

Міністерство освіти і науки України
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра спеціалізованих комп'ютерних систем

**МЕТОДИЧНІ ВКАЗІВКИ
ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ
з дисципліни
“ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ
КОМП'ЮТЕРИЗОВАНИХ ВИМІРЮВАЛЬНИХ
ПРИСТРОЇВ”**

для студентів освітньо-кваліфікаційного рівня «бакалавр»
спеціальності 175 «Інформаційно-вимірювальні технології»

ТЕРНОПІЛЬ – 2025

Методичні вказівки до виконання лабораторних робіт з дисципліни “Програмне забезпечення комп’ютеризованих вимірювальних пристроїв” для студентів освітньо-кваліфікаційного рівня «бакалавр» спеціальності 175 «Інформаційно-вимірювальні технології» / Укл.: Попик Ю. І., Сегін А. І. – Тернопіль: ЗУНУ, 2025. – 43 с.

Методичні рекомендації для виконання лабораторних робіт складаються з тем, що рекомендовані освітньою програмою підготовки на основі Стандарту вищої освіти для спеціальності 151 «Автоматизація та комп’ютерно-інтегровані технології»

Укладачі: Попик Юрій Ігорович, асистент
 Сегін Андрій Ігорович, к.т.н., доцент

Рецензенти:

Франко Ю. П. к.т.н., завідувач кафедри комп’ютерних технологій
Тернопільського національного педагогічного університету

Возняк С. І. Голова правління Тернопільської міської ГО «Тернопільський
освітній комунікаційний центр»

*Затверджено на засіданні кафедри спеціалізованих комп’ютерних систем,
протокол № 3 від 16.10.2024,*

*Розглянуто та схвалено групою забезпечення спеціальності
175 «Інформаційно-вимірювальні технології»
протокол № 2 від 16.10.2024.*

*Розглянуто та схвалено вченою радою факультету комп’ютерних інформаційних
технологій, протокол № 5 від 05.02.2025 р.*

© Попик Ю. І., Сегін А. І., 2025

© ЗУНУ, 2025

Зміст

Загальні положення	4
Оформлення звіту.....	4
Захист звіту	4
Оцінювання лабораторної роботи	4
Завдання до лабораторної роботи № 1.....	6
Завдання до лабораторної роботи № 2.....	8
Завдання до лабораторної роботи № 3.....	10
Завдання до лабораторної роботи № 4.....	12
Завдання до лабораторної роботи № 5.....	14
Завдання до лабораторної роботи № 6.....	16
Завдання до лабораторної роботи № 7.....	19
Налагодження середовища розробки.....	29
Встановлення Python	29
Встановлення Visual Studio Code (VS code)	30
Написання першої програми.....	31
Відлагодження програми.....	33
Встановлення модулів Python	35
Приклад виконання лабораторної роботи № 6.....	37
Приклад виконання лабораторної роботи № 7	40
Рекомендована література	43

Загальні положення

Оформлення звіту

Звіт друкується з однієї сторони на аркушах формату А4 шрифтом 14pt з міжрядковим інтервалом 1,5. Дозволяється об'ємні завдання, тексти програм, розрахунки та інше друкувати читабельним шрифтом 12pt з міжрядковим інтервалом 1, використовуючи певні виділення курсивом чи жирним.

Звіт повинен містити титульну сторінку з номером практичної роботи. темою і іншими атрибутами згідно вимог навчального закладу. На наступній сторінці – мета роботи та персональне завдання згідно варіанту. Потім приводяться короткі теоретичні відомості необхідні для виконання завдання, хід виконання самої роботи з відповідними розрахунками, таблицями, схемами, програмами та іншим матеріалом. В кінці роботи необхідно написати висновок за отриманими результатами та аналізом їх достовірності.

Захист звіту

Захист звіту здійснюється студентом викладачу при наявності роздрукованого звіту оформленого згідно вимог, електронного варіанту відповідних схем, програм та інших допоміжних матеріалів, які можна виконати на комп'ютері для підтвердження отриманих в роботі результатів. При цьому студент повинен володіти теоретичним матеріалом по темі роботи, вміти практично виконати завдання та пояснити хід виконання.

Для виконання завдань потрібно використовувати ті засоби і функціонал Python, які вивчаються в конкретній лабораторній роботі. Наприклад, якщо метою завдання є перетворення стрічки в число, то це потрібно реалізувати самому, замість того, щоб використовувати існуючі функції Python, такі як `int()` чи `hex()`.

Оцінювання лабораторної роботи

Оцінка відмінно чи відповідна їй оцінка в іншій шкалі оцінювання ставиться студенту, який самостійно виконав поставлене у лабораторній роботі завдання в повній мірі, розроблена ним програма працює правильно, може пояснити написану програму,

правильно відповідає на контрольні запитання і володіє теоретичним матеріалом необхідним для виконання завдання поставленого в лабораторній. При цьому звіт по лабораторній оформлений у відповідності до вимог.

Оцінка добре чи відповідна їй оцінка в іншій шкалі оцінювання ставиться студенту, який самостійно виконав поставлене у лабораторній роботі завдання в достатній мірі, розроблена ним програма працює правильно, допускаються незначні помилки, які не впливають на правильний хід виконання програми. Може пояснити написану програму. Здебільшого правильно відповідає на контрольні запитання і володіє основним теоретичним матеріалом необхідним для виконання завдання поставленого в лабораторній. При цьому звіт по лабораторній оформлений у відповідності до вимог. Можливі незначні відхилення в оформленні, які не порушують загальну структуру звіту.

Оцінка задовільно чи відповідна їй оцінка в іншій шкалі оцінювання ставиться студенту, який самостійно виконав поставлене у лабораторній роботі завдання в необхідному мінімумі, розроблена ним програма працює правильно, допускаються помилки, які не впливають на загальний правильний хід виконання програми. Може пояснити написану програму. Правильно відповідає на більшість контрольних запитань і володіє в загальному теоретичним матеріалом необхідним для виконання завдання поставленого в лабораторній. При цьому звіт по лабораторній роботі оформлений у відповідності до вимог. Можливі відхилення в оформленні, але залишаються основні структурні розділи звіту.

Завдання до лабораторної роботи № 1.

Ознайомлення з основними властивостями мови Python.

Оператори умови та циклів.

1. Розробити програму, яка буде зчитувати дані з клавіатури (ріст в метрах і вагу в кілограмах) і обчислюватиме індекс маси тіла (w/h^{**2}). Скористуйтеся даними з таблиць для опису результату.

Вага	Індекс
недостатня	17.0 – 18.4
нормальна	18.5 – 24.9
надлишкова	25.0 – 29.9

Також програма повинна порахувати співвідношення між окружністю талії та стегон (t/s).

Вага	Жінка	Чоловік
нормальна	< 0.8	< 0.9
надлишкова	0.8 – 0.85	0.9 – 1
ожиріння	> 0.85	> 1

Потрібно вивести обидва результати на екран з їхньою назвою і описом. Після цього програма повинна знову запропонувати ввести дані.

2. Розробити програму, яка буде зчитувати температура з клавіатури в Fahrenheit, переводитиме її в Celsius. Програма повинна визначити і вивести на екран чи температура тіла в межах норми. Після цього потрібно знову запропонувати ввести дані.
3. Користувач вводить число N за допомогою клавіатури і програма повинна порахувати окремо суми всіх чисел від 1 до N які є: непарні, парні, діляться на 5, діляться одночасно на 3 і на 7.
4. Користувач вводить N за допомогою клавіатури і програма визначає чи це просте число
5. Користувач вводить з клавіатури дані про сторони трикутника, програма визначає його тип (рівносторонній, рівнобедрений, різносторонній, неможливий) і виводить результат на екран. Після цього програма пропонує ввести дані знову. Трикутник неможливий, якщо $a+b \leq c$ АБО $a+c \leq b$ АБО $b+c \leq a$.
6. Розробити програму, в якій користувач зможе ввести дані необхідні для обчислення периметру і площі кола. Після цього програма знову пропонує ввести дані.
7. Розробити програму, в якій користувач зможе ввести дані необхідні для обчислення периметру і площі прямокутника. Після цього програма знову пропонує ввести дані.

8. Розробити програму, в якій користувач зможе ввести дані необхідні для обчислення периметру і площі трикутника. Після цього програма знову пропонує ввести дані.
9. Розробити програму, в якій користувач зможе ввести дані необхідні для обчислення периметру і площі трапеції. Після цього програма знову пропонує ввести дані.
10. Написати програму-гру - вгадай число. Потрібно згенерувати випадкове число в межах від 1 до 100 включно. Далі програма повинна зчитувати число введене користувачем і порівняти його зі згенерованим. Потрібно вивести відповідне повідомлення - “число менше за загадане”, “більше за загадане” і “ви відгадали”. Якщо користувач не відгадав числа, то запропонувати йому ввести нове число.
11. Розробити програму, яка зчитає число введене користувачем і порахує кілька одиниць і нулів в двійковій формі числа.
Наприклад 100 в десятковій системі числення — це 01100100 в двійковій. В даному числі є 3 одиниці і 5 нулів.
12. Розробити програму, яка перевірити чи введене число є паліндромом (читається однаково в обидві сторони)
Наприклад: 123321 або 23432.
13. Перевірити чи число є розрідженим. Це двійкове число в якому немає двох послідовних одиниць.
Наприклад: число 170 в десятковій системі числення є розрідженими, бо в двійковий його вигляд такий — 10101010.
14. Розробити програму, яка зчитує координати точки (x;y) і визначає в якій чверті площини вона знаходиться. Після цього програма знову пропонує ввести дані.
Наприклад: точка (1;1) розміщена в першій чверті, а (-1, 2) в другій.
15. Розробити програму, яка зчитує рік введений користувачем і визначає чи він є високосним. Після цього програма знову пропонує ввести дані.
Високосний - це такий рік, що ділиться без остачі на 400 або не ділиться на 100 і ділиться на 4.
16. Розробити програму, яка зчитуватиме координати двох точок введених з клавіатури і знаходитиме відстань між ними.
17. Розробити програму, яка буде виводити на екран таблицю множення для числа, яке користувач введе з клавіатури.
Наприклад: 7
 $7 * 1 = 7$
 $7 * 2 = 14$
 $7 * 3 = 21$
...
18. Розробити програму, яка зчитуватиме числа введені користувачем і перемножатиме їх між собою.

Завдання до лабораторної роботи № 2.

Використання списків, кортежів, наборів, словників.

1. Порахувати скільки всього букв у введеному тексті і вивести 10 символів, які зустрічаються найчастіше. Реалізувати програму використовуючи структури даних та оператори циклів.
2. Використовуючи функцію `random.randint(a,b)` згенерувати послідовність з 20 елементів і посортувати її бульбашковим методом.
3. Розробити програму, яка перевірить чи введені `ip/mask` вірні і обчислить мережу.
Наприклад:
`ip = 192.168.1.100/24`
`netmask = 24 біти => 255.255.255.0`
`network = 192.168.1.0`
4. Програма пропонує користувачу ввести число і систему числення (двійкова, вісімкова чи шістнадцяткова) в яку потрібно його перевести. Результат переведення вивести на екран.
5. За допомогою функції `random.randint()` згенерувати список чисел і відсортувати його методом `sorted()`. Вивести список на екран і запропонувати користувачу ввести число зі списку. За допомогою двійкового пошуку знайти число у списку і вивести його індекс.
6. Розробити програму, яка згенерує число за допомогою функції `random.randint()`. Потрібно створити дві функції, перша перетворить число в текст, друга перетворить цей текст в число. Порівняти результат виконання другої функції зі згенерованим числом.
7. Написати програму, яка в запропонує користувачу вибрати фігуру і потім виведе її на екран.
Для квадрата — зчитати довжину сторони.
Для трикутник — зчитати довжину основи.
Наприклад:
Якщо користувач вказує вивести квадрат зі стороною 4 на екран, то програма повинна вивести наступне:

Якщо вказано вивести трикутник з основою 5, то результат повинен бути таким
*
**

8. Розробити програму, яка зчитає введену користувачем діагональ ромба і виведе його на екран за допомогою символів `*` і пробіл.
Наприклад:
діагональ = 5


```

*
* *
* * *
* *
*
```

9. Розробити програму, яка знаходитиме всі числа Армстронга до 10000.
Число Армстронга — це таке число, яке дорівнює сумі своїх цифр піднесених до степеня N, де N — кількість цифр в числі.
Наприклад:
Число 153 є числом Армстронга, бо $N = 3$ і $1^3 + 5^3 + 3^3 = 153$
Число 1634 також є числом Армстронга, бо $N = 4$ і $1^4 + 6^4 + 3^4 + 4^4 = 1634$
10. Використовуючи функцію `random.randint()` згенерувати список чисел і в ньому знайти 3 найбільше число.
Наприклад:
В списку чисел [37, 8, 33, 2, 4] — 8 є 3 найбільшим числом.
11. Написати програму, яка буде двічі зчитує текст введений користувачем і знаходить ті символи, які попадаються тільки в першому з текстів.
Наприклад:
text1 = "123456"
text2 = "1235"
результат - "46"
12. Розробити програму, яка зчитуватиме список чисел, які вводитиме користувач. Потрібно перевірити чи список відсортований і визначатиме порядок сортування: по зростанню чи спаданню.
Наприклад:
ввід1 — [1,3,5,9,7] — список не відсортований
ввід2 — [5,4,3] — список відсортований, порядок сортування — по спаданню.
13. Розробити програму, яка буде створювати 2 списки з 10 чисел за допомогою функції `random.randint(0,9)`. Потрібно, щоб програма знаходила переріз цих двох списків чисел (тобто, потрібно знайти спільні елементи).
Наприклад:
list1 = [4,9,5,7,2,5,9,3,7,6]
list2 = [2,0,8,6,4,7,5,9,4,4]
спільні елементи: [2, 4, 5, 6, 7, 9]
14. Розробити програму, яка буде змінювати порядок цифр в числі на протилежний.
Наприклад:
12376 => 67321
15. Розробити програму, яка буде за допомогою функції `random.randint(1, 20)` наповнювати список з 15 елементів. Після цього, програма повинна перемістити всі парні числа на початок списку, а непарні в кінець.
Наприклад:
input: [5,3,2,6,4,3]
output: [2,6,4,5,3,3]

Завдання до лабораторної роботи № 3.

Робота зі стрічками, регулярні вирази. Робота з функціями.

1. Реалізувати функцію, яка буде працювати аналогічно до `string/split`.
2. Реалізувати функцію, яка буде працювати аналогічно до `string/strip`.
3. Реалізувати функцію, яка буде працювати аналогічно до `string/find`.
4. Реалізувати функцію, яка змінюватиме порядок символів у стрічці на протилежний. Наприклад: `'abc' => 'cba'`
5. Реалізувати функцію, яка замінюватиме всі символи в стрічці, які відмінні від цифр і букв латинського алфавіту, на символ підкреслення.
6. Реалізувати функцію, яка видалятиме в стрічці ті символи, які йдуть послідовно і повторюються.
7. Реалізувати функцію, яка прийматиме необмежену кількість параметрів - стрічок і з'єднуватиме їх в одну стрічку: перший символ з першої стрічки + перший символ з другої стрічки + перший символ з третьої стрічки + другий символ з першої стрічки + ...
Наприклад: `'abc', 'def', 'ghi' => 'adgbehcfi'`
8. Написати програму, яка зчитує текст введений користувачем і шукає в ньому найдовшу послідовність символів, які повторюються.
Наприклад:
`aabbbbcccd => bbbb`
9. Реалізувати функцію, яка сортуватиме всі слова у стрічці в порядку спадання. Враховувати регістр букв не потрібно. Тобто В і в — це одне і теж саме.
Наприклад:
`Жили у бабусі два веселих гусі => бабусі веселих гусі два Жили у`
10. Розробити програму, яка буде розбивати введений текст на слова і змінюватиме порядок слів на протилежний.
Наприклад:
`Несе Галя воду, => гнеться коромисло воду, несе Галя.`
11. Реалізувати функцію, яка рахуватиме кількість різних типів символів в тексті: букви, числа, математичні операції, розділові знаки.
Наприклад:
текст — `К0л0б0к, який т4хенько п1ш0в у л1с, зл0вив т6м ли_си_цю і зайц8 !!!`.
результат:
39 букв (б, в, е, и, й, з, к, л, м, н, о, с, т, у, х, ц, ш, я, ю, ь)
10 цифр (0, 1, 4, 6, 8)
11 символів пунктуації і пробілів (, ! .)
12. Розробити програму, яка зчитуватиме текст і у всіх словах, що йдуть на початку речень замінить всі букви на великі.
Наприклад:
`Жили-були дід та баба. Була в них курочка ряба. => ЖУЛИ-БУЛИ дід та баба. БУЛА в них курочка ряба.`
13. Реалізувати функцію, яка рахуватиме кількість унікальних символів в тексті і повертатиме ті, що зустрічаються найчастіше і найрідше.
Наприклад:

ввід: Вовк Вовченко біг лісом, поки не зустрів їжака.

результат:

о - 5

ж - 1

- 14.Реалізувати функцію, яка знаходитиме числа у вказаному тексті і повертатиме їх у списку.

Наприклад:

тест: 150 років тому, люди мріяли полетіти до 4-ї планети Сонячної системи.

результат: [150, 4]

- 15.Розробити програму, яка зашифруватиме англійський текст, введений з клавіатури, шифром Цезаря і виводитиме його на екран. Потім програма повинна розшифрувати зашифрований текст і також вивести його на екран. Зсув алфавіту - 13 букв.

Наприклад:

abcdefg => nopqrst

- 16.Розробити програму, яка за допомогою регулярних виразів знайде в тексті всі номери телефонів, що відповідають вказаним шаблонам - +(xxx) xx xxxxxxxx або xxx xxxxxxxx.

Використати модуль re

Наприклад:

текст: Зв'яжіться з нами за тел. 065 1234567 або 0677654321

результат: ['065 1234567', '0677654321']

- 17.Розробити програму, яка за допомогою регулярних виразів знайде всі адреси електронної пошти у вказаному тексті. Використати модуль re

Наприклад:

текст: email1: bean@gmail.com, email2: mr.bean@outlook.com

результат: ['bean@gmail.com', 'mr.bean@outlook.com']

- 18.Розробити програму, яка за допомогою регулярних виразів знайде всі ір-адреси у вказаному тексті.

Наприклад:

input: dns servers: 192.168.0.1, 1.1.1.1, 8.8.8.8

result: ['192.168.0.1', '1.1.1.1', '8.8.8.8']

- 19.Розробити програму, яка за допомогою регулярних виразів змінить порядок букв у всіх слова у вказаному тексті.

Наприклад:

Hello world => world Hello

- 20.Розробити програму, яка за допомогою регулярних виразів видалить всі xml теги з вказаного тексту, при цьому не видаляючи їх вміст.

Наприклад:

input: <html><body><div>Hello</div><div>world</div></body></html>

result: Helloworld

Завдання до лабораторної роботи № 4.

Робота з файлами

Функції для отримання переліку файлів і каталогів знаходяться в модулі `os`:
`os.scandir()`, `os.walk()`

1. Розробити програму, яка знаходитиме найдовше і найкоротше слова у вказаному файлі. Також потрібно визначити середню довжину слів.
2. Розробити програму, яка рахуватиме кількість стрічок та букв у файлі для кожної стрічки них окремо.
3. Розробити програму, яка у файлі знаходитиме крапки і перед кожною з них вставлятиме новий рядок. Також потрібно зробити всі букви великими у слові, яке йде після крапки.
4. Розробити програму, яка зчитуватиме дані з клавіатури і записуватиме їх у кінець файлу додаючи до кожного з записів текучу дату. При повторному запуску програма повинна виводити три останні записи.
5. Розробити програму, яка зчитуватиме кількість файлів (N) з клавіатури. Потрібно створити N файлів іменуючи кожен з них `{DATE}_{I}.txt`, де `{DATE}` - текуча дата і час, а `{I}` змінюється в межах 1..N.
В кожен файл потрібно записати `{I}` в 2-му степені раз слово Hello.
6. Розробити програму, яка відсортує всі стрічки у файлі і на початку рядка додасть кількість слів у цьому рядку.
7. Розробити програму, яка знайде всі рядки у файлі, які містять цифри і замінить їх на подібні букви.

Наприклад:

0 => O
1 => I
2 => Z
3 => E
4 => H
5 => S
6 => G
7 => T
8 => V
9 => J

8. Розробити програму, яка пострічково зчитуватиме з лог файлу записи і підраховуватиме скільки байт було скачано кожним з користувачів індивідуально і виводитиме 2 з максимальним обсягом.

Формат файлу (всі поля розділені пробілами):

Дата	Користувач	Сайт	Розмір
2024-01-01	Ivan	https://google.com	10000
2024-01-02	Oleg	https://ukr.net	8000

9. Розробити програму, яка зчитуватиме текст з клавіатури і для кожного слова створюватиме свій підкаталог. Перед створенням каталогів, їхні назви слід очистити від усіх символів які не є літерами алфавіту та числами.
Наприклад:
input: Автобус їде по алеї.
result: 'Автобус' - назва першого підкаталога. Каталог 'їде' повинен бути підкаталогом каталога 'Автобус'. Підкаталог 'по' має бути підкаталогом каталога 'їде' і т.д.
10. Розробити програму, яка знаходитиме всі файли у вказаному каталозі і підкаталогах, дані яких містять текст введений користувачем.
11. Розробити програму, яка буде виводити список усіх файлів і підкаталогів у вказаному каталозі, ім'я яких містить вказаний текст
12. Розробити програму, яка порахує загальний розмір усіх файлів у вказаному каталозі і його підкаталогах, а також знайде файли з максимальним, мінімальним розміром і загальну кількість файлів та каталогів.

Завдання до лабораторної роботи № 5.

Робота з процесами

1. Розробити програму, яка буде запускати `systeminfo.exe`, як дочірній процес. Потрібно з даних, які виводить ця програма, отримати наступну інформацію і вивести її на екран
 - Host Name
 - OS Name
 - OS Version
 - OS ManufacturerВикористати `subprocess.run`. Передати параметри `stdin=subprocess.PIPE`, `stderr=subprocess.DEVNULL`
2. Розробити програму, яка буде запускати `tasklist.exe`, як дочірній процес. Потрібно з даних, які виводить ця програма, отримати наступну інформацію і вивести її на екран
 - Загальна кількість процесів
 - Загальним об'єм пам'яті, яку вони використовуютьВикористати `subprocess.run`. Передати параметри `stdin=subprocess.PIPE`, `stderr=subprocess.DEVNULL`
3. Розробити програму, яка буде визначати кількість доступних процесорів і запускати таку ж кількість дочірніх процесів. Кожен процес повинен вивести повідомлення "Hello from {PROCESSID}. My parent is {PARENTID}", де {PROCESSID} – ідентифікатор процесу, а {PARENTID} – ідентифікатор батьківського процесу. Використати `multiprocessing.cpu_count`, `os.getppid()`, `os.getpid()`
4. Розробити програму, яка буде рахувати суму чисел від 1 до мільярда за допомогою 10 паралельних процесів. Результат обчислень потрібно передавати в основний процес через чергу (Queue) Використати модуль `multiprocessing`
5. Розробити програму, яка створюватиме 10 підпроцесів, кожен з яких отримуватиме об'єкт Pipe з основного процесу. Підпроцес повинен зчитувати число з Pipe, підносити його до квадрату і відсилати назад в Pipe. Основний процес повинен генерувати випадкові числа і записувати їх в Pipe, очікувати результат і виводити його на екран разом з ідентифікатором підпроцесу, який виконав обчислення. Використати модуль `multiprocessing`
6. Розробити програму, яка буде створювати 10 процесів, кожен з яких буде виводити на екран свій PID. За допомогою синхронізації, зробити так, щоб лише 1 процес в конкретний момент часу виводив повідомлення. Використати модуль `multiprocessing`
7. Розробити програму, яка буде виводити інформацію про операційну систему, таку, як:

- кількість процесорів (фізичних і логічних) та їхню частоту
- рівень завантаження
- загальну кількість пам'яті
- час запуску ОС (відобразити в локальному часі)
- кількість запущених процесів
- імена користувачів, які запустили ці процеси

Використати модуль `psutil`. Ознайомитися з класом `psutil.Process`.

Використати функції `psutil.boot_time()`, `psutil.process_iter()`, `psutil.cpu_count()`, `psutil.cpu_freq()`, `psutil.getloadavg()`, `psutil.virtual_memory()`.

Для перетворення UNIX часу в локальний використати функцію `time.ctime()`.

8. Розробити програму, яка виведе список користувачів, які на даний момент мають запущені процеси.

Після вводу імені користувача, програма має вивести на екран всі його процеси з наступною інформацією:

- номер процесу (`pid`)
- ім'я (`name`)
- шлях до програми (`exe`)
- текучий каталог (`cwd`)
- час створення (`create_time`, відобразити в локальному часі)
- ім'я користувача (`username`)
- кількість мережевих з'єднань (`net_connections`)

Використати функцію `psutil.process_iter()`.

Завдання до лабораторної роботи № 6

Візуалізація даних

1. Завантажити csv файл з цінами на товари і послуги з <https://stat.gov.ua/uk/datasets/zminy-tsin-taryfiv-na-spozhyvchi-tovary-posluhy>
Відобразити графіки для наступних показників та порівняти їхні значення

Показник: Середні споживчі ціни на товари (послуги)

Територіальний розріз:

- Івано-Франківська
- Тернопільська

Тип товарів і послуг:

- Хліб пшеничний з борошна вищого гатунку
- Хліб житній, житньо-пшеничний

Період часу: Весь період

2. Завантажити csv файл з цінами на товари і послуги з <https://stat.gov.ua/uk/datasets/zminy-tsin-taryfiv-na-spozhyvchi-tovary-posluhy>
Відобразити графіки для наступних показників та порівняти їхні значення

Показник: Середні споживчі ціни на товари (послуги)

Територіальний розріз:

- Тернопільська
- Хмельницька

Тип товарів і послуг:

- Бензин А-95
- Дизельне пальне
- Газ скраплений для автомобілів

Період часу: Весь період

3. Завантажити csv файл з цінами на товари і послуги з <https://stat.gov.ua/uk/datasets/zminy-tsin-taryfiv-na-spozhyvchi-tovary-posluhy>
Відобразити графіки для наступних показників та порівняти їхні значення

Показник: Середні споживчі ціни на товари (послуги)

Територіальний розріз:

- Рівенська
- Тернопільська

Тип товарів і послуг:

- Яловичина
- Свинина
- Філе куряче

Період часу: Весь період

4. Завантажити csv файл з цінами на товари і послуги з <https://stat.gov.ua/uk/datasets/zminy-tsin-taryfiv-na-spozhyvchi-tovary-posluhy>

Відобразити графіки для наступних показників та порівняти їхні значення

Показник: Середні споживчі ціни на товари (послуги)

Територіальний розріз:

- Тернопільська

- Україна

Тип товарів і послуг:

- Рис

- Борошно пшеничне

- Крупи гречані

Період часу: Весь період

5. Завантажити json файл з інформацією про викрадені або загублені мобільні телефони з <https://data.gov.ua/dataset/5c6c156f-21ee-42cd-8da3-dcde6828be97>

ID - Унікальний ідентифікатор запису

OVD - Назва підрозділу, що зареєстрував інформацію

INSERT_DATE - Дата внесення інформації

IMEI - Унікальний ідентифікатор телефону

NZ - Марка/модель

NK - Номер реєстрації в журналі єдиного обліку підрозділу, що зареєстрував інформацію

DK - Дата реєстрації в журналі єдиного обліку

DTL - інші деталі

6. Завантажити csv файл із справами про банкрутство з <https://data.gov.ua/en/dataset/vidomosti-pro-spravi-pro-bankrutstvo-1>
Відобразити та порівняти графіки з кількістю записів по значеннях в полі type за останні пів-року та порівняти їх.

Поля (колони) з файлу

date:

court_name:

- Господарський суд Тернопільської області

- Господарський суд Львівської області

type:

- Повідомлення про визнання боржника банкрутом і введення процедури погашення боргів боржника

- Повідомлення про відкриття провадження у справі про банкрутство

7. Завантажити csv файл із даними зі статистикою праці з <https://stat.gov.ua/uk/datasets/obstezhennya-pidpryyemstv-iz-pytan-statystyky-pratsi-0>
Відобразити графіки для наступних показників та порівняти їхні значення

Періодичність: місячна

Вид економічної діяльності:

- Сільське господарство, лісове господарство та рибне господарство

- Інформація та телекомунікації

- Будівництво

Показник:

- Середньооблікова кількість штатних працівників

- Середня заробітна плата (номінальна) штатного працівника

8. Завантажити csv файл із даними про діяльність закладів освіти з

<https://stat.gov.ua/uk/datasets/merezha-ta-diyalnist-zakladiv-osvity-0>

Відобразити графіки для наступних показників та порівняти їхні значення

Розріз: Інформаційні технології

Період часу: Весь період

Освітній ступінь:

- Інженерія програмного забезпечення (бакалавр)

- Комп'ютерні науки (бакалавр)

- Кібербезпека (бакалавр)

Показник: Кількість студентів у закладах вищої освіти

9. Завантажити csv файл із даними зі статистикою праці з

<https://stat.gov.ua/uk/datasets/zminy-tsin-na-rynku-zhytla-0>

Відобразити графіки для наступних показників та порівняти їхні значення

Показник: Індекс цін на житло

Періодичність: Квартальна

Період часу: Весь період

Інші розрізи:

- квартири однокімнатні на первинному ринку

- квартири однокімнатні на вторинному ринку

Базисний період:

- до попереднього кварталу

- До відповідного кварталу попереднього року

10. Завантажити csv файл із змінами цін у будівництві з

<https://stat.gov.ua/uk/datasets/zminy-tsin-u-budivnytstvi-0>

Відобразити графіки для наступних показників та порівняти їхні значення

Базисний період:

- До попереднього місяця

- До відповідного місяця попереднього року

Інші розрізи:

- Житлові будинки

- Нежитлові будівлі

Період часу: Весь період

Завдання до лабораторної роботи № 7

Створення веб-сервісів за допомогою web-фреймворка Flask

1. Створити web-додаток для тестування знань мови програмування Python. Питання для тесту знаходяться в модулі `data.questions`. При надсиланні відповідей програма повинна порахувати оцінку і відобразити відповідне повідомлення на сторінці.
Дані в цей шаблон потрібно передати у змінній `questions`. Вона повинна містити список елементів, у формі словників, кожен з яких описує запитання, відповіді до нього і тип запитання: `single` (можлива єдина відповідь), `multi` (можливо багато відповідей). Відповідь має бути теж у формі словника: запитання => булеве значення.

Вимоги

Каталог з файлами - `task_python_test`.

Шаблони:

- сторінка тесту: `templates/index.html`
дані приходять в полях форми. Поле має таку ж назву, як і запитання в тексті. З прикладу нижче, відповідь на запитання 'Як створити змінну?' буде міститися в полі з такою ж назвою - 'Як створити змінну?'. Зверніть увагу, що запитання з типом `'multi'` можуть містити декілька значень і тому відповідь буде міститися в списку.

приклад даних для шаблону:

```
questions = [  
    {  
        "question": "Як створити змінну?",  
        "options": {  
            "a := 10": False,  
            "a << 10": False,  
            "a = 10": True,  
            "a <- 10": False,  
        },  
        "type": "single"  
    },  
    {  
        "question": "Як створити цикл?",  
        "options": {  
            "використати оператор foreach": False,  
            "використати оператор while": True,  
            "використати оператор for": True,  
        },  
        "type": "multi"  
    },  
    ...  
]
```

]

- сторінка результатів: templates/result.html
приклад даних для шаблону:
data={
 "grade": 98.2
}

2. Створити web-додаток, який буде відображати форму реєстрації на сайті, яка складається з 3 кроків. Кожен крок відображається на окремій сторінці.

Сторінка №1

поля: прізвище, ім'я, рік народження
кнопки: 'Далі'

Сторінка №2:

поля: місто проживання, вулиця
кнопки: 'Назад', 'Далі'

Сторінка №3:

поля: професія, хобі
кнопки: 'Назад', 'Надіслати'

Сторінка №4: відображає всю введену інформацію

Коли сайт завантажується вперше, відображається перша сторінка, на якій всі поля будуть порожні. Коли користувач заповнює їх і натискає 'Далі', заповнені дані відсилаються на сервер і відображається наступна сторінка. На сервері дані мають зберегтися в сесію. Якщо користувач натисне 'Назад', то має відобразитися попередня сторінка з даними, які користувач ввів. На третьому кроці, після натискання кнопки 'Надіслати', відображається вся введена інформація.

Вимоги

Каталог з файлами - task_registration_form.

Шаблони:

- сторінка продуктів: templates/page1.html
дані приходять в полях форми: name, surname, age

приклад даних для шаблону:

```
data = {  
    "name": "Іван",  
    "surname": "Іваненко",  
    "age": "22"  
}
```

- сторінка продуктів: templates/page2.html
дані приходять в полях форми: city, street

приклад даних для шаблону:

```
data = {  
    "city": "Тернопіль",
```

```
"street": "Центральна"
}
```

- сторінка продуктів: templates/page3.html
дані приходять в полях форми: profession, hobby

приклад даних для шаблону:

```
data = {
    "profession": "Рибак",
    "hobby": "Програмування на Python"
}
```

- сторінка продуктів: templates/result.html
дані приходять в полях форми: profession, hobby

приклад даних для шаблону:

```
data = {
    "name": "Іван",
    "surname": "Іваненко",
    "age": "22",
    "city": "Тернопіль",
    "street": "Центральна",
    "profession": "Рибак",
    "hobby": "Програмування на Python"
}
```

3. Створити web-додаток, де буде відображатися сторінка з різними продуктами і їхніми цінами. Дані по продуктах знаходяться в модулі data.products. Користувач зможе ввести кількість продуктів, які він хоче придбати. Після натискання кнопки 'Купити', інформація відправляється на сервер. На сервері потрібно зберегти інформацію про продукти і їхню кількість в сесію, також треба порахувати суму за кожен продукт і загальну суму. Цю інформацію потрібно ввести користувачу. На сторінці результату є кнопка 'Назад', при натисканні на яку користувач повертається на сторінку продуктів, яка заповнюється даними, які збережені в сесії.

Вимоги

Каталог з файлами - task_products.

Шаблони:

- сторінка продуктів: templates/index.html
дані приходять в полях форми. Поле має таку ж назву, як і назва продукту. З прикладу нижче, кількість продуктів "яблука" буде міститися в полі з такою ж назвою - 'яблука'.

приклад даних для шаблону:

```
products = [
    {
        "name": "яблука",
        # назва продукту
```

```

        "price": "10",
        "quantity": 0
    },
    ....
]

```

° сторінка результату: templates/result.html
 приклад даних для шаблону:
 products = [
 {
 "name": "яблука",
 "price": "10",
 "quantity": 2,
 "total": 20
 },
 {
 "name": "груші",
 "price": "13",
 "quantity": 2,
 "total": 26
 },

],
 total = 46

ціна за кілограм яблук
 # кількість кілограмів
 # ціна за яблука
 # ціна за груші
 # загальна ціна за продукти

4. Створити web-додаток, який виводитиме таблицю на екран. Дані для таблиці знаходяться в модулі data.students. На сторінці потрібно також показувати текстове поле. Якщо користувач введе текст в цьому полі і натисне “Фільтрувати”, то потрібно вивести лише ті рядки з таблиці, де колонка ПІБ містить введений текст.

Вимоги

Каталог з файлами - task_table_filter.

Шаблони:

- ° сторінка перегляду: templates/index.html
 дані приходять в полях форми:
 - text – введений текст

приклад даних для шаблону:

```

students = [
    {
        "fullname": "Іваненко Олексій Андрійович",
        "group": "АКІТ-21",
        ....
    },
    ....
],

```

```
text = "іване"
```

5. Створити web-додаток, який виводитиме таблицю на екран. Дані для таблиці знаходяться в модулі `data.students`. Якщо користувач клікне на заголовок таблиці ПІБ, то рядки в таблиці посортуються в порядку спадання по цій колонці. Якщо користувач ще раз клікне на цей заголовок, то рядки посортуються в порядку спадання

Вимоги

Каталог з файлами - `task_table_sort`.

Шаблони:

- сторінка перегляду: `templates/index.html`
дані приходять в полях форми:
 - `sort` – напрямок сортування: `asc` (А-Я), `desc` (Я-А)

приклад даних для шаблону:

```
students = [  
    {  
        "fullname": "Іваненко Олексій Андрійович",  
        "group": "AKIT-21",  
        .....,  
    },  
    .....,  
],  
sort_dir = "asc"    # якщо з форми передано asc, то тут має бути desc. І  
навпаки
```

6. Створити web-додаток, який виводитиме таблицю з даними на екран. Дані для таблиці знаходяться в модулі `data.students`. Також потрібно додати форму, за допомогою якої користувач зможе додати рядок до таблиці. Крім цього треба додати можливість видалення рядка.

Вимоги

Каталог з файлами - `task_table_manage`.

Шаблони:

- сторінка перегляду: `templates/index.html`
якщо в даних форми є поле `'add_student'`, то в наступних полях форми містяться дані про студента, якого потрібно додати: `fullname`, `group`, `age`, `gender`, `grade`
якщо в даних форми є поле `'delete_student'`, то це поле містить індекс запису, який потрібно видалити з списку студентів

приклад даних для шаблону:

```
students = [  
    {  
        "fullname": "Іваненко Олексій Андрійович",
```

```

        "group": "AKIT-21",
        .....
    },
    .....
]
```

7. Створити web-додаток, який дозволить завантажувати будь-яку кількість файлів і додаватиме їх у zip-архів. Цей архів потрібно повернути користувачу.

Файли надіслані користувачем отримайте з поля форми — files. Створіть zip файл на диску, додайте в нього файли надіслані користувачем і відішліть файл за допомогою функції flask.send_file().

Вимоги

Додаткові модулі:

- zipfile – модуль для роботи zip-архівами

Каталог з файлами - task_zip_files.

Шаблони:

- сторінка завантаження: templates/index.html
дані приходять в полях форми:
 - files – завантажені файли

Допоміжна інформація

Додати файли в zip-архів можна використавши наступний код
import zipfile

```

files = .... # файли завантажені користувачем
zip_filename = 'data.zip'
with zipfile.ZipFile(zip_filename, 'w') as z:
    for file in files:
        if file:
            z.writestr(file.filename, file.read())
```

8. Створити web-додаток, який дозволить завантажити малюнок і оберне його на 90, 180 або 270 градусів, відобразити зліва направо або навпаки (випадковим чином). Обернений файл потрібно повернути користувачу. Зробіть це за допомогою функції flask.send_file().

Вимоги

Додаткові модулі:

- pillow – модуль для роботи з малюнками

Каталог з файлами - task_transform_image.

Шаблони:

- сторінка завантаження: templates/index.html

дані приходять в полях форми:

- image – завантажений малюнок

Допоміжна інформація

Для того, щоб зчитати завантажений файл, як малюнок, можна використати наступний код

```
from PIL import Image
```

```
file = ..... # завантажений файл
```

```
img = Image.open(file)
```

```
# обертання малюнка на 90 градусів
```

```
img = img.transpose(Image.Transpose.ROTATE_90)
```

```
image_filename = 'image.jpg'
```

```
with open(image_filename, 'wb') as f:
```

```
    img.save(f, format='jpeg')
```

9. Створити web-додаток, який буде запитувати ім'я користувача і його пароль. Користувачі знаходяться в модулі data.auth_users. Це список має такий вигляд:

```
[
    {
        'name': 'ivan',
        'pass': 'Ivan#2023!',
        'fullname': 'Іван Іванович Шевченко',
        'role': 'admin'
    },
    {
        'name': 'olena',
        'pass': 'Olena@2023$',
        'fullname': 'Олена Петрівна Коваленко',
        'role': 'user'
    },
    ...
]
```

Вам потрібно перевірити чи введене ім'я існує в цьому списку і чи введений пароль співпадає. Якщо такого користувача немає, то потрібно відобразити сторінку з помилкою. Якщо ж користувач існує, то залежно від його ролі (admin чи user) відобразити відповідну сторінку. Для користувача з роллю admin потрібно відобразити список імен існуючих користувачів.

Вимоги

Каталог з файлами - task_check_auth.

Шаблони:

- сторінка входу: templates/login.html

дані приходять в полях форми: username, password

приклад даних для шаблону, якщо дані на формі введені неправильно:
error = "Доступ заборонено!"

- сторінка для ролі admin: templates/admin.html

приклад даних для шаблону:

fullname = "Іван Іванович Шевченко",

all_users = ["Іван Іванович Шевченко", "Олена Петрівна Коваленко"]

- сторінка для ролі user: templates/user.html

приклад даних для шаблону:

fullname = "Олена Петрівна Коваленко"

10. Створити web-додаток, який буде звертатися до web-сервіса, який надає інформацію про землетруси. Користувач вказує початкову і кінцеву дати, і сайт повинен робити http-запит до
["https://earthquake.usgs.gov/fdsnws/event/1/query?format=geojson&starttime={start_time}&endtime={end_time}"](https://earthquake.usgs.gov/fdsnws/event/1/query?format=geojson&starttime={start_time}&endtime={end_time})

де {start_time} – початковий час, наприклад 2024-01-01T10:00:00,

{end_time} – кінцевий час, наприклад 2024-01-02T10:00:00,

Після виконання запиту потрібно перевірити response.status_code. Якщо він не дорівнює 200, то сталася помилка.

Результат прийде у форматі JSON і буде мати наступний вигляд (частковий результат)

```
{
  "type": "FeatureCollection",
  "metadata": {...},
  "features": [
    {
      "type": "Feature",
      "properties": {
        "mag": 2.1,
        "place": "53 km NNE of Chase, Alaska",
        "time": 1729865629156,
        "tsunami": 0
      }
    }
  ]
}
```

Потрібно з цього результату відобразити наступні дані:

result

features

properties

mag – мігнітуда (сила) землетрусу

place – місце де він відбувся

time – часовий відбиток в мікросекундах. Поділіть це число на 1000 і перетворіть його в стрічку часу за допомогою time.ctime()

tsunami – 1 або 0 в залежності чи є загроза цунамі чи ні.

Вимоги

Каталог з файлами - task_earthquakes.

Шаблони:

- сторінка запиту: templates/index.html
дані приходять в полях форми: start_time, end_time

- сторінка результатів: templates/result.html

приклад даних для шаблону:

```
earthquakes=[  
  {  
    "place": "53 km NNE of Chase, Alaska",  
    "mag": 2.1,  
    "time": "Fri Oct 25 17:13:49 2024",  
    "tsunami": 0  
  },  
  .....  
]
```

приклад даних для шаблону, якщо під час отримання даних виникла помилка:
error = "Не вдалося отримати дані."

11. Створити web-додаток, де можна буде вказати назву сайту або ір-адресу і ваш сайт запустити програму ping до нього. Результат виведеться користувачу.

Вимоги

Каталог з файлами - task_ping.

Шаблони:

- сторінка запиту: templates/index.html
дані приходять в полях форми:
 - host – інтернет адреса

приклад даних для шаблону:

```
output="Pinging 127.0.0.1 ..."  
host="127.0.0.1"
```

12. Створити web-додаток, який буде відображати вміст каталога (файли і під-каталоги). Якщо користувач клікне на підкаталог, то потрібно відобразити вміст цього під-каталога. Вверху списку файлів і під-каталогів потрібно додати посилання на батьківський каталог.

Вимоги

Каталог з файлами - task_filemanager.

Шаблони:

- сторінка запиту: templates/index.html
дані приходять в полях форми:
 - directory — каталог введений користувачем

приклад даних для шаблону:

```
items=[
    {
        "name": "...назва каталога...",
        "type": "directory"
    },
    {
        "name": "...назва файла...",
        "type": "file"
    }
],
directory="тут має бути шлях до текущего каталога",
parent_directory="тут має бути шлях до батьківського каталога"
```

Корисні посилання

1. Функція зберігання даних в zip-архів, ZipFile.writestr -
<https://docs.python.org/uk/3/library/zipfile.html#zipfile.ZipFile.writestr>
2. Функція обертання малюнка, Image.transpose -
<https://pillow.readthedocs.io/en/stable/reference/Image.html#PIL.Image.Image.transpose>

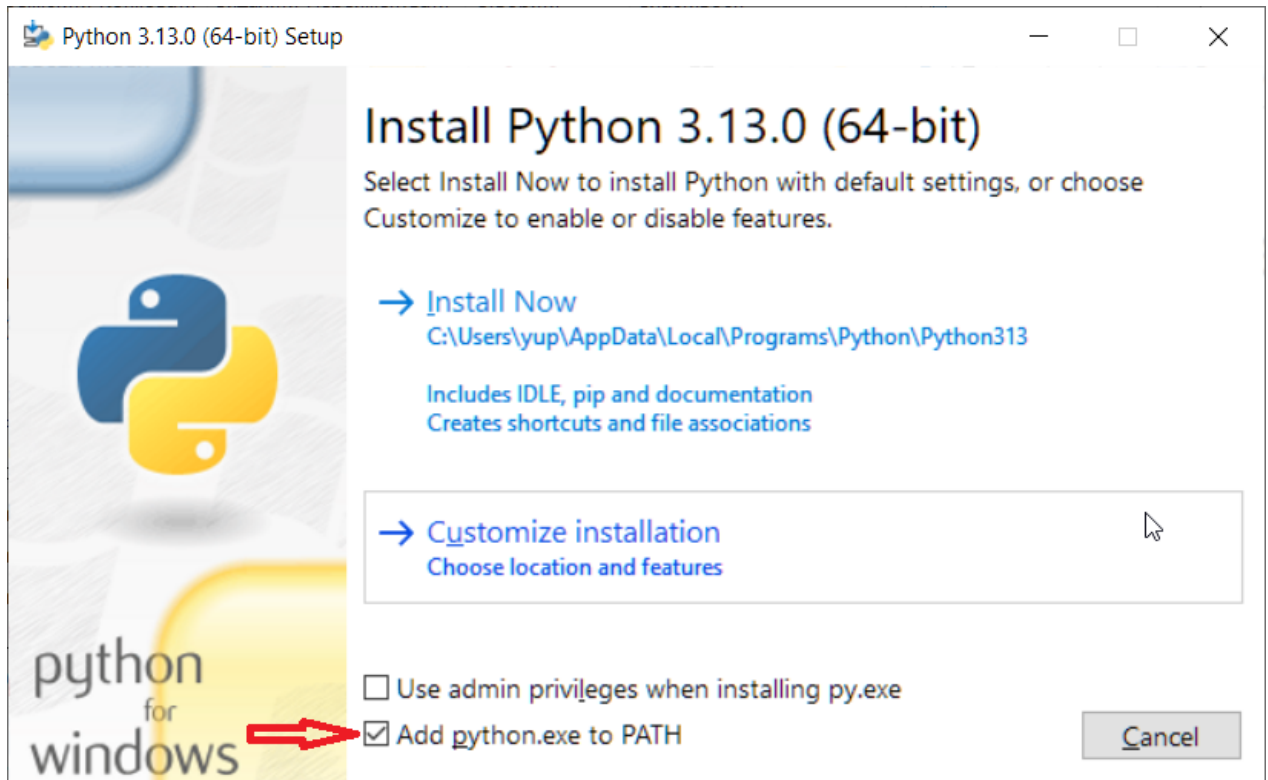
Налагодження середовища розробки

Встановлення Python

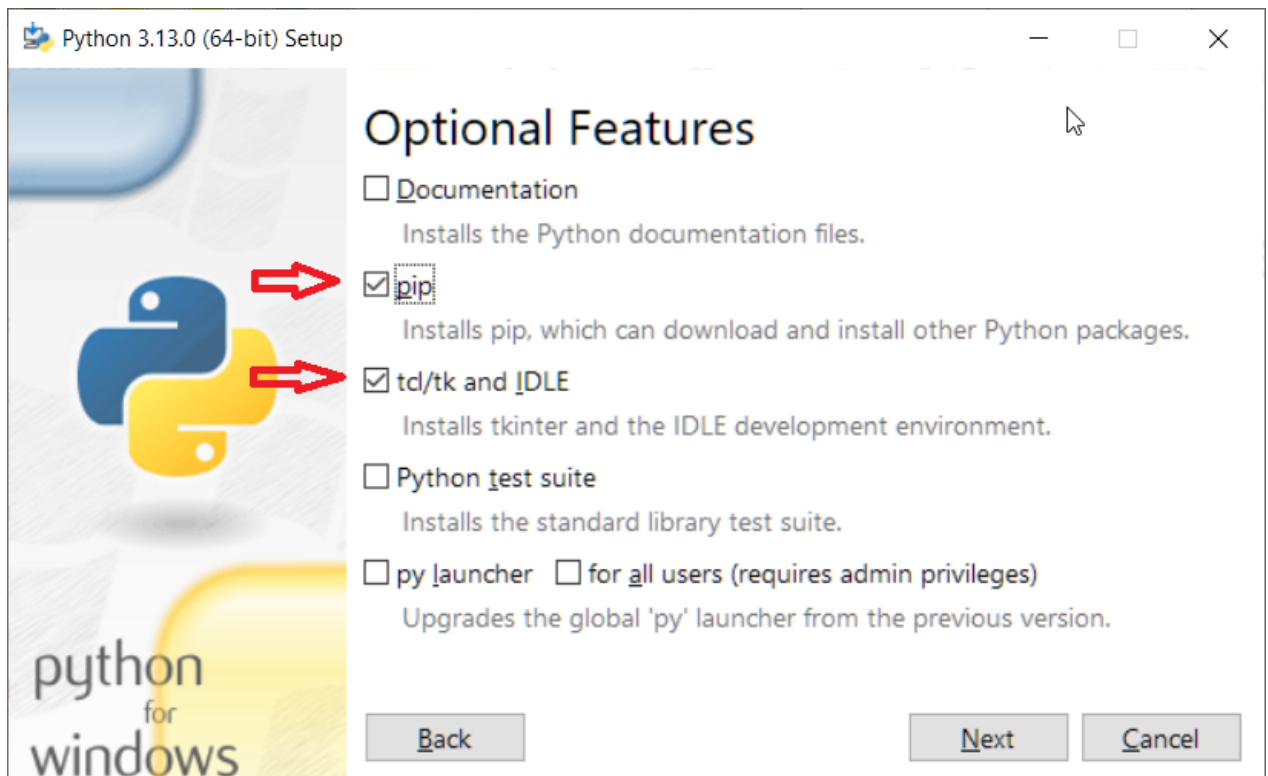
Завантажте пакет встановлення Python з <https://www.python.org/downloads/> і запустіть встановлення.

Переконайтеся, що всі зміни, позначені червоними стрілками, виконані.

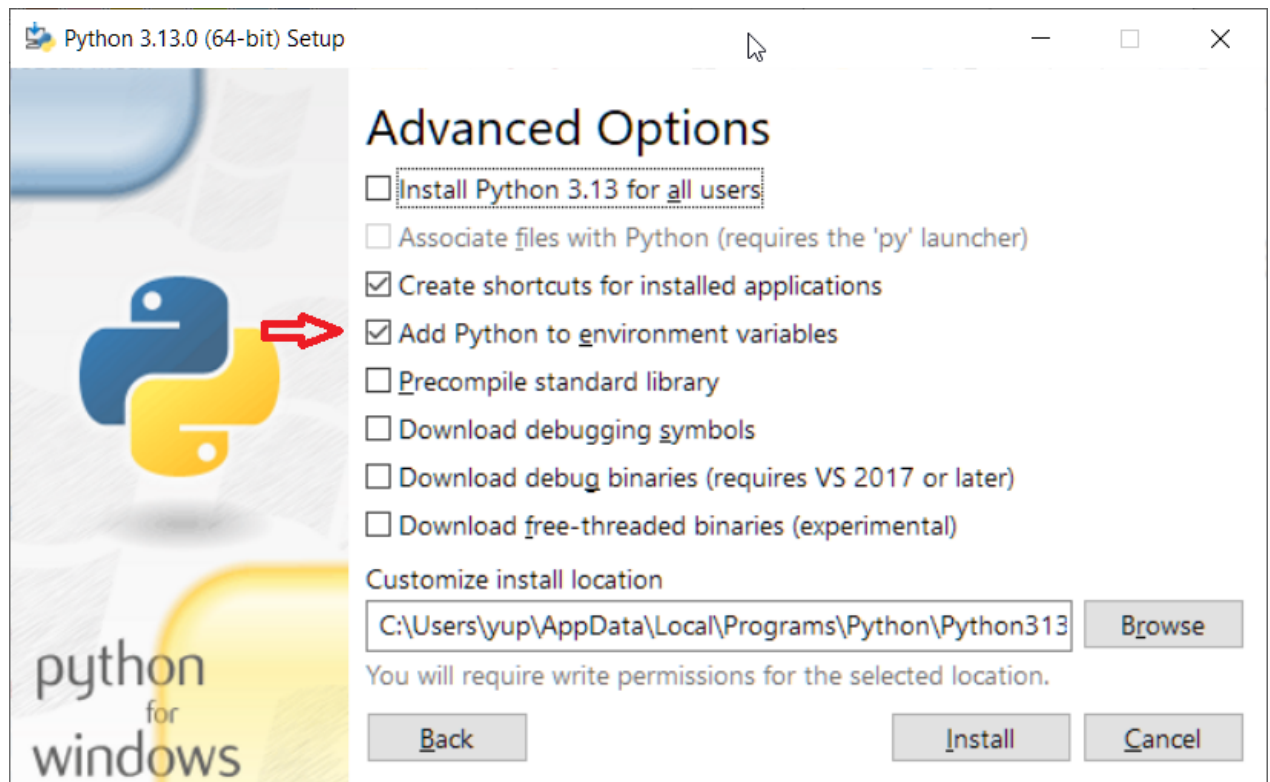
1.



2.



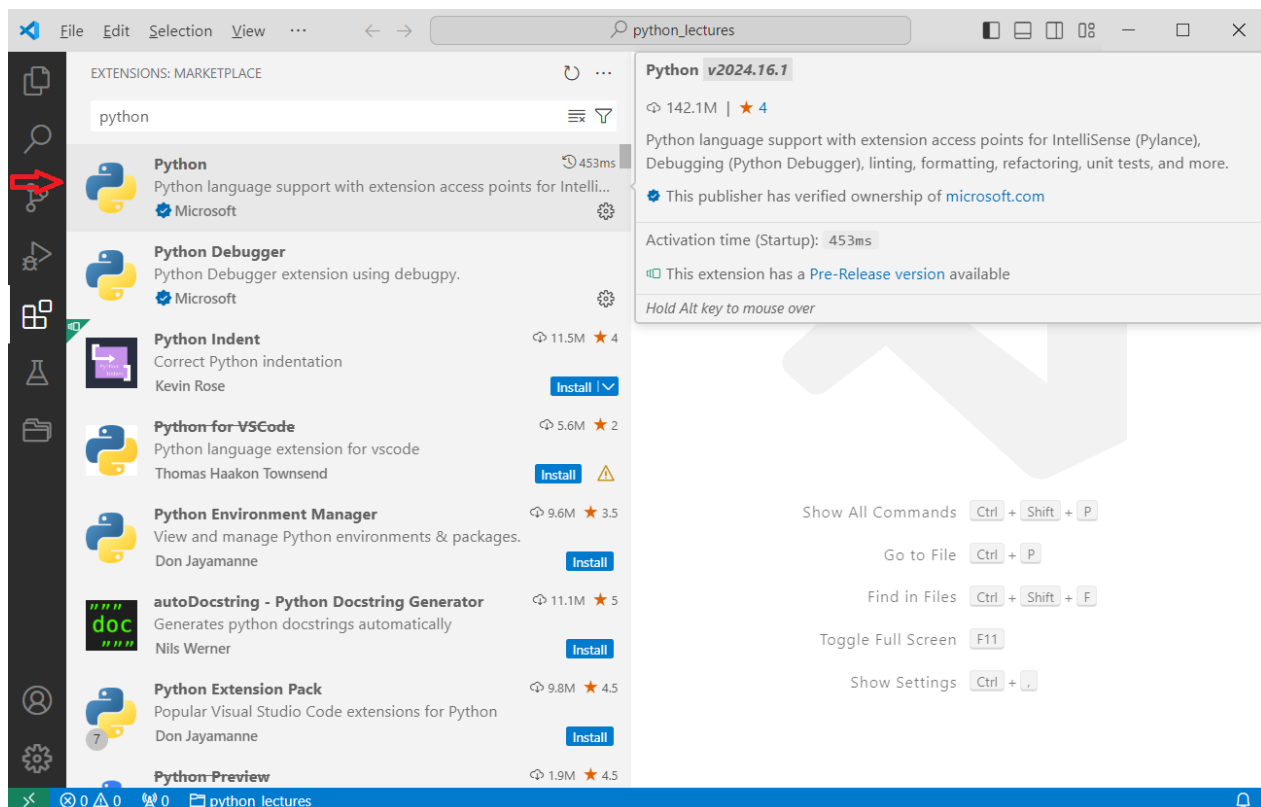
3.



Встановлення Visual Studio Code (VS code)

Завантажте пакет встановлення VS code <https://code.visualstudio.com/download> і запустіть встановлення.

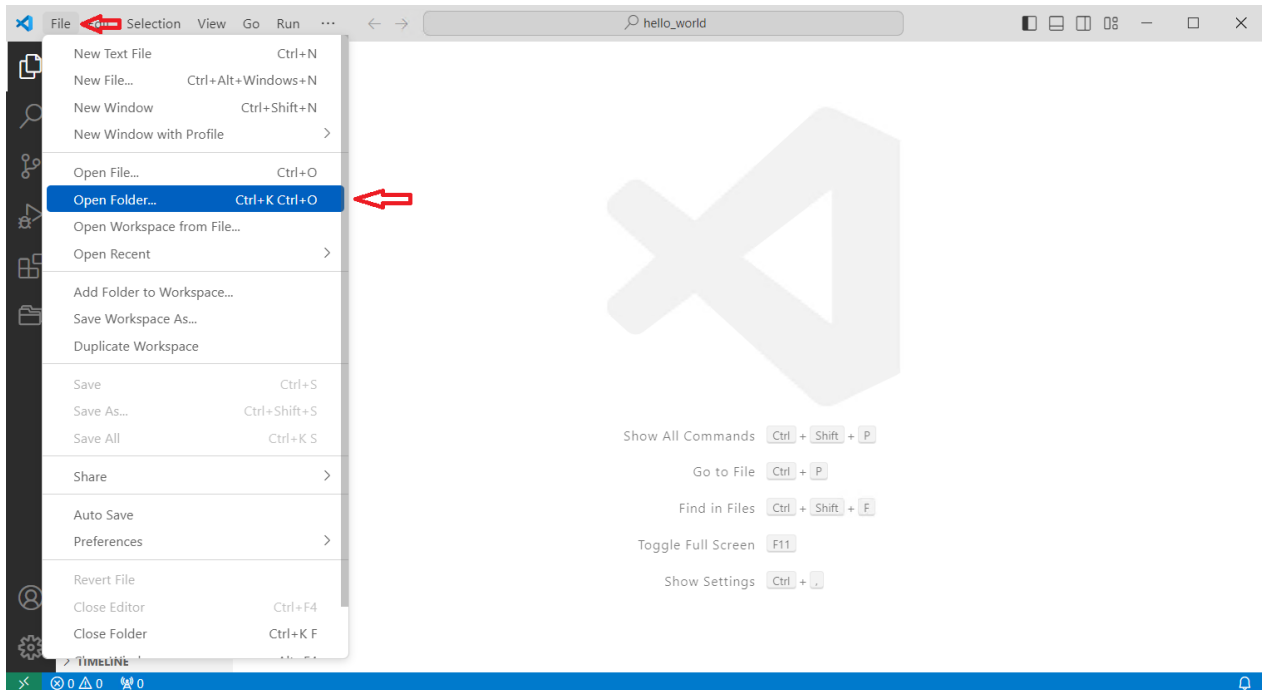
Після встановлення запустіть VS code і встановіть модуль підтримки програм на Python



Написання першої програми

- Запустіть VS code.
- Виберіть меню File > Open Folder або Файл > Відкрити каталог і вкажіть каталог, в якому ви будете зберігати свої програми.

Наприклад C:\python_dev

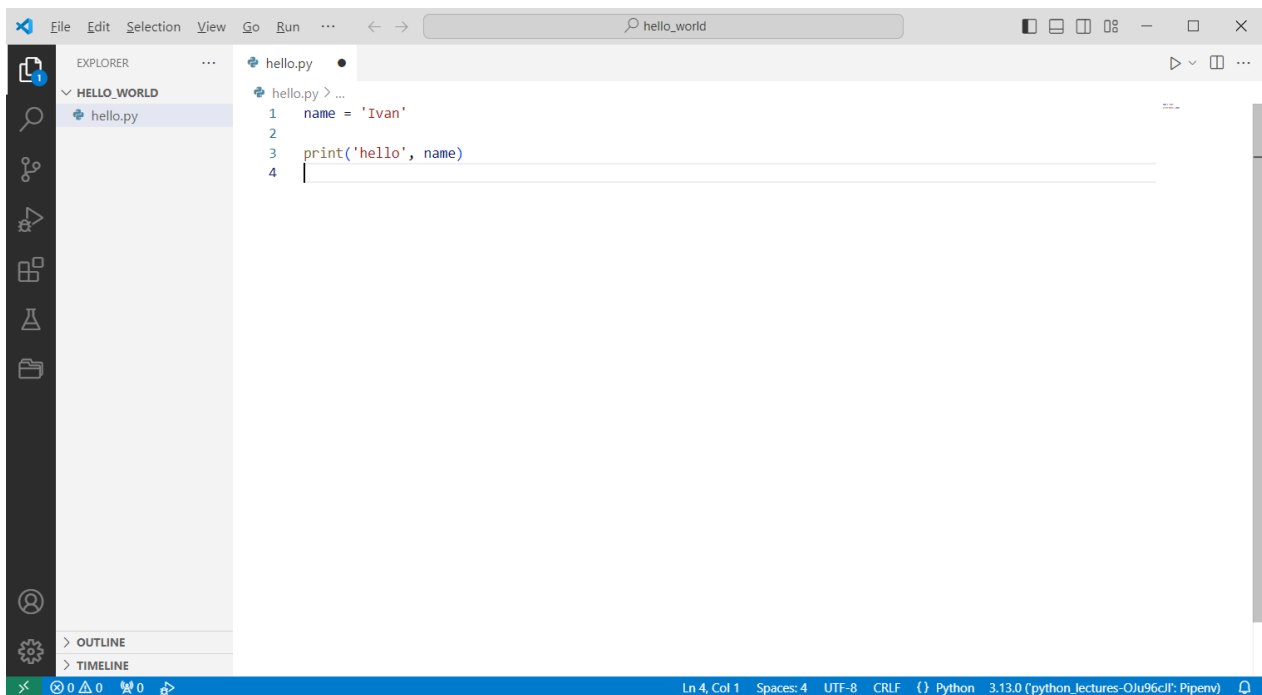


- Далі виберіть в меню File > New File або Файл > Новий файл
- вкажіть йому ім'я hello.py і введіть наступний код

```
name = 'Ivan'
```

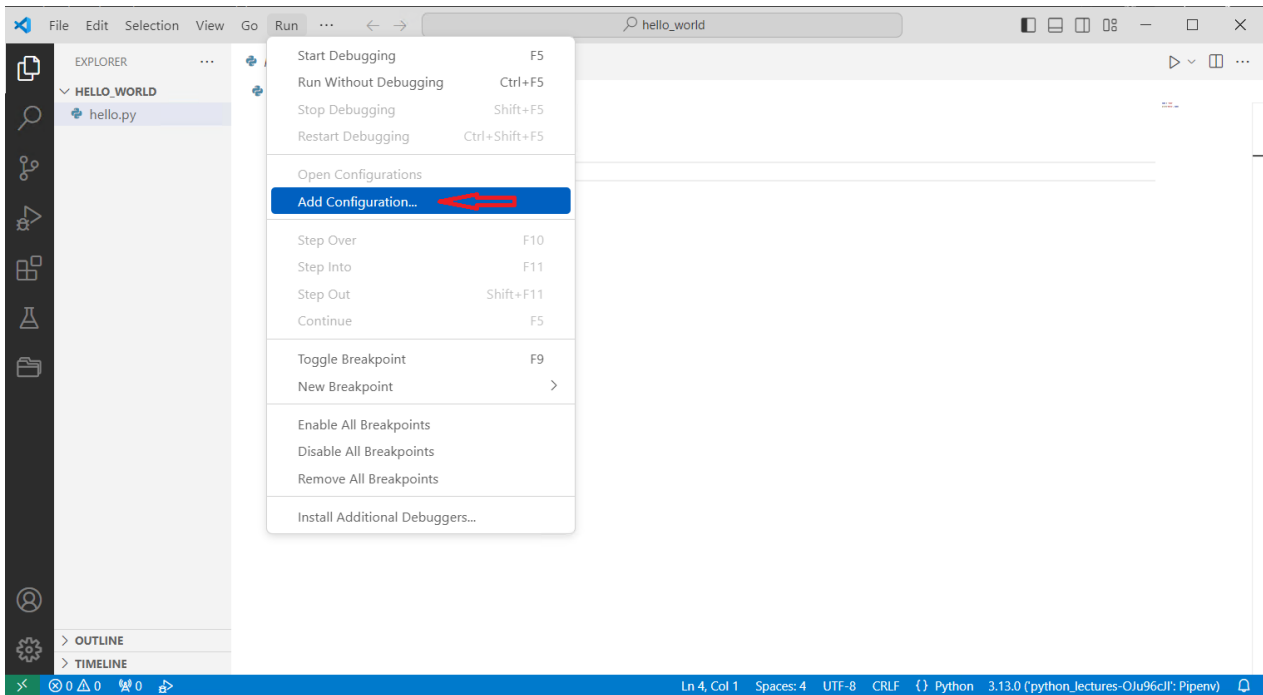
```
print('hello', name)
```

В результаті у вас повинно вийти так, як показано на зображенні

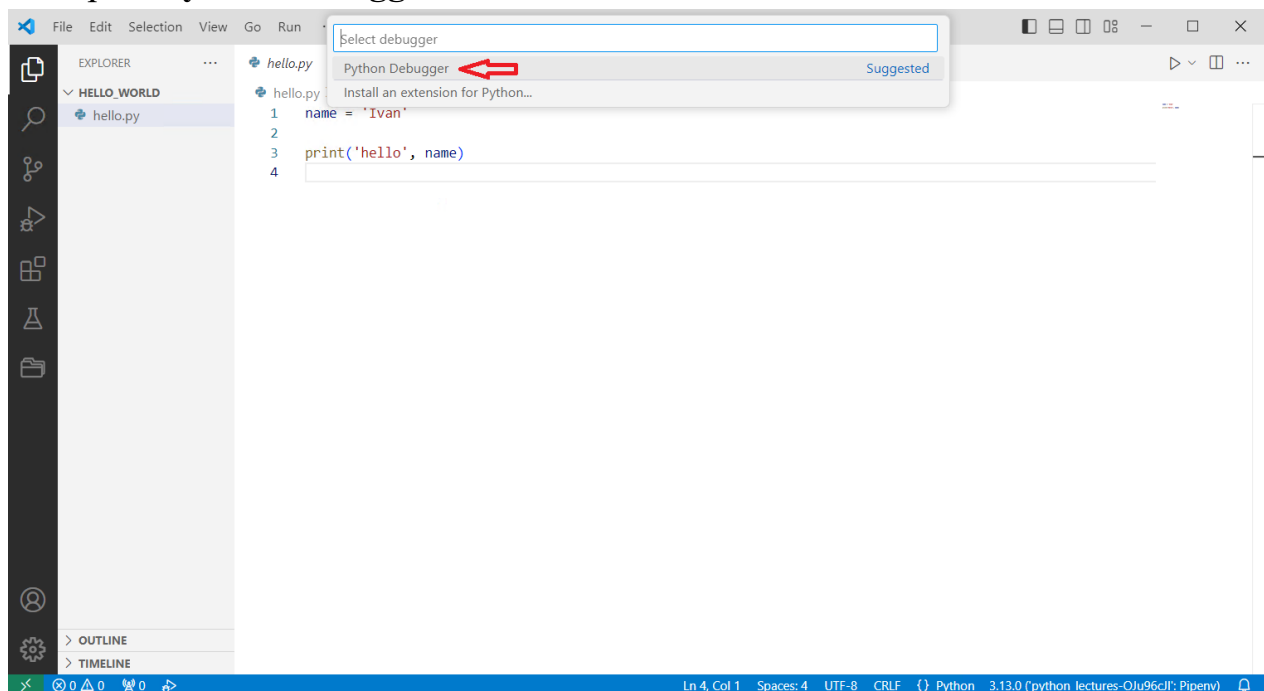


Далі виконайте наступні дії, так як показано на зображеннях

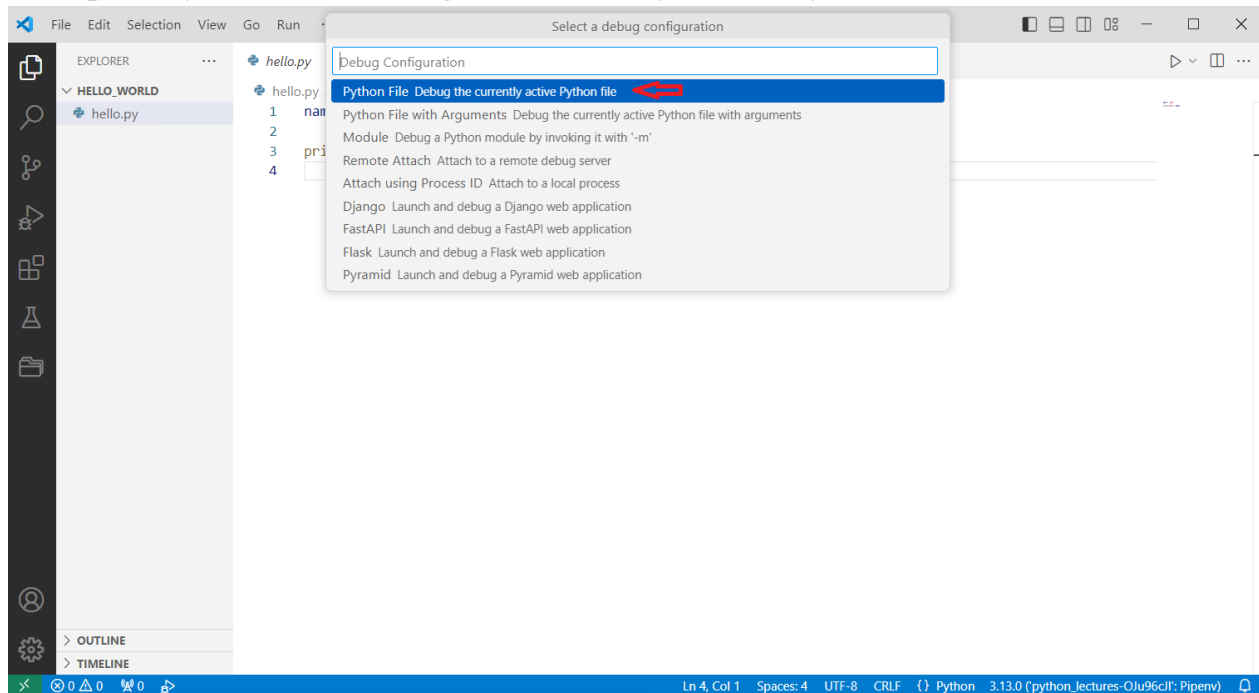
Меню Run > Add Configuration



Виберіть Python Debugger



Виберіть Python File Debug the currently active Python file



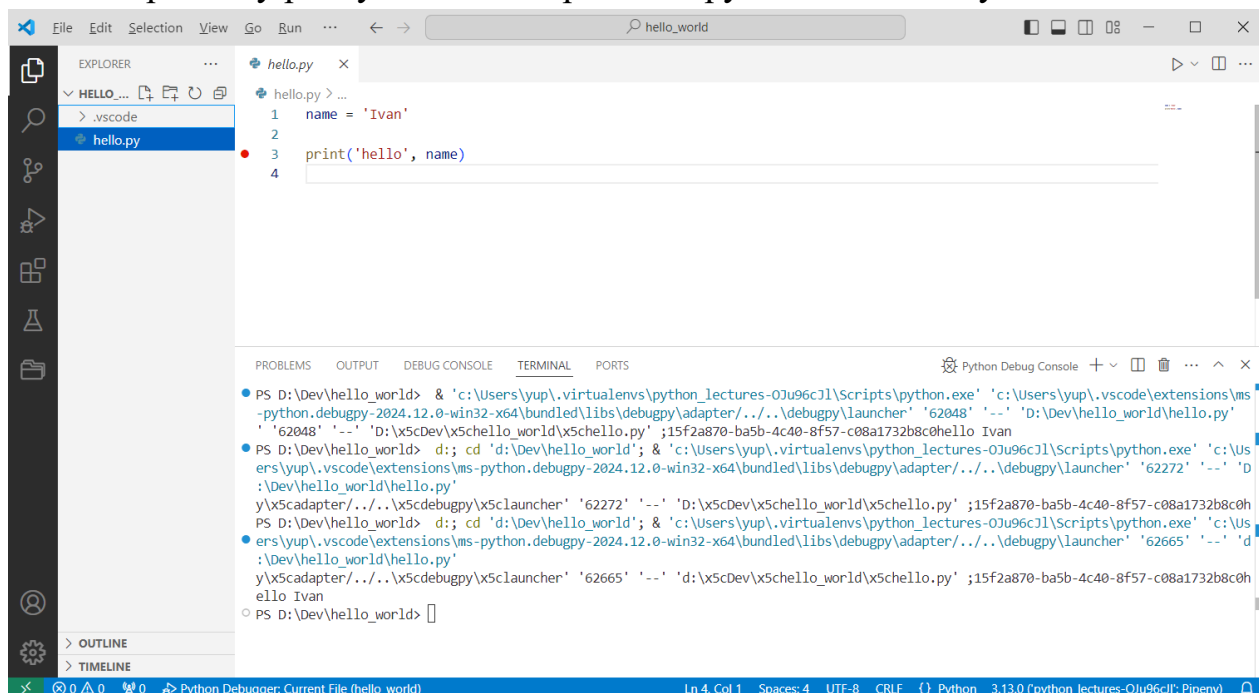
Для того, щоб запустити програму натисніть клавішу F5 або виберіть дію з меню Run > Start Debugging

Перегляньте у вікні виконання результат роботи програми

hello Ivan

Відлагодження програми

Клікніть зліва від третього рядка коду або станьте курсором на 3 рядок і натисніть F9, щоб на третьому рядку з'явився червоний кружечок - точка зупинки.

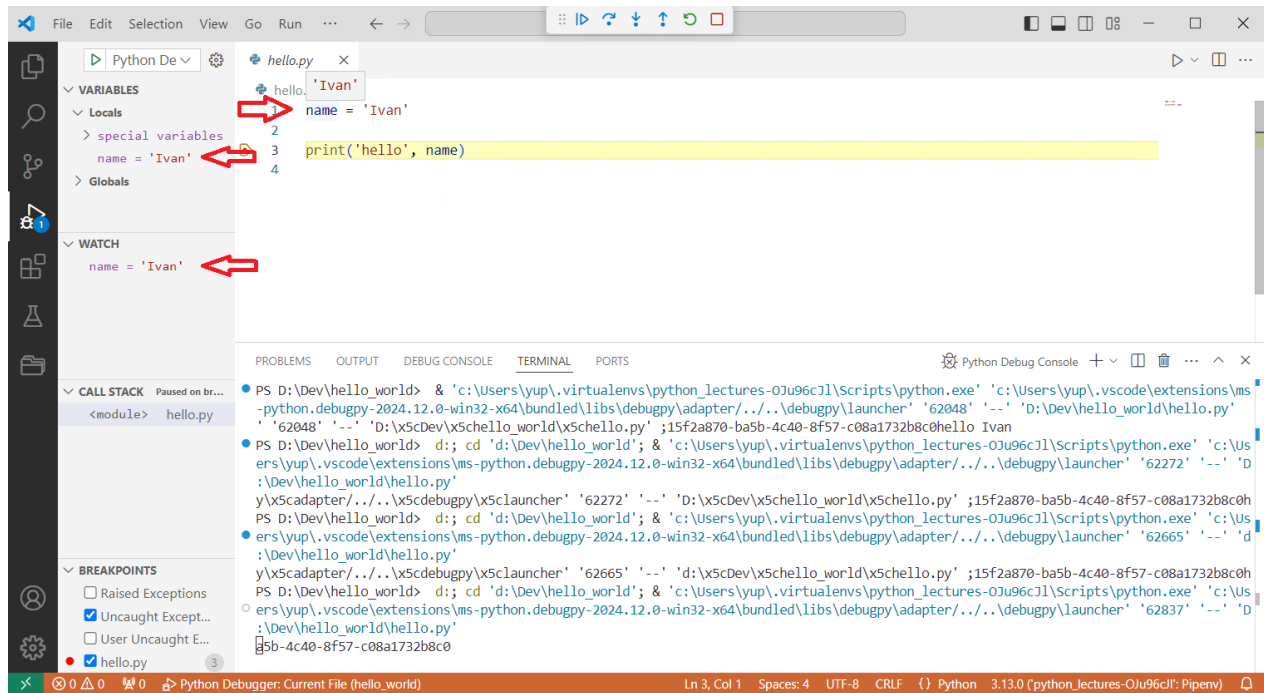


Тепер, коли ви запустите програму на виконання (F5 або Run > Start Debugging) програма запуститься і зупиниться на третьому рядку.

В режимі відлагодження можна перевіряти значення змінних.

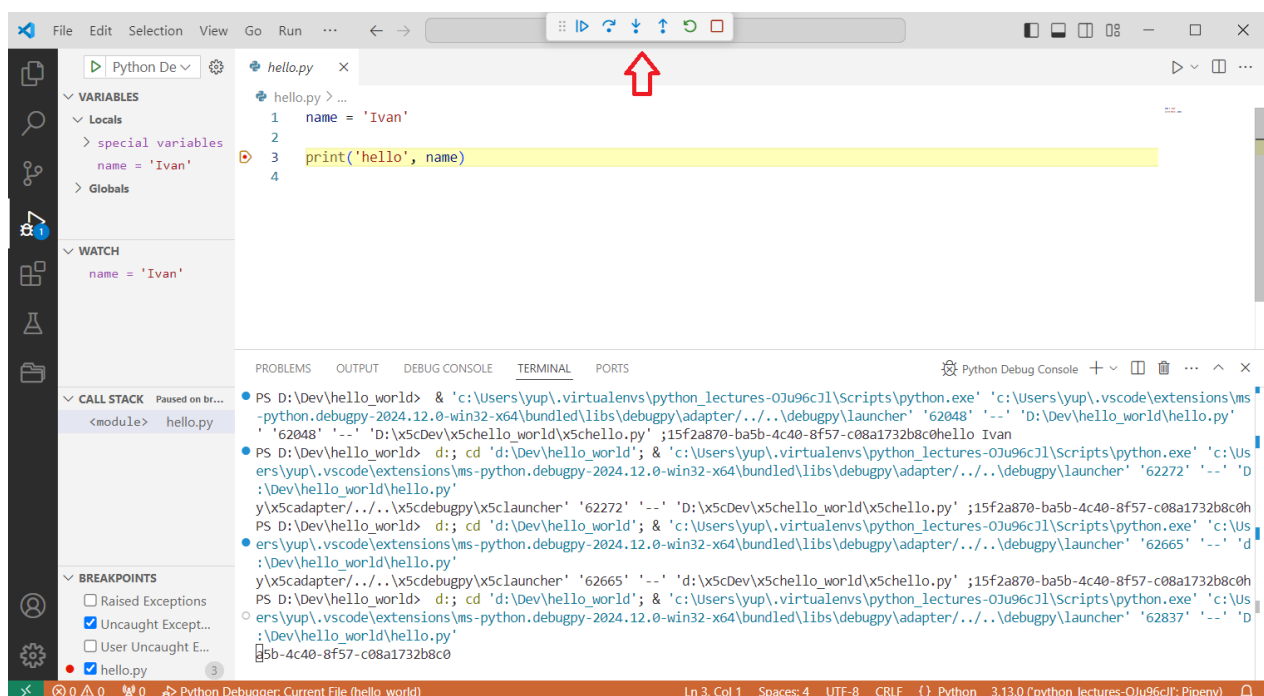
Наведіть курсор мишки на змінну `name` і ви побачите яке значення є в цій змінній на даний момент.

Також можна переглянути на панелях зліва. У вікні **WATCH** можна додати свої вирази і їхнє значення буде відображатися там



Також, в режимі відлагодження, можна покроково виконувати програму. Для цього використовуйте гарячі клавіші * F5 - продовжити виконання до наступної точки зупинки * F10 - крок вперед (виконати наступну інструкцію) * F11 - зайти всередину наступної інструкції (це можливо у випадку текуча інструкція - виклик функції) * Shift + F11 - вийти на рівень вище (у випадку, якщо текуча інструкція знаходиться у функції)

Або скористайтеся кнопками

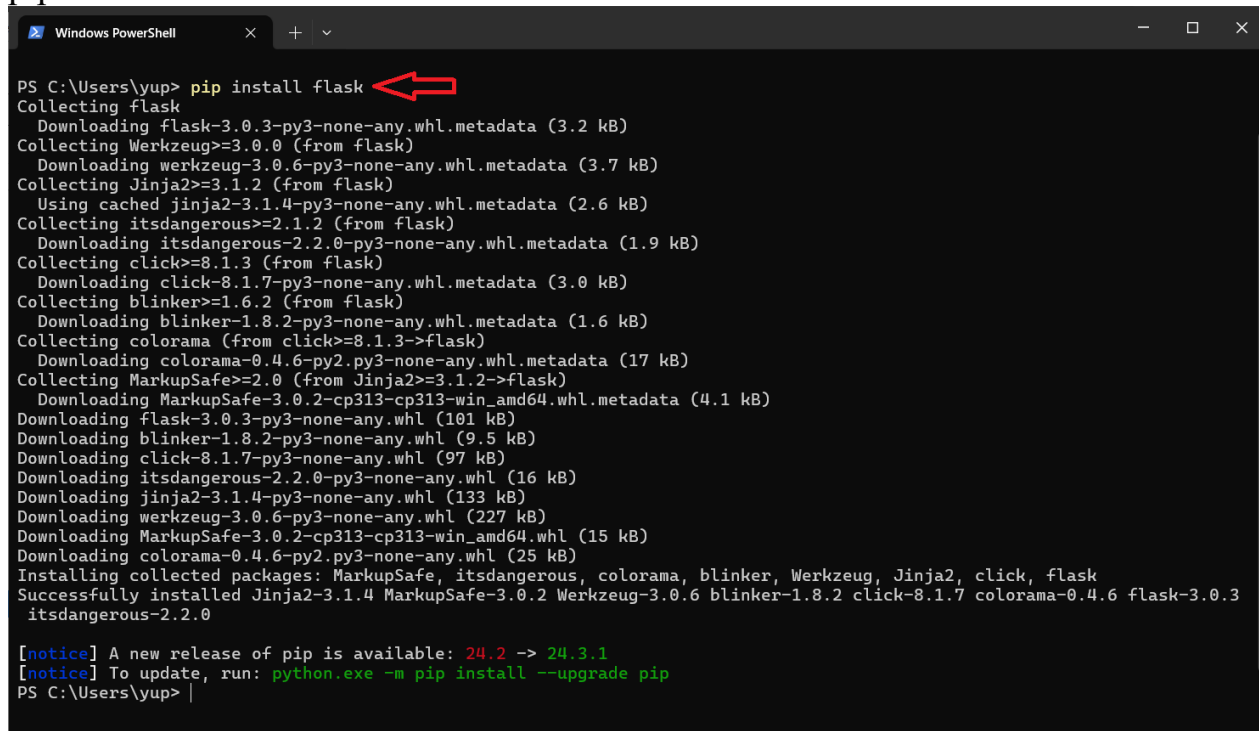


Встановлення модулів Python

Для виконання лабораторних робіт потрібно буде встановити додаткові модулі, наприклад такі як: pandas та flask.

Зробити це можна запустивши термінал, або командну стрічку (cmd).

pip install flask



```
Windows PowerShell
PS C:\Users\yup> pip install flask
Collecting flask
  Downloading flask-3.0.3-py3-none-any.whl.metadata (3.2 kB)
Collecting Werkzeug>=3.0.0 (from flask)
  Downloading werkzeug-3.0.6-py3-none-any.whl.metadata (3.7 kB)
Collecting Jinja2>=3.1.2 (from flask)
  Using cached jinja2-3.1.4-py3-none-any.whl.metadata (2.6 kB)
Collecting itsdangerous>=2.1.2 (from flask)
  Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting click>=8.1.3 (from flask)
  Downloading click-8.1.7-py3-none-any.whl.metadata (3.0 kB)
Collecting blinker>=1.6.2 (from flask)
  Downloading blinker-1.8.2-py3-none-any.whl.metadata (1.6 kB)
Collecting colorama (from click>=8.1.3->flask)
  Downloading colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Collecting MarkupSafe>=2.0 (from Jinja2>=3.1.2->flask)
  Downloading MarkupSafe-3.0.2-cp313-cp313-win_amd64.whl.metadata (4.1 kB)
Downloading flask-3.0.3-py3-none-any.whl (101 kB)
Downloading blinker-1.8.2-py3-none-any.whl (9.5 kB)
Downloading click-8.1.7-py3-none-any.whl (97 kB)
Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Downloading jinja2-3.1.4-py3-none-any.whl (133 kB)
Downloading werkzeug-3.0.6-py3-none-any.whl (227 kB)
Downloading MarkupSafe-3.0.2-cp313-cp313-win_amd64.whl (15 kB)
Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Installing collected packages: MarkupSafe, itsdangerous, colorama, blinker, Werkzeug, Jinja2, click, flask
Successfully installed Jinja2-3.1.4 MarkupSafe-3.0.2 Werkzeug-3.0.6 blinker-1.8.2 click-8.1.7 colorama-0.4.6 flask-3.0.3 itsdangerous-2.2.0

[notice] A new release of pip is available: 24.2 -> 24.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\yup> |
```

У випадку, якщо програма рір не знайдена, то можливо, що ви встановили Python, не так, як показано вище.

Спробуйте запустити командну стрічку і перейти в наступний каталог %USERPROFILE%\AppData\Local\Programs\Python\Python313\Scripts і запустити програму там.

```
cd %USERPROFILE%\AppData\Local\Programs\Python\Python313\Scripts
pip install flask
```

Шлях каталога буде відрізняти залежно від версії Python, яку ви використовуєте. Якщо такий спосіб теж не спрацює, то спробуйте перевстановити Python.

```
Command Prompt
Microsoft Windows [Version 10.0.19045.5011]
(c) Microsoft Corporation. All rights reserved.

C:\Users\yup>cd %USERPROFILE%\AppData\Local\Programs\Python\Python313\Scripts
C:\Users\yup\AppData\Local\Programs\Python\Python313\Scripts>pip install flask
Requirement already satisfied: flask in c:\users\yup\appdata\local\programs\python\python313\lib\site-packages (3.0.3)
Requirement already satisfied: Werkzeug>=3.0.0 in c:\users\yup\appdata\local\programs\python\python313\lib\site-packages (from flask) (3.0.6)
Requirement already satisfied: Jinja2>=3.1.2 in c:\users\yup\appdata\local\programs\python\python313\lib\site-packages (from flask) (3.1.4)
Requirement already satisfied: itsdangerous>=2.1.2 in c:\users\yup\appdata\local\programs\python\python313\lib\site-packages (from flask) (2.2.0)
Requirement already satisfied: click>=8.1.3 in c:\users\yup\appdata\local\programs\python\python313\lib\site-packages (from flask) (8.1.7)
Requirement already satisfied: blinker>=1.6.2 in c:\users\yup\appdata\local\programs\python\python313\lib\site-packages (from flask) (1.8.2)
Requirement already satisfied: colorama in c:\users\yup\appdata\local\programs\python\python313\lib\site-packages (from click>=8.1.3->flask) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\yup\appdata\local\programs\python\python313\lib\site-packages (from Jinja2>=3.1.2->flask) (3.0.2)

[notice] A new release of pip is available: 24.2 -> 24.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\yup\AppData\Local\Programs\Python\Python313\Scripts>_
```

Приклад виконання лабораторної роботи № 6

Завдання: вивести графік цін в Тернопільській області за 2024 рік на продукти: Батон і Хліб пшеничний з борошна вищого ґатунку

Тепер завантажимо файл з даними згідно завдання.

В даному прикладі ми будемо використовувати файл з даними - Зміни цін на споживчі товари.csv.

Далі встановимо модулі pandas та matplotlib за допомогою команди pip

```
! pip install pandas matplotlib
```

Імпортуємо модулі matplotlib.pyplot і pandas та завантажимо дані з csv файлу у DataFrame

```
import matplotlib.pyplot as plt
import pandas as pd
df = pd.read_csv('Зміни цін на споживчі товари.csv')
```

Переглянемо загальну інформацію про наш DataFrame

```
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2446 entries, 0 to 2445
Columns: 443 entries, Показник to 2024-M09
dtypes: float64(438), object(5)
memory usage: 8.3+ MB
```

За допомогою методів head(N) і tail(N) можна переглянути N перших або останніх рядків з таблиці. По замовчуванню N = 5

```
df.tail(3)
```

	Показник	Базисний період	Територіальний розріз	Тип товарів і послуг	Періодичність	1991	1991-M01	1991-M02	1991-M03	1991-M04	...	2023-M12	2024-M01	2024-M02	2024-M03	2024-M04	2024-M05
2443	Індекс споживчих цін	До попереднього місяця	Україна	Ресторани та готелі	Місячна	NaN	NaN	NaN	NaN	NaN	...	100.7	100.9	101.1	100.5	100.6	101.0
2444	Індекс споживчих цін	До попереднього місяця	Україна	Різні товари та послуги	Місячна	NaN	NaN	NaN	NaN	NaN	...	103.8	101.0	101.0	100.0	96.4	100.6
2445	Індекс споживчих цін	До попереднього місяця	Україна	Хліб	Місячна	NaN	NaN	NaN	NaN	NaN	...	100.3	100.3	100.6	100.4	100.8	100.8

3 rows × 443 columns

Тепер знайдемо всі унікальні значення в колонці 'Територіальний розріз'

```
df['Територіальний розріз'].unique()
array(['Україна', 'Вінницька', 'Волинська', 'Дніпропетровська',
       'Донецька', 'Житомирська', 'Закарпатська', 'Запорізька',
       'Івано-Франківська', 'Київська', 'Кіровоградська', 'Луганська',
```

```
'Львівська', 'Миколаївська', 'Одеська', 'Полтавська',
'Рівненська',
'Сумська', 'Тернопільська', 'Харківська', 'Херсонська',
'Хмельницька', 'Черкаська', 'Чернівецька', 'Чернігівська',
'Київ',
'Автономна Республіка Крим', 'Севастополь'], dtype=object)
Виберемо дані по Тернопільській області для продуктів: Батон і Хліб
пшеничний з борошна вищого ґатунку
```

```
ternopil_region = df.loc[
    (df['Територіальний розріз'] == 'Тернопільська')
    & df['Тип товарів і послуг'].isin(['Батон', 'Хліб пшеничний з
борошна вищого ґатунку'])]
ternopil_region.head(3)
```

	Показник	Базисний період	Територіальний розріз	Тип товарів і послуг	Періодичність	1991	1991- M01	1991- M02	1991- M03	1991- M04	...	2023- M12	2024- M01	2024- M02	2024- M03	2024- M04	2024- M05
Тепер виберемо всі	991 Середні споживчі ціни на товари (послуги)	Не застосовується	Тернопільська	Хліб пшеничний з борошна вищого ґатунку	Місячна	NaN	NaN	NaN	NaN	NaN	...	31.41	32.43	32.43	32.43	32.59	32.59
	994 Середні споживчі ціни на товари (послуги)	Не застосовується	Тернопільська	Батон	Місячна	NaN	NaN	NaN	NaN	NaN	...	23.10	23.55	23.55	23.55	23.77	23.97

КОЛО 2 rows × 443 columns

нки

за 2024 рік і встановимо Тип товарів і послуг індексною колонкою.

Індексна колонка відображається як номер рядка. Це потрібно для відображення графіка

```
ternopil_region_2024 = ternopil_region[['Тип товарів і послуг', '2024-
M01', '2024-M02', '2024-M03', '2024-M04', '2024-M05', '2024-M06', '2024-
M07', '2024-M08']]
ternopil_region_2024 = ternopil_region_2024.set_index('Тип товарів і
послуг')
display(ternopil_region_2024)
```

	2024-M01	2024-M02	2024-M03	2024-M04	2024-M05	2024-M06	2024-M07	2024-M08
Хліб пшеничний з борошна вищого ґатунку	32.43	32.43	32.43	32.59	32.59	32.89	33.77	34.35
Батон	23.55	23.55	23.55	23.77	23.97	24.27	24.98	25.13

Тепер нам потрібно поміняти місцями рядки з колонками. Для цього використаємо метод `transpose()`

```
ternopil_region_2024 = ternopil_region_2024.transpose()
display(ternopil_region_2024)
```

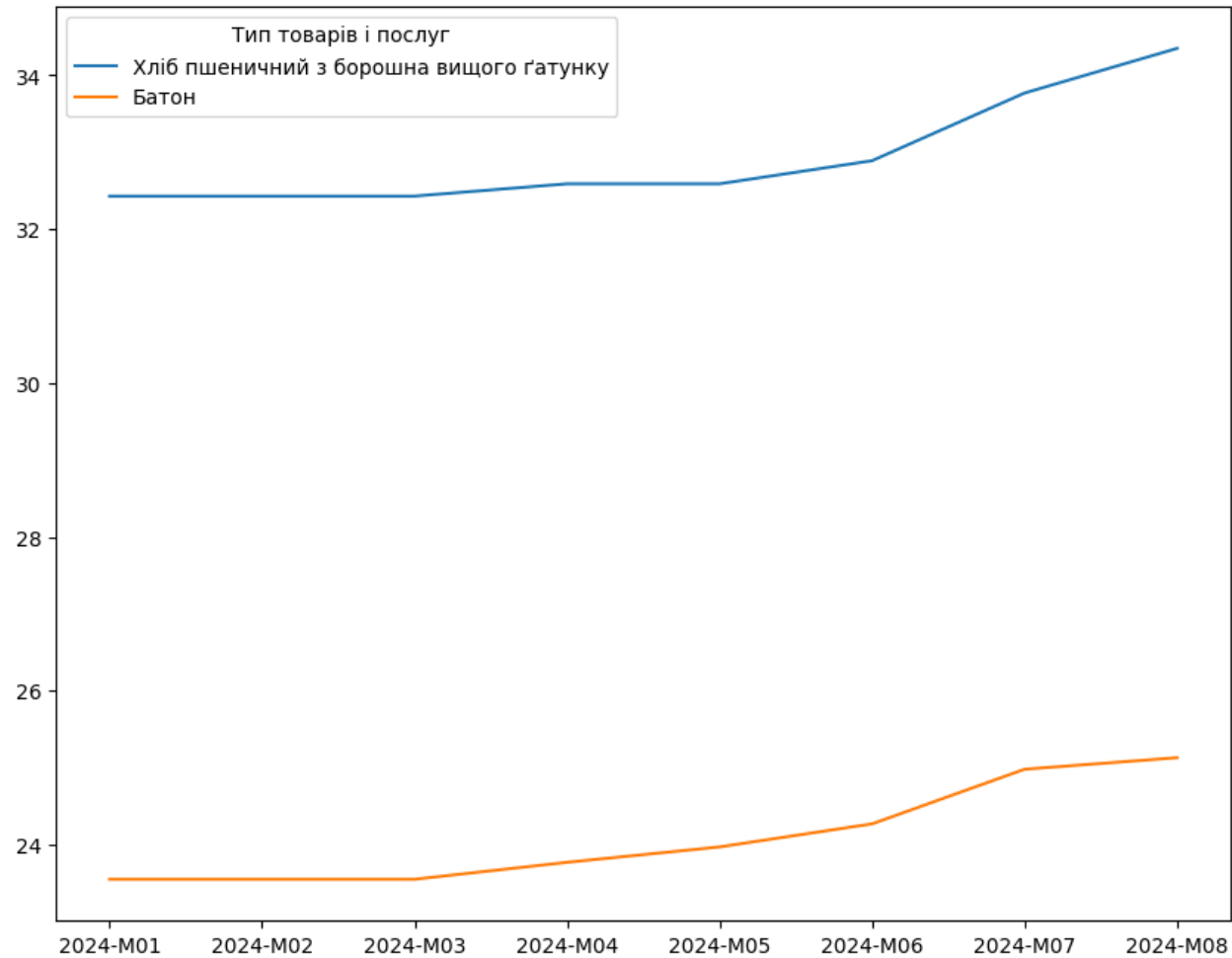
Тепер відобразимо графік для отриманих даних.

Значення з першої колонки будуть використані для осі X, а дані з інших колонок для побудови графіків.

Тип графіку може бути одним з наступних: 'line' або 'bar'

```
ternopil_region_2024.plot(kind='line', figsize=(10.24, 8.0))
<Axes: >
```

Тип товарів і послуг	Хліб пшеничний з борошна вищого ґатунку	Батон
2024-M01	32.43	23.55
2024-M02	32.43	23.55
2024-M03	32.43	23.55
2024-M04	32.59	23.77
2024-M05	32.59	23.97
2024-M06	32.89	24.27
2024-M07	33.77	24.98
2024-M08	34.35	25.13



```
plt.show()
```

Приклад виконання лабораторної роботи № 7

Підготовка до написання програми

Встановіть модуль flask, так як це описано в документі Налаштування середовища розробки

```
pip install flask
```

В VS code створіть наступні файли та каталоги:

- файл app.py
- каталог templates і в ньому файл index.html
- каталог .vscode і в ньому файл launch.json

В результаті повинна бути така структура файлів

```
| - app.py
| - templates
| ---- index.html
| -- .vscode
| ---- launch.json
```

launch.json повинен містити наступний код

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Python Debugger: Flask",
      "type": "debugpy",
      "request": "launch",
      "module": "flask",
      "env": {
        "FLASK_APP": "app.py",
        "FLASK_DEBUG": "1"
      },
      "args": [
        "run"
      ],
      "jinja": true,
      "autoStartBrowser": false
    }
  ]
}
```

Написання програми для визначення периметру та площі кола

Додайте в файл app.py наступний код

```
import flask
import math
```



```

# створимо об'єкт, який описує сайт і буде обробляти запити
app = flask.Flask(__name__)

# функція-обробник (хендлер) запитів
@app.route('/', methods=['GET', 'POST'])
def index():

    # отримуємо з даних форми значення радіуса
    r = flask.request.form.get('radius')

    try:
        # всі значення форми є стрічками, тому перетворемо це значення в
        # дробове число
        r = float(r)
    except ValueError:
        r = 0

    if r > 0:
        # обчислимо периметр та площу
        perimeter = 2 * math.pi * r
        area = math.pi * r * r

        # згенеруємо html код з шаблону `index.html` і виведемо значення
        # периметру та площі
        return flask.render_template('index.html', radius = r, result = {
            'perimeter': perimeter,
            'area': area
        })

    # якщо радіус не було передано з запитом то відобразимо початкову
    # сторінку
    return flask.render_template('index.html')

if __name__ == '__main__':
    app.run(debug = True)

```

А в файл templates/index.html такий код html

```

<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.mi
n.css" rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMHjY6hW+ALEwIH"
crossorigin="anonymous">
    <title>Обчислення периметру та площі кола</title>
</head>

```

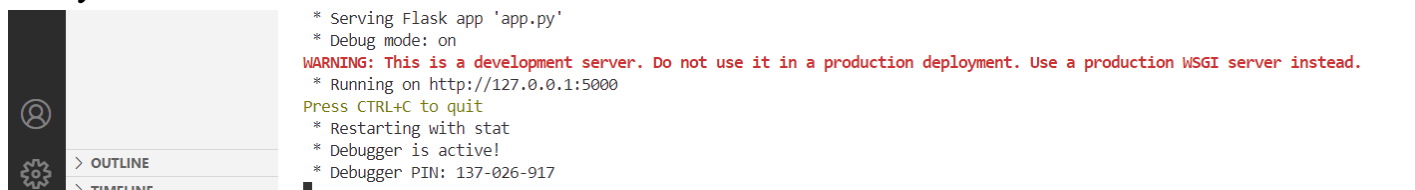
```

<body>
  <div class="container">
    <div class="row justify-content-center">
      <div class="col-sm-6">
        <h2>Обчислення периметру та площі кола</h2>
        <form method="post">
          <input type="number" min="0" step="0.01"
name="radius" value="{{ radius }}" placeholder="Радіус" class="mb-3
form-control">
          <input type="submit" value="Обчислити" class="form-
control mb-3 btn btn-success">
        </form>
        {% if result %}
        <div class="mb-3">Периметр кола: {{ result.perimeter
}}</div>
        <div class="mb-3">Площа кола: {{ result.area }}</div>
        {% endif %}
      </div>
    </div>
  </div>
</body>
</html>

```

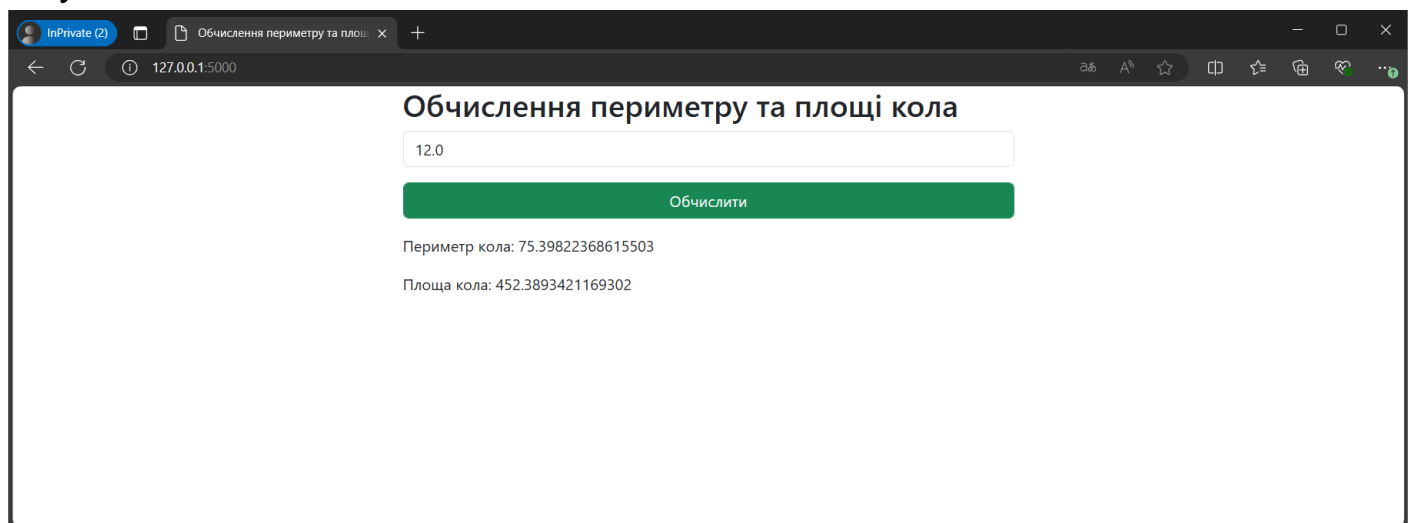
Запуск програми

Запустіть app.py на виконання. В результаті в консолі VS code можна буде побачити наступне



У веб-браузері відкрийте сайт <http://127.0.0.1:5000/> і попробуйте ввести декілька різних значень для радіуса

Результат виконання



Рекомендована література

1. Пришвидшений курс Python. Практичний, проєктно-орієнтований вступ до програмування: Ерік Маттесс; перекл. з англ. Ольги Белової. - Львів: Видавництво Старого Лева, 2021. - 600с.
2. Основи програмування алгоритмічною мовою Python [Електронний ресурс]: навч. посіб. для студ. освітньої програми «Комп'ютерні системи та мережі» спеціальності 123 «Комп'ютерна інженерія» / М.А. Новотарський – Електронні текстові дані (1 файл: 18 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2022. – 701с.
3. Head First Python. Publisher: O.Reilly, 2023. 663 p.
4. Python official docs [Електронний ресурс]. – Режим доступу: <https://docs.python.org/uk/3/>
5. Think Python [Електронний ресурс]. – Режим доступу: <https://allendowney.github.io/ThinkPython/>
6. Automate the Boring Stuff with Python [Електронний ресурс]. – Режим доступу: <https://automatetheboringstuff.com/>
7. The Hitchhiker's Guide to Python [Електронний ресурс]. – Режим доступу: <https://docs.python-guide.org/>
8. Python for Everybody [Електронний ресурс]. – Режим доступу: <https://www.py4e.com/book>
9. Python module "psutil" [Електронний ресурс]. – Режим доступу: <https://pypi.org/project/psutil/>
10. Flask micro web framework [Електронний ресурс]. – Режим доступу: <https://flask.palletsprojects.com/en/3.0.x/>
11. Pandas documentation [Електронний ресурс]. – Режим доступу: <https://pandas.pydata.org/docs/>
12. Getting started with plotly in Python [Електронний ресурс]. – Режим доступу: <https://plotly.com/python/getting-started/>
13. Tkinter Introduction: A Beginner's Guide [Електронний ресурс]. – Режим доступу: <https://ultrapythonic.com/tkinter-introduction/>

Навчально-методичне видання

**Попик Юрій Ігорович
Сегін Андрій Ігорович**

**МЕТОДИЧНІ ВКАЗІВКИ
ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ
з дисципліни
“ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ
КОМП’ЮТЕРИЗОВАНИХ ВИМІРЮВАЛЬНИХ
ПРИСТРОЇВ”**

для студентів освітньо-кваліфікаційного рівня «бакалавр»
спеціальності 175 «Інформаційно-вимірювальні технології»

Підписано до друку 12.02.2025 р.
Формат 60х84/16. Папір офсетний.
Друк офсетний. Зам. № 25-229
Умов.-друк. арк. 1,9. Обл.-вид. арк. 2,1.
Тираж 30 прим.

Віддруковано ФО-П Шпак В. Б.
Свідоцтво про державну реєстрацію В02 № 924434 від 11.12.2006 р.
м. Тернопіль, бульвар Просвіти, 6/4. тел. 097 299 38 99.
E-mail: tooums@ukr.net

*Свідоцтво про внесення суб'єкта видавничої справи до державного
реєстру видавців, виготовлювачів і розповсюджувачів видавничої продукції
ДК № 7599 від 10.02.2022 р.*