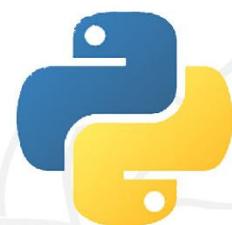


INFORMATIČKI KLUB

# FUTURA

## LIGA PROGRAMIRANJA



# python #3

## LIGA PROGRAMIRANJA U PYTHONU ZA OSNOVNE ŠKOLE – 3. RADIONICA

Tomo Sjekavica, Mario Miličević *Informatički klub FUTURA*  
Dubrovnik, 03. prosinca 2016.



Zaštićeno licencom <http://creativecommons.org/licenses/by-nc-sa/3.0/hr/>



# Creative Commons

---



- slobodno smijete:**
  - dijeliti — umnožavati, distribuirati i javnosti priopćavati djelo
  - remiksirati — prerađivati djelo
- pod slijedećim uvjetima:**
  - **imenovanje.** Morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
  - **nekomercijalno.** Ovo djelo ne smijete koristiti u komercijalne svrhe.
  - **dijeli pod istim uvjetima.** Ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, preradu možete distribuirati samo pod licencem koja je ista ili slična ovoj.

U slučaju daljnog korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela. Najbolji način da to učinite je linkom na ovu internetsku stranicu.

Od svakog od gornjih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava.

Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava.

Tekst licence preuzet je s <http://creativecommons.org/>.

# Raspored Lige programiranja

---

- 03.12.2016. – **3. radionica**
- 17.12.2016. – **2. kolo Lige programiranja**
- 21.01.2017. – 4. radionica
- 04.02.2017. – 3. kolo Lige programiranja
- ...
- Web stranica Lige programiranja:  
[www.futura.com.hr/liga-programiranja-u-pythonu-2016-2017/](http://www.futura.com.hr/liga-programiranja-u-pythonu-2016-2017/)



# Ponavljanje gradiva s 2. radionice

- Donošenje odluka (grananje programa) - u slučaju višestrukog izbora:

...

```
if uvjet_1:  
    blok naredbi_1  
elif uvjet_2:  
    blok naredbi_2  
...  
elif uvjet_n:  
    blok naredbi_n  
else:  
    blok naredbi  
...
```

Izvršit će se samo JEDAN od blokova naredbi!

# Zadatak: Bodovi



- Iz pravila Lige programiranja: "*Od ukupnog broja bodova učenika za poredak se svakom učeniku oduzima najlošije kolo.,,*"
- Napisati program kojim će se unijeti bodovi za četiri kola Lige programiranja za jednog učenika i izračunati njegov ukupan broj bodova.
- Ulazni podaci:
  - Četiri cijela broja, svaki u svom retku
- Izlazni podaci:
  - Ukupan broj bodova bez najlošijeg kola

Primjeri testnih podataka

ULAZ	ULAZ	ULAZ
100	50	95
100	100	65
90	70	65
80	60	85
IZLAZ	IZLAZ	IZLAZ
290	230	245

# Zadatak: Bodovi - rješenje



```
kolo1 = int(input('Unesi broj bodova 1. kola: '))
kolo2 = int(input('Unesi broj bodova 2. kola: '))
kolo3 = int(input('Unesi broj bodova 3. kola: '))
kolo4 = int(input('Unesi broj bodova 4. kola: '))

najlosije = kolo1

if kolo2 < najlosije:
    najlosije = kolo2
if kolo3 < najlosije:
    najlosije = kolo3
if kolo4 < najlosije:
    najlosije = kolo4

ukupno = kolo1 + kolo2 + kolo3 + kolo4 - najlosije
print(ukupno)
```



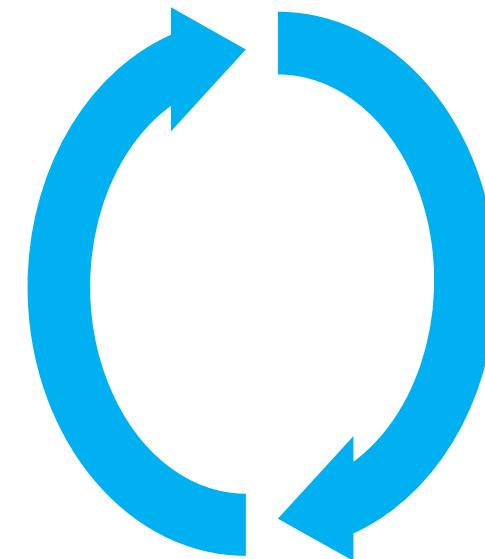
bodovi.py



# Ponavljanje bloka naredbi

- Često je u programima potrebno određeni broj puta ponoviti blok istih naredbi:

```
...  
za i := 1 do n činiti  
    naredba_1  
    ...  
    naredba_z  
...  
...
```



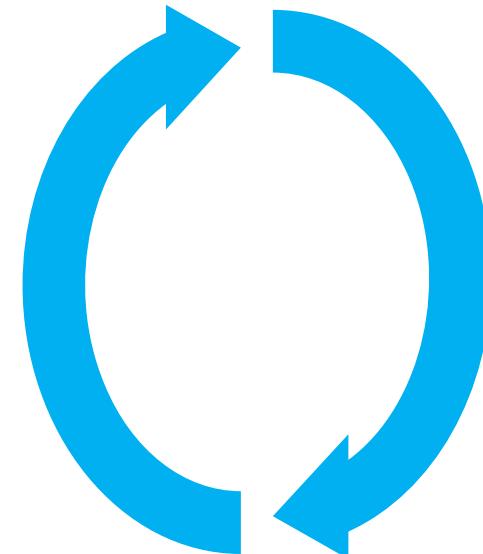
Programska petlja



# Ponavljanje bloka naredbi

- U Pythonu:

```
...  
for i in range(n):  
    naredba_1  
    ...  
    naredba_z  
    ...
```



Programska petlja

- Blok naredbi će se izvesti **n puta**, za vrijednosti varijable **i** od **0** do **n-1**.



# Ponavljanje bloka naredbi

- Provjeriti u interaktivnom sučelju (*shell*):

```
>>> for i in range(5):
    print(i)

0
1
2
3
4
>>>
```

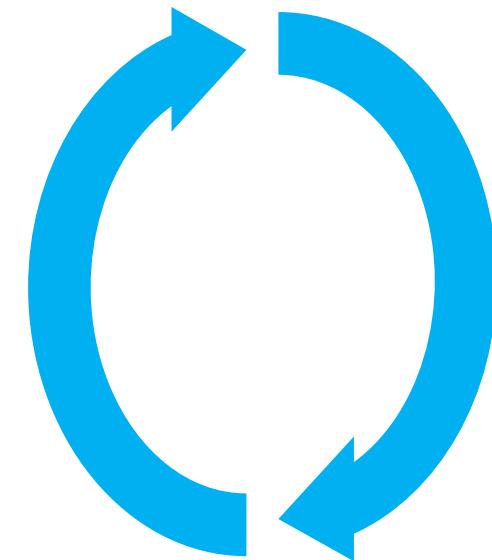
- Blok naredbi će se izvesti **5 puta**, za sve vrijednosti varijable **i** od **0** do **4**.



# Ponavljanje bloka naredbi

- Ako nam ne odgovara da vrijednost varijable **i** kreće od 0:

```
...  
for i in range(n,m):  
    naredba_1  
    ...  
    naredba_z  
    ...
```



Programska petlja

- Blok naredbi će se izvesti **m-n** puta, za sve vrijednosti varijable **i** od **n** do **m-1**.



# Ponavljanje bloka naredbi

- Provjeriti u interaktivnom sučelju (*shell*):

```
>>> for i in range(3, 10):  
    print(i)  
  
3  
4  
5  
6  
7  
8  
9  
>>>
```

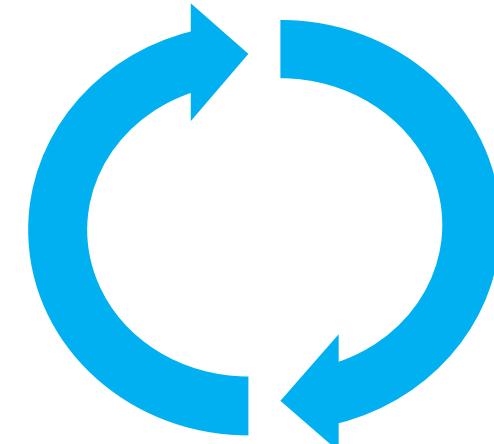
- Blok naredbi će se izvesti **7 puta**, za sve vrijednosti varijable *i* od 3 do 9.



# Ponavljanje bloka naredbi

- Isto tako možda nam ne odgovara da se vrijednost varijable **i** mijenja u koracima po 1:

```
...
for i in range(n,m,k):
    naredba_1
    ...
    naredba_z
...
...
```



Programska petlja

- Blok naredbi će se izvesti **m-n** puta, za vrijednosti varijable **i** od **n** do **m-1**, ali će se **i** svaki put uvećati za **k**.



# Ponavljanje bloka naredbi

- Provjeriti u interaktivnom sučelju (*shell*):

```
>>> for i in range(5, 20, 3):
    print(i)

5
8
11
14
17
>>>
```

- Varijabla **i** mijenja vrijednost od **5** do **19** (tj. 20-1), a korak promjene je **3**.



# Ponavljanje bloka naredbi

- Može li korak biti negativan broj?

```
>>> for i in range(10, 4, -1):  
    print(i)
```

```
10  
9  
8  
7  
6  
5  
>>>
```

- Varijabla **i** mijenja vrijednost od **10** do **5** (tj.  $4+1$ ), a korak promjene je **-1**.

# Zadatak: Bodovi 2



- Unos i izračun bodova realizirati s petljom**
- Iz pravila Lige programiranja: "*Od ukupnog broja bodova učenika za poredak se svakom učeniku oduzima najlošije kolo.,,*"
- Napisati program kojim će se unijeti bodovi za četiri kola Lige programiranja za jednog učenika i izračunati njegov ukupan broj bodova.
- Ulazni podaci:
  - Četiri cijela broja, svaki u svom retku
- Izlazni podaci:
  - Ukupan broj bodova bez najlošijeg kola

Primjeri testnih podataka

ULAZ	ULAZ	ULAZ
100	50	95
100	100	65
90	70	65
80	60	85
IZLAZ	IZLAZ	IZLAZ
290	230	245

# Zadatak: Bodovi 2 - rješenje



```
najlosije = 100
```

```
ukupno = 0
```

```
for i in range(4):  
    kolo = int(input('Unesi broj bodova: '))
```

```
if kolo < najlosije:  
    najlosije = kolo
```

```
ukupno = ukupno + kolo
```

```
ukupno = ukupno - najlosije  
print(ukupno)
```



bodovi2.py

# Zadatak: Zbroj ili umnožak



- Potrebno je s tipkovnice unijeti prirodni broj **N**.
- Ako je **N** neparan, ispisati zbroj svih neparnih prirodnih brojeva manjih od **N**.
- Ako je **N** paran, ispisati umnožak svih parnih prirodnih brojeva manjih od **N**.
- Ulazni podaci:
  - Prirodni broj **N** ( $N > 0$ )
- Izlazni podaci:
  - Zbroj ili umnožak

Vrijeme



Primjeri testnih podataka

ULAZ	ULAZ	ULAZ	ULAZ
13	10	55	8
IZLAZ	IZLAZ	IZLAZ	IZLAZ
Zbroj: 36	Umnožak: 384	Zbroj: 729	Umnožak: 48

# Zbroj ili umnožak - rješenje



```
N = int(input('Unesi prirodni broj: '))

if N%2 == 1:
    zbroj = 0
    for i in range(1, N, 2):
        zbroj = zbroj + i
    print('Zbroj: ', zbroj)
else:
    umnozak = 1
    for i in range(2, N, 2):
        umnozak = umnozak * i
    print('Umnožak: ', umnozak)
```



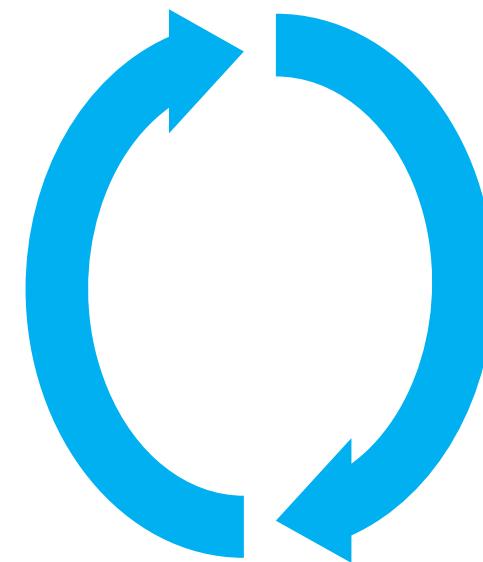
[zbrojiliumnozak.py](#)



# Uvjetno ponavljanje bloka naredbi

- Ponekad je u programima potrebno određeni broj puta ponoviti blok istih naredbi, ali samo dok je neki uvjet ispunjen:

```
...
dok je uvjet činiti
    naredba_1
    ...
    naredba_z
...
...
```



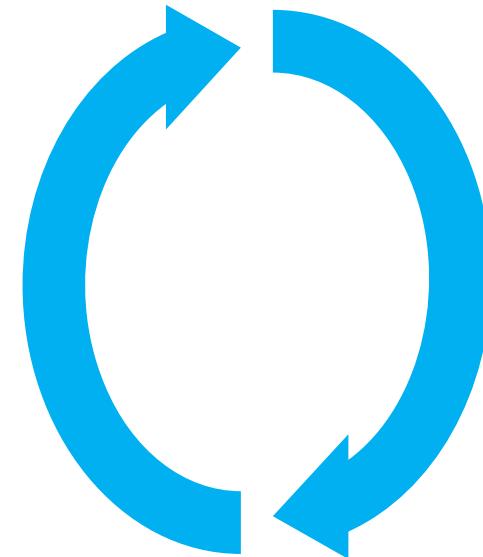
**Uvjetna programska petlja**



# Uvjetno ponavljanje bloka naredbi

- U Pythonu:

```
...  
while uvjet:  
    naredba_1  
    ...  
    naredba_i  
    ...
```



**Uvjetna programska petlja**

- Blok naredbi će se izvoditi dok je *uvjet* ispunjen (daje vrijednost **True**).



# Uvjetno ponavljanje bloka naredbi

- Provjeriti u interaktivnom sučelju (shell):

```
>>> i = 5
>>> while i < 10:
...     print(i)
...     i = i +1
5
6
7
8
9
>>>
```

Početna vrijednost  
variabile i

Dvije naredbe u  
bloku će se  
izvršavati više  
puta – sve dok je  
 $i < 10$

Brojač (ovdje varijabla **i**)  
mora se prije petlje  
inicijalizirati, a u petlji  
povećavati (smanjivati)!



# Uvjetno ponavljanje bloka naredbi

- Treba paziti da se ne napiše "beskonačna" petlja 😞 :

```
>>>  
>>>  
>>> br = 1  
>>> while br < 10:  
      print(br)
```

Problem: brojaču **br** ne mijenjamo vrijednost u petlji, pa je uvjet **br<10** stalno istinit 😞

- Što će se dogoditi?
- Napisali smo "beskonačnu" petlju, pa je moramo prekinuti istovremenim pritiskom na **Ctrl C**



# Uvjetno ponavljanje bloka naredbi

- Treba paziti da se ne napiše "beskonačna" petlja ☹ :

```
>>>  
>>>  
>>> br = 5  
>>> while br != 10:  
        print(br)  
        br = br + 2
```

Problem: brojaču *br* mijenjamo vrijednost u petlji, ali tako da je uvjet *br!=10* stalno istinit ☹

- Što će se dogoditi?
- Napisali smo "beskonačnu" petlju, pa je moramo prekinuti istovremenim pritiskom na **Ctrl C**



# Programske petlje - zadatak

- **for** i **while** programske petlje često se koriste kada je potrebno unijeti više podataka, ali u vrijeme pisanja programa nije poznat njihov broj.
- Primjer: potrebno je napisati program za izračun prosječne ocjene iz nekog predmeta, ali broj ocjena koje se uzimaju u obzir može biti različit.
- U ovom programu moramo koristiti **for** ili **while** programsku petlju!

# Programske petlje - primjer



- Rješenje – pomoću **for** petlje:

```
brOcj = int(input('Koliko ima ocjena? '))
zbroj = 0
for i in range (1, brOcj+1):
    ocjena = int(input('Unesi ocjenu: '))
    zbroj = zbroj + ocjena
print ('Prosjek svih ocjena je ', zbroj;brOcj)
```



projekt1.py

```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
>>>
Koliko ima ocjena? 3
Unesi ocjenu: 5
Unesi ocjenu: 3
Unesi ocjenu: 4
Prosjek svih ocjena je  4.0
>>> |
Ln: 61124 Col: 4
```

# Programske petlje - primjer



- Rješenje – pomoću **while** petlje. Moramo se dogovoriti koji podatak prekida petlju (npr.**0**)

```
zbroj = 0
brOcj = 0
ocjena = int(input('Unesi ocjenu (0 za kraj): '))

while ocjena != 0:
    brOcj = brOcj + 1
    zbroj = zbroj + ocjena
    ocjena = int(input('Unesi ocjenu (0 za kraj): '))

print('Prosjek svih ocjena je ', zbroj;brOcj)
```



projek2.py

# Programske petlje - primjer



- Kako spriječiti unos pogrešnih podataka?

```
br0cj = 0
while br0cj < 1:
    br0cj=int(input('Koliko ima ocjena? '))
zbroj = 0
ocjena = 0
for i in range (1, br0cj+1):
    while ocjena < 1 or ocjena > 5:
        ocjena = int(input('Unesi ocjenu: '))
        if ocjena < 1 or ocjena > 5:
            print('Neispravna ocjena')
    zbroj = zbroj + ocjena
    ocjena = 0
print ('Proshek svih ocjena je ', zbroj;br0cj)
```



projek3.py

Ne može se unijeti 0 ili negativan broj

Ne može se unijeti neispravna ocjena

# Zadatak: Bacanje koplja



- Na kraju svakog treninga trener Sari odredi udaljenost preko koje mora 3 puta baciti koplje da bi završila s treningom.
- Svaka priprema i bacanje koplja traje 35 sekundi.
- Napisati program u kojem se prvo unosi zadana trenerova udaljenost, a nakon toga udaljenosti na koje je Sara bacila koplje. Unos se završava kada Sara 3 puta baci koplje preko zadane udaljenosti.
- Ispisati koliko je Sari trebalo pokušaja da prebaci 3 puta zadalu udaljenost i ukupno vrijeme trajanja kraja treninga u minutama i sekundama.

Vrijeme



# Zadatak: Bacanje koplja

Vrijeme



- Sara završava s treningom kada 3 puta baci koplje preko zadane udaljenosti.
- Svaka priprema i bacanje koplja traje 35 sekundi.



Primjeri testnih podataka

ULAZ	ULAZ	ULAZ	ULAZ
60	63	60	64
50	50	65	0
0	0	64	50
56	64	63	64
59	65		66
61	66		67
62			63
50			68
67			
IZLAZ	IZLAZ	IZLAZ	IZLAZ
8 puta	5 puta	3 puta	7 puta
4 min i 40 sekundi	2 min i 55 sekundi	1 min i 45 sekundi	4 min i 5 sekundi

# Bacanje koplja - rješenje



```
zadanaUdaljenost = int(input('Unesi zadanu '))
brBacanja = 0
brPreko = 0
while brPreko != 3:
    udaljenost = int(input('Udaljenost: '))
    brBacanja = brBacanja + 1
    if udaljenost > zadanaUdaljenost:
        brPreko = brPreko + 1
```

```
vrijeme = brBacanja * 35
print(brBacanja, 'puta.')
print('Kraj treninga je trajao:', vrijeme//60,
      'minuta i', vrijeme%60, 'sekundi.')
```



bacanjekoplja.py



# Zadatak: Palindrom

- Neki broj je palindrom ako ne mijenja vrijednost kada se znamenke broja napišu obrnutim redoslijedom. Npr. 33, 121, 2002, 123454321
- Napisati program u kojem se unosi prirodan broj  $N \geq 10$ . Provjeriti i ispisati da li je uneseni broj palindrom.

- Algoritam:

$N = 2002$

Početak:  $pal = 0$

1. korak:  $pal = pal * 10 + pom \% 10 = 2$

2. korak:  $pal = pal * 10 + pom \% 10 = 20$

3. korak:  $pal = pal * 10 + pom \% 10 = 200$

4. korak:  $pal = pal * 10 + pom \% 10 = 2002$

obrnuti broj  
(palindrom)

pomoćna  
varijabla

$pom = 2002$

$pom = 2002 / 10 = 200$

$pom = 200 / 10 = 20$

$pom = 20 / 10 = 2$

$pom = 2 / 10 = 0$

Vrijeme



# Zadatak: Palindrom - rješenje



```
N = int(input('Unesi prirodni broj N>= 10: '))
palindrom = 0
pom = N

while pom > 0:
    palindrom = palindrom * 10 + pom % 10
    pom = pom // 10

if palindrom == N:
    print('Broj', N, 'je palindrom')
else:
    print('Broj', N, 'nije palindrom')
```



palindrom.py

# Ne zaboravite!

---

- Za 15 dana – **u subotu 17.12.2016.** –  
**2. kolo Lige programiranja**
- 5./6. razredi** - početak **10:00**
- 7./8. razredi** - početak **10:00**
- 3 zadatka rješavate 75 minuta**
- nemojte kasniti!**

