

# Ponavljanje nastavnih sadržaja iz informatike

---

TJEDAN PONAVLJANJA I PLANIRANJA



# Ponavljanje nastavnih sadržaja iz informatike

---

Dragi učenici 7. razreda, ovaj tjedan ćete ponavljati nastavne sadržaje (nastavno gradivo) iz informatike.

Ono što ćete ponavljati, i što ću ocjenjivati, je nastavna tema **4.B. Programski jezik Python**.

Spomenuta nastavna tema započinje na stranici 75 u udžbeniku i završava na stranici 91.

Pročitajte tekst u udžbeniku koji se odnosi na spomenutu nastavnu temu. Na sljedećim stranicama ovog dokumenta ću vam napisati što trebate naučiti i na što trebate obratiti posebnu pažnju.

# Što trebate znati

---

1. Trebate znati koji su to jednostavni tipovi podataka u Pythonu. To su **cijeli brojevi**, **decimalni brojevi**, **znakovni nizovi (tekst)** i **logički tipovi podataka**.
2. Trebate znati da se znakovni niz podataka pretvara u cijeli broj pomoću funkcije **int**, zatim da se znakovni niz pretvara u decimalni broj pomoću funkcije **float** te da se različiti brojevi pretvaraju u znakovni niz pomoću funkcije **str**.

Primjeri:

*Pretvorba teksta u cijeli broj:*

```
a = '8'  
a = int(a)
```

*Pretvorba teksta u decimalni broj:*

```
a = '1.2'  
a = float(a)
```

*Pretvorba cijelog broja u tekst:*

```
a = 12  
a = str(a)
```

# Što trebate znati

---

3. Trebate znati da svaki znak u znakovnom nizu (tekstu) ima svoj redni broj. Ali imajte na umu da redni brojevi počinju od 0, a ne od 1. Dakle prvi znak u znakovnom nizu ima redni broj 0.
4. Trebate znati ispisati pojedine znakove iz znakovnog niza, ili pojedine dijelove znakovnog niza, a sve na temelju rednih brojeva znakova.

## Primjeri

```
a = 'VATRA'  
print(a[0]) → ispisuje slovo 'V'  
print(a[3]) → ispisuje slovo 'R'  
print(a[1:4]) → ispisuje slova 'ATR' (od rednog broja 1 do rednog broja 4, ali ne uključuje znak na rednom broju 4, dakle ide zapravo do 3! Ovo ZAPAMTITE.)
```

# Što trebate znati

---

5. Trebate znati da funkcija **len** vraća koliko znakova ima u nekom znakovnom nizu:

```
a = 'VATRA'  
print(len(a)) → ispisat će 5, jer VATRA ima 5 znakova
```

6. Trebate znati da funkcija **min** vraća slovo koje je prvo po abecedi, a funkcija **max** vraća slovo koje je zadnje po abecedi:

```
a = 'VATRA'  
print(min(a)) → ispisat će slovo 'A'  
print(max(a)) → ispisat će slovo 'V'
```

# Što trebate znati

---

7. Trebate znati da je **lista** zasad jedini *složeni* tip podataka kojeg smo učili u Pythonu. Ona u sebi može sadržavati više različitih vrijednosti.

8. Trebate znati sljedeće naredbe koje se koriste nad listama:

```
lista = [1, 2, 3] → definiramo listu; podaci idu unutar uglatih zagrada, odvojeni su zarezom
```

```
print(lista[0]) → ispisuje broj 1, odnosno element liste s rednim brojem 0 (isto kao i kod znakovnog niza)
```

```
print(lista[2]) → ispisuje broj 3, odnosno element liste s rednim brojem 2
```

```
lista.append(4) → dodaje broj 4 na kraj liste; sada lista ima novi element
```

```
lista.remove(2) → briše broj 2 iz liste, sada lista ima jedan element manje
```

```
lista.sort() → sortira listu, od najmanjeg do najvećeg broja
```

```
print(len(lista)) → ispisuje koliko lista ima elemenata (slično kao i kod znakovnog niza)
```

# Što trebate znati

---

9. Trebate znati napisati kratki program koji traži neki podatak u listi.

U sljedećem primjeru je napisan kratki program koji provjerava postoji li slovo 'A' u listi. Ako postoji, ispisuje na zaslon tekst 'Pronašao sam slovo A'. Slično trebate znati i vi napisati.

```
lista = ['V', 'A', 'T', 'R', 'A']
for k in range(len(lista)):
    if lista[k] == 'A':
        print('Pronašao sam slovo A')
```

Ova će petlja proći kroz sve elemente liste. Varijabla  $k$  će u prvom prolazu petlje imati vrijednost 0, a u zadnjem 4. Dakle u prvom prolazu petlje  $lista[k]$  će zapravo biti  $lista[0]$  i predstavljat će prvi element u listi.

# Što trebate znati

---

10. Trebate znati napisati kratki program koji u sebi sadrži potprogram. Naučili smo da potprogram počinje s riječi **def**. Svaki potprogram može imati 0, 1 ili više parametara. Oni se navode unutar zagrada nakon naziva potprograma.

Evo primjera programa koji u sebi sadrži potprogram. Potprogram ima dva parametra i ispisuje zbroj tih dvaju parametara:

```
def potprogram(a, b):  
    print(a + b)
```

```
a = 2  
b = 3
```

```
potprogram(a, b)
```

Ovdje smo definirali potprogram. Pazite na to da sve naredbe koje se nalaze unutar potprograma budu uvučene **udesno** za 4 mjesta!

Ovdje pozivamo potprogram. Tek sada će se krenuti izvršavati naredbe potprograma.