи перевищують 6.



Щоб застосувати функцію OR:

- клацніть на комірку **F3** (1);
- на вкладці **Формули** в розділі **Бібліотека функцій** клацніть на **Логічні** (2);
- у списку оберіть **OR** (3);
- у вікні **Аргументи функцій** у текстовому полі **Лог_значення1** введіть **B3>6**, а в текстовому полі **Лог_значення2** введіть **C3>6** (4);
- натисніть **ОК** (5);
- використайте **маркер заповнення** (6), щоб заповнити стовпчик за допомогою **Автозаповнення** (7).

*Як і очікували, комірка **F7** має значення **Хибно**, оскільки значення в комірках **B7** і **C7** не перевищують 6.*

Зверніть увагу, що одна з умов має значення **Хибно (6>6)**, але функція **OR** повертає **Істинно**, бо інша умова має значення **Істинно (7>6)**.

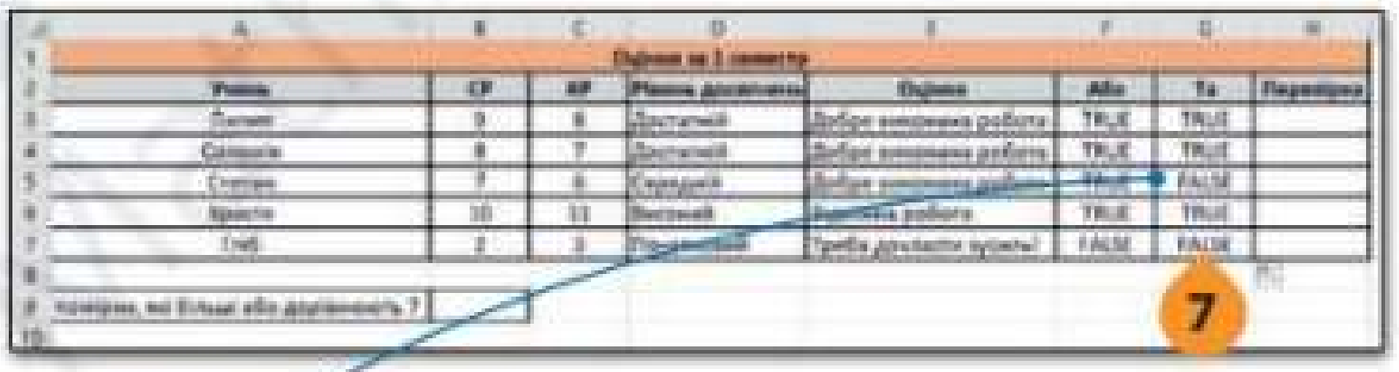
© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Функція AND

За допомогою функції **AND** можна перевірити, чи виконуються всі умови певної групи. В цьому випадку **Microsoft Excel** відображає значення **TRUE**, інакше відображається **FALSE**. Наприклад, у таблиці **Оцінок за 1 семестр** потрібно перевірити, чи оцінки за самостійну **І** контрольну роботи перевищують 6.

Щоб використати функцію AND:

- клацніть на комірку **G3** (1);
- на вкладці **Формули** в розділі **Бібліотека функцій** клацніть на **Логічні** (2);
- у списку оберіть **AND** (3);
- у вікні **Аргументи функцій** у текстовому полі **Лог_значення1** введіть **G3>6**, а в текстовому полі **Лог_значення2 – G3>6** (4);
- натисніть **ОК** (5);
- використайте **маркер заповнення** (6), щоб заповнити стовпчик за допомогою **Автозаповнення** (7).



Зверніть увагу!
На відміну від попередньої функції **OR**, одна з умов має значення **Істинно (7>6)**, але функція **AND** повертає **Хибно**, бо інша умова має значення **Хибно (6>6)**.

ПІДКАЗКА

Завжди перевіряйте формули в електронній таблиці. Крихітна помилка може спричинити величезну проблему!

© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Поєднаємо функції IF та AND

Якщо (**IF**) оцінка за самостійну роботу перевищує 4 або дорівнює 4 і (**AND**) оцінка за контрольну роботу перевищує або дорівнює 4, тоді учень пройшов атестацію, інакше учень не атестований.

Щоб поєднати вкладену функцію IF з функцією AND:

- натисніть на комірку **H3** (1);
- у **Рядку формул** введіть **=IF(AND(B3>=4;C3>=4);"Атестований"; "Не атестований")** (2);
- натисніть **Ctrl** + **Enter**, щоб обчислити формулу та залишитися в тій самій комірці (3);
- використайте **маркер заповнення** (4), щоб заповнити стовпчик за допомогою **Автозаповнення** (5).

Оцінки за 1 семестр							
Учень	СР	КР	Рівень досягнень	Оцінка	Або	Та	Перевірка
Пилип	9	8	Достатній	Добре виконана робота	TRUE	TRUE	Атестований
Соломія	8	7	Достатній	Добре виконана робота	TRUE	TRUE	Атестований
Степан	7	6	Середній	Добре виконана робота	TRUE	FALSE	Атестований
Христя	10	11	Високий	Відмінна робота	TRUE	TRUE	Атестований
Гліб	2	3	Початковий	Треба докласти зусиль!	FALSE	FALSE	Не атестований
Комірки, які більші або дорівнюють ?							

ПІДКАЗКА

У багатьох країнах кома використовується як десятковий розділювач, тоді як в інших використовується крапка. Знайдіть за посиланням, у яких країнах використовують розділювачі у вигляді крапки:
uk.wikipedia.org/wiki/Десятковий_розділювач

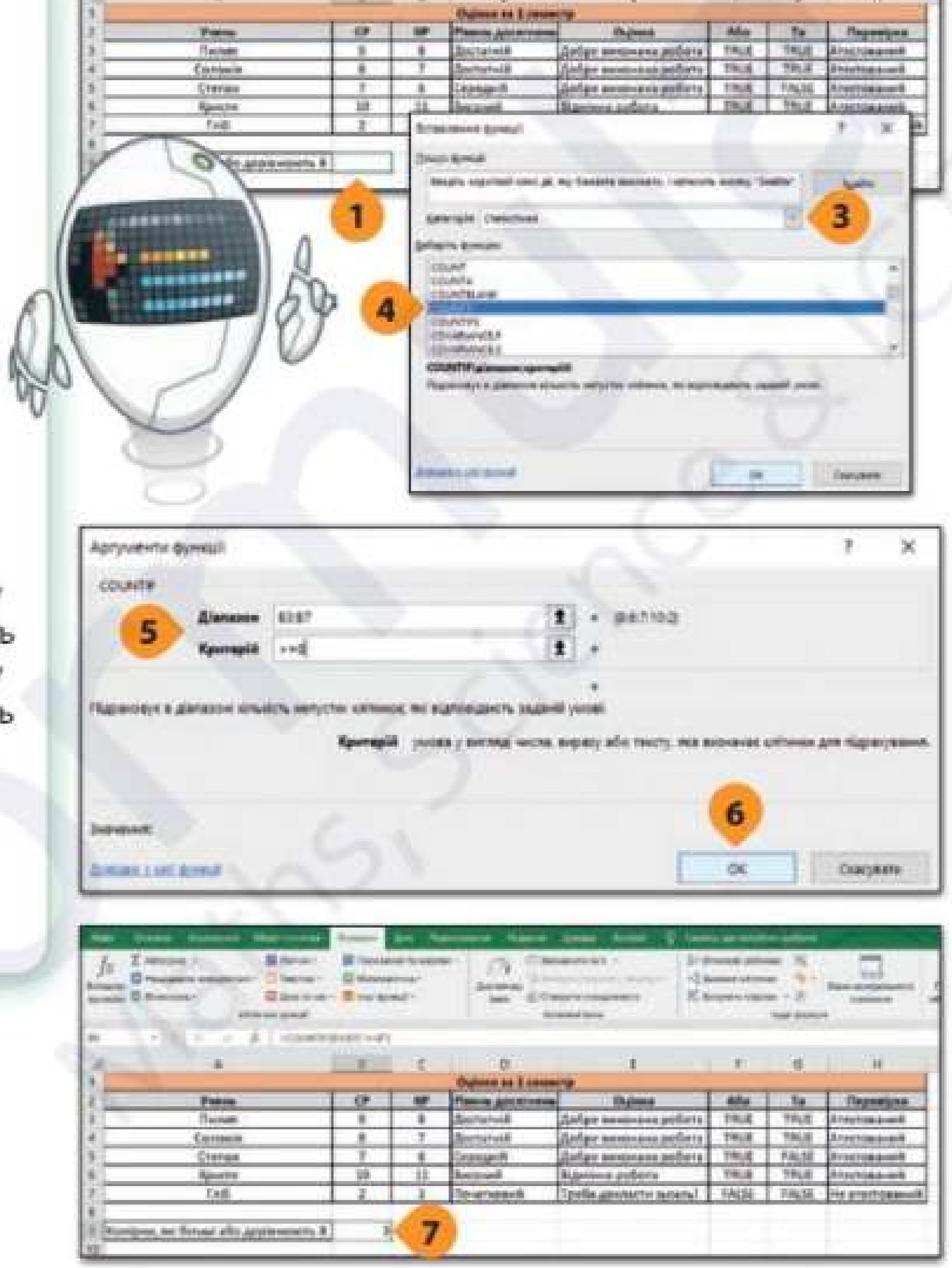
© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Функція COUNTIF

За допомогою функції **COUNTIF** можна порахувати кількість комірок, які відповідають певній умові. Наприклад, скільки комірок має значення більше або дорівнює 8.

Щоб застосувати функцію COUNTIF:

- натисніть комірку **B9** (1);
- на вкладці **Формули** в розділі **Бібліотека функцій** натисніть **Вставити функцію** (2);
- у вікні **Вставлення функції** в списку **Категорія** оберіть **Статистичні** (3);
- натисніть **COUNTIF** (4);
- у вікні **Аргументи функції** у текстовому полі **Діапазон** введіть **B3:B7**, а в текстовому полі **Критерій** введіть **>=8** (5);
- натисніть **ОК** (6);
- результат функції відобразиться в комірці **B9** (7).



Практичне завдання

Вам відома кількість туристів із різних країн світу, які відвідують Австралію. Проаналізуйте наведену інформацію за допомогою файлів-заготовок, розташованих за посиланням на сторінці інтернет-підтримки.

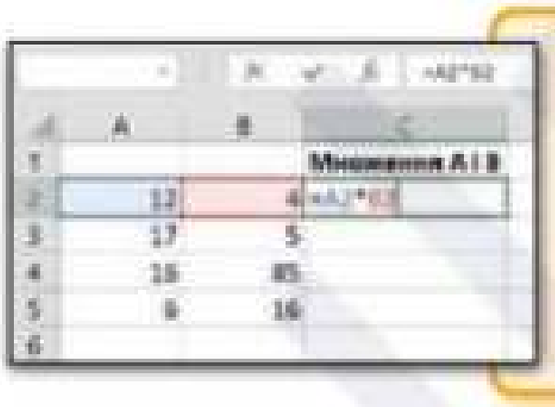
© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 6
Посилання

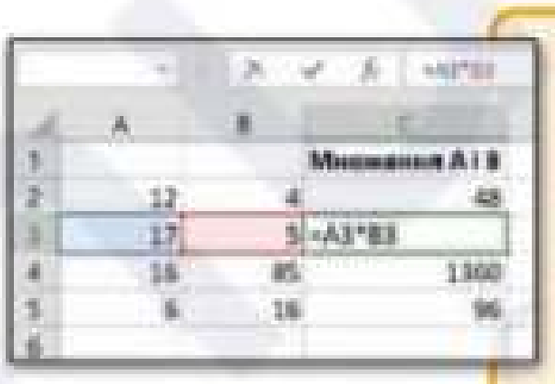
Як відомо, ім'я комірки складається з літери стовпця та номера рядка, до яких вона належить. Посилання на комірку і є адресою комірки та визначає її місцезнаходження. Коли ви копіюєте формулу в нові комірки, то можете використовувати відносні, абсолютні та мішані посилання.

Відносні посилання

Відносне посилання – це посилання на комірку. Під час копіювання формул посилання змінюються за правилом: номери стовпців/рядків в адресах клітинок змінюються на різницю номерів кінцевого та початкового стовпця / рядка. Якщо, наприклад, комірка **C2** посилається на комірку **A2**, фактично ви посилаєтеся на комірку, розташовану на два **(С мінус А)** стовпці ліворуч у тому самому **рядку 2**.



Наприклад, введіть числа в стовпці **A** *і* **B**, *а в комірку* **C2** *введіть* **=A2*B2**.



Якщо скопіювати з комірки **C2** *формулу* **=A2*B2** *до комірки* **C3**, *то отримаємо в ній* **=A3*B3**. *Зі зміною номера рядка змінилися й номери рядків у формулі: з* **2** *на* **3**.

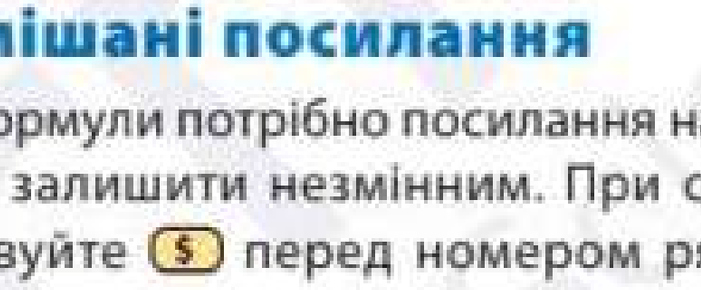
Абсолютні та мішані посилання

Іноді під час копіювання формули потрібно посилання на комірку, рядок або стовпець залишити незмінним. При створенні формули використовуйте $ перед номером рядка та/або літерою стовпця. Так створюється абсолютне посилання на комірку, яке не змінюється при його копіюванні.

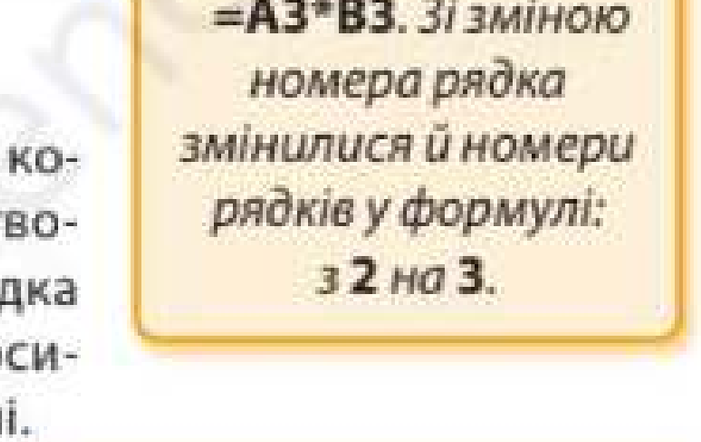
$E$1	Адреса комірки не змінюється під час копіювання. І стовпець, і рядок залишаються постійними.
$E1	Номер рядка змінюється під час копіювання, але стовпець залишається незмінним.
E$1	Стовпець змінюється під час копіювання, але рядок залишається незмінним.

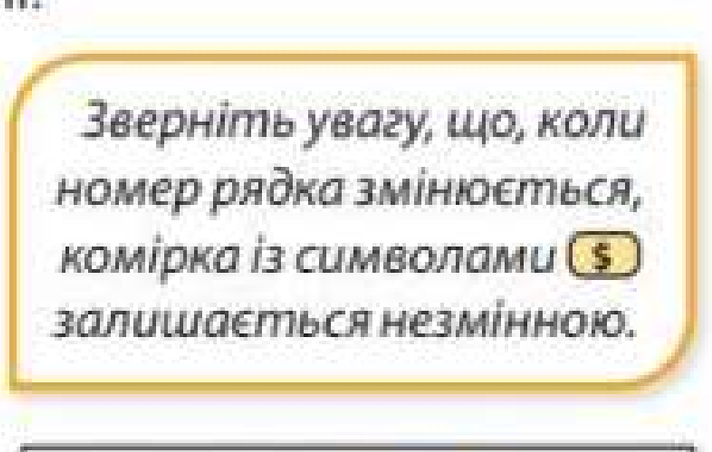
Зверніть увагу, що, коли номер рядка змінюється, комірка із символами $ залишається незмінною.

Наприклад, введіть дані для стовпців **A** і **B**, як показано нижче, а в комірку **C2** введіть **=$D$1*B2**.



Якщо ввести формулу **=$D$1*B2** в стовпець **C**, формула зміниться на **=$D$1*B3**, **=$D$1*B4** тощо.





ПІДКАЗКА

Замість функції **Автозаповнення** можна використовувати команди **Копіювати** Ctrl + C та **Вставити** Ctrl + V.

© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Розглянемо ще один приклад. Створіть наведену таблицю.

Щоб створити й скопіювати формулу з абсолютним посиланням:

- клацніть на комірку **H4** (1);
- у **Рядку формул** введіть **=G4*$G$1** (2);
- натисніть **Ctrl** + **Enter ↵**, щоб обчислити формулу та залишитися в тій самій комірці (3);
- скопіюйте формулу на весь стовпець за допомогою **Автозаповнення** (4).

Найпростіший спосіб запам'ятати, як використовувати $, – це подумати про те, як ви плануєте використати функцію ***Автозаповнення****. Якщо ви хочете використовувати її горизонтально, то $ поставте перед літерою (стовпцем). Якщо ж хочете використовувати її вертикально, введіть $ перед числом (рядком). Таке посилання називають* ***мішаним****.*

Щоб комірка мала абсолютне посилання, під час введення формули, коли ввели адресу комірки, або клацнули на неї, натисніть F4.

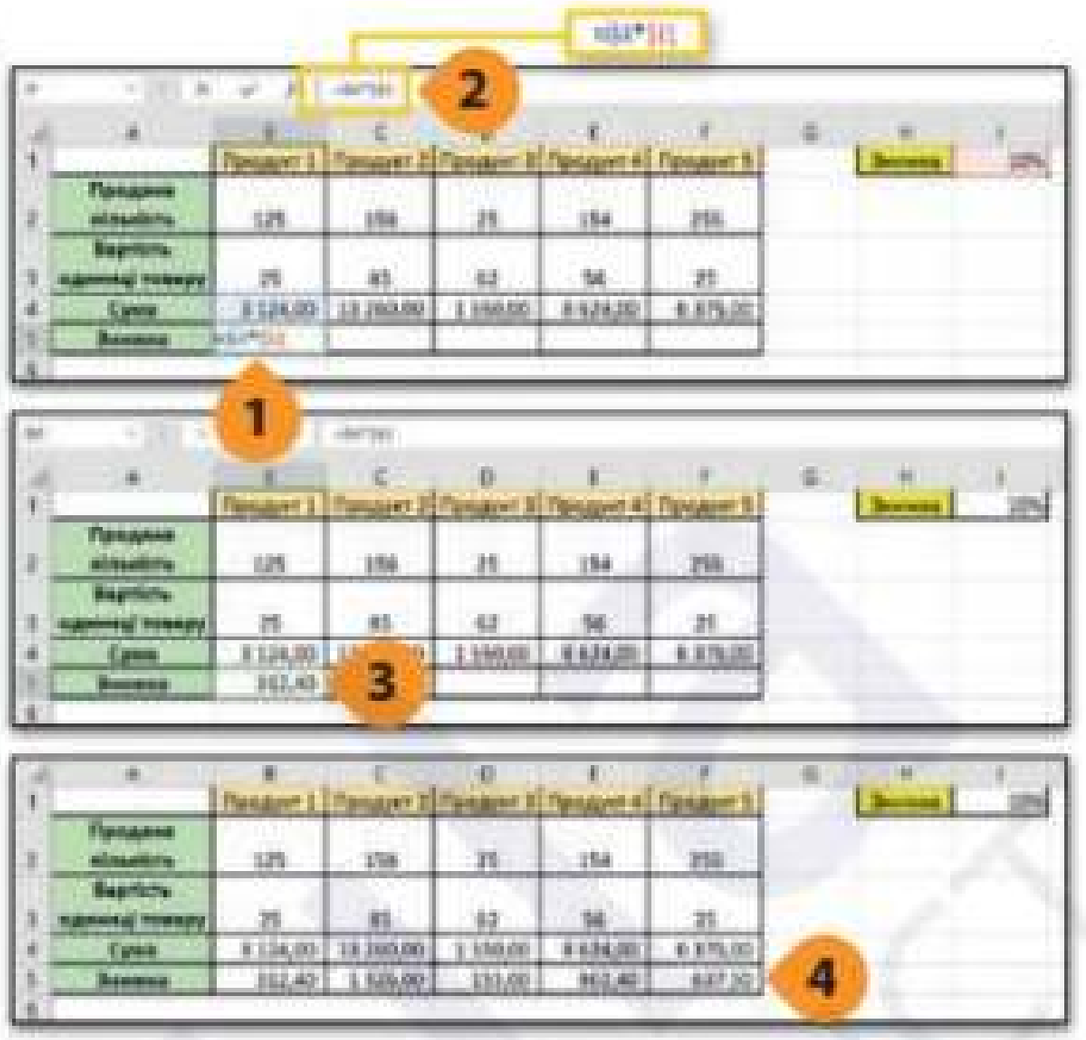
Щоб створити й скопіювати формулу з мішаним посиланням:

- створіть таблицю, що наведено праворуч, та клацніть на комірку **E2** (1);
- у **Рядку формул** введіть **=D2*B$8** (2);
- натисніть **Ctrl** + **Enter ↵**, щоб обчислити формулу та залишитися в тій самій комірці (3);
- скопіюйте формулу на весь стовпець за допомогою **Автозаповнення** (4).

© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Щоб створити й скопіювати формулу з мішаним посиланням:

- створіть таблицю, що наведено праворуч, та клацніть на комірку **B5** (1);
- у **Рядку формул** введіть **=B4*$I1** (2);
- натисніть **Ctrl** + **Enter**, щоб обчислити формулу та залишитися в тій самій комірці (3);
- скопіюйте формулу на весь рядок за допомогою **Автозаповнення** (4).



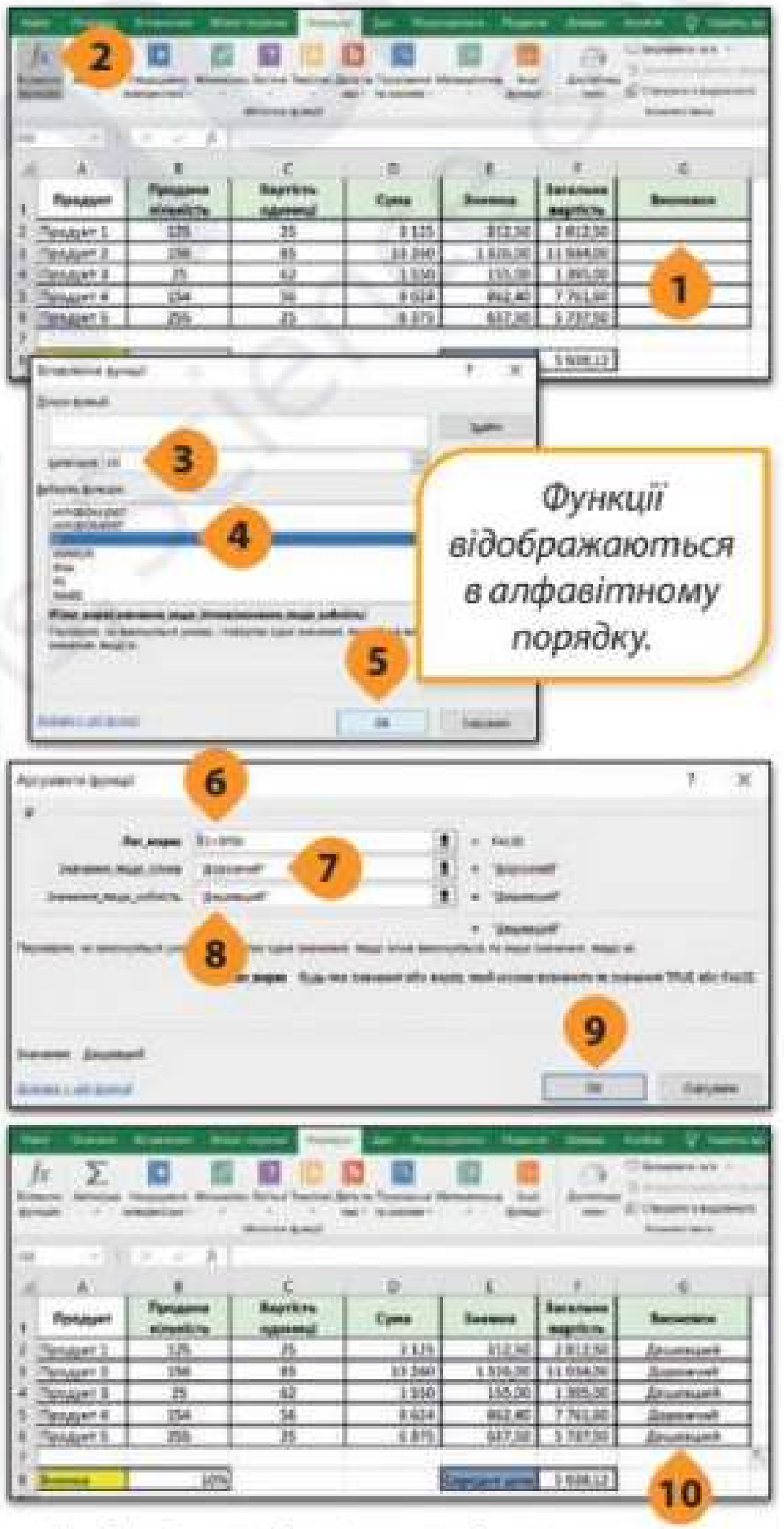
Фукція IF та посилання

Зробимо щось трохи складніше. Поєднаймо функцію **IF** з абсолютним посиланням. Знайдемо **Загальну вартість**, віднімемо **Знижка** від **Суми** і знайдемо **Середню ціну** в комірці **F8**.

Якщо (**IF**) **Загальна вартість** більша за **Середню ціну**, тоді продукт **дорожчий за середню**, якщо менше (**ELSE**), то **дешевше за середню**.

Щоб поєднати функції IF із посиланнями:

- клацніть клітинку **G2** (1);
- на вкладці **Формули** в розділі **Бібліотека функцій** натисніть **Вставити функцію** (2);
- у вікні **Вставлення функції** в спадному списку **Категорія** оберіть **Усі** (3);
- натисніть **IF** (4);
- натисніть **ОК** (5);
- у вікні **Аргументи функцій** введіть **F2>$F$8** у текстовому полі **Лог_вираз** (6);
- у текстовому полі **Значення_якщо_істина** введіть **Дорожчий** (7);
- у текстовому полі **Значення_якщо_хибність** введіть **Дешевший** (8);
- натисніть **ОК** (9);
- скопіюйте функцію за допомогою **Автозаповнення** (10).



© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Повідомлення про помилки

Іноді під час введення формули можна допустити помилки. Коли таке стається, з'являються повідомлення.

Виправити помилку можна, натиснувши на комірку з помилкою. Поряд із нею з'являється піктограма (1). Наведіть вказівник на неї миші на неї та клацніть. У спадному меню програма пропонує варіанти розв'язання проблеми та опис помилки. Натисніть на ***Довідка із цієї помилки*** *(2) та* ***Показати кроки обчислення*** *(3), щоб виправити помилку.*

Повідомлення про найпоширеніші помилки

#####	Таке сповіщення з'являється, коли стовпець не досить широкий, щоб відобразити вміст комірки. Перетягніть межу стовпця, щоб розширити його, або скористайтеся **Автодобір ширини стовпця**.
#ДІЛЕННЯ/0!	Повідомлення з'являється, коли число ділите на 0. Це можна виправити, змінивши дільник у функції або формулі, щоб він не був 0 або порожнім.
#ІМ'Я?	Повідомлення про помилку з'являється, коли ви ввели неправильну назву формули, і Microsoft Excel не може її розпізнати. Виправте це, ввівши правильну назву формули, або скористатися інструментом **Вставити функцію** на вкладці **Формули**.
#ЗНАЧЕННЯ!	Це повідомлення свідчить про те, що введена формула містить помилку або щось не так із комірками, на які вона посилається. Виправити це можна, перевіривши посилання на комірки та їхній формат, а також безпосередньо саму формулу, чи правильно вона складена.

Практичне завдання

Вам відома інформація про дані продажів електроніки в інтернет-магазині. Проаналізуйте ці дані та спрогнозуйте продажі наступного періоду. Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте завдання.

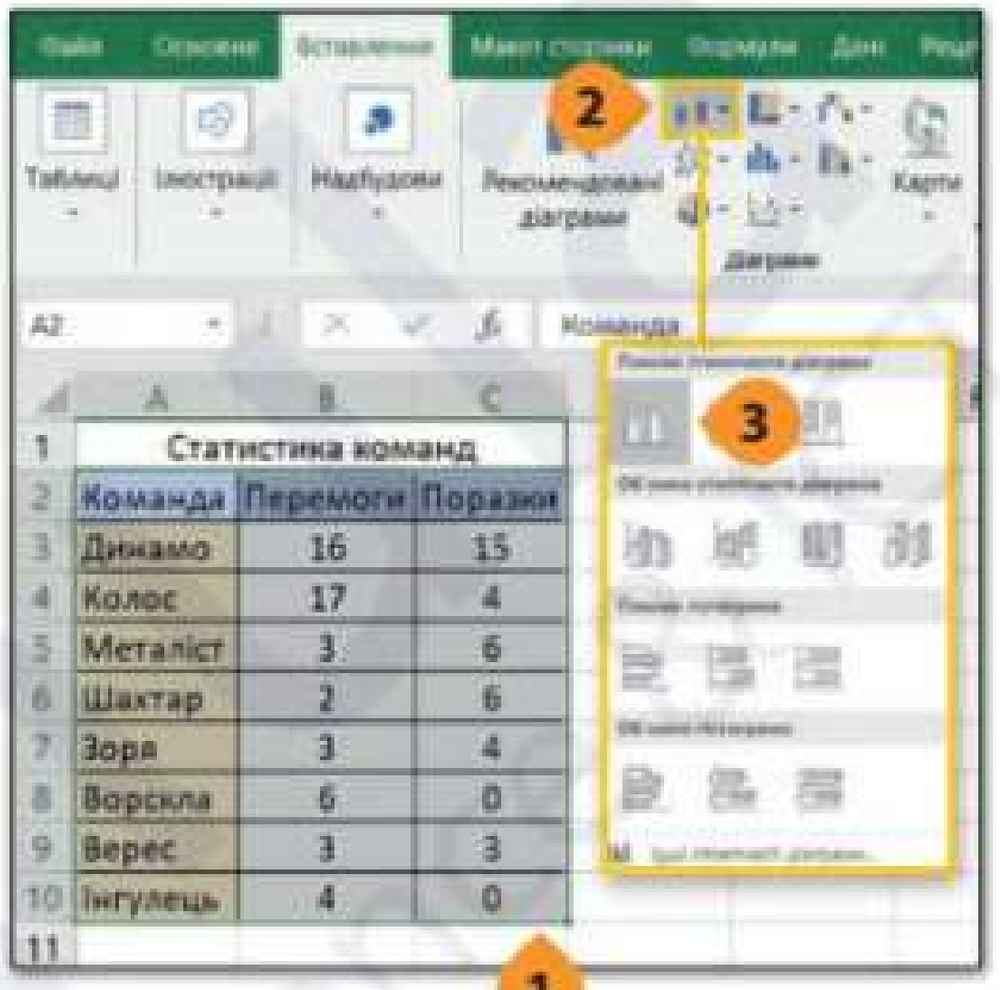
© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

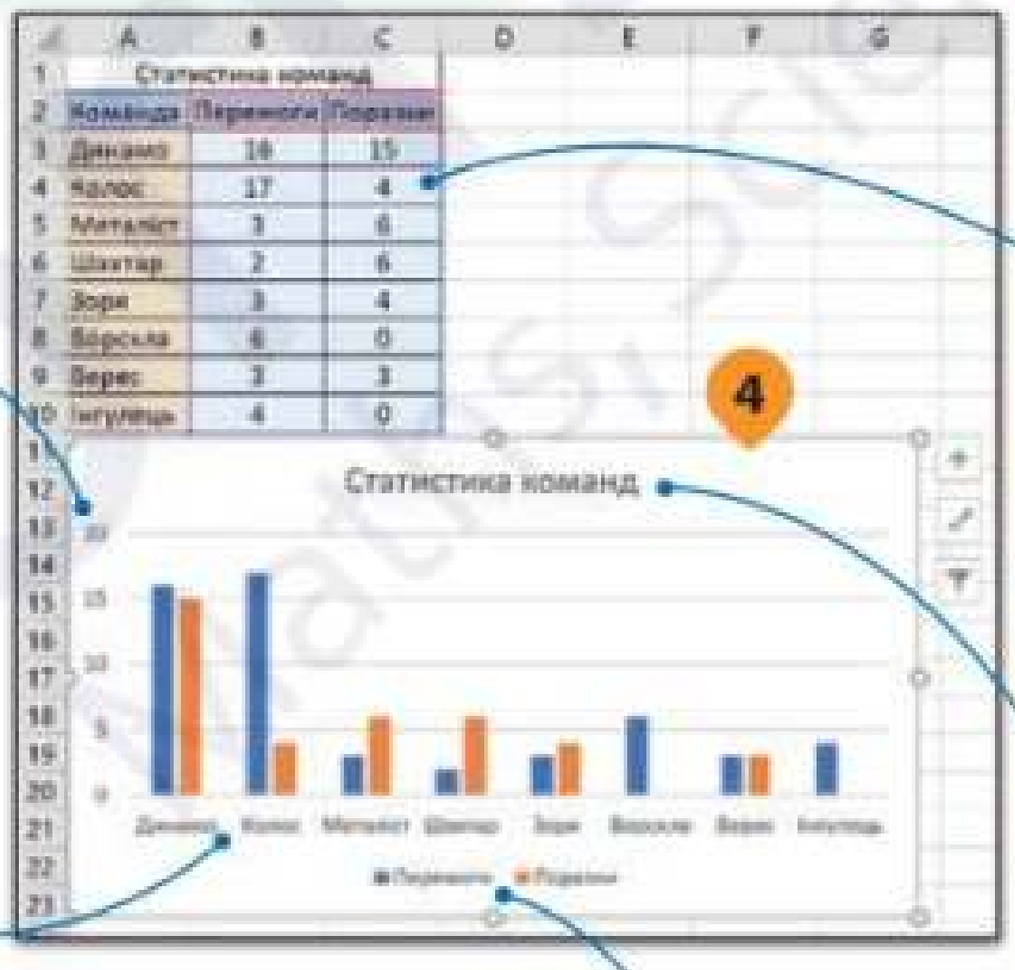
Тема 7
Діаграми та графіки

Діаграма або графік – спосіб візуального подання даних. Вони допомагають краще розуміти дані та аналізувати їх. Зазвичай набагато швидше порівнювати фігури, ніж числа.

Щоб створити діаграму:

- оберіть та виділіть дані, які бажаєте представити у вигляді діаграми, наприклад дані з комірок від **A2** до **C10** (1);
- у вкладці **Вставлення** в розділі **Діаграми** натисніть **Вставити стовпчасту діаграму або гістограму** (2);
- виберіть тип діаграми, який до вподоби, наприклад **Плоска стовпчаста діаграма** (3), та клацніть по ній;
- з'явиться діаграма, яка відображає обрані дані (4).





Вертикальна вісь (Значення) показує діапазон ваших даних від найменшого до найбільшого значення.

Горизонтальна вісь (Категорія) показує категорії, які є у вашій таблиці. У цьому прикладі ви бачите назви команд.

Якщо ви змінити щось у таблиці і натиснете Enter, діаграма автоматично зміниться, щоб показати нову інформацію.

Щоб змінити заголовок діаграми, просто клацніть на нього.

Легенда, використовуючи кольори та назви з ваших даних, пояснює, що являє собою кожний елемент значень на діаграмі.

ПІДКАЗКА

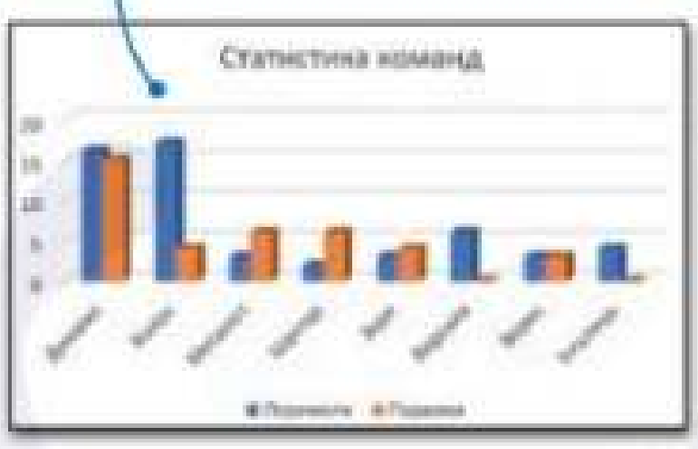
Діаграми бувають різного типу. Як знати, яку краще вибрати для конкретного випадку? Все залежить від того, що ви хочете показати. Стовпці та гістограми підходять для порівняння значень. Графіки зазвичай використовують, щоб відобразити зміну значення протягом певного часу. Секторна діаграма більше підходить для відсотків.

© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

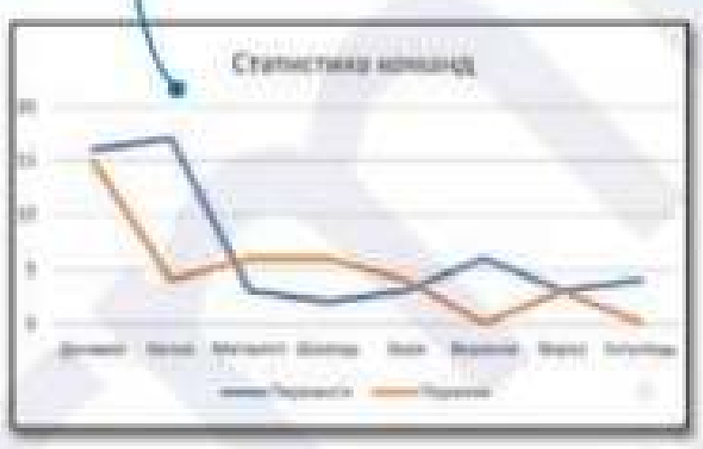
Типи діаграм

Щоб зробити ваші презентації з даними більш живими та цікавішими, слід вибирати «правильну» діаграму. Вона повинна бути простою та доступною.

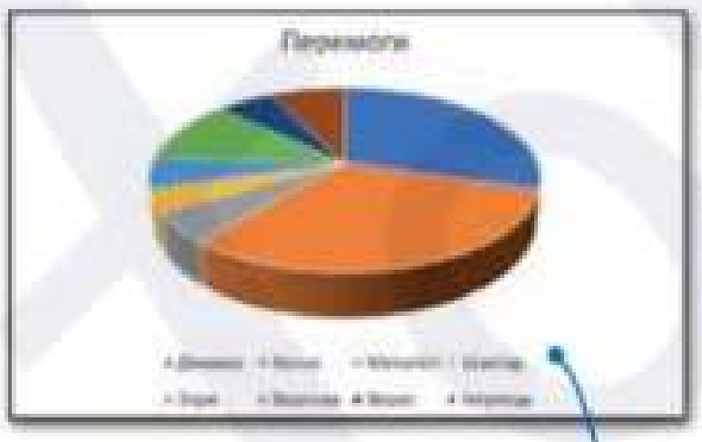
Стовпчаста діаграма або **гістограма** використовується для ілюстрації порівнянь між низкою даних. На стовпчастій діаграмі категорії відображаються горизонтально (вісь **X**), а числові значення – вертикально (вісь **Y**). На гістограмі, одному з найпоширеніших типів діаграм, все відбувається навпаки.



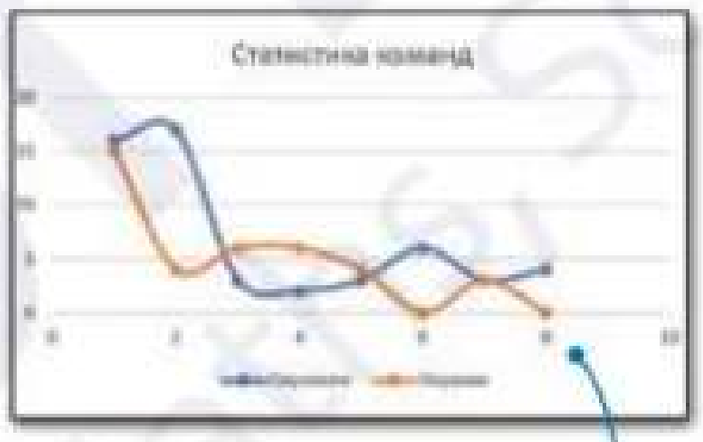
Лінійчата діаграма використовується для відображення тенденцій. Вона показує зміни даних протягом певного періоду часу. Числові значення завжди відображаються вертикально (вісь **Y**), а час – горизонтально (вісь **X**). Вона підходить для відображення даних для великої кількості груп.



Діаграма з областями схожа на лінійчату діаграму, за винятком того, що область під накресленою лінією зафарбована кольором. Вона використовується для відображення тенденцій у часі або деяких інших категорій і підходить для відображення даних для обмеженої кількості груп.



Секторна діаграма використовується для відображення лише однієї серії даних. Вона показує відношення частин до цілого. Зверніть увагу, що вона підходить для показу даних з однієї групи.



Точкова діаграма використовується для відображення значень двох серій та їх порівняння із часом. Це як лінійчата діаграма, за винятком того, що побудована лінія показує точки даних. Вона підходить для показу взаємозв'язку між двома змінними.

Кільцева діаграма використовується для відображення даних у вигляді кілець і схожа на секторну діаграму.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

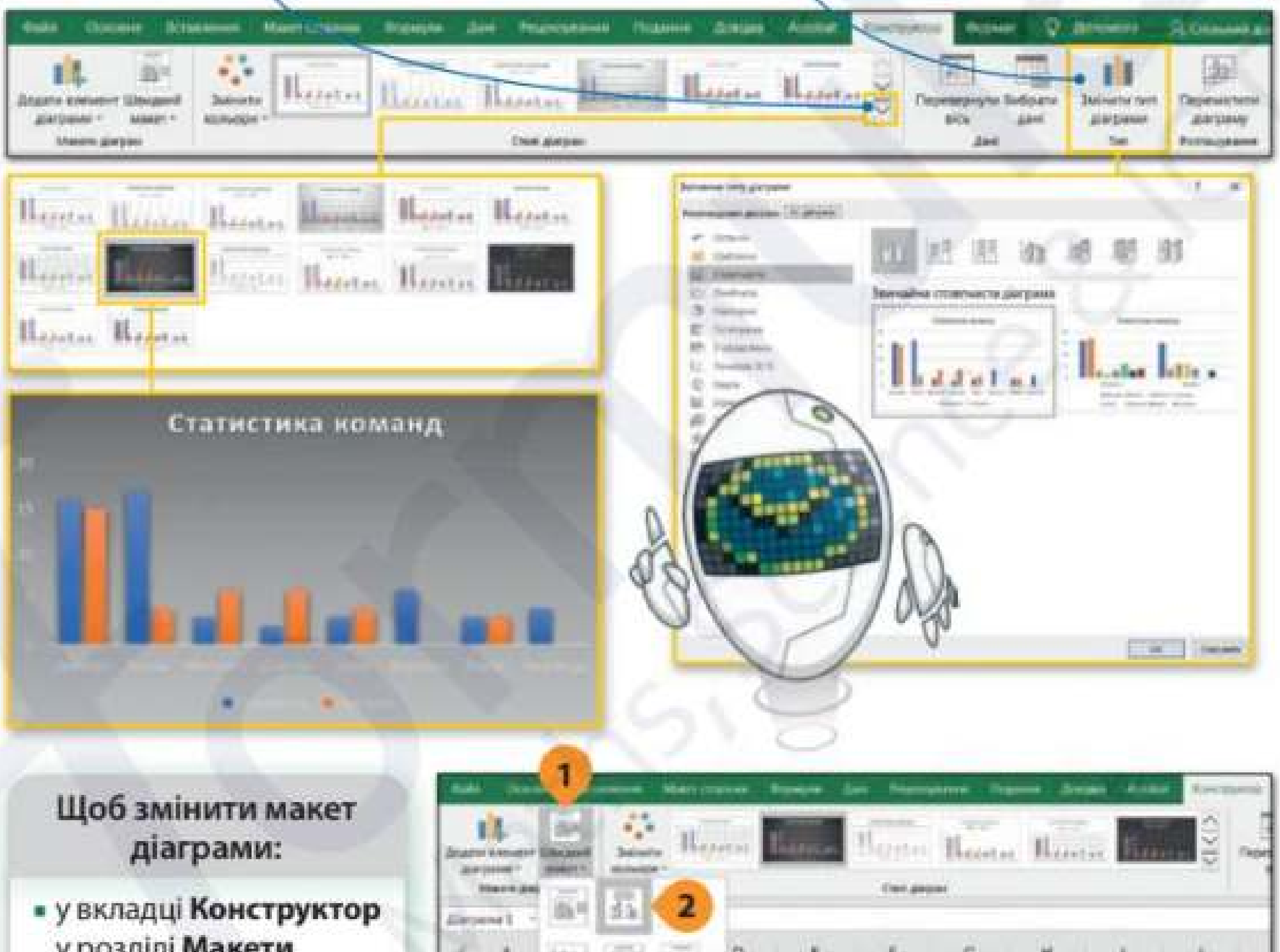
Форматування діаграм

Ви, мабуть, здогадалися, що в програмах Microsoft 365 (до 2023 року це Microsoft Office) можна налаштувати багато цікавих функцій. Те саме стосується і діаграм. Ви можете змінити все що завгодно: кольори, шрифти тощо.

Зверніть увагу, що, коли ви вибираєте діаграму, з'являються дві нові вкладки. Подивимося, як можна їх використовувати.

Клацніть на спадний список **Стилі діаграм**, щоб обрати стиль вашої діаграми.

Якщо ви хочете змінити діаграму, але не хочете починати все спочатку, скористайтеся функцією **Змінити тип діаграми** й у вікні **Змінення типу діаграми** виберіть ту діаграму, яка вам до вподоби.



Щоб змінити макет діаграми:

- у вкладці **Конструктор** у розділі **Макети діаграм** натисніть **Швидкий макет** (1) і виберіть, наприклад **Макет 2** (2);
- макет діаграми буде змінений (3).

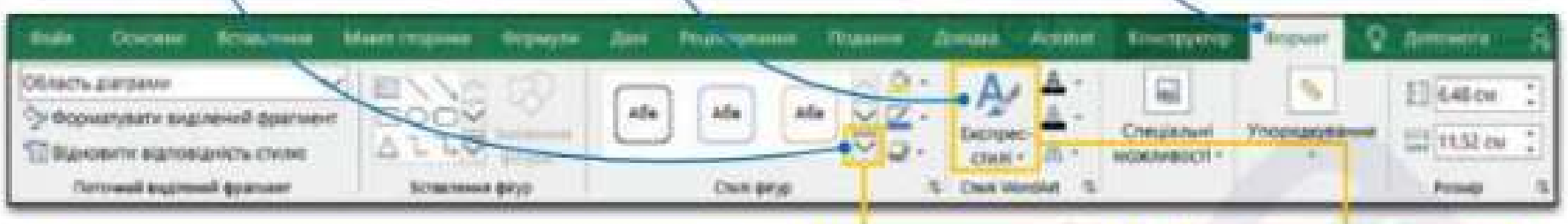
Коли вибираєте діаграму, звертайте увагу не на разючий вигляд, а на простоту й доступність відображення інформації. Звичайно, об'ємна діаграма вражає більше, але її важче зрозуміти.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Використовуйте розділ **Стилі фігур**, щоб змінити стиль фігури вашої діаграми.

За допомогою налаштувань у розділі **Стилі WordArt** можна змінити стиль символів у діаграмі.

На вкладці **Формат** ви можете відформатувати діаграму, змінити її кольори, налаштувати контури тощо. Розглянемо деякі з них.



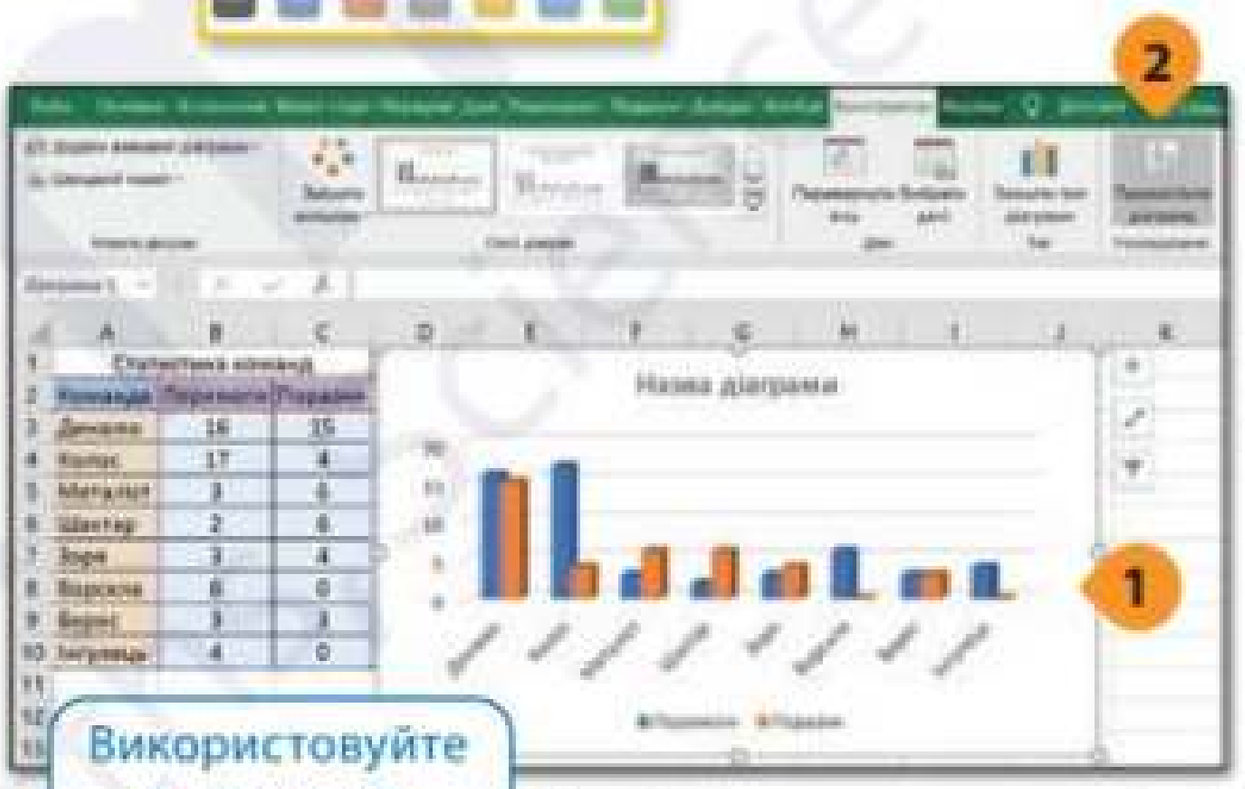
*Вкладка **Формат**, імовірно, вам уже знайома. Багато функцій є спільними для більшості програм Microsoft і не тільки. Експериментуйте! Здебільшого ви навчитеся багато чого самостійно, просто користуючись інструментами.*

Зміна діаграми

Після створення діаграми ви можете її змінити. Наприклад, ви можете змінити її назву або тип.

Щоб перемістити діаграму:

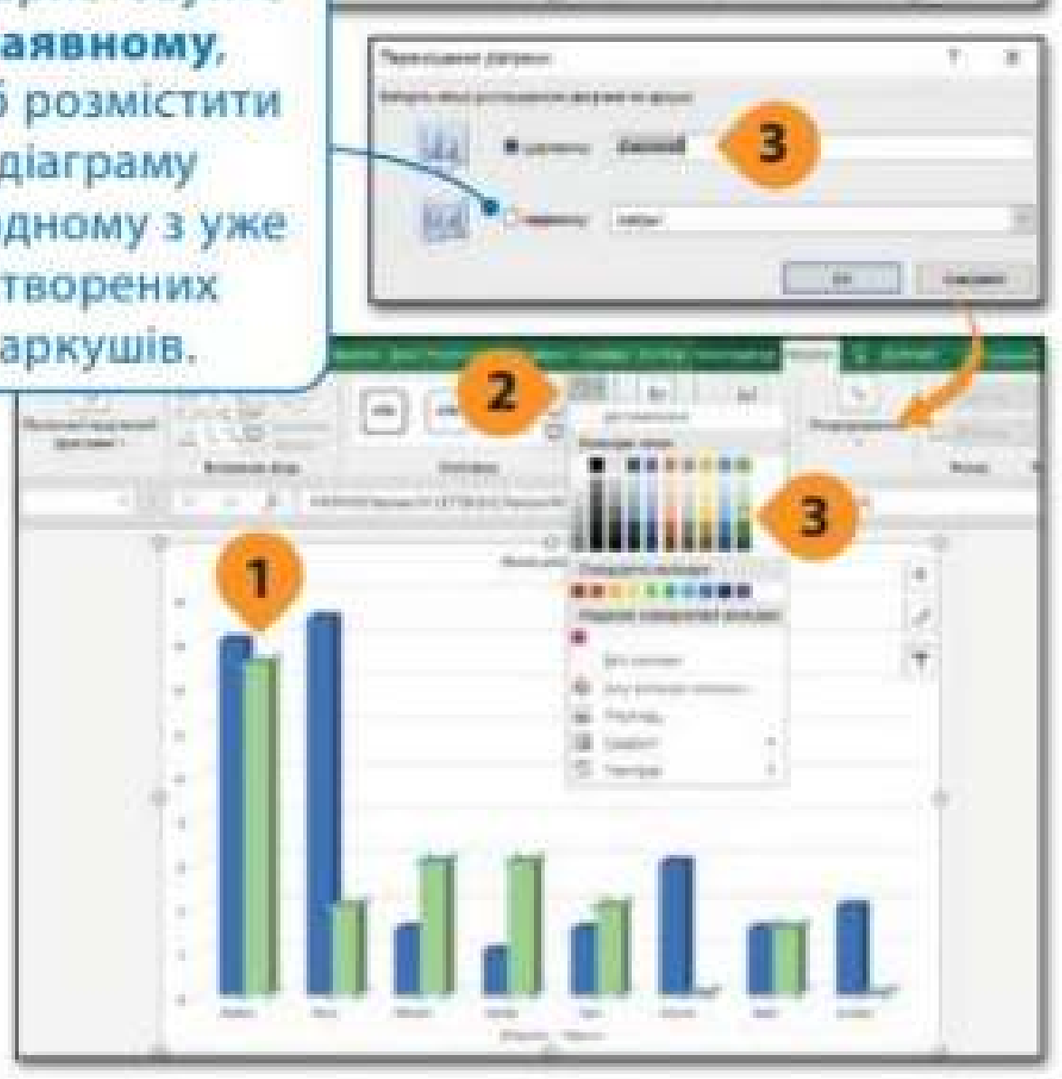
- натисніть на діаграму (1);
- на вкладці **Конструктор** у розділі **Розташування** натисніть кнопку **Перемістити діаграму** (2);
- можете перемістити діаграму на новий аркуш або на будь-який уже створений (3).



Використовуйте **наявному**, щоб розмістити діаграму на одному з уже створених аркушів.

Щоб змінити заливку фігури діаграми:

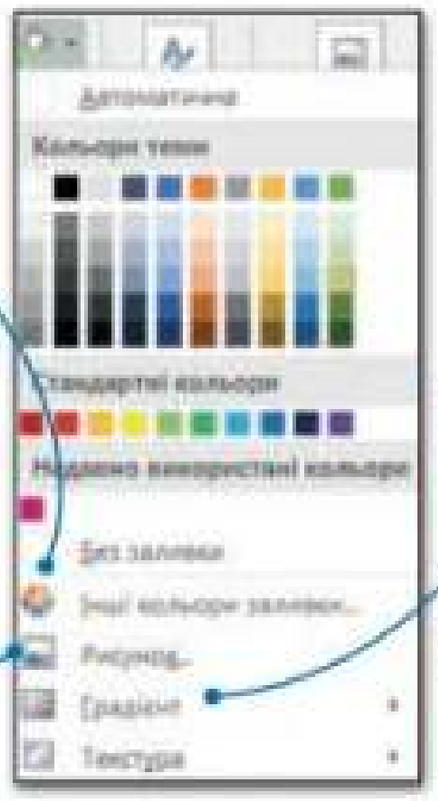
- натисніть на фігуру, до якої потрібно внести зміни. Наприклад, виберіть **Поразки** (1);
- на вкладці **Формат** у розділі **Стилі фігури** натисніть **Заливка фігури** (2);
- натисніть на вибраний вами колір (3).



© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Використовуйте **Інші кольори заливки...**, якщо хочете використати колір, якого немає в зразках **Кольори теми**.

Використовуйте **Рисунок**, коли хочете заповнити форму зображенням.



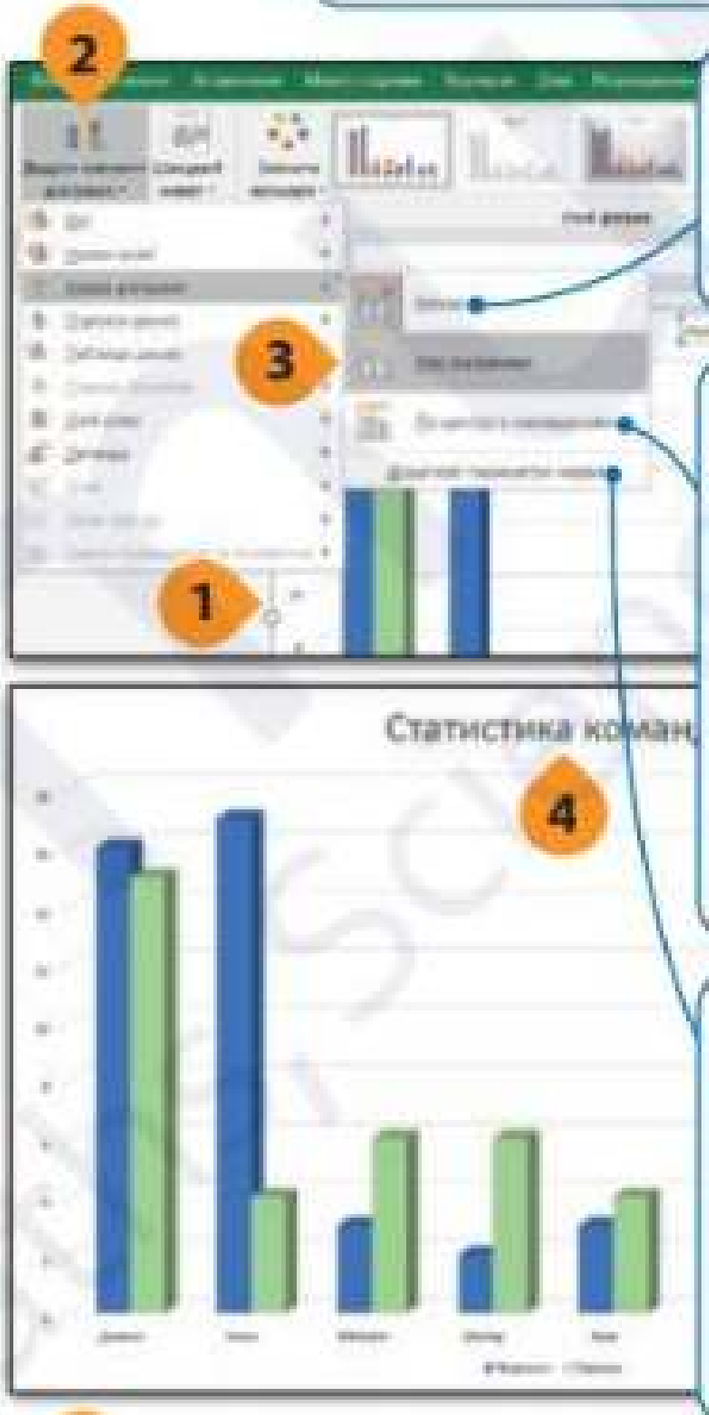
Ви можете розмістити свою діаграму на новому аркуші. У цьому випадку ваша діаграма займе весь робочий аркуш.

Використовуйте **Градієнт**, коли хочете заповнити фігуру кольором, який переходить з одного в інший.

Використовуйте **Текстуру**, коли хочете використовувати заливку з візерунком або текстурою.

Щоб додати назву діаграми:

- натисніть на діаграму (1);
- на вкладці **Конструктор** в розділі **Макети діаграм** натисніть **Додати елемент діаграми** (2);
- оберіть **Заголовок діаграми**, а потім виділить **Назва по центру** та клацніть **Над діаграмою** (3);
- натисніть на заголовок діаграми, виділить текст і видаліть його, введіть **Статистика** (4);
- клацніть будь-де поза назвою діаграми.



Використовуйте **Немає**, якщо не хочете додавати заголовок.

Використовуйте **Назва діаграми → по центру з накладанням**, коли потрібно відцентрувати заголовок на діаграмі без зміни її розміру.

Використовуйте **Додаткові параметри назви**, якщо вам потрібно більше варіантів щодо заливки, стилю рядків тощо.

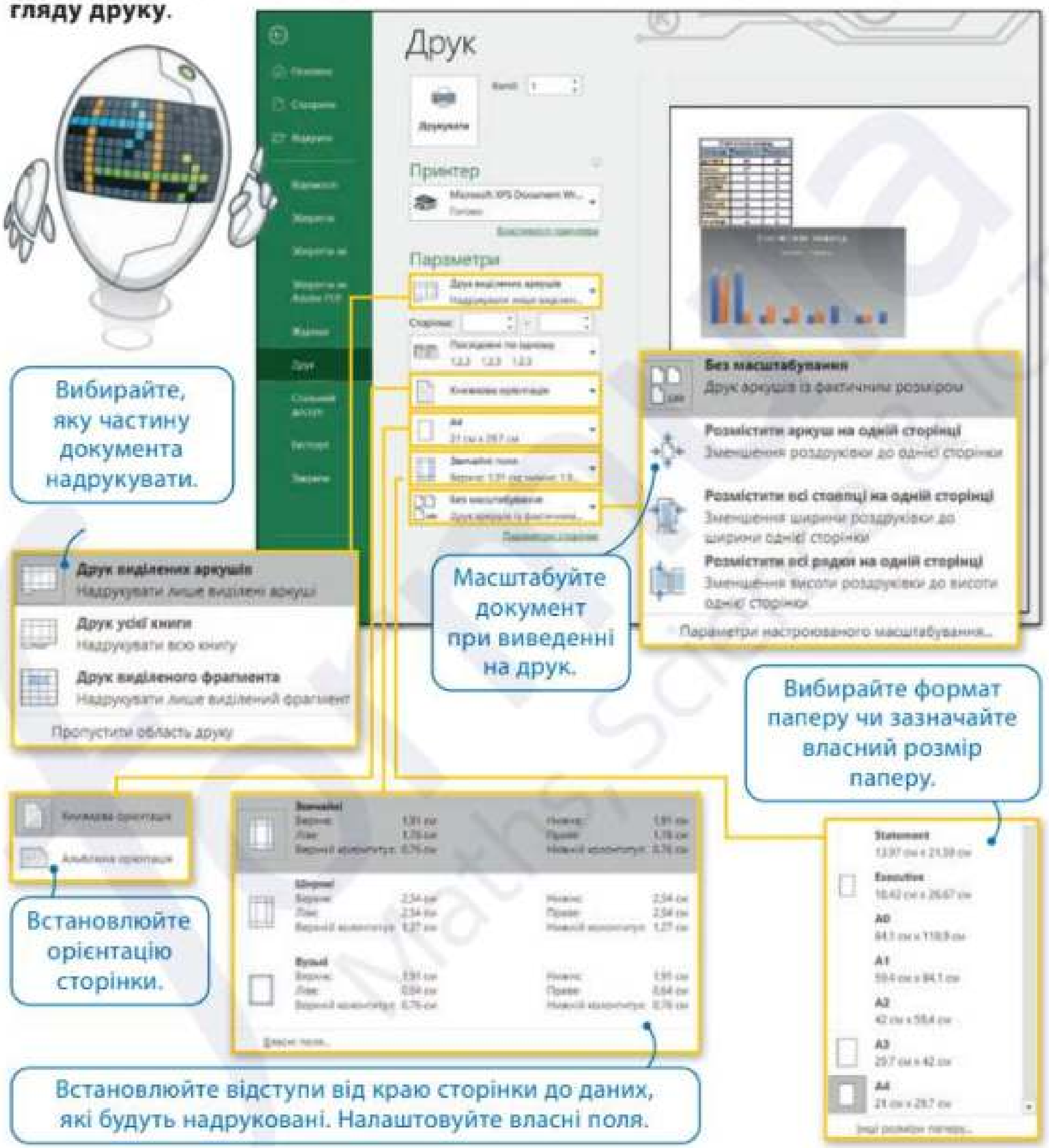
Щоб змінити формат осі:

- натисніть на діаграму (1);
- на вкладці **Конструктор** в розділі **Макети діаграм** натисніть **Додати елемент діаграми** (2);
- у спадному меню оберіть **Назви осей** та клацніть **Горизонтальна для основної осі** (3);
- натисніть на назву осі, виділить текст і видаліть його, введіть **Команди** (4);
- натисніть будь-де поза назвою горизонтальної осі.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Друк

Ви можете надрукувати аркуш із цифрами або діаграмою так само, як звичайний документ. У вкладці **Файл** клацніть **Друк** або натисніть Ctrl + P. Налаштуйте **Параметри друку** в лівій частині екрана **Попереднього перегляду друку**.



Практичне завдання

Перейдіть за посиланням на сторінку інтернет-підтримки та завантажте файл-заготовку. Поверніться до туризму в Австралії та проілюструйте дані за допомогою діаграм.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 8

Дослідження та гіпотези

Електронні таблиці можуть бути корисними не тільки в повсякденних справах, а й для обробки великих масивів даних, які є невід'ємною частиною різноманітних досліджень. На основі такого обсягу інформації, використовуючи Microsoft Excel, можна не тільки впорядкувати дані, а й робити різноманітні припущення, передбачати взаємозв'язок між даними тощо.

Процес дослідження *– систематичний підхід до вивчення певної теми чи проблеми з метою здобуття нових знань або перевірки відомих гіпотез.*

Гіпотеза *– це припущення або передбачення про взаємозв'язок між змінними, яке можна перевірити емпірично.*

За допомогою сучасних комп'ютерів та ПЗ це можна робити значно скоріше. Але не завжди є можливість перевіряти гіпотези в реальному житті. Використовуючи ПК є змога імітувати такі ситуації, перевіряти дані, аналізувати та розуміти їх. Такий процес називають моделюванням. Модель може представляти об'єкти, умови та процеси за допомогою математичних рівнянь і правил, які їх визначають. Як інструмент моделювання можна використовувати електронну таблицю.

Наприклад, компанія може використовувати електронну таблицю для з'ясування, що станеться, якщо вона підвищить ціну на свій продукт, і як це вплине на її дохід від продажів. Отже, якщо хтось підвищить значення в колонці ціни, дані в колонці доходу від продажів будуть автоматично перераховані.

Моделювання та симуляція в повсякденному житті

Чи стикалися ви з різноманітними моделями в повсякденному житті?

Моделювання та симуляція дозволяють людям тестувати систему без необхідності створювати її в реальному житті, прогнозувати, що може статися із системою в майбутньому, та навчати людей користуватися нею, не наражаючи себе на ризик.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Симуляція

Коли ви вводите значення в модель чи прикладаєте до моделі зусилля й отримуєте результати, відбувається симуляція. **Симуляція** – процес відтворення поведінки реальної системи або процесу за допомогою математичної моделі, комп'ютерної програми чи фізичної моделі.

Фінансові та бізнес-моделі

Фінансові моделі використовують математичні рівняння для проведення фінансового аналізу. Такі моделі зазвичай будують у Microsoft Excel для прийняття рішень та розробки рекомендацій. **Бізнес-моделі** дозволяють робити прогнози та формувати тенденції, пов'язані з виробничими розрахунками.

Наукові моделі

Моделювання змін клімату

Кліматичні моделі використовують математичні рівняння для опису того, як атмосфера, океани, поверхня суші, лід і сонячна енергія впливають одне на одного й на клімат Землі.

Прогнозування погоди

Комп'ютерне моделювання погоди можна використовувати для прогнозування погоди. Ці моделі беруть дані спостережень, як-от швидкість і напрямок вітру, температура повітря, тиск і вологість, зібрані з багатьох місць і джерел по всьому регіону. Потім, використовуючи математичні рівняння, передбачають, що станеться в майбутньому, зокрема появу штормів або інших погодних явищ.

Прогнозування землетрусів

Моделювання землетрусів ґрунтується на статистиці теперішніх і минулих землетрусів. Однак учені ще дуже далекі від того, щоб передбачати землетруси. Прогноз повинен бути точним щодо того, де станеться землетрус, коли він станеться та якої магнітуди він буде.

Моделювання цунамі

Якщо землетрус виникає в океані, він може створити потужні хвилі, відомі як цунамі. Чисельні моделі можна використовувати для імітації цунамі від моменту зародження до появи в прибережній зоні.

Гідрологічне моделювання

Гідрологічна модель – комп'ютерний програмний інструмент, який імітує потік і поведінку води в річковій системі. Ці моделі використовують різноманітні вхідні дані, як-от інформацію про температуру, кількість опадів і швидкість течії за певний період часу.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Молекулярне моделювання

Молекулярне моделювання – набір комп'ютерних методів, які прогнозують і моделюють хімічні реакції та їхні результати.

Навчальні моделі

Викладання математики та природничих наук за допомогою симуляцій

Багато природничо-математичних понять можна вивчати за допомогою віртуального навчання. Симуляції пропонують більш інтерактивний підхід до навчання, ніж вправи з олівцем і папером, можуть допомогти учням візуалізувати складні проблеми та ідеї.

Симуляція водіння

Симулятори водіння використовують як для розваг, так і для тренування базових навичок водіння. Система контролю правил дорожнього руху, яку використовують в автотренажерах, попереджає користувачів у разі порушення правил дорожнього руху.

Льотні симулятори

Льотний симулятор – система, яка дозволяє користувачеві відчути, як це – пілотувати певний тип літального апарата. Пілоти-стажери проводять багато годин навчання на авіасимуляторах, перш ніж їм дозволять керувати справжнім літаком.

Силові тренажери

Силові тренажери допомагають підготувати поліцейських, військових та охоронців до будь-яких ситуацій. Реалістичні тренування на основі сценаріїв відбуваються без потреби в тирі, рольових іграх, а також не вимагають присутності співробітників служби безпеки.

Моделювання громадської думки

Математичні моделі використовуються для дослідження тенденцій громадської думки в суспільстві. Ці моделі ґрунтуються на аналізі опитувань, які стосуються широкого кола питань, як-от економіка чи політика.

Промислові моделі

Комп'ютерні моделі можна використовувати для виготовлення таких об'єктів, як автомобілі чи літаки, за допомогою програмного забезпечення CAD/CAM. Інші моделі можна використовувати для перевірки безпечності спроєктованих об'єктів у разі аварії. Віртуальні об'єкти можна розбивати знов і знов, досліджувати наслідки та легко змінювати конструкцію, поки вона не стане максимально безпечною.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Будівельні моделі

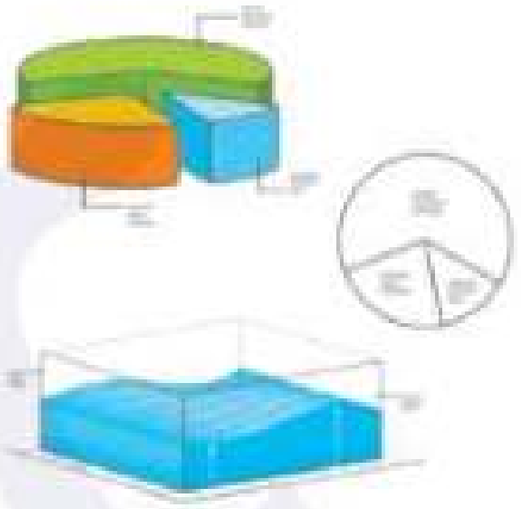
Комп'ютерні моделі інфраструктури, наприклад доріг, мостів чи тунелів, можна використовувати для тестування проєкту. Так, мости повинні витримувати екстремальні погодні умови. Відповідні моделі можна протестувати під час віртуальних штормів. На віртуальному мосту можна створити затор із віртуальних вантажівок, щоб перевірити, чи не зруйнується він.

Подібні системи використовують інженери-архітектори, особливо для дуже великих або високих будівель, як-от хмарочоси чи вежі.

Розважальні моделі

Віртуальні світи – тривимірні середовища, де можна взаємодіяти з іншими людьми та створювати об'єкти в межах цієї взаємодії. Віртуальні світи використовуються в різних сферах, зокрема в освіті й дослідженнях.

Симулятори використовують також для ігор. Можна імітувати реальні життєві ситуації, як-от управління фермою чи навіть риболовля.



Microsoft Excel має функції, які допоможуть у моделюванні фінансових, ділових і наукових гіпотез. Це пошук цілей, зведені таблиці, діаграми, звіти. Завдяки функціям аналізу можна отримати потужне рішення з підтримкою прийняття рішень та аналітикою. Для більш складних моделей і використання майже безмежних ресурсів обробки та зберігання можна перенести свої моделі в хмарні сховища, щоб скористатися перевагами хмарних обчислень.

Практичне завдання

Перейдіть за посиланням на сторінку інтернет-підтримки та завантажте файл-заготовку.

https://formula.education/online-ict-8-klas/

Інші платформи

Проєкт

Підсумки

Глосарій

© Binary Logic та © Видавництво «Пінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

3. Програмування

Хоча комп'ютери вже оснащені багатьма корисними програмами, знання програмування відкриває двері до безмежних можливостей. Програми роблять комп'ютер потужним інструментом, а знання програмування перетворює їх на ваших особистих помічників. Ви зможете створювати унікальні рішення, що відповідають вашим конкретним потребам та цілям. Це доступно кожному та приносить радість творчості.
Зануримося в захопливий світ програмування!

У цьому розділі ви дізнаєтеся:

- що таке програма;
- як розробити програму;
- які існують середовища програмування;
- як працюють середовища програмування;
- як описати розв'язок задачі;
- як написати код для розв'язання проблеми;
- як перевірити, чи коректно працює програма.

У цьому розділі ви навчитеся:

- описувати кроки рішення за допомогою коду;
- розрізняти типи даних;
- працювати зі змінними;
- використовувати команди;
- використовувати умови та цикли;
- використовувати функції;
- обробляти дії користувача;
- працювати зі списками;
- складати набори даних для перевірки роботи програми;
- перевіряти наявність помилок.

Програмне забезпечення:

- Python IDLE;
- Microsoft Visual Studio;
- Thonny;
- PyCharm;
- Python онлайн;
- Eclipse.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 1

Що таке програма

Як працюють комп'ютери

Апаратне забезпечення – термін, який ми використовуємо для опису електронних та механічних частин комп'ютера. Щоб мати змогу ним користуватися, потрібні програми. Комп'ютерна програма – список апаратних команд, який зберігається у вигляді файлу внутрішнього програмного забезпечення на накопичувачі. Коли ви запускаєте пристрій, програма, яка вбудована в комп'ютер, «читає» команди та виконує те, що програма диктує йому робити.

Як можна написати програму

Комп'ютер може розуміти лише команди, які подаються у вигляді послідовності одиниць та нулів, що називають машинним кодом. Такі команди – найнижчий рівень програмного забезпечення. Програмісти не можуть написати програму в операційній системі на машинному коді, тому вони використовують мову програмування більш високого рівня. Після того як програма написана, програмісти використовують інструменти, щоб перетворити ці команди в машинний код, який може бути виконаний комп'ютером.

Існує сотні мов програмування. Деякі з них вам уже відомі. Середовища для написання коду, відомі як інтегровані середовища розробки (IDE), онлайн-редактори або звичайні текстові редактори допомагають людям створювати програми. Програми, які перетворюють код, написаний мовою програмування, в іншу форму або мову, називають **трансляторами**.

Мова програмування використовує слова з англійської мови, а також спеціальну граматику та синтаксис, за допомогою яких можна описати команди для комп'ютера.

ДОВІДКА

Мова асемблера – низькорівнева мова програмування, у якій кожний оператор відповідає одній команді машинного коду.
Низькорівнева мова програмування максимально близька до машинного коду, а її команди менш зрозумілі для людини.
Високорівнева мова програмування максимально наближена до людської мови, дозволяє швидко писати код та легко читати й розуміти.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Типи трансляторів

Компілятори	Інтерпретатори	Асемблери
• перетворюють весь вихідний код у машинний перед виконанням; • створюють виконуваний файл.	• виконують код рядок за рядком; • не створюють окремого виконуваного файлу.	• перетворюють код асемблера в машинний код; • працюють на найнижчому рівні абстракції; • використовуються для оптимізації критичних частин програм.
Приклади: C, C++, Rust. **Переваги:** швидке виконання, виявлення помилок перед запуском. **Недоліки:** довший процес розробки, залежність від платформи.	**Приклади:** Python, Ruby, JavaScript. **Переваги:** швидкий цикл розробки, кросплатформеність. **Недоліки:** повільніше виконання, потребують інтерпретатора для запуску.	

Одна й та сама програма на різних мовах програмування!

```
' Hello world! in Small Basic
TextWindow.WriteLine ("Hello world!")
```

```
/* Hello world! in C Ansi */
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
        puts("Hello world!");
        return EXIT_SUCCESS;
}
```

```
# Hello world! in Ruby
puts "Hello world!"
```

```
//Hello world! in C#
class Hello world
{
        static void Main()
        {
          System.Console.
WriteLine("Hello world!");
        }
}
```

Транслятори відіграють ключову роль у сучасному програмуванні, дозволяючи розробникам писати код на високорівневих мовах, які потім ефективно виконуються на різних платформах.

Багато розробників використовують різні середовища для різних завдань або мов програмування. Вибір середовища для написання коду часто залежить від особистих уподобань розробника та специфіки проєкту.

Сьогодні програму написати не так і складно, це можуть не тільки фахівці – програмісти чи розробники.

Абстракція – це виділення загальних характеристик об'єктів, їхніх властивостей і методів при ігноруванні деталей реалізації. Наприклад, можна створювати моделі комп'ютерів, турбін або людського тіла без згадки окремих деталей їхньої структури та функціонування.

*Для розробки ігор частіше використовують **C++** або **C#**, для створення десктопних застосунків – **Java** чи **C#**, мобільних застосунків – **Swift**, **Kotlin** чи **Java**, **React Native** (**JavaScript**).*

© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Програмний проєкт

Програмний проєкт – це не просто готова програма, як-от текстовий чи табличний процесор, а комплексне завдання з розробки програмного забезпечення, яке охоплює кілька етапів.

Кроки створення програмного проєкту:

1	З'ясуйте, що треба вирішити, у чому полягає проблема.
2	Розбийте проблему на невеликі завдання.
3	Почніть розробляти алгоритм для кожного маленького завдання. Знайдіть кроки для їх виконання.
4	Складіть блок-схему з кроками для кожного маленького завдання.
5	Напишіть код для кожного маленького завдання.
6	Об'єднайте код усіх цих завдань, щоб створити цілу програму.
7	Протестуйте отриману програму, використовуючи коректні та некоректні дані.
8	Попросіть друзів та подруг протестувати вашу програму. Вони найкращі мисливці за багами.
9	Якщо все гаразд, програма готова. Єдине, що вам знадобиться, – доповнити її новими функціями в майбутньому.

Середовище програмування – комплексний набір інструментів, який використовується розробниками для створення, редагування, тестування та відлагодження програмного коду.

Редактор коду

Дозволяє писати та редагувати вихідний код.

Компілятор / інтерпретатор

Перетворює код на виконуваний формат.

Відлагоджувач

Допомагає знаходити та виправляти помилки в коді.

Інструменти автоматизації

Спрощують рутинні завдання розробки.

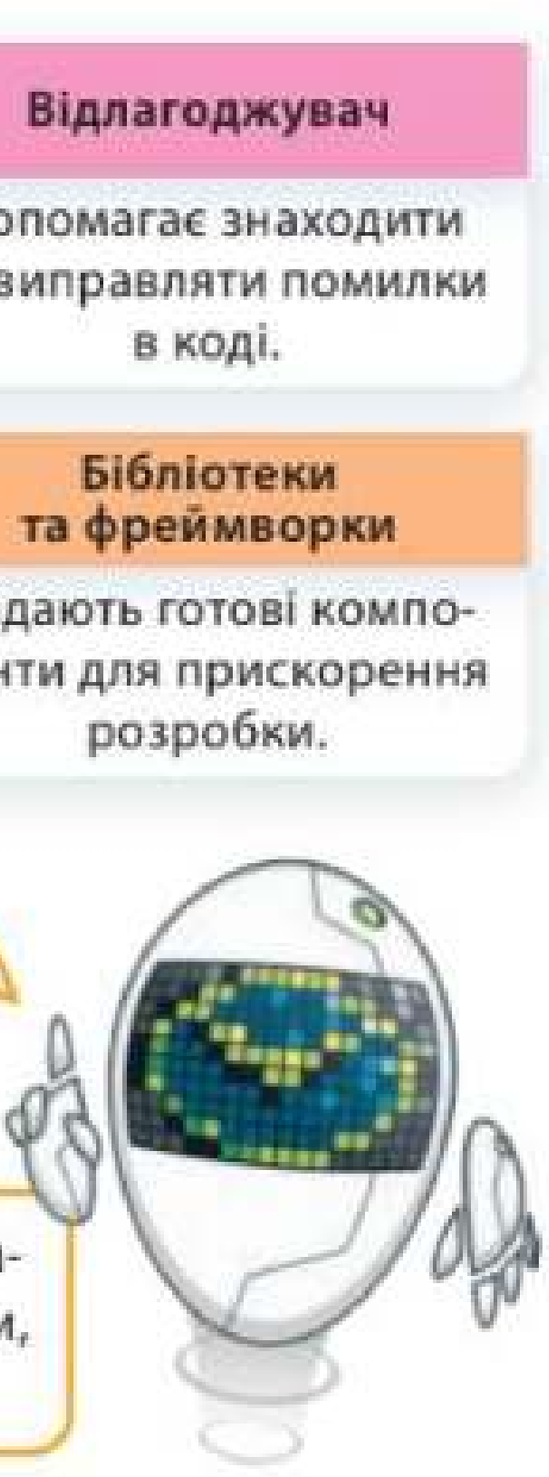
Система контролю версій

Відстежує зміни в коді та полегшує співпрацю.

Бібліотеки та фреймворки

Надають готові компоненти для прискорення розробки.

Компіляція *– процес перетворення коду, написаного людиною, у мову, яку розуміє комп'ютер.*



ДОВІДКА

Фреймворк – це як набір інструментів для розробки програмного забезпечення. Він надає готову структуру та компоненти, тож ви можете використовувати вже готові частини.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Середовища програмування:

1	Інтегровані середовища розробки (IDE)	Eclipse, Visual Studio, IntelliJ IDEA, PyCharm
2	Текстові редактори з підтримкою програмування	Sublime Text, Atom
3	Онлайн-середовища	Repl.it, CodePen, JSFiddle, JDoodle
4	Середовища для конкретних мов	IDLE (для Python), RStudio (для R), Xcode (для Swift/Objective-C)
5	Мобільні середовища розробки	Android Studio, Xcode (для iOS)

Перевірка правильності коду є важливим етапом розробки ПЗ.

На етапі компіляції коду середовище програмування виявляє синтаксичні помилки, перевіряє типи даних. Відлагоджувач шукає помилки та пропонує їх виправлення.

Важливий етап у процесі тестування – підбір наборів даних для перевірки коду. Програмісти перевіряють, як їхній код працює в різних випадках, щоб переконатися, що програма не «зламається» в незвичайних ситуаціях.

Типи наборів даних

Нормальні дані

типові вхідні дані, які програма повинна обробляти щодня.

Це дані, введення яких очікується. Вони легко обробляються в програмах.

Граничні випадки

дані на межі допустимих значень.

Наприклад, якщо працюєте з текстом – порожній рядок (""), дуже довгий текст, текст із спеціальними символами (@#$%^&*).

Екстремальні випадки

дані за межами очікуваного діапазону.

Це особливі ситуації, наприклад, порожній набір даних (немає елементів), дуже великий набір даних, набори з однаковими елементами.

Недійсні дані

неправильні або неочікувані вхідні дані.

Недійсні дані можуть призвести до помилок виконання або неочікуваної поведінки програми. Вони часто виникають через неправильне введення користувачем або помилки в логіці програми.

Додайте до коду перевірку коректності введених користувачем даних.

Валідація вхідних даних – процес перевірки та підтвердження, що дані, які надходять до програми, відповідають очікуваним форматам, типам та обмеженням. Це критично важливий аспект розробки ПЗ для забезпечення надійності та безпеки.

Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 2
Змінні та команди

Ви вже ознайомлені з Python – мовою програмування, розробленою для того, щоб зробити кодування простим і цікавим. Відкрийте Thonny, середовище, яке будете використовувати для написання та запуску коду на Python.

Структуру вашої програми можна змінювати залежно від призначення та її складності.

Змінні

Щоб робити обчислення і показувати результати, потрібно використовувати змінні. Змінна – контейнер для збереження та обробки даних у програмуванні. Для зберігання різних типів даних можна використовувати дві основні категорії змінних: числа й текст. Текстові змінні також називають рядками.

```
level=5
Score=1200
TotalAmount=120.50
```

Числа (числові змінні)

```
name='Богдан'
Message="Ви будете грати знов? Так/Ні"
email='john@binary-academy.com'
```

Текст (рядки)

Типи даних

Типи даних виникають внаслідок поділу даних на певні категорії. Тип даних задає як множину можливих значень, так і операції, які можна виконувати над цими значеннями. Python містить деякі вбудовані типи даних.

ПІДКАЗКА

Пам'ятайте, що текст для рядкових змінних потрібно брати в лапки – одинарні ' ' чи подвійні " ". Вони повинні бути однакові на початку та в кінці.

Структура програми на Python

Імпорт модулів

```
import pygame
pygame.init()
```

Глобальні змінні

```
width, height = 800, 600
```

Визначення функцій

```
def sum(a,b):
    c=a+b
    return c
```

Визначення класів

```
class MyClass:
    def __init__(self):
```

Основний код програми

```
while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
```

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

ТИП ДАНИХ	НАЗВА В PYTHON	ПРИКЛАД
Цілі числа	int	12, -999, 0, 900000
Дійсні числа	float	4.5, 0.0003, -90.5, 3.0
Символи текстові, числові та спеціальні	str	"Привіт!", "Іванка", "$$$"
Логічні (булеві) дані	bool	False, True

Список використовують для зберігання кількох елементів зі схожими характеристиками. Зазвичай ці елементи мають однаковий тип.

Списки

У Python можна зберігати дані в змінних, але їх можна розміщувати й у списках. Створити список так само просто, як ввести різні значення, розділені комами, між квадратними дужками. Кожному елементу списку присвоюється значення за допомогою **індексу**.

```
MyList=[15, 20, 35.78, 'Четвер']
```

Команда `MyList=[2]` відобразить третій елемент списку, який має значення **35.78**. Аналогічно можна змінити третій елемент, просто призначте його безпосередньо до списку, набравши `MyList[2]="Субота"`.

ПІДКАЗКА

Деякі імена не можна використовувати, оскільки вони є спеціальними словами, які вже використовуються в мові програмування. Такі слова зарезервовані системою: **print**, **return**, **while**, **True**, **False**, **else**, **and**, **not**, **import**, **None**, **global**, **break**.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Видалення та додавання елементів списку

Якщо точно відомо, який елемент списку треба видалити, скористайтеся командою **del**. Протестуйте наведений код.

```
JumpList=[6.27, 5.20, 7, 5.5, 6.2]
print(JumpList)
del JumpList[2]
print("Після видалення елемента з індексом 2")
print(JumpList)
```

```
[6.27, 5.2, 7, 5.5, 6.2]
Після видалення елемента з індексом 2
[6.27, 5.2, 5.5, 6.2]
```

Якщо треба додати елемент до списку, використайте команду **append** із назвою списку. Наприклад, щоб уставити значення 6.6 як елемент у кінець наявного списку, введіть наведений нижче код.

```
JumpList=[6.27, 5.20, 7, 5.5, 6.2]
print(JumpList)
JumpList.append(6.6)
print("Після додавання елемента")
print(JumpList)
```

```
[6.27, 5.2, 7, 5.5, 6.2]
Після додавання елемента
[6.27, 5.2, 7, 5.5, 6.2, 6.6]
```

Для отримання частини списку використайте двокрапку (:) між квадратними дужками списку. Наприклад, команда **Mylist[2:5]** повертає підсписок із третього елемента до п'ятого, не включаючи сам елемент з індексом 5.

```
JumpList=[6.27, 5.20, 7, 5.5, 6.2, 7]
print(JumpList[2:3])
print(JumpList[2:5])
```

```
[7]
[7, 5.5, 6.2]
```

Операції зі списками

len() показує кількість елементів у списку.

```
grades=[89,88,35,95]
print(grades)
gradesItems=len(grades)
print(gradesItems)
```

```
[89, 88, 35, 95]
4
```

sum() додає всі елементи списку та повертає їхню суму.

```
grades=[89,88,35,95]
myGrades=sum(grades)
print("Сума оцінок:",myGrades)
```

```
Сума оцінок: 307
```

max() показує найбільший елемент зі списку.

```
grades=[89,88,35,95]
maxGrade=max(grades)
print("Максимальна оцінка:",maxGrade)
```

```
Максимальна оцінка: 95
```

min() показує найменший елемент зі списку.

```
grades=[89,88,35,95]
minGrade=min(grades)
print("Мінімальна оцінка:",minGrade)
```

```
Мінімальна оцінка: 35
```

Коли вводите код, будьте уважні. Завжди використовуйте правильний синтаксис мови.

*Функція **sum()** працює тільки із числовими значеннями.*

ПІДКАЗКА

Перший елемент списку в Python має індекс 0, другий – 1 і далі. Тому, щоб отримати третій елемент списку **MyList**, треба надрукувати **MyList[2]**.

© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

append() можна використати для створення списку на основі вхідних даних від користувача. Значення додаються в кінець списку.

Створіть пустий список.

```
subjects=[ ]
print("введіть назву предмета",0)
subjects.append((input()))
print("введіть назву предмета",1)
subjects.append((input()))
print("введіть назву предмета",2)
subjects.append((input()))
print(subjects)
```

```
введіть назву предмета 0
фізика
введіть назву предмета 1
математика
введіть назву предмета 2
історія
['фізика', 'математика', 'історія']
```

На екран виводиться створений список.

remove() видаляє вказаний елемент зі списку.

Видаляє елементу зі списку за його значенням.

```
grades=[89,35,98,18,95]
grades.remove(18)
print(grades)
grades.remove(grades[2])
print(grades)
```

```
[89, 35, 98, 95]
[89, 35, 98]
```

Видаляє елемент зі списку по індексу.

*Функції **min()** та **max()** не працюють зі змішаними списками, які містять і числа, і символи.*

count(x) підраховує, скільки разів указаний елемент відображається в списку.

```
grades=[89,35,98,18,95]
grades.append(100)
grades.remove(18)
print(grades)
y=grades.count(100)
print(y)
```

```
[89, 35, 98, 95, 100]
1
```

sort() сортує елементи списку за зростанням.

```
grades=[89,35,98,18,95]
grades.append(100)
grades.remove(18)
print(grades)
grades.sort()
print(grades)
```

```
[89, 35, 98, 95, 100]
[35, 89, 95, 98, 100]
```

reverse() змінює послідовність елементів списку на зворотний.

```
grades=[89,35,98,18,95]
grades.append(100)
grades.remove(18)
print(grades)
grades.reverse()
print(grades)
```

```
[89, 35, 98, 95, 100]
[100, 95, 98, 35, 89]
```

clear() видаляє всі елементи списку.

```
grades=[89,35,98,18,95]
grades.append(100)
grades.remove(18)
print(grades)
grades.clear()
print(grades)
```

```
[89, 35, 98, 95, 100]
[]
```


© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Вкладені списки

Список може містити елементи різного типу, навіть інший список. Вам уже відомо, що цикл може бути всередині іншого циклу. Так само й список може бути всередині іншого списку.

```
list=[3,["a","b","c"],7.5,-2,"апельсин"]
print(list)
print(list[0])
print(list[1])
print(list[2])
print(list[3])
print(list[4])
```

```
[3, ['a', 'b', 'c'], 7.5, -2, 'апельсин']
3
['a', 'b', 'c']
7.5
-2
апельсин
```

Другий елемент списку є інший список.

Внутрішній список подібний до інших елементів списку. Він опрацьовується так, як решта елементів.

Довжина списку не змінилася.

```
list=[3,["a","b","c"],7.5,-2,"апельсин"]
l=len(list)
print("довжина списку:",l)
```

```
довжина списку: 5
```

Щоб вивести на екран елемент вкладеного списку, потрібні два числа. Перше – це індексний номер елемента в зовнішньому списку, а другий – індекс внутрішнього списку.

Виведемо на екран вкладені списки та їхні елементи.

```
list=[[1,2],['c','d'],[15,62,79]]
#print the 1st item
print(list[0])
print(list[0][0])
print(list[0][1])
#print the 2nd item
print(list[1])
print(list[1][0])
print(list[1][1])
#print 3rd item
print(list[2])
print(list[2][0])
print(list[2][1])
print(list[2][2])
```

```
[1, 2]
1
2
['c', 'd']
c
d
[15, 62, 79]
15
62
79
```

Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 3
Введення та виведення даних

Знак рівності

Знак рівності (=) в Python використовується не так, як у математиці. **SwimScore=70** означає, що змінній **SwimScore** присвоюється число 70. Знак рівності можна також використовувати для обчислення з правого боку від знаку рівності, а потім присвоїти результат змінній із лівого боку.

input()

Щоб отримати будь-яку інформацію від користувача, скористайтеся функцією **input()**.

Наприклад, можна попросити користувача гри назвати ім'я свого спортсмена. У наведеній нижче програмі після її запуску система чекає, поки користувач введе деякий текст і натисне Enter. Потім функція **input()** приймає введені дані та присвоює їх змінній **HeroName**.

```
print ("Надрукуйте ім'я свого героя ")
HeroName=input()
print(HeroName)
```

```
Надрукуйте ім'я свого героя
Stepan
Stepan
```

Функція **input()** також може використовувати рядок як повідомлення, що допоможе користувачеві зрозуміти, які саме дані потрібно ввести. Для нашого прикладу можна використати одну команду.

```
HeroName=input("Надрукуйте ім'я свого героя ")
print(HeroName)
```

```
Надрукуйте ім'я свого героя Romchick
Romchick
```

Якщо ви хочете, щоб користувач вводив числа замість тексту, просто використовуйте команди **int(input())** для цілих чисел і **float(input())** для чисел із рухомою крапкою (дійсне число).

print()

Найпростіша команда, яку можна дати комп'ютеру, – вивести дані на екран. Наприклад, змінній **SwimScore** присвоєно число **70**.

```
SwimScore=70
print(SwimScore)
```

```
70
```

Запустіть програму. У такий спосіб можна вивести на екран будь-які повідомлення або дані, які вам потрібні. **print()** у Python – це функція, що виводить вказаний об'єкт на екран.

Якщо потрібно зробити інформацію, яку виводять на екран, зрозумілішою та зручнішою для використання, змініть код.

Ви бачите різницю?

```
SwimScore=70
print("Ваш рахунок у грі ", SwimScore)
```

```
Ваш рахунок у грі  70
```


© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Обчислення

Python також можна використовувати для виконання будь-яких обчислень: додавання, віднімання, множення, ділення тощо. Правила, які ви вивчили в Microsoft Excel для використання круглих дужок, застосовуються й тут. Припустимо, що ви хочете обчислити середній бал.

Арифметичні операції

Арифметичні оператори використовуються для виконання математичних операцій, як-от додавання, віднімання, множення тощо.

//	Цілочисельне ділення
%	Залишок від ділення
**	Піднесення до степеня

** та / обчислюються перед + та -. Це означає, що **4 + 2 * 5** дає в результаті **14**, а не **30**. Використовуйте ( ), щоб розставити пріоритети в послідовності обчислень.*

Оператори присвоювання

Для надання значень змінним використовуємо **оператори присвоювання**.

=	Присвоювання	a=5
+=	Додавання з присвоюванням	b += 7 → b = b + 7
-=	Віднімання з присвоюванням	b -= 1 → b = b - 1
*=	Множення з присвоюванням	c *= 2 → c = c * 2
/=	Ділення з присвоюванням	c /= 4 → c = c / 4
%=	Залишок від ділення з присвоюванням	d %= 5 → d = d % 5
**=	Піднесення до степеня з присвоюванням	d **= 3 → d = d ** 3

Підключіть модуль math та використовуйте такі функції:

sqrt()	Квадратний корінь
ceil()	Найближче ціле (більше)
floor()	Найближче ціле (менше)
factorial()	Факторіал
fmod(a,n)	Остача від ділення числа **a** на число **n**
max()	Повертає найбільший елемент із двох чи більше аргументів
min()	Повертає найменший елемент із двох чи більше аргументів
len()	Повертає кількість символів рядка або числове значення
abs()	Повертає абсолютне значення числа
round()	Округлює число до вказаної кількості цифр
pow(x,y)	Повертає **x** у степені **y**

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Робота з рядками

За допомогою (+) можна об'єднати два чи більше рядки, або створити більший рядок. Для цього використайте рядкові змінні або безпосередньо текст. Наприклад,

```
Sport="SprintRace"
Events="100 метрів"+Sport
```

У вас є дані:

```
Distance="100 метрів"
Sport="SprintRace"
```

Додайте наступні рядки:

```
Events=Distance+Sport
print(Events)
```

На екрані з'явиться:

```
100 метрівSprintRace
```

Не забувайте додавати Space, *коли об'єднуєте текстові змінні, щоб вони були зрозумілими.*

Але більш грамотний спосіб поєднання рядків буде такий:

```
Events=Distance+" "+Sport
```

```
100 метрів SprintRace
```

Коментарі

Використовуйте коментарі, щоб пам'ятати, що робить ваш код. Вони знадобляться й у майбутньому, якщо захочете внести зміни у вашу програму. Щоб додати коментар, використовуйте (#) на початку фрази. Все, що записане в рядку коментаря, комп'ютер проігнорує.

```
# Рахунок у грі
SwimScore=70
print("Ваш рахунок у грі ", SwimScore)
```

```
import math
r = 10

'''Це коментар
Він буде ігноруватися при виконанні програми.
Тут можна написати все що завгодно'''

long = 2 * math.pi * r
print(long)
```

```
62.83185307179586
```

Щоб додати багаторядковий коментар, створіть новий рядок і візьміть свій коментар у потрійні лапки. Такі лапки дозволяють об'єднати між собою групу текстових рядків.

Введені коментарі в редакторі коду виділяються іншим кольором. Це зроблено, щоб полегшити вам читання коду.

ПІДКАЗКА

Завжди використовуйте назви змінних, які легко запам'ятати.
Наприклад, для підрахунку балів, набраних гравцем у стрибках у довжину, використовуйте таку змінну, як **LongJumpScore**.

Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 4
Умови та розгалуження

Умови

Програмісти повинні вміти навчити комп'ютер або програму приймати рішення. Це дуже важливо для створення складних, але більш якісних програм. Ці рішення приймає комп'ютер, але саме програміст повинен написати код для цього. Можна написати програму, яка буде реагувати на вхідні дані від користувача під час її виконання та видавати різні результати для різних вхідних даних, наприклад дій користувача в грі. Цей процес стає можливим за допомогою умов.

Уявіть, за ігровим сценарієм для кожної події існує час або рівень, якого гравець повинен досягти, щоб перейти на наступний рівень.

```
#Кваліфікація пройдена = 100
if TriathlonPoints<100:
    print('Спробуйте ще раз')
if TriathlonPoints>=100:
    print('Готуйтеся до наступної події')
```

*Ви використали слово **if** і двокрапку :.*
*За словом **if** слідує умова.*
*У цьому випадку **TriathlonPoints<100**.*
*Комп'ютер порівнює значення в змінній **TriathlonPoints**. Якщо воно менше 100, він виконує команди, що стоять нижче двокрапки : – умовне виконання.*

У цьому прикладі дві умови: **TriathlonPoints<100** і **TriathlonPoints>=100**. Однак немає необхідності перевіряти виконання другої умови (**TriathlonPoints>=100**). Можна змінити код так, щоб він мав такий вигляд:

```
#Кваліфікація пройдена = 100
if TriathlonPoints<100:
    print('Спробуйте ще раз')
else:
    print('Готуйтеся до наступної події')
```

else (інакше) робить ваш код набагато коротшим.

Коли ви хочете присвоїти дані змінній, використовуйте =, наприклад **RunningScore=60**. Але = можна використовувати й для порівняння, коли порівнюєте змінні чи змінну зі значенням, як-от **RunningScore==60**.

Вкладені умови

Іноді потрібно зробити вибір між більш ніж двома можливими випадками. У Python можна використовувати **elif**. Команда буде виконана, як тільки одна з умов стане істинною.

```
#Отримання медалі
if TriathlonPoints>=120:
    MedalName='Золото'
elif TriathlonPoints>=115:
    MedalName='Срібло'
elif TriathlonPoints>=110:
    MedalName='Бронза'
else:
    print('Наступного разу все вийде')
```


© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Умовні та логічні оператори

Під час написання умов можна використовувати наведені нижче умовні та логічні оператори.

Умовні оператори

ОПЕРАТОР	ЗНАЧЕННЯ
==	Дорівнює
>	Більше ніж
<	Менше ніж
>=	Більше або дорівнює
<=	Менше або дорівнює
!=	Не дорівнює

Логічні оператори

ОПЕРАТОР	ЗНАЧЕННЯ
and	Обидві сторони повинні бути істинними
or	Одна зі сторін повинна бути істинною
not	Заперечує істинність

Цикли

Припустимо, у вас є список зі спробами гравців у стрибках у довжину, і треба відобразити найвищий стрибок без використання функції **max()**.

Ви створили змінну, надали їй початкове й кінцеве значення та дозволили комп'ютеру збільшити змінну. Кожного разу, коли змінна збільшується, комп'ютер виконує команди після оператора **for**.

```
LongJumpList=[6, 7.25, 5.5, 6.2]
BestTry=0
for i in range(0,4):
    if LongJumpList[i]>BestTry:
        BestTry=LongJumpList[i]
print('Ваш найкращий результат:', BestTry)
```

Цей цикл виконується чотири рази. Змінна **i** починається з **0** й закінчується значенням **3**.

Зазвичай приріст дорівнює **1**, але його можна змінити на будь-яке значення, визначивши крок у параметрах діапазону. Можна вказати навіть від'ємний приріст, щоб рахувати назад.

```
for i in range(25,1,-2):
    print(i)
```

Значення **–2** в параметрах діапазону задає приріст змінної **i**.

```
25
23
21
19
17
15
13
11
9
7
5
3
```

Цикл **for** використовують, коли знають, скільки разів будуть виконані команди. Кількість повторів указують за допомогою параметра **range**. Але іноді невідомо, скільки разів буде виконуватися цикл. У цьому випадку скористайтеся циклом **while**, який виконується до тих пір, поки виконується задана умова.

```
#Пароль повинен мати 6 символів
password=input('Введіть пароль: ')
while len(password)<6:
    print('Ваш пароль повинен мати 6 символів. Спробуйте ще раз.')
    password=input('Введіть пароль: ')
```

Всі команди після слова **while** будуть виконуватися до тих пір, поки кількість символів у змінній **password** буде меншою за 6.

© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

За допомогою вкладених циклів можна значно скоротити код.

Виведемо на екран вкладені списки та їхні елементи.

```
list=[[1,2],['c','d'],[15,62,79]]
for i in list:
#виводить на екран елементи зовнішнього списку
    print(i)
    for j in i:
#виводить на екран елементи внутрішнього списку
        print(j)
```

```
[1, 2]
1
2
['c', 'd']
c
d
[15, 62, 79]
15
62
79
```

Оператор **break** дозволяє достроково перервати виконання поточного процесу, як-от цикли **for** або **while**, чи перервати виконання умови **if ... else**.

```
#Пароль повинен мати 6 символів
password=input('Введіть пароль: ')
while True:
    if len(password)<6:
        print('Ваш пароль повинен мати 6 символів. Спробуйте ще раз.')
        password=input('Введіть пароль: ')
    else:
        break
```

```
Введіть пароль: 321
Ваш пароль повинен мати 6 символів. Спробуйте ще раз.
Введіть пароль: 8hiO123ikj
```

Оператор **continue** повертає управління на початок циклу **for** або **while**. Він дозволяє «перестрибнути» поточний крок і перейти до наступної ітерації.

```
i = 0
while i < 9:
    i += 1
    if i == 5:
        print("Число 5 пропустимо")
        continue
        print("Це число ніхто не побачить")
    else:
        print("Поточний індекс: ", i)
```

```
Поточний індекс:  1
Поточний індекс:  2
Поточний індекс:  3
Поточний індекс:  4
Число 5 пропустимо
Поточний індекс:  6
Поточний індекс:  7
Поточний індекс:  8
Поточний індекс:  9
```

У програмуванні зазвичай існує багато способів виконання однієї і тієї самої дії. Іноді один спосіб кращий за інший. Продуктивність, розмір пам'яті та інші фактори можуть бути важливими. Вибір залишається за вами.

Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 5
Функції

Функції та події є елементами програми, які можна використовувати безліч разів. Вони дають можливість присвоювати ім'я певному набору команд і запускати їх за допомогою цих назв у будь-якому місці програми.

Користувацька функція в програмуванні – це функція, яку можна створити, щоб вирішити певні завдання / проблеми, які часто повторюються. Ви вже використовували багато допоміжних функцій: **print ()**, **input ()**, **len ()** та **range ()**.

Створення власної функції

У Python функція складається з двох частин. Перша частина – визначення функції, позначається ключовим словом **def**. Друга частина – назва функції. Після заголовка функції завжди ставлять двокрапку. Вона вказує на те, що далі йде блок команд, який повинен мати відступ.

Визначимо функцію, яка виводить на екран повідомлення **Привіт!**

def повідомляє комп'ютеру про визначення нової функції.

```
def my_function():
    print("Привіт!")
```

Назва функції.

Зверніть увагу, функції Python завжди мають круглі дужки. Якщо функція не використовує параметрів, дужки залишаються порожніми.

Щоб викликати функцію, треба написати її назву та поставити круглі дужки.

```
def my_function():
    print("Привіт!")

my_function()
```

```
Привіт!
```

Параметри та аргументи

Функція може мати параметри – значення, які присвоюють їй для виконання певної дії. Параметри схожі зі змінними, але відрізняються тим, що значення цих змінних встановлюються, коли викликають функцію.

Змінні, які додають під час визначення функції, називають **параметрами**, а значення, які вводять під час виклику функції, називають **аргументами**.

Розглянемо приклад. Визначимо функцію з назвою **printMax**, яка має два параметри **a** і **b**. Знайдемо найбільше число за допомогою простої команди **if ... else** та виведемо його на екран.

Параметри встановлюють під час визначення функції в дужках та розділяють комою. У такий самий спосіб вводять значення під час виклику функції.

ДОВІДКА

Головна перевага функції полягає в тому, що її можна викликати в основній програмі безліч разів.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

```
def printMax(a, b):
    if a > b:
        print(a, 'найбільше.')
    elif a == b:
        print(a, 'дорівнює', b)
    else:
        print(b, 'найбільше.')

printMax(3, 4)
```

Параметри

```
4 найбільше.
```

Аргументи

*Оператор **return** не можна використовувати за межами функції.*

return

У Python оператор **return** завершує виконання функції та повертає її значення. Поява команди **return** у будь-якому місці в тілі функції припиняє виконання команд функції та повертає виконання програми до місця виклику функції.

```
def squareRoot(a):
    return a*a

#основна програма
finalNumber = squareRoot(5)
print(finalNumber)
```

```
25
```

return повертає значення функції, у той час як **print** виводить на екран текст.

У наведеному нижче прикладі функція **maximum** повертає параметр із найбільшим значенням. У цьому випадку у функцію введені числа. Функція використовує просту команду **if ... else** для пошуку більшого значення, а тоді повертає це значення.

```
def maximum(x, y):
    if x > y:
        return x
    elif x == y:
        return "Числа рівні"
    else:
        return y

print(maximum(2, 7))
```

```
7
```

*Якщо не додавати оператор **return** у функцію, вона не буде повертати результат своєї дії.*

Значення аргументу за замовчуванням

Визначення значень для параметрів у функції може бути необов'язковим. Якщо користувач не хоче присвоювати значення цим параметрам, будуть використовуватися значення за замовчуванням. Для цього слід встановити ці значення.

Встановіть значення функції за замовчуванням. Якщо викликають таку функцію, але забувають вказати аргументи, функція використовує заздалегідь підготовлене значення.

```
def hello(message="Привіт"):
    print(message)

hello()

hello(message="Привіт від Python!!")
```

```
Привіт
Привіт від Python!!
```

Якщо викликати функцію без аргументу, вона буде використовувати значення за замовчуванням.

Значення за замовчуванням.

Якщо потрібно присвоїти інше значення, це можна зробити так.

ПІДКАЗКА

Щоб установити для параметрів аргументи за замовчуванням, у визначенні функції після назви параметра поставте оператор присвоєння (=), тоді введіть значення за замовчуванням.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Локальні та глобальні змінні

Змінні, які з'являються всередині функції, називають **локальними**. Їх можна використовувати тільки всередині цієї функції. Вони не взаємодіють за її межами. Різні функції можуть мати локальні змінні з однаковими іменами, і вони не будуть впливати одна на одну.

Частина програми, у якій можна отримати доступ до змінної та використовувати її, називається межею дії змінної. Локальні змінні мають межу дії в тій частині програми, у якій створені. Вони починають діяти із місця у функції, де визначена її назва, та закінчують місцем, де функція припиняє свою роботу.

Розглянемо приклад функції, у якій використовується значення **x**. Python використовує всередині цієї функції значення параметра, яке було встановлене перед тим. Присвоюємо **x** значення **2**. Значення **x**, визначене в основному коді програми, не впливатиме на значення **x**, визначене всередині функції.

Якщо викликати функцію **print** зі значенням **x** в основній частині коду, то можна побачити, що значення **x** всередині попередньої функції не вплинуло на значення **x** в основній частині коду.

Змінна **x** – локальна для функції.

```
x = 50
def func(x):
    print("x дорівнює", x)
    x = 2
    print("Змінюємо x на", x)

func(x)
print("x залишається", x)
```

```
x дорівнює 50
Змінюємо x на 2
x залишається 50
```

Коли потрібно, щоб змінна працювала в усій програмі, а не тільки в окремій її частині, зробіть її глобальною. Це можна зробити за допомогою команди **global**.

Розглянемо приклад, як за допомогою команди **global** можна зазначити, що **x** – глобальна змінна. Якщо **x** присвоїти всередині функції певне значення, це значення **x** відображатиметься також і в основному коді програми.

```
x = 50
def func():
    global x
    print("x дорівнює",x)
    x = 2
    print("Змінюємо global x на", x)

func()
print("Значення x дорівнює", x)
```

```
x дорівнює 50
Змінюємо global x на 2
Значення x дорівнює 2
```

ПІДКАЗКА

Визначення функцій допомагає зменшити кількість коду, який доводиться вводити, а також усунути дублювання коду, який має однакову функціональність. Код функції перевіряють один раз, щоб переконатися, що він правильний і працює так, як запланували.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Переваги визначених користувачем функцій

Визначення функцій допомагає зменшити кількість коду, який потрібно вводити, та усуває повторюваний код, який має однакову функціональність. Дуже зручно один раз перевірити код функції та переконатися, що він коректний і працює, як планували.

Коли ви визначили функцію, її можна викликати з будь-якого місця в програмі. Крім того, визначення функції може допомогти розділити складні проблеми на простіші.

Визначення функції покращує читабельність коду. Якщо у вас визначені функції для вашої програми, код стане легшим для розуміння, виправлення та подальшого розвитку. Це важливо, якщо хтось інший має внести зміни у вашу програму. І дуже корисно, коли ви захочете прочитати свій код через деякий час після створення, десь через місяць або пів року.

ДОВІДКА

Розробники мови Python – прихильники філософії «The Zen of Python» («Дзен Пайтона»). Ось декілька формулювань.

- Помилки ніколи не повинні проходити непомітно.
- Якщо їх приховування не прописано явно.
- Зустрівши неоднозначність, опирайтеся спокусі вгадати.
- Має бути один – і, бажано, тільки один – очевидний спосіб зробити це.
- Зараз – краще, ніж ніколи.
- Хоча ніколи, найчастіше, – краще, ніж просто зараз.
- Якщо реалізацію важко пояснити – задум поганий.
- Якщо реалізацію легко пояснити – можливо, задум добрий.
- Простори імен – чудова річ, тож робімо їх більше!

Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

https://formula.education/online-ict-8-klas/

Інші платформи

Проєкт

Підсумки

Глосарій

© Binary Logic та © Видавництво «Пінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

4. Проєктування графічного інтерфейсу

Уявіть! Ви створюєте новий застосунок для смартфона. У вас є цікава ідея, ви знаєте, які функції будуть у ньому. Але як зробити так, щоб користувачі змогли легко та із задоволенням користуватися вашим застосунком? Треба розробити зручний інтерфейс. У цьому розділі ви дізнаєтеся, як це зробити.

У цьому розділі ви навчитеся:

- використовувати інструменти у векторному редакторі;
- застосовувати заливку, контур і градієнт;
- створювати об'єкти за допомогою кривих Безьє;
- використовувати типографіку;
- створювати композиції та макети;
- створювати логотипи, банери, кнопки, піктограми, етикетки, елементи шрифтів;
- розробляти графічний інтерфейс та прототипи.

У цьому розділі ви дізнаєтеся:

- що таке графічний дизайн;
- що таке UX/UI-дизайн;
- як створити зручні інтерфейси;
- які інструменти використовують для створення графічних інтерфейсів;
- як використовувати колір та шрифт в інтерфейсі;
- чому важливо враховувати доступність інтерфейсу.

Програмне забезпечення:

- Inkscape;
- Figma;
- Xara Designer;
- Wickeditor;
- Artweaver;
- Canva;
- VistaCreate;
- Adobe Fireworks;
- Adobe Director;
- Adobe Illustrator.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 1

Графічний інтерфейс продукту. UI/UX-дизайн

Як працюють комп'ютери

Розуміння основ графічного дизайну – універсальна навичка. Знання принципів проєктування графічних інтерфейсів допоможуть вам краще орієнтуватися в цифровому середовищі, розуміти логіку створення зручних застосунків і сайтів, більш креативно мислити.

Графічний дизайн – це мова сучасної комунікації, яку варто вивчати незалежно від вашої майбутньої професії. Це допоможе вам створювати кращі цифрові продукти, ефективніше використовувати наявні технології та розвинути критичне мислення щодо дизайну, з яким стикаєтеся щодня.

Готові зануритися у світ графічного дизайну та UI/UX?

До появи інтерфейсу, який ви бачите на екрані цифрових пристроїв, зокрема комп'ютерів, смартфонів, смарт-браслетів чи фітнес-трекерів тощо, користувачі комп'ютерів послуговувалися **текстовим інтерфейсом**, який ґрунтується на використанні тексту, текстовому наборі команд і текстовій навігації.

Інтерфейс, який ви бачите на екранах сучасних пристроїв, – графічний.

Графічний інтерфейс користувача (**GUI** – Graphical User Interface) – це спосіб взаємодії людини з комп'ютером або іншим цифровим пристроєм через графічні елементи та візуальні підказки: вікна, піктограми, меню, кнопки тощо. GUI використовується не тільки в операційних системах, але й у застосунках, вебсайтах, іграх тощо.

ІСТОРІЯ

Графічний інтерфейс користувача був реалізований корпорацією **Apple Computer** у 1984 році на комп'ютері **Macintosh**. А ідею GUI запропонували вчені з дослідницької лабораторії **Xerox PARC** у 1970-х роках.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Виконання дій у GUI – це безпосередня маніпуляція з графічними елементами. GUI зазвичай не застосовують на пристроях із низькою роздільною здатністю екранів, а у відеоіграх використовують інтерфейс HUD.

HUD (Heads Up Display) – частина GUI, яка дозволяє отримати різноманітну ігрову інформацію, не викликаючи додаткові меню. Це область дисплея, де гравці можуть бачити життєво важливі дані свого персонажа, як-от поточне здоров'я, додаткові атрибути, рівень броні, кількість боєприпасів тощо. Свою назву отримала через схожість із технологією, яка була розроблена для сучасних військових літаків та дозволяє пілотові бачити важливу інформацію на лобовому склі.

ПЕРЕВАГИ GUI	НЕДОЛІКИ GUI
Інтуїтивність GUI зазвичай прості у використанні, оскільки вони застосовують візуальні елементи (кнопки, меню, іконки), які легко зрозуміти. Це дозволяє користувачам швидше навчатися та ефективно виконувати завдання.	**Високі вимоги до ресурсів** GUI потребує більше системних ресурсів (пам'яті, процесорної потужності) якщо порівняти з текстовими інтерфейсами, що може бути проблемою для старих або малопотужних пристроїв.
Візуальна привабливість Яскравий та естетично приємний інтерфейс може покращити загальне враження від програми, що робить її більш привабливою для користувачів.	**Складність у налаштуванні** Деякі користувачі можуть вважати GUI менш гнучким якщо порівняти з текстовими інтерфейсами, оскільки налаштування можуть бути обмеженими.
Багатозадачність GUI дозволяє відкривати кілька вікон одночасно, що полегшує виконання кількох завдань.	**Перенавантаженість інформацією** Якщо інтерфейс містить занадто багато елементів або інформації, це може заплутати користувача та ускладнити навігацію.
Доступність для широкої авдиторії Програми та сайти з GUI стають доступними для людей без спеціального досвіду, що розширює коло потенційних користувачів.	**Залежність від графічного обладнання** Для оптимальної роботи GUI потрібне якісне графічне обладнання, що може бути недоступним для деяких користувачів чи техніки.

Розуміння основ графічного інтерфейсу та дизайну UI/UX допоможе створювати ефективні й привабливі цифрові продукти.

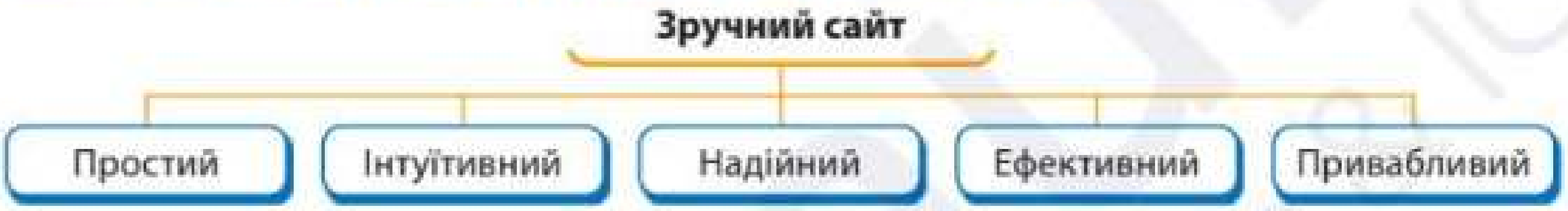
© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

На створення зручних, привабливих та ефективних інтерфейсів спрямований UI/UX-дизайн.

UI (User Interface – англ. інтерфейс користувача) – це зовнішній вигляд продукту, те що ви бачите на екрані, коли користуєтеся сайтом або програмою. UI-дизайнери відповідають за те, який вигляд має інтерфейс. Вони вибирають кольори, шрифти та графічні елементи. Головне завдання UI-дизайну – зробити інтерфейс красивим і зрозумілим.

UX (User Experience – англ. користувацький досвід або дизайн взаємодії) – це зручність використання продукту, це те, як ви почуваєтеся під час використання програми чи сайту: наскільки легко знайти потрібну інформацію, чи зрозумілі кнопки та меню, а також чи є загальне задоволення від використання. UX-дизайнери працюють над тим, щоб зробити взаємодію з продуктом приємною та інтуїтивно зрозумілою.

Поради щодо ефективного вебдизайну

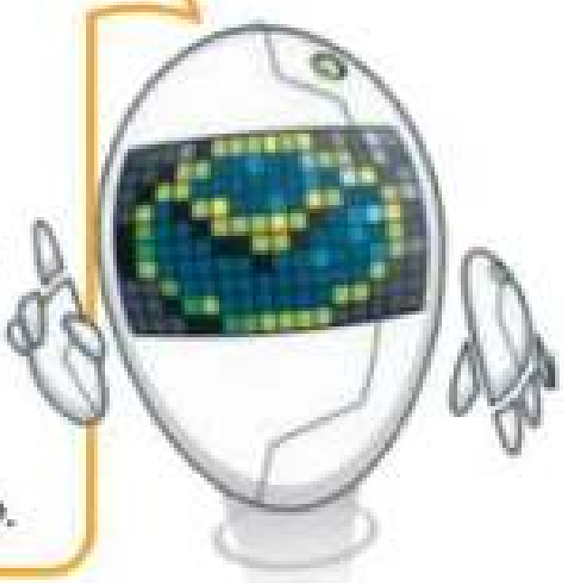


Зручність сайтів забезпечують дизайн, контент, технічна реалізація.

Інтуїтивний інтерфейс легко зрозуміти без додаткових інструкцій. Наприклад, використання звичних іконок.

Логічне розташування елементів на екрані та зручний інтерфейс дозволяють швидко, ефективно орієнтуватися та виконувати завдання.

Іноді кажуть: «UX – це душа інтерфейсу, UI – його тіло». Але разом вони створюють гармонійний та ефективний продукт, який приносить користувачам позитивний досвід. Хоча директор агентства дизайну UX / UI Awsmd Антон Міхальцов описує дизайн інтерфейсу користувача інакше: «Якщо UX – це мозок продукту, то UI – це його душа».



Головне в дизайні інтерфейсу – дотримуватися відступів між блоками інформації, не перевантажувати сторінку елементами, кольорами та шрифтами. Мінімалізм – найкращий тренд для вашого дизайну.

Під час розробки дизайну враховуйте, що сайт працюватиме на різних пристроях – смартфонах, планшетах, комп'ютерах. Дизайн сайту має бути адаптивним – він повинен мати однаковий вигляд як на екрані смартфона чи планшета, так і на моніторі ПК чи ноутбука, незалежно від їхнього розміру, роздільної здатності чи операційної системи. Тобто адаптивність – здатність дизайну й контенту продукту адаптуватися під розмір вікна браузера та інші характеристики.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Інклюзивний дизайн – це філософія, яка визнає унікальність кожної людини та прагне зробити світ доступнішим.

Інклюзивний дизайн враховує потреби різних користувачів, зокрема людей з обмеженими можливостями. Наприклад, можливість збільшення розміру тексту (екранна лупа), використання контрастних кольорів для кращої читабельності (висока контрастність), додавання альтернативного тексту до зображень для людей із вадами зору або додавання читача тексту з екрана.

ПІДКАЗКА

Гарний UX-дизайн неможливий без якісного UI, і навпаки.

Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 2

Векторний графічний редактор

Перш ніж почати створювати цифровий дизайн, важливо розуміти два основні типи графіки – растрову та векторну.

Растрова графіка

- зображення складається з пікселів;
- кожен піксель має унікальний колір;
- висока деталізація зображення;
- великий обсяг файлів через зберігання інформації про колір кожного пікселя;
- втрата якості під час масштабування;
- приклади форматів файлів: **JPEG**, **PNG**, **BMP**, **TIFF**.

Формати файлів

- **BMP** – зберігання без стиснення, досить великий розмір файлів. Формат програми Paint.
- **JPEG** – стиснення даних із погіршенням якості, розширення імен файлів jpg або jpeg. Для зберігання фотографій.
- **GIF** – стиснення без втрати даних. Підтримує анімацію та прозорість, має тільки 256 кольорів. Для мальованих та анімованих ілюстрацій.
- **PNG** – стиснення без втрати даних, підтримує прозорість фону, має більше 16 млн кольорів.
- **TIFF** – зберігання даних без втрат для поліграфії або під час сканування.

Векторна графіка

- зображення складається з графічних примітивів (прямокутників, кіл тощо);
- зберігає інформацію про форму за допомогою математичних рівнянь;
- необмежене масштабування без втрати якості;
- малий розмір файлів;
- легко редагується;
- підходить для логотипів, схем, ілюстрацій;
- приклади форматів файлів: **SVG**, **AI**, **EPS**, **CDR**.

Формати файлів

- **AI** – файли програми Adobe Illustrator.
- **CDR** – файли програми CorelDraw.
- **SVG** – зберігає також анімацію. Використовується в інженерній графіці та під час розробки вебсайтів.
- **WMF** – універсальний формат для програм, що працюють в ОС Windows.

Основні відмінності: растрове зображення нагадує мозаїку, воно реалістичне, а векторне – це математичний конструктор зображень.

Наприклад, якщо ви хочете намалювати пряму лінію за допомогою растрового зображення, треба зберігати інформацію про положення кожного пікселя прямої та його колір. Для векторної графіки треба мати тільки початкову та кінцеву точку, а всі проміжки між ними створюються з використанням математичних формул. Це дозволяє мати більш якісну графіку, навіть якщо суттєво збільшити масштаб.

Якщо збільшити растрову графіку, можна побачити пікселі, з яких складається зображення, але при збільшенні векторної графіки зображення залишається чітким.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Існує безліч векторних редакторів, які можуть допомогти у створенні якісної графіки. Ось кілька можливих варіантів:

Canva

- Платформа графічного дизайну, що дозволяє користувачам створювати графіку, презентації, афіші та інший візуальний контент.
- Користувачі можуть вибирати між багатьма шаблонами, редагувати їх та завантажувати власні фотографії.

Figma

- Онлайн-редактор, який дозволяє працювати над проєктами в реальному часі спільно з іншими користувачами.
- Зручний для вебдизайну, оскільки підтримує прототипування та інтерактивні елементи.

Inkscape

- Безплатний векторний редактор з відкритим кодом, що має потужні інструменти для редагування векторної графіки.
- Підходить для створення логотипів та ілюстрацій, а також підтримує багато форматів файлів.

Corel Draw

- Потужний векторний редактор, доступний як у вебверсії, так і для ПК.
- Має інтуїтивно зрозумілий інтерфейс і підходить для створення різноманітної графіки, зокрема інтерфесів та анімації.

Vectr

- Безкоштовний онлайн-редактор, простий у використанні й підходить для створення логотипів та ілюстрацій.
- Має можливості спільного редагування, що робить його зручним для командної роботи.

Adobe Illustrator

- Професійний векторний редактор, надає широкий спектр інструментів для створення ілюстрацій, логотипів та графіки для вебсайтів.
- Ідеально підходить для досвідчених дизайнерів завдяки своїм потужним функціям.

Ці редактори забезпечують різноманітні можливості для створення графіки: від простих логотипів, піктограм, кнопок до банерів та складних інтерфейсів для застосунків і сайтів. Вибір конкретного інструменту залежить від ваших потреб і рівня досвіду.

ІСТОРІЯ

Перший векторний графічний редактор був розроблений Айвеном Сазерлендом у 1963 році. Цю систему називали Sketchpad, вона дозволяла користувачам створювати та редагувати векторну графіку на екрані комп'ютера.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Створення векторних зображень

Типи векторних зображень

Ілюстрації
зображення, що супроводжує текст твору, статті для візуалізації змісту

Креслення
графічне зображення предмета, дані для його виготовлення та контролю виробництва

Логотипи
унікальні символи, що презентують бренд

Банери
графічні рекламні елементи на вебсторінках

Кнопки
елементи інтерфейсу для взаємодії

Елементи шрифтів
створення унікальних букв і символів

Піктограми
невеликі умовні зображення

Етикетки
інформаційні елементи для товарів

Інтерфейс
сукупність засобів, методів і правил взаємодії (керування, контролю тощо) між елементами сайту

Раніше дизайнери для створення інтерфейсів сайтів використовували здебільшого Photoshop.

Поява Figma зробила створення макетів сайтів набагато простішим. **Figma** – інструмент для роботи з векторною графікою, що дозволяє створювати різні зображення, ілюстрації, елементи інтерфейсу, анімацію, макети вебсайтів та застосунків тощо. За допомогою плагінів функціонал Figma можна значно розширити.

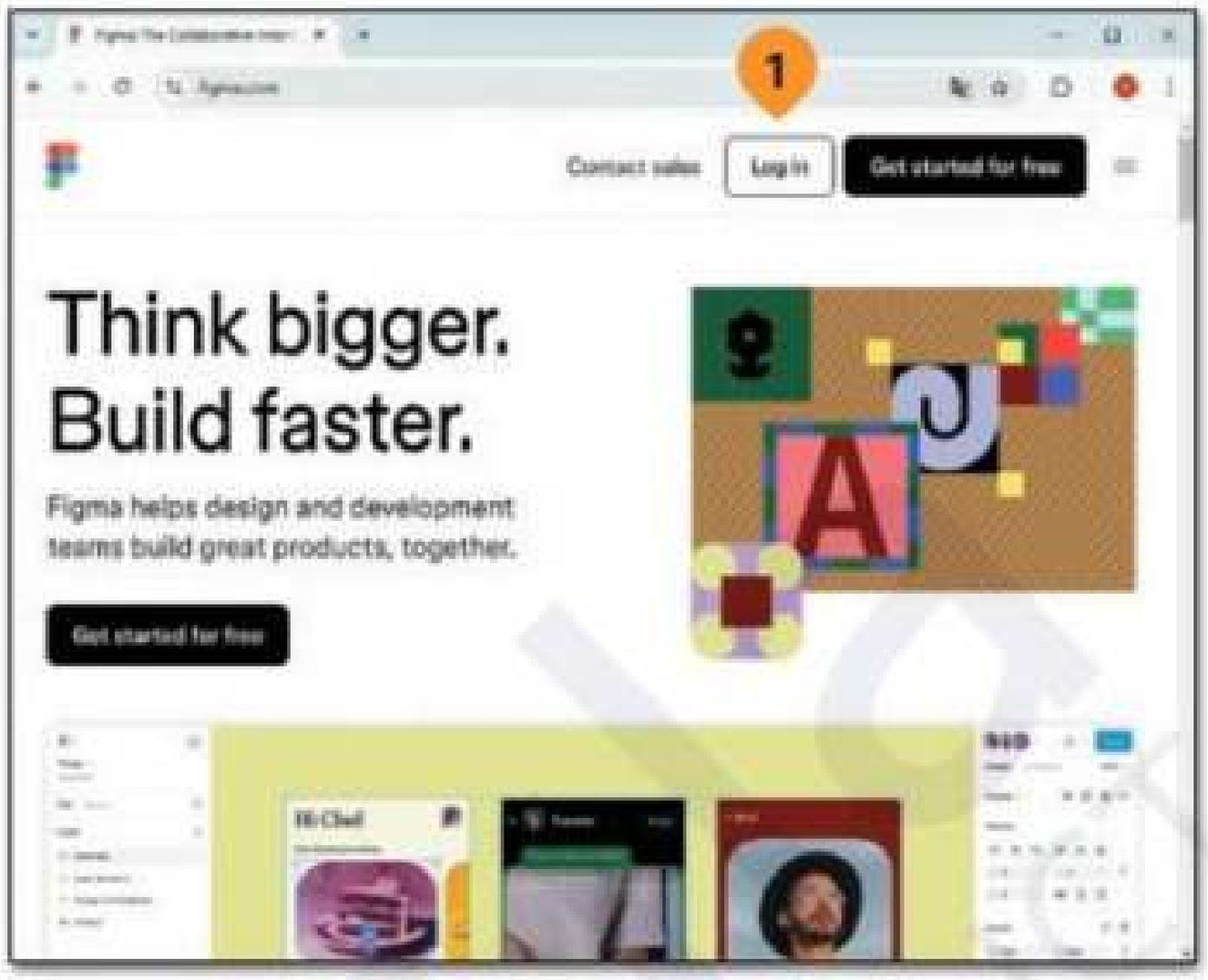
Він популярний серед веб- та UI/UX-дизайнерів, графічних дизайнерів, зручний у використанні, доступний, дає можливість працювати над проєктом віддалено разом з іншими учасниками.

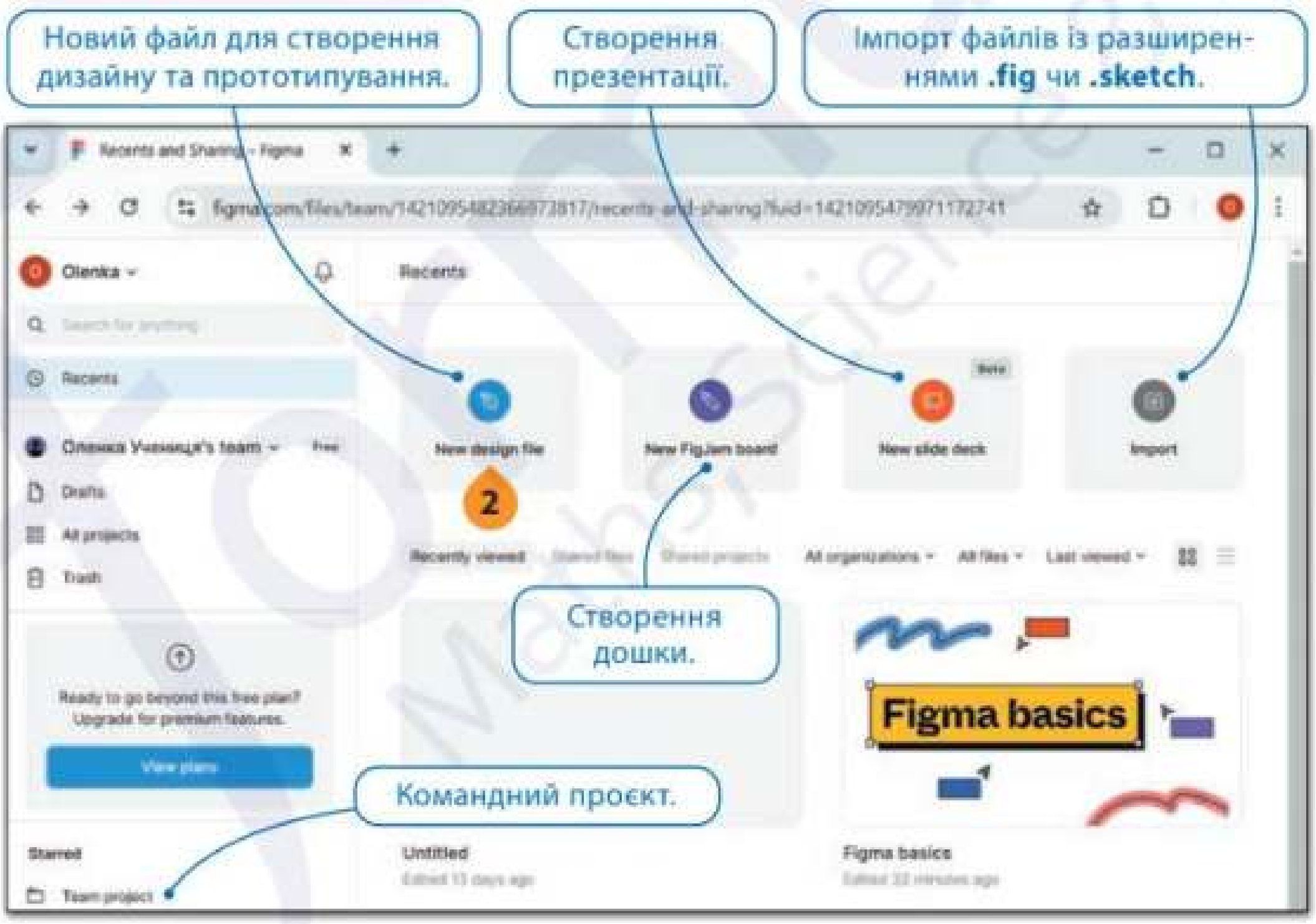
Можливість працювати у Figma онлайн або в застосунку робить інструмент універсальним.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Щоб створити новий проєкт у Figma:

- перейдіть за посиланням **figma.com** та зареєструйтеся за допомогою електронної пошти (1);
- натисніть **New design file** (2).

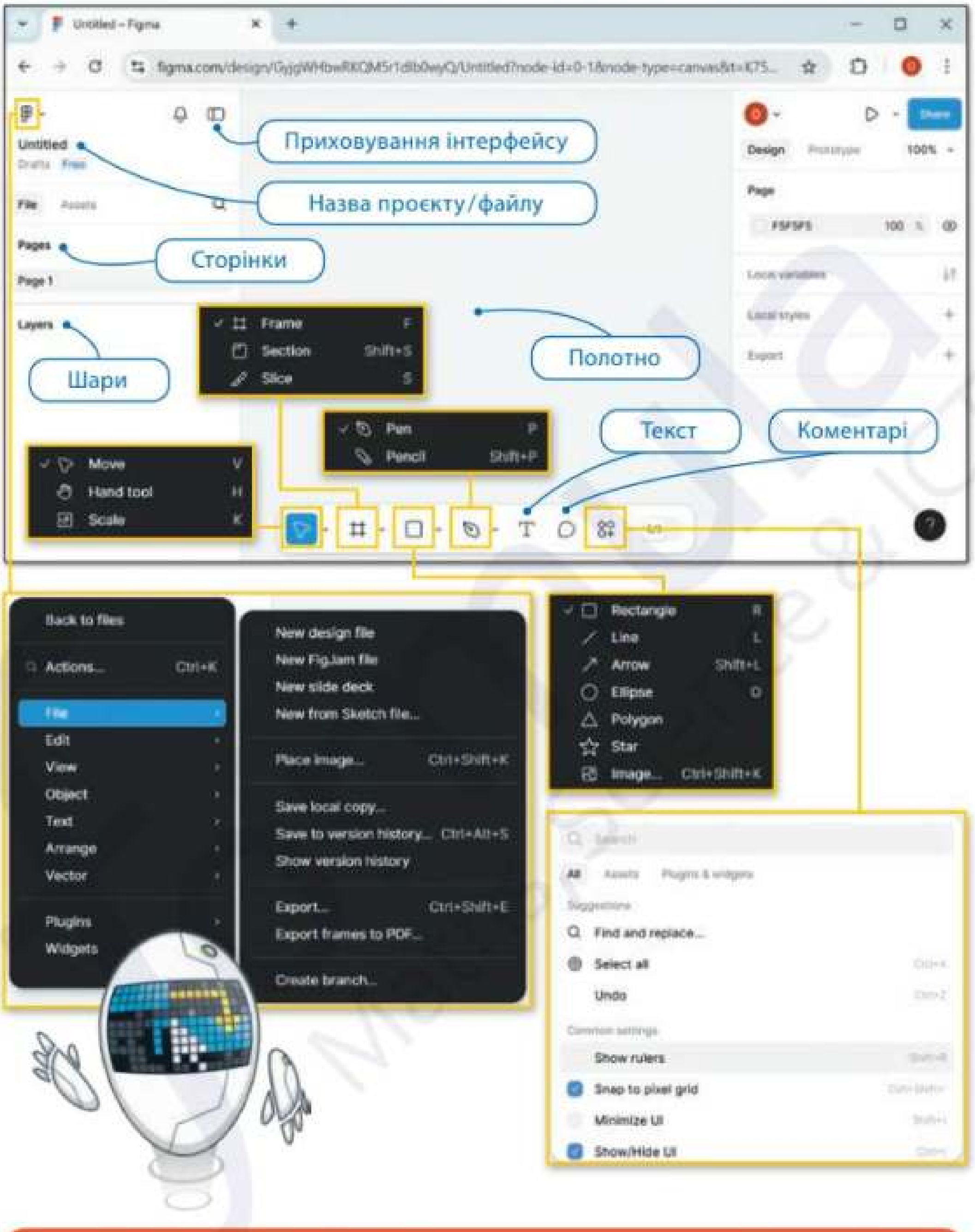




ПІДКАЗКА

Figma – це інструмент векторного дизайну. На відміну від пікселів, які мають фіксований розмір, векторні контури не залежать від роздільної здатності. Ви можете їх легко масштабувати відповідно до своїх проєктів. Можете використовувати попередній перегляд у пікселях, щоб побачити, який вигляд матимуть проєкти, якщо ви експортуєте їх як растрові ресурси.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.



Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 3
Графічні примітиви

Полотно – це тло для ваших робіт. Саме на ньому створюємо фрейми, фігури, тексти, зображення.

Фрейм (англ. frame – рамка, каркас) – основа для створення структурованих та організованих дизайн-макетів у Figma, це базовий елемент документа, ще їх називають **артборди**. Він як сторінка паперу, на якій створюють проєкт чи малюють зображення. Фрейми допомагають візуалізувати, компонувати та експортувати елементи проєкту.

У процесі створення графіки ви малюєте графічні примітиви, а потім змінюєте їх та перетворюєте на більш складні фігури й зображення. Фігура в графічному редакторі складається з двох компонентів – заливки та контура, кожен із яких має власний колір. Заливка та контур фігури – незалежні, і ви можете переміщати, видаляти чи змінювати деякі з них навіть після того, як створили фігуру.

***Графічні примітиви** – основні геометричні фігури та елементи, що використовують для створення та побудови більш складних графічних об'єктів. Вони як «цеглинки» для створення візуальної інформації.*

У графічних та векторних редакторах шари – унікальний інструмент організації та управління елементами зображення.

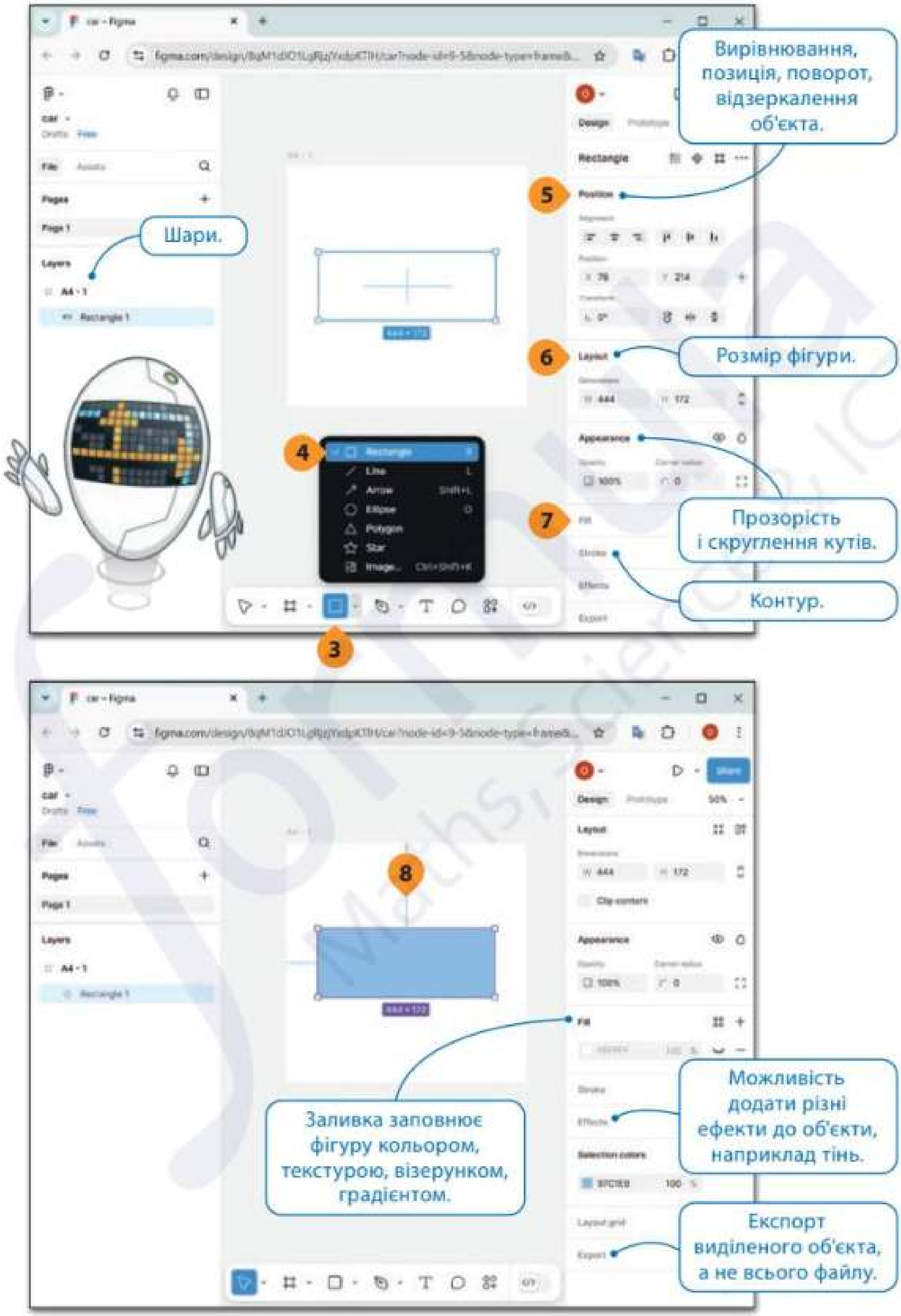
Кожен шар – окремий елемент зображення. Це дає можливість їх швидко редагувати без впливу на інші частини зображення. Можна заблокувати шар, щоб захистити його від редагування, чи зробити невидимим.

Робота із шарами дозволяє створювати складні багатошарові зображення з максимальною зручністю та контролем.

Щоб створити графічний об'єкт:

- створіть **New design file** у Figma;
- натисніть **Frame** (1) та оберіть бажаний формат полотна (2);
- клацніть (3) та оберіть фігуру (4);
- на панелі інструментів налаштуйте розташування фігури (5), її розмір (6), оберіть колір заливки (7) та інші параметри;
- вирівняйте відносно центру (8).

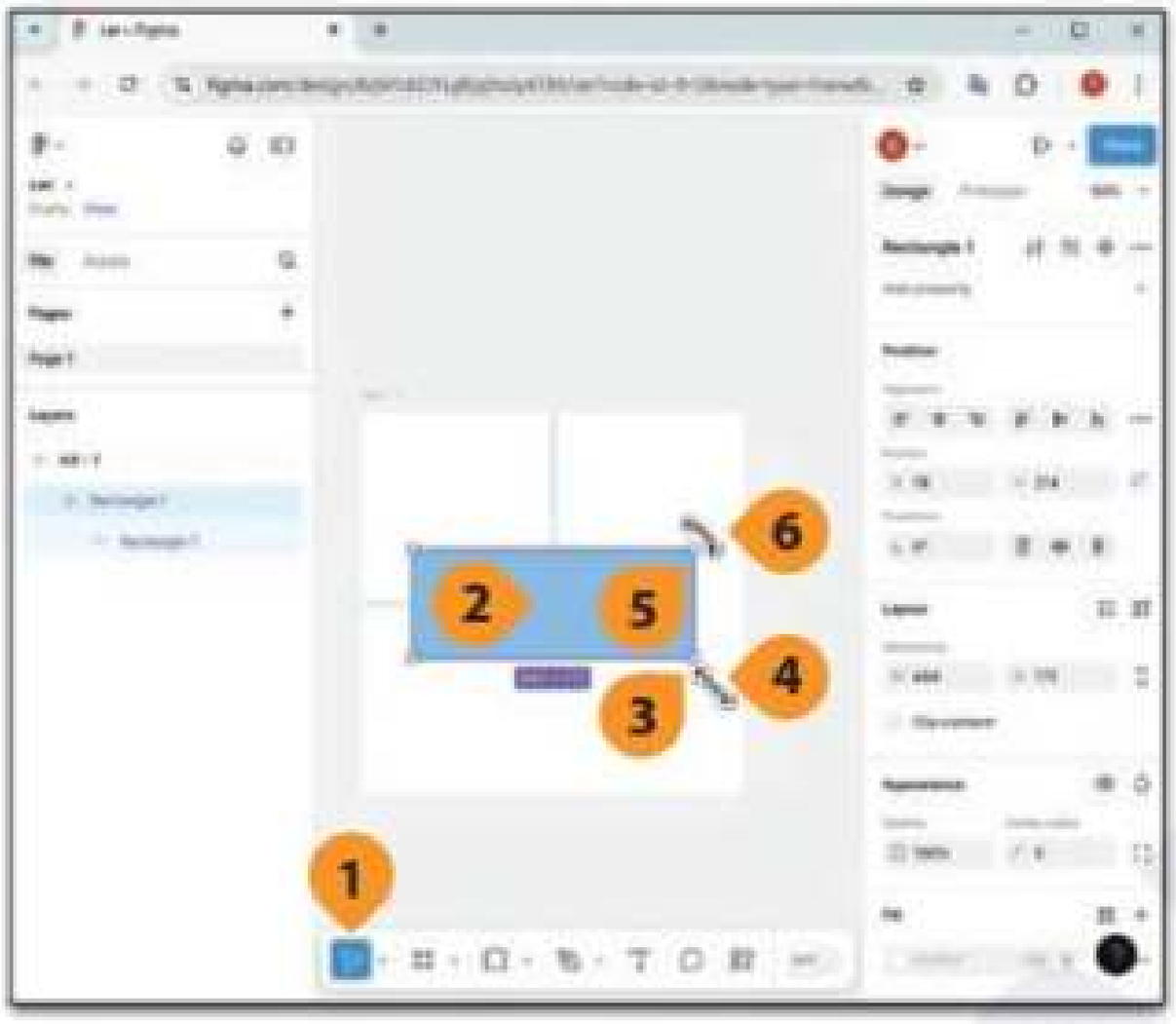
© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.



© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Інструмент Виділення

Інструмент **Виділення** дає змогу редагувати фігуру більш детально.



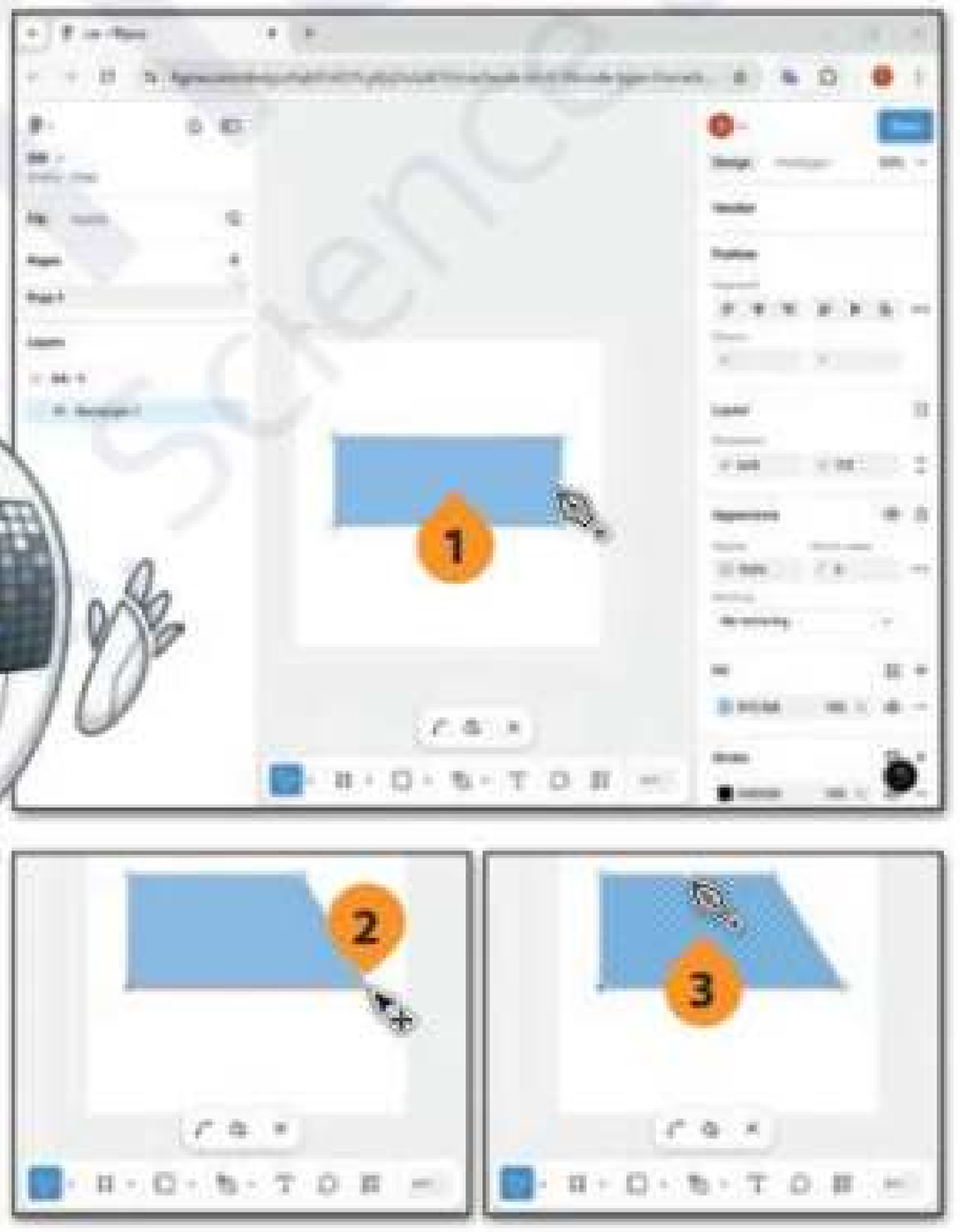
Щоб змінити фігуру:

- виберіть інструмент **Move** (1) на панелі **Інструментів**;
- клацніть усередині фігури (2), щоб виділити її;
- наведіть курсор на вузол у куті (3) до появи стрілки (4);
- потягніть за неї для зміни розміру;
- наведіть курсор біля вузла в куті (5) до появи стрілки (6);
- поверніть стрілку навколо центра для зміни кута нахилу.

За допомогою функцій інструменту **Виділення** є можливість створювати складніші фігури. Наприклад, вигнувши інші сторони вашого прямокутника, можна створити форму автомобіля.

Щоб трансформувати фігуру:

- двічі клацніть всередині фігури (1);
- наведіть вказівник миші на правий нижній кут прямокутника, затисніть його. Коли вказівник зміниться, перетягніть його праворуч (2), щоб змінити фігуру;
- наведіть вказівник миші на сторону і, коли вказівник (3) зміниться, знову клацніть;
- натисніть на інструмент **Криві** (4) та потягніть назовні, щоб створити криву (5).



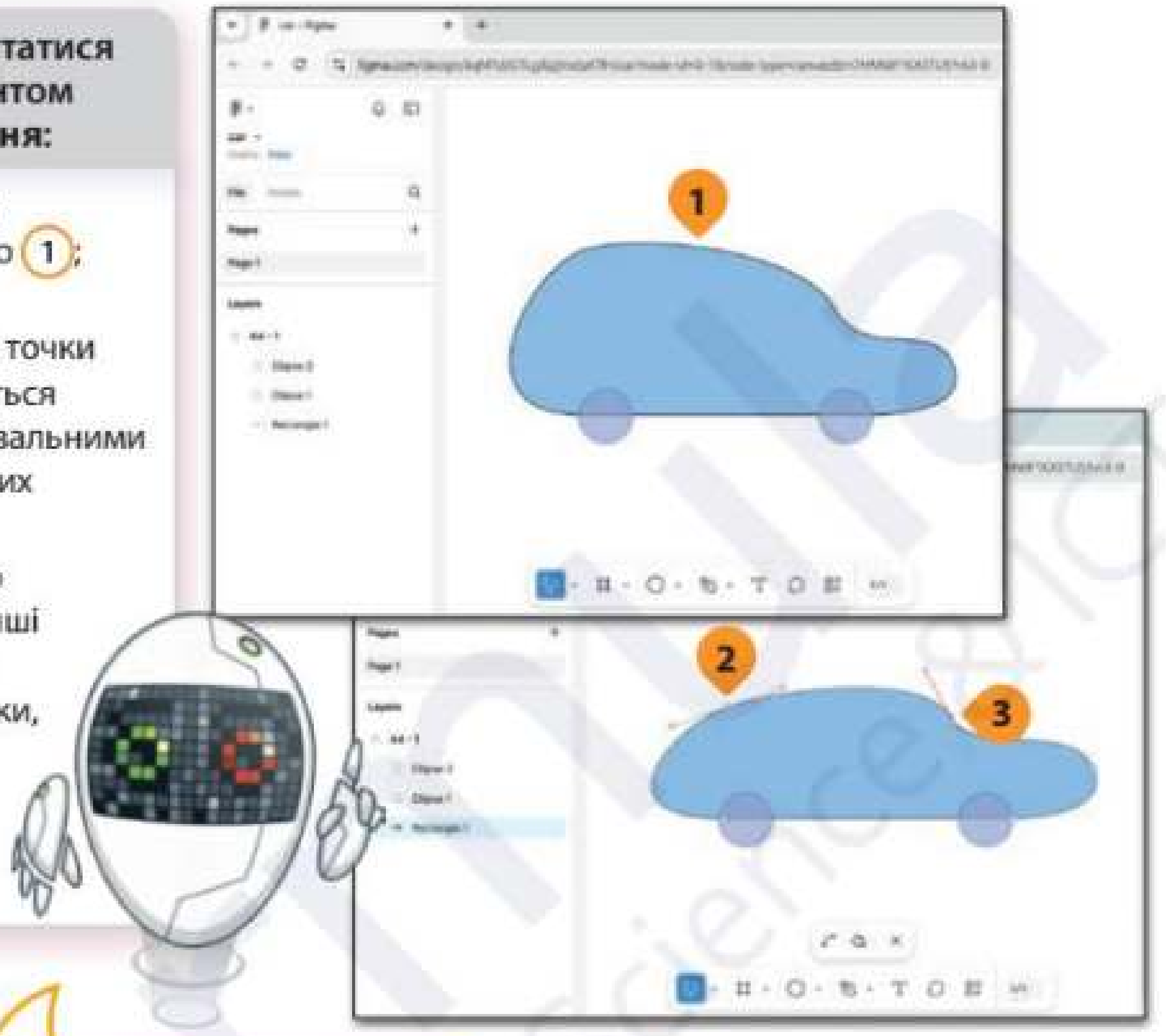
Керувальні вузли, або вуса Безьє, дають можливість змінювати форму відрізка, управляти нахилом і вигином кривої лінії контуру.

© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Інструмент **Виділення** насправді набагато потужніший, ніж просте виділення. Він має режим швидкого редагування, який дозволяє змінювати форму фігур.

Щоб скористатися інструментом Виділення:

- клацніть двічі на контур авто (1);
- окремі кути та контрольні точки фігури з'являться разом із керувальними вузлами кривих Безьє (2);
- за допомогою вказівника миші переміщайте ці вузли й точки, що дає вам можливість змінювати фігуру (3).



***Крива Безьє** – параметрично задана крива, яку використовують у комп'ютерній графіці. У векторній графіці криві Безьє використовують для моделювання гладких кривих, які можна масштабувати до нескінченності.*

Щоб створити ідеальне коло, утримуйте клавішу Shift ↑ під час створення фігури.

Інструмент **Олівець** дає можливість намалювати будь-яку фігуру довільної форми.

Інструмент **Перо** дозволяє малювати контури за допомогою кривих Безьє.

ІСТОРІЯ

Криві Безьє були запроваджені в 1962 році П'єром Безьє в компанії Renault. Але ще в 1959 році їх використовував Поль де Кастельє з компанії Citroën, та його дослідження не публікували. Компанія їх приховувала як комерційну таємницю до кінця 1960-х років.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Інструмент Еліпс

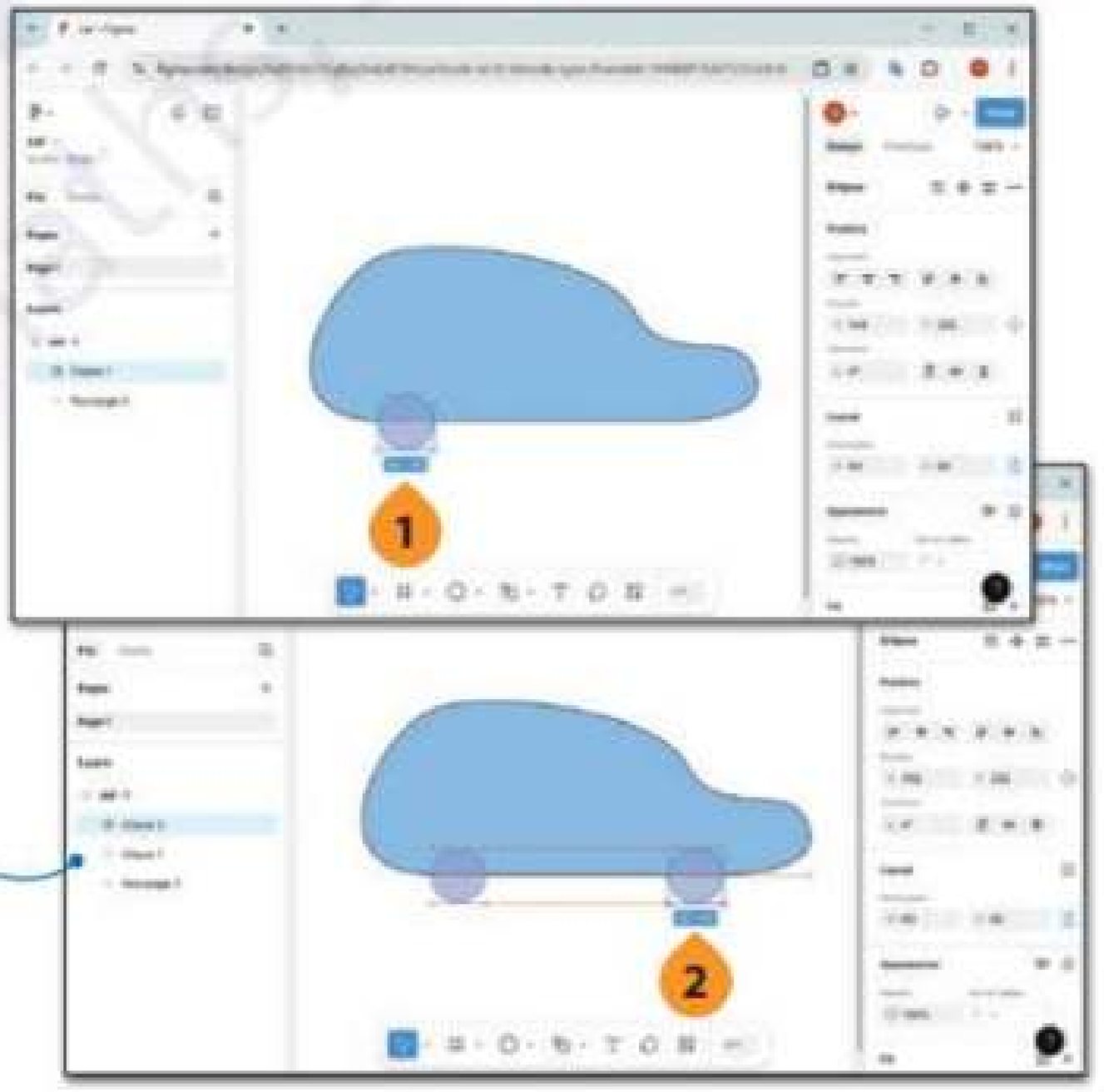
Для створення колеса нашого автомобіля використайте інструмент **Еліпс**.

Щоб створити колесо:

- оберіть інструмент **Ellipse** (1);
- утримуйте Shift ↑, клацніть лівою кнопкою миші та потягніть, щоб намалювати коло (2);
- оберіть **Fill** (3) та клацніть (4). Натисніть панель кольорів (5). Оберіть колір (6), відтінок (7), прозорість (8). Закрийте вікно (9);
- повторіть дії щодо кольору з інструментом **Stroke** (10);
- оберіть **Move** (11);
- затисніть ліву кнопку миші на фігурі та перетягніть коло, щоб розмістити фігуру в бажаному місці.

Щоб скопіювати та вставити фігуру:

- клацніть на коло, щоб виділити його (1);
- натисніть Ctrl + C, щоб скопіювати фігуру;
- клацніть Ctrl + V, щоб уставити її. Копію буде розміщено поверх виділеного об'єкта;
- перемістіть копію, перетягнувши її туди, де має бути колесо авто (2).



Порядок шарів, у якому фігури розташовані на панелі, визначаються порядком їхнього створення. Фігура, яку створили останньою, накриє будь-які ділянки фігур, створених раніше. Саме тому колеса нашого автомобіля покривають кузов.

© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

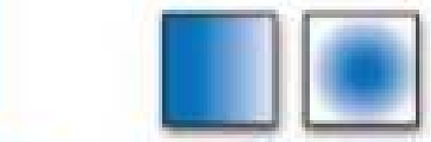
Градієнти

Створіть вікно для автомобіля та використайте гарний градієнт як колір.

Щоб створити градієнт:

- оберіть інструмент **Pen** (1), намалюйте форму вікна авто;
- на панелі параметрів інструменту оберіть **Fill** (2) та клацніть + (3);
- натисніть (4) для відкриття панелі кольорів, оберіть **Gradient** (5);
- у спадному меню оберіть **Лінійний градієнт** (6);
- клацніть лівий маркер (7), оберіть колір, відтінок і прозорість (8). Закрийте вікно (9);
- повторіть дії з правим маркером (10);
- закрийте вікно (11).

Розрізняють лінійні та радіальні градієнти



© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Інструменти вирівнювання та розподілення

Інструменти вирівнювання та розподілення у Figma – функції, які дозволяють точно розташувати графічні елементи на артборді. Вони дозволяють миттєво вирівняти об'єкти по горизонталі та вертикалі: по лівому, правому краях, по центру або з рівномірними інтервалами між ними.

Такі інструменти допомагають створювати професійні макети з бездоганною геометрією та естетикою, забезпечуючи абсолютну точність.

Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 4

Типографіка. Ефекти та фільтри

Чи може така дрібниця, як шрифт, змінити значення слів або весь дизайн презентації, плаката, книги, сайту?

Мистецтво володіння типографікою, безсумнівно, одна з базових цифрових навичок. Зрештою, ця річ може зробити дизайн кращим або зруйнувати його. Шрифт може додати новий рівень тексту та значення вашому повідомлення. Це може допомогти вам налагодити контакт із користувачем, сформувати бренд і задати тон усьому проєкту. Неправильний шрифт може зробити дизайн плоским, розрізненим або навіть дати користувачам неправильне уявлення про ваш бренд.

Типографіка *– це мистецтво оформлення тексту на екрані чи в друкованих матеріалах. Вона охоплює вибір шрифтів, розміру, інтерліньяжу, кернінгу, а також використання кольорів та інших ефектів для покращення сприйняття інформації. Правильне використання типографіки є важливим елементом дизайну інтерфейсів та вебсайтів, підручників і книг, плакатів, листівок, іншої друкованої продукції.*

У **Figma** можна налаштувати різні типографічні параметри, щоб створити привабливі текстові елементи.

Параметри типографіки

Вибір шрифту

Figma надає широкий вибір системних та вебшрифтів, які можна використовувати у проєктах. Завантажуйте додаткові шрифти з Google Fonts чи інших джерел.

Розмір

Дозволяє задати оптимальний розмір тексту для різних елементів: заголовків, підзаголовків, основного тексту. Оптимальний розмір залежить від цілі використання та платформи.

Інтерліньяж

Це відстань між рядками тексту. Правильний інтерліньяж покращує читабельність і сприйняття тексту.

Вирівнювання

Дозволяє задати відповідне вирівнювання тексту – за лівим краєм, по центру, за правим краєм або по ширині сторінки.

Кернінг

Це налаштування відстані між окремими літерами. Підбір правильного кернінгу робить текст більш гармонійним та естетично приємним.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Окрім типографічних налаштувань, Figma також дозволяє застосовувати різноманітні ефекти та фільтри до текстових елементів.

Ефекти та фільтри

Тінь
Додайте тінь до тексту, щоб надати глибини та виразності.

Контур
Намалюйте контур навколо тексту для виділення.

Розмиття
Застосуйте ефект розмиття, щоб створити ілюзію глибини чи фокусу.

Градієнт
Заповніть текст плавним переходом кольорів для більшої виразності.

Накладання
Накладіть текст на фонове зображення чи колір для створення цікавих композицій.

Експериментуючи з різними типографічними налаштуваннями та ефектами у Figma, можна створювати унікальні, привабливі та функціональні текстові елементи для своїх дизайн-проєктів.

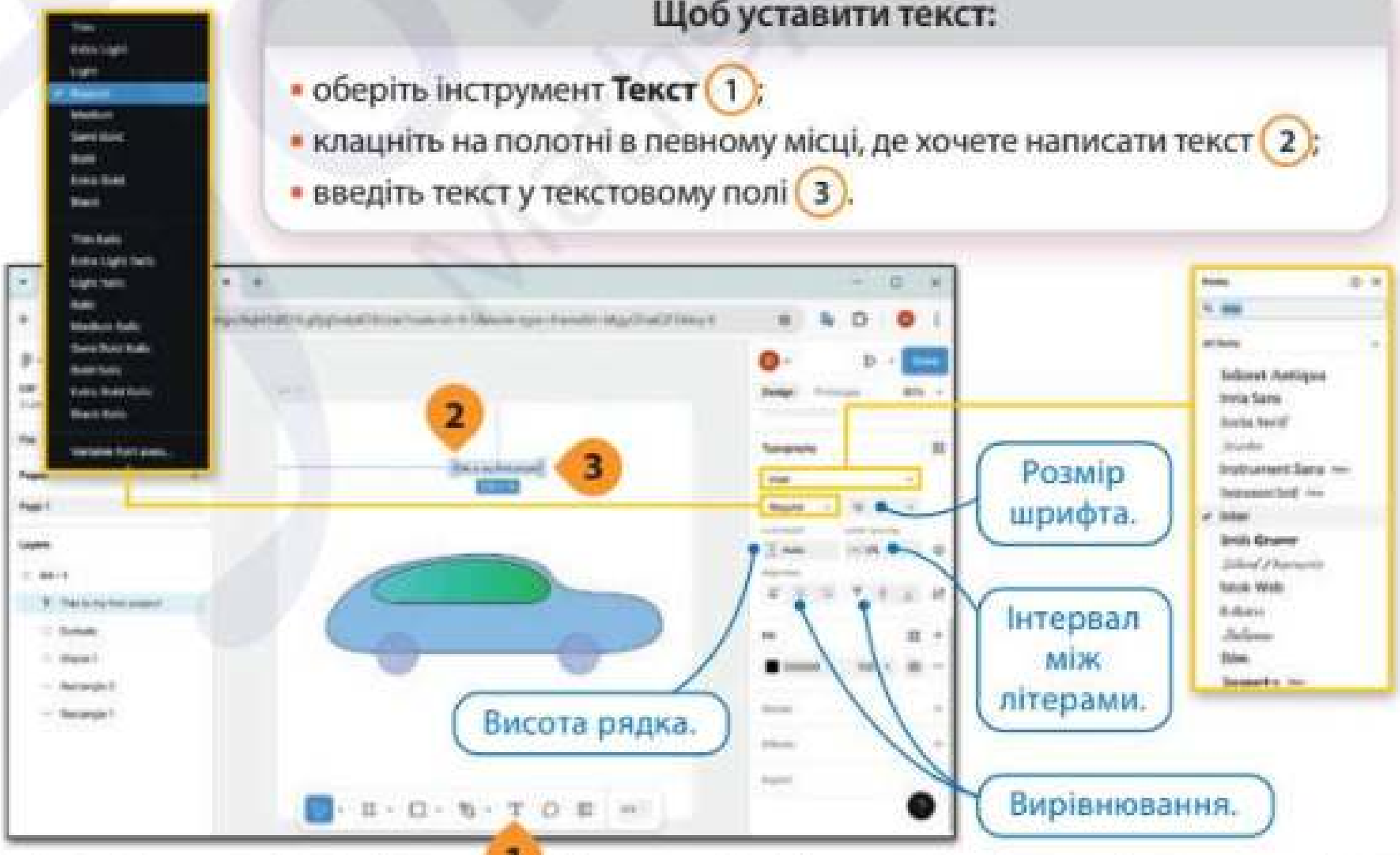
Робота з текстом доволі звична і схожа на роботу в інших графічних редакторах.

Є ще один елемент, який ви не можете контролювати, коли справа доходить до типографіки та настрою. Це те, як ваша авдиторія сприйме ваш контент і варіації шрифтів.

Користувачі підходитимуть до дизайну з різних поглядів. Заздалегідь зверніть до уваги вашу авдиторію та спробуйте передбачити, як вона сприйме дизайн. Що вона подумає про ваш вибір типографіки? Чи можете ви ухвалити рішення, які збігатимуться з тим, чого хочуть або чекають від проєкту?

Щоб уставити текст:

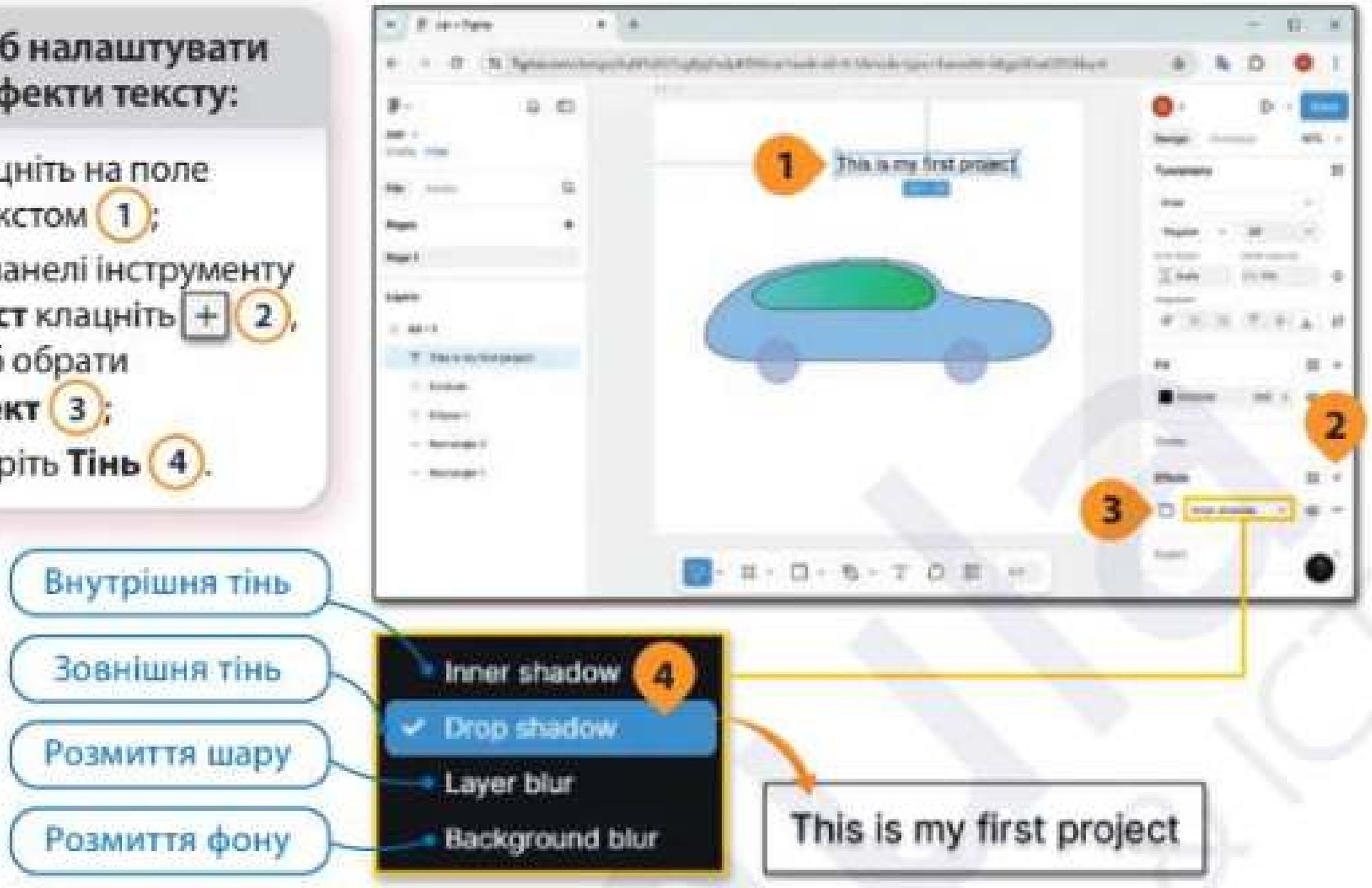
- оберіть інструмент **Текст** (1);
- клацніть на полотні в певному місці, де хочете написати текст (2);
- введіть текст у текстовому полі (3).



© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Щоб налаштувати ефекти тексту:

- клацніть на поле з текстом (1);
- на панелі інструменту **Текст** клацніть + (2), щоб обрати **Ефект** (3);
- оберіть **Тінь** (4).



Кожен ефект має власні налаштування. Ефекти можна додавати, ховати в разі потреби та видаляти.

Щоб налаштувати ефект Тінь:

- на панелі інструменту в розділі **Effects** клацніть (1);
- налаштуйте на панелі **Тінь** (2) розташування, розмиття тощо за власним бажанням.

ДОВІДКА

Гарна типографіка – це якість, яку людина, заходячи на ваш сайт, не помічає, але відчуває.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Ці правила допоможуть зробити читання легким, приємним для очей, зрозумілим та організованим.

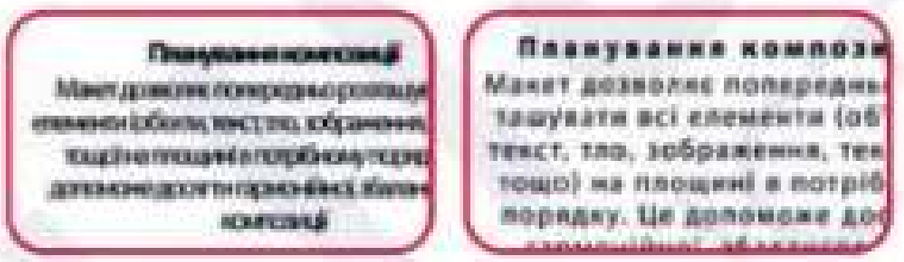
Особливості вибору шрифтів

1 **Використовуйте оптимальну кількість шрифтів**

Краще використовувати 2–3 шрифти, не більше: один – для заголовків, інший – для основного тексту. Забагато різних шрифтів створює хаос, зменшується розуміння тексту та важко засвоюється інформація

2 **Стежте за кернінгом**

Літери не повинні «злипатися» або бути надто далеко одна від одної. Текст має бути розподілений рівномірно



3 **Менше декоративних шрифтів**

Декоративні шрифти мають складні візерунки чи незвичайну форму, їх краще використовувати тільки для коротких заголовків. Для основного тексту обирайте прості, чіткі шрифти

4 **Розмір тексту має значення**

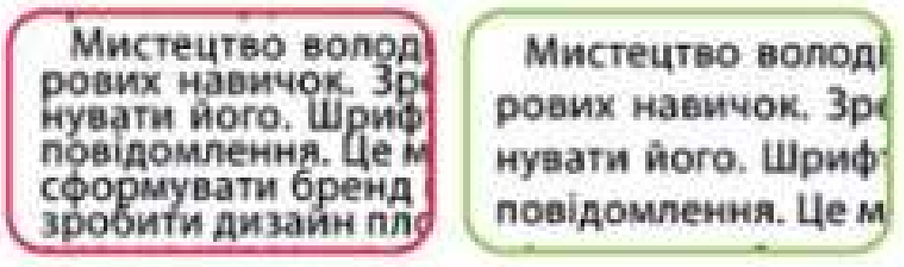
Заголовки завжди більші за основний текст, але основний текст має бути достатньо великим, щоб легко було читати. Підписи можуть бути дрібнішими

5 **Стежте за читабельністю**

Текст повинен легко читатися, літери мають бути чіткі, зрозумілі, а кольори – контрастні

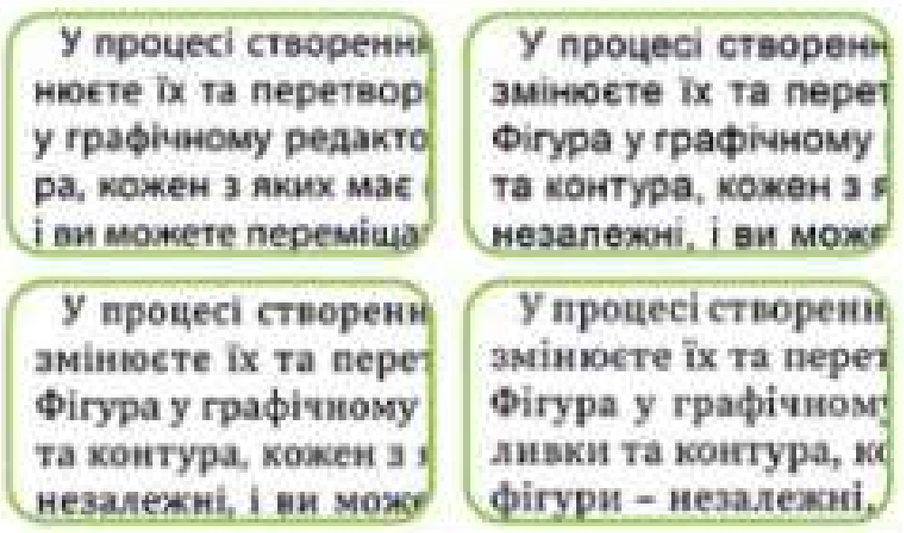
6 **Достатній інтерліньяж**

Рядки не повинні перекривати один одного, зазвичай відстань між ними більша за висоту літер



7 **Настрій шрифту**

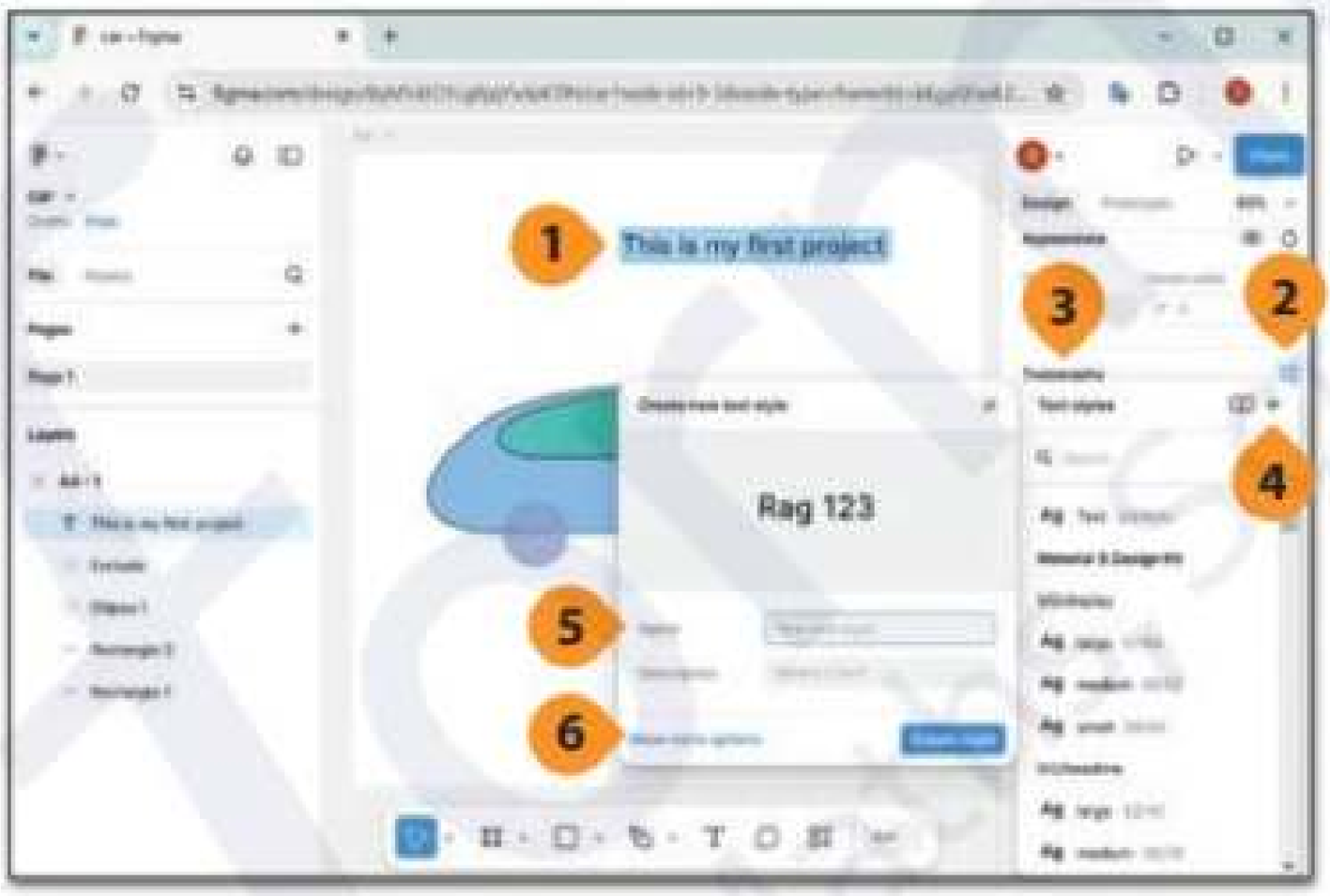
Різні шрифти створюють різний настрій: округлі, м'які шрифти – дружній та неформальний; прямі, геометричні шрифти – серйозний і діловий. Експериментальні шрифти можна використати для креативного проєкту

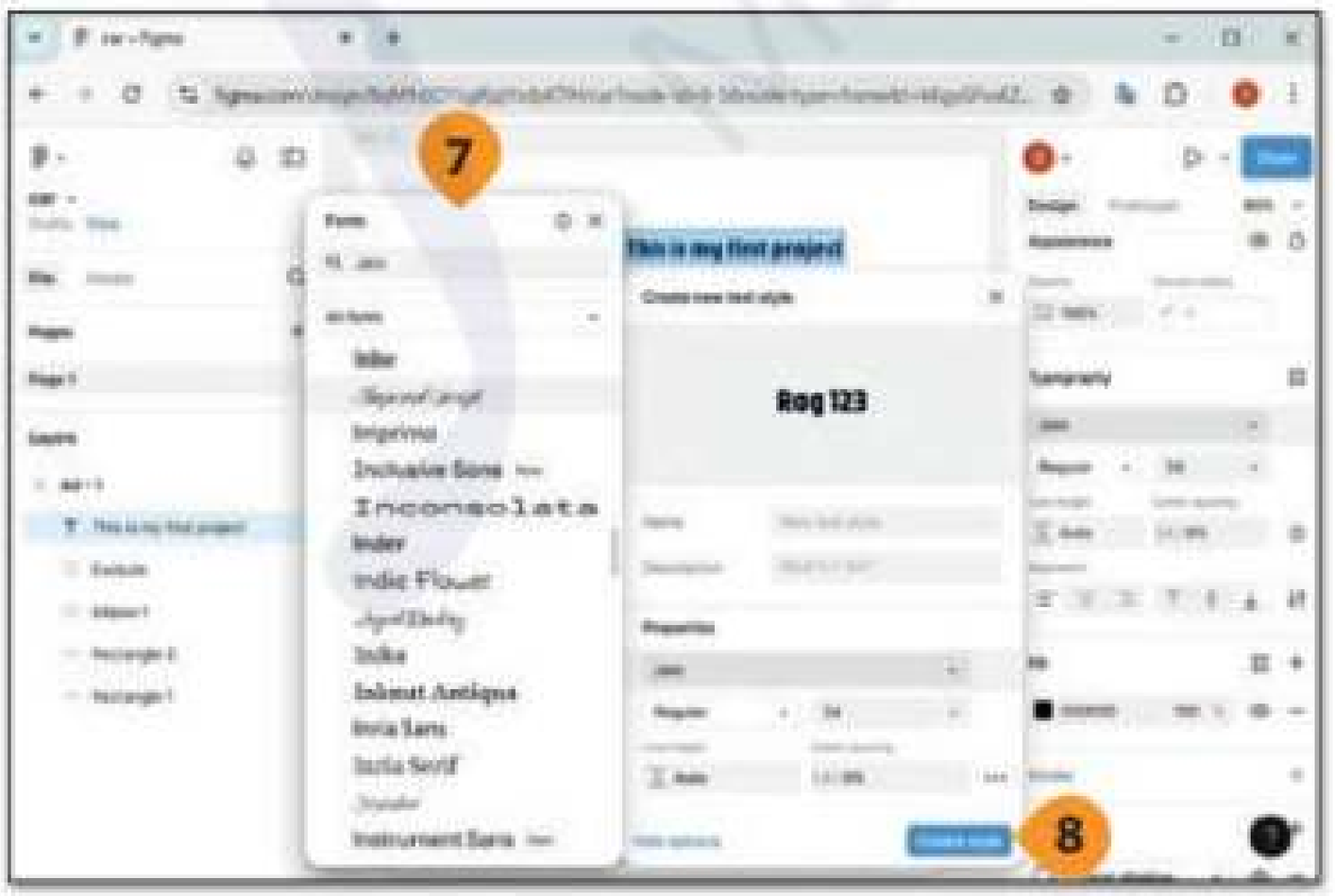


© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

8	**Враховуйте інші елементи дизайну** Шрифт має пасувати до загального дизайну. Узгоджуйте його зі стилем кнопок, піктограм, зображеннями та іншими елементами. Думайте, де проєкт буде використовуватися	• оберіть інструмент **Ellipse** (1); • утримуйте **Shift** ⇧, клацніть лівою
9	**Абзаци важливі** Розділяйте текст на логічні частини, створюйте відступи між абзацами. Перший рядок абзацу можна виділити	**Композиція** – це поєдна проєкті, які втілюють певну меблі в кімнаті – всі елемент дають гарно, гармонійно та Композиція – це мова диза чення, а їх взаємодія розпо
10	**Акценти в тексті** Головні фрази та слова можна виділити жирним, курсив використовуйте для цитат, але не перестарайтеся з виділенням – може загубитися важливе	*Не забудьте вказати в документації всі* **дати якусь дію, треба наступним станом екр проти Interactions кла дії, які будуть можлив**

Створення стилів тексту для шрифтів прискорює процес створення проєктів.





Щоб створити новий стиль для шрифту:

- оберіть поле з текстом (1);
- клацніть ⊞ (2) на панелі інструменту **Typography** (3);
- натисніть + (4) для створення нового стилю шрифту;
- клацніть у полі **Name** (5) та введіть назву стилю;
- клацніть **Show more options** (6);
- зі спадного списку (7) оберіть потрібний шрифт та інші налаштування;
- натисніть **Create style** (8).

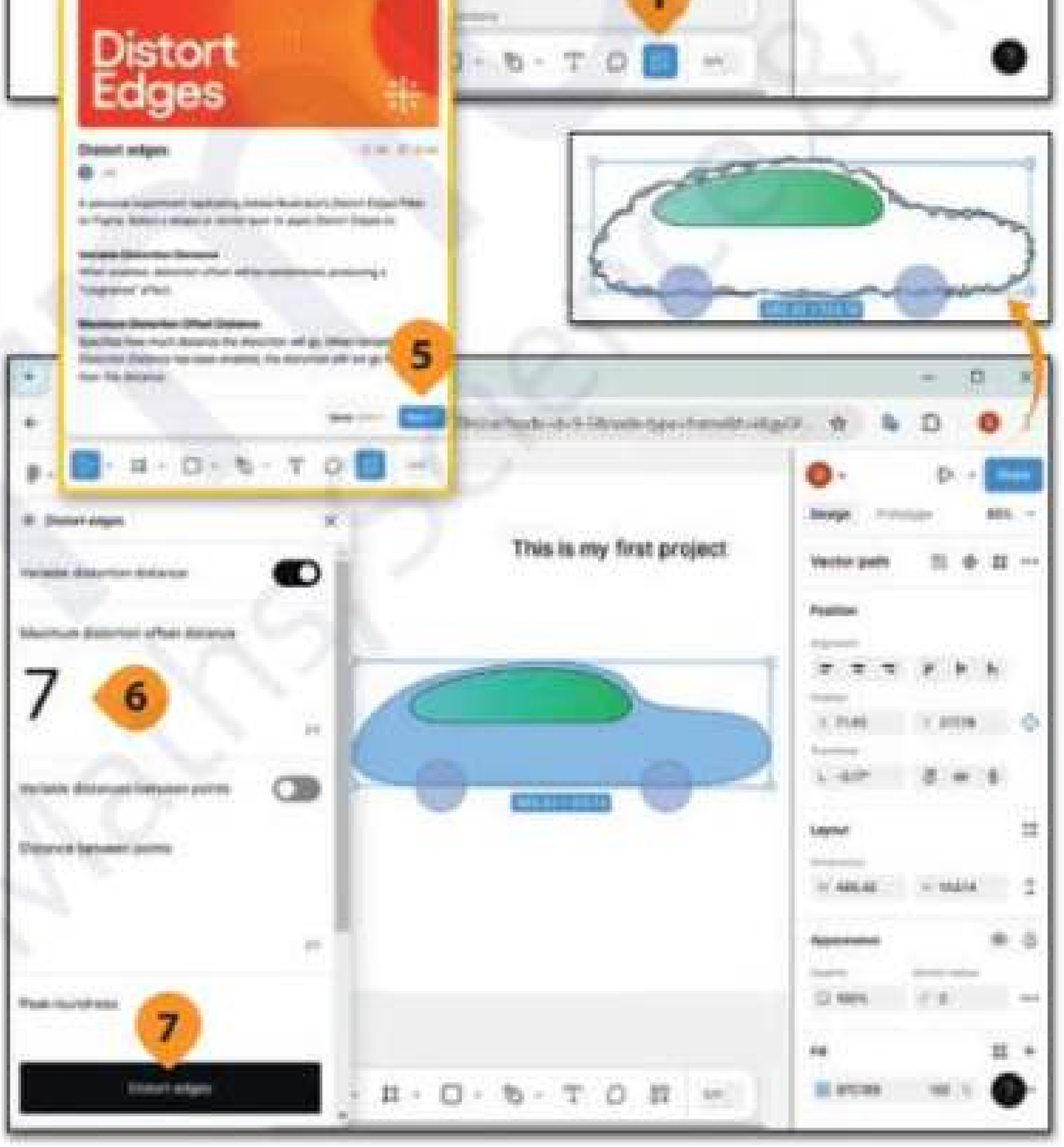
© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Плагіни

Стандартних ефектів у **Figma** небагато. Але є багато плагінів, які розширюють можливості програми.

Щоб використати плагін:

- оберіть об'єкт на вашому проєкті, який бажаєте деформувати;
- натисніть на (1);
- оберіть **Plugins&widgets** (2);
- у рядку пошуку надрукуйте **Distort** (3);
- оберіть **Distort edges** (4);
- клацніть **Run** (5) для запуску плагіна;
- у вигульковому вікні плагіна встановіть відстань зміщення (6);
- натисніть **Distort edges** (7) для запуску плагіна.



Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 5

Композиція та макет

Крім шрифтів і кольору, ще одним складником гарного дизайну є композиція.

Композиція – це поєднання різних елементів у вашому проєкті, які втілюють певну ідею. Уявіть, що ви розставляєте меблі в кімнаті – всі елементи меблів та декору створюють гарний, гармонійний і функціональний інтер'єр.

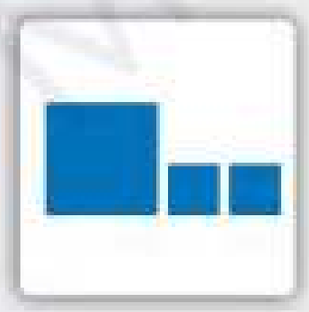
Композиція – це мова дизайну, де кожен елемент має значення, а їх взаємодія розповідає історію. Мета композиції – створити гармонійний, зрозумілий і привабливий графічний твір, який ефективно комунікує з глядачем.

Але, як і всюди, у композиції теж є певні правила.

Основні правила композиції

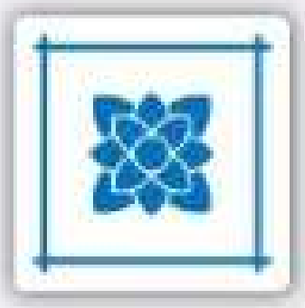
Баланс

Розміщуйте великий об'єкт з одного боку та кілька маленьких з іншого, щоб акцент не зміщався в один бік



Центр уваги

Як і головний герой на сцені, так і головний елемент дизайну, – одразу помітний

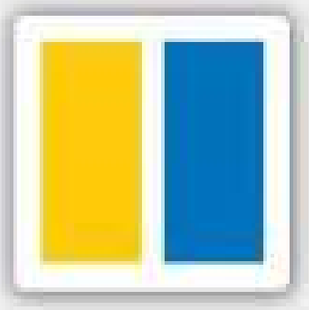


Ритм

Повторення елементів, як музичний біт, вибудовують ритм

Контраст

Поєднання протилежностей: темне і світле, велике і маленьке, гладке і текстуроване



© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Макет

Для того щоб було зрозуміло, який вигляд матиме ваш проєкт (сайт, листівка, афіша, сторінка журналу, книги тощо), фахівці використовують макет. Макет – це попередній ескіз або план того, якою буде кінцева візуалізація проєкту чи зображення. Макет – важливий етап у процесі створення будь-якої візуальної роботи.

Макет може бути намальований олівцем на папері чи створений на цифровому пристрої в графічних редакторах або інших спеціалізованих програмах для макетування.

Чому створювати макет важливо

Планування композиції
Макет дозволяє попередньо розташувати всі елементи (об'єкти, текст, тло, зображення, текстури тощо) на площині в потрібному порядку. Це допоможе досягти гармонійної, збалансованої композиції

Визначення пропорцій
На етапі макетування можна експериментувати з розмірами та пропорціями об'єктів для досягнення оптимальних співвідношень у кінцевій візуалізації

Вибір кольорової схеми
Макет дає можливість підібрати та протестувати різні кольорові рішення, щоб створити гармонійну, виразну колірну палітру

Перевірка концепції
Макет дозволяє оцінити, наскільки вдало реалізується задумана ідея чи концепція, внести корективи та доопрацювати задумане, щоб отримати бажаний результат

Тестування інтерфейсів
Макети дають можливість протестувати інтерфейси та функціональність застосунків. Це допомагає покращити якість і точність проєктів, створювати зручні та інтуїтивні продукти, мінімізуючи помилки

Координація роботи команди
Наявність макета полегшує взаємодію та узгодження дій між учасниками команди: дизайнерами, художниками, аніматорами, верстальниками, програмістами тощо

Класифікація схематичних макетів:

- ескізи (Sketches);
- низькодеталізовані макети (Low-Fidelity Wireframes);
- високодеталізовані макети (High-Fidelity Wireframes).

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Із чого складається сайт? Які елементи є обов'язковими, а без яких можна обійтися?

Анатомія макета сайту

«Шапка», або хедер (Header)
Логотип
Навігаційне меню
Кнопки входу/реєстрації
Пошук

Основний контент (Main)
Центральна область сторінки
Основна інформація
Текст, зображення, відео

Бічна панель (Sidebar)
Додаткова інформація
Категорії
Теги
Бічне меню
Реклама

«Підвал», або футер (Footer)
Контактні дані
Посилання на соціальні мережі
Карта сайту
Авторські права
Юридична інформація

Додаткові елементи
Кнопка «Наверх»
Вигулькові вікна
Змістовні блоки
Модальні вікна
Адаптивне меню

Як правильно побудувати композицію? Які правила та інструменти допоможуть розташувати елементи на сторінці чи екрані?

Типи макетів

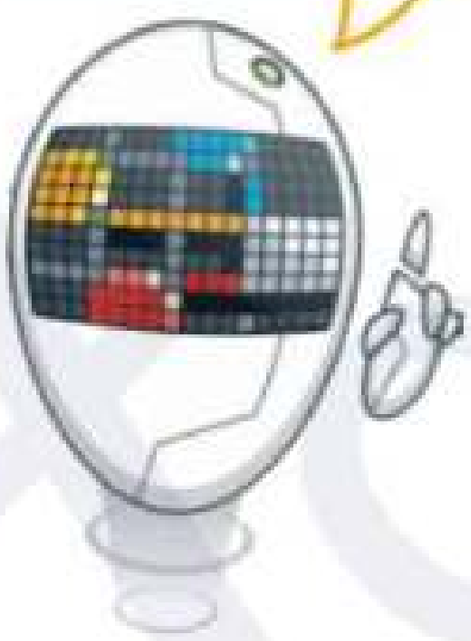
- Макет в одну колонку (як у книжці)
- Макет у дві колонки (як у газеті)
- Розклад карток (як в Pinterest)
- Асиметричний (вільне розміщення)

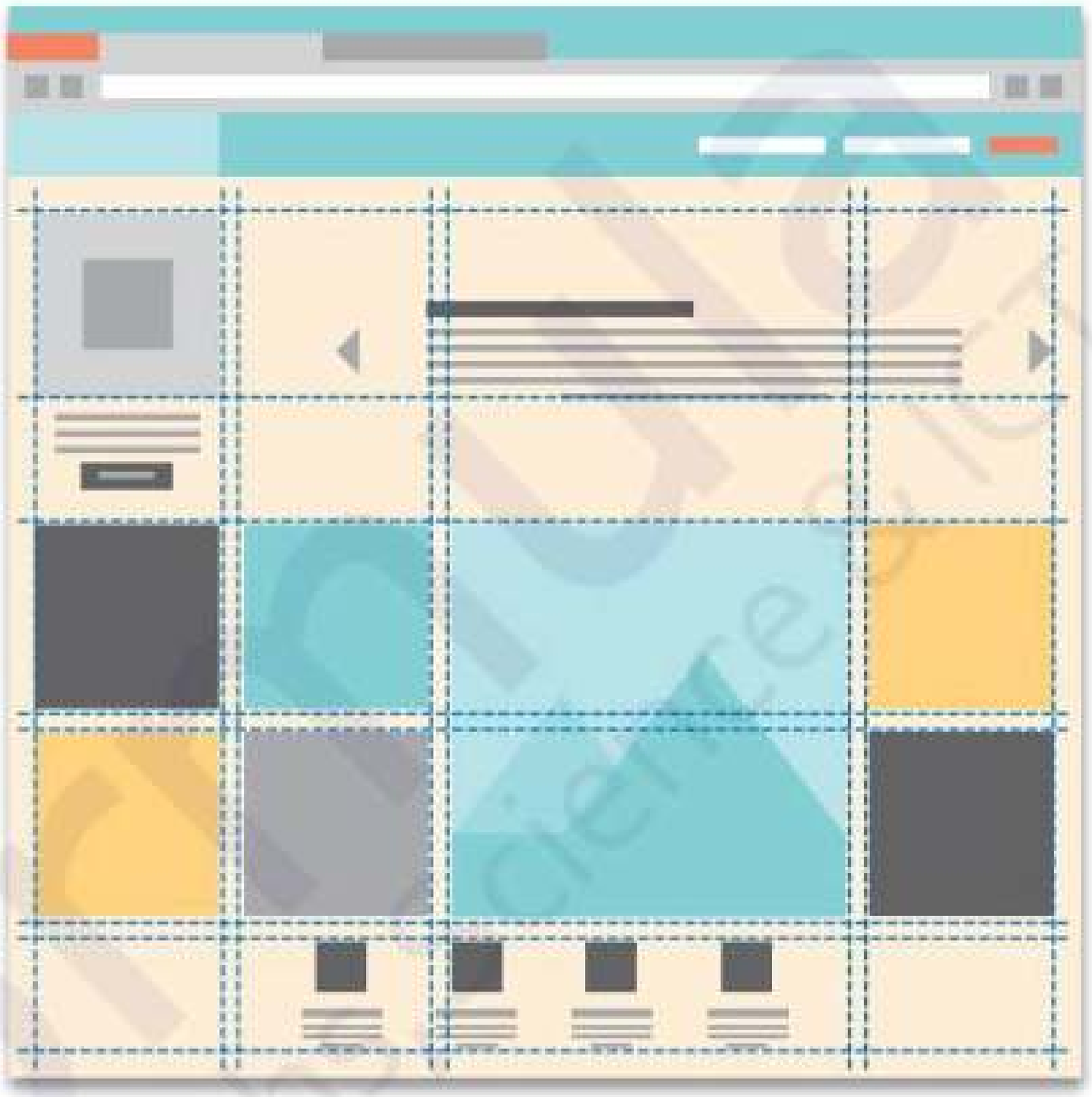
© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Правила розміщення

- Важливе – вгорі або в центрі
- Пов'язані об'єкти – поруч
- Між блоками має бути відступ
- Текст не повинен розміщуватися близько до країв

Сітка (Grid) *допомагає акуратно розмістити текст і картинки. Уявіть, що на аркуші намальовані невидимі лінії, як у зошиті в клітинку, тільки клітинки можуть бути різного розміру.*





Як оцінити власний макет сайту? Чи легко в ньому орієнтуватися?

Перевірте свій макет

- ☑ Чи все видно здалеку?
- ☑ Чи зрозуміло, куди притягується погляд?
- ☑ Чи не тісно елементам?
- ☑ Чи має все разом гармонійний вигляд?

Практичне завдання

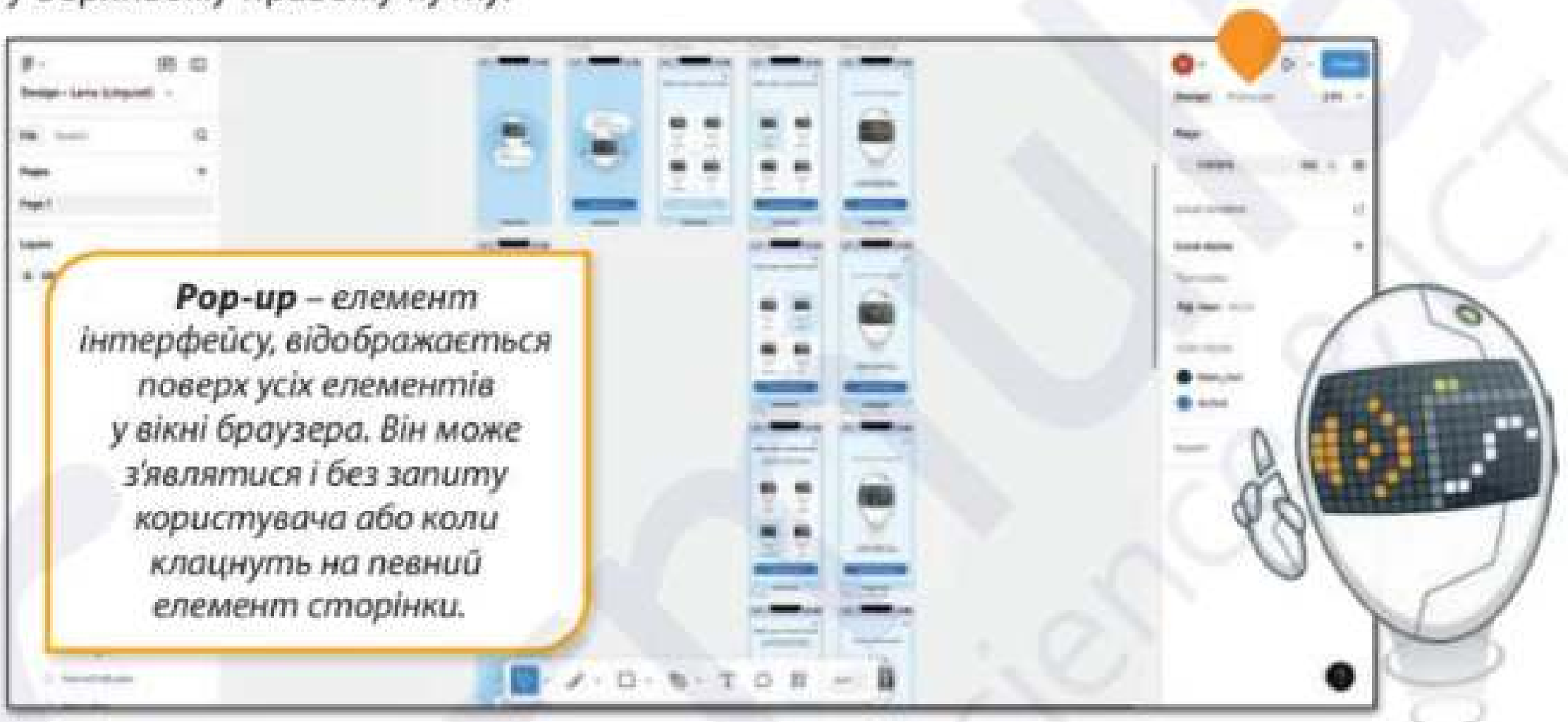
Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 6

Прототип. Документація інформаційного продукту

Прототипування – це створення макета майбутнього інтерфейсу. Після підготовки всіх можливих екранів його анімують, тобто створюють імітацію роботи реального сайту або застосунку за допомогою вкладки **Prototype** у верхньому правому кутку.

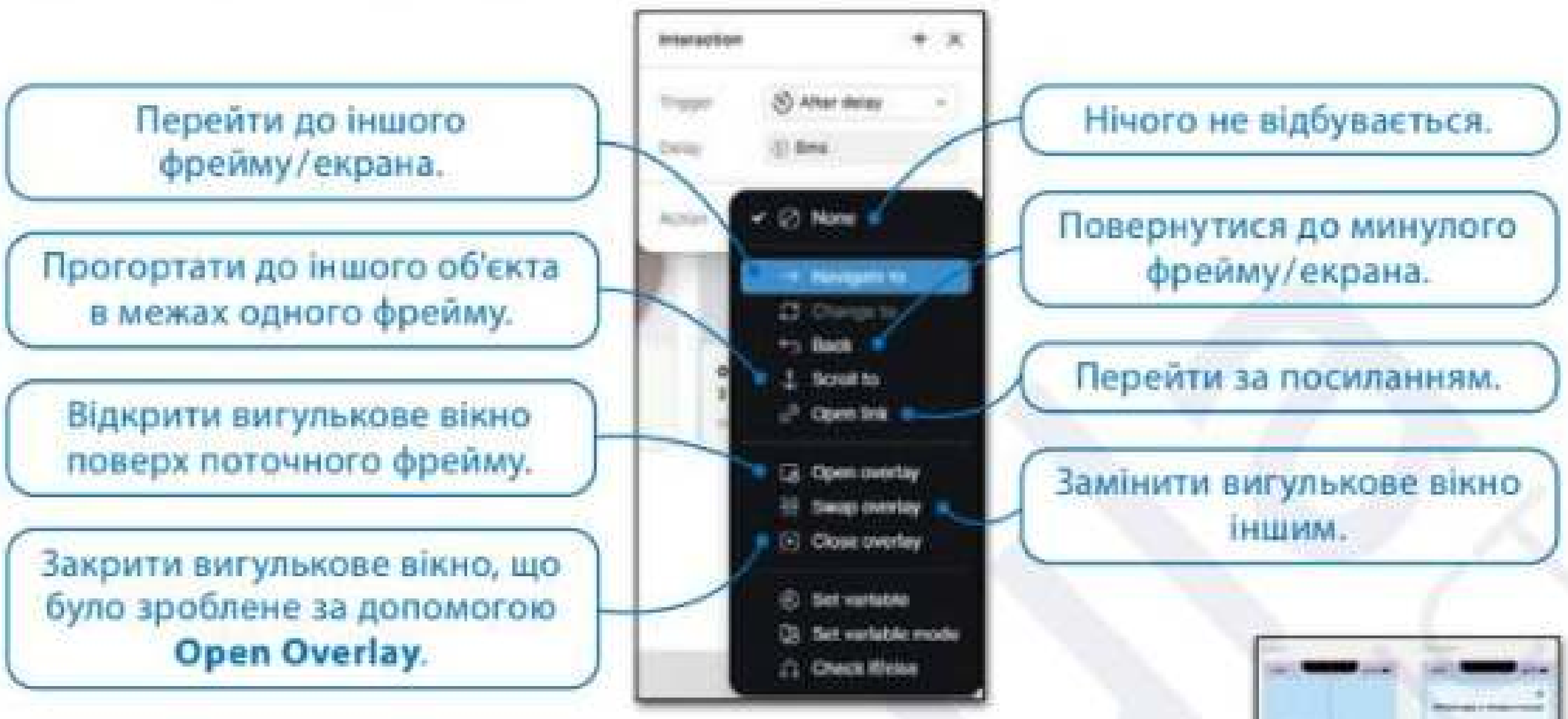


***Pop-up** – елемент інтерфейсу, відображається поверх усіх елементів у вікні браузера. Він може з'являтися і без запиту користувача або коли клацнуть на певний елемент сторінки.*

Щоб додати якусь дію, треба обрати екран чи окремий елемент інтерфейсу й створити зв'язок із наступним екраном. Оберіть потрібний екран, далі в бічній панелі навпроти **Interactions**, клацніть (+), натисніть **Click**, а потім у полі навпроти **Trigger** виберіть потрібний тригер (при якій дії відкриватиметься наступний екран).

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Далі оберіть час чекання, якщо вибрано триггер **After delay**. В інших випадках потрібно вибрати дію (**Action**), яка відбудеться.



Після того як екран із роботом зникне, повинен з'явитися наступний екран із чотирма об'єктами.

Кожен об'єкт матиме власну дію, що відкриватиме наступний стан екрана. Додайте тригер натискання та дію зміни екрана до кожного.

Продовжуйте історію, додавши відповідні тригери та дії до потрібних елементів інтерфейсу.

Щоб передивитися інтерактивний інтерфейс, натисніть **Flow** біля першого екрана інтерфейсу.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Документація інформаційного продукту

Документація інформаційного продукту – це структурований набір текстових та графічних матеріалів, що повністю описують проєкт та все, що з ними пов'язано. Основне завдання документації – максимально зрозуміло та зручно описати проєкт для його подальшого розвитку.

Технічна документація

Специфікації продукту
Вимоги до системи
Інструкції зі встановлення
Архітектура системи
API-документація
Схеми баз даних

Користувацька документація

Посібник користувача
FAQ (часті запитання)
Інструкції з використання
Навчальні матеріали
Відеоуроки
Довідка онлайн

Обов'язково вкажіть в документації всі використані джерела, ресурси, ліцензії.

Маркетингова документація

Опис продукту
Презентації
Брошури
Технічні характеристики
Порівняльні таблиці
Кейси використання

Юридична документація

Ліцензійна угода
Умови використання
Політика конфіденційності
Гарантійні зобов'язання
Сертифікати

Документація з підтримки

Інструкції з усунення несправностей
База знань
Процедури оновлення
Контакти служби підтримки

Безпека

Політики безпеки
Процедури резервного копіювання
Протоколи відновлення
Рекомендації з безпеки

Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

https://formula.education/online-ict-8-klas/

Інші платформи

Проєкт

Підсумки

Глосарій

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

5. Сайт і його верстка

У цьому розділі ви дізнаєтеся, як створювати вебсайт. Визначите, які інструменти знадобляться, вивчите структуру HTML-коду. Зрозумієте, як урізноманітнити вебсторінки за допомогою контенту, як-от зображення, малюнки, відео тощо. Дізнаєтеся як отримати зворотній зв'язок та слідкувати за версіями продукту, які технології допомагають будувати роботу команди.

У цьому розділі ви навчитеся:

- створювати вебсайт за допомогою HTML;
- вставляти та форматувати текст на вебсторінці;
- створювати гіперпосилання;
- додавати зображення, відео та звуки на сайт;
- використовувати стилі CSS для оформлення сайту;
- створювати вебформи;
- використовувати інструменти для координування роботи команди;
- використовувати інструменти для контролю версій програми.

У цьому розділі ви дізнаєтеся:

- що таке вебсайт;
- як зверстати просту вебсторінку;
- як переходити з однієї вебсторінки на іншу;
- як опублікувати свій сайт в інтернеті;
- як розмістити на вебсторінці мультимедійні файли;
- як застосовувати бібліотеки CSS;
- як додавати меню на вебсайт;
- як отримати зворотній зв'язок;
- як планувати роботу команди;
- як контролювати версії програм.

Програмне забезпечення:

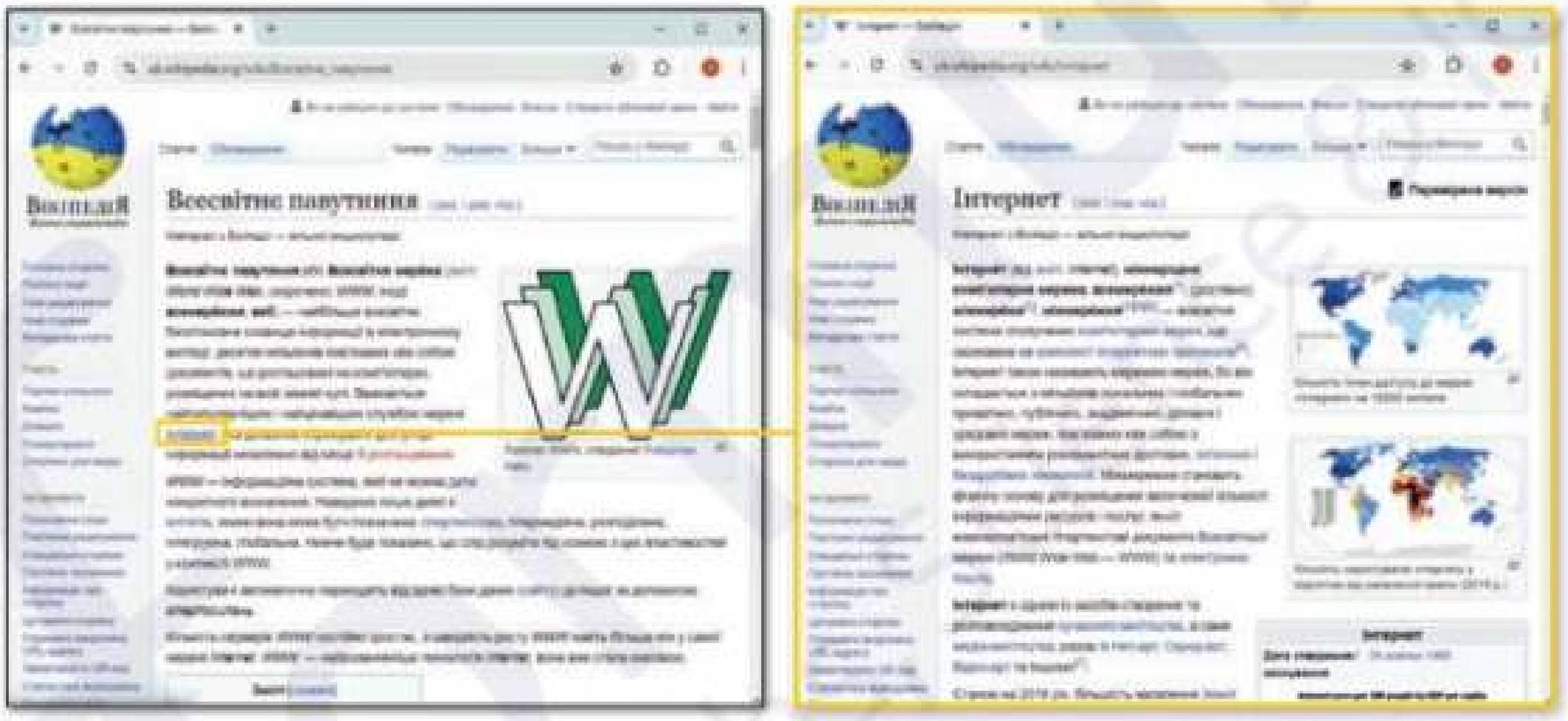
- Visual Studio Code;
- Eclipse;
- CoffeeCup;
- Notepad++;
- Sublime Text;
- Nova;
- Adobe Dreamweaver.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 1
Створення вебсторінки

Переглядаючи сайт, по суті, ви бачите серію вебсторінок. **Вебсторінка** – це структурований документ, що містить текст, зображення та інший мультимедійний контент, який ви можете переглядати за допомогою вебпереглядача. Вебсторінка відрізняється від простого мультимедійного документа, що зберігається на вашому комп'ютері, тим, що вона містить гіпертекст, або посилання, які забезпечують можливість переходу на інші вебсторінки.

Візьмемо, сторінку у Вікіпедії. Вона містить різні посилання на інші статті. Натиснувши на одне з них, браузер показує нам ту вебсторінку, на яку вказує це посилання.



Браузер дає можливість переходити на інші вебсторінки, клацнувши на посилання.

Набір пов'язаних між собою вебсторінок, які можна знайти під одним доменним ім'ям, наприклад, **wikipedia.org**, утворюють **вебсайт**. Він зазвичай має початкову сторінку, яку називають **домашньою**. З неї відвідувач може перейти на інші сторінки.

Вебсторінка може містити посилання на сторінки що розташовані на тому ж самому сайті, то й на сторінки інших сайтів.

ДОВІДКА

Доменне ім'я (наприклад, google.com) – це ім'я-ідентифікатор, яке можна придбати у реєстрі доменних імен. За кожне доменне ім'я сплачується щорічна плата, щоб зберігати його доки є потреба.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

HTML – мова вебсторінок

Вебсторінки створюють за допомогою **мови розмітки гіпертексту** (**HTML**). Вона визначає структуру сторінки й те, як інформація буде відображатися у браузері.

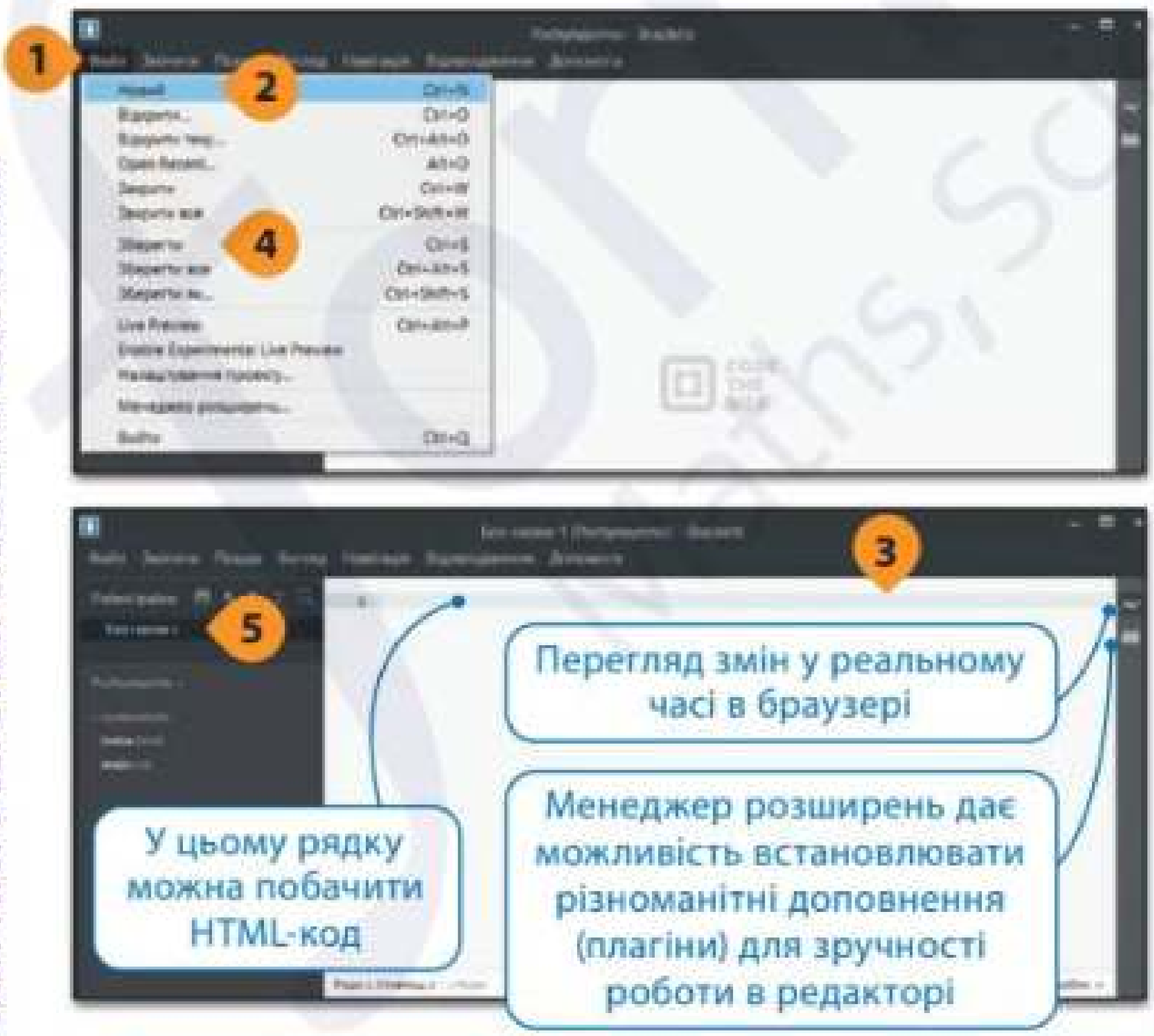
HTML використовує теги для позначення різних елементів на сторінці, наприклад, заголовки, абзаци, зображення, посилання тощо.

Теги – спеціальні позначки, укладені у кутові дужки (наприклад, <html>), в межах вмісту вебсторінки. Теги HTML найчастіше зустрічаються в парах, наприклад, **<h1>** (відкриваючий тег) і **</h1>** (закриваючий тег), хоча деякі теги, можуть бути непарними, наприклад, **<img>**. Між цими тегами можна додавати текст, інші теги, коментарі та інші типи текстового контенту.

Вебпереглядач здатний інтерпретувати HTML-код вебсторінки та точно відображати його.

Створення вебсторінки

Все, що вам потрібно для написання власної вебсторінки – це текстовий редактор, наприклад, **Блокнот**, і заміна розширення створеного файлу на **.html**. Однак набагато простіше та швидше використовувати HTML-редактор, наприклад, **Sublime Text** або **Eclipse** тощо. Скористаємося текстовим редактором від компанії Adobe **Brackets** для створення простої вебсторінки.



Плагін *(англ. plug-in) – програмний компонент, який додає специфічні функції до основної програми або розширює її можливості.*

HTML – не мова програмування, а мова стилів.

Щоб створити вебсторінку:

- на початковій сторінці **Brackets** клацніть **Файл** (1) та оберіть **Новий** (2);
- у редакторі відкриється новий порожній файл (3);
- збережіть файл (4) під назвою **index.html** (5). Ця назва відображатиметься у вікні браузера, коли ви відкриєте в ньому цю вебсторінку.

© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

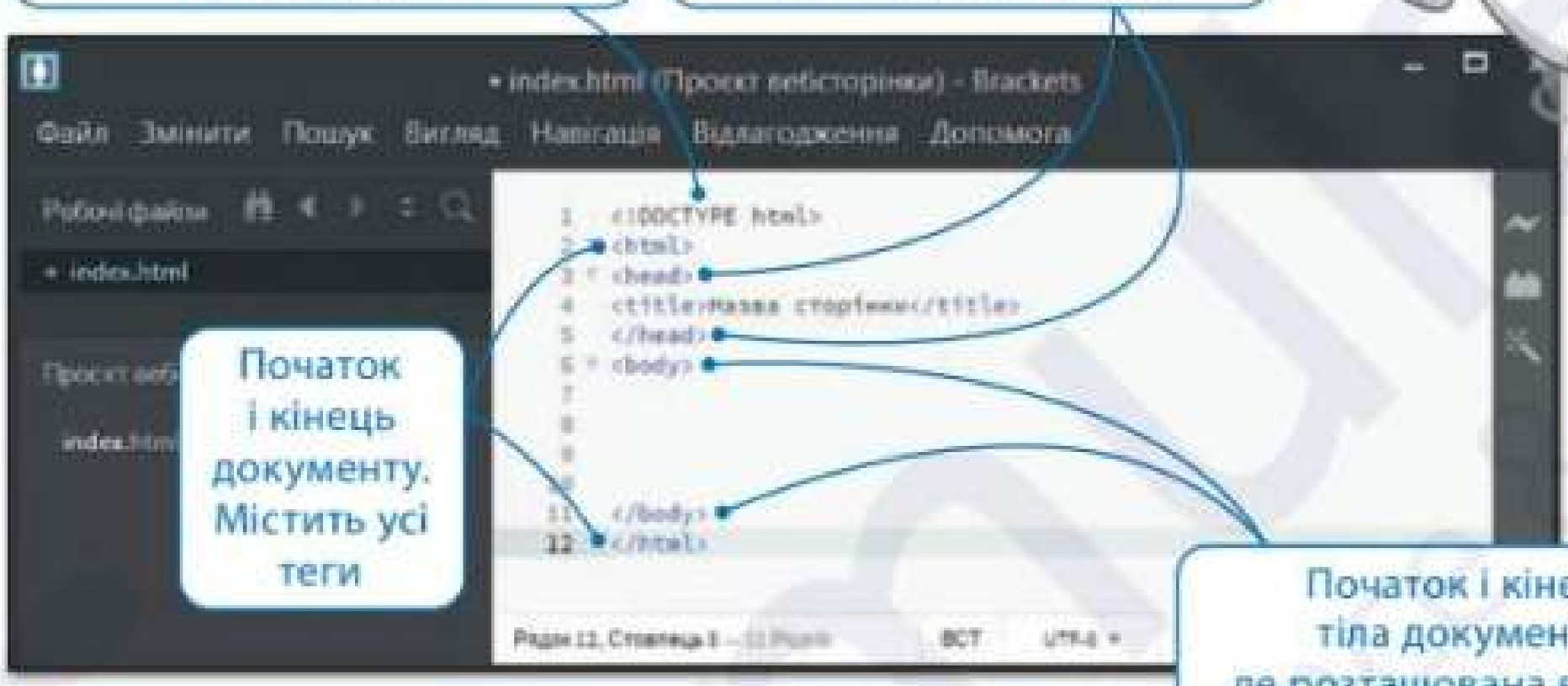
Структура HTML-документа

Вже було сказано, HTML-документ – простий текстовий документ, який можна створити у будь-якому текстовому редакторі. Але він має певну структуру та містить обов'язковий набір тегів.

*Використовуйте **HTML5** – найсвіжішу версію HTML.*

Вказує тип документу, інформація для браузера, як потрібно інтерпретувати вебсторінку

Початок та кінець тега, який зберігає службову інформацію для браузерів і пошукових систем

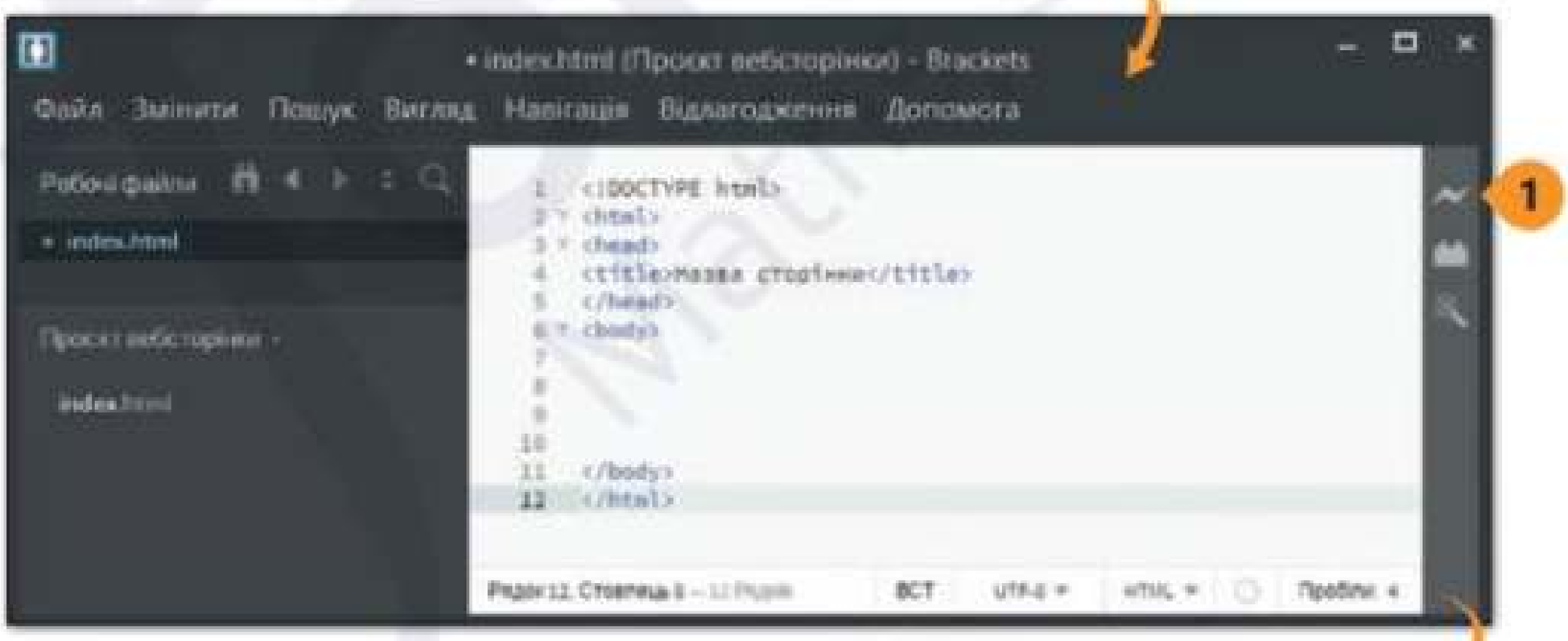


Початок і кінець документу. Містить усі теги

Початок і кінець тіла документу, де розташована видима частина вебсторінки (текст, таблиці, зображення тощо)

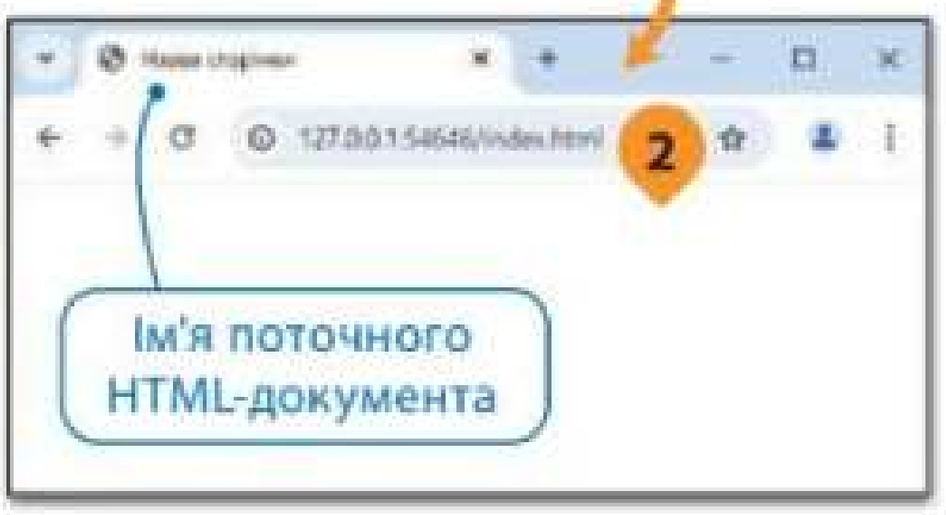
*Вміст **<head>** не відображається на вебсторінці, за винятком тега **<title>**. Він відображає у заголовку браузера ім'я поточного HTML-документа.*

Як буде виглядати вебсторінка за наведеним кодом?



Щоб переглянути вебсторінку у браузері:

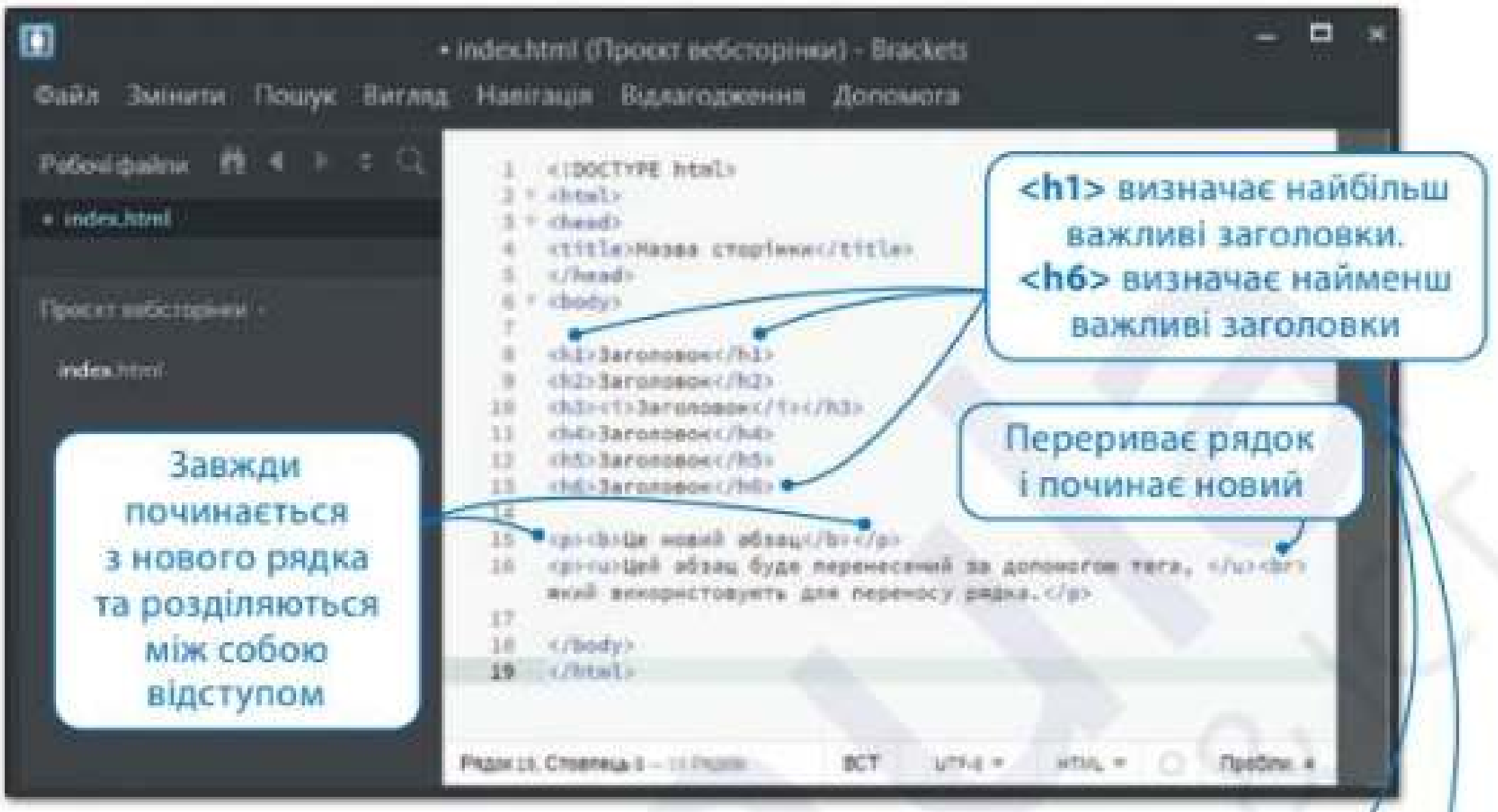
- клацніть **Перегляд у реальному часі** (1);
- у вікні браузера, що відкриється, буде ваша вебсторінка (2).



© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Основні елементи розмітки тексту

Додайте вміст на вебсторінку.

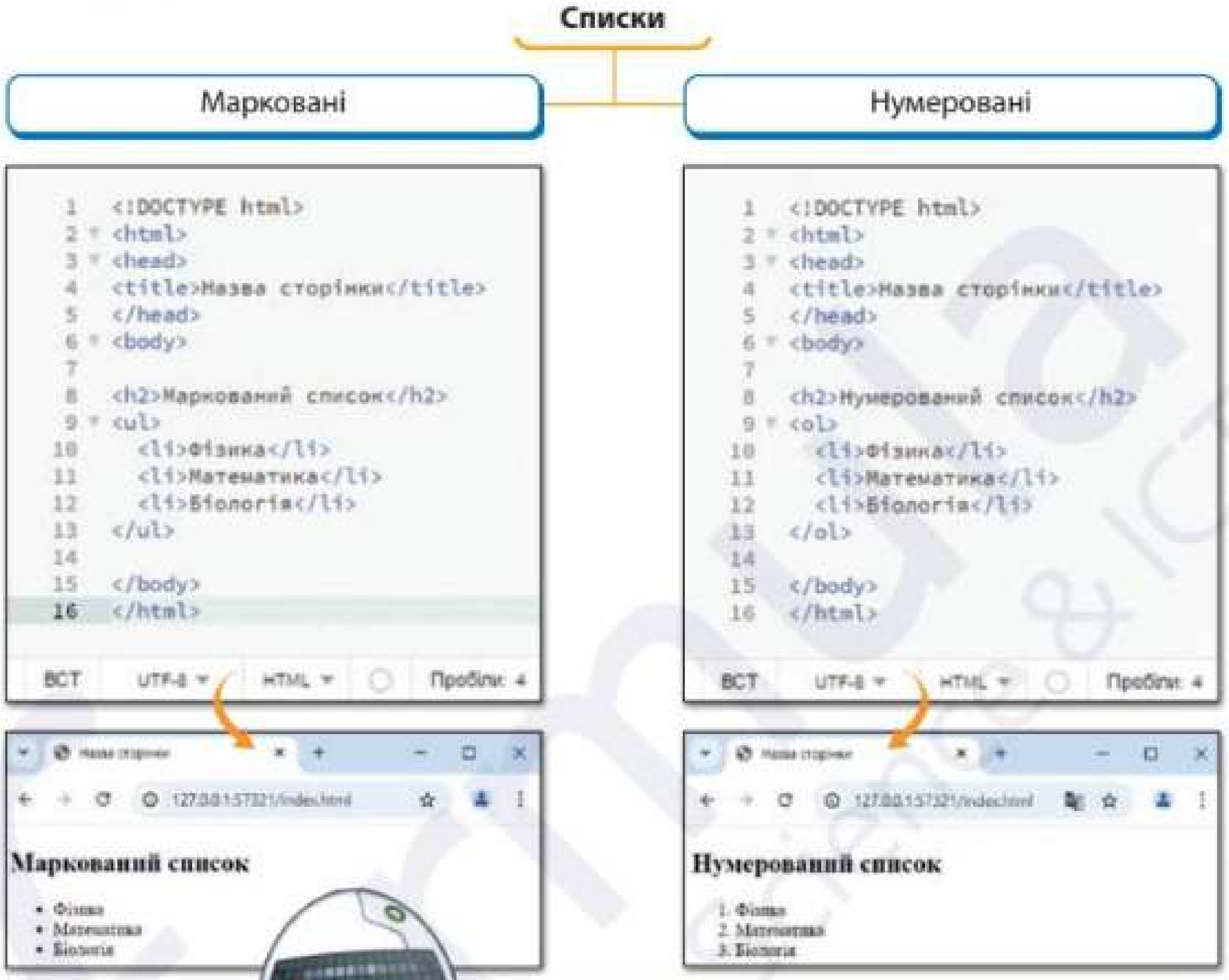


Теги, що вказані з помилками, або такі, що не підтримує браузер, будуть проігноровані.

Тег	Значення тегу
... ...	Визначає заголовки, <h1> визначає найбільший заголовок, <h6> – найменший заголовок
...	Початок та кінець абзацу з відступом
...	Текст стає курсивом
...	Текст стає жирним
<u> ... </u>	Текст стає підкресленим
 	Переривається рядок і починається новий
<!- - ... - ->	Коментарі, текст, розміщений між тегами не відображається на вебсторінці

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Щоб контент легко сприймався, використовують різні структури: списки, таблиці тощо.



Тег	Значення тегу
...	Маркований список
...	Нумерований список
...	Елемент списку

Стилі маркованого списку

Тег	Значення тегу
<ul type = "disc"> ...	Список з маркерами у вигляді •
<ul type = "circle"> ...	Список з маркерами у вигляді ◦
<ul type = "square"> ...	Список з маркерами у вигляді ▪

HTML Атрибути

Атрибути надають додаткову інформацію про HTML-елементи та дозволяють модифікувати їх зовнішній вигляд або поведінку.

Для маркованого списку за замовчуванням використовують крапки.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Без уточнення атрибуту в нумерованому списку застосовується список з арабськими числами.

Значення атрибута зазвичай беруть в одинарні чи подвійні лапки.

Стилі нумерованого списку

Тег	Значення тегу

`<ol type = "1">`
`  <li> ... </li>`
`</ol>`
	Список з арабськими числами (1, 2, 3, …)

`<ol type = "A">`
`  <li> ... </li>`
`</ol>`
	Список з великими буквами латинського алфавіту (A, B, C, …)

`<ol type = "a">`
`  <li> ... </li>`
`</ol>`
	Список з малими літерами латинського алфавіту (a, b, c, …)

`<ol type = "I">`
`  <li> ... </li>`
`</ol>`
	Список з римськими числами в верхньому регістрі (I, II, III, …)

`<ol type = "i">`
`  <li> ... </li>`
`</ol>`
	Список з римськими числами в нижньому регістрі (i, ii, iii, …)

Зверніть увагу, що теги можуть мати більш ніж один атрибут.

<tag attribute1="value1" attribute2="value2" attribute3="value3">content</tag>

tag	Початок тегу
attribute1, attribute2, attribute3	Назва атрибута
"value1", "value2", "value3"	Значення атрибуту
content	Вміст
</tag>	Кінець тегу

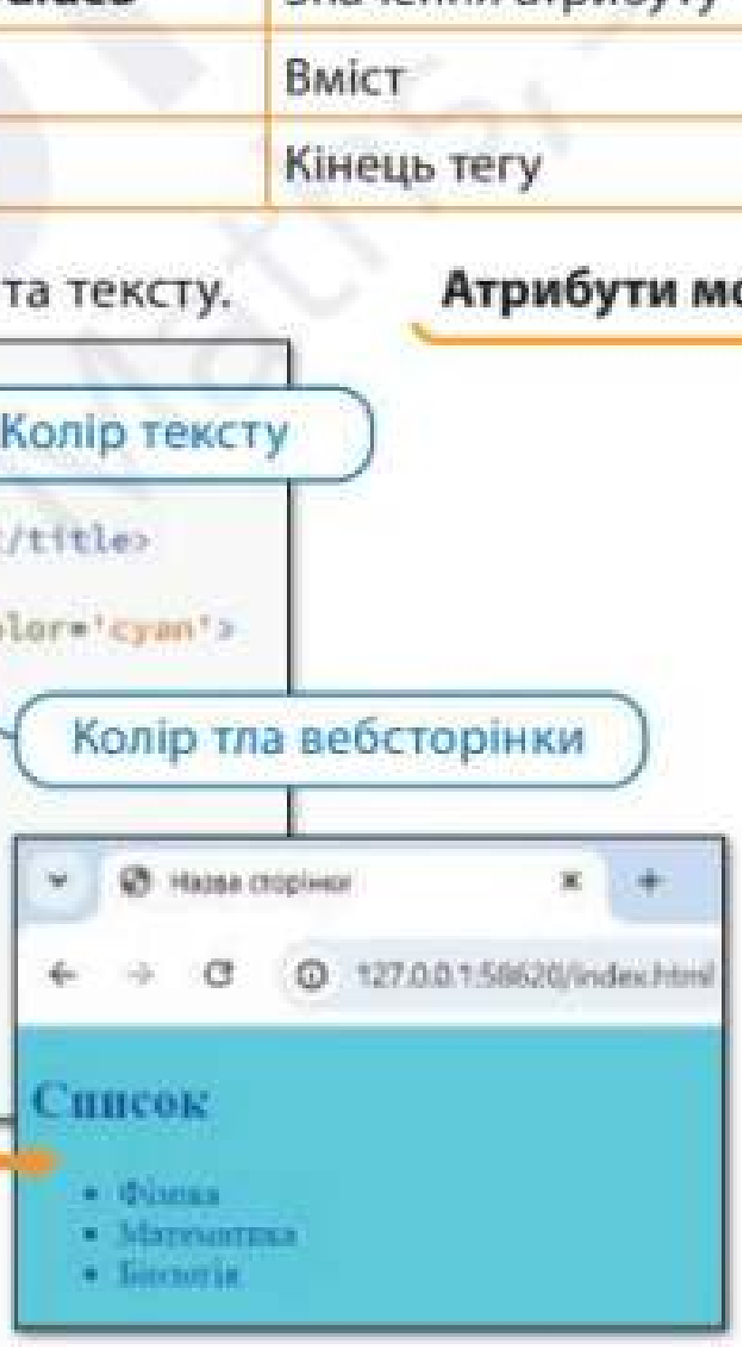
Атрибути між собою розділяють Space

Змінимо колір сторінки та тексту.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Назва сторінки</title>
5  </head>
6  <body text='blue' bgcolor='cyan'>
7
8  <h2>Список</h2>
9  <ul type='square'>
10   <li>Фізика</li>
11   <li>Математика</li>
12   <li>Біологія</li>
13 </ul>
14
15 </body>
16 </html>
```

Колір тексту

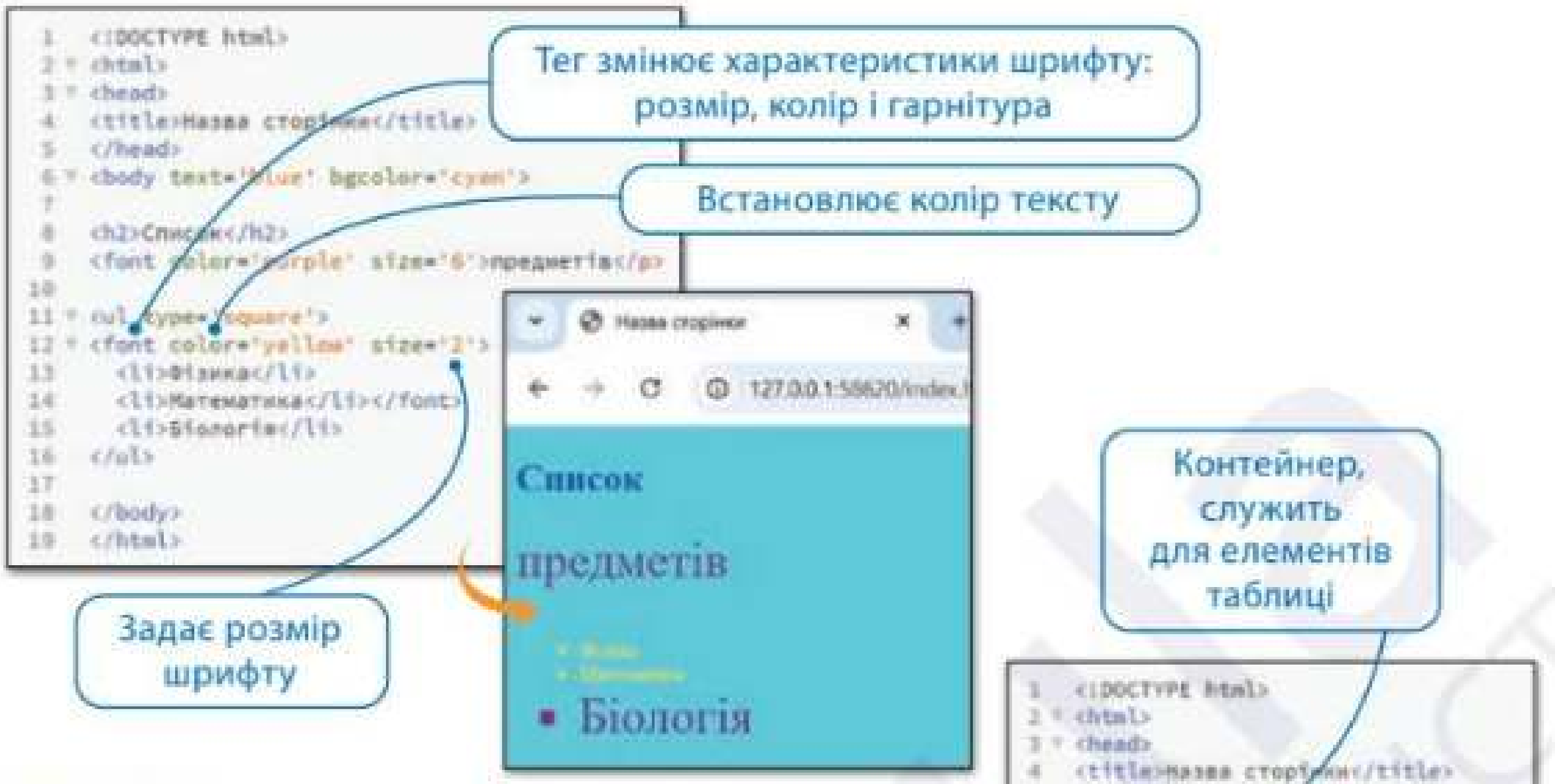
Колір тла вебсторінки



Атрибути можуть мати одне зі значень

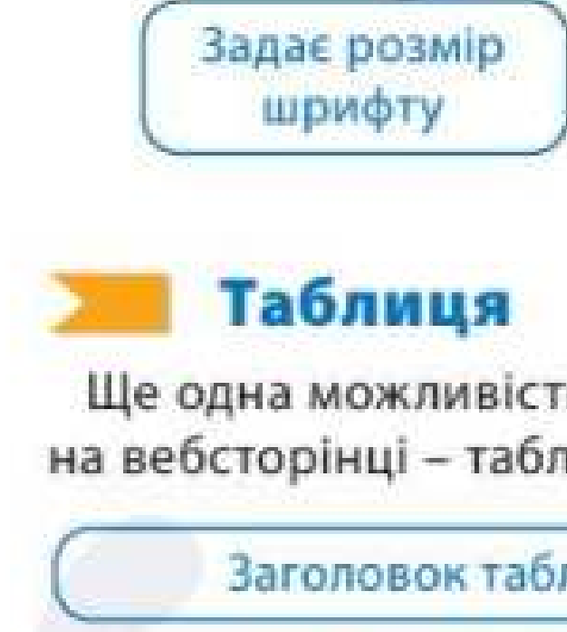
- назва кольору, наприклад, "red"
- колір тексту за HEX-кодом, як-от, "#ff0000"
- колір тексту за RGB-кодом, "RGB (255.0.0)"

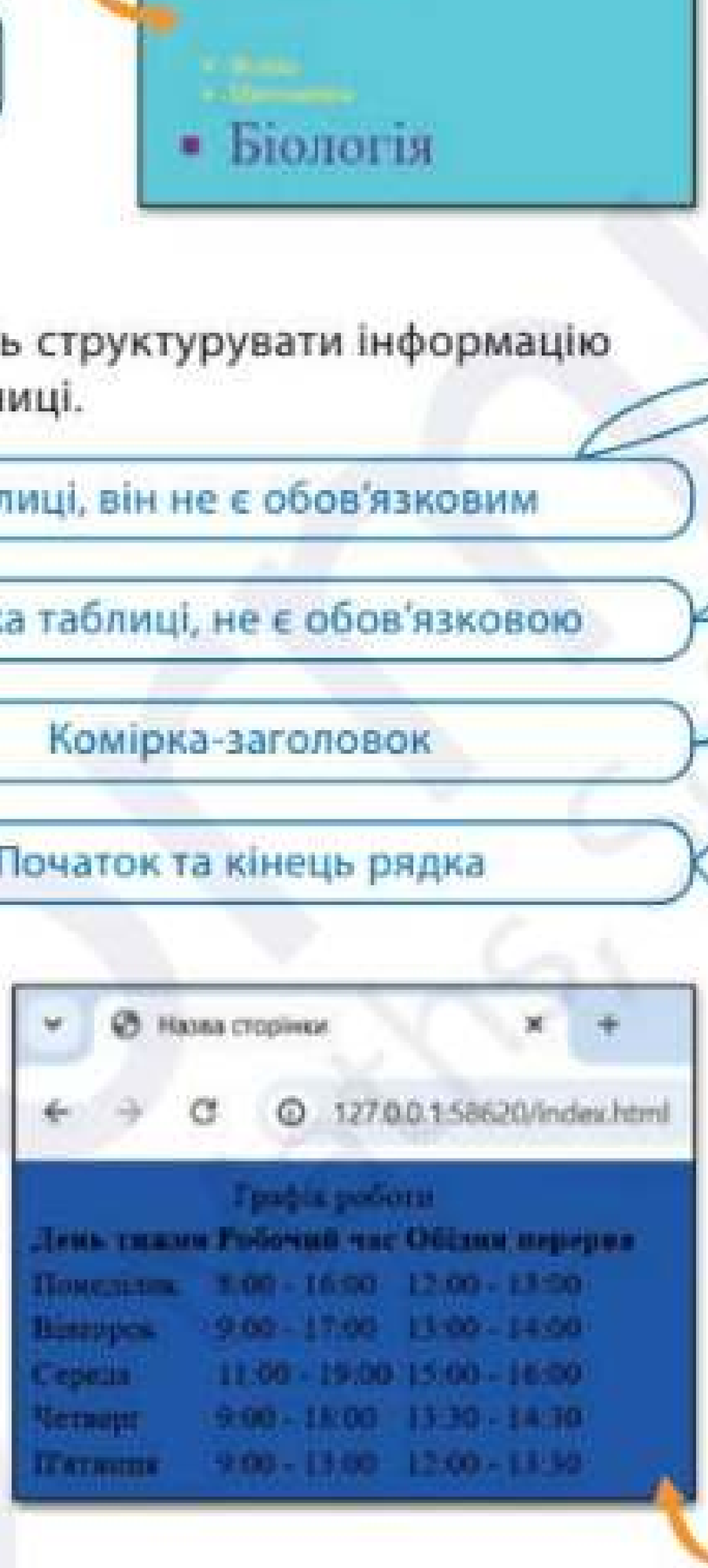
© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.



Таблиця

Ще одна можливість структурувати інформацію на вебсторінці – таблиці.





Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 2
HTML та CSS

Як і всі технології, HTML постійно розвивається. Поточна версія є HTML5. Вона запроваджує чимало нових функцій, підтримуючи відтворення відео та аудіо, а також 2D та 3D графіки. Використання HTML5 дозволяє створювати більш легкі вебсторінки, які завантажуються швидше та менше навантажують вебсервери. Багато функцій HTML5 були розроблені з можливістю їх запуску на малопотужних пристроях, як-от смартфони та планшети.

Додаємо відео

За допомогою HTML5 додавати відео на ваші вебсторінки легко. Подивимося, як це зробити.

```
<!DOCTYPE html>
<html>
<head>
<title>Моя перша вебсторінка</title>
</head>
<body>

<h1>Це особиста сторінка Оленки</h1>
<p>Ласкаво просимо на мою особисту вебсторінку!</p>
<p>Ось моє коло інтересів:</p>
<ul>
  <li>Фото</li>
  <li>Арт</li>
  <li>Комп'ютери</li>
</ul>

    <h3>Моє відео</h3>

    <!-- Локальне відео -->
    <video width="640" height="360" controls>
        <source src="design T-shot.mp4" type="video/mp4">
    </video>

    <!-- Відео з YouTube -->

    <iframe
      width="560"
      height="315"
      src="https://www.youtube.com/embed/KzAIBGPsA0Iyl-46bCOEyIlwtDSTTA" title="YouTube video player" frameborder="0"
      allowfullscreen>
    </iframe>

</video>
</body>
</html>
</body>
</html>
```

Вказує ширину та висоту вікна відео

Якщо відеофайл розташований в тій самій теці, що й вебсторінка, просто напишіть ім'я файлу. Якщо відеофайл лежить в іншій теці – вкажіть шлях до цього файлу. Наприклад, **<source src="F:\LEGO\Дизайн.mp4" type="video/mp4">**.

Якщо ваш відеофайл розміщений на сайті в інтернеті, відтворити його можна за допомогою такого коду.

Атрибути тега ***<video>****:*

autoplay *– відео починає відтворюватися автоматично;*

loop *– відео відтворюється циклічно;*

muted *– відео відтворюється без звуку.*

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

CSS

Каскадні таблиці стилів (**CSS**) – це мова, яка дозволяє змінювати зовнішній вигляд вебсторінок. CSS співпрацює з HTML. За допомогою CSS можна змінювати колір, розмір, розташування візуальних елементів, додавати відео та анімацію, перетворюючи звичайний текстовий контент у привабливі та функціональні вебсторінки.

Теги **<style type=«text/css»> ... </style>** оточують елементи CSS, які застосовуються до вебсторінки, усередині тегів **<head>** ... **</head>** HTML-коду. Загальний синтаксис правил CSS наступний:

Перша частина коду – **селектор**. Він є структурою, яку використовують для визначення вигляду певного елемента (або набору елементів) у браузері.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Моя перша вебсторінка</title>
5  <style type='text/css'>
6      body {
7          background-color: #6FF;
8      }
9      </style>
10 </head>
11 <body>
12
13 <h1>Це особиста сторінка Оленки</h1>
```

Звичайно, правила CSS можуть містити багато пар властивостей та їх значень. Кожна з них виконує певне завдання в структурі, щоб надати сторінці потрібний вам вигляд.

Властивості CSS-коду визначають конкретну характеристику елементу, наприклад, властивість background-color.

Значення властивості задає параметр вигляду. Наприклад, колір фону.

Додамо новий CSS-код. Застосуємо до тегу **<p>**, що визначає зовнішній вигляд усіх абзаців на нашій вебсторінці.

```
5  <style type='text/css'>
6      body {
7          background-color: #6FF;
8      }
9      p {
10         background-color: #3CF;
11         padding: 5px;
12         color: #FFFFFF;
13         width: 300px;
14     }
15 </style>
```

Кожен елемент HTML міститься в невидимому прямокутнику. Змінивши **background-color** абзацу, цей контейнер забарвлюється, й прямокутник можна побачити. Властивість **color** змінює колір тексту елемента. **Padding** – це відстань від вмісту елемента до внутрішніх стінок прямокутника.

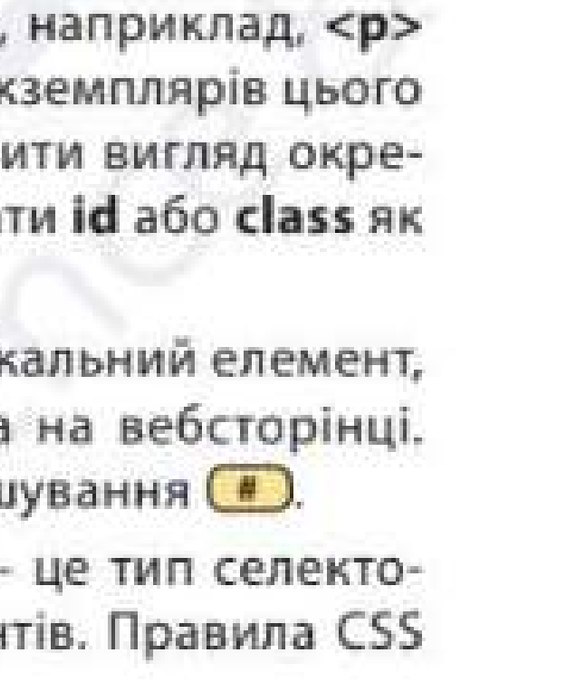
Вказавши значення **width**, встановлюєте точну ширину нашого прямокутника.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

```
5  <style type='text/css'>
6      body {
7          back
8      }
9      p {
10         bac
11         pad
12         col
13         wid
14     }
15 </style>
16 </head>
17 <body>
18 background-color: #6FF;
```

backface-visibility
background
background-attachment
background-blend-mode
background-clip
background-color

Коли починаєте вводити новий CSS-код в редакторі коду, відображається список підказок, що містить назви всіх доступних властивостей, які можна змінити.



*Стилі CSS краще розміщувати на початку коду, всередині тегу **<style>**. У такий спосіб браузер відформатує сторінку під час її відкриття. В іншому випадку він спочатку відкриває вебсторінку, а потім застосовує будь-які стилі.*

Коли селектором коду CSS є стандартний елемент HTML, наприклад, **<p>** або **<h1>**, це означає, що правило застосовується до всіх екземплярів цього елемента по всій вебсторінці. Однак, якщо ви хочете змінити вигляд окремого елемента чи набору елементів, можна використовувати **id** або **class** як селектор.

У першому правилі створюємо **id** з назвою **video** – це унікальний елемент, який може бути застосований лише до одного елемента на вебсторінці. Правила CSS для ідентифікаторів починаються зі знака хешування (#).

У другому правилі створюємо клас з назвою **txtCenter** – це тип селектора, який можна застосувати до будь-якої кількості елементів. Правила CSS для класів починаються з крапки (.).

```
15     #video {
16         width: 400px;
17     }
18     .txtCenter {
19         text-align: center;
20     }
```

Властивостей CSS багато. Більшість назв властивостей можна зрозуміти за їх назвою. Експериментуйте з різними властивостями елементів.

ПІДКАЗКА

Якщо ви зіткнулися з властивістю CSS, функцію якої не можете зрозуміти, або якщо хочете дослідити та дізнатися більше про інші властивості CSS, скористайтеся улюбленою пошуковою системою та надрукуйте **CSS властивості**.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Правила до будь-яких елементів на вебсторінці застосувати дуже легко.

У наведеному прикладі встановили стиль для елемента **video** ідентифікатор **video**, а для другого абзацу – клас **txtCenter**.

```
<!DOCTYPE html>
<html>
<head>
<title>Моя перша вебсторінка</title>
<style type='text/css'>
    body {
        background-color: #6FF;
    }
    p {
        background-color: #3CF;
        padding: 5px;
        color: #FFFFFF;
        width: 300px;
    }
    #video {
       width: 400px;
    }
    .txtCenter {
        text-align: center;
    }
</style>
</head>

<body>
<h1>Це особиста сторінка Оленки</h1>
<p class="txtCenter">Ласкаво просимо на мою особисту вебсторінку.</p>
<p class="txtCenter">Ось коло моїх інтересів:</p>
<ul>
  <li>Фото</li>
  <li>Арт</li>
  <li>Комп'ютери</li>
</ul>

    <h3>Моє відео</h3>

    <!-- Локальне відео -->
    <video id="video" src="design T-shot.mp4"></video>
</body>
</html>
```

Каскадний ефект CSS

Правила CSS можна успадковувати та перевизначати, створюючи ієрархію та взаємозв'язок стилів.

Тут ми визначаємо два правила для елементів **body**: колір шрифту – **#000**, або чорний, а текст має бути вирівняний по правому краю. Однак у правилі CSS для елемента параграфа перевизначаємо властивість **text-align** як **right**.

Визначимо колір шрифту – #000 або чорний, текст вирівняємо по правому краю.

```
<!DOCTYPE html>
<html>
<head>
<title>Моя перша вебсторінка</title>
    <style>
        body {
            color: #000;
            text-align: center;
        }

        p {
            text-align: right;
        }
    </style>
</head>
<body>
    <p>This is a paragraph</p>
</body>
</html>
```


© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Отриманий абзац матиме чорний колір шрифту й буде вирівняний по правому краю. Оскільки елемент **<p>** міститься в елементі **<body>**, він успадковує всі властивості, які не визначені явно в правилі **<p>**, від свого батьківського елемента **<body>**. Так, колір шрифту абзацу успадковується від правила **<body>**. Однак, коли ця ж властивість перевизначається, застосовується правило найбільш специфічного селектора, і саме тому абзац вирівнюється по правому краю.

Зовнішні таблиці стилів

У попередньому прикладі була використана внутрішня таблиця стилів – код CSS розташований в самому файлі **index.html**. Це підходить для невеликих проєктів або в ситуаціях, коли стилі, які ви визначаєте, будуть використовуватися лише на одній сторінці. Буває багато випадків, коли стилі застосовують до багатьох сторінок. У такому випадку копіювання та вставка коду CSS на кожну сторінку – клопітка справа. Окрім того, що сторінки будуть захаращені однаковим кодом CSS, доведеться редагувати кожну з цих сторінок, якщо ви захочете змінити стиль.

Створіть окремий файл CSS – зовнішню таблицю стилів, з усіма вашими правилами, а потім зв'яжіть його з вебсторінкою, до якої треба застосувати ці правила. Введіть правила CSS у файл з розширенням **.css**, наприклад **style.css**, збережіть його в тій самій теці, що й ваш файлі **index.html**. Не додавайте до файлу жодних тегів, наприклад **<style>**, просто введіть правила. Потім видаліть усі внутрішні визначення таблиці стилів з файлу **index.html** та розмістіть наступний рядок між тегами **<head> ... </head>** вашої сторінки.

```
<link href='style.css' rel='stylesheet' type='text/css'>
```

Ваш код буде виглядати так.

```
1  <!DOCTYPE html>
2  <html lang='uk'>
3  <head>
4      <meta charset="UTF-8">
5      <title>Моя перша вебсторінка</title>
6      <meta name='keywords' content='Персональна сторінка Оленки, Оленка Учениця'>
7      <meta name='description' content='Персональна сторінка Оленки Ученицi'>
8      <link href='style.css' rel='stylesheet' type='text/css'>
9  </head>
10
11 <body>
12   <h1>Це особиста сторінка Оленки</h1>
```

Тег **<meta>** повідомляє браузеру користувача, який набір символів і яке кодування необхідно встановити на вебсторінці. Атрибут **charset** задає кодування символів для HTML-документа:

- UTF-8 – кодування символів для Unicode;
- ISO-8859-1 – кодування символів для латинського алфавіту.

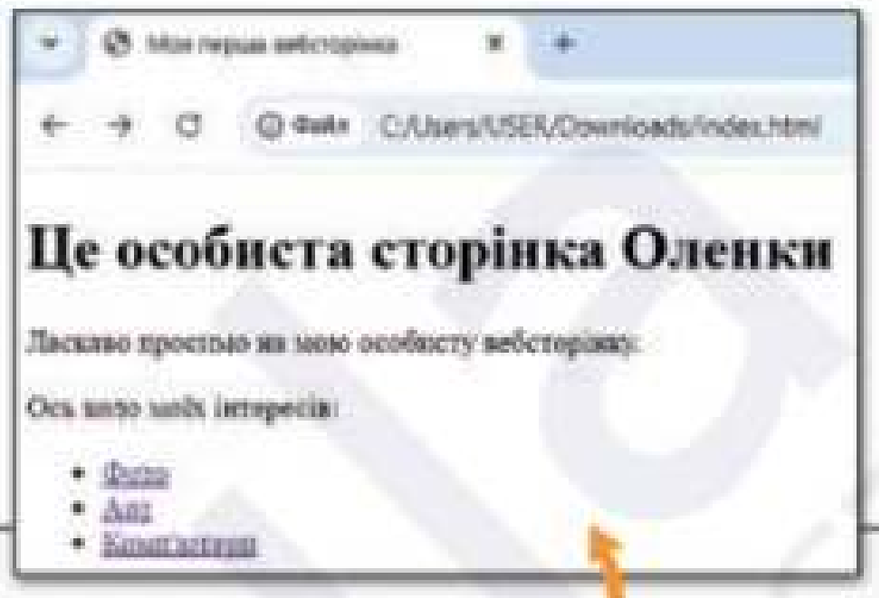
Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 3
Додаємо контент

Досі ви працювали з однією вебсторінкою. Розширимо сайт і додамо декілька сторінок. Щоб полегшити відвідувачам перехід з однієї сторінки на іншу, варто створити навігаційну панель, яка міститиме посилання на інші сторінки вашого сайту.



Створюємо гіперпосилання

```
1  <!DOCTYPE html>
2  <html lang='uk'>
3  <head>
4      <meta charset="UTF-8">
5      <title>Моя перша вебсторінка</title>
6      <meta name='keywords' content='Персональна сторінка Оленки, Оленка Учениця'>
7      <meta name='description' content='Персональна сторінка Оленки Учениці'>
8  </head>
9
10 <body>
11          <h1>Це особиста сторінка Оленки</h1>
12          <p>Ласкаво просимо на мою особисту вебсторінку.</p>
13
14     <p>Ось мої інтереси:</p>
15     <ul>
16         <li><a href="kaboompics_sunset-reflections-a-tranquil-coastal-journey-36819.jpg">Фото</a></li>
17         <li><a href="../page2.html">Арт</a></li>
18         <li><a href="http://htmlbook.at.ua/">Комп'ютери</a></li>
19     </ul>
20
21
22 </body>
23 </html>
```

Позначає **Якір** (**anchor**)

Гіпертекстове посилання, яке вказує місце, куди виконується перехід за наведеним посиланням.

Поєднання **../** вказує на теку, розташовану на один рівень вище від даної позиції файлу, з якого робиться посилання.

Звично, щоб посилання, які вказують на вебсторінки поза вашим сайтом, відкривалися в тому ж вікні або вкладці, щоб користувач не переходив з вашого сайту.

Спробуємо змінити вигляд сторінки за допомогою шрифтів.

Встановлено розмір шрифта 20

```
11 <body>
12     <h1>Це особиста сторінка Оленки</h1>
13     <p style="font-size: 20px;">Ласкаво просимо на мою особисту вебсторінку.</p>
14     <p style="font-family: courier;">Ласкаво просимо на мою особисту вебсторінку.</p>
15     <p style="font-size:20px; font-family: courier ">Ласкаво просимо на мою особисту вебсторінку.</p>
```

Встановлено шрифт Courier

Встановлено шрифт Courier розміром 20

Це особиста сторінка Оленки

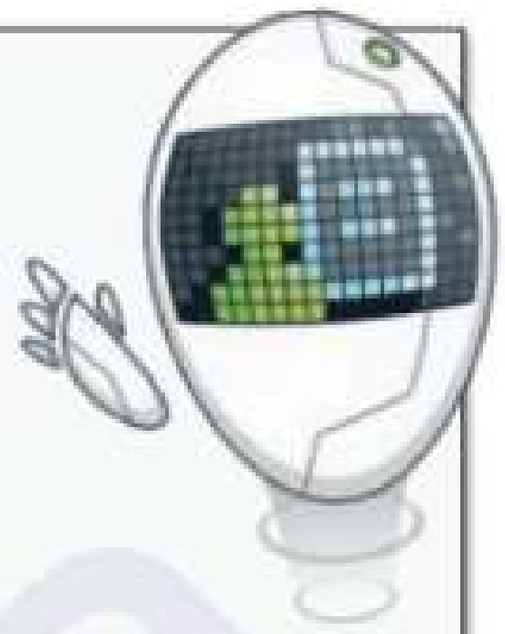
Ласкаво просимо на мою особисту вебсторінку.

Ласкаво просимо на мою особисту вебсторінку.

Ласкаво просимо на мою особисту вебсторінку.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

```html
<!DOCTYPE html>
<html lang='uk'>
<head>
    <meta charset="UTF-8">
    <title>Моя перша вебсторінка</title>
    <meta name='keywords' content='Персональна сторінка Оленки, Оленка Учениця'>
    <meta name='description' content='Персональна сторінка Оленки Учениці'>
</head>

<body>
        <h1>Це особиста сторінка Оленки</h1>
        <p>Ласкаво просимо на мою особисту вебсторінку.</p>
        <nav>Тут буде розміщений тег для створення панелі навігації</nav>

    <p>Ось коло моїх інтересів:</p>
    <ul>
        <li><a href="kaboompics_sunset-reflections-a-tranquil-coastal-journey-30019.jpg">Фото</a></li>
        <li><a href="../page2.html">Арт</a></li>
        <li><a href="http://htmlbook.at.ua/">Комп'ютери</a></li>
    </ul>
</body>
</html>
```



Це особиста сторінка Оленки

Ласкаво просимо на мою особисту вебсторінку.

Тут буде розміщений тег для створення панелі навігації

Ось коло моїх інтересів:

- Фото
- Арт
- Комп'ютери

Визначає набір навігаційних посилань, ці посилання ведуть на інший розділ, головну сторінку, контакти та інше.

```html
<body>
    <h1>Це особиста сторінка Оленки</h1>
    <p>Ласкаво просимо на мою особисту вебсторінку.</p>

    <nav>
        <a href="index.html">Home</a>
        <a href="art.html">Art</a>
        <a href="contact.html">Contact me</a>
    </nav>
</body>
```

Це особиста сторінка Оленки

Ласкаво просимо на мою особисту вебсторінку.

Home Art Contact me

Додаємо зображення

Щоб сторінка привертала увагу, додамо зображення на нашу вебсторінку.

```html
<body>
    <h1>Це особиста сторінка Оленки</h1>
    <p>Ласкаво просимо на мою особисту вебсторінку.</p>

    <nav>
        <a href="index.html">Home</a>|
        <a href="art.html">Art</a>|
        <a href="contact.html">Contact me</a>
    </nav>
    <img src="Зчімок.JPG" width="170" height="300" alt="Робот"/>
</body>
</html>
```

Вказує ім'я графічного файлу, який зберігається в поточному каталозі

Ширина та висота зображення

Визначає альтернативний текст для зображення, якщо воно не може бути відображено з якоїсь причини

Це особиста сторінка Оленки

Ласкаво просимо на мою особисту вебсторінку.

Home Art Contact me

ПІДКАЗКА

Перше навігаційне посилання **Home** веде на домашню сторінку, яка являє собою файл **index.html**. Інші два посилання, **Art** та **Contact me**, ведуть на інші вебсторінки, які ще не створені.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Щоб сторінка виглядала більш привабливо, вирівняємо зображення по середині сторінки, додамо рамку та заллємо кольором простір всередині рамки.

```
<!DOCTYPE html>
<html lang='uk'>
<head>
    <meta charset="UTF-8">
    <title>Моя перша вебсторінка</title>
    <meta name='keywords' content='Персональна сторінка Оленки, Оленка Учениця'>
    <meta name='description' content='Персональна сторінка Оленки Учениці'>
    <link href='style.css' rel='stylesheet' type='text/css'>
</head>

<body>
    <h1>Це особиста сторінка Оленки</h1>
    <nav>
        <a href="index.html">Home</a>|
        <a href="art.html">Art</a>|
        <a href="contact.html">Contact me</a>
    </nav>
    <center>
        <img src="Screen.JPG" alt="Робот" border="5" width="150"
         style="background-color:lightgray; padding:10px;">
    </center>
</body>
</html>
```

Вирівнює зображення по центру

Додає рамку навколо зображення товщиною в 5 пікселів

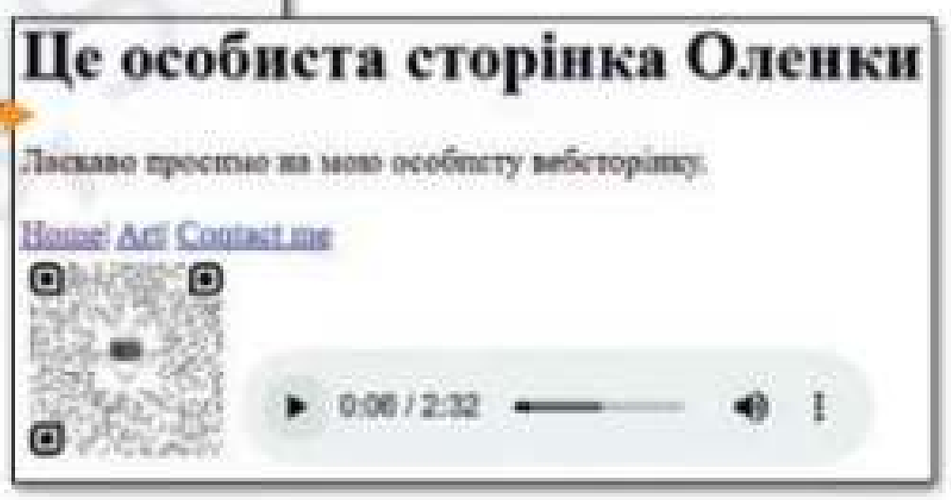
Робить світло-сірим тло в середині рамки

Додає внутрішній відступ товщиною в 10 пікселів навколо зображення

Додаємо звук

За допомогою HTML5 дуже легко додати звук на будь-яку вашу вебсторінку.

```
    <img src="qr-code.png" width="100" height="100" alt="qr-code"/>
    <audio controls>
        <source src="Etude.mp3" type="audio/mp3">
    </audio>
</body>
</html>
```



ПІДКАЗКА

Якщо не вказано ширину та висоту зображення, під час завантаження зображення сторінка може мерехтіти.

Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 4
Вебформи

Створимо останню вебсторінку вашого сайту. Це сторінка контактів, на якій відвідувачі вашого сайту зможуть заповнити вебформу та надіслати вам повідомлення.

Створіть нову HTML-сторінку з ім'ям **contact.html**, скопіюйте початковий HTML-код з домашньої сторінки, щоб зберегти загальний вигляд сайту. Ось так виглядає HTML-форма в режимі перегляду дизайну.

А ось HTML-код, який створив його.

** ** – нерозривний пробіл. Вставляється в HTML-код замість звичайного пробілу. Найкраще використовувати ** ** як вміст порожньої комірки таблиці, оскільки деякі браузери не можуть коректно відображати їх без нього.

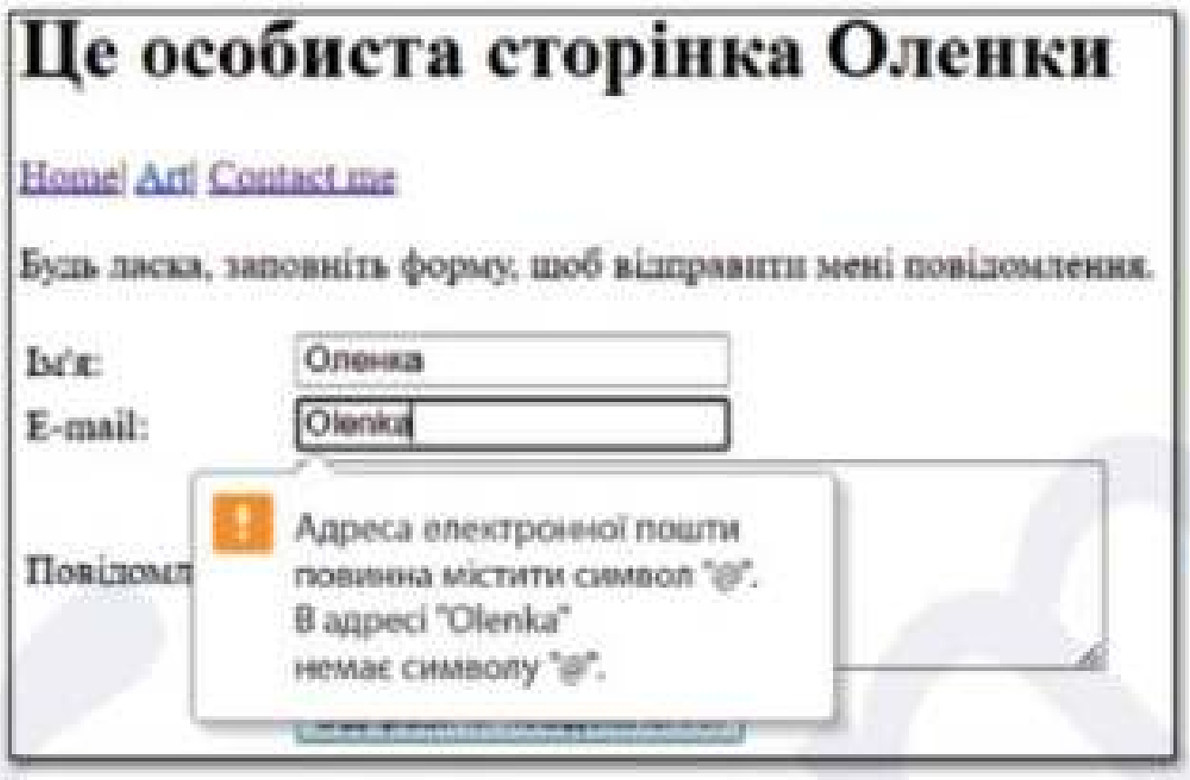
```
18      <p>Будь ласка, заповніть форму, щоб відправити мені повідомлення.</p>
19      <form action="contactForm.php" method="post" id="form1"> <table width="400">
20          <tr>
21              <td>Ім'я:</td>
22              <td><input name="name" type="text"></td>
23          </tr>
24          <tr>
25              <td>E-mail:</td>
26              <td><input type="email" name="email"></td>
27          </tr>
28          <tr>
29              <td>Повідомлення:</td>
30              <td><textarea name="message" cols="40" rows="5" id="message"></textarea></td>
31          </tr>
32          <tr>
33              <td> </td>
34              <td><input type="submit" value="Відправити повідомлення"></td>
35          </tr>
36      </table>
37      </form>
```

HTML-форма завжди міститься в тегах **<form>**. Цей тег має дві важливі властивості: **action** і **method**. Властивість **action** визначає сторінку, на яку буде відправлено форму, а властивість **method** – спосіб відправлення. Властивість **method** може мати значення **post** або **get** – два стандартних методи для відправки даних з однієї сторінки на іншу.

У цьому випадку форма надсилається на сторінку **contactForm.php** за допомогою методу **post**. Сторінка **contactForm.php** – це серверний скрипт. По суті, це набір команд для обробки переданих даних форми.

© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Усередині **<form>** у нас є кілька тегів **<input>**. Тип кожного поля у формі визначається властивістю **type** тегу **<input>**. Перше поле є простим текстовим полем. Друге має тип **email** – це текстове поле, але воно підтримує автоматичну перевірку електронної пошти. Це функція HTML5, яка не дозволяє користувачеві відправити форму без правильно написаної адреси електронної пошти в полі **email**. Якщо користувач намагається надіслати форму без коректної адреси електронної пошти, форма не буде надіслана та з'явиться вигулькове повідомлення. Останнє поле форми – текстова область, яка визначається тегом **<textarea>**. Текстова область – це, по суті, багаторядкове текстове поле. Властивості **cols** і **rows** визначають розмір текстової області.

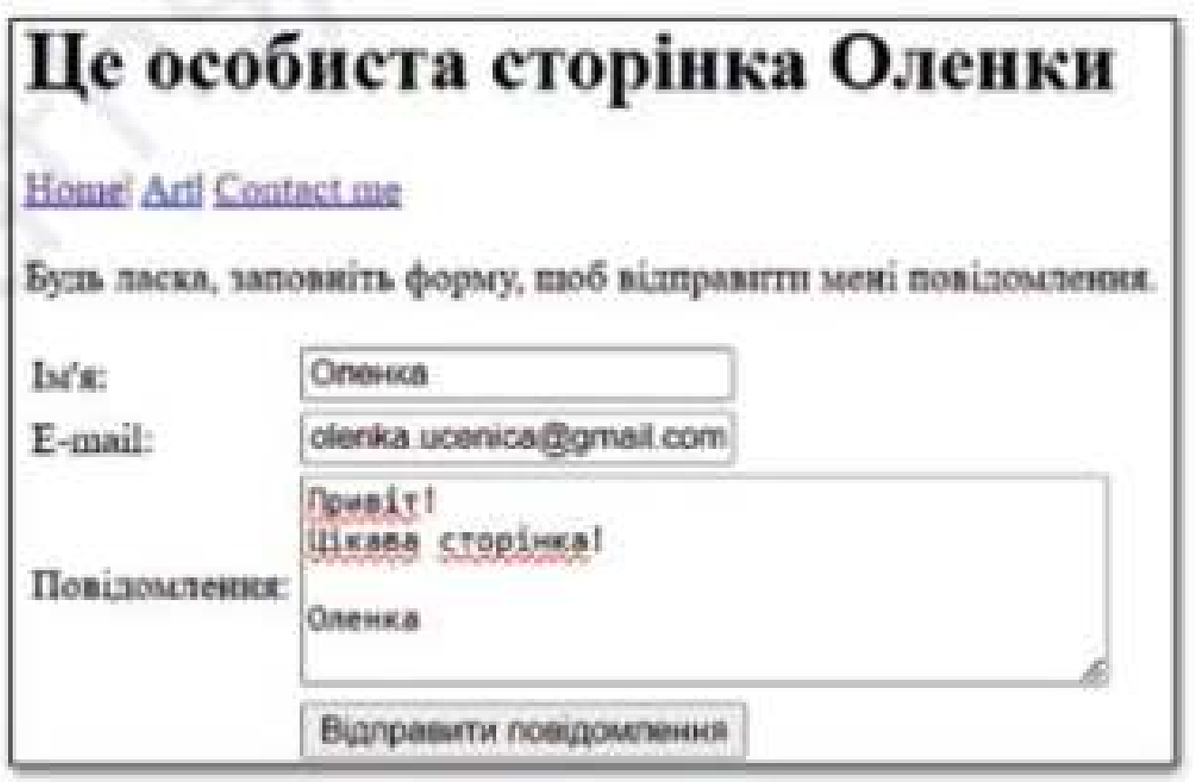


*При роботі з полями HTML-форми важливо задати властивість **name** для кожного поля. Це ім'я буде назвою змінної, яка буде використовуватися в серверному php-скрипті для доступу до кожного поля форми.*

Форма має елемент управління типу **submit** – це кнопка, що натискається користувачем для відправлення форми. Властивість **value** дозволяє визначити підпис кнопки.

Зверніть увагу, що всі поля форми містяться в тегах **<table>**. Тут використовуємо структуру HTML-таблиці, щоб легко вирівняти поля форми з їхніми підписами. Усередині HTML-таблиці теги **<tr>** визначають рядок таблиці, а теги **<td>** – стовпець таблиці. У цій формі створили таблицю з чотирьох рядків, що містять по два стовпці.

Коли відвідувач сайту заповнює форму й надсилає її, дані з кожного поля надсилаються на нашу php-сторінку, яка обробляє дані та надсилає нам електронного листа з повідомленням.



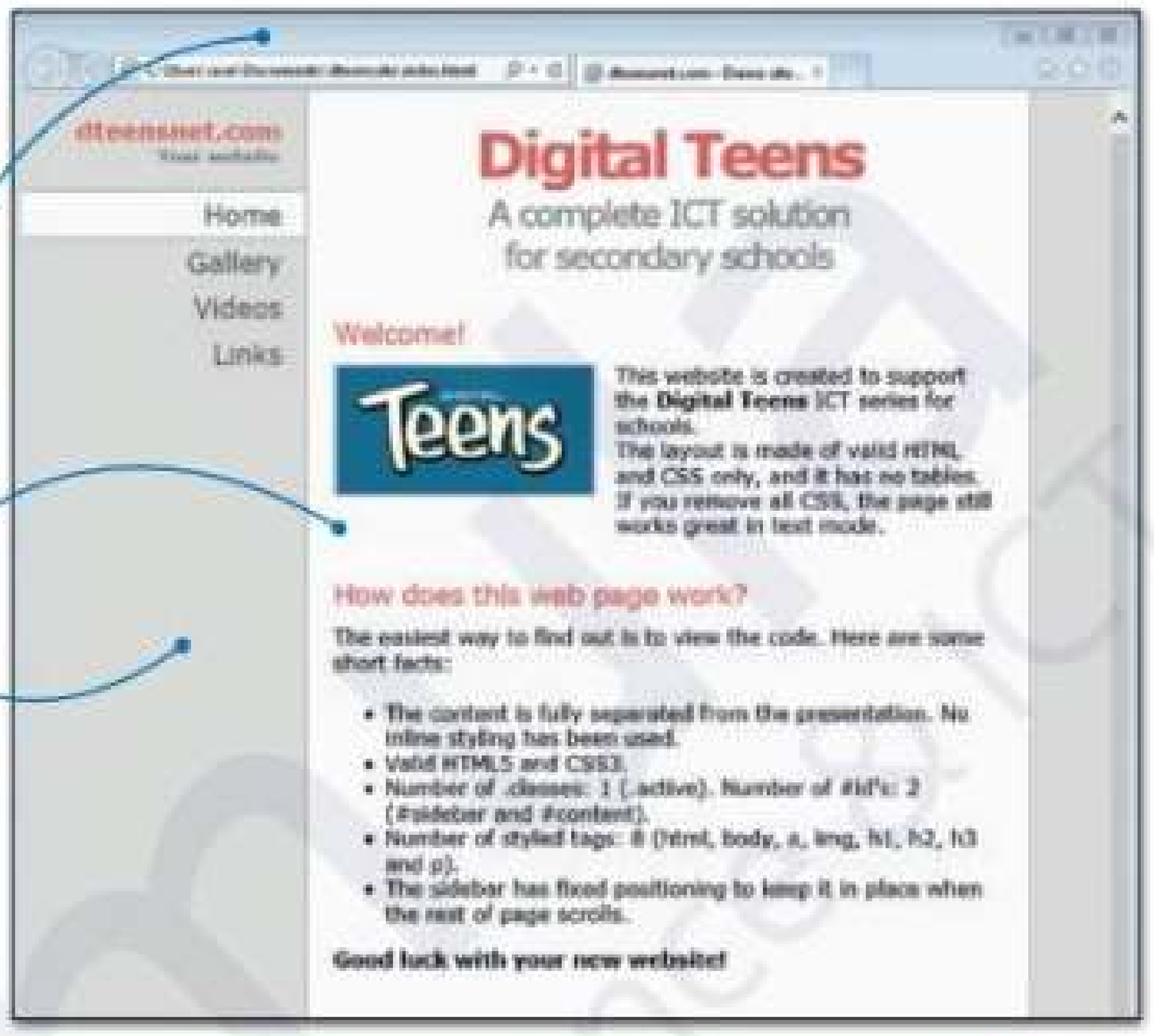
БЕРЕЖИ СЕБЕ

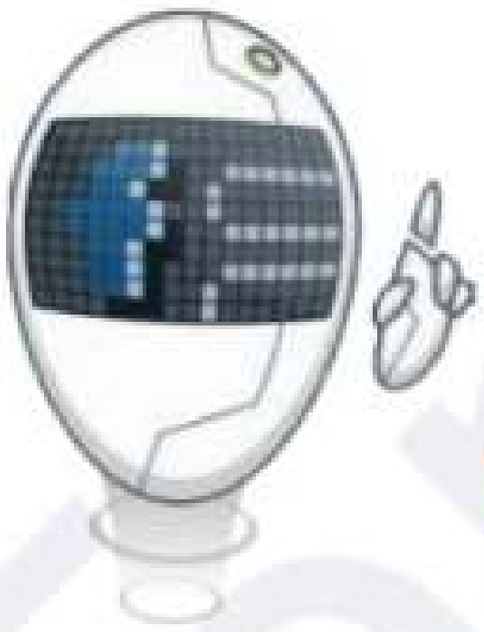
У 2024 році Оксфордський словник обрав словом року вираз «brain rot» (англ. – «гниття мозку»). Цей термін описує погіршення розумових здібностей людини через надмірне споживання неякісного або беззмістовного контенту, особливо в інтернеті.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Створення більш складних макетів

Ви вже знаєте, як створювати простий вебсайт. Створимо вебсайт на основі макета з двох колонок. Наша мета полягає в тому, щоб остаточний макет сайту виглядав, як показано на наступному зображенні.





Верстка сайтів колонками стала популярною серед персональних блогів, але зараз вона використовується й на багатьох найбільш відвідуваних вебсайтах, як-от ***Facebook*** *і* ***Gmail****.*

Створіть теку для сайту. Потім створіть новий HTML-файл і збережіть його у вашій теці під назвою **index.html**. Також створіть файл **CSS** і збережіть його в тій самій теці під назвою **style.css**. Як завжди, зв'яжіть ваш HTML-файл з файлом таблиці стилів у розділі **<head>** вашого HTML-файлу. Додайте заголовок сторінки, опис і деякі ключові слова.

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>dteensnet.com - Demo site for Digital Teens ICT series</title>
<meta name="description" content="Demo site for Digital Teens ICT series by Binary Logic">
<meta name="keywords" content="Digital Teens, ICT, Binary Logic, Technology, Computers in schools">
<link href="style.css" rel="stylesheet" type="text/css">
</head>

<body>
</body>
</html>
```


© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Створіть базову HTML-структуру нашої сторінки.

```
<body>
<div id="sidebar">Content for  id "sidebar" Goes Here</div>
<div id="content">Content for  id "content" Goes Here</div>
</body>
```

Використовуємо HTML-елемент **div** – контейнер, якому можна присвоїти ідентифікатор або клас. Він використовується, щоб групувати блоки інформації та форматувати її за допомогою CSS. Елементи **div** дуже корисні для створення макетів вебсторінок. Створіть елемент **div** з ідентифікатором **sidebar та** другий **div** елемент з ідентифікатором **content**. Ці дві структури будуть нашими двома колонками. Поки вони виглядають не дуже привабливо, оскільки ще не створені правила **CSS** для цих ідентифікаторів. Текст всередині елементів **div** – просто заповнювач, щоб можна було побачити, як вони виглядають, перш ніж додавати реальний контент.

```
Content for id "sidebar" Goes Here
Content for id "content" Goes Here
```

```
#sidebar {
    width: 220px;
    background-color: #E0E0E0;
}
#content {
    width: 520px;
    background-color: #FAFAFA;
}
```

Створіть правила CSS, щоб оформити наш макет. У файлі CSS додайте правила для бічної панелі та основної колонки. Почніть з визначення ширини та кольору фону для кожного з елементів **div**.

```
Content for id "sidebar" Goes Here
Content for id "content" Goes Here
```

Зараз блоки **div** не відображаються поруч один з одним, як колонки, а розташовуються один над одним за замовчуванням.

Зробіть їх більш схожими на колонки.

Не забувайте зберігати файли, коли вносите зміни та перед тим, як переглядати їх у вебпереглядачі.

```
#sidebar {
    width: 220px;
    background-color: #E0E0E0;
    position: fixed;
    top: 0px;
    left: 0px;
    border-right: 2px solid #C0C0C0;
    text-align: right;
}

#content {
    width: 520px;
    background-color: #FAFAFA;
    margin: 0 0 0 222px;
    padding: 20px;
    border-right: 2px solid #C0C0C0;
}
```

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Зафіксуйте положення елементу **sidebar** у верхньому лівому кутку сторінки за допомогою властивостей:

```
position: fixed;
top: 0px;
left: 0px;
```

Це означає, що **sidebar** завжди буде з'являтися на тому ж місці, навіть якщо решта сторінки, тобто блок **content**, прокручується. Бічна панель – закріплена, завжди видима. Додайте праву рамку та визначте, що весь текст всередині **sidebar** буде вирівняний по правому краю. Для блоку **content** найважливіша властивість для макета – відступ ліворуч.

Встановлює зовнішні відступи (поля) навколо елемента

```
margin: 0 0 0 222px;
```

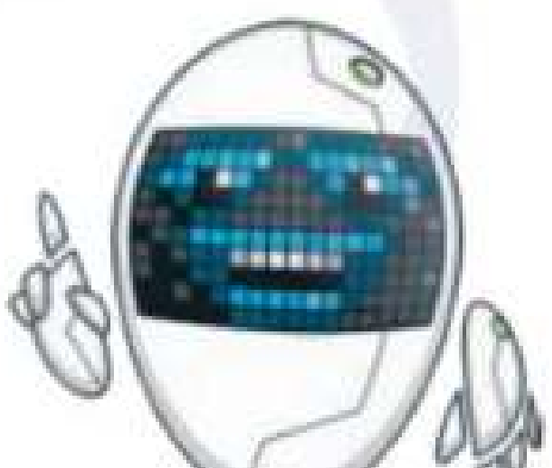
Ця властивість визначає, що верхнє, праве та нижнє поля навколо блоку **content** дорівнюють **0**, а ліве – **222px.** По суті, весь блок **content** на 222px відступає вправо, щоб він з'явився поруч з блоком **sidebar**. Чому 222px? Блок **sidebar** має ширину 220px, додамо 2px з правого краю поруч із **content**.

```
html {
    height: 100%;
}

body {
    background-color: #E0E0E0;
    font-family: Verdana, sans-serif;
    margin: 0;
    height: 100%;
}

#sidebar {
    width: 220px;
    background-color: #E0E0E0;
    position: fixed;
    top: 0px;
    left: 0px;
    border-right: 2px solid #C0C0C0;
    text-align: right;
    height: 100%;
}

#content {
    width: 520px;
    background-color: #FAFAFA;
    margin: 0 0 0 222px;
    padding: 20px;
    border-right: 2px solid #C0C0C0;
    min-height: 100%;
}
```

Чому вигляд не відповідає очікуванням?

Проблема з невідповідним розміщенням елементів та наявністю небажаних проміжків (невеличкий відступ у верхній частині сторінки) виникає через вбудовані параметри браузера за замовчуванням. Зокрема, елемент **<body>** має стандартні відступи, які порушують бажаний макет сторінки.

Щоб досягти ідеального розташування та створити суцільні колонки, які розтягуються на повну висоту екрана, необхідно встановити властивість **height: 100%** для всіх контейнерів. Це означає, що кожен елемент повністю займе доступний вертикальний простір. Макет сторінки буде без зайвих проміжків.



Зараз це дійсно виглядає, як справжній макет з двох колонок. Зверніть увагу, що для елемента **<html>** треба встановити висоту 100%, оскільки елемент **<body>** включено в елемент **<html>**, а все інше знаходиться всередині елемента **<body>**. Додано колір фону до елемента **<body>**, який відповідає кольору **sidebar**, змінено шрифт за замовчуванням для всієї сторінки. У блоці **content** властивість **min-height** має значення 100% замість **height**. Це змушує ** ** виходити за видимий низ сторінки у випадках, коли вміст ** ** є довшим і з'являється смуга прокрутки.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Додайте в структуру HTML вміст блоку **sidebar**: лише заголовок, підзаголовок та навігаційні посилання.

```
<body>
<div id="sidebar">
  <h1>dteensnet.com</h1>
  <h2>Your website</h2>
  <a class="active" href="index.html">Home</a>
  <a href="gallery.html">Gallery</a>
  <a href="videos.html">Videos</a>
  <a href="links.html">Links</a>
</div>
```

Застосуйте деякі правила CSS до елементів, щоб створити привабливе навігаційне меню.

```
#sidebar h1 {
    font-size: 120%;
    color: #085050;
    margin: 20px 10px 0 5px;
}

#sidebar h2 {
    font-size: 80%;
    color: #808080;
    margin: 0 20px 10px 5px;
    font-weight: bold;
}

#sidebar a {
    display: block;
    color: #606060;
    background-color: #E0E0E0;
    padding: 5px 10px 5px 0;
    font-size: 130%;
    font-weight: normal;
    text-decoration: none;
}

#sidebar a:hover {
    color: #303030;
    background-color: #F0F0F0;
}

#sidebar a.active {
    background-color: #FAFAFA;
    border-top: 2px solid #C0C0C0;
    border-bottom: 2px solid #C0C0C0;
}

#sidebar a.active:hover {
    color: #505050;
    background-color: #FAFAFA;
}
```

Селектор **#sidebar h1** означає застосування цього правила до всіх заголовків **h1**, які знаходяться всередині елемента з ідентифікатором **sidebar**. Це спосіб створити правила стилів для елементів тільки всередині певного контейнера.

Всі розміри шрифтів визначені у відсотках від розміру шрифту за замовчуванням, який у всіх браузерах дорівнює 16px. При визначенні розміру шрифтів сторінок у відсотках, текст вашого сайту краще масштабується при перегляді з мобільних пристроїв з маленькими екранами або людьми з вадами зору, які використовують допоміжні засоби доступності.

ІСТОРІЯ

До того, як використання тегів **div** і правил CSS стало стандартом верстки вебсторінок, вебдизайнери створювали свої макети за допомогою HTML-таблиць. **div**'и схожі на таблиці, але вони простіші у використанні, їх можна налаштовувати за допомогою CSS та їх завантаження відбувається швидше, ніж таблиць.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

У правилі CSS для елементів **<a>**, які є нашими навігаційними посиланнями, використано властивість **display: block;**. Це змушує елементи **<a>** поводитися, як блокові елементи (як елементи ** **, які були використані раніше), тому кожен елемент розширюється та займає стільки місця по горизонталі, скільки займає контейнер, в якому він знаходиться. Це змушує посилання переміщатися один над одним, а також робить всю область блоку доступною для кліку, що полегшує навігацію. Властивість **text-decoration: none;** прибирає підкреслення елементів **<a>** за замовчуванням.

a:hover – спеціальний селектор, який застосовує правило CSS, коли курсор миші розміщується над вказаним елементом. В даному випадку це посилання **<a>**. Коли користувач наводить курсор миші на посилання в розділі **sidebar**, створюється динамічний ефект: змінюється колір фону та шрифту.

Клас з ім'ям **.active** застосовується до посилань поточної сторінки та виділяє його, шляхом зміни кольору тла та додавання верхньої та нижньої меж.

Додайте HTML-структуру до блоку **content**.

```
<div id="content">
  <h1>Digital Teens</h1>
  <h2>A complete ICT solution <br />
    for secondary schools</h2>

  <h3>Welcome!</h3>
  <p><img src="photo.jpg" alt=""/>This website is created to support the <strong>Digital Teens</strong> ICT
series for schools. <br />
    The layout is made of valid HTML and CSS only, and it has no tables. If you renove all CSS, the page still
works great in text node.</p>

  <h3>How does this web page work?</h3>
  <p>The easiest way to find out is to view the code. Here are some short facts:</p>
  <ul>
    <li>The content is fully separated from the presentation. No inline styling has been used.</li>
    <li>Valid HTML5 and CSS3.</li>
    <li>Number of .classes: 1 (.active). Number of #id's: 2 (#sidebar and #content).</li>
    <li>Number of styled tags: 8 (html, body, a, img, h1, h2, h3 and p).</li>
    <li>The sidebar has fixed posttioning to keep it in place when the rest of page scrolls.</li>
  </ul>
  <p><strong>Good luck with your new website!</strong></p>

  <h3>Site info</h3>
  <p>&copy; 2013 Binary Logic SA. <br />
    Digital Kids and Digital Teens are tradenarks of Binary Logic SA.</p>
</div>
</body>
</html>
```

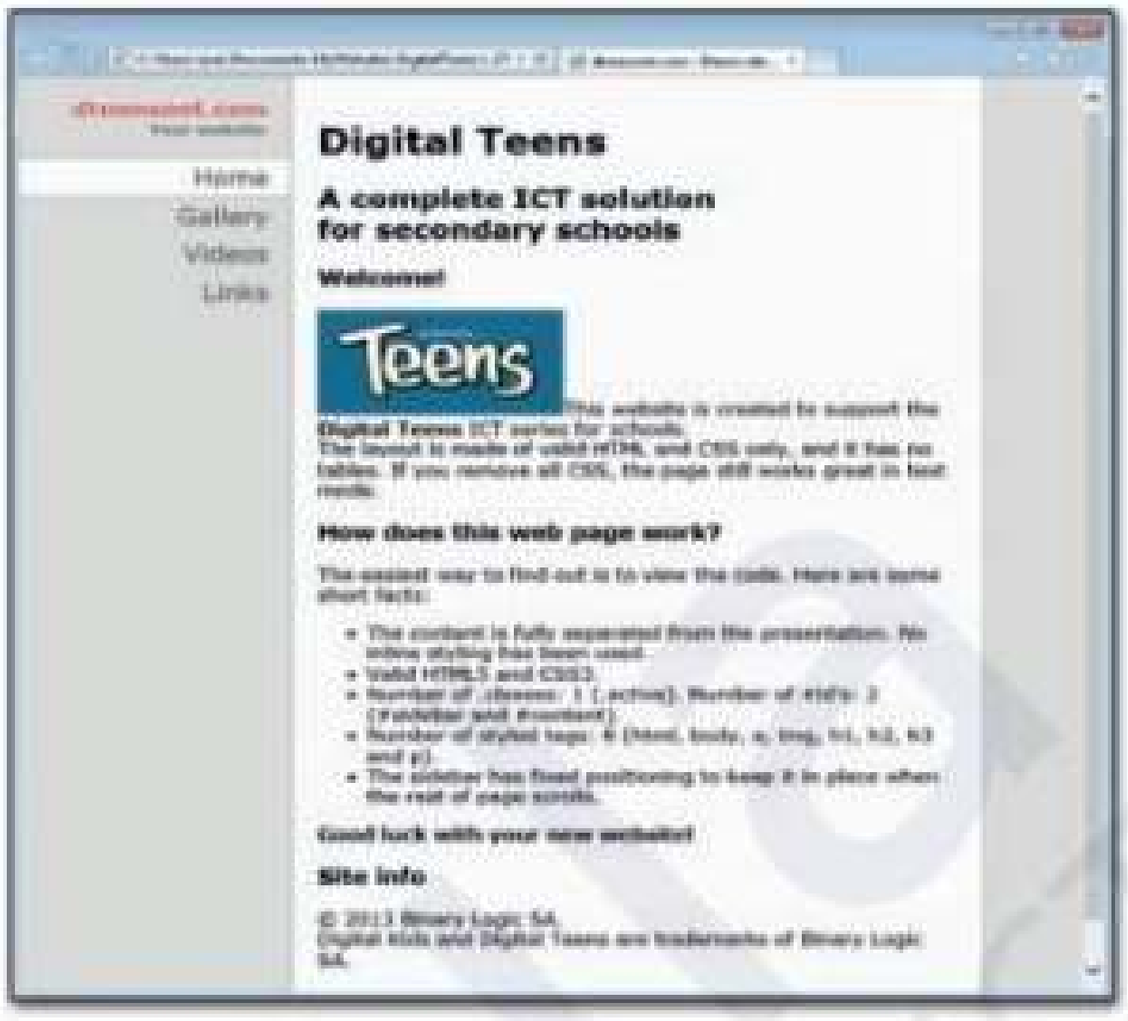
Переконайтеся, що ваш код працює в різних браузерах. Перевірте його в останніх версіях топ-5 браузерів: Internet Edge, Firefox, Chrome, Safari та Opera.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Поля та відступи

У CSS можна вказати різні відступи для кожної зі сторін:

```
margin-top:100px;
margin-bottom:100px;
margin-right:50px;
margin-left:50px;
```



Поле – властивості-скорочення

Щоб скоротити код, можна вказати всі властивості полів в одній властивості. Це називається властивості-скорочення.

Скорочена властивість для всіх властивостей полів – **margin**, а для властивостей відступів – **padding**.

Властивість **margin** (аналогічно з властивістю **padding**) може мати від одного до чотирьох значень.

margin:25px 50px 75px 100px;	верхнє поле 25px, праве поле 50px, нижнє поле 75px, ліве поле 100px
margin:25px 50px 75px;	верхнє поле – 25px, праве та ліве - 50px, нижнє – 75px
margin:25px 50px;	верхнє та нижнє поля – 25px, праве та ліве – 50px
margin:25px;	всі чотири поля по 25px

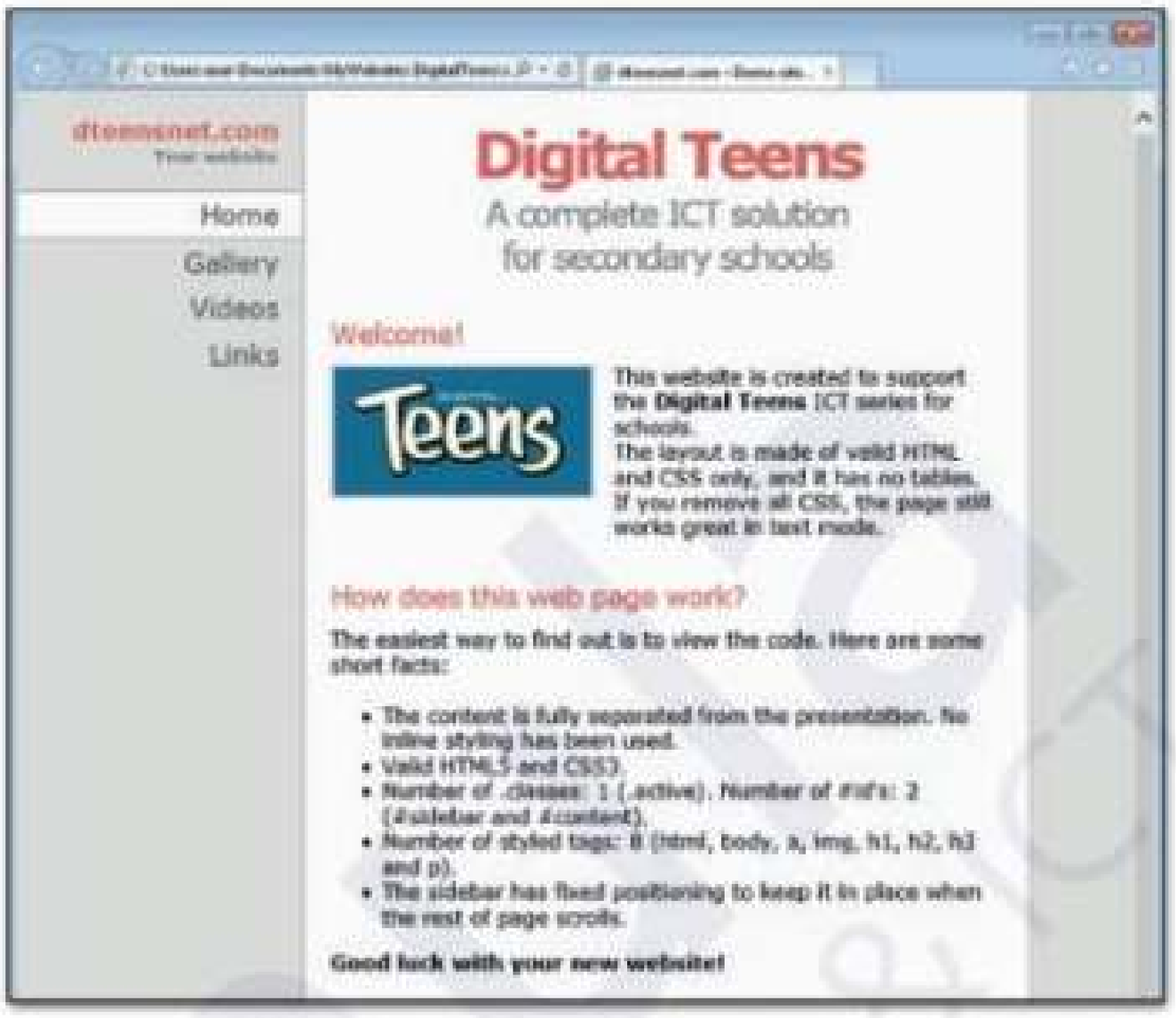
Поле очищає область навколо елемента (поза межами). Воно не має кольору фону та повністю прозоре.

Відступ очищає область навколо вмісту (всередині рамки) елемента. Відступ – прозорий, тому приймає колір фону елемента.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Додайте відповідний стиль за допомогою деяких правил CSS.

```
a {
    color: #d85d5d;
    font-weight: bold;
    text-decoration: none;
}

a:hover {
    color: #505050;
    text-decoration: underline;
}

img {
    float: left;
    margin: 0 15px 15px 0;
    padding: 1px;
    background: #ffffff;
    border: 1px solid #d0d0d0;
}

#content h1 {
    margin: 0;
    font-size: 300%;
    color: #D85D5D;
    text-align: center;
    letter-spacing: -4px;
}

#content h2 {
    margin: 0;
    font-size: 170%;
    font-weight: normal;
    color: #808080;
    text-align: center;
    letter-spacing: -2px;
}

#content h3 {
    margin: 30px 0 10px 0;
    font-size: 140%;
    font-weight: normal;
    color: #D85D5D;
    letter-spacing: -1px;
}

#content p {
    margin: 0 0 20px 0;
}
```



Правила для елементів **<a>** та особливого випадку **a:hover** визначають поведінку всіх посилань на сторінці. Зауважте, що оскільки в CSS правила каскадні, всі властивості, визначені тут, успадковуються іншими правилами CSS. І далі інші правила CSS можуть визначати більш специфічні випадки того ж самого селектора, наприклад **#sidebar a**. Однак, якщо більш спеціалізований селектор перевизначає успадковану властивість, успадковане значення буде перевизначено.

Всі інші правила – базовий стиль. Зверніть увагу на властивість **float: left;** у правилі CSS для елемента **<img>**. Це дозволить усім зображенням спливати ліворуч, а іншим елементам, як-от текст елемента абзацу в даному випадку, займати простір, що залишився праворуч від зображення. На цьому завершимо макет двоколонної вебсторінки. Використайте цей файл CSS для інших сторінок і заповніть кожну з них відповідним контентом. Пам'ятайте, що при створенні нової сторінки для вебсайту обов'язково застосовуйте активний клас до посилання поточної сторінки в навігаційній панелі. Це створить враження, що поточна сторінка «обрана».

ПІДКАЗКА

Назви ідентифікаторів або класів елементів важливі.
Ніколи не використовуйте назви, які описують фізичні атрибути (наприклад, redLink), оскільки зміна значення пізніше може зробити назву непридатною.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Типи медіа в CSS

Залежно від типу медіа сторінка може мати один макет для екрану комп'ютера, інший – для друку, ще один – для портативних пристроїв тощо. Можна створити різні файли CSS для кожного пристрою та зв'язати їх на вашій сторінці. У наступному прикладі визначено дві різні таблиці стилів для різних типів медіа (екран і друк).

```
<head>
<link rel="stylesheet" type="text/css" href="style.css">
<link rel="stylesheet" type="text/css" href="print.css" media="print">
</head>
```

Якщо опустити властивість медіа у тезі посилання, то значенням за замовчуванням буде **screen**, призначене для комп'ютерних моніторів. Як правило, потрібно використовувати різні значення для одних й тих самих властивостей для різних пристроїв. Наприклад, документ зазвичай потребує більшого розміру шрифту на екрані, ніж на папері, а шрифти без зарубок легше читати на екрані, тоді як шрифти із зарубками легше читати на папері.

Зверніть увагу, що для шрифту встановлено напівжирне креслення, як на екрані, так і на папері.

Правило @media

Правило **@media** дозволяє застосовувати різні правила стилів для різних медіа в одній таблиці стилів. Стиль у наведеному нижче коді вказує браузеру відображати на екрані шрифт Verdana розміром 14 пікселів. Але якщо сторінка буде надрукована, вона буде мати шрифт Times розміром 10 пікселів.

```
@media screen
 {
 p {font-family:verdana,sans-serif;font-size:14px;}
 }
@media print
 {
 p {font-family:times,serif;font-size:10px;}
 }
@media screen,print
 {
 p {font-weight:bold;}
 }
```

Якщо бажаєте заощадити чорнило та папір ваших відвідувачів, важливо створити зручні для друку таблиці стилів. Використовуйте правило друку ***@media*** *та видаляйте все, що не потрібно на друкованій сторінці. Наприклад, кольорові фони, стилістичні зображення, навігаційне меню тощо.*

Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 5
Хостинг та SEO

Отже, ви навчилися створювати вебсторінку, але зараз її бачите тільки ви на своєму комп'ютері. Для того, щоб ваша вебсторінка була доступна будь-кому у світі через інтернет, вам потрібно завантажити її на вебсервер.

Вебсервер – це комп'ютерна система, яка забезпечує доставку вебсторінок. Кожен вебсервер має IP-адресу і здебільшого відповідне доменне ім'я. Наприклад, коли ви вводите **www.wikipedia.org** в адресний рядок браузера або натискаєте на посилання, яке вказує на **Вікіпедію**, ваш браузер надсилає запит на вебсервер, доменне ім'я якого **wikipedia.org**. Вебсервер відповідає на цей запит та передає комп'ютеру потрібну вебсторінку. В цьому випадку – головна, тому сервер отримує сторінку за замовчуванням з назвою **index.html**, якщо вона існує, і надсилає її назад браузеру, який її запитував.

ДОВІДКА

Технічно будь-який комп'ютер з доступом до інтернету може працювати як вебсервер після встановлення відповідного серверного програмного забезпечення. Для популярних вебсайтів з високим навантаженням використовують інший підхід. Такі сайти розміщують на спеціальних серверних фермах – це кластери потужних комп'ютерів, об'єднаних високошвидкісним інтернет-підключенням, які здатні обробляти велику кількість одночасних запитів від користувачів.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Послуги вебхостингу

Найпростіший спосіб зробити вашу вебсторінку або вебсайт доступним в інтернеті – зареєструватися на вебхостингу. **Вебхостинг** – це компанія, яка підтримує певну кількість вебсерверів, які ви можете використовувати для розміщення вашого вебсайту. Існує багато компаній, що пропонують як безкоштовні, так і платні послуги вебхостингу. Щоб знайти варіанти безкоштовного хостингу, скористайтеся пошуком «безкоштовний хостинг» у вашій улюбленій пошуковій системі, наприклад, **000webhost.com**, **webfreehosting.net** або **freehosting.com**.

Щоб зареєструвати безкоштовний акаунт на хостингу:

- перейдіть за посиланням **webfreehosting.net**;
- у віконці **Free web hosting** натисніть **order now** (1);
- оберіть безкоштовний вебсайт, клацнувши **Free** (2);
- натисніть **Continue** (3);
- заповніть обов'язкові поля, промарковані * (4) та за бажанням – * (5). Підтвердить, що ознайомились з угодами (6);
- підтвердіть, що ви не робот (7);
- клацніть **Continue** (8);
- на адресу електронної пошти, яку ви вказали у формі, буде надіслано лист із підтвердженням (9).

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Перевірте свою електронну пошту та перейдіть за посиланням для підтвердження вашого акаунту. Ви перейдете на сторінку свого облікового запису.

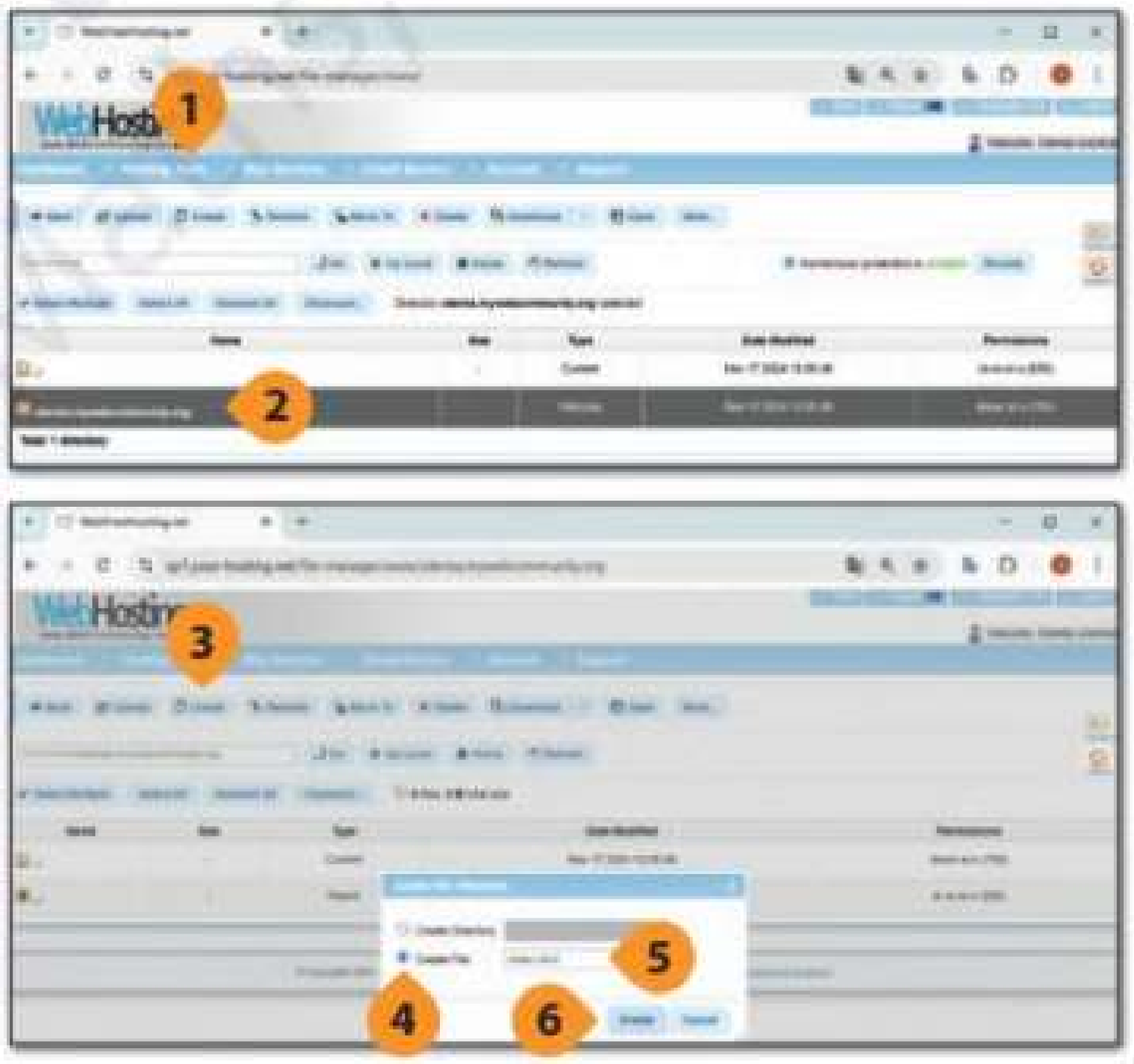
Щоб створити сайт:

- на панелі меню натисніть **Hosting Tools** (1) та у спадному меню оберіть **Domain Manager** (2), або на сторінці сайту клацніть **Domain Manager** (3);
- оберіть вкладку **Create a Free Subdomein** (4);
- у рядку введіть бажану URL-адресу (5);
- у спадному меню оберіть бажаний домен (6);
- клацніть **Create** (7);
- у вкладці **Subdomains** (8) з'явилася назва вашого сайту (9).

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Щоб завантажити вашу вебсторінку:

- на панелі меню натисніть **Hosting Tools** (1) та у спадному меню оберіть **File Manager**;
- відкрийте теку з назвою вашого сайту (2);
- клацніть **Create** (3);
- у вигульковому вікні оберіть **Create File** (4). Надрукуйте назву файлу **index.html** (5);
- клацніть **Create** (6);
- відкрийте файл (7), та вставте необхідні теги й текст для створення сторінки (8);
- натисніть **Save** (9);
- відкрийте ваш сайт через URL-адресу (10).



© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Ваша вебсторінка тепер доступна будь-кому в інтернеті. Введіть адресу вашого сайту в браузері, щоб переглянути сторінку, яку ви щойно завантажили. Дотримуючись цієї ж процедури, ви можете завантажити інші сторінки на свій вебсайт пізніше.

Пошукова оптимізація

Тепер ви знаєте, як завантажити свої вебсторінки на вебсервер.

Як люди можуть знайти ваш сайт та відвідати його? Звичайно, ви можете надіслати адресу своїм друзям і родичам, але як бути з іншими?

Щоб люди могли легко знайти ваш сайт через Google чи інші пошукові системи, потрібно зробити його «помітним» для них. Цей процес називається **SEO** (англ. Search Engineering Optimization – пошукова оптимізація) – вона робить сайт більш зрозумілим для пошукових систем, щоб вони могли показувати його людям, які шукають схожу інформацію чи послуги.

Це схоже на те, як розставляти товари в магазині так, щоб покупці могли їх легко знайти. Тільки тут ви «розставляєте» інформацію на сайті так, щоб пошукова система могла її легко знайти та показати певним людям.

Як правило, пошукова система сканує вміст ваших вебсторінок і шукає ключові слова. Якщо користувач шукає певне ключове слово або комбінацію ключових слів, які є на вашій сторінці, ваша сторінка з більшою ймовірністю з'явиться в результатах пошуку. На кожній сторінці є інформація, невидима для користувача, який переглядає вашу сторінку, але видима для пошукових систем, які сканують ваш сайт. Ця додаткова інформація називається метаданими й допомагає пошуковій системі надавати користувачам більш релевантні результати, а також збільшити відвідуваність вашого сайту.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Додаємо description

Тег **description** – це тип метаданих для вашої вебсторінки, який підсумовує вміст сторінки. Важливо надавати короткі і добре написані описи всіх ваших вебсторінок. Опис зазвичай з'являється під заголовком вашої вебсторінки в результатах пошуку в пошуковій системі. Гарний якісний опис привабить більше користувачів на ваш сайт.

```
1   <!DOCTYPE html>
2 ▼ <html lang='uk'>
3 ▼ <head>
4       <meta charset="UTF-8">
5       <title>Моя перша вебсторінка</title>
6       <meta name='keywords' content='Персональна сторінка Оленки, Оленка Учениця'>
7       <meta name='description' content='Персональна сторінка Оленки Ученицi'>
8       <link href='style.css' rel='stylesheet' type='text/css'>
9   </head>
10
11 ▼ <body>
12      <h1>Це особиста сторінка Оленки</h1>
13      <p>Ласкаво просимо на мою особисту вебсторінку.</p>
14
15 ▼    <nav>
16          <a href="index.html">Home</a>|
17          <a href="art.html">Art</a>|
18          <a href="contact.html">Contact me</a>
19      </nav>
20      <img src="Снимок.JPG" width="170" height="300" alt="Робот"/>
21 ▼    <audio controls>
22          <source src="Etude.mp3" type="audio/mp3">
23      </audio>
24  </body>
25  </html>
```

keywords

description

Додаємо keywords

Тег **keywords** – це ще один тип метаданих на вебсторінці. Раніше власникам сайту приходилося самостійно вказувати ключові слова на сайті для врахування пошуковими системами при індексації сайту.

Але в 2009 році в **Google** заявили, що пошукова система не враховуватиме **keywords**.

Зараз пошукові системи самостійно знаходять і визначають ключові слова, скануючи контент вашої вебсторінки з урахуванням заголовків (H1-H6), тегу **description**, тексту посилань.

Рекомендовано приділяти увагу найбільш впливовим тегам – **title**, **description**.

Велика кількість вихідних посилань на сторінці може негативно вплинути на ранжування сайту.

ДОВІДКА

Пошукові системи оцінюють сайт за більш ніж 800 факторами.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Відредагуйте вашу вебсторінку безпосередньо у форматі HTML, додавши необхідні метадані.

Назва вашої сторінки

Теги заголовка містять усі метадані вашої вебсторінки

Атрибут **charset** тегу **<meta>** визначає кодування символів HTML-документа. За замовчуванням має значення UTF-8 – універсальне кодування символів, яке підтримує багато мов і символів.

```
<!DOCTYPE html>
<html lang='uk'>
<head>
    <meta charset="UTF-8">
    <title>Моя перша вебсторінка</title>
    <meta name='keywords' content='Персональна сторінка Оленки, Оленка Учениця'>
    <meta name='description' content='Персональна сторінка Оленки Учениці'>
    <link href='style.css' rel='stylesheet' type='text/css'>
</head>

<body>
    <h1>Це особиста сторінка Оленки</h1>
    <p style="font-size: 20px;">Ласкаво просимо на мою особисту вебсторінку.</p>
    <p style="font-family: courier;">Ласкаво просимо на мою особисту вебсторінку.</p>
    <p style="font-size:20px; font-family: courier ">Ласкаво просимо на мою особисту вебсторінку.</p>


    <nav>
        <a href="index.html">Home</a>|
        <a href="art.html">Art</a>|
        <a href="contact.html">Contact me</a>
    </nav>
    <img src="Снимок.JPG" width="170" height="300" alt="Робот"/>
    <audio controls>
        <source src="Etude.mp3" type="audio/mp3">
    </audio>
</body>
</html>
```

Ключові слова, які ви визначили, як приклад

Опис вашої вебсторінки

Пошукові системи намагаються ранжувати результати пошукового запиту. Найбільш релевантні та якісні вебсторінки з'являються у верхній частині результатів пошуку. Релевантність визначається головним чином тим, як часто й де зустрічаються слова із запиту у змісті вебсторінки, а якість – тим, скільки інших сайтів містять посилання на цю вебсторінку.

На ранжування сайту впливають й текстові фактори.

Текст повинен буди корисний для відвідувачів, мати цінність, бути унікальним. Оптимальний варіант – більш ніж 90 % унікальності.

Форматуйте текст – виділяйте списки, таблиці, блоки уваги, підзаголовки.

Додайте ключові слова та фрази у текст. Перевіряйте тексти на помилки перед публікацією. Пошукові системи враховують орфографічні помилки та пунктуацію, та знижують позицію сайта у видачі.

Оновлюйте сайт, доповнюйте новими та цікавими матеріалами та не забувайте й про технічну складову.

Швидкість завантаження сторінки має велике значення. Оптимальний час завантаження сторінки – до 3 сек.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Технічне завдання з розробки вебсайту

Для розробки сайту потрібне технічне завдання. **Технічне завдання** (ТЗ) для розробника сайтів є ключовим документом, що визначає всі вимоги та специфікації проєкту перед початком розробки програми. Воно містить детальний опис функцій та особливостей архітектури, інтерфейсів та інших важливих аспектів сайту. Наприклад, чи матиме сайт можливість покупки, чи буде функціонал додавання у вибране або чи міститиме сайт калькулятор. Завдяки йому розробник розуміє як повинен виглядати та працювати сайт за задумом замовника.

Важливим аспектом створення сайту є розуміння цільової авдиторії. **Цільова авдиторія сайту** – це група людей, які є основною метою або цільовою групою для конкретного вебсайту. Визначення цільової авдиторії для сайту є ключовим етапом в процесі розробки та маркетингу, оскільки воно визначає хто та для яких цілей буде використовувати сайт, а також допоможе зрозуміти «больові» та проблемні точки цих користувачів.

Аналіз відвідування сайту

Робота над сайтом не завершується навіть тоді, коли він запущений та працює. Однією з таких робіт є аналіз відвідування сайту. Це робиться для того, щоб розуміти як користувачі поводяться на сайті, чи подобається їм його вміст, чи досягають вони своїх цілей тощо. В інтернеті є безліч інструментів, що можуть допомогти це зробити. **Google Analytics 4** допоможе переглянути дані відвідування, а **HotJar** або **Clarity** наддасть візуалізацію цих відвідувань за допомогою теплових карт або записів екранів.

Введіть пошуковий запит...

Рядків на сторінці: 10 1–4 з 4

	Категорія пристрою	Користувачі	Нові користувачі	Сеанси з взаємодією	Коефіцієнт взаємодії	Кількість сеансів взаємодії на користувача	Сер
		43 604 100% від загальної кількості	28 367 100% від загальної кількості	104 004 100% від загального підсумку	93,54% Сер. 0%	2,43 Сер. 0%	9 хв
1	mobile	40 389	26 099	99 568	93,51%	2,47	9 хв
2	tablet	2 348	1 434	7 125	92,24%	3,03	10 хв
3	desktop	913	725	742	67,83%	0,81	
4	smart tv	4	4	3	75%	0,75	

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Визначення цільової авдиторії вебсайту

Для визначення та розуміння цільової авдиторії сайту використовують декілька різних методів. Один з них це прописати **Портрет цільового користувача** – детальний опис представника цільової авдиторії, для якої розробляється продукт чи послуга. У портреті відмічають загальний опис людини, його хобі, звички, сім'ю та домашніх улюбленців. Не менш важливо вказати цілі та розчарування користувача, щоб зрозуміти що додати на сайт або прибрати.

Персона

Катерина
32 роки
Харків
Домогосподарка

Опис

Її основні задачі дня – це займатись домом та дітьми.

Катерина любить, коли все охайно та гарно. Останнім часом зацікавилась нутриціологією та здоровим способом життя.

Почала розбиратись у темі та намагається готувати лише поживну та якісну їжу. Також шукає способи економії свого часу на приготування.

Цілі

- Готувати якісну їжу
- Мати здорове тіло
- Піклуватись про дітей та чоловіка

Розчарування

- Не задовольняє якість продуктів у супермаркеті
- Не знає, де купити якісні продукти
- Не знає, які заклади готують з фермерських продуктів

Персона

Михайло
27 років
Львів
Підприємець

Опис

Його основна задача дня – це розвивати власний бізнес.

Михайло любить, коли все систематизовано та упорядковано. Цікавить здоровий спосіб життя.

Відвідує фітнес-центр та намагається готувати лише поживну та якісну їжу. Шукає способи швидко та просто готувати корисну їжу.

Цілі

- Мати здорове тіло
- Готувати корисну їжу
- Готувати просто та швидко

Розчарування

- Не задовольняє якість продуктів у супермаркеті
- Не встигає купити фермерські продукти на ринку
- Не знає, які заклади готують з фермерських продуктів

Ще одним методом для вивчення проблем та думок користувача є карта емпатії. Вона допомагає відобразити та зрозуміти потреби, бажання, проблеми, мотивації та переживання користувачів. Карта емпатії зазвичай представляється у вигляді великого аркуша або дошки, розділеної на секції, які представляють різні аспекти життя цільового користувача: що говорить, що думає, що робить та що відчуває. Часто карту емпатії розробляють після первинних інтерв'ю з поточними або майбутніми користувачами.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Карта емпатії

Говорить

Як я можу отримати здорову та якісну їжу, щоб зекономити час на приготуванні?

Де я можу знайти здорові продукти з ферми?

Де знайти магазини з фермерськими продуктами?

Як я можу знайти ресторани зі здоровою їжею?

Думає

Тренд здорового способу життя у соцмережах

Як зекономити час на приготуванні якісної та здорової їжі?

Хочу мати здорове тіло

Як мені перевірити чи готують у цьому закладі страви з фермерських продуктів?

Як мені перевірити чи продаються в цьому магазині фермерські товари?

Хочу, щоб мої діти були здорові

Робить

Шукаю якісні продукти, без пестицидів

Готую корисні та збалансовані страви

Уважно читаю склад продуктів

Виконую фізичні вправи

Переглядаю відео та статті про нутриціологію

Відчуває

Зневіру у продуктах з супермаркету

Втрату часу на приготування їжі

Мотивацію готувати здорову їжу

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Щоб зрозуміти проблеми та цілі користувачів. Використовують **User Stories** (Користувацькі історії). Це коротке речення, що будується за певною формулою:

Я ***тип користувача***, хочу ***дія користувача***,
щоб ***ціль користувача***.

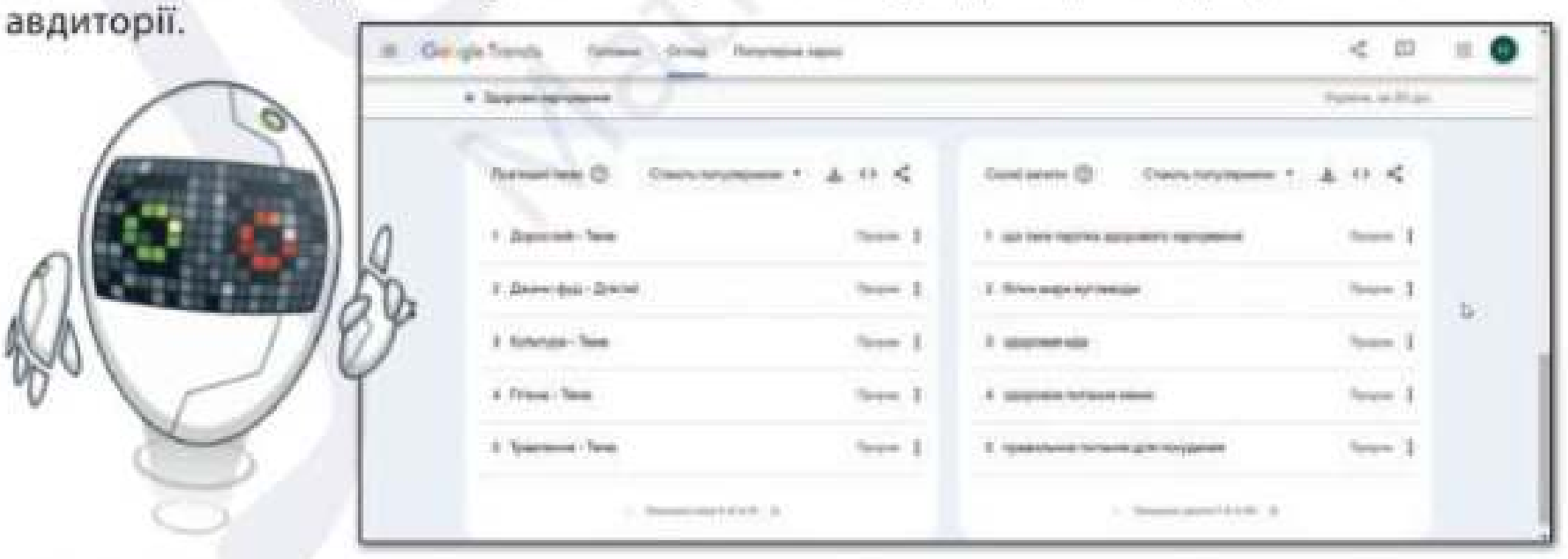
Таким чином команда, що працює над сайтом, чітко бачить основні запити та цілі користувача та підлаштовує сайт, що розробляється, під них. Користувацькі історії часто також пишуть після інтерв'ю з користувачами.

User Stories

Я як домогосподарка хочу знайти магазини з фермерськими продуктами, щоб покращувати своє здоров'я та своєї сім'ї. Олена	Я як підприємиця хочу знати ресторани, де подаються страви виключно з фермерських продуктів, щоб економити свій час та підтримувати своє здоров'я. Вікторія	Я як спортсмен хочу знати місця зі здоровою їжею, щоб не перейматись за прийоми їжі поки готуюсь до змагань. Олег	Я як блогерка нутриціолог хочу знати місця з якісною їжею, щоб розповісти у блозі та створювати контент. Діана

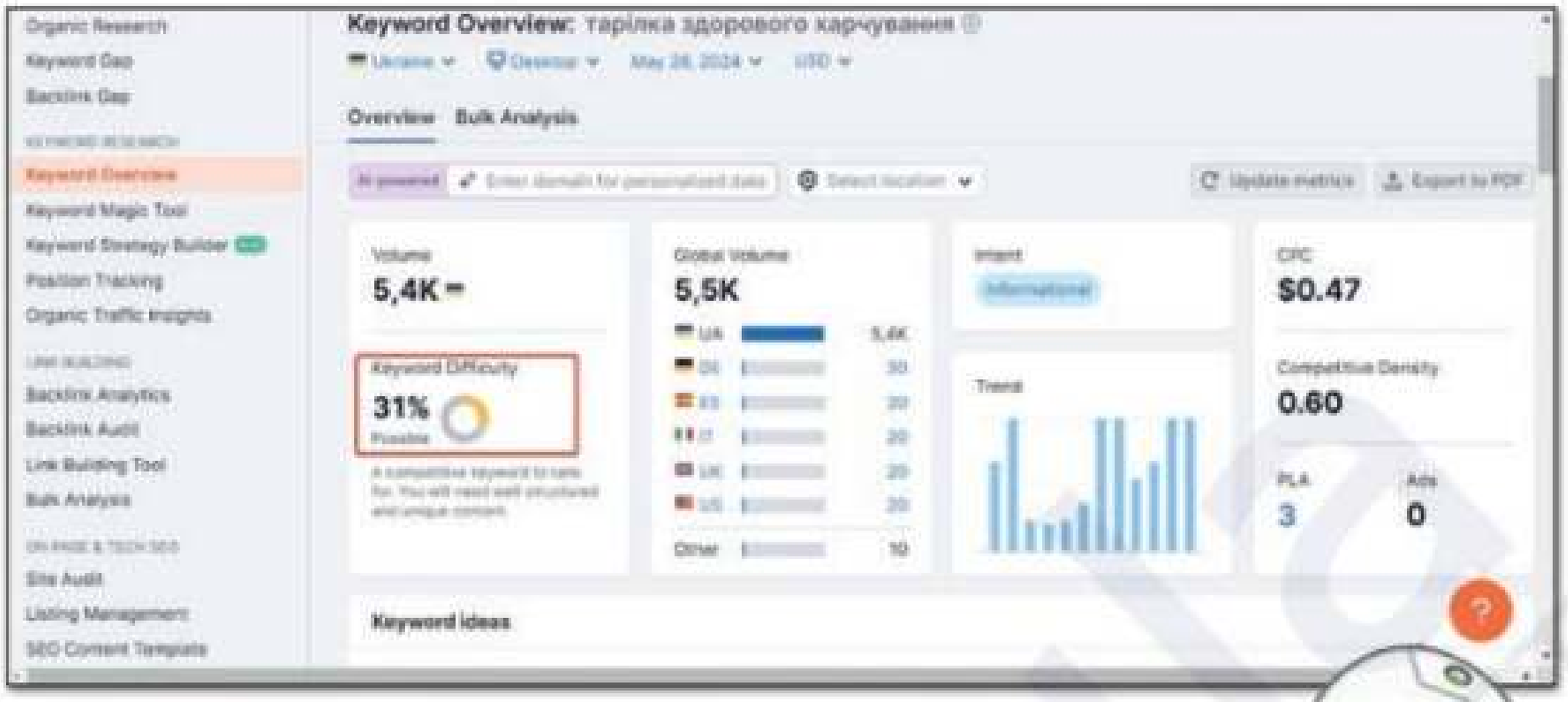
Вибір ключових слів вебсайту та дослідження їх конкурентності

Вибір ключових слів для вебсайту та дослідження їх конкурентності є важливим етапом в процесі оптимізації пошукових систем (SEO) і розробки контентної стратегії. Ключові слова допомагають вебсайту з'являтися у результатах пошуку, коли користувачі шукають інформацію, товари чи послуги, пов'язані з певним бізнесом. Вони також допомагають визначити теми та напрямки контенту для сайту, забезпечуючи більшу відповідність до цільової авдиторії.



Щоб підібрати ключові слова, спочатку потрібно провести дослідження цільової авдиторії, бізнесу та ринку. Використовуючи **Google Trends**, **SEMrush**, **Ahrefs** або **Ubersuggest** можна знайти потенційні слова та фрази, пов'язані з потрібно тематикою. При цьому потрібно звертати увагу на обсяг пошуку, релевантність та потенційний трафік.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.



Далі потрібно перевірити конкурентність вибраних слів та фраз. Можна використовувати наведені вище інструменти. Потрібно оцінити рівень конкурентності на основі показників, таких як кількість сайтів, які конкурують за це слово, та їхній авторитет. Ключові слова з помірним рівнем конкурентності можуть забезпечити більше можливостей для підняття сайту в пошукових результатах.

Збір відгуків відвідувачів сайтів

Після створення сайту важливо отримувати зворотній відгук. Таким чином власник сайту зможе орієнтуватись у тому що подобається користувачам, що не виходить та чого не вистачає. Часто це роблять за допомогою опитувань, що вбудовуються на сайт та виникають на різних сторінках. Як приклад, може бути коротке опитування, де користувач відповідає на питання оцінкою від 1 до 5, а у кінці має можливість розгорнуто описати. Крім того, таке опитування є легким та дешевим інструментом, що заміняє користувацькі дослідження.

Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

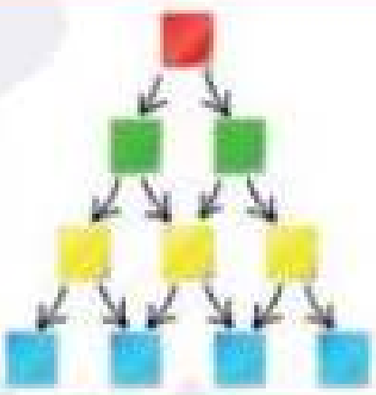
© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 6

Координація роботи команди. Система контролю версій

Створення сайту починається задовго до першого рядка коду, він починається з ідеї. Зазвичай це мета – те що ви плануєте робити зі своїм сайтом: продавати товари, надавати послуги чи інше. Вибір цільової авдиторії, структура сайту, ескізи сторінок сайту тощо. Фактично, кодування – це майже фінальний крок у процесі.

Більшістю цих робіт займається Product-owner (власник продукту), він зацікавлений у тому щоб сайт мав успіх. Він пише документацію для проєкту, як-от структуру сайту, контент-план (наповнення сайту та коли воно буде розміщено). Product-owner аналізує які можливості надає сайт конкурентів, щоб зробити власний сайт технологічнішим. Він допомагає дизайнерам – бере участь у створені концепції дизайну, визначає основні елементи інтерфейсу. І наприкінці допомагає розробникам – пояснює ТЗ, відповідає на їх питання та контролює процес розробки.



Ієрархічна структура сайту схожа на дерево. Головна сторінка – стовбур дерева, категорії товарів – великі гілки, від яких йдуть гілки поменше – підкатегорії. Листочки – це по суті сторінки товару.

Роботу над сайтом ділять між собою декілька фахівців: дизайнер, верстальник, програміст, тестувальник, контент-менеджери та інші.

Дизайнер створює прототип сайту, малює іконки, підбирає шрифти, кольорову палітру сайту. До того ж дизайн повинен відповідати цільовій авдиторії, на яку він буде розрахований.

Верстальник (відомий як front-end розробник) – це програміст, який перетворює узгоджений дизайн у реальний сайт, який працює. Саме він створює інтерактивні елементи: кнопки, форми, спадні меню тощо, а також стилізує їх під дизайн сайту. Він намагається зробити сайт, який не буде перевантажувати систему та швидко завантажуватися на різних цифрових пристроях користувачів.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Програміст (відомий як back-end розробник) – це фахівець, який займається «зворотньою частиною» сайту. Тобто розробляє взаємодію користувача з сервером, на якому розташовується сайт. Працює з базою даних, на яких знаходяться товари чи дані клієнтів, оптимізує час відповіді між сайтом і користувачем. І найголовніше, він працює над безпекою як самого сайту, так й даних, які там зберігаються.

Тестувальник – людина, що відповідає за перевірку коректності роботи сайту. Його основною метою є пошук помилок і передача їх програмістам, для виправлення. Він перевіряє чи відповідає сайт вимогам наданих Product-owner. У роботі використовує спеціальні програми для тестування сайту та складання некоректних звітів про його роботу.

Система контролю версій

Створення сайту – складний процес із залученням великої кількості людей. Як уникнути хаосу, коли так багато людей працює над проєктом?

На допомогу приходять **системи контролю версій** (**СКВ** або **VCS**) – це інструмент для відстеження змін у файлах проєкту та керування різними версіями коду. Простішим прикладом СКВ – історія редагування статей на Вікіпедії. Вони є незамінними при роботі з сайтом та надають наступні переваги:

- допомагають зберігати та відстежувати зміни, які внесли у проєкт, точний час та автора змін;
- дають можливість повернути попередню версію, якщо щось пішло не так;
- надають можливість розгалуження – вести розробку у різних напрямах;
- дають можливість командної роботи над одним файлом одночасно;
- допомагають документувати поетапні зміни у проєкті та коментувати їх;
- зберігають історію усіх змін.

Контент-менеджер – людина, що наповнює наш сайт текстовою та графічною інформацією. Зазвичай робить це через запис адміністратора сайту, так звану «адмінку» або окремий програмний продукт, через який завантажує матеріали на сайт.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Види СКВ

- розподілена
- централізована
- локальна

У централізованій системі усі оновлення файлів проходять через серверне сховище, однак ви можете скопіювати файли зі сховища на свій пристрій. До того ж можна заблокувати файли та обмежити доступ певних осіб до файлу. Єдиним недоліком цієї системи є потреба у постійному інтернет-з'єднанні з сервером для оновлення та збереження змін. Прикладами можна назвати Subversion та Azure DevOps Server.

Розподілена СКВ дає можливість кожному програмісту працювати зі своєю локальною копією репозиторію, не потрібно підтримувати постійне інтернет-з'єднання. Такий вид СКВ має певні недоліки – відсутність контролю доступу до файлів, можливості блокувати файли, самі файли займають набагато більше дискового простору що на сервері, то й на локальному пристрої. Представниками такого типу можна назвати Git, Bitbucket та GitLab.

Репозиторій *(англ. repository) – місце, де зберігаються та підтримуються дані. В репозиторії дані зберігаються у вигляді файлів, доступних для подальшого завантаження в мережі.*

Локальна СКВ зберігається тільки на особистому комп'ютері, де кожна зміна файлу є новою версією. Недоліки такого способу – неможливість роботи у команді над одним проєктом, при пошкодженні комп'ютера, всі дані проєкту можуть бути втрачені незворотньо.

Використання системи контролю версій

Скористаємося **Git** – розподіленою системою контролю версій, яка допомагає відстежувати зміни в коді та працювати над проєктами в команді. Основним інтерфейсом для роботи з Git є консоль або термінал, що може бути незручним для новачків. Тому рекомендують встановлювати додаткову програму з графічним інтерфейсом, яка спрощує роботу з Git.

ІСТОРІЯ

Сервіс GitHub був розроблений Крісом Ванстратом, П. Д. Хаєттом, Томом Престон-Вернером і Скоттом Чейконом з використанням Ruby on Rails і запущений у лютому 2008 року. Компанія GitHub, Inc. існує з 2007 року та знаходиться в Сан-Франциско.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Щоб встановити Git на пристрій:

- перейдіть за посиланням **https://git-scm.com/**;
- оберіть версію під вашу ОС (1) та її розрядність (2);
- завантажте програму та запустить інсталяцію;
- ознайомтеся з ліцензійною угодою;
- клацніть **Next** (3);
- визначте налаштування, натискаючи **Next**;
- у вікні вибору редактору оберіть **Notepad** (4);
- у наступному вікні оберіть (5);
- далі знов клацніть **Next**, щоб прийняти налаштування за замовчуванням;
- завершіть інсталяцію, клацнувши **Finish** (6).

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Налаштування

Після встановлення Git, слід його налаштувати, щоб при створенні **commit** відображалося ім'я та e-mail автора змін.

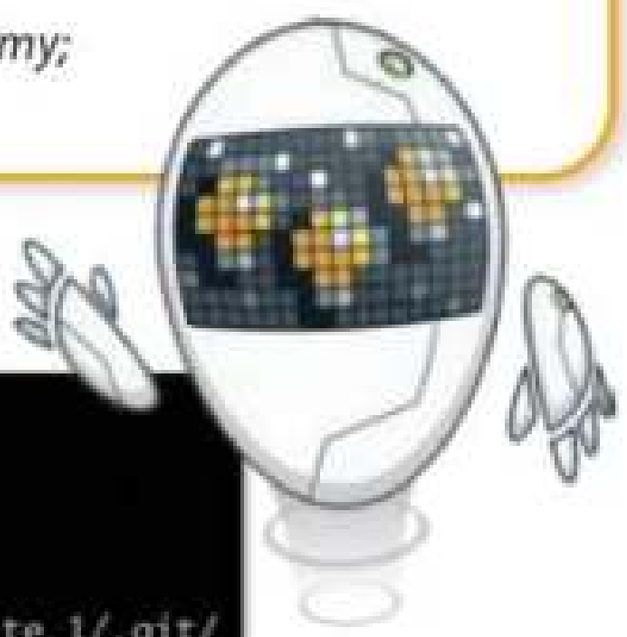
Щоб налаштувати Git:

- клацніть меню **Пуск** та запустіть **Git Bash** (1);
- завантажте програму та запустіть інсталяцію;
- встановіть ім'я користувача (2);
- встановіть e-mail користувача (3).

```
MINGW64:/c/Users/User
User@Z24 MINGW64 ~                                          (2)
$ git config --global user.name "Olenka Ucenica"
User@Z24 MINGW64 ~                                          (3)
$ git config --global user.email "olenka.ucenica@gmail.com"
```

Commit* у *Git *– знімок стану проєкту в певний момент часу, він зберігає інформацію:*

- *унікальний ідентифікатор (хеш);*
- *автор коміту;*
- *дата та час;*
- *повідомлення коміту;*
- *зміни в файлах.*



Створення репозиторію

(1)

```
User@Z24 MINGW64 ~
$ cd /c/Users/User/Documents/Site_1

User@Z24 MINGW64 ~/Documents/Site_1
$ git init
Initialized empty Git repository in C:/Users/User/Documents/Site_1/.git/
```

```
User@Z24 MINGW64 ~/Documents/Site_1 (main)
$ git add first.py

User@Z24 MINGW64 ~/Documents/Site_1 (main)
$ git commit -m 'Перша версія'
[main (root-commit) 7bc4414] Перша версія
 1 file changed, 6 insertions(+)
 create mode 100644 first.py
```

Щоб налаштувати Git локально:

- перейдіть до теки з вашим проєктом (1);
- ініціалізуйте репозиторій (2);
- додайте файли, які плануєте контролювати (3);
- додайте **commit** (4).

© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Робота з Git

Створюйте commit у таких випадках

- Додано новий функціонал
- Зроблено значні зміни у верстці
- Виправлено помилку
- Завершення робочого дня

Це допоможе підтримувати чистоту у вашій історії змін.

Візуальний інтерфейс

Для полегшення роботи з Git можна встановити графічний інтерфейс: **GitHub Desktop**, **Sourcetree**, **GitKraken** тощо. Перейдіть за посиланням

git-scm.com/downloads/guis

та оберіть потрібну ОС та бажаний **GUI**.

Завантаження та оновлення проєкту

Щоб отримати зміни з віддаленого репозиторію:

- перейдіть до основної гілки командою git checkout master;

```
$ git checkout edited
Switched to a new branch 'edited'
Branch 'edited' set up to track remote branch 'edited' from 'origin'.
```

- оновіть локальну версію командою git pull origin master.

```
$ git pull origin master
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 655 bytes | 3.00 KiB/s, done.
From https://github.com/akashadr/Git
 * branch            master     -> FETCH_HEAD
   5a457ea..4913939  master     -> origin/master
Merge made by the 'ort' strategy.
 New.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 New.txt
```

ПІДКАЗКА

Якщо ви використуєте **Visual Studio Code** (**VS Code**), у вкладці Git натисніть кнопку для публікації на **GitHub** та оберіть опцію для створення публічного репозиторію.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Основні команди Git

Створення нової гілки:

git branch <назва_гілки>
git checkout -b <назва_гілки>

Перемикання між гілками:

git checkout <назва_гілки>

Злиття гілки в основну:

git merge <назва_гілки>

```
$ git merge next-five-even-odd
Updating a8d9e6b..3c477e1
Fast-forward
 Even-numbers.txt | 10 ++++++++++-
 Odd-Number.txt   | 10 ++++++++++-
 2 files changed, 18 insertions(+), 2 deletions(-)
```

Відправка змін на GitHub:

git push origin <назва_гілки>

Клонування репозиторію:

git clone <адреса_репозиторію>

```
$git clone https://github.com/vertexacademy/first-repo.git newRepo
Cloning into 'newRepo'...
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 22 (delta 3), reused 21 (delta 2), pack-reused 0
Unpacking objects: 100% (22/22), done.
$
```

Інструменти координації роботи команди

Нам відомо, як завантажувати та працювати разом над одним проєктом. Однак, як правильно розрахувати час та кількість людей, які потрібні для виконання певного завдання? Хто буде займатися підбором кольорів вашого сайту? Програміст? А який? А якщо він зараз зайнятий роботою над іншим елементом дизайну?

На допомогу нам приходять спеціальні програми для координації роботи в команді та методології.

Методологія розробки програмного забезпечення включає в себе принципи, техніки та процедури, які використовують фахівці, що працюють у цій сфері. Найбільш популярні методи відрізняються один від одного не лише своїм способом виконання, а й вимагають використання різних процесів і навіть програмного забезпечення для управління проєктами.

Розглянемо найпопулярніші рішення на сьогодні – це Scrum та Kanban.

Інструменти координації роботи команди – *програми та сервіси, які допомагають розподілити роботу у команді, спілкуватися та відстежувати процес виконання завдань.*

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Scrum

Scrum – це як гра в команди, де кожен гравець знає свою роль. Ми розбиваємо роботу на короткі етапи, які називають спринтами. Наприкінці кожного спринту оцінюємо, що вийшло, та плануємо наступні кроки. Це допомагає всім залишатися на одній хвилі й швидко реагувати на зміни. Однак слід пам'ятати, що Scrum працює тільки для команд з кількістю людей не більш ніж 10 людей.

Kanban

Kanban – це організація списку справ, де ми бачимо, що робимо, над чим працюємо та що вже закінчили.
Ми використовуємо картки, щоб візуально показати, на якому етапі кожне завдання.
Це допомагає зрозуміти, що потрібно зробити в першу чергу. До того ж він підходить для маленьких, середніх та навіть великих організацій.

Отже ви обрали методологію, за якою будете створювати свій сайт. Перейдемо до готових рішень з управління вашим проєктом.

Для Kanban чудово підійде **Trello** або звичайна дошка та стікери, яка розділена на три або більше колонок. Основними є колонки **To Do**, **Doing** та **Done**. Можете додати свої колонки, де створите потрібну саме вам колонку, на кшталт **Тестування функцій** або ж **Перевірене завдання.**

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Однак для Scrum треба брати вже більш складну програму, як **Jira** або **FlexiProject**. У них вам доведеться поставити всі завдання для кожного учасника вашої команди, виділити час, який потрібен на виконання завдань, продумати, яке завдання йде по черзі, яку кількість часу буде тривати спринт, запланувати час на щоденні зустрічі, щоб відслідковувати прогрес по кожному завданню. Це займе у вас велику кількість часу, щоб це все налаштувати, однак цей досвід допоможе краще бачити проблеми, з якими стикається проєкт, краще розуміти можливості та навички інших людей, а також допоможе якісно планувати час.

Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

https://formula.education/online-ict-8-klas/

Інші платформи

Проєкт

Підсумки

Глосарій

© Binary Logic та © Видавництво «Пітгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

6. Дизайн-мислення

Уявіть! Вам треба зробити популярним ваш блог, зробити його брендом. Що потрібно зробити, щоб він став впізнаваним, як зробити його привабливим для потенційних відвідувачів? Визначте цінність, місію та візію бренду. Сформуйте корпоративний стиль. За допомогою технологій генерування ідей розробіть власний проєкт та контент-план просування продукту. Саме в цьому розділі ви дізнаєтеся, як це зробити.

У цьому розділі ви навчитеся:

- використовувати різноманітні інструменти для створення фірмового стилю;
- розрізняти логотипи за видами;
- розуміти психологію кольору;
- створювати інформаційні продукти бренду;
- розвивати власну креативність;
- використовувати дизайн-мислення у повсякденних завданнях;
- складати контент-план.

У цьому розділі ви дізнаєтеся:

- що таке бренд, брендбук;
- що таке цінність, місія та візія бренду;
- як розробити фірмовий стиль;
- як спроєктувати інформаційний продукт;
- які технології використовують для генерації ідей;
- що таке дизайн-мислення;
- що таке контент-план.

Програмне забезпечення:

- Figma;
- Adobe XD;
- Sketch;
- Notion;
- Trello.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 1
Бренд і фірмовий стиль

Наш мозок влаштований так, що основний канал отримання інформації – зір. Тому й візуальне представлення інформації дуже важливо.

Результати досліджень показують, що при використанні візуальної інформації, продуктивність людини на 17 % вища. Ми краще запам'ятовуємо такі деталі, які б не привернули уваги при читанні просто тексту.

Саме завдяки таким якостям ми швидко впізнаємо образи, і на цьому будується необхідність створення бренду. Це може бути ваш особистий бренд або бренд вашого закладу освіти чи компанії. Це як шкільна форма – одразу видно, хто звідки, будує командний дух.

Що таке бренд

Бренд – набір асоціацій, емоцій та вражень, які виникають у людей при згадці про певну компанію чи продукт. Це не лише логотип чи назва, а й певний посил, який компанія транслює своїм клієнтам. Уявіть собі компанію **Nike**. Коли ви чуєте цю назву, у вас одразу виникають певні асоціації: спортивний одяг, якісне взуття, сучасний дизайн і відомий слоган **Just Do It**. Всі ці елементи формують бренд, який асоціюється зі спортом і активним способом життя. Бренд передає певні цінності, місію та візію.

Цінності бренду – це основні принципи компанії, які формують її унікальність та закладають основу для створення емоційного зв'язку з клієнтами.

Google	Apple	Нова пошта
Доброчесність	Інновації	Оперативність

Місія бренду – це основна мета компанії, яку вона транслює світу. Вона відповідає на питання: «Чому ми це робимо?», «Для чого ми це робимо?» та «Чому саме наш продукт клієнти мають вибирати?». Місія допомагає сформувати шлях до кінцевого результату, який прагне отримати компанія.

IKEA	Vodafone	Сільпо
Поліпшення повсякденного життя кожного	Об'єднувати людей через інноваційні телекомунікаційні рішення	Робити покупки приємними та доступними для всіх

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Візія бренду – це те, до чого прагне компанія в майбутньому, якою вона хоче стати.

Київстар	Disney	Google
Створення безперервного зв'язку між людьми та можливість ділитися важливими моментами Бренд прагне до розвитку інновацій у телекомунікаціях та наданні послуг, які полегшують життя.	**Робити людей щасливими** Бренд створює казкові парки з атракціонами, мультфільми, відео, позитивні історії.	**Організувати світову інформацію та зробити її доступною для всіх** Бренд створює інноваційні програми, як-от пошуковик, карти, перекладач.

Що таке фірмовий стиль

Фірмовий або **корпоративний стиль** – це набір візуальних та комунікаційних елементів, завдяки яким формується унікальність та ідентичність компанії. Основна його мета це створення впізнаваності бренду, передачі його цінностей, місії та візії. Фірмовий стиль є важливим інструментом, за допомогою якого компанія виділяється серед конкурентів і забезпечує емоційний зв'язок зі споживачами.

У сприйнятті бренду зображення відіграють дуже важливу роль. Як ви пам'ятаєте, саме через зір ми отримуємо більшість інформації. Зображення – це найбільш зручний спосіб візуально передати цінності, ідеї та атмосферу, яку хоче донести компанія до свого споживача.

ПІДКАЗКА

При створенні бренду поміркуйте, які емоції та асоціації ви хочете, щоб викликав ваш продукт.

Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 2
Елементи стилю та брендбук

Для створення фірмового стилю використовують брендбук. Це збірник, в якому зібрані всі елементи бренду та правила, як ними користуватися, щоб створити єдиний та цілісний образ компанії.

Основні елементи фірмового стилю

Логотип – графічний елемент, який представляє компанію. Його основні завдання – бути простим, запам'ятовуваним і відображати суть бренду. Логотипи діляться на види:

Типографічні або текстові логотипи

Ці логотипи складаються з тексту, зазвичай це назва компанії, і виготовленні завдяки використанню різних шрифтів та стилів написання. Частіше за все вони базуються на типографії та стилізації букв.

Графічні логотипи

В цьому випадку логотип складається з графічного елемента або символу, який представляє бренд. Логотип може бути простим або складним, але він завжди повинен бути зрозумілим та запам'ятовуваним.

Комбіновані логотипи

Ці логотипи поєднують текст і графічний елемент. Це найбільш розповсюджений тип, бо він дозволяє бренду використовувати як назву, так і символ, що забезпечує йому більшу впізнаванність.

Логотипи-маскоти

При створенні цього логотипу використовують персонажа або графічне зображення тварини, що надає індивідуальності та характеру. Маскоти зазвичай використовуються для створення емоційного зв'язку з авдиторією, а також їх зручно використовувати в анімованих рекламних кампаніях для передання цінності або ідеї бренду через свої образи та поведінку.

Логотипи-символи

В цьому логотипі основну роль грають літери, зазвичай це ініціали або перші букви назви компанії, абревіатури. Цей логотип простий у створенні, але забезпечує собі впізнаваность бренду.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Абстрактний логотип

Для створення цього логотипу використовують геометричні форми, кольори та текстури для створення унікального і незвичного зображення, що не має конкретного зображення предметів чи істот. Основна особливість абстрактного логотипу – це надання свободи для уяви та асоціацій у споживача через відсутність прив'язки до конкретних об'єктів.

Логотип-емблема

Він поєднує в собі текст та графічний елемент у єдину композицію. Основна відмінність від комбінованого логотипу це те що текст не просто стоїть десь поряд з символом, а вмонтований в графічний елемент або обрамлення, таке як круг, щит або інша форма. Такі логотипи часто асоціюються з традиціями, офіційністю та надійністю.

Фірмові кольори – це кольори, які використовує бренд для своєї візуальної ідентичності, реклами та комунікації зі споживачами. Кольори мають таку ж важливість як і зображення, бо так само сприймаються людьми через зір. Також вони здатні легко та швидко пробудити в підсвідомості потрібні асоціації у споживачів що забезпечує впізнаваність та вирізняє продукт серед конкурентів та підкреслює його унікальність. А ще це один із успішних психологічних інструментів, який використовують в маркетингу.

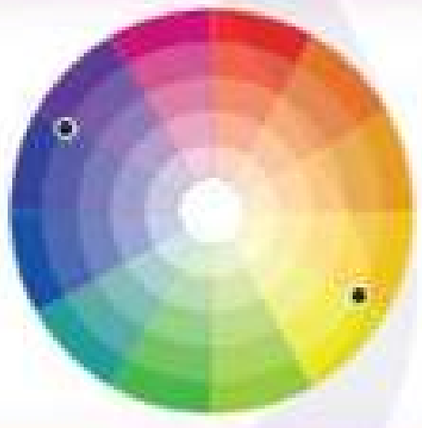
Коли постає завдання обирати кольори для фірмового стилю, важливо, щоб вони гармонійно поєднувалися один з одним, оскільки кольори впливають на сприйняття бренду та його імідж. Є кілька основних принципів, як обрати кольори, які добре комбінуються.

***Маркетинг** – комплексна діяльність, спрямована на вивчення ринку, визначення потреб споживачів, розроблення, просування та реалізацію товарів і послуг, що задовольняють ці потреби, з метою отримання прибутку.*

Колірне коло

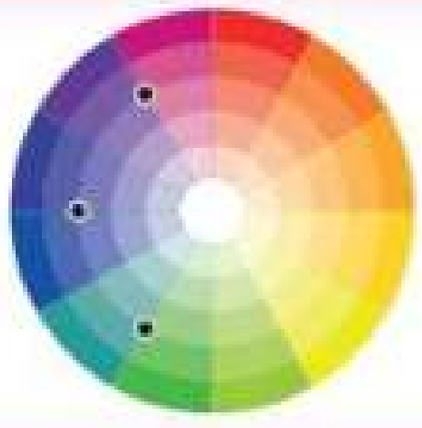
Усі кольори можна комбінувати на основі колірного кола. Це самий зручний і легкий спосіб перевірити кольори на сумісність.

Комплементарні кольори



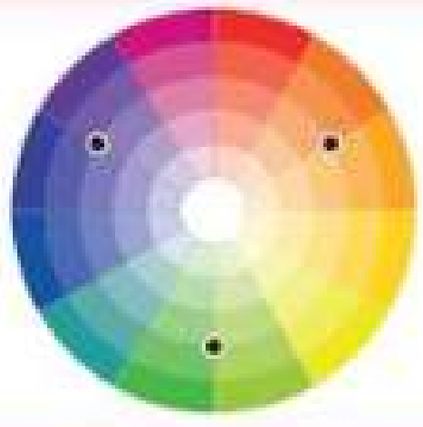
Це кольори, які знаходяться навпроти один одного на колірному колі. Вони створюють сильний контраст і привертають увагу.

Аналогічні кольори



Це кольори, які знаходяться поруч один з одним на колірному колі. Вони створюють гармонійний вигляд.

Тріадні кольори



Три кольори, рівномірно розташовані на колірному колі. Вони забезпечують баланс і динамічність.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Значну роль у формуванні першого враження про бренд чи продукт відіграють кольори.

Психологія кольору

Різні кольори викликають різні емоції. Наприклад, червоний може символізувати енергію або пристрасть, тоді як синій асоціюється з довірою та спокоєм. Вибір кольору відповідає повідомленню, яке хоче донести бренд.

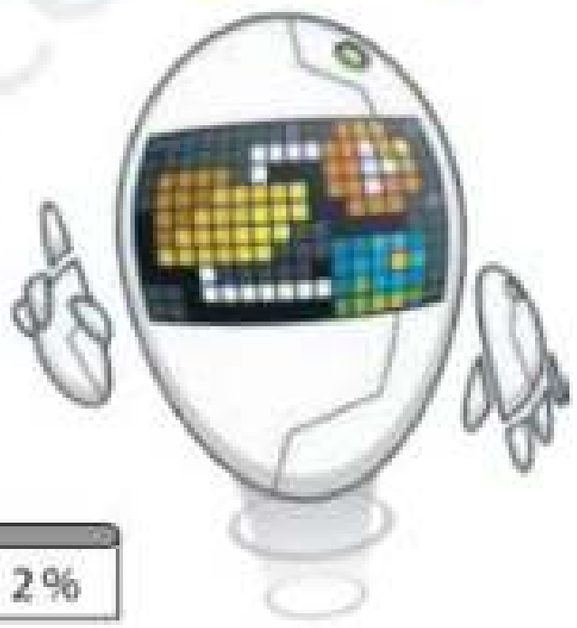
Використання нейтральних кольорів

Додавання нейтральних кольорів (білий, чорний, сірий) допоможе збалансувати більш яскраві кольори і надати дизайну професійний вигляд.

Асоціації з кольором частково визначаються нашими відчуттями, мовою, об'єктами або формами та характеристиками особистості, – стверджує автор книги «Психологія кольору та кольоротерапія» Фабер Біррен. Колір майже автоматично передає настрої, пов'язані з людськими почуттями та нашим психічним складом.

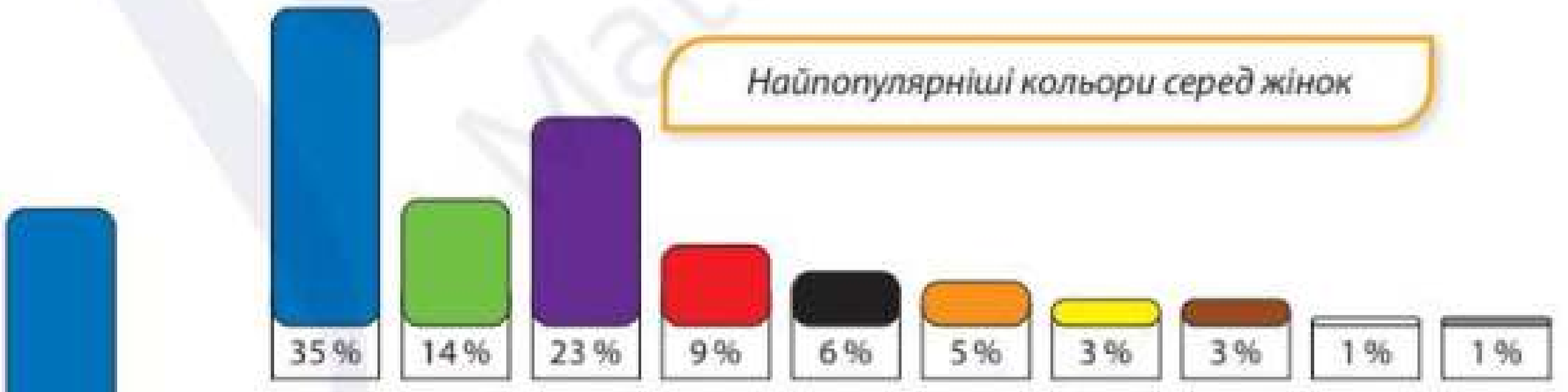
Джо Халлок провів дослідження, де з'ясував цікаві факти.

Приклади фірмових кольорів

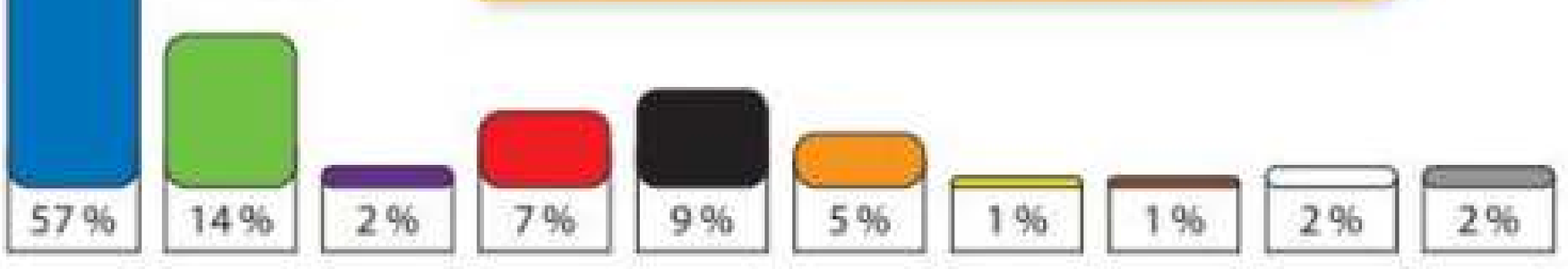


Найпопулярніші кольори у світі

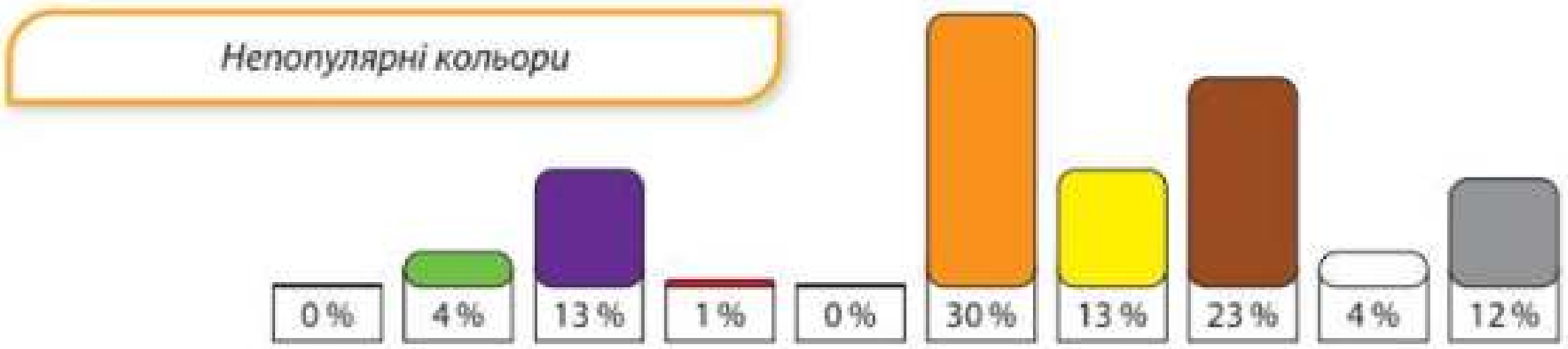
Найпопулярніші кольори серед жінок



Найпопулярніші кольори серед чоловіків



© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.



Але ніхто не відміняє неочікуваний ефект від використання нестандартних кольорів, який допоможе допомогти виділитися серед конкурентів та збільшити обізнаність бренду.

Інші елементи стилю

Шрифти – це важливий елемент фірмового стилю, який також несе в собі функцію вирізнити бренд серед інших, транслювати ідентичність та впливати на весь зовнішній вигляд. Розрізняють такі види шрифтів:

Times New Roman
Georgia

Антиква – це тип шрифта, який має серіфи (маленькі виступи на кінцях літер або «засічки»), характеризується класичними формами. Приклади: Times New Roman, Georgia.

Arial
Helvetica

Гротеск – це тип шрифта, який характеризується своєю простотою форм без виступів (серіфів або «засічок») на кінцях літер. Такі шрифти мають сучасний і мінімалістський вигляд, що робить їх популярними в графічному та вебдизайні. Приклади: Arial, Helvetica.

Pacifico
Lobster

Рукописні шрифти – можуть додати індивідуальність і творчість, але їх слід використовувати обережно, оскільки вони можуть бути важкими для читання. Приклади: Pacifico, Lobster.

DAYPOSTERBLACK
Euclid Fraktur

Декоративні шрифти – варто використовувати рідше, більше як невеликий графічний елемент, оскільки вони можуть відволікати увагу від основного повідомлення і бути незрозумілими для читання. Приклади: DayPosterBlack, Euclid Fraktur.

Гасло – коротка фраза, яка передає головну ідею бренду. Наприклад, Рошен – «Солодкий смак щастя»; Київстар – «Київстар – це просто»; Укрзалізниця – «З нами у подорож»; Нова Пошта – «Швидше, ніж ви думаєте».

Комунікації – це всі дії компанії, через які вона взаємодіює зі своєю авдиторією, забезпечуючи те чи інше сприйняття її бренду. Приведемо кілька прикладів таких комунікацій від відомих брендів: Coca-Cola змінила традиційний дизайн своїх пляшок, замінивши логотип на популярні імена (наприклад, «Олександр», «Марія»). Це заохочувало людей купувати пляшку зі своїм або знайомим іменем та ділитися напоєм з іншими. McDonald's запропонував випадковим клієнтам у певний проміжок часу «заплатити» за своє замовлення не грошима, а проявами любові: обіймами, дзвінком мамі чи обіцянкою зробити щось приємне для іншого. Метою було підвищити лояльність клієнтів та показати, що бренд цінує людські емоції та моменти, які приносять радість.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Що таке брендбук

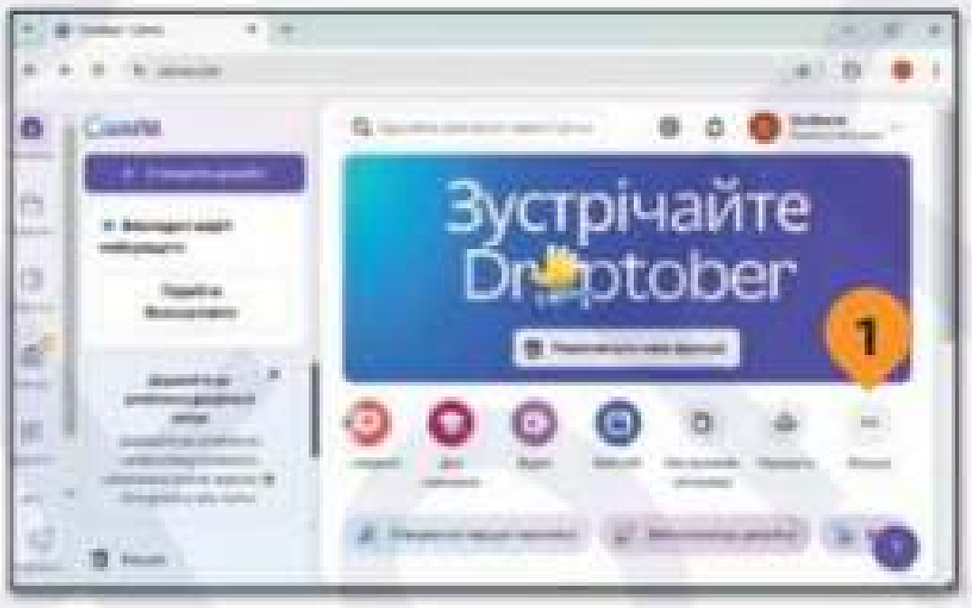
Брендбук – це документ, що містить в собі правила використання елементів фірмового стилю. У ньому описано, як правильно використовувати логотип, кольори, шрифти та інші елементи, щоб зберігати цілісність бренду.

Як створити успішний бренд

1. **Дослідження цільової авдиторії:** зрозумійте, хто є вашим клієнтом – дослідіть його вік, інтереси, потреби та очікування.
2. **Мозковий штурм:** розробіть ідеї щодо візуальної ідентичності бренду та способів комунікації.
3. **Розробка дизайну:** створіть дизайн шаблонів для реклами, соціальних мереж та інших інформаційних продуктів відповідно до погодженого стилю.

Безперечно головним елементом бренду є логотип. Створити його можна за допомогою векторного графічного редактору, як-от Figma, Canva, Inkscape, BoxySVG, Gravit Designer, CorelDRAW, Adobe Illustrator тощо. Розглянемо приклад створення логотипу.

В дослідженнях цільової авдиторії прийнято брати платоспроможних людей віком від 18 до 60.



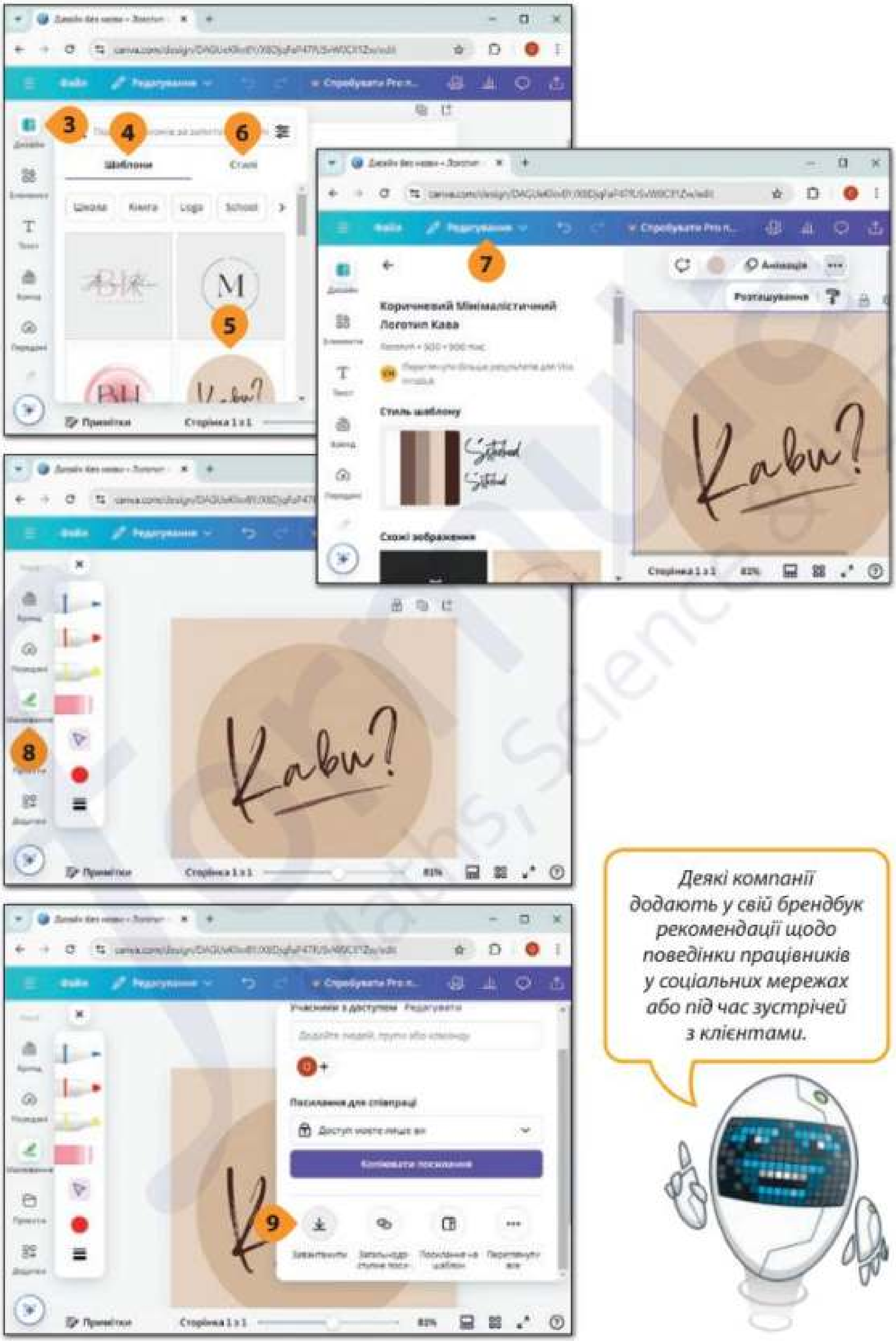
Щоб створити логотип:

- перейдіть за посиланням на сайт **canva.com**;
- увійдіть в акаунт або зареєструйтеся за допомогою електронної пошти;
- натисніть (1) та клацніть **Логотип** (2);
- оберіть **Дизайн** (3) на бічній панелі меню;
- за допомогою **Шаблони** (4) оберіть лого в залежності від того, в який сфері (5) планується бізнес чи особистий бренд;
- за бажанням змініть **Стиль** (6), або відредагуйте (7) логотип;
- за допомогою **Малювання** (8) додайте індивідуальності логотипу за бажанням;
- збережіть свій проєкт та завантажте (9) на свій комп'ютер.

ІСТОРІЯ

Один із перших брендів, який систематично використовував фірмовий стиль, був Coca-Cola. Їхній червоний колір та унікальний шрифт стали впізнаваними у всьому світі ще на початку XX століття.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.



Деякі компанії додають у свій брендбук рекомендації щодо поведінки працівників у соціальних мережах або під час зустрічей з клієнтами.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Проєктування інформаційного продукту

Інформаційний продукт бренду – це будь-який вид цифрового контенту чи ресурсу, створеного брендом, метою якого є передача інформації до своєї авдиторії. Такий продукт часто використовується як частина маркетингової стратегії для побудови довіри, підвищення обізнаності про бренд та залучення клієнтів. До інформаційних продуктів можуть відноситися: електронні книги (детальні матеріали, що пояснюють певну тему); вебінари або онлайн-семінари (інтерактивні події, на яких обговорюються актуальні питання); курси (освітні матеріали з різних тем, пов'язаних з послугами чи продуктами бренду); статті, блоги або гайди (тексти з корисною інформацією, яка відповідає на запити авдиторії); інфографіки (візуальні представлення даних або ідей); подкасти (аудіоінформація з певної теми).

Мета інформаційного продукту полягає у формуванні позитивного сприйняття бренду і може сприяти збільшенню продажів.

Як створити інформаційний продукт бренду?

1. **Проєктування:** визначте, який тип контенту вам потрібен (текст, зображення, відео, анімації тощо).
2. **Розробка:** створіть шаблони для публікацій на основі брендбуку.
3. **Публікація та представлення:** оберіть платформи для публікацій (сайт, соціальні мережі) та представте ваш контент.

Контент-план для публікацій бренду

Контент-план – це графік або список того, коли, де та які матеріали будуть публікуватися. Він допомагає підтримувати активну присутність бренду в інформаційному просторі.

БЕРЕЖІТЬ СЕБЕ

Не забувайте про короткі перерви під час роботи за комп'ютером! Це допоможе зберегти вашу увагу та здоров'я.

Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 3

Технології генерування ідей

Іноді, створюючи контент для інформаційних продуктів, буває так, що дуже важко придумати нову ідею для публікації або посту, створити новий дизайн логотипу чи сторінку сайту. Щоб підштовхнути креативність, існують різні технології.

Мозковий штурм (Brainstorming):

- колективне генерування ідей;
- заборона критики;
- кількість ідей важливіша за якість;
- заохочення креативності;
- різноманітність учасників.

Метод 6 капелюхів (Едвард де Боно):

- **білий**: факти та інформація;
- **червоний**: емоції та почуття;
- **чорний**: критичне мислення;
- **жовтий**: позитивні сторони;
- **зелений**: креативність;
- **синій**: контроль процесу.

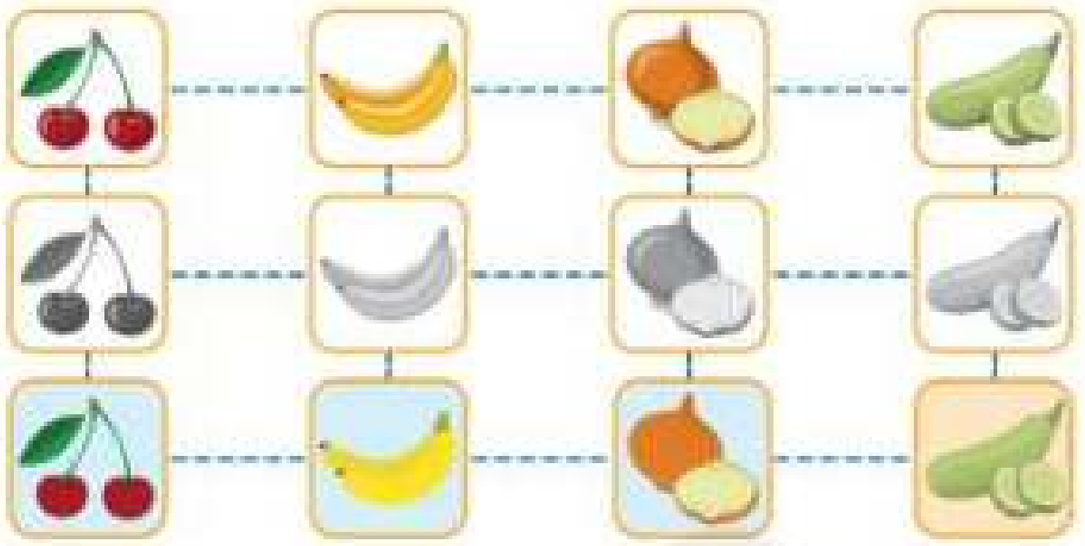
Метод Уолта Діснея:

- **мрійник**: генерація ідей;
- **реаліст**: практична оцінка;
- **критик**: аналіз недоліків;
- послідовний перехід між станами.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Метод аналогій:

- пошук схожості в різних сферах;
- перенесення рішень з однієї галузі в іншу;
- природні, технічні, соціальні аналогії.



Mind Mapping:

- графічне представлення ідей;
- головна проблема;
- гілки та підгілки;
- зв'язки між поняттями;
- просторове мислення.

SCAMPER-метод:

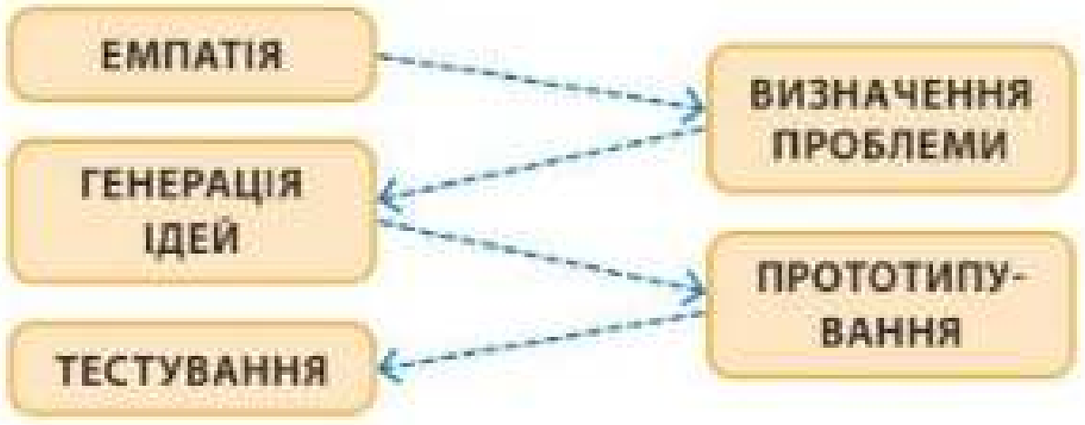
- **S**ubstitute (замінити) – замінити частину проєкту на щось інше.
- **C**ombine (комбінувати) – поєднати два та більше елементів для створення нового рішення.
- **A**dapt (адаптувати) – змінити проєкт для нових умов або потреб.
- **M**odify (модифікувати) – змінити розмір, форму або інші характеристики продукту.
- **P**ut to another use (використати інакше) – знайти нові способи використання проєкту.
- **E**liminate (вилучити) – видалити непотрібні частини проєкту.
- **R**everse (інверсія) – змінити порядок або напрямок процесу.

Провокаційні питання:

- «що буде, якщо...»;
- зміна контексту;
- руйнування стандартних припущень;
- парадоксальні запитання.

Дизайн-мислення (Design Thinking):

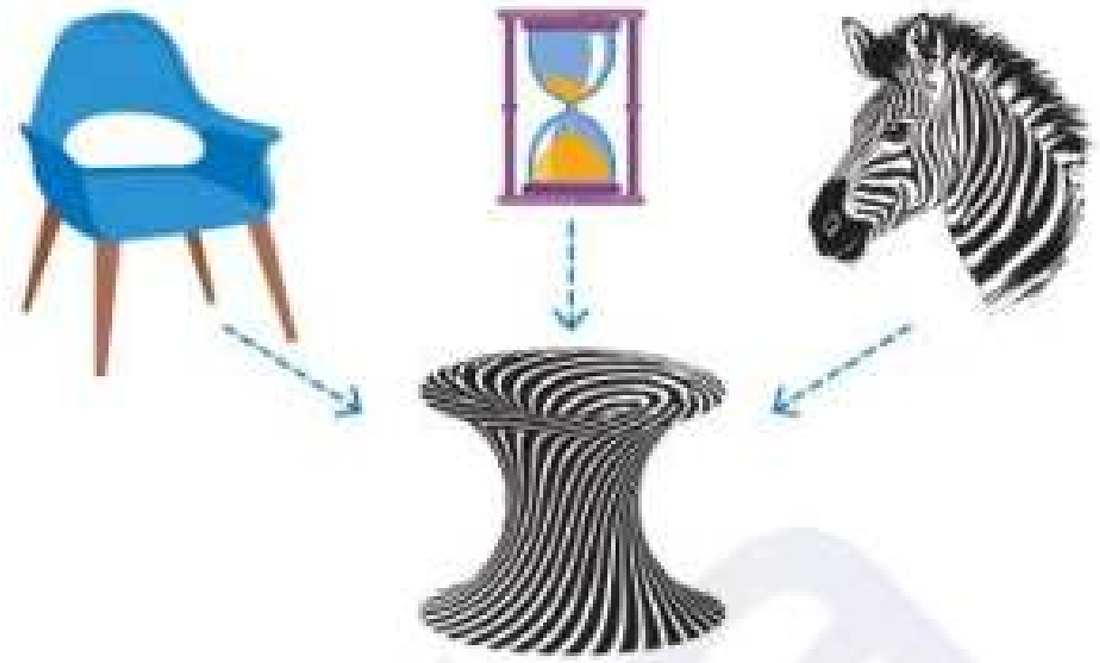
- емпатія: розуміння користувача;
- визначення проблеми;
- генерація ідей;
- прототипування;
- тестування.



© Binary Logic та © Видавництво «Лінгвіст». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

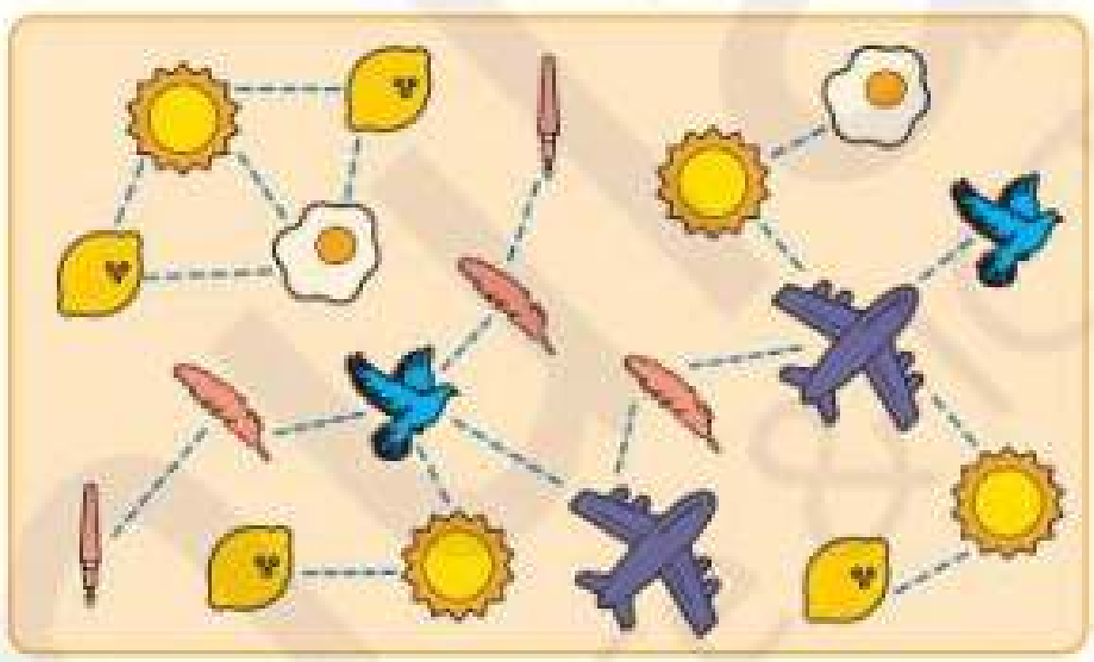
Метод фокальних об'єктів:

- випадкові об'єкти;
- перенесення властивостей;
- несподівані поєднання;
- народження нових концепцій.



Технологія Сінектики:

- несподівані аналогії;
- особиста аналогія;
- символічна аналогія;
- фантастична аналогія.



Reverse Brainstorming:

- постановка проблеми навпаки;
- як її посилити;
- пошук протилежних рішень;
- зміна кута зору.

Алгоритм вибору технології:

Оцініть проблему

Визначте ресурси

Виберіть підхід

Експериментуйте

Ви мабуть помічали, що коли залучені в процес, як-от уважно слухаєте подкаст або виступ цікавого фахівця, часто малюєте по аркушах або серветках лінії, кола, квадрати, квіти тощо. Це природний процес самоорганізації мозку. Він дозволяє сконцентрувати увагу, активувати мозок, активізувати дрібну моторику та стимулювати нейронні зв'язки, а головне – виявляє підсвідому креативність.

Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Тема 4

Дизайн-мислення

Дизайн-мислення (англ. **Design Thinking**) – це підхід до вирішення проблем, орієнтований на людину, алгоритм вирішення певних проблем з використанням нестандартних підходів, інноваційних рішень, досліджень. Зверніть увагу, що цей алгоритм орієнтований не на бізнес-завдання, а саме на вирішення проблем людини. Він поєднує емпатію, креативність та раціональність. Питання, які ставлять фахівці при застосуванні цього методу в роботі: «Хто користуватиметься продуктом?», «Як це вплине на життя користувача?», а не «Що отримає бізнес?».

Дизайн-мислення – не просто метод, а філософія творчого підходу до вирішення складних завдань.

Процес дизайн-мислення побудований на кількох принципах:

- творчість, а не аналітичний підхід до проблеми. Аналіз все ж таки буде присутній – для кращого розуміння питання та підбору найефективнішого рішення;
- можливість охопити проблему як в цілому, так й у всіх деталях;
- вміння поставити себе на місце споживача, зрозуміти, які почуття та емоції він відчуває – всі інновації мають бути орієнтовані на людей;
- готовність до експериментів, виходу за межі стандартного та шаблонного бачення.

Ціль використання дизайн-мислення – розробка максимально корисних продуктів для користувачів. За допомогою п'яти графічних елементів можна намалювати все що завгодно.

Дизайн-мислення застосовують при розробці цифрових продуктів. Продуктовий дизайн, освіта, соціальні інновації тощо – далеко не повний перелік сфер застосування цього методу.

Ви вже стикалися з цим методом при розробці прототипу сайту: створювали карту емпатій, описували потенційного відвідувача сайту, при розробці сайту брали до уваги UI/UX-дизайн.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Етапи дизайн-мислення

Емпатія

Глибинне вивчення користувача
Методи дослідження:
- Інтерв'ю
- Спостереження
- Занурення в середовище

Створення карти емпатій
Розуміння потреб та «болю» користувача

Визначення проблеми

Структуризація зібраної інформації
Формування проблемного запиту
Побудова карти проблем
Фокусування на найбільш важливому

Генерація ідей

Технології брейнстормінгу
Методи колективної генерації
Кількість важливіша за якість
Заборона критики на етапі генерації

Прототипування

Створення моделей рішення
Швидкі макети
Мінімальний життєздатний продукт
Різні рівні деталізації

Тестування

Перевірка гіпотез
Зворотний зв'язок від користувачів
Аналіз результатів
Готовність до змін

Для реалізації алгоритму дизайн-мислення зазвичай використовують не тільки аркуш паперу та олівець, але й програми, які вам вже знайомі.

Контент-план

Розробка вдалого дизайну цифрового продукту – це тільки перший крок до створення цифрового образу, бренду продукту. Коли цифровий продукт виходить у світ він може привернути увагу. Але, щоб підтримувати постійний інтерес до нього цього замало. Кількість інформації, яку ви отримуєте кожен день може перемикати увагу людей від вашого проєкту / продукту. Тож потрібно розробити стратегію, яка дозволить нагадувати про ваш проєкт потенційним користувачам вашого продукту. В цьому вам допоможе контент-план.

Часто до графіка контенту додають типи публікацій, теми й начерки щодо контенту – щоб простіше вести та заповнювати. Зазвичай контент-план роблять у вигляді таблиці або карток із завданнями, термінами і відповідальними.

Контент-план – це інструмент планування. Він виконує багато корисних завдань для компанії або окремих людей.

***Контент-план** – це графік публікацій про будь-який проєкт/продукт з метою просування його на ринку або серед авдиторії.*

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Робота з контент-планом складається з наступних етапів

Підготовчий етап

Аналіз цілей бізнесу
Вивчення цільової авдиторії
Визначення унікальної пропозиції
Окреслення комунікаційних каналів

Дослідження авдиторії

Створення персональних профілів
Карти інтересів
Аналіз проблем та потреб потенційного користувача
Вподобання в контенті

Типи контенту

Освітній: навчальні статті, гайди, інструкції, лайфхаки
Розважальний: меми, гумористичний контент, інтерактивні опитування, цікаві факти
Експертний: аналітичні матеріали, дослідження, експертні думки, кейси
Контент, який продає: огляди продуктів, порівняння, відгуки, акційні пропозиції

Структура контент-плану

Тема контенту
Формат
Канал публікації
Дата публікації
Мета контенту
Метрики

Інструменти планування

Notion
Trello
Google Sheets
Asana
Canva

Контент-матриця

Стовпці:
канали комунікації
Рядки:
типи контенту
На перетині – контент-одиниці

Частота публікацій

Instagram: 1–2 пости щодня
Facebook: 3–4 пости на тиждень
YouTube: 1–2 відео на тиждень
Блог: 2–3 статті на тиждень

Алгоритм створення

Генерація ідей
Перевірка релевантності
Підготовка контенту
Узгодження
Публікація
Аналіз результатів

Метрики ефективності

Охоплення
Залученість
Конверсія
Лояльність авдиторії

Інструменти аналітики

Google Analytics
Facebook Insights
YouTube Analytics
Similarweb

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

У сучасному цифровому світі стиль більше, ніж просто естетика – це потужна мова комунікації, що транслює цінності, емоції та унікальність бренду. Дизайн-мислення перетворює статичні візуальні рішення на динамічний діалог між брендом та авдиторією, де кожен елемент, від кольору до типографіки, стає носієм глибинного сенсу та провідником стратегічних намірів. Цифровий образ еволюціонує від інструменту презентації до самостійного феномену, що формує сприйняття, будує довіру та створює емоційний зв'язок, який виходить далеко за межі традиційних уявлень про маркетинг і комунікацію.

Саме дизайн-мислення в сучасному цифровому просторі допомагає перетворити звичайні візуальні елементи на інструмент взаємодії з авдиторією.

Дизайн-мислення – неймовірно потужний спосіб розв'язання проблем, хоча він може здатися вам не звичним, по суті, ми вже давно та постійно його використовуємо.

Не бійтеся експериментувати, бажати, мріяти. У вас все вийде!

ДОВІДКА

Конверсія – процес перетворення відвідувачів сайту або потенційних клієнтів у реальних покупців чи користувачів, які виконують цільову дію (покупка, реєстрація, підписка на розсилку тощо).

Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

https://formula.education/online-ict-8-klas/

Інші платформи

Проєкт

Підсумки

Глосарій

© Binary Logic та © Видавництво «Пінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Навчальне видання

Джон Ендрю **Біос**

ІНФОРМАТИКА

Підручник для 8 класу закладів загальної середньої освіти

Підручник відповідає Державним санітарним нормам і правилам
«Гігієнічні вимоги до друкованої продукції для дітей»

Список використаних ілюстрацій

Головна редакторка *Діана Головань*
Редакторка-методистка *Олена Вернигора*
Дизайнерка *Вікторія Субочева*
Коректорка *Олена Богатир*
Перекладач *Ростислав Вернигора*

Формат 60×90/8. Ум. друк. арк. 26,0. Обл.-вид. арк. 23,4.
Наклад 00000 прим. Зам. № 00000.

ТОВ «Видавництво "Лінгвіст"». 04208, м. Київ, просп. Василя Порика, 9-Г, к. 58.
Адреса редакції: 04071, м. Київ, вул. Нижній Вал, 17/8. E-mail: publishing@linguist.ua. Тел. (044) 599-32-28.
Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру видавців,
виготівників і розповсюджувачів видавничої продукції серія ДК № 6168 від 03.05.2018 р.

Please note: This book contains links to websites that are not maintained by Binary Logic. Although we make every effort to ensure these links are accurate, up-to-date and appropriate, Binary Logic cannot take responsibility for the content of any external websites.
Trademark notice: Product or corporate names mentioned herein may be trademarks or registered trademarks and are used only for identification and explanation without intent to infringe. Binary Logic disclaims any affiliation, sponsorship, or endorsement by the respective trademark owners. Microsoft, Windows, Windows Live, Office, Office 365, Access, Excel, PowerPoint, Outlook, Teams, OneDrive, OneNote, Bing, Edge, Skype, and Kodu are trademarks or registered trademarks of Microsoft Corporation. Google, Gmail, Chrome, Google Docs, Google Drive, Android, Blogger, Google Maps, Google Translate, and YouTube are trademarks or registered trademarks of Google Inc. Apple, iPad, iPhone, Pages, Numbers, Keynote, iCloud, and Safari are registered trademarks of Apple Inc. LibreOffice is a registered trademark of the Document Foundation. Facebook, Messenger, Instagram, and WhatsApp are trademarks owned by Facebook and its affiliate companies. Twitter is a trademark of Twitter, Inc. Zoom is a trademark of Zoom Video Communications, Inc. Cisco WebEx is a trademark of Cisco Systems, Inc. The above companies or organizations do not sponsor, authorize, or endorse this book.
The publisher has made every effort to trace all copyright holders, but if they have inadvertently overlooked any they will be pleased to make the necessary arrangements at the first opportunity.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

«Десять евристик юзабіліті для дизайну інтерфейсів» (1994) від Якоба Нільсена

1.
Видимість стану системи

2.
Взаємозв'язок між системою та реальним світом

3.

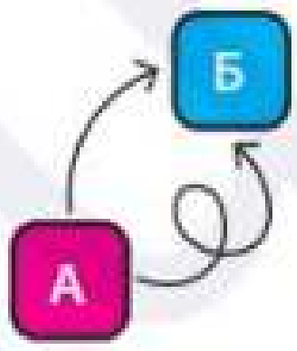
Користувацький контроль і свобода

4.
Послідовність і стандарти

5.
Запобігання помилкам

6.

Розпізнавання, а не пригадування

7. 

Гнучкість та ефективність використання

8.
Естетичний і мінімалістичний дизайн

9.
Допомога користувачам у розпізнаванні, діагностуванні та виправленні помилок

10.
Допомога та документація

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

Корисні плагіни для роботи у Figma

Figmotion	Плагін для створення анімації, він дозволяє додавати до макета анімаційні ефекти.
Color Kit	Плагін, який генерує світліші й темніші відтінки певного кольору, його зручно використовувати для створення кольорових шкал.
Mesh Gradient	Створює градієнтне зображення на основі двовимірної сітки, дозволяє редагувати вершини та краї сітки для отримання ідеального градієнта.
BeatFlyer Lite	Плагін додає анімаційні ефекти окремо для кожного шару, комбінуючи безліч ефектів.
Content Reel	Плагін від Microsoft створює контент для заповнення макета: тексти, аватарки, імена користувачів, згенеровані номери телефонів, адреси електронної пошти, назви компаній, URL-адреси.
Vectary 3D elements	Плагін дозволяє створювати й редагувати тривимірні графічні об'єкти.
Design Lint	Плагін, який знаходить помилки в стилі вашого дизайну: невідповідність кольорів, шрифтів, заливки.
Isometric	Дозволяє створювати ізометричні шари без необхідності їх ручного налаштування.
remove.bg	Плагін, який дозволяє швидко видалити фон із зображення.

Структура брендбуку

© Binary Logic та © Видавництво «Лінгвіст» . Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в пошуковій системі або передана в будь-якій іншій формі будь-якими способами без письмового дозволу видавця. За жодних обставин ніяка частина цього матеріалу не може бути скопійована для перепродажу або для іншого використання.

НОВА УКРАЇНСЬКА ШКОЛА

Джон Ендрю Біос

ІНФОРМАТИКА 8

ЦИФРОВІ ПІДЛІТКИ

- Виклад навчального матеріалу представлений у доступній, чіткій та ілюстративній формі відповідно до вікових особливостей учнів.
- Підручник створено з урахуванням компетентнісного підходу в освіті.
- Відповідає міжнародним стандартам навчання ІКТ-грамотності та цифрових компетентностей DigComp.
- Містить цифровий контент із додатковими інтерактивними вправами та файлами-заготовками до практичних робіт.

ISBN 978-617-8146-47-4

© Binary Logic та © Видавництво «Формула». Усі права захищені. Жодна частина цієї публікації не може бути відтворена, збережена в поширеній системі або передана в будь якій якій формі будь якими способами без письмового дозволу видавця. За жодних обставин жодна частина цього матеріалу не може бути скопійована для подальшого або для масового використання.