Іменні функції, інструкція def

Функція в Python - об'єкт, який приймає аргументи і повертає значення. Зазвичай функція визначається за допомогою інструкції def.
Визначимо найпростішу функцію:

```
def add (x, y):
    return x + y
```

Інструкція return вказує, що потрібно повернути значення. У нашому випадку функція повертає суму значень двох змінних, які функція отримує в якості параметрів і для себе називає їх **x** та **y**.

Приклад як ми можемо викликати дану функцію, приклад програми:

```
def add (x, y):
    return x + y

a = add(1, 10)
print(a)
```

Результат: 11

Ось приклад виклику цієї функції у випадку, коли ми передаємо даній функції текстові дані:

```
def add (x, y):
    return x + y

a = add('все', 'добре')
print(a)
```

Результат: 'вседобре'

Функція може бути будь-якої складності і повертати будь-які об'єкти (списки, кортежі, і навіть функції). Функція може і не закінчуватися інструкцією return, при цьому функція поверне значення **None**.

Аргументи функції

Функція може приймати будь-яку кількість аргументів чи не приймати їх зовсім. Також поширені функції з довільним числом аргументів, функції з позиційними і іменованими аргументами, обов'язковими і необов'язковими.

Приклади:

```
def func(a, b, c=2): # c – необов’язковий аргумент
    return a + b + c


func(1, 2)  # a = 1, b = 2, c = 2 (по замовчуванню)
```

Результат: 5

```
func(1, 2, 3)  # a = 1, b = 2, c = 3
```

Результат: 6

```
func(a=1, b=3)  # a = 1, b = 3, c = 2
```

Результат: 6

Функція також може приймати змінну кількість позиційних аргументів, тоді перед ім'ям ставиться *:

```
def func(*args):
    return args
```

Приклад:

```
def add (*args):
    return sum(args)

a = add(1,2,4)
print(a)
```

Результат: 7

Анонімні функції, інструкція lambda

Анонімні функції можуть містити лише один вислів, але і виконуються вони швидше. Анонімні функції створюються за допомогою інструкції **lambda**. Крім цього, їх не обов'язково привласнювати змінній, як з інструкцією def func ():

```
func = lambda x, y: x + y
print(func(1, 2))
```

Результат: 7

```
func = lambda x, y: x + y
func('a', 'b')
```

Результат: 'ab'

```
print((lambda x, y: x + y)(1, 2))
```

Результат: 3

lambda функції, на відміну від звичайної, не потрібна інструкція **return**