

Python (#3)

Nemanja Maček

- “Kraljevski pitoni nisu baš socijalna bića, ali ukoliko se navikne na svog vlasnika, njima se može manipulirati bez većih poteškoća. Potrebno je naglasiti da mogu biti i agresivni ukoliko smatraju da je ugrožena njihova teritorija i to pogotovo za vreme presvlačenja.” [1]



Slika preuzeta sa: <https://www.xyzreptiles.com/product/baby-super-mystic-ball-python/>

- Stringovi
- Liste

Dodela vrednosti.

```
>>> s = "Python string"
>>> print(s)
Python string
>>> # upotreba jednostrukih i običnih navodnika
>>> s = "A Poor Woman's Journey."
>>> print(s)
A Poor Woman's Journey.
>>> s = "I read the article, 'A Poor Woman's Journey.'"
>>> print(s)
I read the article, 'A Poor Woman's Journey.'
>>> s = 'dir "c:\&temp\*.sas" /o:n /b > "c:\&temp\abc.txt"'
>>> print(s)
dir "c:\&temp\*.sas" /o:n /b > "c:\&tempbc.txt"
```

Dodela vrednosti.

```
>>> # višestruke linije
>>> s = """jQuery exercises
        JavaScript tutorial
        Python tutorial and exercises ..."""
>>> print(s)
jQuery exercises
        JavaScript tutorial
        Python tutorial and exercises ...
>>> s = 'jQuery exercises\n JavaScript tutorial\n Python tutorial and exercises
... '
>>> print(s)
jQuery exercises
        JavaScript tutorial
        Python tutorial and exercises
```

Pristupanje karakteru u stringu.

```
>>> a = "Python string"
>>> print (a)
Python string
>>> b = a[2]
>>> print(b)
t
>>> a[0]
'p'
```

Pristupanje karakteru u stringu

- Upotreba izraza u indeksu.

```
>>> a = "Python string"
```

```
>>> b = a[4+3]
```

```
>>> print(b)
```

```
s
```

Pristupanje karakteru u stringu

- Indeks mora biti celobrojna vrednost.

```
>>> a = "Python string"
>>> a[2.3]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: string indices must be integers
```


Pristupanje karakteru u stringu

- Negativni indeksi.

```
>>> a = "Python string"
```

```
>>> a[-2]
```

```
'n'
```

```
>>> a[-8]
```

```
'n'
```

Elementi stringa se ne mogu menjati.

```
>>> a = "PYTHON"
>>> a[0] = "x"
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' object does not support item assignment
>>> a = "PYTHON"
# Konkatenacijom se može zameniti neka vrednost na sledeći način:
>>> b = "X" + a[1:]
>>> print(b)
XYTHON
# Pri tome se ne menja originalni string.
>>> print(a)
PYTHON
```

Konkatenacija stringova.

```
>>> a = "Python" + "String"
>>> print(a)
PythonString
# možete da koristite i operator +=
>>> a = "Java"
>>> b = "Script"
>>> a+=b
>>> print(a)
JavaScript
```

Upotreba operatora *.

```
>>> a = "Python" + "String"  
>>> b = "<" + a*3 + ">"  
>>> print(b)  
<PythonStringPythonStringPythonString>
```

Dužina stringova.

- Funkcija `len()` vraća broj karaktera u stringu.
- Obratite pažnju da brojanje kreće od nule, tako da ukoliko `len()` vrati `n` za string `x`, poslednji karakter stringa `x` očitavate kao `x[n-1]`.

```
>>> a = "Python string"
>>> len(a)
13
>>> a[13]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: string index out of range
>>> a[12]
'g'
```

Prolaz kroz string petljom while.

```
a = "STRING"  
i = 0  
while i < len(a):  
    c = a[i]  
    print(c)  
    i = i + 1
```

- Šta je izlaz ovog programa?

Prolaz kroz string petljom while.

```
a = "STRING"  
i = 0  
while i < len(a):  
    c = a[i]  
    print(c)  
    i = i + 1
```

- Šta je izlaz ovog programa?

S
T
R
I
N
G

Prolaz kroz string petljom while.

```
a = "STRING"  
i = 0  
while i <= len(a):  
    c = a[i]  
    print(c)  
    i = i + 1
```

- Da li je ovakav program ispravan?

Prolaz kroz string petljom while.

```
a = "STRING"  
i = 0  
while i <= len(a):  
    c = a[i]  
    print(c)  
    i = i + 1
```

- Da li je ovakav program ispravan?
- Nije, vratiće grešku prilikom pokušaja očitavanja elementa a[len(a)].

Prolaz kroz string petljom for.

```
a = "Python"
i = 0
new=""
for i in range (0,len(a)):
    b=a[i]
    # podsetnik: operator + se koristi za konkatenciju
    new = new+b
    i = i + 1
    print(b)
    print(new)
```

- Šta je izlaz ovog programa?

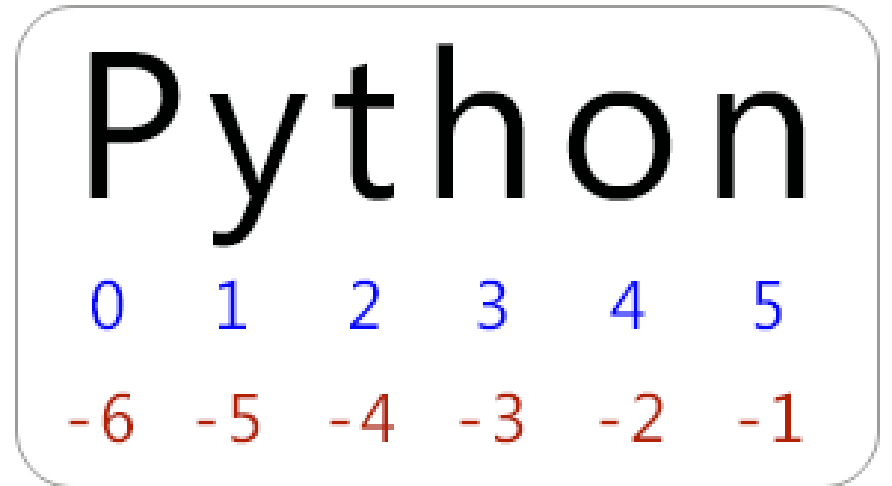
Prolaz kroz string petljom for.

- Izlaz je:

```
P
 P
y
 Py
t
 Pyt
h
 Pyth
o
 Pytho
n
 Python
```

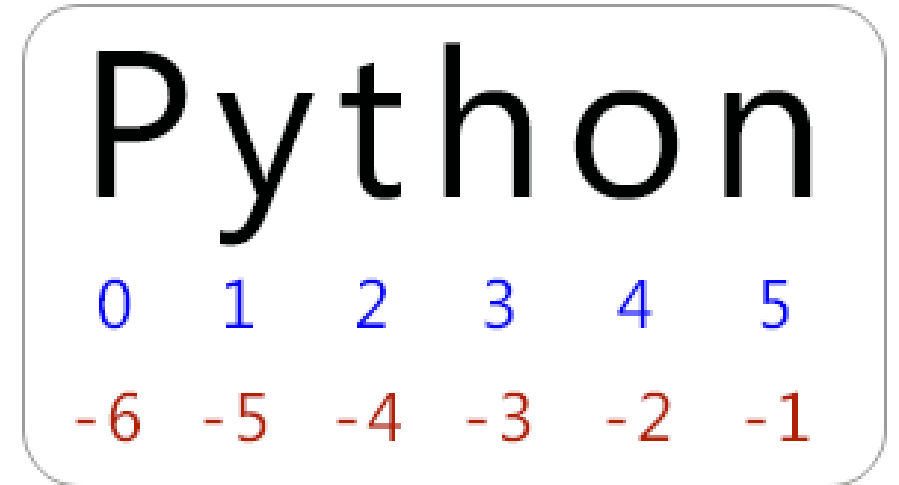
Seckanje stringova.

- `s[1:4]` je 'yth'
 - Kreće od prvog, ali ne uključuje peti znak
- `s[1:]` je 'ython'
 - Ukoliko jedan indeks ne navedete, kao podrazumevana vrednost uzima se početak, odnosno kraj stringa.
- `s[:]` je 'Python'
 - Ceo string.
- `s[1:100]` je 'ython'
 - Preveliki indeks se skraćuje na dužinu stringa.



Seckanje stringova.

- `s[-1]` je 'n'
 - Poslednji karakter, tj. prvi s kraja.
- `s[-4]` je 't'
 - Četvrti karakter sa kraja.
- `s[:-3]` is 'pyt'
 - Svi karakteri do trećeg s kraja (koji nije uključen u podstring).
- `s[-3:]` is 'hon'
 - Svi karakteri počev od trećeg s kraja (koji je uključen u podstring).



Seckanje stringova.

Primeri.

```
>>> a = "Python String"
>>> print (a[0:5])
Pytho
>>> print(a[6:11])
Stri
>>> print(a[5:13])
n String
>>> print(a[:8])
Python S
>>> print(a[4:])
on String
>>> print(a[6:3])
```

Traženje karaktera u stringu.

```
def search(char, str):
    L=len(str)
    print(L)
    i = 0
    while i < L:
        if str[i]== char:
            return 1
        i = i + 1
    return -1
print(search("P", "PYTHON"))
```

- Šta je izlaz ovog programa?

Traženje karaktera u stringu.

```
def search(char, str):  
    L=len(str)  
    print(L)  
    i = 0  
    while i < L:  
        if str[i]== char:  
            return 1  
            i = i + 1  
    return -1  
print(search("P", "PYTHON"))
```

- Šta je izlaz ovog programa?

6
1

Traženje karaktera u stringu.

```
def search(char, str):
    L=len(str)
    print(L)
    i = 0
    while i < L:
        if str[i]== char:
            return 1
        i = i + 1
    return -1
print(search("S", "PYTHON"))
```

- Šta je izlaz ovog programa?

Traženje karaktera u stringu.

```
def search(char, str):  
    L=len(str)  
    print(L)  
    i = 0  
    while i < L:  
        if str[i]== char:  
            return 1  
            i = i + 1  
    return -1  
print(search("S", "PYTHON"))
```

- Šta je izlaz ovog programa?

6
-1

Formatiranje stringova – poravnanje.

```
# poravnanje udesno
>>> '{:>15}'.format('Python')
'          Python'
# korišćenje znaka * za dopunu
>>> '{:*<15}'.format('Python')
'Python*****'
# poravnanje ulevo
>>> '{:15}'.format('Python')
'Python          '
# centriranje
>>> '{:^16}'.format('Python')
'      Python      '
```

Formatiranje stringova – skraćivanje.

```
>>> '{:.10}'.format('Python Tutorial')
'Python Tut'
# upotreba argumenta
>>> '{:.{}}'.format('Python Tutorial', 10)
'Python Tut'
```

Primeri.

```
>>> my_list1 = [5, 12, 13, 14] # lista celobrojnih vrednosti
>>> print(my_list1)
[5, 12, 13, 14]
>>> my_list2 = ['red', 'blue', 'black', 'white'] # lista stringova
>>> print(my_list2)
['red', 'blue', 'black', 'white']
>>> my_list3 = ['red', 12, 112.12] # lista sadži string, ceo broj i float
>>> print(my_list3)
['red', 12, 112.12]
>>> my_list=[] # prazna lista
>>> print(my_list)
[]
```

Konkatenacija listi (operator +).

```
>>> color_list1 = ["White", "Yellow"]
>>> color_list2 = ["Red", "Blue"]
>>> color_list3 = ["Green", "Black"]
>>> color_list = color_list1 + color_list2 + color_list3
>>> print(color_list)
['White', 'Yellow', 'Red', 'Blue', 'Green', 'Black']
```

Ponavljjanje listi (operator *).

```
>>> number = [1,2,3]
>>> print(number[0]*4)
4
>>> print(number*4)
[1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3]
```

Indeksi.

```
>>> color_list=["RED", "Blue", "Green", "Black"]
```

Item	RED	Blue	Green	Black
Index (s leva)	0	1	2	3
Index (s desna)	-4	-3	-2	-1

Indeksi.

```
>>> color_list = ["RED", "Blue", "Green", "Black"]
>>> color_list[0]
'RED'
>>> print (color_list[0],color_list[3])
RED Black
>>> color_list[-1]
'Black'
>>> color_list[4]
Traceback (most recent call last):
  File "<pyshell#94>", line 1, in <module>
    color_list[4]
IndexError: list index out of range
```

Dodavanje elementa na kraj liste.

- Koristi se funkcija `append()`.

```
>>> color_list=["Red", "Blue", "Green", "Black"]
>>> print(color_list)
['Red', 'Blue', 'Green', 'Black']
>>> color_list.append("Yellow")
>>> print(color_list)
['Red', 'Blue', 'Green', 'Black', 'Yellow']
```

Umetanje elementa na određenu poziciju u listi.

- Koristi se funkcija insert().

```
>>> color_list=["Red", "Blue", "Green", "Black"]
>>> print(color_list)
['Red', 'Blue', 'Green', 'Black']
>>> color_list.insert(2, "White") # Umetni element na treću poziciju u listu.
>>> print(color_list)
['Red', 'Blue', 'White', 'Green', 'Black']
```

Izmena elementa liste.

- Elementu se pristupa preko indeksa.

```
>>> color_list=["Red", "Blue", "Green", "Black"]
>>> print(color_list)
['Red', 'Blue', 'Green', 'Black']
>>> color_list[2]="Yellow" # Izmeni treći element
>>> print(color_list)
['Red', 'Blue', 'Yellow', 'Black']
```

Uklanjanje svih elemenata liste.

- Koristi se funkcija `clear()`.

```
>>> color_list=["Red", "Blue", "Green", "Black"]
>>> print(color_list)
['Red', 'Blue', 'Green', 'Black']
>>> color_list.clear()
>>> print(color_list)
[]
```

Seckanje liste.

- Liste se seckaju preko indeksa.

```
>>> color_list=["Red", "Blue", "Green", "Black"]
# Lista ima četiri elementa
# Indeksi počinju sa nulom i završavaju se sa 3.
>>> print(color_list[0:2]) # Iseckaćemo prva dva elementa
['Red', 'Blue']
```

Seckanje liste.

- Liste se seckaju preko indeksa.

```
>>> color_list=["Red", "Blue", "Green", "Black"]
# Lista ima četiri elementa
# Indeksi počinju sa nulom i završavaju se sa 3.
>>> print(color_list[0:2]) # Prva dva elementa
['Red', 'Blue']
>>> print(color_list[1:2]) # Drugi element
['Blue']
>>> print(color_list[1:-1]) # Drugi i treći element (drugi indeks je negativan)
['Blue', 'Green']
>>> print(color_list[:3]) # Prva tri elementa
['Red', 'Blue', 'Green']
```

Čitanje i uklanjanje elementa sa date pozicije.

- Koristi se funkcija pop().

```
>>> color_list=["Red", "Blue", "Green", "Black"]
>>> print(color_list)
['Red', 'Blue', 'Green', 'Black']
>>> color_list.pop(2) # Uklanja drugi element i vraća ga
'Green'
>>> print(color_list)
['Red', 'Blue', 'Black']
```


Vraćanje indeksa elementa koji ima datu vrednost.

- Koristi se funkcija `index()`.

```
>>> color_list=["Red", "Blue", "Green", "Black"]
```

```
>>> color_list.index("Red")
```

```
0
```

```
>>> color_list.index("Black")
```

```
3
```

Sortiranje elemenata liste.

- Koristi se funkcija `sort()`.

```
>>> color_list=["Red", "Blue", "Green", "Black"]
>>> print(color_list)
['Red', 'Blue', 'Green', 'Black']
>>> color_list.sort(key=None, reverse=False)
>>> print(color_list)
['Black', 'Blue', 'Green', 'Red']
```

Kreiranje liste sa inverznim redosledom elemenata.

- Koristi se funkcija `reverse()`.

```
>>> color_list=["Red", "Blue", "Green", "Black"]
>>> print(color_list)
['Red', 'Blue', 'Green', 'Black']
>>> color_list.reverse()
>>> print(color_list)
['Black', 'Green', 'Blue', 'Red']
```

Čemu služi [: :]?

```
>>> listx=[1, 5, 7, 3, 2, 4, 6]
>>> print(listx)
[1, 5, 7, 3, 2, 4, 6]
>>> sublist=listx[2:7:2] # list[start:stop:step], step je inkrement.
>>> print(sublist)
[7, 2, 6]
>>> sublist=listx[::3] # vraća svaki treći element.
>>> print(sublist)
[1, 3, 6]
>>> sublist=listx[6:2:-2] # ako je inkrement negativan, ide se unazad kroz listu
>>> print(sublist)
[6, 2]
```

Pronalaženje najvećeg i najmanjeg elementa liste.

- Koriste se funkcije `min()` i `max()`.

```
>>> listx=[5, 10, 3, 25, 7, 4, 15]
```

```
>>> print(max(listx))
```

```
25
```

```
>>> print(min(listx))
```

```
3
```

Pitanja su dobrodošla.

1. <http://moj.pet-centar.rs/Teraristika/5404-Kraljevski-piton.html>
2. Milan Bjelica (2016): Programski jezik Python. Dostupno u sekciji “download” na sledećem linku: http://www.etf.bg.ac.rs/etf_files/udzbenici/python.pdf
3. Charles R. Severance (2016): Python for Everybody – Exploring Data Using Python 3. Dostupno u sekciji “download” na stranici predmata i na sledećem linku: http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf
4. A Bite of Python. Dostupno u sekciji “download” na sledećem linku (na engleskom jeziku): <https://python.swaroopch.com/>.
5. <https://www.w3resource.com/python/python-tutorial.php>