

3. RADIONICA

LIGA PROGRAMIRANJA



python

#5

Dubrovnik, 16. siječnja 2024.



INFORMATIČKI KLUB
FUTURA



SVEUČILIŠTE
U DUBROVNIKU
UNIVERSITY
OF DUBROVNIK



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 licencom koja je ista ili slična ovoj.

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

- ...
- 16.01.2024 – **3. radionica**
- 30.01.2024 – **3. kolo Lige programiranja**
- 13.02.2024 – 4. radionica
- 27.02.2024 – 4. kolo Lige programiranja
- ...

□ Web stranica Lige programiranja:

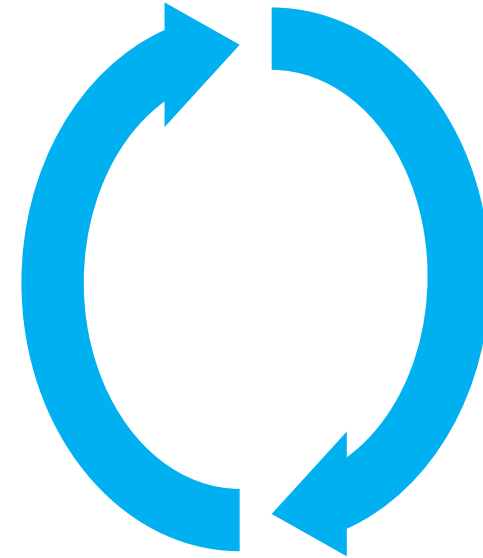
<https://www.futura.com.hr/liga-programiranja-u-pythonu-2023-2024/>



Ponavljjanje gradiva s 2. radionice

□ Ponavljjanje bloka naredbi:

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



Programska petlja

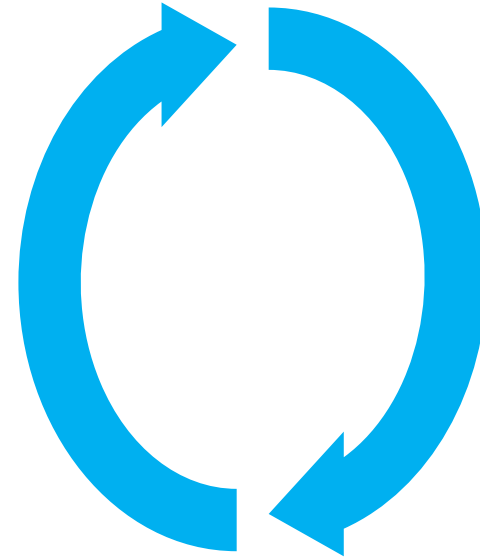
Blok naredbi će se izvesti **n puta**, za vrijednosti varijable **i** od **0** do **n-1**.



Ponavljanje gradiva s 2. radionice

□ Ponavljanje bloka naredbi:

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



Programska petlja

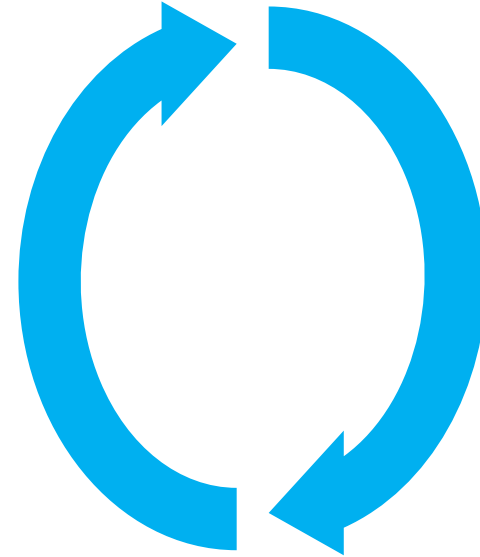
Blok naredbi će se izvesti **m-n puta**, za sve vrijednosti varijable **i** od **n** do **m-1**.



Ponavljjanje gradiva s 2. radionice

□ Ponavljjanje bloka naredbi:

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



Programska petlja

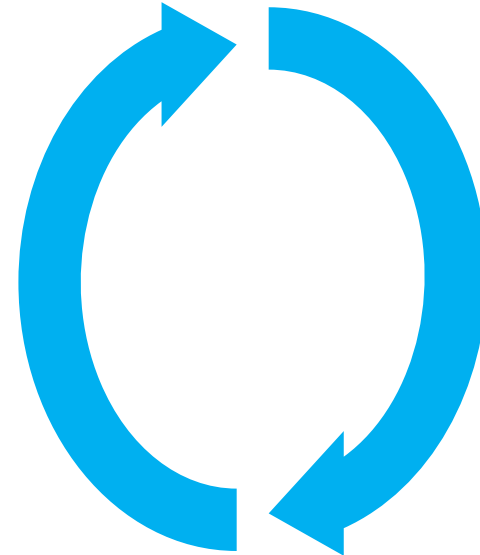
Blok naredbi će se izvesti **m-n puta**, za vrijednosti varijable **i** od **n** do **m-1**, ali će se **i** svaki put uvećati za **k**.



Ponavljanje gradiva s 2. radionice

- Uvjetno ponavljanje bloka naredbi:

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



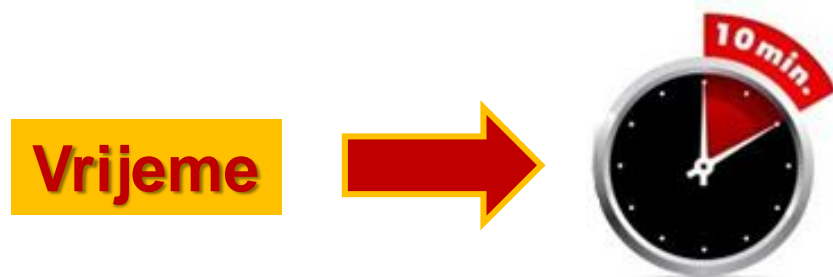
Uvjetna programska petlja

Blok naredbi će se izvoditi dok je **uvjet** ispunjen (daje vrijednost **True**). Potrebno **inicijalizirati** brojač prije petlje, a u petlji ga **povećavati** (smanjivati).



Zadatak: Odbrojavanje

- Napisati program u kojem se unosi početno vrijeme u minutama i sekundama, te se ispisi odbrojavanje do nule samo u sekundama (5./6. razredi), ili u formatu ukupne sekunde – minute : sekunde (7./8. razredi). Npr.



(5./6. razredi)	(7./8. razredi)
Unesi minute: 3	Unesi minute: 3
Unesi sekunde: 2	Unesi sekunde: 2
182	182 – 3 : 2
181	181 – 3 : 1
180	180 – 3 : 0
179	179 – 2 : 59
...	...
2	2 – 0 : 2
1	1 – 0 : 1
0	0 – 0 : 0



Odbrojavanje – rješenje (5./6.)

```
minute = int(input('Unesite broj minuta: '))
sekunde = int(input('Unesite broj sekundi: '))
sekunde = minute*60 + sekunde
while sekunde >= 0:
    print(sekunde)
    sekunde = sekunde-1
```



odbrojavanje5-6.py

```
Python 3.4.1 Shell
File Edit Shell Debug Options Window Help
Unesite broj minuta: 1
Unesite broj sekundi: 15
75
74
73
72
71
70
69
68
67
66
65

Python 3.4.1 Shell
64

Python 3.4.1 Shell
51

Python 3.4.1 Shell
38

Python 3.4.1 Shell
25

Python 3.4.1 Shell
12

Python 3.4.1 Shell
11
10
9
8
7
6
5
4
3
2
1
0
Ln: 40 Col: 0
```



Odbrojavanje – rješenje (7./8.)

```
minute = int(input('Unesite broj minuta: '))
sekunde = int(input('Unesite broj sekundi: '))
sekunde = minute*60 + sekunde
while sekunde >= 0:
    print(sekunde, '-', sekunde//60, ':',
          sekunde%60)
    sekunde = sekunde-1
```



odbrojavanje7-8.py

Python 3.4.1 Shell	Python 3.4.1 Shell	Python 3.4.1 Shell	Python 3.4.1 Shell	Python 3.4.1 Shell	Python 3.4.1 Shell	Python 3.4.1 Shell	Python 3.4.1 Shell
Unesite broj minuta: 1 Unesite broj sekundi: 15 75 - 1 : 15 74 - 1 : 14 73 - 1 : 13 72 - 1 : 12 71 - 1 : 11 70 - 1 : 10 69 - 1 : 9 68 - 1 : 8	67 - 1 : 7 66 - 1 : 6 65 - 1 : 5 64 - 1 : 4 63 - 1 : 3 62 - 1 : 2 61 - 1 : 1 60 - 1 : 0 59 - 0 : 59 58 - 0 : 58	57 - 0 : 57 56 - 0 : 56 55 - 0 : 55 54 - 0 : 54 53 - 0 : 53 52 - 0 : 52 51 - 0 : 51 50 - 0 : 50 49 - 0 : 49 48 - 0 : 48	47 - 0 : 47 46 - 0 : 46 45 - 0 : 45 44 - 0 : 44 43 - 0 : 43 42 - 0 : 42 41 - 0 : 41 40 - 0 : 40 39 - 0 : 39 38 - 0 : 38	37 - 0 : 37 36 - 0 : 36 35 - 0 : 35 34 - 0 : 34 33 - 0 : 33 32 - 0 : 32 31 - 0 : 31 30 - 0 : 30 29 - 0 : 29 28 - 0 : 28	27 - 0 : 27 26 - 0 : 26 25 - 0 : 25 24 - 0 : 24 23 - 0 : 23 22 - 0 : 22 21 - 0 : 21 20 - 0 : 20 19 - 0 : 19 18 - 0 : 18	17 - 0 : 17 16 - 0 : 16 15 - 0 : 15 14 - 0 : 14 13 - 0 : 13 12 - 0 : 12 11 - 0 : 11 10 - 0 : 10 9 - 0 : 9 8 - 0 : 8	7 - 0 : 7 6 - 0 : 6 5 - 0 : 5 4 - 0 : 4 3 - 0 : 3 2 - 0 : 2 1 - 0 : 1 0 - 0 : 0 >>> >>>

Funkcija split



- Funkcija **split** vraća listu riječi iz zadanog niza znakova (standardni razdjelnik je praznina ' ')

```
>>> tekst = 'Liga programiranja u Pythonu'  
>>> tekst.split()  
['Liga', 'programiranja', 'u', 'Pythonu']
```

- Korisnik može kod poziva funkcije **split** postaviti razdjelnik po želji

```
>>> tekst = 'Liga programiranja u Pythonu'  
>>> tekst.split('a')  
['Lig', ' progr', 'mir', 'nj', ' u Pythonu']
```



Varijable i višestruko pridruživanje

- Višestruko pridruživanje vrijednosti varijablama

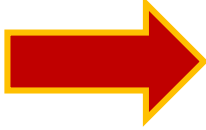

```
>>> broj1, broj2 = 1, 5
>>> print('broj1 =', broj1, 'broj2 =', broj2)
broj1 = 1 broj2 = 5
```

- Višestruko pridruživanje vrijednosti varijablama sa funkcijom **split**

```
>>> tekst = 'Liga programiranja u Pythonu'
>>> t1, t2, t3, t4 = tekst.split()
>>> print('t1', t1, 't2', t2, 't3', t3, 't4', t4)
t1 Liga t2 programiranja t3 u t4 Pythonu
```

Zadatak: Odbrojavanje split



- Nadograditi zadatak odbrojavanje tako da se vrijeme unosi u jednoj liniji u formatu mm:ss. Npr. 3:2
- Kod prethodnog zadatka Odbrojavanje: **Vrijeme**  

```
minute = int(input('Unesite broj minuta: '))
sekunde = int(input('Unesite broj sekundi: '))
sekunde = minute*60 + sekunde
```

```
while sekunde >= 0:
    print(sekunde, '-', sekunde//60, ':',
          sekunde%60)
    sekunde = sekunde-1
```



odbrojavanje7-8.py



Odbrojavanje split – rješenje

```
vrijeme = input('Unesite vrijeme: ')\nminute, sekunde = vrijeme.split(':')
```

```
minute = int(minute)\nsekunde = int(sekunde)
```

```
sekunde = minute*60 + sekunde
```

```
while sekunde >= 0:\n    print(sekunde, '-', sekunde//60, ':',\n          sekunde%60)\n    sekunde = sekunde-1
```

Uneseno vrijeme se sprema u varijable minute i sekunde, razdjelnik je ':'.

Pretvaranje minuta i sekunda iz niza znakova u cijele brojeve.



odbrojavanje_split.py

Liste



- ❑ Kako stvoriti niz od 10 cijelih brojeva?
- ❑ 1. pokušaj (10 varijabli s 10 vrijednosti):

```
>>> br1, br2, br3, br4, br5, br6, br7, br8, br9, br10 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
```

- ❑ 2. pokušaj:

```
>>> brojevi = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
>>> print(brojevi)
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Brojevi (elementi polja).

1	2	3	4	5	6	7	8	9	10
0	1	2	3	4	5	6	7	8	9

Indeksi elemenata polja.
Indeksi idu od 0 do n-1
(u ovom primjeru do 10-1 = 9).

Pristup elementima liste



- Elementima liste se može pristupiti preko indeksa elementa liste

```
>>> brojevi[0]
```

```
1
```

```
>>> brojevi[1]
```

```
2
```

```
>>> brojevi[9]
```

```
10
```

```
>>> brojevi[10]
```

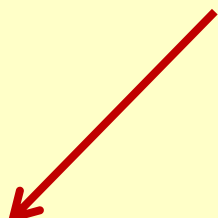
```
Traceback (most recent call last):
```

```
  File "<pyshell#89>", line 1, in <module>
```

```
    brojevi[10]
```

```
IndexError: list index out of range
```

GREŠKA: u listi **brojevi** ne postoji element s indeksom **10**, zadnji element liste s 10 brojeva ima indeks **9**.



Unos elemenata liste s tipkovnice

```
lista = [0] * 5
```

Deklaracija liste koja sadži 5 elemenata i inicijalizacija svih elemenata na vrijednost 0.

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

```
    lista[i] = float(input('Unesite broj: '))
```



lista.py

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

```
    print(i, 'element liste:', lista[i])
```

Unos brojeva s tipkovnice i spremanje u listu pomoću for petlje.

i = 5
kraj petlje

Pristup elementima liste pomoću for petlje i ispis vrijednosti na ekran.

1.1	2.2	3.3	4.4	5.5
0	1	2	3	4

```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
Unesite broj: 1.1
Unesite broj: 2.2
Unesite broj: 3.3
Unesite broj: 4.4
Unesite broj: 5.5
0 element liste: 1.1
1 element liste: 2.2
2 element liste: 3.3
3 element liste: 4.4
4 element liste: 5.5
Ln: 926 Col: 4
```

Zadatak: Najmanji element



- Napisati program koji pronalazi i ispisuje na ekran najmanji element liste cijelih brojeva. Broj elemenata liste i vrijednosti elemenata liste se unose s tipkovnice.
- Testni podaci:

<u>Ulaz</u>	<u>Ulaz</u>	<u>Ulaz</u>	<u>Ulaz</u>	<u>Ulaz</u>
5	4	5	3	4
3	32	74	489	971
15	43	84	264	453
4	17	54	534	415
8	54	86		544
9		22		
<u>Izlaz</u>	<u>Izlaz</u>	<u>Izlaz</u>	<u>Izlaz</u>	<u>Izlaz</u>
3	17	22	264	415

Vrijeme





Najmanji element – rješenje

```
n = int(input('Unesi broj elemenata liste: '))  
lista = [0] * n
```

Deklaracija liste koja sadrži **n** elemenata i inicijalizacija svih elemenata na vrijednost 0.

```
for i in range(n):  
    lista[i] = int(input('Unesite broj: '))
```

```
najmanji = lista[0]  
for i in range(n):  
    if lista[i] < najmanji:  
        najmanji = lista[i]
```

```
print('Najmanji element je:', najmanji)
```

```
Python 3.4.1 Shell  
File Edit Shell Debug Options Windows Help  
Unesi broj elemenata liste: 5  
Unesite broj: 3  
Unesite broj: 15  
Unesite broj: 4  
Unesite broj: 8  
Unesite broj: 9  
Najmanji element je: 3  
>>>
```



lista_min.py

Zadatak: Par - nepar



- ❑ Napisati program u kojem se s tipkovnice unosi broj elemenata prve liste i vrijednosti elemenata prve liste.
- ❑ Treba stvoriti drugu listu čiji će elementi imati vrijednosti elemenata prve liste umanjenih ili uvećanih za 3:
 - Neparnim brojevima umanjuje vrijednost za 3
 - Parnim brojevima uvećava vrijednost za 3
- ❑ Na kraju se ispisuju elementi obje liste jedan do drugog.



Zadatak: Par - nepar



□ Testni podaci:

<u>Ulaz</u> 5 3 15 4 8 9	<u>Ulaz</u> 4 32 43 17 54	<u>Ulaz</u> 5 74 84 54 86 22	<u>Ulaz</u> 3 489 264 534	<u>Ulaz</u> 4 971 453 415 544
<u>Izlaz</u> 3 0 15 12 4 7 8 11 9 6	<u>Izlaz</u> 32 35 43 40 17 14 54 57	<u>Izlaz</u> 74 77 84 87 54 57 86 89 22 25	<u>Izlaz</u> 489 486 264 267 534 537	<u>Izlaz</u> 971 968 453 450 415 412 544 547

Vrijeme





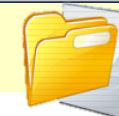
Par – nepar – rješenje

```
n = int(input('Unesi broj elemenata liste: '))  
lista1 = [0] * n  
lista2 = [0] * n
```

Deklaracija lista koje sadrže **n** elemenata i inicijalizacija svih elemenata na vrijednost 0.

```
for i in range(n):  
    lista1[i] = int(input('Unesite broj: '))  
    if lista1[i]%2 != 0:  
        lista2[i] = lista1[i] - 3  
    else:  
        lista2[i] = lista1[i] + 3  
  
for i in range(n):  
    print(lista1[i], lista2[i])
```

```
Python 3.4.1 Shell  
File Edit Shell Debug Options Windows Help  
Unesi broj elemenata liste: 4  
Unesite broj: 32  
Unesite broj: 43  
Unesite broj: 17  
Unesite broj: 54  
32 35  
43 40  
17 14  
54 57  
Ln: 27 Col: 4
```



[par-nepar.py](#)

Zadatak: Reflektori



- Na igralištu se nalazi n reflektora. Svaki reflektor može biti u stanju ugašen (broj **0**) ili upaljen (broj **1**). Na početku su svi reflektori ugašeni.
- Napisati program u kojem se unosi broj reflektora, pa zatim parovi vrijednosti broj reflektora i stanje (**0** ili **1**), sve dok se ne unese kombinacija **0 0** ili **0 1**.
- Ispisati koliko je reflektora upaljeno, a koliko ih je ugašeno

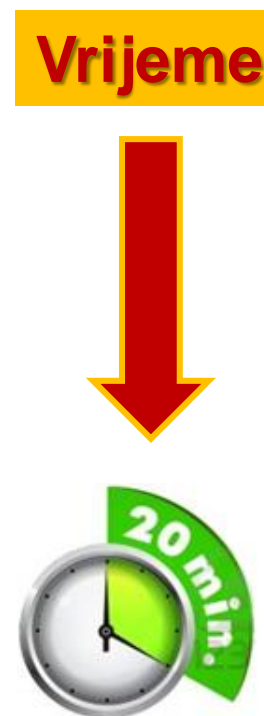


Zadatak: Reflektori



□ Testni podaci:

<u>Ulaz</u> 10 1 1 5 0 6 1 3 0 4 1 8 1 0 0	<u>Ulaz</u> 15 13 1 9 0 7 1 2 1 6 1 7 0 10 1 14 1 0 1	<u>Ulaz</u> 5 1 1 2 1 3 1 5 1 5 0 4 1 5 1 0 1	<u>Ulaz</u> 10 2 1 8 1 9 1 8 0 3 0 4 1 9 0 2 0 4 0 0 0	<u>Ulaz</u> 20 19 1 5 1 3 1 20 1 13 0 14 1 8 0 4 1 11 1 5 0 0 0
<u>Izlaz</u> 4 6	<u>Izlaz</u> 6 9	<u>Izlaz</u> 5 0	<u>Izlaz</u> 0 10	<u>Izlaz</u> 6 14





Reflektori– rješenje

```
n = int(input('Unesi broj reflektora: '))
lista = [0] * (n + 1)
rbr = 1
onoff = 0
on = 0
while rbr != 0:
    unos = input('Unesite broj reflektora i 0/1: ')
    rbr, onoff = unos.split()
    rbr = int(rbr)
    onoff = int(onoff)
    lista[rbr] = onoff
for i in range (1, n+1):
    if lista[i] == 1:
        on = on + 1
print('Upaljeno', on, 'a ugašeno', n-on, 'reflektora.')
```



reflektori.py

Ne zaboravite!

- Za 15 dana – u utorak 30.01.2024. u 17:00 –
3. kolo Lige programiranja

