

Funkcije i uvod u fajlove



Softverske i informacione tehnologije, 1. godina osnovnih strukovnih studija

- Funkcija je blok (parče) koda koji se pokreće samo onda kada je pozvan.
- Do sada ste videli funkciju *main* u kojoj ste pisali čitavo izvršavanje vaših programa.
- Ideja jeste da učinite delove programa ponovno iskoristivim tako što ćete pozivati funkcije kada su vam one potrebne.

```
1  def main():  
2      |    print("Hello World!")  
3  
4  main()
```

- Funkciji možete prosleđivati vrednosti poznatije kao argumenti.
- Funkcija vam može vraćati vrednosti (jednu ili više) kao odgovor.
- U Python-u funkcija se definiše ključnom reči *def*.

```
1  def my_function():  
2      print("Hello from a function")  
3  
4  my_function()  
5
```

Poziv funkcije



Argumenti

- Informacije (parametri) mogu biti prosleđeni funkciji kao argumenti.
- Argumenti se specificiraju nakon imena funkcije, unutar zagrada.
- Možete dodati koliko želite parametara samo ih razdvojte zarezima.

```
1  def my_function(fname):  
2      print(fname + " Refsnes")  
3  
4      my_function("Emil")  
5      my_function("Tobias")  
6      my_function("Linus")  
7
```

- Funkcija mora biti pozvana sa onoliko argumenata koliki je broj parametara definisan u funkciji.
- Terminologija: Parametri <-> Argumenti
 - Termin parametari i argumenti se mogu koristiti za istu stvar: informacije koje su prosleđene funkciji.
 - Moglo bi se reći da je parametar varijabla izlistana unutra zagrada definicije funkcije, dok su argumenti vrednosti koje su prosleđene funkciji prilikom njenog poziva.

```
1  def my_function(fname, lname):  
2      |   print(fname + " " + lname)  
3  
4      my_function("Emil", "Refsnes")  
5
```

Povratne vrednosti

- Kako bi funkcija vraćala vrednost koristi se *return*.

```
1  def my_function(x):  
2  |    return 5 * x  
3  
4  print(my_function(3))  
5  print(my_function(5))  
6  print(my_function(9))
```

Fajlovi


- Python pruža podršku za kreiranje, čitanje, ažuriranje (update) i brisanje fajlova.
- Ključna funkcija za rad sa fajlovima u Python-u jeste *open()* funkcija.

```
f = open("demofile.txt")
```

- Postoje četiri režima otvaranja fajla:

- “r”- read – Otvara fajl za čitanje, prijavljuje grešku ukoliko fajl ne postoji.
- “a” – append – Otvara fajl za dodavanje novog sadržaja ukoliko fajl ne postoji kreira ga.
- “w” – write – Otvara fajl da piše u njega (sve prethodno se briše), kreira fajl ukoliko ne postoji.
- “x” – create – Kreira fajl, prijavljuje grešku ukoliko fajl postoji.

- Dodatno možemo specificirati sa kojim tipom fajla radimo.
 - “t” – Text – tekstualni fajl, default vrednost.
 - “b” – Binary – binarni fajl.

`f = open("demofile.txt")`  `f = open("demofile.txt", "rt")`

Dakle koristićemo open funkciju na sledeće **načine**.

`f = open("demofile.txt", "r")`

`f = open("demofile.txt", "a")`

`f = open("demofile.txt", "w")`

Primeri

Čitanje fajla

```
2 def citanje_fajla():
3     file = open('myfile.txt', 'r')
4     lines = file.readlines()
5
6     count = 0
7     for line in lines:
8         count = count + 1
9         print(count, line.strip())
10
```

Pisanje u fajl

```
12 def pisanje_u_fajl():
13     primer = ["Ovo\n", "je\n", "primer\n"]
14     file = open('myfile.txt', 'w')
15     file.writelines(L)
16     file.close()
17
```

Dodavanje u fajl

```
18
19 def dodavanje_u_fajl():
20     dodatak = ["dodaj\n", "ovo\n", "u fajl\n"]
21     file = open('myfile.txt', 'a')
22     file.writelines(L)
23     file.close()
24
```