

# Python (#5)

*Nemanja Maček*

- “Kako bi se agresija Vaše zmije svela na najmanju moguću meru možete je hraniti izvan terarijuma u priručnoj kutiji čime se zmija navikava na kontakt, a ona sama se oseća sigurnom i izvan terarijuma.” [1]



Slika preuzeta sa: <https://www.xyzreptiles.com/product/baby-super-mystic-ball-python/>

- Funkcije
- Primeri nekih funkcija

## Primer funkcije.

```
def printt():  
    print("This is Python Tutorial")  
    print("This is Python Tutorial")
```

- Pozivanje funkcije:

```
printt()
```

- Izlaz:

```
This is Python Tutorial  
This is Python Tutorial
```

## Primer funkcije sa argumentima.

```
def sumofsquares(x, y):  
    return (x*x + y*y)
```

- Pozivanje funkcije:

```
print("The sum squares of 3 and 4 is : ", sumofsquares(2, 3))
```

- Izlaz:

```
The sum squares of 3 and 4 is : 25
```

## Podrazumevane vrednosti argumenata.

```
def sumofsquares(x, y = 2):  
    return (x*x + y*y)
```

- Pozivanje funkcije:

```
print("The sum squares of 3 and 2 is : ", sumofsquares(3))  
print("The sum squares of 3 and 4 is : ", sumofsquares(3, 4))
```

- Izlaz:

```
The sum squares of 3 and 2 is : 13  
The sum squares of 3 and 4 is : 25
```

## Prosleđivanje vrednosti po ključnim rečima.

```
def marks(english, math = 85, science = 80):  
    print('English is ', english, ', Math ', math, ', Science ', science)
```

- Pozivanje funkcije:

```
marks(71, 77)  
marks(65, science = 74)  
marks(science = 70, math = 90, english = 75)
```

- Izlaz:

```
English is 71 , Math 77 , Science 80  
English is 65 , Math 85 , Science 74  
English is 75 , Math 90 , Science 70
```

## Funkcije sa fiksno-neodređenim brojem argumenata.

```
def sum(*numbers):  
    s = 0  
    for n in numbers:  
        s += n  
    return s
```

- Pozivanje funkcije:

```
print(sum(1,2,3,4))
```

- Izlaz:

10



## Rekurzivne funkcije.

```
def factorial(n):  
    if n <= 1:  
        return 1  
    else:  
        return n * (factorial(n - 1))
```

- Pozivanje funkcije:

```
print(factorial(5))
```

- Izlaz:

120

## Lambda.

- Primer funkcije:

```
def average(x, y):  
    return (x + y)/2  
print(average(4, 3))
```

- U lambda formi:

```
print((lambda x, y: (x + y)/2)(4, 3))
```

## Lambda.

- Gde se koristi lambda?
  - Lambda izrazi su pogodni za upotrebu kao argument druge funkcije gde su objekti koji se prosleđuju sintaksno limitirani na jedan izraz.

## Primer 1.

- Definisati funkciju koja množi sve elemente liste koja je prosleđena kao argument.

```
def multiply(numbers):  
    total = 1  
    for x in numbers:  
        total *= x  
    return total
```

## Primer 2.

- Definisati funkciju koja invertuje string koji je prosleđen kao argument.

```
def string_reverse(str1):  
    rstr1 = ''  
    index = len(str1)  
    while index > 0:  
        rstr1 += str1[ index - 1 ]  
        index = index - 1  
    return rstr1
```

## Primer 3.

- Definisati funkciju koja proverava da li broj pripada datom opsegu. Podrazumevani opseg je (3,9) ukoliko korisnik drugačije ne naznači.

```
def test_range(n,x=3,y=9):  
    if n in range(x,y):  
        print( " %s is in the range"%str(n))  
    else :  
        print("The number is outside the given range.")
```

## Primer 4.

- Definisati funkciju koja vraća listu u kojoj se elementi liste prosleđene kao argument ne ponavljaju.

```
def unique_list(l):  
    x = []  
    for a in l:  
        if a not in x:  
            x.append(a)  
    return x
```

## Primer 5.

- Definisati funkciju koja proverava da li je broj prosleđen kao argument prost.

```
def test_prime(n):
    if (n==1):
        return False
    elif (n==2):
        return True;
    else:
        for x in range(2,n):
            if(n % x==0):
                return False
        return True
print(test_prime(9))
```



## Primer 6.

- Definisati funkciju koja vraća listu sačinjenu od svih parnih brojeva koji se nalaze u listi koja je prosleđena kao argument.

```
def is_even_num(l):  
    enum = []  
    for n in l:  
        if n % 2 == 0:  
            enum.append(n)  
    return enum
```

## Primer 7.

- Definisati funkciju proverava da li je string prosleđen kao argument palindrom ili ne.

```
def isPalindrome(string):
    left_pos = 0
    right_pos = len(string) - 1
    while right_pos >= left_pos:
        if not string[left_pos] == string[right_pos]:
            return False
        left_pos += 1
        right_pos -= 1
    return True
```

## Primer 8.

- Definisati rekurzivnu funkciju koja računa sumu cifara ne-negativnog celog broja prosleđenog kao argument.

```
def sumDigits(n):  
    if n == 0:  
        return 0  
    else:  
        return n % 10 + sumDigits(int(n / 10))
```

## Primer 9.

- Definisati rekurzivnu funkciju koja računa sumu pozitivnih brojeva  $n + (n-2) + (n-4) + \dots$ , sve dok je  $(n-x) \leq 0$ , pri čemu se vrednost  $n$  prenosi kao argument.

```
def sum_series(n):  
    if n < 1:  
        return 0  
    else:  
        return n + sum_series(n - 2)
```

## Primer 10.

- Definisati rekurzivnu funkciju koja vraća najveći zajednički delilac dva broja koji su prosleđeni kao argumenti.

```
def Recurgcd(a, b):  
    low = min(a, b)  
    high = max(a, b)  
    if low == 0:  
        return high  
    elif low == 1:  
        return 1  
    else:  
        return Recurgcd(low, high%low)
```

**Pitanja su dobrodošla.**

1. <http://moj.pet-centar.rs/Teraristika/5404-Kraljevski-piton.html>
2. Milan Bjelica (2016): Programski jezik Python. Dostupno u sekciji “download” na sledećem linku: [http://www.etf.bg.ac.rs/etf\\_files/udzbenici/python.pdf](http://www.etf.bg.ac.rs/etf_files/udzbenici/python.pdf)
3. Charles R. Severance (2016): Python for Everybody – Exploring Data Using Python 3. Dostupno u sekciji “download” na stranici predmata i na sledećem linku: [http://do1.dr-chuck.com/pythonlearn/EN\\_us/pythonlearn.pdf](http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf)
4. A Bite of Python. Dostupno u sekciji “download” na sledećem linku (na engleskom jeziku): <https://python.swaroopch.com/>.
5. <https://www.w3resource.com/python/python-tutorial.php>