

PROGRAMIRANJE 1

Funkcije i biblioteke

doc. dr. sc. Tomislav Volarić
mag. ing. comp. Robert Rozić



Creative Commons

- **Slobodno smijete:**
 - **dijeliti** - umnožavati, distribuirati i javnosti priopćavati djelo
 - **remiksirati** - prerađivati djelo
- **pod sljedećim uvjetima:**
 - **Imenovanje.** Morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
 - **Nekomercijalno.** Ovo djelo ne smijete koristiti u komercijalne svrhe.
 - **Dijeli pod istim uvjetima.** Ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, prerađivanje možete distribuirati samo pod licencom koja je ista ili slična ovoj.

U slučaju daljnjeg korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela. Najbolji način da to učinite je linkom na ovu internetsku stranicu. Od svakog od gornjih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava.

Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava.

Tekst licence preuzet je s <https://creativecommons.org/>.



Procedure

Procedura je kolekcija instrukcija koje se mogu izvršiti bilo gdje unutar programa.

Postoje brojni razlozi i prednosti korištenja procedura:

- Smanjiti ponavljanja (redundancije) istih ili sličnih sekvenci programskog koda.
- Omogućiti ponovnu uporabu dijelova programskog koda u različitim programima.
- Dekompozicija složenih problema u manje i jednostavnije cjeline.
- Poboljšanje "čitljivosti" i preglednosti programa.
- Repliciranje rješenja za matematičke funkcije.
- Sakrivanje informacija i nevažnih detalja unutar pojedinih dijelova programa.
- Smanjenje rizika od grešaka, lakše održavanje i nadogradnja velikih sustava.

Funkcije

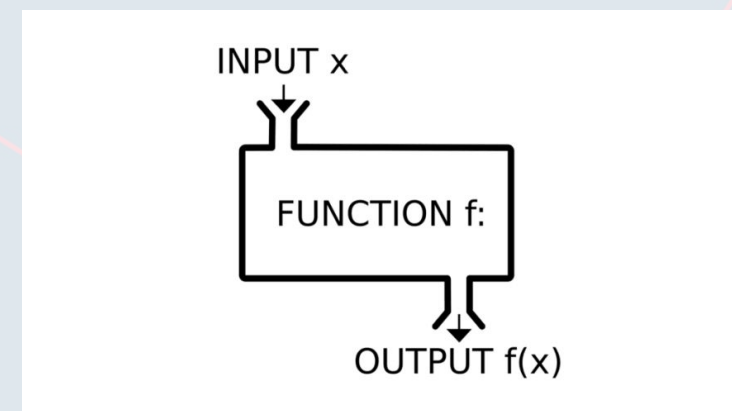
U Python-u funkcija (engl. function) je grupa povezanih iskaza koji izvršavaju određeni zadatak.

- blok organiziranog koda koji se može ponovno koristiti (reusable)
- obavlja jednu određenu akciju
- omogućuju bolju **modularnost** aplikacije
- pisanje čitljivijeg i bolje organiziranog koda
- Korištenjem funkcija izbjegavamo ponavljanje koda.
- **DRY!** - Don't Repeat Yourself

Python pruža brojne **ugrađene** funkcije npr. poput `print()`, `abs()`...

Korisnik može kreirati **vlastite** funkcije.

Paradigma - **Funkcionalno programiranje**



Definiranje funkcije

```
def <ime_funkcije>(<parametri>):  
    <tijelo funkcije>
```

Ime funkcije mora pratiti python pravila.

Funkcija može imati 0 ili više parametara.

Funkcija može, ali ne mora vraćati vrijednosti - tada koristimo naredbu return

- naredba return može se pojaviti bilo gdje u tijelu funkcije
- završava se izvršavanje funkcije i vraća rezultat pozivatelju

```
def <ime_funkcije>(<parametri>):  
    <tijelo funkcije>  
    return <vrijednost>
```

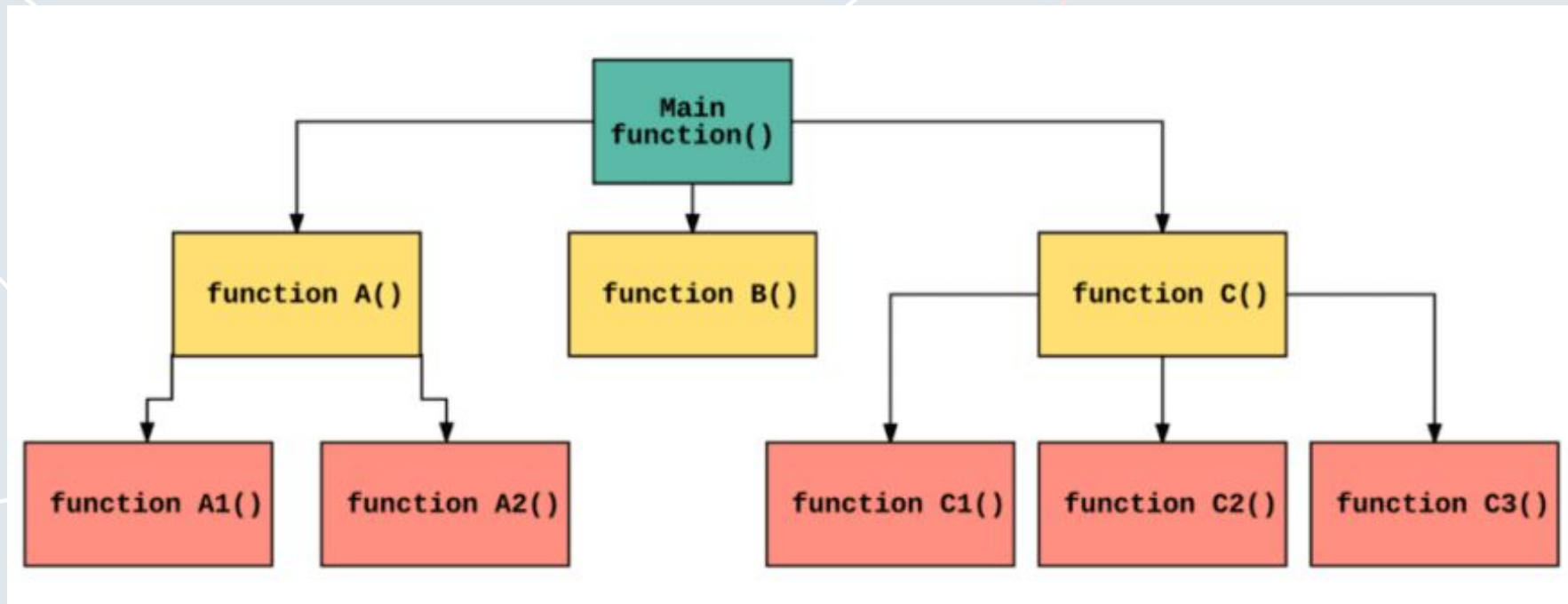
Definiranje funkcije

```
def pozdrav(ime):  
    print("Dobar dan " + ime);  
  
>>> pozdrav("Ivan")
```

Funkcija koja vraca vrijednost

```
def zbroj(broj):  
    if broj >= 0:  
        return broj  
    else:  
        return -broj  
  
>>> print(apsolutna(-12))
```

Modularnost



Aritmetičke funkcije

Funkcija	Opis	Primjena	Ispis
abs()	apsolutna vrijednost	<code>abs (-5)</code>	5
max()	najveći broj u nizu	<code>max (15, -9, 56, 123, 0)</code>	123
min()	najmanji broj u nizu	<code>min (15, -9, 56, 123, 0)</code>	-9

Funkcije nad stringovima

Funkcija	Opis	Primjena	Ispis
len()	duljina stringa (i niza)	<code>len("abcde")</code>	5
min()	znak s najmjanjom kodnom vrijednošću	<code>min("abcde")</code>	a
max()	znak s najvećom kodnom vrjednošću	<code>max("abcde")</code>	e
ord()	brojčani kod za znak	<code>ord("A")</code>	65
chr()	znak koji odgovara brojčanom kodu	<code>chr(65)</code>	"A"

```
>>>min("abcdEfg")  
???
```

Metoda vs. Funkcija

- **Metoda je pojam u OOP (objektno orijentirano programiranje)**
- **u Javi i C# postoje samo metode**
- **u C - samo funkcije**
- **C++ i Python oboje - ovisno jesmo li u objektu/klasi(class)**
- **Sve metode su funkcije ali nisu sve funkcije metode.**

Metode sa stringovima

- **Zamjena dijela stringa - `string.replace(arg1, arg2)`**

```
str = 'jabuka, banana'  
print(str.replace('banana', 'naranca'))
```

- **Pozicija prvog pojavljivanja - `string.find(needle)`**

```
str = 'Danas je četvrtak.'  
print(str.find("je"))
```

- **Pretvaranje u velika i mala slova - `str.lower()` i `str.upper()`**

```
str = 'Danas je četvrtak.'  
print(str.lower())  
print(str.upper())
```

Moduli (zbirke funkcija)

Matematičke funkcije s realnim brojevima: math

Matematičke funkcije s kompleksnim brojevima: cmath

Slučajni brojevi: random

Pozivaju se na 2 načina:

1) Na vrhu programa uvodimo:

**import <ime_biblioteke>,
a funkcije pozivamo na način: ime_biblioteke.ime_funkcije**

```
import math  
x = 16  
print("Korijen: ", math.sqrt(x))
```

Moduli (zbirke funkcija)

2) Nacin importa

```
from <ime_biblioteke> import <ime_funkcije>
```

```
<ime_funkcije>(x)
```

Osnovne funkcije Math biblioteke

- **sqrt(x)**
- **pi**
- **sin(x)**
- **cos(x)**
- **tan(x)**
- **exp(x)**
- **log(x,b)**

Osnovne funkcije Random biblioteke

randint(a,b) – generiranje slučajnog cijelog broja iz intervala $[a, b]$

random() – generiranje slučajnog realnog broja iz intervala $[0, 1]$

randrange(start,stop,step) – generiranje slučajnog cijelog broja iz intervala $[start, stop]$, s tim da se od početne vrijednosti povećava za korak **step**

Ako se korak ne navede, podrazumijeva se da je 1.

- **Primjer:**

random.randrange(2,100,2)

Birat će brojeve iz $[2,100]$ s korakom 2, što znači da će se birati samo parni brojevi.

Zadatak 1.

Napisati program koji će ispisati umnožak 5 slučajno odabranih brojeva od 1 do 100.

Zadatak 2.

**Računalo zamisli jedan broj od 1 do 10. Korisnik ga pokušava pogoditi.
Ukoliko ga pogodi ispisati „Pogodak”, inače „Broj je veći od zamišljenog” ili „Broj je manji od
zamišljenog”.**

Zadatak

**Napisati funkciju koja vraća je li unesena riječ/rečenica palindrom.
Pripaziti na velika i mala slova, te ignorirati razmake.
Koristiti naučene metode i operatore za rad sa nizovima i stringovima.**

**Palindrom (obrtaljka) - riječ ili fraza koja se može čitati i naopako, a da joj se značenje ne promijeni.
npr. oko, kapak, kuk, “Ana voli Milovana”, “Ana nabra par banana.”**

Literatura



- Think IT 1, (2019.) Udžbenik iz informatike za prvi razred gimnazije, ALFA d. d.
- <https://www.programiz.com/python-programming/function>
- https://www.tutorialspoint.com/python/python_functions.htm

sumarum.sum.ba



Hvala na pozornosti!



tomislav.volaric@fpmoz.sum.ba

robert.rozic@fpmoz.sum.ba