

Python з нуля

Функції, об'єкти, стандартні структури даних

Богдан Кулинич

Web Academy

17 лютого 2015 р.

- ▶ Код, який можна повторно використовувати (reuse)
- ▶ Один з найголовніших елементів будь-якого програмного забезпечення

```
def convert_to_celcius(fahrenheit):  
    ...return (fahrenheit - 32) * 5 / 9
```

```
def convert_to_celcius(32)  
def convert_to_celcius(451)
```

Функції

scope

- ▶ Область видимості (scope)
- ▶ global, local

```
x = 'plumbum'
```

```
def meh():  
    ...x = 'gold'
```

```
meh()  
print(x)      # 'plumbum'
```

```
def magic():  
    ...global x  
    ...x = 'gold'
```

```
magic()  
print(x)      # 'gold'
```

- ▶ Передача літералів або незмінних типів (immutable)

```
x = 10                                # 10
```

```
def square(x):
```

```
    ... x = x * x
```

```
square(x)
```

```
print(x)                              # 10
```

- ▶ Передача змінних типів (mutable)

```
a = [1, 2, 3]
```

```
def mutate(iterable):
```

```
    ... iterable.append(4)
```

```
mutate(a)
```

```
print(a)                              # [1, 2, 3, 4]
```

- ▶ Параметри за замовчуванням

```
def say(message, times=1):  
    ...print(message * times)  
  
say('Hello')           # Hello  
say('Hello', 2)        # HelloHello
```

- ▶ Параметри за ключовими словами. Оператори *, **

```
def hello(**kwargs):  
    ...print('Hello',  
    .....kwargs[firstname],  
    .....kwargs[lastname])  
  
hello(name='Petryk', lastname='Pyatochkin')  
# Hello Petryk Pyatochkin
```

► Docstrings

```
def convert_to_celcius(fahrenheit):  
    """Convert Fahrenheit degrees to Celcius  
    .  
    . . . . Use formula  $C = (F - 32) * 5 / 9$   
    . . . . """  
    . . . . return (fahrenheit - 32) * 5 / 9
```

► Лямбди

```
square = lambda x: x**2  
square(4)      # 16
```

► Рекурсія

```
def factorial(n):  
    ....if n == 0:  
    .....return 1  
    ....elif n > 0:  
    .....return n * factorial(n-1)
```

► Функціональне програмування

Об'єкти

Об'єкт — значення з певним станом і поведінкою

```
numbers = [1, 3, 5, 7]
numbers.append(9) # [1, 3, 5, 7, 9]
numbers.extend([11, 13]) # [1, 3, 5, 7, 9, 11, 13]
numbers.pop().pop() # [1, 3, 5, 7, 9]
numbers.reverse() # [9, 7, 5, 3, 1]
numbers.sort() # [1, 3, 5, 7, 9]

# Contrast to:
reversed(numbers)
sorted(numbers)
```


Детальніше про структури даних

- ▶ Списки (list),
- ▶ Словники (dict)
- ▶ Множини (set)
- ▶ Кортежі (tuple)

Comprehensions

- ▶ list comprehensions

```
squares = [x**2 for x in range(5)]  
# [0, 1, 4, 9, 16]
```

```
list(map(lambda x: x**2, range(10)))
```

- ▶ set comprehension (Python 3)

```
{x for x in [5, 5, 2, 2]}  
# {2, 5}
```

- ▶ dict comprehension (Python 3)

```
{name: i for i, name in enumerate(['a', 'b'])}  
# {'a': 0, 'b': 1}
```

Generator expression

► Генератор vs. список

```
(x**2 for x in range(5))  
# <generator object <genexpr> at 0x7f516cb622d0>  
[x**2 for x in range(5)]  
# [0, 1, 4, 9, 16]
```