

INFORMATIČKI KLUB

FUTURA

LIGA PROGRAMIRANJA



python

#4

LIGA PROGRAMIRANJA U PYTHONU ZA

OSNOVNE ŠKOLE – 3. RADIONICA



Tomo Sjekavica, Mario Miličević *Informatički klub FUTURA*
Dubrovnik, 20. siječnja 2018.



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



Creative Commons

-  ☐ **slobodno možete:**
 - **Dijelite dalje** — možete umnažati i redistribuirati materijal u bilo kojem mediju ili formatu
 - **Stvarajte prerade** — možete remiksirati, mijenjati i prerađivati djelo
-  ☐ **pod slijedećim uvjetima:**
 - **Imenovanje** — Morate adekvatno navesti autora, uvrstiti link na licencu i naznačiti eventualne izmjene. Možete to učiniti na bilo koji razuman način, ali ne smijete sugerirati da davatelj licence izravno podupire Vas ili Vaše korištenje djela.
 - **Nekomercijalno** — Ne smijete koristiti materijal u kommercijalne svrhe.
 - **Dijeli pod istim uvjetima** — Ako remiksirate, mijenjate ili prerađujete materijal, Vaše prerade morate distribuirati pod istom licencom pod kojom je bio izvornik.

U slučaju daljnjeg 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

- ☐ 20.01.2018. – 3. radionica
- ☐ 03.02.2018. – Sveti Vlaho
- ☐ 3. kolo Lige programiranja
10.02.2018. – novi termin
- ☐ 17.02.2018. – 4. radionica programiranja
- ☐ 03.03.2018. – 4. kolo Lige programiranja
- ☐ ...



- ☐ Web stranica Lige programiranja:

www.futura.com.hr/liga-programiranja-u-pythonu-2017-2018/

Ponavljjanje gradiva s 2. radionice



□ Pisanje Python programa:

The image shows the Python IDLE environment with several windows and annotations:

- Python IDLE**: The main application window.
- Python IDLE editor**: The window where the code is written. It contains the following code:

```
#Jednolinijski komentari
#Ovo je moj prvi Python program.

"""
Komentar u više linija.
Mogu se koristiti i jednostruki navodnici.
Ovo je moj prvi Python progr.
"""

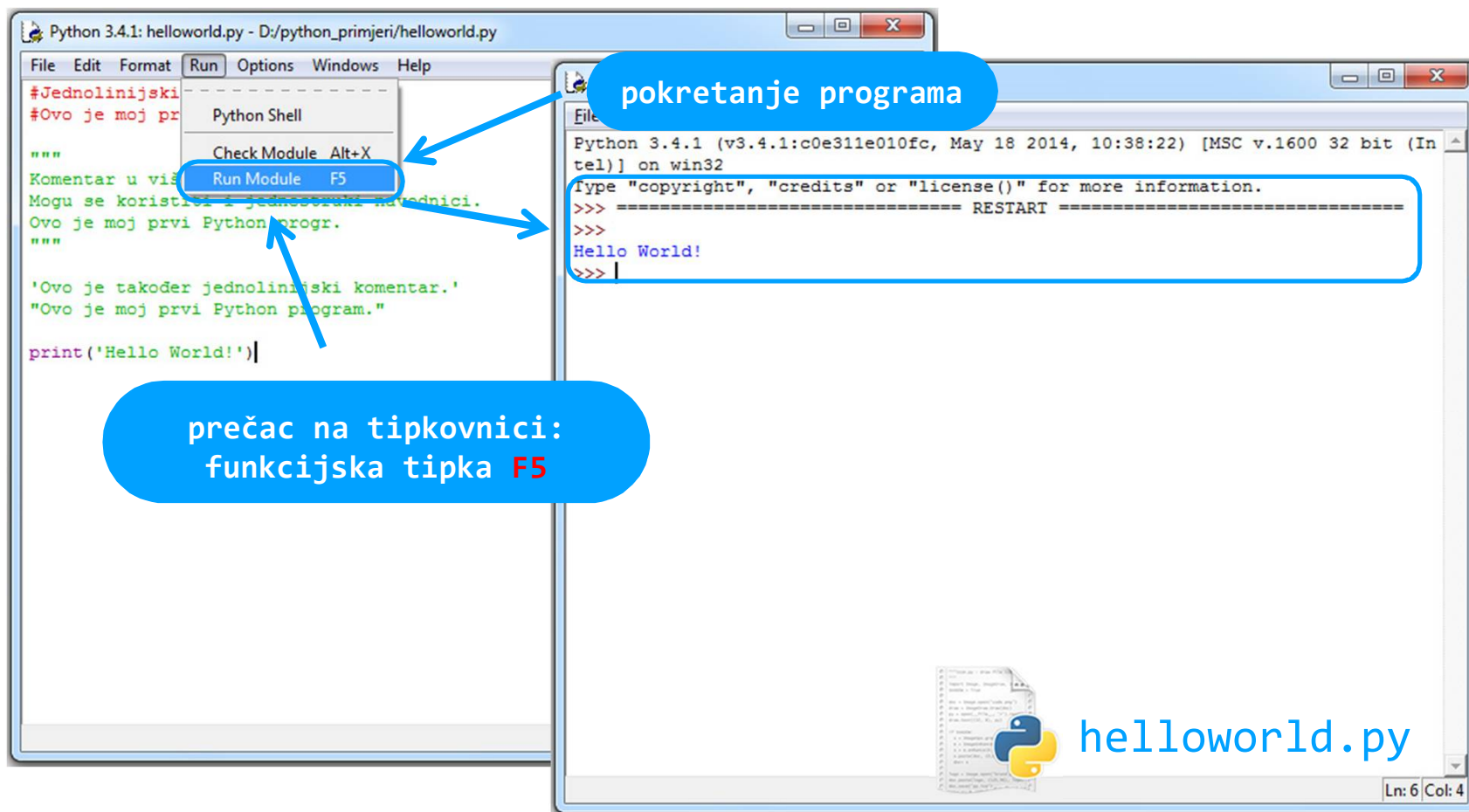
'Ovo je također jednolinijski komentar.'
'Ovo je moj prvi Python program.'

print('Hello World!')
```
- komentari**: Annotation pointing to the multi-line comment in the code.
- ekstenzija .py**: Annotation pointing to the file extension in the save dialog.
- unos naziva programa**: Annotation pointing to the file name input field in the save dialog.
- spremanje programa**: Annotation pointing to the save button in the save dialog.
- kod programa**: Annotation pointing to the code in the editor.
- Save As**: A dialog box for saving the file. The file name is `helloworld.py` and the save as type is `Python files (*.py;*.pyw)`.

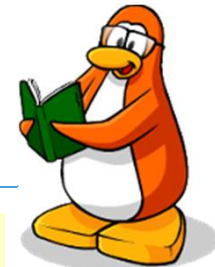
Ponavljjanje gradiva s 2. radionice



□ Pokretanje programa u Python IDLE-u:



Ponavljjanje gradiva s 2. radionice



□ Aritmetički operatori

zbrajanje	+
oduzimanje	-
množenje	*
dijeljenje	/
cjelobrojno dijeljenje	//
modulo (ostatak od dijeljenja)	%
potenciranje	**

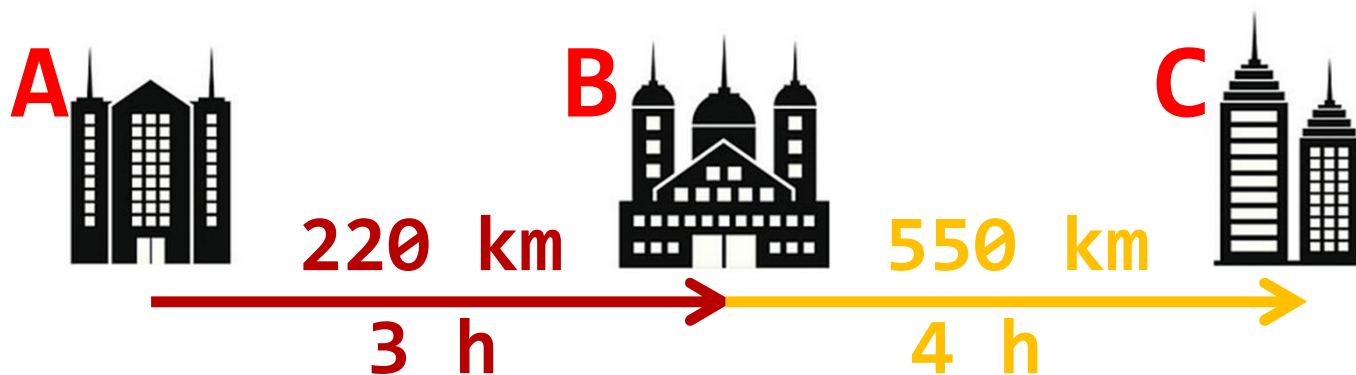
```
>>> a = 4
>>> b = 5
>>> a + b
9
>>> a / b
0.8
>>> a // b
0
>>> a - b * 2
-6
>>> (a - b) * 2
-2
>>> a % b
4
```

- Prvenstvo pri izvođenju ima potenciranje, pa nakon toga množenje, dijeljenje, cjelobrojno dijeljenje i modulo, te na kraju zbrajanje i oduzimanje

Ponavljjanje gradiva s 2. radionice



- ☐ Program u kojem se unose:
 - udaljenost između gradova A i B (u **km**),
 - udaljenost između gradova B i C (u **km**),
 - vrijeme vožnje od grada A do grada B (u **h**)
 - vrijeme vožnje od grada B do grada C (u **h**)
- ☐ Program izračunava prosječnu brzinu na putu iz grada A u grad C (preko grada B).



Ponavljjanje gradiva s 2. radionice



```
a_b = int(input('Udaljenost između A i B: '))
b_c = int(input('Udaljenost između B i C: '))
vr_a_b = int(input('Trajanje puta od A do B: '))
vr_b_c = int(input('Trajanje puta od B do C: '))
pros_brz = (a_b + b_c) / (vr_a_b + vr_b_c)
print('Prosječna brzina:', pros_brz, 'km/h')
```

A screenshot of a Python 3.6.3 Shell window. The window has a menu bar with File, Edit, Shell, Debug, Options, Window, and Help. The main text area shows the program's execution with user input: 'Udaljenost između A i B: 220', 'Udaljenost između B i C: 550', 'Trajanje puta od A do B: 3', 'Trajanje puta od B do C: 4', and the output 'Prosječna brzina: 110.0 km/h'. The prompt '>>>' is visible at the bottom. The status bar at the bottom right shows 'Ln: 10 C'.

prosjecnaBrzina.py

Ponavljanje gradiva s 2. radionice



- Izvođenje jedne ili više naredbi na temelju ispitivanja nekog uvjeta:

```
...
if uvjet:
    naredba1_1
    ...
    naredba1_n
...
```

- Odabir jedne od dvije mogućnosti (izvršava se samo jedan blok naredbi):

```
...
if uvjet:
    naredba1_1
    ...
    naredba1_n
else:
    naredba2_1
    ...
    naredba2_m
...
```

- Višestruki izbor izvršava se samo jedan od blokova naredbi):

```
...
if uvjet_1:
    blok_naredbi_1
elif uvjet_2:
    blok_naredbi_2
...
elif uvjet_n:
    blok_naredbi_n
else:
    blok_naredbi
...
```

Ponavljjanje gradiva s 2. radionice



□ Primjer: usporedba brojeva

```
A = int(input('Unesi prirodni broj A: '))
B = int(input('Unesi prirodni broj B: '))
if A > B:
    print('Broj A je veći!')
elif B > A:
    print('Broj B je veći!')
else:
    print('Brojevi A i B su jednaki!')
```

Izvršit
će se
samo
JEDAN od
ova tri
bloka
naredbi!

```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Unesi prirodni broj A: 234
Unesi prirodni broj B: 317
Broj B je veći!
>>>
```

 usporedba.py

```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Unesi prirodni broj A: 105
Unesi prirodni broj B: 105
Brojevi A i B su jednaki!
>>>
```

Ponavljjanje gradiva s 2. radionice



- Što ako je uvjet na temelju kojeg treba donijeti odluku složen?

- Logički operatori:

logička I operacija	and
logička ILI operacija	or
NE operacija (negacija)	not

- Redoslijed izvođenja logičkih operacija:

1.	not
2.	and
3.	or

```
>>> a = 2
>>> b = 3
>>> c = 10
>>> a > b
False
```

```
>>>
>>> c > b
True
```

```
>>>
>>> a > b and c > b
False
```

```
>>> a > b or c > b
True
>>>
```

za logičku **AND** operaciju rezultat će biti **True** (istina) samo ako su oba izraza **True** (istinita)

za logičku **OR** operaciju rezultat će biti **True** (istina) već ako je jedan od izraza **True** (istinit)

Zadatak: Natjecanje



- ☐ Na jednodnevnom natjecanju iz programiranja Futura je odlučila podijeliti više zlatnih, srebrnih i brončanih medalja prema sljedećim kriterijima:
 - Zlatna medalja za sve koji ostvare 90 bodova ili više.
 - Srebrna medalja za sve koji ostvare između 75 i 90 bodova.
 - Brončana medalja za sve koji ostvare od 60 do 75 bodova, uključujući i 60 i 75 bodova.
- ☐ S tipkovnice unijeti ostvaren broj bodova (prirodni broj).
- ☐ Ispisati koju je medalju osvojio natjecatelj ili poruku da nije osvojio medalju.

Vrijeme





Zadatak: Natjecanje - rješenje

```
bodovi = int(input('Unesi broj bodova: '))
```

```
if bodovi >= 90:
```

```
    print('Zlatna medalja')
```

```
elif bodovi > 75 and bodovi < 90:
```

```
    print('Srebrna medalja')
```

```
elif bodovi >= 60 and bodovi <= 75:
```

```
    print('Brončana medalja')
```

```
else:
```

```
    print('Nije osvojena medalja')
```

```
Unesi broj bodova: 100
Zlatna medalja
>>>
=====+===== RESTART: C:
/Users/Tomo/Desktop/natjecan
je.py =====
Unesi broj bodova: 80
Srebrna medalja
>>>
===== RESTART: C:
/Users/Tomo/Desktop/natjecan
je.py =====
Unesi broj bodova: 75
Brončana medalja
>>>
===== RESTART: C:
/Users/Tomo/Desktop/natjecan
je.py =====
Unesi broj bodova: 50
Nije osvojena medalja
>>>
```



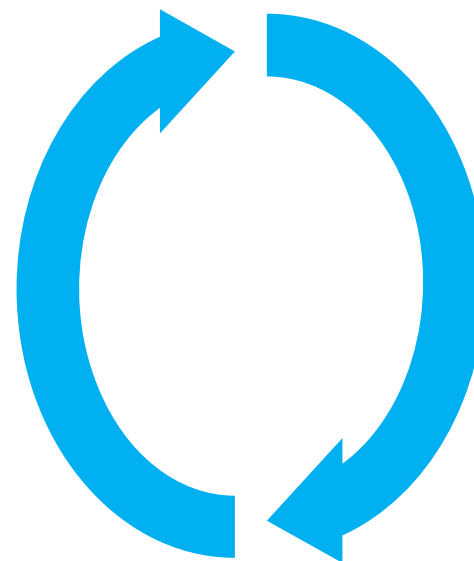
natjecanje.py

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  
...
```



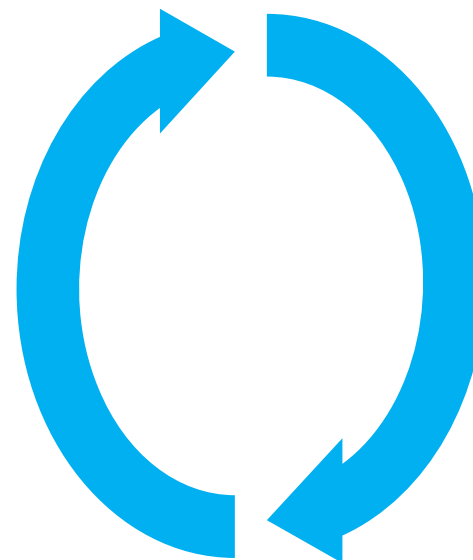
Programska petlja

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(6):  
        print(i)
```

0

1

2

3

4

5

```
>>>
```

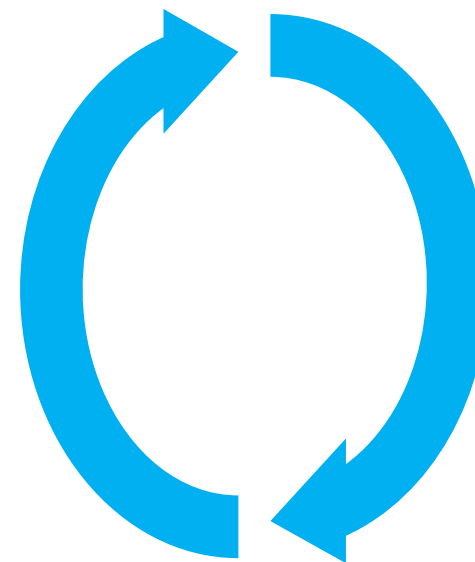
- Blok naredbi će se izvesti **6 puta**, za sve vrijednosti varijable **i** od **0** do **5**.

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(6, 13):  
        print(i)
```

```
6  
7  
8  
9  
10  
11  
12  
>>>
```

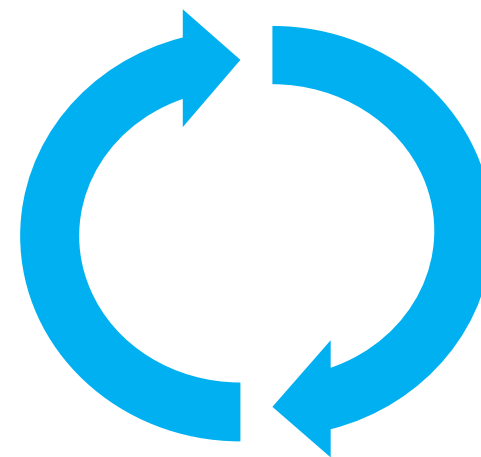
- Blok naredbi će se izvesti **7 puta**, za sve vrijednosti varijable **i** od **6** do **12**.

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 korak **k**.

Ponavljanje bloka naredbi



- Provjeriti u interaktivnom sučelju (*shell*):

```
>>> for i in range(5, 23, 3):  
        print(i)
```

5

8

11

14

17

20

>>>

- Varijabla *i* mijenja vrijednost od **5** do **22** (tj. $23-1$), a korak promjene je **3**.

Ponavljanje bloka naredbi



□ Može li korak biti negativan broj?

```
>>> for i in range(12, 6, -1):  
        print(i)
```

12

11

10

9

8

7

```
>>>
```

□ Varijabla **i** mijenja vrijednost od **12** do **7** (tj. $6+1$), a korak promjene je **-1**.

Zadatak: Višekratnici



- Napisati program u kojem se unosi jedan prirodni broj. Potrebno je ispisati sve dvoznamenkaste višekratnike unesenog broja.

Primjeri testnih podataka

ULAZ	ULAZ	ULAZ	ULAZ
7	8	9	10
IZLAZ	IZLAZ	IZLAZ	IZLAZ
14	16	18	10
21	24	27	20
28	32	36	30
35	40	45	40
42	48	54	50
49	56	63	60
56	64	72	70
63	72	81	80
70	80	90	90
77	88	99	
84	96		
91			
98			

Vrijeme





Zadatak: Višekratnici - rješenje

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

```
for i in range(10, 100):  
    if i % broj == 0:  
        print(i)
```



visekratnici.py

```
===== RESTART: C:/Users/Tomo/Desktop/visekratnici.py =====
```

```
Unesi prirodni broj: 7
```

```
14  
21  
28  
35  
42  
49  
56  
63  
70  
77  
84  
91  
98  
>>>
```

```
===== RESTART: C:/Users/Tomo/Desktop/visekratnici.py =====
```

```
Unesi prirodni broj: 8
```

```
16  
24  
32  
40  
48  
56  
64  
72  
80  
88  
96  
>>>
```

```
===== RESTART: C:/Users/Tomo/Desktop/visekratnici.py =====
```

```
Unesi prirodni broj: 10
```

```
10  
20  
30  
40  
50  
60  
70  
80  
90  
>>> |
```

Zadatak: Liga



- ☐ Zbog velikog interesa umjesto jednodnevnog natjecanja Futura je organizirala Ligu, na kojoj se polaznici natječu na **N** kola.
- ☐ Na kraju Lige zlatne, srebrne i brončane medalje se dijele prema sljedećim kriterijima:
 - Zlatna medalja za sve koji ostvare 90% ukupnih bodova ili više.
 - Srebrna medalja za sve koji ostvare između 75% i 90% ukupnih bodova.
 - Brončana medalja za sve koji ostvare od 60% do 75% ukupnih bodova, uključujući i 60% i 75%.
- ☐ Maksimalan broj bodova na kolu je 100.

Zadatak: Liga

Vrijeme



□ Ulazni podaci:

- Prirodni broj N – broj kola Lige
- N prirodnih brojeva – ostvaren broj bodova na pojedinom kolu Lige

□ Izlazni podaci:

- prosjek ostvarenih bodova
- poruka koju je medalju osvojio ili poruku da nije osvojio medalju



prosijek ≥ 90



prosijek > 75 and prosijek < 90



prosijek ≥ 60 and prosijek ≤ 75

Primjeri testnih podataka

ULAZ	ULAZ
4	3
100	30
90	40
95	35
90	
IZLAZ	IZLAZ
93.75	35.0
Zlatna medalja	Nije osvojena medalja

Primjeri testnih podataka

ULAZ	ULAZ
5	4
100	98
50	74
66	80
80	75
70	
IZLAZ	IZLAZ
73.2	81.75
Brončana medalja	Srebrna medalja



Zadatak: Liga - rješenje

```
n = int(input('Unesi broj kola Lige: '))
suma = 0
for i in range(n):
    bodovi = int(input('Unesi broj bodova: '))
    suma = suma + bodovi
prosjeak = suma / n
print(prosjeak)
if prosjeak >= 90:
    print('Zlatna medalja')
elif prosjeak > 75 and prosjeak < 90:
    print('Srebrna medalja')
elif prosjeak >= 60 and prosjeak <= 75:
    print('Brončana medalja')
else:
    print('Nije osvojena medalja')
```

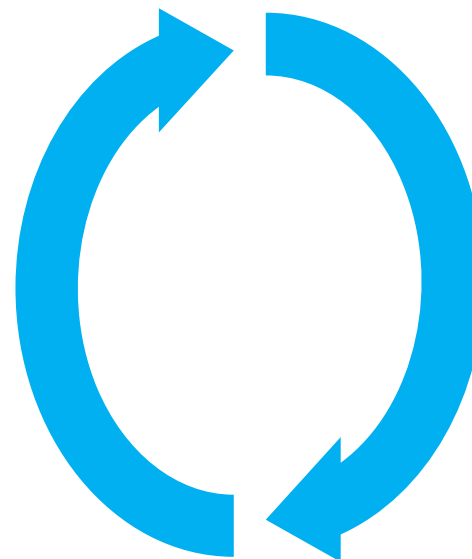


Uvjetno ponavljanje bloka naredbi



- 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  
...
```



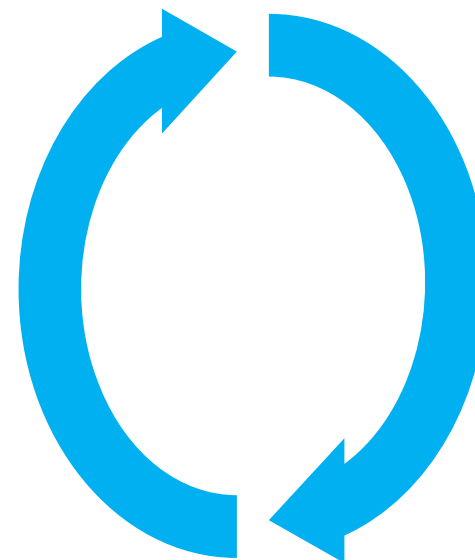
Uvjetna programska petlja

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 = 1
>>> while i < 9:
    print(i)
    i = i + 1
```

Početna vrijednost
varijable *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)!

1
2
3
4
5
6
7
8
>>>

Uvjetno ponavljanje bloka naredbi



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

```
>>>  
>>>  
>>> br = 2  
>>> 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** i **C**

Uvjetno ponavljanje bloka naredbi



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

```
>>>
>>>
>>> br = 1
>>> 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** i **C**

Programske petlje - primjer



- ❑ **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)
```



prosjek1.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
>>> |
```

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)
```



prosjek2.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 ('Prosjek svih ocjena je ', zbroj/br0cj)
```

Ne može se
unijeti 0
ili
negativan
broj



prosjek3.py

Ne može se
unijeti
neispravna
ocjena

Ponavljanje blokova naredbi



- ❑ Probleme je često moguće riješiti upotrebom i **for** i **while** programske petlje, ali kod nekih problema ipak prednost ima jedna od varijanti.

```
>>> for i in range(1, 6, 2):  
        i2 = i ** 2  
        print(i, '\t', i2)  
  
1          1  
3          9  
5         25  
>>>
```

```
>>> i = 1  
>>> while i < 6:  
        i2 = i ** 2  
        print(i, '\t', i2)  
        i = i + 2  
  
1          1  
3          9  
5         25  
>>>
```

- ❑ Ako je poznat broj izvođenja naredbi unutar petlje obično se koristi **for** petlja.
- ❑ Ako nije poznat broj ponavljanja naredbi unutar petlje obično se koristi **while** petlja.

Zadatak: Trening



- ☐ Luka trenira nogomet, pa da bi bio u dobroj formi svako jutro prije škole trči oko svoje zgrade. Kako ne bi zakasnio u školu odlučio je stati s trčanjem nakon što mu za neki krug treba više od 60 sekundi.
- ☐ Napisati program u kojem se unosi vrijeme u sekundama za svaki krug redom, a nakon posljednjeg kruga se ispisiuje broj pretrčanih krugova, i ukupno vrijeme u minutama i sekundama.

Zadatak: Trening



- Luka prestaje s trčanjem nakon što mu za neki krug treba više od 60 sekundi.

Primjeri testnih podataka

ULAZ	ULAZ	ULAZ	ULAZ
50	30	66	30
55	40		35
60	50		30
65	60		40
	70		55
			60
			58
			61
IZLAZ	IZLAZ	IZLAZ	IZLAZ
4	5	1	8
3 min. i 50 sek.	4 min. i 10 sek.	1 min. i 6 sek.	6 min. i 9 sek.

Vrijeme





Zadatak: Trening - rješenje

```
sekunde = 0
```

```
krugovi = 0
```

```
vrijeme = 0
```

```
while vrijeme <= 60:
```

```
    vrijeme = int(input('Unesi vrijeme za krug:'))
```

```
    sekunde = sekunde + vrijeme
```

```
    krugovi = krugovi + 1
```

```
print('Broj pretrčanih krugova: ', krugovi)
```

```
print('Luka je trčao', sekunde // 60,  
      'minuta i', sekunde % 60,  
      'sekundi.')
```



trening.py

Zadatak: Temperature



- Unijeti prirodni broj N u intervalu od 4 do 44. Ako se unese broj izvan intervala potrebno je ponoviti unos broja.
- Unijeti N brojeva s pomičnom točkom koje predstavljaju temperature u Celzijevim stupnjevima.
- Za svaku unesenu vrijednost izračunati i ispisati temperaturu u Fahrenheitovim stupnjevima.

$$T_F = \frac{9}{5} T_C + 32$$

Vrijeme





Zadatak: Temperature - rješenje

```
N = 0
```

```
while N < 4 or N > 44:
```

```
    N = int(input('Unesi broj [4,44]: '))
```

```
for i in range(N):
```

```
    tempC = float(input('Unesi temperaturu: '))
```

```
    tempF = (9/5) * tempC + 32
```

```
    print('Temperatura u Fahrenheitovim:', tempF)
```



temperature.py

Ne zaboravite!

- ☐ Za 21 dan – u subotu 10.02.2018. –
3. kolo Lige programiranja
- ☐ **5./6. razredi** - početak **10:00**
- ☐ **7./8. razredi** - početak **10:00**
- ☐ 3 zadatka rješavate **75** minuta
- ☐ nemojte kasniti!

