

Конспект лекції (01)

▼ про Python і чому він (переваги)

Python — добре зарекомендована мова програмування загального призначення, яку вперше випустив її творець, Гвідо ван Россум, у 1991 році. Ця стабільна та зріла мова є високорівневою, динамічною, об'єктно-орієнтованою та кросплатформною — усе це дуже привабливо. Python працює на macOS, найсучасніших варіантах Unix, включаючи Linux, Windows і, з деякими виключеннями, на мобільних платформах.

Python забезпечує високу продуктивність на всіх етапах життєвого циклу програмного забезпечення: аналіз, проектування, створення прототипів, кодування, тестування, налагодження, налаштування, документування та, звичайно, обслуговування. Популярність цієї мови невпинно зростала протягом багатьох років, ставши лідером індексу TIOBE у жовтні 2021 року. Сьогодні знайомство з Python є плюсом для кожного програміста: він проник у більшість ніш, відіграючи корисні ролі в будь-якому програмному рішенні.

Python забезпечує унікальне поєднання елегантності, простоти, практичності та величезної потужності. Ви швидко станете продуктивними з Python завдяки його узгодженості та регулярності, багатій стандартній бібліотеці та багатьом пакетам і інструментам сторонніх розробників, які легко доступні для нього. Python простий у вивченні, тому він цілком підходить, якщо ви новачок у програмуванні, але також достатньо потужний для найдосвідченішого експерта.

Основні переваги коротко:

- Широко поширений
- Легкий для вивчення
- Гарна робота з відгадуванням та опис помилок у програмі

▼ які версії Python використовують

Зараз поширений Python версії 3, однак ще можна зустріти код чи проекти написані на Python версії 2.

Ми вивчаємо лише нову версію 3. Однак, розуміючи код версії 3 ви без проблем прочитаете і розберетеся з кодом версії 2, якщо колись зустрінете його.

▼ інтерпретований чи компільований?

Є два підходи до перетворення коду, що пише програміст у код, що розуміє комп'ютер:



Компільовані мови програмування (C, C++, Java, etc.)

Спочатку Ви створюєте програму, використовуючи команди, які є в цій мові програмування, а потім Вам необхідно перевести (компілювати) цю програму до рівня, який розуміє процесор. Компільована програма виконується безпосередньо ОС.



Інтерпретовані мови програмування (Python, PHP, etc.)

Ви створюєте програму, використовуючи доступний набір команд, який є в цій мові програмування, а потім, в процесі виконання програми, інтерпретатор переводить команди, які розуміє людина, в команди, які може виконати процесор.

Інтерпритована програма виконується через інтерпритатор, що слід встановити в ОС.

Перші кроки

▼ Завантажити інтерпретатор Python

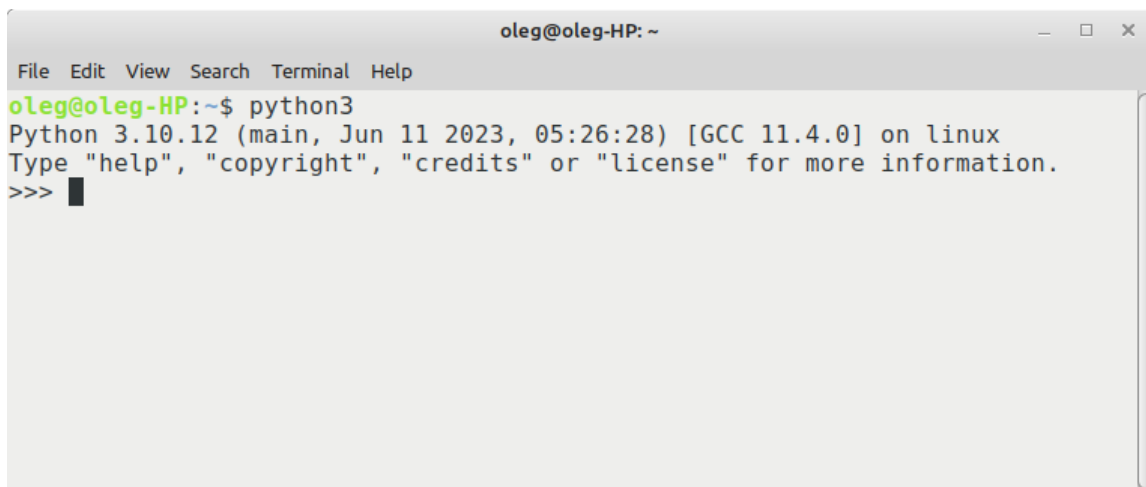
Завантажити можна з офіційного сайту:

<https://www.python.org/downloads/>

▼ Як перевірити те, що Python встановлений у системі?

У терміналі (для **Windows** потрібно використовувати **cmd.exe**) необхідно ввести `python3` (або `python` , якщо **Windows**).

Повинно з'явитися щось на зразок цього

A screenshot of a terminal window titled 'oleg@oleg-HP: ~'. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal shows the command 'python3' being executed, which outputs 'Python 3.10.12 (main, Jun 11 2023, 05:26:28) [GCC 11.4.0] on linux' and 'Type "help", "copyright", "credits" or "license" for more information.' followed by a prompt '>>>' and a cursor.

Ми бачимо працюючий інтерпретатор, і це є ознакою того, що Python встановлений коректно.

Якщо такого напису немає.

Два варіанти:

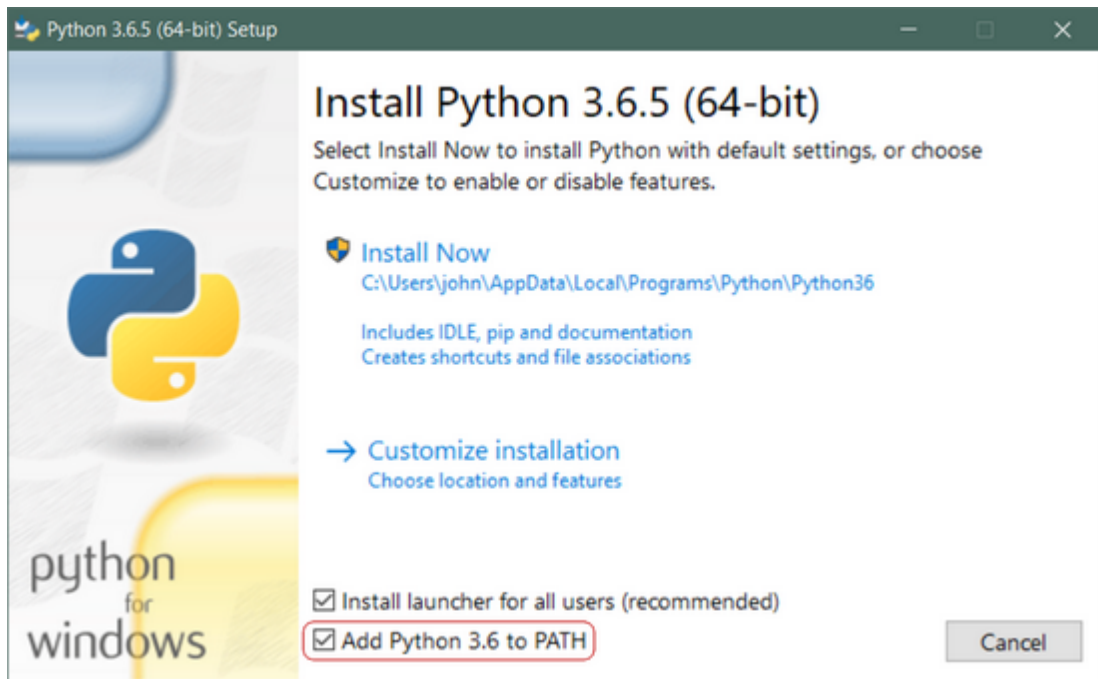
- або Python встановлено некоректно,
- або взагалі не встановлено.

Якщо Python встановлено некоректно, його потрібно виділити.

За адресою <https://www.python.org/downloads/> завантажуйте інсталятор під вашу ОС. І становлюєте.

Важливо!

Якщо у Вас ОС Windows, обов'язково додайте Python до PATH (як зазначено на зображенні).



▼ Приклади виконання найпростіших команд із консолі інтерпретатора

```
print("Hello world")
```

```
IDLE Shell 3.11.2
File Edit Shell Debug Options Window Help
Python 3.11.2 (tags/v3.11.2:878ead1, Feb 7 2023, 16:38:35)
4 bit (AMD64) on win32
Type "help", "copyright", "credits" or "license()" for more
>>> print("Hello world")
Hello world
>>>
```

▼ Створення найпростішої програми та збереження її у файл

File > New file

Ввести код

File > Save

Вказати ім'я та розташування

▼ Запуск збереженого файлу на виконання

Для того, щоб запустити файл на виконання, потрібно відкрити термінал і перейти в ту папку, куди було збережено файл та набрати команду **python3 hello.py** та нажати Enter (якщо Windows то **python hello.py**) В терміналі ми побачимо результат роботи нашої програми

Знайомство з IDE (Integrated Development Environment)

▼ Що таке IDE і навіщо воно треба

Термін IDE розшифровується як Integrated Development Environment або «інтегроване середовище розробки». Це набір інструментів, які дозволяють створювати програми. Іншими словами, якщо говорити дуже просто, то IDE – це програма, де створюються інші програми.

Які переваги отримує програміст використовуючи IDE?

- Підсвітка синтаксису
- Рання перевірка і вказівки на явні помилки або одруківки
- Підтягування документації або гайдів до коду
- Всі дії пов'язані з програмуванням та збереженням коду виконуються в одній програмі

▼ Visual Studio Code або PyCharm

ПРИМІТКА ДЛЯ ВИКЛАДАЧА:
показуємо і використовуємо щось одне, ту IDE яка вам особисто зручніше.

Visual Studio Code (УВАГА! ЦЕ НЕ Visual Studio Community!!!)

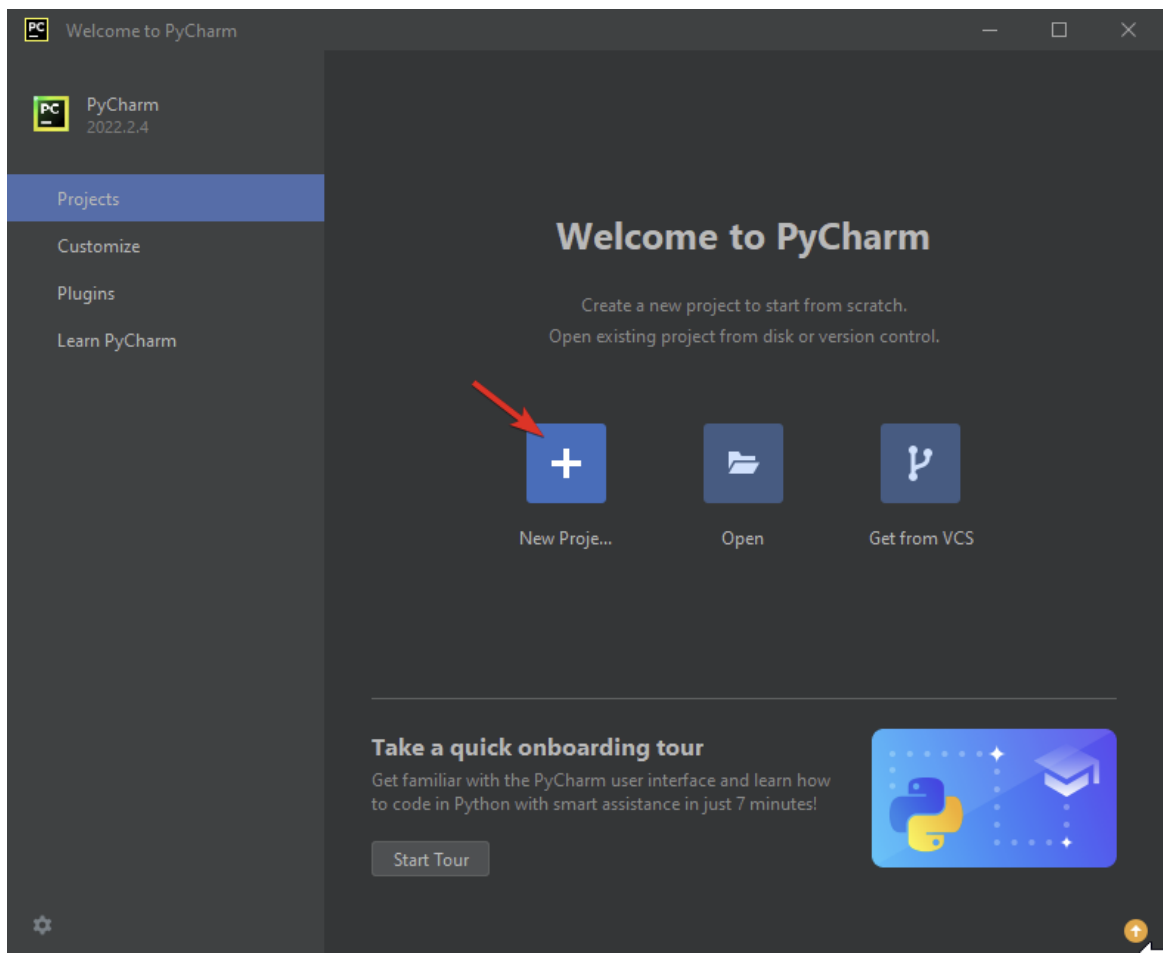
- [Скачати](#)

PyCharm

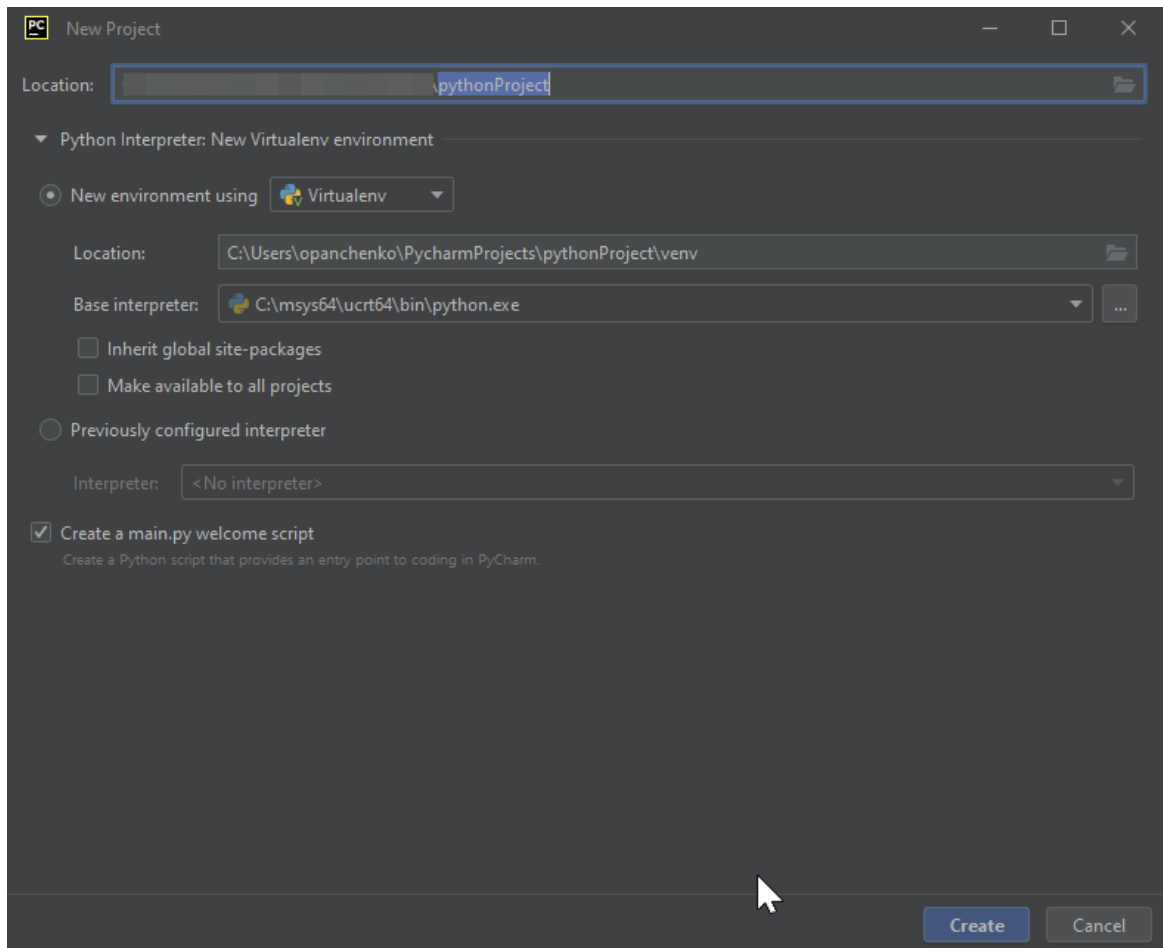
- [Скачати](#)

▼ PyCharm: Створення першого проекту

Все дуже очевидно:

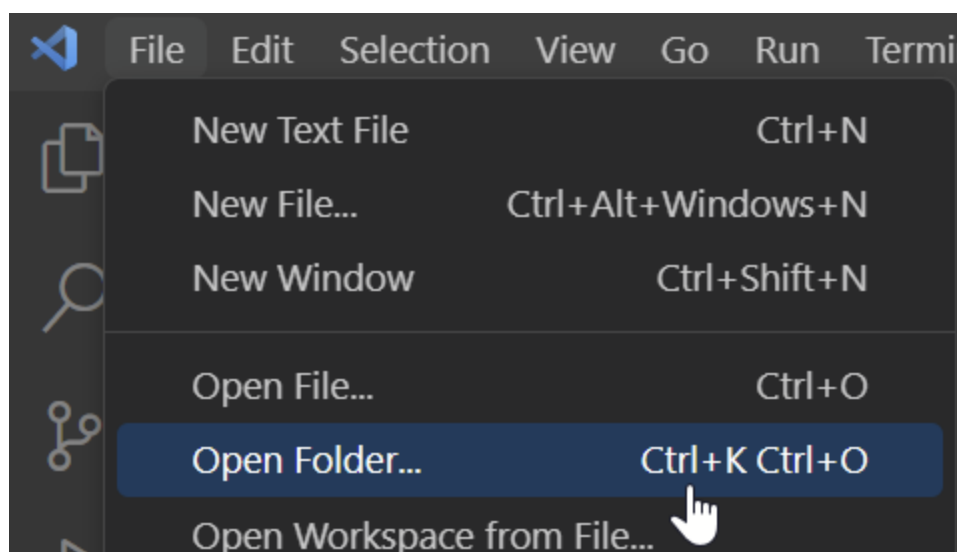


Далі вказуємо шлях і натискаємо Create (Створити):



▼ VS Code: Створення першого проекту

- Створіть порожню папку засобами ОС
- Відкрийте її через меню File > Open Folder

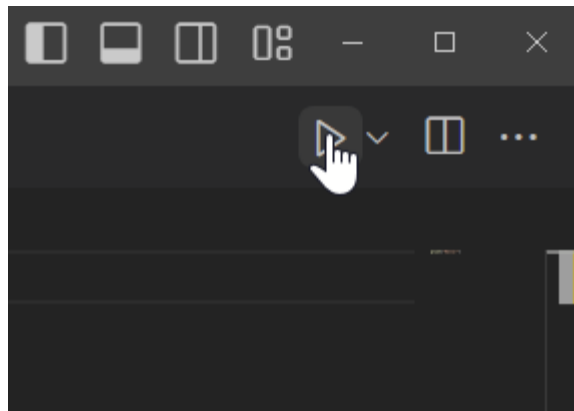


▼ VS Code: Додавання Python файлу до проекту. Запуск створеної програми на виконання

Щоб додати оберіть File > New

Задайте ім'я та розширення .py

Щоб виконати натисніть трикутник у верхній частині:



Початок програмування у Python

▼ Змінні (Variables), створення та використання змінних

Змінна — це іменована область пам'яті. Вона може містити будь-який тип даних: числа, рядки, масиви, об'єкти. Змінні використовуються для збереження та отримання значень, які можуть змінюватися під час виконання програми. Змінна складається з двох основних компонентів: **ім'я** та **значення**.

Ім'я — ідентифікатор, що дозволяє нам посилатися на дані, що зберігаються у пам'яті.

Значення змінної — дані різних типів, що зберігаються в ній.

Python не має строгої типізації, тобто ви можете змінювати тип значень у змінній, однак хорошим тоном програмування буде цього не робити.

У Python можна не вказувати наперед ім'я змінної, якимось чином ініціалізувати чи резервувати її. Достатньо просто задати ім'я = значення.

= — знак присвоєння

== — знак "дорівнює"

```
age = 18
name = "Stephen"
```



```
hourly_price = 12.56
```

При створенні та використанні змінних дуже важливими є їх імена, які повинні підказувати вам, як програмісту, що саме містить данна змінна.

Використовуйте прості та зрозумілі англійські іменники та слова.

За домовленістю змінні називаються лише латинецею, маленькими літерами. Для розділення слів змінній використовується символ підкреслення.

▼ Відступи (Indentation)

Відступ це певна кількість пробілів на початку рядка коду.

Якщо в інших мовах програмування відступи в коді призначені лише для зручності читання, у Python відступи дуже важливі.

Python використовує відступи для позначення блоку коду.

приклад:

```
age = 18
name = "Stephen"
if age == 18:
    print("Congratulations, you are no longer a teenager!")
# next code here
```

Якщо умова вірна, правильна — то виконається весь код, після слова if перед яким стоїть 4 пробіли. У даному випадку — це одна команда print.

▼ Найпростіші математичні операції: додавання, віднімання, множення, ділення

Математичні операції можуть виконуватися з цілими числами, числами з комою та числами з уявною часткою.

Для виконання операції можна вказати безпосередньо значення або змінні, в яких міститься відповідний тип даних.

```
a = 5
b = 3

summ = a + b # Додавання
diff = a - b # Віднімання
mult = a * b # Множення
divd = a / b # Ділення
```

Результат математичної операції присвоюється у нову або існуючу змінну.

▼ Вивід даних через print()

Результати роботи програми треба виводити на екран через print()

```
print(object(s), sep=separator, end=end, file=file, flush=flush)
```

В функції print() ми можемо виводити значення змінних, текст, числа і т. ін.

▼ Перше ДЗ та як його робити

Код, що було розглянуто на лекції та домашні завдання доступні у Github.

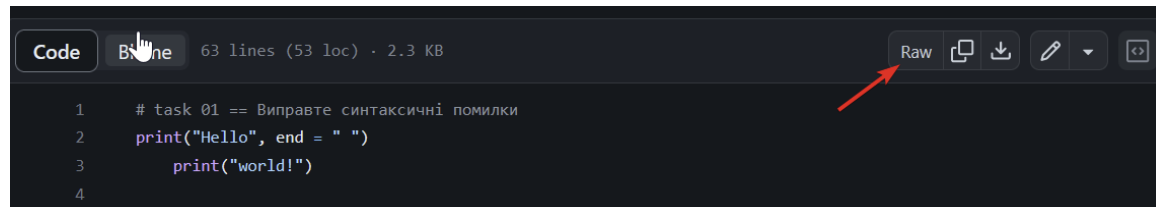
Що таке Git ми розглянемо дещо пізніше, поки що ви можете просто перейти за посиланням на навчальний репозиторій:

https://github.com/dntpanix/automation_ga

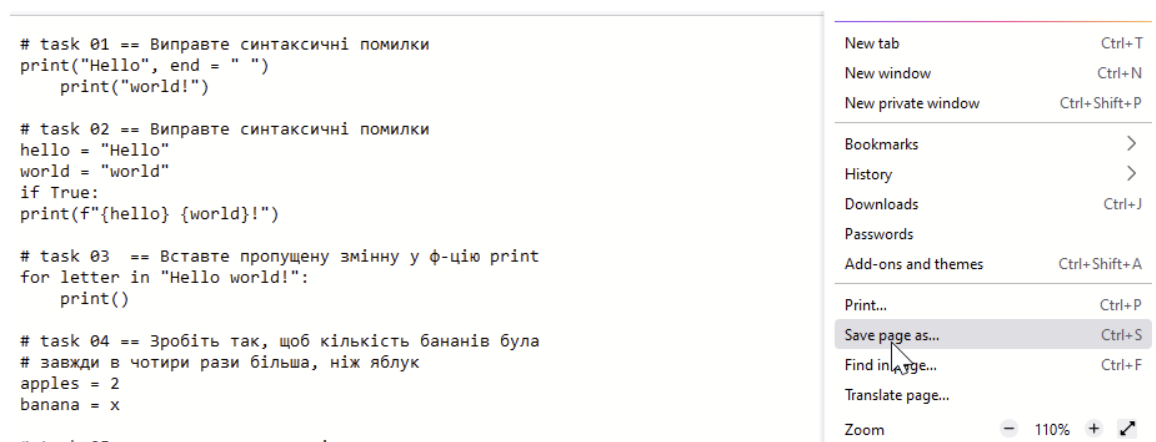
Завантажте файл

https://github.com/dntpanix/automation_ga/blob/main/lesson_01/homework_01.py

натисніть Raw



Оберіть “Зберегти як” і збережіть в папці вашого навчального проекту



виконайте ДЗ згідно з інструкціями написаними в коментарях до коду.

Завершене ДЗ завантажте у навчальну систему через форму відповіді.
Переконайтеся що завантажуєте правильну і останню версію роботи.