



НПГ по КТС гр.Правец

ТЕМА

2

Основи на програмирането
с Python

ФУНКЦИИ

PYTHON



Автор: Елена Първанова

PYTHON



Определение за функция

Какво представлява функцията?

- Когато пишем код улесняваме нашата работа като разделяме програмата на части.
- **функция** (метод) се нарича **именувана група от инструкции**, които изпълняват дадена функционалност.
- Стартирането на изпълнението на инструкциите във функцията се нарича **извикване на функцията**.
- Една функция може да бъде извикана толкова пъти, колкото ние е необходимо.
- Функцията *спестява повторението на един и същи код* няколко пъти, което от своя страна е лоша практика.

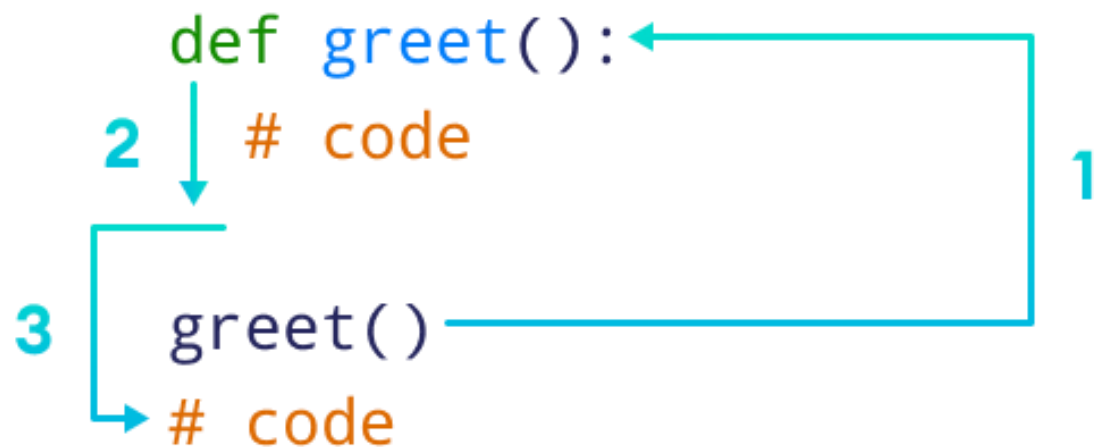
PYTHON



Дефиниране на функции

Дефиниране на функции

- **Дефиниране на функция** представлява регистрирането на функцията в програмата, за да може впоследствие да бъде.



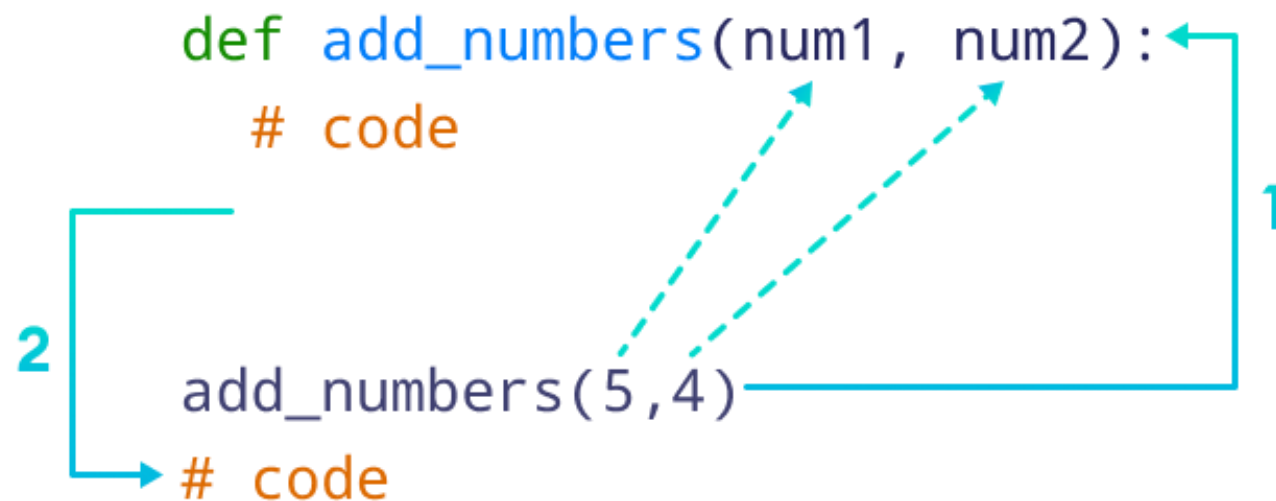
- **def** - ключовата дума в Python, с която се дефинира функция
- **име на функцията** - определя се програмиста
- **списък с параметри** - изброява се поредицата от параметри, които функцията ще използва в (). Може да има няколко, един или да липсват
- **двуточие** - следва тялото на функцията
- **тялото на функцията** - описва алгоритъма (инструкциите) на задачата, т.е това е кода, който реализира логиката на функцията

Извикване на функции

- Извикването на функция **стартира изпълнението на кода**, който се намира в тялото на функцията.
- Това става с **името на функцията**, кръглите скоби **()** и **подадените параметри**.
- Ако функцията няма параметри (входни данни), скобите са празни.
- Ако функцията има параметри, те се подават в скобите **()**, те се *изброяват със запетая*.

Параметри във функциите

- При декларирането на параметри, се използват *различни типове данни* като всеки един параметър трябва да има име (което да е смислено).

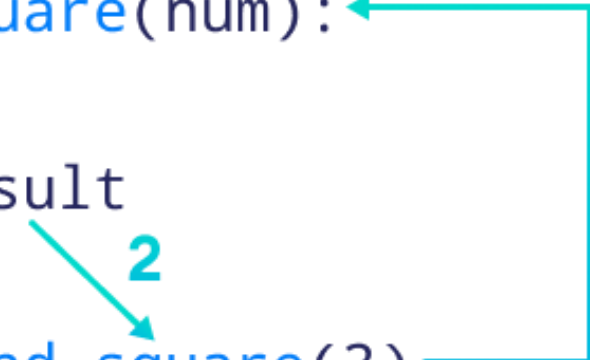


- При извикване на функцията, трябва се подават параметрите в реда, в който са декларирани. Те се наричат **аргументи на функцията**.

Връщане на резултат от функция

- Връщането на резултат от функция става с оператор **return**.
- След return не трябва да се записва друг оператор, защото указва на програмата да **спре изпълнението** си.

```
def find_square(num):  
    # code  
    return result  
  
Square = find_square(3)  
# code
```



The diagram consists of two teal arrows. Arrow 1 starts at the 'return result' line in the function definition and points to the 'find_square(3)' call in the code below. Arrow 2 starts at the 'return result' line and points to the 'result' variable, indicating the value being returned.

След като дадена функция е изпълнена и върне стойност, то тази стойност може да се **присвои на променлива**.

Връщане на резултат с няколко стойности

```
def divideNumbers(x, y):  
    result = x // y  
    reminder = x % y  
    return result, reminder
```

```
num1, num2 = divideNumbers(27, 5)  
print(num1)  
print(num2)
```

Резултат:

5
2

- Когато се връщат няколко стойности, те се записват със запетая в оператор **return**.
- При извикването на функцията, върнатите стойности се присвояват **на няколко променливи**

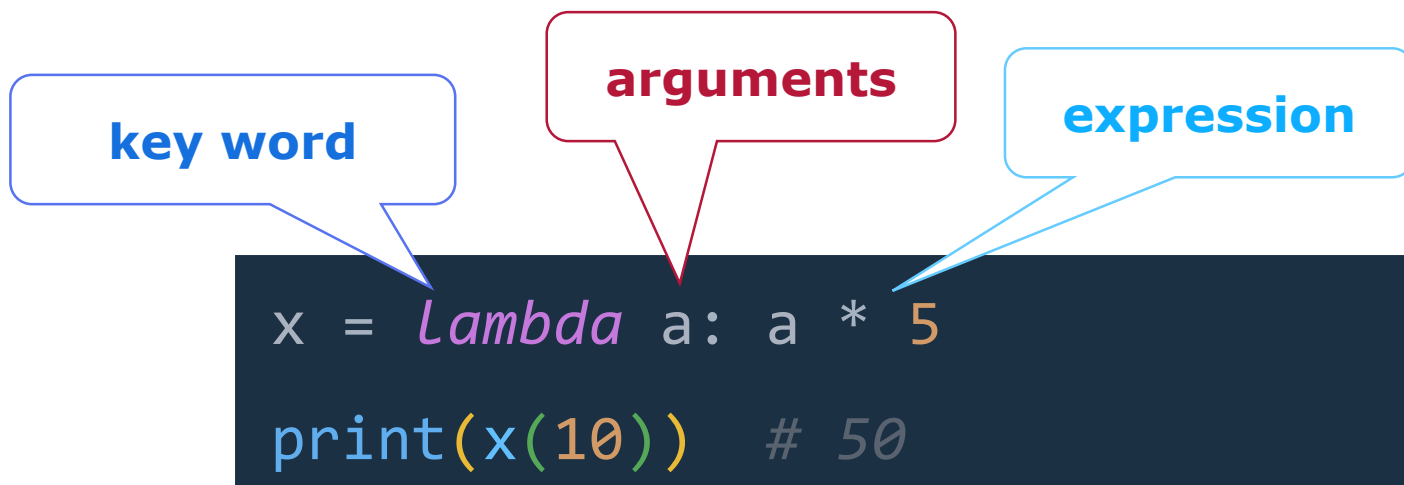
PYTHON



Lambda функция

Функция `lambda`

- В Python **`lambda`** функцията е специален тип функция без име.
- Подобно на функция, тя може да приема параметъри и да върне резултат.



The diagram illustrates the components of the lambda function call in the code snippet below. Three callout boxes are present: a blue box labeled 'key word' pointing to the word 'lambda', a red box labeled 'arguments' pointing to the parameter 'a', and a blue box labeled 'expression' pointing to the expression 'a * 5'.

```
x = lambda a: a * 5  
print(x(10)) # 50
```

Множество параметри с lambda

- Функция **lambda** може да приема множество параметри.
- Колкото са параметрите, толкова са и аргументите при извикване.

```
x = lambda a, b: a * b  
print(x(3, 5)) # 15
```