

Uvod u Python

Sledeći kod je većinski preuzet od asistenata Anđelke Zečević i Milana Čugurovića sa časova vežbi iz Mašinskog učenja.

Ispis podataka

```
In [12]: print(10)

10
```

```
In [2]: print("NISKA")

NISKA
```

```
In [3]: a = 10
print(a)

10
```

Formatiran ispis

```
In [4]: n = 10
print('Vrednost promenljive n je {}'.format(n))

Vrednost promenljive n je 10
```

```
In [5]: a = 5
b = 6
print('a = {}, b = {}'.format(a, b))

a = 5, b = 6
```

If-else, for, while, obrada izuzetaka

```
In [13]: try:
    a = 10
    if a % 3 == 0:
        print('Broj a je deljiv brojem 3')
    elif a % 3 == 1:
        print('Broj a daje ostatak 1 pri deljenju sa brojem 3')
    else:
        print('Broj a daje ostatak 2 pri deljenju sa brojem 3')
except:
    print('Neispravna vrednost broja a!')
```

Broj a daje ostatak 1 pri deljenju sa brojem 3

```
In [14]: godina = 2000
if (godina % 4 == 0 and godina % 100 != 0) or (godina % 400 == 0):
    print('Godina {} je prestupna'.format(godina))
else:
    print('Godina {} je prosta'.format(godina))

Godina 2000 je prestupna
```

```
In [15]: for i in range(0,5): # C-ovski: for(i = 0; i < 5; i++)
    print(i)

0
1
2
3
4
```

```
In [16]: i = 0
while i < 5:
    print(i)
    i += 1 # operator ++ nije podrzan

0
1
2
3
4
```

Liste

```
In [17]: lista = [1, 2, 3, 4]
print(lista)

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

```
In [18]: lista = [1, 2, 3, 4]
n = len(lista)
print('Duzina liste je: {}'.format(n))

Duzina liste je: 4
```

Iteracija kroz listu po indeksima

```
In [19]: lista = [1, 2, 3, 4]
n = len(lista)

for i in range(0, n):
    print(lista[i])

1
2
3
4
```

Iteracija kroz listu po elementima (foreach petlja)

```
In [20]: lista = [1, 2, 3, 4]
n = len(lista)

for element in lista:
    print(element)

1
2
3
4
```

List comprehension

```
In [21]: lista = [1, 5, 2, 4, 3]
kvadrirana = [x**2 for x in lista] # ** - stepen
print(kvadrirana)

[1, 25, 4, 16, 9]
```

```
In [22]: lista = [1, 2, 3, 4, 5]
neparni = [x for x in lista if x % 2 == 1]
print(neparni)

[1, 3, 5]
```

Zipovanje lista

```
In [23]: lista1 = [1, 2, 3, 4]
lista2 = ['a', 'b', 'c', 'd']
zipovana = zip(lista1, lista2)
print(zipovana)

for el in zipovana:
    print(el)
```

<zip object at 0x0000021C0E99D240>

(1, 'a')

(2, 'b')

(3, 'c')

(4, 'd')

Obrtanje liste

```
In [24]: lista = [1, 2, 3, 4, 5]
lista.reverse()
print(lista)

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

Sortiranje liste

```
In [25]: lista = [1, 5, 2, 4, 3]
lista.sort()
print(lista)

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

```
In [26]: lista = [1, 5, 2, 4, 3]
lista.sort(reverse=True)
print(lista)

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

Izdvajanje podlisti

[polazni_indeks;zavrzni_index(n+1):korak]

```
In [27]: lista = [1, 2, 3, 4]
print(lista[1:3])

[2, 3]
```

```
In [28]: lista = [1, 2, 3, 4]
print(lista[1:4:2])

[2, 4]
```

```
In [29]: lista = [1, 2, 3, 4]
print(lista[-4:-1])

[1, 2, 3]
```

Dodavanje elemenata u listu i konkatencija listi

```
In [37]: lista = [1, 2, 3, 4]
lista.append(5) # Dodavanje elementa na kraj liste
print(lista)

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

```
In [38]: lista = [1, 2, 3, 4]
lista.insert(3,3.4) # Dodavanje vrednosti 3.4 na poziciju u listi sa indeksom 3
print(lista)

[1, 2, 3, 3.4, 4]
```

```
In [39]: lista = [1, 2, 3, 4]
lista.pop() # Uklanjanje elementa sa kraja liste
print(lista)

[1, 2, 3]
```

```
In [40]: lista1 = [1, 2, 3, 4]
lista2 = [5, 6, 7, 8]
print(lista1 + lista2) # Konkatencija listi

[1, 2, 3, 4, 5, 6, 7, 8]
```

```
In [41]: element = 0
lista = [1, 2, 3, 4]
print([element] + lista) # Dodavanje elementa na pocetak liste pomocu konkatencije

[0, 1, 2, 3, 4]
```

Pseudoslučajni brojevi

```
In [34]: import random
x = random.random()
print(x)

0.8674554092476019
```

```
In [35]: lista = [10, 100, 1000, 10000]
izabrani = random.choice(lista)
print(izabrani)

10000
```

```
In [36]: lista = [10, 100, 1000, 10000]
podskup = random.sample(lista, 3)
print(podskup)

[10, 100, 10000]
```

Funkcije

```
In [31]: def saberi(a, b, c=0):
    return a + b + c

saber(1,2,3)
```

```
Out[31]: 6
```

```
In [32]: # Funkcija moze vratiti vise argumenata kroz strukturu tuple
def minimum_i_maksimum(a, b):
    minimum = min(a,b)
    maksimum = max(a,b)
    return (minimum, maksimum)

a = 10
b = 6.7

(mini, maks) = minimum_i_maksimum(a, b)
print(mini, maks)
```

```
6.7 10
```

```
In [33]: # Funkcija moze vratiti niz
def niz_neparnih():
    return [1, 3, 5, 7, 9, 11, 13, 15, 17, 19]

svi_neparni = niz_neparnih()
[prvi, drugi, treci] = niz_neparnih()[:3]

print(svi_neparni)
print(prvi, drugi, treci)
```

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