



**ALGEBRA**


**PROGRAMIRANJE**  
Predavanje 06

Ishod 3

1

**FUNKCIJE**

Strana • 3



3

## Uvod

- Funkcija je dio imenovani kôda koji napišemo jednom, a zatim koristimo više puta u programu
- Implementira princip redukcije (ili "**podijeli pa vladaj**")
  - Princip dijeljenja glavnog problema na manje probleme sve do lako rješivih problema
  - Spajanjem rješenja više problema niže razine rješavamo problem više razine
- Dvije glavne prednosti funkcija
  - Omogućuju razbijanje složenijih problema u jednostavnije
  - Omogućuju ponovnu iskoristivost dijelova kôda (engl. *reuse*)

Strana • 4



4

## Princip crne kutije

- Jedna funkcija koristi drugu kao **crnu kutiju**
- Korisnik crne kutije zna sljedeće:
  - Što crna kutija radi
  - Koje joj parametre mora predati
  - Što će dobiti kao rezultat (što će kutija učiniti)
- Korisnika crne kutije **ne zanima način rada kutije**
- Primjer crne kutije za množenje:
  - Funkcija: množi cijele brojeve
  - Parametri: dva cijela broja
  - Rezultat: umnožak proslijeđenih brojeva
- Primjer crne kutije koju smo koristili: algoritam `print()`

Strana • 5



5

## Princip crne kutije

- U izradi programa ćemo jedan program dijeliti na manje cjeline zvane **funkcije** (engl. *function*)
  - **Funkcija je imenovani dio kôda** koji rješava manji dio početnog problema
  - U drugim programskim jezicima i programskim paradigmama funkcija se može nazivati i **rutina, potprogram, procedura, metoda, ...**
  - Funkcije se koriste po principu crne kutije (poznato je što prima i vraća te što radi, ali ne i kako to radi)

Strana • 6



6

## Ugrađene funkcije

- Python dolazi s nizom funkcija spremnih za korištenje
  - Neke su ugrađene u sam jezik, neke su dostupne u dodatnim paketima
    - Već smo upoznali funkcije `print()`, `type()`, `list()` ...
  - Nekoliko dodatnih korisnih ugrađenih funkcija:
    - `abs(broj)` – vraća apsolutnu vrijednost broja
    - `pow(baza, eksponent)` – podiže broj na zadanu potenciju
    - `min(kontejner)` – vraća najmanju vrijednost u kontejneru
    - `max(kontejner)` – vraća najveću vrijednost u kontejneru
    - `round(broj, znamenki)` – zaokružuje broj
    - `format(tekst, specifikacija)` – vraća formatirani tekst
- Strana • prema zadanoj specifikaciji



7

## Detalji funkcije format()

- Funkcija prolazi kroz zadani tekst i svaki *placeholder* zamijeni sa vrijednostima zadanim iza
  - U *placeholderima* se navode imenima unutar vitičastih zagrada
  - Opcionalno se mogu navesti redni brojevi ili čak i ništa

- Najbolja opcija:

```
msg = 'Vrijeme {prognoza}, temperatura {temp} stupnjeva'
      .format(prognoza='sunčano', temp=37)
print(msg)
```

- Ostale opcije:

```
msg = 'Vrijeme {0}, temperatura {1} stupnjeva'
      .format('sunčano', 37)
msg = 'Vrijeme {}, temperatura {} stupnjeva'
      .format('sunčano', 37)
```

Strana • 8



8

## VLASTITE FUNKCIJE

Strana • 9



9

## Sastavni elementi funkcija

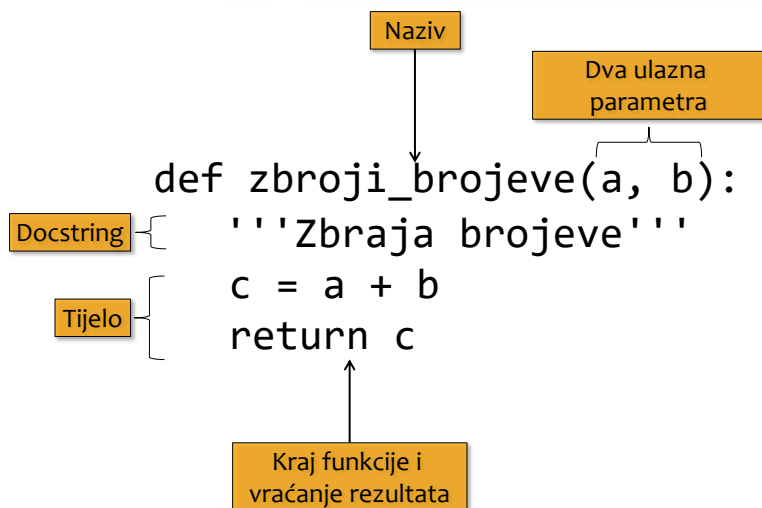
- Funkcija se sastoji od sljedećih elemenata:
  - Ključna riječ **def**
  - **Naziv** – kao kod imenovanja varijabli
  - **Ulazni parametri (argumenti)** – nula ili više naziv odvojenih zarezima
  - **Docstring** – opcionalni tekst koji opisuje što funkcija radi
  - **Tijelo funkcije** – niz naredbi za rješavanje problema
  - **Naredba return** – opcionalna naredba, ima dvije funkcije:
    - Završava izvršavanje funkcije
    - Opcionalno vraća vrijednost pozivatelju

Strana • 10



10

## Primjer funkcije



Strana • 11



11

## Pozivanje funkcija

- Funkciju pozivamo (koristimo) tako da:
  1. Navedemo njezin naziv
  2. Iza naziva u okruglim zagradama navedemo vrijednosti ulaznih parametara (ako nema parametara moramo pisati prazne zagrade)
    - Redoslijed vrijednosti mora odgovarati redoslijedu parametara
    - Druga opcija su imenovani parametri (kasnije)
- Kao vrijednost ulaznih parametara možemo navesti bilo koji izraz
  - Izraz čini kombinacija literala, operatora i varijabli
- Funkcija se može pozvati tek nakon što je definirana
  - Dakle, prvo napišemo funkcije, a tek nakon toga je pozivamo

Strana 12



12

## Globalne i lokalne varijable (pojednostavljeno)

- Varijable definirane izvan funkcija su globalne varijable
  - Vidljive su u svakoj funkciji
  - Funkcija ih može čitati, ali ih ne može mijenjati (ali ako je kontejner, može manipulirati elementima)
  - Ako ih želi mijenjati, mora koristiti ključnu riječ `global`
- Varijable definirane unutar funkcije su lokalne varijable
  - Vidljive su samo u toj funkciji
  - Uništavaju se kad završi izvršavanje funkcije
  - Parametri se također smatraju lokalnim varijablama
- Ako funkcija definira varijablu istog imena kao što ga ima postojeća globalna varijabla, nova lokalna varijabla će skriti globalnu varijablu

Strana 13



13

## Primjer 1

```
ime = 'Marija'

def f1():
    godina = 29
    print(ime)
    print(godina)

print(ime)
f1()
print(ime)
```

Strana \* 14



14

## Primjer 2

```
ime = 'Marija'

def f1():
    ime = "Jurica"
    godina = 29
    print(ime)
    print(godina)

print(ime)
f1()
print(ime)
```

Strana \* 15



15

## Primjer 3

```
ime = 'Marija'

def f1():
    global ime
    ime = "Jurica"
    godina = 29
    print(ime)
    print(godina)

print(ime)
f1()
print(ime)
```

Strana \* 16



16

## Primjeri

9. Napišite funkciju koja prima broj i vraća njegov faktorijel. Učitajte od korisnika broj i ispišite njegov faktorijel pozivom funkcije.
10. Napišite funkciju koja prima string i vraća isti string, ali tako da se između svaka dva znaka nalazi crtica. Primjerice, ako funkcija primi „med”, neka vrati „m-e-d”. Učitavajte od korisnika string dok to želi i ispisujte rezultate koje vrati funkcija.
11. Napišite funkciju koja prima string sastavljen samo od malih slova i vraća koliko kojih samoglasnika ima u tom stringu.

Strana \* 17



17



## Primjeri

12. Napišite funkciju koja prima string i vraća koliko kojih znakova ima u tom stringu.
13. Napišite funkciju koja prima listu brojeva i vraća novu listu sastavljenu samo od parnih brojeva.
14. Napišite funkciju koja prima broj  $n$  i iscrtava prazni kvadrat stranica  $n$ .
15. Napišite funkciju koja prima tri broja i vraća najmanjeg od njih.
16. Napišite funkciju koja prima listu brojeva i vraća novu listu sastavljenu samo od jedinstvenih brojeva (duplikati se uklanjaju).

Strana • 18



18

## Za sljedeće predavanje

- ✓ Ponoviti sve iz ovog predavanja
- ✓ Pročitati i isprobati primjere:
  - <https://www.programiz.com/python-programming/function>
  - <https://www.programiz.com/python-programming/methods/built-in>
    - Samo one koje smo obradili na predavanjima

Strana • 19



19