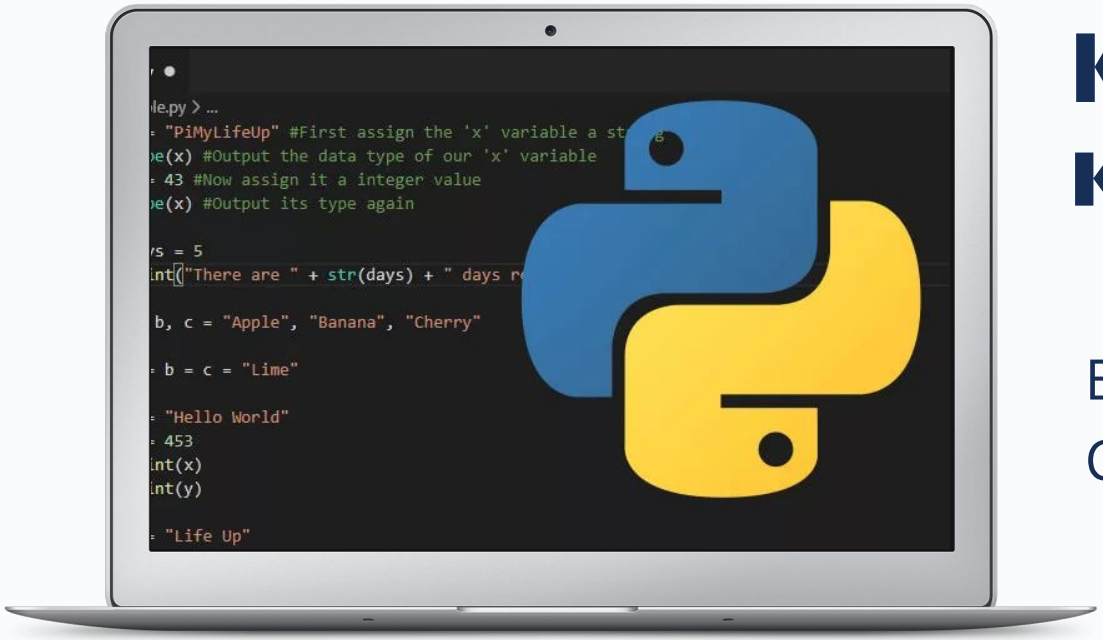
Become QA Auto

PROMETHEUS



Конструктор класу

Бутенко
Сергій

План лекції

- Поняття конструктора класу
- Конструктор як засіб для ініціалізації атрибутів об'єкта
- Конструктор з вказанням параметрів за замовчуванням

Поняття конструктора класу

Конструктор класу – це спеціальні функції, що викликаються під час створення об'єкта (`__new__()` та **`__init__()`**).

Загальний вигляд конструктора класу:

```
def __init__(self[, <параметри>]):
        <тіло конструктора>
```

Види конструкторів:

- конструктор за замовчуванням
- не параметризований конструктор
- параметризований конструктор.

`self` – це посилання саме на той об'єкт класу, що створюється.

Приклад 1. Використання конструктора з параметром self

shopConstructor.py

PYTHON_BASICS > shopConstructor.py > ...

```
1  class ShopWorker:
2      count_workers = 0
3      def __init__(self):
4          ShopWorker.count_workers +=1
5
6  print("всіх працівників ", ShopWorker.count_workers)
7
8  worker_one = ShopWorker()
9  worker_one.name = 'Іван'
10 worker_one.age = 25
11
12 print("Працівник 1: ",worker_one.name," ", worker_one.age,
13 " всіх працівників ",  worker_one.count_workers)
14
15 worker_two = ShopWorker()
16 worker_two.name = 'Петро'
17 worker_two.age = 32
18
19 print("Працівник 2: ", worker_two.name, " ", worker_two.age,
20 " всіх працівників ", worker_two.count_workers)
```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE JUPYTER Python

```
всіх працівників  0
Працівник 1:  Іван   25  всіх працівників  1
Працівник 2:  Петро   32  всіх працівників  2
```

Опис класу **ShopWorker**
Створення атрибуту класу **count_workers**
Конструктор класу
Збільшення атрибуту **count_workers** на 1.

Вивід атрибуту класу **ShopWorker**

Створення об'єкту **worker_one**
Створення атрибутів об'єкта **worker_one**

Вивід атрибутів об’єкту **worker_one** та атрибуту класу

Створення об'єкту **worker_two**
Створення атрибутів об'єкта **worker_two**

Вивід атрибутів об’єкта **worker_two** та атрибуту класу

Результат роботи коду

Конструктор як засіб для ініціалізації атрибутів об'єкта

Синтаксис параметризованого конструктора:

```
def __init__(self, <параметр1>, [<параметр2>, …]):
    self.<атрибут1>=<параметр1>
    [self.<атрибут2>=<параметр2>  … ]
```

Приклад конструктора для ініціалізації атрибутів **name** і **age** та збільшення значення атрибуту **count_workers** класу **ShopWorker** на 1.

```
def __init__(self, name1, age1):
        self.name = name1
        self.age = age1
        ShopWorker.count_workers +=1
```

Приклад створення об'єкта

```
worker_one = ShopWorker('Іван', 25)
```

Приклад 2. Використання конструктора з параметрами

shopConstructor1.py

PYTHON_BASICS > shopConstructor1.py > ...

```
1   class ShopWorker:
2       count_workers = 0
3       def __init__(self, name1, age1):
4           self.name = name1
5           self.age = age1
6           ShopWorker.count_workers +=1
7
8   print("всіх працівників ", ShopWorker.count_workers)
9   #worker_one = ShopWorker()
10  worker_one = ShopWorker('Іван', 25)
11  #worker_one.name = 'Іван'
12  #worker_one.age = 25
13
14  print("Працівник 1: ",worker_one.name," ", worker_one.age,
15  " всіх працівників ",  worker_one.count_workers)
16
17  worker_two = ShopWorker('Петро', 32)
18
19  print("Працівник 2: ", worker_two.name, " ", worker_two.age,
20  " всіх працівників ", worker_two.count_workers)
```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE JUPYTER Python

```
всіх працівників  0
Працівник 1:  Іван   25  всіх працівників  1
Працівник 2:  Петро   32  всіх працівників  2
```

Опис класу **`ShopWorker`**
Створення атрибуту класу **`count_workers`**
Конструктор класу

Створення та ініціалізація атрибутів об'єкта

Збільшення атрибуту **`count_workers`** на 1.

Вивід атрибуту класу **`ShopWorker`**

Створення об'єкту **`worker_one`** та ініціалізація його полів

Вивід атрибутів об’єкту **`worker_one`** та атрибуту класу

Створення об'єкту **`worker_two`** та ініціалізація його полів

Вивід атрибутів об’єкта **`worker_two`** та атрибуту класу

Результат роботи коду

Конструктор з вказанням параметрів за замовчуванням

Синтаксис конструктора з параметрами за замовчуванням:

```
def __init__(self, <параметр1>[=<значення1>],
[<параметр2>[=<значення2>], …]):
    self.<атрибут1>=<параметр1>
    [self.<атрибут2>=<параметр2>  … ]
```

Приклад конструктора для ініціалізації атрибутів **name** і **age** та збільшення значення атрибуту **count_workers** класу **ShopWorker** на 1.

```
def __init__(self, name1= '', age1=0):
        self.name = name1
        self.age = age1
        ShopWorker.count_workers +=1
```

Приклад створення об'єкта

```
worker_one = ShopWorker()
worker_two = ShopWorker('Петро', 32)
worker_three = ShopWorker('Семен')
```

Приклад 3. Використання конструктора з вказанням параметрів за замовчуванням

shopConstructor2.py

PYTHON_BASICS > shopConstructor2.py > ...

```
1  class ShopWorker:
2      count_workers = 0
3      def __init__(self, name1='', age1=0):
4          self.name = name1
5          self.age = age1
6          ShopWorker.count_workers +=1
7
8  print("всіх працівників ", ShopWorker.count_workers)
9
10 worker_one = ShopWorker()
11
12 print("Працівник 1: ",worker_one.name," ", worker_one.age,
13 " всіх працівників ",  worker_one.count_workers)
14
15 worker_two = ShopWorker('Петро', 32)
16
17 print("Працівник 2: ", worker_two.name, " ", worker_two.age,
18 " всіх працівників ", worker_two.count_workers)
19
20 worker_three = ShopWorker('Семен')
21
22 print("Працівник 3: ", worker_three.name, " ", worker_three.age,
23 " всіх працівників ", worker_three.count_workers)
```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE JUPYTER Python

```
всіх працівників  0
Працівник 1:     0  всіх працівників  1
Працівник 2:  Петро   32  всіх працівників  2
Працівник 3:  Семен   0  всіх працівників  3
```

Опис класу **ShopWorker**
Створення атрибуту класу **count_workers**
Конструктор класу
Створення та ініціалізація атрибутів об'єкта
Збільшення атрибуту **count_workers** на 1.

Вивід атрибуту класу **ShopWorker**

Створення об'єкту **worker_one** та ініціалізація його полів

Вивід атрибутів об’єкту **worker_one** та атрибуту класу

Створення об'єкту **worker_two** та ініціалізація його полів

Вивід атрибутів об’єкта **worker_two** та атрибуту класу

Створення об'єкту **worker_three**та ініціалізація його полів

Вивід атрибутів об’єкта **worker_three** та атрибуту класу

Результат роботи коду

Підсумки

Конструктор класу – це спеціальні функції, що викликаються під час створення об'єкта

Призначення конструктора — встановити початковий стан об'єкта з можливою ініціалізацією його атрибутів.

Кожен клас має конструктор

`self` – це посилання саме на той об'єкт класу, що створюється.

Синтаксис конструктора:

def __init__(self, [<параметр1> [=<значення1>], [<параметр2> [=<значення2>], ...]]**):**
    [self.<атрибут1>=<параметр1>]
    [self.<атрибут2>=<параметр2> ...]