

## Листе

-5	-4	-3	-2	-1
Ana	Marko	Sara	Luka	Iva
0	1	2	3	4

`ucenik = ["Ana", "Marko", "Sara", "Luka", "Iva"]`

**Листа** је врста променљиве која омогућава меморисање више података у себи. Ти подаци могу бити различитог типа (бројеви, слова, речи...). Подаци који чине листу пишу се у угластим заградама одвојени зарезом.

Чланови листе зову се елементи листе, а места (позиције) у листи – индекси. Индекси не почињу од 1, већ од 0 (**важно**: на ово посебно обрати пажњу). Када не знаш колико листа има елемената, онда за индексе последњих чланова употреби негативне бројеве (-1, -2, -3...).

Пример листе:

`brojevi = [198, 203, 198, 203, 206, 199, 193]`

`a = brojevi[0]` Променљива **a** ће садржати број \_\_\_\_\_

`b = brojevi[-1]` Променљива **b** ће садржати број \_\_\_\_\_

Ако желимо да прикажемо бројеве 206 и 199, онда

`print(brojevi[4:6])` јер последњи број (овде 6) се не приказује (зато овде увећамо за 1)

Прикажи 3 последња броја.

`print(brojevi [-3:])` када оставимо празно после двотачке, приказују се сви следећи елементи

Када треба да се нађе број индекса, користимо наредбу **index**. (Пази, овде не пишемо угласте заграде).

`print(ucenik.index("Marko"))` добијамо индекс \_\_\_\_\_

Функцијом **len** израчунавамо дужину листе, функцијом **sum** израчунавамо збир елемената листе, функцијом **min** најмањи елемент у листи, а функцијом **max** највећи.

На пример, у задатку за израчунавање просека:

`brojevi = [198, 203, 198, 203, 206, 199, 193]`

`prosek = sum(brojevi) / len(brojevi)`

`print(prosek)` добијамо решење \_\_\_\_\_

Елементе листе веома једноставно можемо уредити од најмањег до највећег или по абecedном реду. Тада користимо функцију **sorted**.

`brojevi = [198, 203, 198, 203, 206, 199, 193]`

`poredjani_brojevi = sorted(brojevi)`

`print(poredjani_brojevi)` добијамо решење \_\_\_\_\_

# овде може и `print(sorted(brojevi))`

Задаци:

1. НПК уписује у листу **bodovi** 5 резултата такмичења (целе бројеве) и приказује најбољи и најслабији резултат.

```
bodovi=[]
for i in range(5):
    a=int(input("unesi bodove "))
    bodovi.append(a)
print(max(bodovi))
print(min(bodovi))
```

2. НПК уписује у листу **izostanci** изостанке ученика за 2 седмице (10 целих бројева), сабира их и приказује изостанке за 5. дан.

```
izostanci=[]
for i in range(10):
    a=int(input("unesi izostanke "))
    izostanci.append(a)
print(sum(izostanci))
print(izostanci[4])
```

## Скупови, торке и речници

**Скуп** (енгл. SET) је структура података која не дозвољава понављање својих елемената (нема дупликата података).

Скуп садржи елементе раздвојене зарезима између витичастих (великих) заграда.

```
strelci={"Mesi", "Ronaldo", "Nejmar", "Mbape"}
```

Листа може лако да се претвори у скуп наредбом set(). На пример:

```
golovi=["Mesi", "Mesi", "Ronaldo", "Ronaldo", "Nejmar", "Mbape"]
strelci=set(golovi)
```

добивамо: `strelci={"Mesi", "Ronaldo", "Nejmar", "Mbape"}`

Исто је и са стрингом: `print(set("blebetalo"))`

**Торка** је уређени пар или уређена n-торка.

Нпр. позиција фигуре на шаховској табли ("**b**", **6**)

Време које чине сат, минут и секунд `hms=(7, 25, 37)`

Торке су веома сличне листама. Главна разлика је та што се у листе могу додавати елементи, док су торке фиксирани дужине и једном када направимо торку нема начина да се она промени.

Не може: `hms[2]=10` Python јавља "TypeError".

**Речник** се користи за чување парова података **кључ/вредност**, где сваки кључ има своју вредност. Кључ је најчешће стринг, а вредност било који тип података.

```
cene_automobila = {"fiat 500I": 11990,
                  "renault clio": 9650,
                  "toyota corolla": 13990}
automobil = input("Unesi model automobila:")
print(cene_automobila[automobil])
```

"fiat 500I" је кључ  
11990 је вредност

Задатак: Прикажи елементе речника петљом “for”.

```
cene_automobila = {"fiat 500l": 11990,  
                  "renault clio": 9650,  
                  "toyota corolla": 13990}
```

```
for k, v in cene_automobila.items():  
    print(k,"vredi ",v)
```