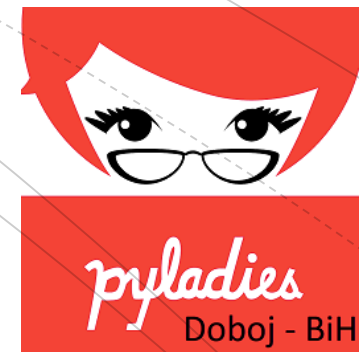
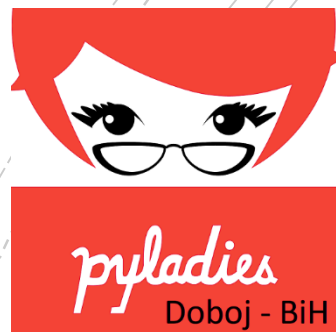


Sponsor



python SOFTWARE  
FOUNDATION





Virtualna radionica

Osnove Pythona i  
analiza i  
vizualizacija  
podataka

SPONZOR



MENTORI

- Tijana Paunović
- Jelena Mitrović
- Siniša Bubonja

WHILE PETLJA

# PONAVLJANJE NAREDBI - WHILE

---

- Python pruža mehanizam da se jedna ili više naredbi ponovi određen broj puta ili dok je neki uslov ispunjen

```
while(uslov):  
    naredba1  
    naredba2
```

- Naredbe se izvršavaju sve dok je uslov tačan.
- Uslov ne mora biti u zagradi.



# PONAVLJANJE NAREDBI -WHILE

---

Ako želimo 5 puta da šampamo 'Zdravo!'

```
br=0  
while br<5:  
    print('Zdravo!')  
    br=br+1
```

```
Zdravo!  
Zdravo!  
Zdravo!  
Zdravo!  
Zdravo!
```



# PONAVLJANJE NAREDBI -WHILE

---

Šta radi dati deo koda?

```
n=5
while n > 0:
    print(n)
    n = n-1    #šta se dešava bez ovog reda?
print("Gotovo!")
```



# PRIMER

---

Šta je rezultat sledećeg koda?

```
i = 1
while i < 6:
    print(i)
    if i == 3:
        break
    i = i + 1
```

Odgovor:

```
1
2
3
```

# PRIMER

---

Šta je rezultat sledećeg koda?

```
i = 0
while i < 6:
    i = i + 1
    if i == 3:
        continue
    print(i)
```

Odgovor:

1  
2  
4  
5  
6



# PRIMER

---

Šta je rezultat sledećeg koda?

```
i= 1
while i<= 6:
    print(5*i, '\t', end=' ')
    i= i+ 1
print()
```

Odgovor:      5          10          15          20          25          30



# PRIMER

---

Koliko puta će se ponoviti ciklus i šta će biti vrednosti promenljivih **a**, **b** i **s** posle izvršenja ovog niza naredbi?

```
a = 1
b = 1
while a+b<8:
    a = a+1
    b = b+2
s = a+b
print (s)
```

Odgovor: 8



# ZADACI

---

## zadatak\_while\_1

Napisati program koji vrši ispis prirodnih brojeva od 1 do 17 koji su deljivi sa 3.

## zadatak\_while\_2

Izračunati zbir unetih brojeva. Brojeve unositi dok se ne unese 0.

Primer testiranja:

```
Unesi broj: 6
Unesi broj: 7
Unesi broj: 5
Unesi broj: 5
Unesi broj: 4
Unesi broj: 9
Unesi broj: 0
Zbir unetih brojeva je 36
```



# ZADACI

---

## zadatak\_while\_3

Napiši program koji unosi imena studenata sve dok se ne unese prazan string i na kraju prijavljuje koliko imena je uneto.

## zadatak\_while\_4

Napisati program koji uneti ceo broj *n* ispisuje sa ciframa u obrnutom poretaku.

Primer testiranja:      `Unesi broj: 123456`  
                              `654321`



# FUNKCIJE

# FUNKCIJA

---

- Funkcija je samostalan deo programa koji obavlja određeni zadatak
- Preko svog naziva i liste parametara, delu programa iz koga je pozvana vraća odgovarajuće rezultate
- Svaka funkcija ima jedinstveni naziv preko koga se može pozvati proizvoljan broj puta iz bilo kog dela programa u cilju izvršenja tog zadatka.



# FUNKCIJA

---

Python pruža mogućnost definisanja korisničkih funkcija

```
def ime_funkcije (spisak parametara):  
    blok_naredbi  
    return vrednost
```



# FUNKCIJA

---

- Funkcija mora biti definisana pre prvog korišćenja (poziva)
- Da biste pozvali funkciju, upotrebite naziv funkcije, a zatim zagrade

Primer:

```
def moja_funkcija():  
    print ("Moja prva funkcija.")  
    print ("Jednostavno, zar ne?")
```

```
moja_funkcija()    #poziva funkciju
```





# FUNKCIJE SA PARAMETRIMA

---

- Funkcija može biti definisana sa jednim ili više parametara.
- Poziv funkcije mora odgovarati definiciji funkcije po broju i redosledu parametara.
- Parametri navedeni u opisu funkcije nazivaju se formalni parametri.
- Parametri navedeni pri pozivu funkcije nazivaju se stvarni parametri.
- Formalni i stvarni parametri se moraju podudarati po broju i po tipu.
- Pri izvršavanju funkcije, formalnim parametrima se dodeljuju vrednosti stvarnih parametara.



# FUNKCIJE SA PARAMETRIMA

---

Sedeći primer ima funkciju s jednim parametrom (ime). Kad se funkcija pozove, prosleđujemo ime, koje se unutar funkcije koristi za ispis punog imena:

```
def moja_funkcija(ime):  
    print(ime + " Simpson")
```

```
moja_funkcija("Homer")  
moja_funkcija("Marge")  
moja_funkcija("Bart")  
moja_funkcija("Lisa")  
moja_funkcija("Maggie")
```



# PRIMER

---

Šta je rezultat sledećeg koda?

```
def print_hello(n):  
    print('Hello ' * n)  
    print()
```

```
print_hello(3)  
print_hello(5)
```



# LOKALNE PROMENLJIVE FUNKCIJE

---

Promenljive definisane i korišćene unutar funkcije, nisu vidljive van nje

```
def saberi(a,b):  
    c = a + b  
    print (c)
```

```
saberi(3,5)  
print(c)
```

8

Traceback (most recent call last):

File "C:\Users\Tiki\Desktop\primer1.py", line 5, in <module>

print(c)

NameError: name 'c' is not defined



# LOKALNE PROMENLJIVE FUNKCIJE

---

Promenljive definisane i korišćene unutar funkcije, nisu vidljive van nje

```
c = 10
def saberi(a,b):
    c = a + b
    print (c)
saber(3,5)
print(c)
```

Izlaz:

8

10



# VRAĆANJE VREDNOSTI IZ FUNKCIJE

---

Funkcija kao rezultat može da vrati jednu vrednost

```
def obim (a):  
    O = 4*a  
    return O
```

```
print (obim(5))  
print (obim(7))  
print (obim(9))
```



# FUNKCIJE SA VIŠE REZULTATA

---

U nekim situacijama funkcija treba da vrati više vrednosti. Na primer, želimo da izračunamo obim i površinu.

```
def obim_i_povrsina (a):  
    O = 4*a  
    P = a*a  
    return (O,P)
```

```
obim, povrsina = obim_i_povrsina(5)  
print (obim, povrsina)
```



# ZADACI

---

## zadatak\_funkcije\_1

Napišite funkciju pod nazivom pravougaonik koja prihvata dva cela broja m i n kao argumente i štampa  $m \times n$  boks koji se sastoji od zvezdica. Pored je prikazan izlaz funkcije pravougaonik(3,7):

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

## zadatak\_funkcije\_2

Napišite funkcije za sabiranje dva broja i pozovite je u glavnom delu programa. Omogućite izvršavanje te funkcije 3 puta.

Unesite prvi broj: 5

Unesite drugi broj: 66

Zbir unetih brojeva je: 71

-----

Unesite prvi broj: 7

Unesite drugi broj: 8

Zbir unetih brojeva je: 15

-----

Unesite prvi broj: 99

Unesite drugi broj: 2

Zbir unetih brojeva je: 101

-----





**STRINGOVI**

# RAD SA STRINGOVIMA

---

String je niz karaktera ograničen jednostrukim ili dvostrukim navodnicima

```
a="Tekst sa dvosturkim navodnicima"  
b='Tekst sa jednostrukim navodnicima'  
print(a)  
print(b)
```

```
Tekst sa dvosturkim navodnicima  
Tekst sa jednostrukim navodnicima
```



# RAD SA STRINGOVIMA

---

String mora početi i završiti se istim navodnicima

```
s = 'On je rekao, "Zdravo, svete!'"  
t = "Vise navodnika'''' ali i dalje ispravno."  
print(s)  
print(t)
```

Izlaz:

On je rekao, "Zdravo, svete!"

Vise navodnika'''' ali i dalje ispravno.



# RAD SA STRINGOVIMA

---

String se može štampati u više linija

- Ako se započne sa tri navodnika (jednostruka ili dvostruka) i tako i završi
- Ako se u string umetne specijalni karakter \n

```
s1 = """Ovo  
je string koji ima  
više linija."""  
print(s1)  
s2 = 'Ovo\ntakodje'  
print (s2)
```



# RAD SA STRINGOVIMA

---

## Specijalni karakteri

`\n` –završava trenutnu liniju i nastavlja u sledećoj

`\t` –ubacuje tab u string

`\'` –ubacuje ' u string

`\"` –ubacuje " u string

`\\` –ubacuje \ u string



# SPAJANJE STRINGOVA

---

Dva stringa se spajaju u jedan korišćenjem operatora +

```
s1 = "Zdravo"
```

```
s2 = "Svete"
```

```
print(s1+s2)
```

```
print(s1+' '+s2)
```

```
print("Halo," + "gde si?")
```

```
s1 = "Prvi string"
```

```
s2 = ", drugi string"
```

```
print(s1+s2)
```



# SPAJANJE STRINGOVA

---

Više stringova se spajaju u jedan, na isti način, korišćenjem operatora +

```
s0 = "Spajanje"
```

```
s1 = "vise"
```

```
s2 = "stringova"
```

```
s3 = "zajedno"
```

```
razmak= " "
```

```
s = s0 + razmak+ s1 + razmak+ s2 + razmak+ s3
```

```
print (s)
```



# KOPIRANJE STRINGOVA

---

```
s = 'Ha'  
print (s * 10)  
a = 'Ćao'  
print(a * 3)
```

HaHaHaHaHaHaHaHaHaHa

ĆaoĆaoĆao

- Množenje nulom i negativnim brojem ne daje nikakav rezultat
- Nije dozvoljeno množenje razlomljenim brojem i sabiranje sa celim brojem

```
print('Pozdrav' * 8.1)  
print('123' + 4)
```





# DELJENJE STRINGA

---

Delovima stringa može se pristupiti preko indeksa navedenog u zagradama [ ]

```
a="Dobar dan"
```

```
print (a[1])
```

```
print (a[2:7]) # dobijate znakove od pozicije 2 do 7 (7 nije uključen)
```

Negativno indeksiranje

```
a="Dobar dan"
```

```
print (a[-5:-2])
```



# DELJENJE STRINGA

---

Delovima stringa može se pristupiti preko indeksa navedenog u zagradama [ ]

```
s = 'programiranje'
print ('s = ', s)
print ('s[0] = ', s[0], ' s[3] = ', s[3])
print ('s[-1] = ', s[-1])
print ('s[1:5] = ', s[1:5])
print ('s[5:-2] = ', s[5:-2])
```

```
s = programiranje
s[0] = p s[3] = g
s[-1] = e
s[1:5] = rogr
s[5:-2] = amiran
```



# STRING FUNKCIJE

---

Dužina stringa

len() – funkcija len vraća dužinu stringa

```
s = "Zdravo! "  
print (len(s))  
print (len("Jedan obican string "))  
  
print (s[2:-1],s[2:len(s)])
```



---

Python ima niz ugrađenih metoda koje možete koistiti nad stringovima

- Broj ponavljanja stringa u stringu – count
- Konvertovanje svih slova u velika, odnosno mala – upper, lower
- Promena dela stringa novim stringom –replace

```
s = ' Python je zanimljiv programski jezik! '  
print (s.count(' '))  
print (s.upper())  
print (s.lower())  
print (s.replace('a', 'A'))
```



# FORMAT

---

format() - metoda uzima argumente i stavlja ih u string gde su rezervisana mesta {}

```
godine = 25
```

```
tekst = "Perica ima {} godina."
```

```
print(tekst.format(godine))
```

Metoda format () uzima neograničen broj argumenata i stavlja ih u string na odgovarajuća mesta:

```
mat = 5
```

```
inf= 5
```

```
fiz= 4
```

```
poruka="Perica ima sledeće ocene iz matematike {}, informatike {} i fizike {}."
```

```
print(poruka.format(mat,inf,fiz))
```



# STRINGOVI SU NEPROMENLJIVI

---

Posmatrajmo sledeći deo koda (testirati):

```
rec = "Osnovi programiranja"  
rec[0] = 'A'  
print(rec)
```

Predloženo rešenje –formiranje nove reči na osnovu stare:

```
nova_rec = 'A'+rec[1:]  
print(nova_rec)
```



# OPERATOR IN I IN NOT

---

Operator **in** je koristan kada treba da znamo da li string sadrži nešto što tražimo.

Na primer:

```
tekst = "Čini mi se da Python nije težak"  
x = " Python" in tekst  
print (x)
```

Možete da kombinujete operator **in** sa operatorom **not** da zaključite da string ne sadrži nešto:

```
tekst = "Čini mi se da Python nije težak"  
x = "program" not in tekst  
print (x)
```



# ZADACI

---

## zadatak\_stringovi\_1

Napisati program kojim se za dati jedinstveni matični broj građanina određuju dan, mesec i godina rođenja i ispisuju u formatu DD.MM.GGGG. (Smatrati da je taj građanin rođen u periodu od 1100. do 2020. godine.)



LISTE

# LISTE

---

- Lista u jeziku Python najjednostavnije se definiše kao niz vrednosti u uglastim zagradama.
- Može da sadrži podatke proizvoljnog tipa
- Svaki podatak u listi se naziva element

`[1, 4, 17, 256, 3, 59, 45]` – lista celih brojeva

`['april', 'jun', 'septembar', 'novembar']` – lista stringova

`[]` – prazna lista

`['Pera', 28, 2020, 'VPTS']` – lista elemenata različitog tipa

`[1, 4, [7, 6, [13]], [9, 5]]` – lista čiji elementi su liste



# GENERISANJE LISTE

---

- Primer 1:

Izlaz:

```
zivotinje = ['pas', 'macka', 'mis', 'majmun', 'slon']  
print (zivotinje)
```

```
['pas', 'macka', 'mis', 'majmun', 'slon']
```

- Primer 2:

```
lista=list(range(1,15))  
print (lista)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
```

- Primer 3:

```
lista=list(range(1,20,2))  
print (lista)
```

```
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19]
```



# ELEMENTI LISTE, INDEKSIRANJE

- Elementima liste se pristupa preko rednog broja pozicije – indeksa
- Indeks prvog elementa u listi je 0

Primer:

```
lista= ['Ejda', 2, 'Tjuring', 4,  
        'Bebidz', 6, 'Bul']
```

```
print (lista[2], lista[-1])
```

```
print (lista[1:5])
```

```
print (lista[2:])
```

```
print (lista[:5])
```

```
print (lista[12])
```

Tjuring Bul

[2, 'Tjuring', 4, 'Bebidz']

['Tjuring', 4, 'Bebidz', 6, 'Bul']

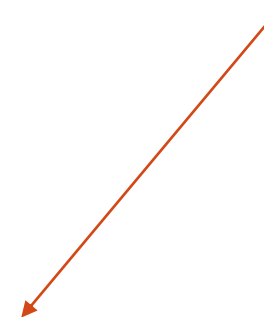
['Ejda', 2, 'Tjuring', 4, 'Bebidz']

Traceback (most recent call last):

File "C:/Users/Tiki/Desktop/primer2.py", line 6, in <module>  
 print (lista[12])

IndexError: list index out of range

Izlaz



# PROMENA VREDNOSTI

---

Elementima liste je moguće promeniti vrednost

Primer:

```
lista= [2, 3, 6, 10]  
lista[1] = 4  
lista[-1] = 8  
print (lista)
```

Izlaz: [2, 4, 6, 8]



# SPAJANJE LISTI

---

Dve liste se spajaju operatorom +

Primer:

```
lista1 = ["a", "b" , "c"]  
lista2 = [1, 2, 3]  
lista3 = lista1 + lista2  
print(lista3)
```

Izlaz: ['a', 'b', 'c', 1, 2, 3]



# PONAVLJANJE ELEMENATA LISTE

---

Elementi liste se mogu ponavljati određen broj puta operatorom \*

```
C = ['Ćao']
```

```
A = [1, 2, 3]
```

```
B = 3 * A
```

```
D = 4 * C
```

```
print(B, D)
```

Izlaz:

```
[1, 2, 3, 1, 2, 3, 1, 2, 3] ['Ćao', 'Ćao', 'Ćao', 'Ćao']
```



# FUNKCIJE ZA RAD SA LISTAMA

---

Dužina liste, odnosno broj elemenata liste – **len**

Primer:

```
samoglasnici = ['a', 'e', 'i', 'o', 'u']  
print (len(samoglasnici))
```





# FUNKCIJE ZA RAD SA LISTAMA

---

Funkcija	Opis
len	Vraća dužinu liste (broj elemenata liste)
sum	Vraća sumu elemenata u listi
min	Vraća najmanji element u listi
max	Vraća najveći element u listi



# FUNKCIJE ZA RAD SA LISTAMA

---

Operator in

Primer:

```
lista=[1,2,3,4,5]
print(5 in lista)
print(8 in lista)
if (not 10 in lista):
    print(10,"nije u listi")
else:
    print(10,"jeste u listi")
```

Izlaz:

```
True
False
10 nije u listi
```



# METODE ZA RAD SA LISTAMA

---

Python ima skup ugrađenih metoda koje možete koristiti na listama.

Dodavanje elemenata na kraj liste –append

Primer:

```
lista = ["jabuka", "kruska", "banana"]  
lista.append("kivi")  
print(lista)
```

Izlaz:

```
['jabuka', 'kruska', 'banana', 'kivi']
```



# METODE ZA RAD SA LISTAMA

---

Dodavanje elemenata na željenu poziciju –insert

Primer:

```
lista = ["jabuka", "kruska", "banana"]  
lista.insert(1, "kivi")  
print(lista)
```

Izlaz:

```
['jabuka', 'kivi', 'kruska', 'banana']
```



# METODE ZA RAD SA LISTAMA

---

Brisanje elementa iz liste –del, pop, remove

Primer:

```
lista = ["jabuka", "kruska", "banana"]  
lista.remove("kruska")  
print(lista)
```

Izlaz:

```
['jabuka', 'banana']
```



# METODE ZA RAD SA LISTAMA

---

Primer:

```
lista = [1, 2, 3, 4, 5, 6, 7]
lista.pop()
print(lista)
lista.pop(2)
print(lista)
del lista[0]
print (lista)
```

Izlaz:

```
[1, 2, 3, 4, 5, 6]
[1, 2, 4, 5, 6]
[2, 4, 5, 6]
```



# METODE ZA RAD SA LISTAMA

---

clear() – prazni listu

```
lista = ["jabuka", "kruska", "banana"]  
lista.clear()  
print(lista)
```



# METODE ZA RAD SA LISTAMA

---

Broj ponavljanja elementa –count

Primer:

```
L = [2, 3, 4, 1, 7, 2, 3, 1, 1, 0, 9, 1]
```

```
print (L.count(1))
```

```
formula1= ['Hamilton', 'Bottas', 'Vettel', 'Hamilton', 'Albon']
```

```
print (formula1.count('Hamilton'))
```

Izlaz:

```
4  
2
```





# METODE ZA RAD SA LISTAMA

---

Sortiranje i okretanje elemenata liste –sort, reverse

Primer:

```
A = [2,6,1,9,3,5,4]
```

```
A.sort()
```

```
print (A)
```

```
A=[1, 2, 3, 4, 5, 6, 9]
```

```
A.reverse()
```

```
print (A)
```

```
A = ['Pera', 'Laza', 'Mika', 'Aca']
```

```
A.sort()
```

```
print (A)
```

Izlaz:

```
[1, 2, 3, 4, 5, 6, 9]
```

```
[9, 6, 5, 4, 3, 2, 1]
```

```
['Aca', 'Laza', 'Mika', 'Pera']
```



# LISTE I PETLJE

---

Primer:

```
voce = ["banana", "jabuka", "kruska"]  
for x in voce:  
    print (x)
```

Izlaz:

```
banana  
jabuka  
kruska
```



# LISTE I STRINGOVI

---

Koliko data rečenica ima reči?

```
recenica= "Volim da učim programiranje"  
lista_reci= list(recenica.split())  
print(lista_reci)  
print(len(lista_reci))
```

Izlaz: `['Volim', 'da', 'učim', 'programiranje']`  
`4`

- `recenica.split()` – od date rečenice pravi listu čiji su elementi dobijeni razdvajanjem rečenice po jednom (ili više) blanko karakteru
- Funkcija `split()` se može primeniti direktno na ulaz:
- `lista_reci= list(input().split())`



# LISTE I STRINGOVI

---

Metoda join() poziva se nad listom stringova, formirajući jedan string od elemenata liste

```
L = ['Volim', 'da', 'učim', 'Python']  
recenica=' '.join(L)  
print(recenica)
```



# ZADACI

---

## zadatak\_liste\_1

Napisati funkciju koja datu listu sortira u rastućem poretku. U glavnom delu programa uneti listu celih brojeva (korisnik unosi brojeve sa tastature sve dok ne unese broj 0), a zatim korišćenjem napisane funkcije sortirati elemente liste.

zadatak\_kviz\_2  
zadatak\_kviz\_2\_liste } Kviz

Rešenje (zadatak\_kviz):

<https://youtu.be/wcDEYEg7o4Y>

