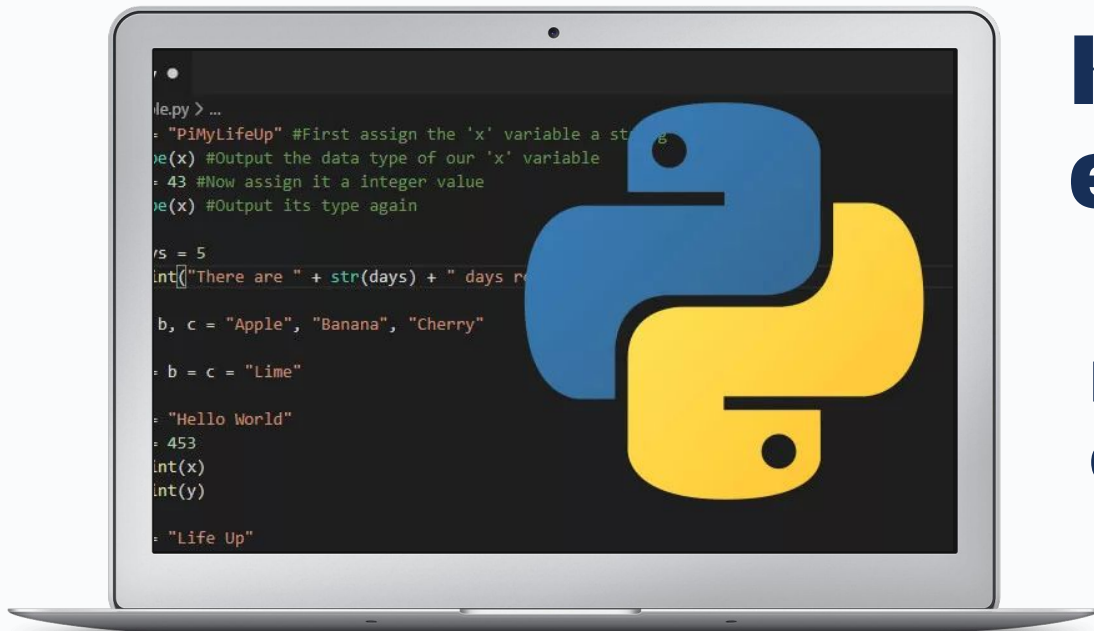




Become QA Auto



Класи та об'єкти

Бутенко
Сергій



План лекції



Етапи створення програми в ООП



Атрибути об'єкта



Поняття класу



Атрибути класу



Поняття об'єкта



Етапи створення програми в ООП



● Оголошення класів

визначають
структуру та
поведінку об'
єктів



● Створення об'єктів

екземплярів
оголошених
класів, задання
їхніх полів



● Організація комунікації

між об'єктами
у вигляді
послідовності
повідомлень



Поняття класу



Клас – це план для створення об'єкту, користувацький тип даних.

Клас містить опис структури об'єкта: імена його атрибутів, оголошення методів

Структура класу в Python:

```
class <ім'я класу>:  
    <атрибути класу>  
    <конструктор класу>  
    <методи>
```

Приклад найпростішого опису класу:

```
class ShopWorker:  
    pass
```



Поняття об'єкта



Об'єкт - це програмний компонент, який зберігає дані та функції для роботи з цими даними.

Об'єкт - це екземпляр класу (конкретний представник)

Одна із головних задач об'єктно-орієнтованих програм - правильно визначити, які об'єкти необхідні для вирішення задачі.

Створення об'єкта в Python:

`<ім'я об'єкта> = <ім'я класу>()`

Наприклад

```
worker_one = ShopWorker()
```

```
worker_two = ShopWorker()
```

Приклад 1 створення класу та об'єктів



The screenshot shows a VS Code editor with a file named `shop.py` open. The file contains the following Python code:

```
1 class ShopWorker:
2     pass
3 worker_one = ShopWorker()
4 worker_two = ShopWorker()
5 print(worker_one)
6 print(worker_two)
7
```

The terminal at the bottom shows the command to run the script and the resulting output:

```
PS D:\PYTHON_BASICS> & C:/Users/user/AppData/Local/Programs/Python/Python38/python.exe d:/PYTHON_BASICS/shop.py
<__main__.ShopWorker object at 0x000001E8C3168880>
<__main__.ShopWorker object at 0x000001E8C3168EB0>
PS D:\PYTHON_BASICS>
```

The output lines are highlighted with a red box.



Атрибути (властивості, поля)



Атрибути для працівників магазину

1. ім'я працівника (name)
2. вік працівника (age)
3. кількість працівників в магазині (count_workers)

Атрибути об'єкта

властивості є унікальними для кожного об'єкта

Атрибути класу

властивості однакові для всіх об'єктів

Створення атрибута об'єкта

`<ім'я об'єкта>.<ім'я атрибута об'єкта>=`
`<Значення атрибута>`

Наприклад:

```
worker_one.name= 'Іван'  
worker_two.name= 'Петро'
```

Приклад 2. Створення атрибутів об'єкта



shopAttr.py X

PYTHON_BASICS > shopAttr.py > ...

```
1 class ShopWorker:
2     pass
3
4 worker_one = ShopWorker()
5 worker_two = ShopWorker()
6
7 worker_one.name = 'Іван'
8 worker_one.age = 25
9
10 worker_two.name = 'Петро'
11 worker_two.age = 32
12
13 print("Працівник 1: ", worker_one.name, " ", worker_one.age)
14 print("Працівник 2: ", worker_two.name, " ", worker_two.age)
```

PROBLEMS OUTPUT TERMINAL ...

Python + - [] [X]

PS D:\PYTHON_BASICS> & C:/Users/user/AppData/Local/Programs/Python/Python38/python.exe d:/PYTHON_BASICS/shop.py

Працівник 1: Іван 25
Працівник 2: Петро 32

Опис класу **ShopWorker**

Створення об'єктів класу **worker_one** та **worker_two**

Створення атрибутів об'єкта **worker_one**

Створення атрибутів об'єкта **worker_two**

Вивід атрибутів **name** та **age** об'єктів **worker_one** та **worker_two**

Результат роботи коду

◎ Створення атрибутів класу



Атрибути класу створюються в розділі опису класу відразу під рядком, що починається словом **class**.

Всі створені об'єкти отримують цей атрибут та його значення.

Використовуйте атрибути класу для визначення властивостей, які мають однакове значення для кожного екземпляра класу.

Синтаксис написання:

```
class <Ім'я класу>:  
    <ім'я атрибуту>= <значення атрибуту>
```

Наприклад

```
class ShopWorker:  
    count_workers=2
```

Рекомендується:

<об'єкт>.<атрибут> - звернення до атрибута об'єкта
<клас>.<атрибут> - звернення до атрибута класу

Приклад 3. Створення атрибуту класу



shopAttr1.py X

PYTHON_BASICS > shopAttr1.py > ...

```
1 class ShopWorker:
2     count_workers = 2
3
4 worker_one = ShopWorker()
5 worker_two = ShopWorker()
6
7 worker_one.name = 'Іван'
8 worker_one.age = 25
9
10 worker_two.name = 'Петро'
11 worker_two.age = 32
12
13 print("Працівник 1: ", worker_one.name, " ", worker_one.age,
14      " всіх працівників ", worker_one.count_workers)
15 print("Працівник 2: ", worker_two.name, " ", worker_two.age,
16      " всіх працівників ", worker_two.count_workers)
17 print("всіх працівників ", ShopWorker.count_workers)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Python + - []

```
Працівник 1: Іван 25 всіх працівників 2
Працівник 2: Петро 32 всіх працівників 2
всіх працівників 2
PS D:\PYTHON_BASICS> []
```

Опис класу **ShopWorker**

Створення атрибуту класу **count_workers**

Створення об'єктів класу **worker_one** та **worker_two**

Створення атрибутів об'єкта **worker_one**

Створення атрибутів об'єкта **worker_two**

Вивід атрибутів об'єктів **worker_one** та **worker_two** та атрибуту класу

Вивід атрибуту класу **ShopWorker**

Результат роботи коду



Підсумки



Клас (class) – це шаблон для створення об'єкту, користувацький тип даних



Для створення об'єкта в Python використовують конструкцію:
<ім'я об'єкта> = <ім'я класу>()



Об'єкт - це екземпляр класу (конкретний представник)



Атрибути об'єкта - властивості є унікальними для кожного об'єкта
Атрибути класу - властивості однакові для всіх об'єктів



Для створення класу в Python використовують конструкцію:
class <ім'я класу>:



Рекомендується:
<об'єкт>.<атрибут> - звернення до атрибута об'єкта
<клас>.<атрибут> - звернення до атрибута класу