

Uvod u programiranje



2023/24



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)
```

```
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
vise linija."""
print(s1)
s2 = 'Ovo\ntakodje'
print (s2)
```

```
Ovo
je string koji ima
vise linija.
Ovo
takodje
```

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

Konkatenacija stringova

- Dva stringa se spajaju u jedan korišćenjem operatora +

```
s0 = "Zdravo"  
s1 = "Svete"  
print(s0+s1)  
print(s0+' '+s1)  
print("Halo, " + "gde si?")  
s1 = "Prvi string"  
s2 = ", drugi string"  
print(s1+s2)
```

```
ZdravoSvete  
Zdravo Svete  
Halo, gde si?  
Prvi string, drugi string
```

Konkatenacija 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)
```

Spajanje vise stringova zajedno

Kopiranje stringova

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

HaHaHaHaHaHaHaHaHaHa
ProgramProgramProgram

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

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

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`

```
s = "Hello!"  
print (len(s))  
print (len("Jedan obican string "))
```

6

20

```
print (s[2:-1],s[2:len(s)])
```

llo llo!

String funkcije

- String u stringu – **find, index**

```
string.find(pattern[,begin[,end]])
```

```
str1 = "Ovo je reprezentacija nekog stringa!"  
str2 = "taci"  
print (str1.find(str2))           15  
print (str1.find(str2, 10))      15  
print (str1.find(str2, 16))      -1  
print (str1.find(str2, 10, 20))  15  
print (str1.find(str2, 10, 16))  -1
```

String funkcije

- String u stringu – **find, index**

```
string.index(pattern[,begin[,end]])
```

```
str1 = "Ovo je reprezentacija nekog stringa!"  
str2 = "taci"  
print (str1.index(str2))  
print (str1.index(str2, 16))
```

15

ValueError: substring not found

String funkcije

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

```
s = 'Ovaj predmet se zove PRAKTIKUM'
```

```
print (s.count(' '))           6
print (s.upper())             OVAJ PREDMET SE ZOVE PRAKTIKUM
print (s.lower())            ovaj predmet se zove praktikum
print (s.replace('a', 'A'))   OvAj predmet se zove PrAktikum
```

Stringovi su nepromenljivi

- Posmatrajmo sledeći deo koda:

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

- Umesto očekivanog izlaza:
Asnovi programiranja
- Izlaz je runtime error:
TypeError: 'str' object does not support item assignment

- Predloženo rešenje – formiranje nove reči na osnovu stare:

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

Poređenje stringova

```
ime = input()
```

```
if ime=="Marko":  
    print("Da, moje ime je Marko")
```

ili svi slučajevi:

```
if ime < "Marko":  
    print("Uneto ime je ispred Marka")  
elif ime > "Marko":  
    print("Uneto ime je iza imena Marko")  
else:  
    print("Uneto ime je Marko")
```


Primer

- Šta je rezultat rada sledećeg programa?

```
prvi_sabirak = input("Unesi prvi sabirak: ")  
drugi_sabirak = input("Unesi drugi sabirak: ")  
zbir = prvi_sabirak + drugi_sabirak  
print(zbir)
```

Primer

- Tekst je palindrom ako se čita sdesna nalevo isto kao i sleva nadesno (radar, kajak). Napisati funkciju koja testira da li je neki tekst palindrom. Problem rešiti kako iterativno, tako i rekurzivno.

```
def palindromI(tekst):  
    n = len(tekst)  
    limit = n//2  
    for i in range(limit):  
        if tekst[i] != tekst[-i-1]:  
            return False  
    return True
```

Primer

```
def palindromR(tekst):  
    if tekst == '':  
        return True  
    elif tekst[0] != tekst[-1]:  
        return False  
    else:  
        return palindromR(tekst[1 : -1])
```

```
tekst = input()  
if (palindromI(tekst)): # moze se koristiti i palindromR(tekst)  
    print("Jeste")  
else:  
    print("Nije")
```