

INFORMATIČKI KLUB
FUTURA

LIGA PROGRAMIRANJA



python

#2

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

Tomo Sjekavica, Informatički klub FUTURA
Dubrovnik, 16. siječnja 2016.



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 davalac licence izravno podupire Vas ili Vaše korištenje djela.



- Nekomercijalno** — Ne smijete koristiti materijal u komercijalne svrhe.



- Djeli pod istim uvjetima** — Ako remiksirate, mijenjate ili prerađujete materijal, Vaše prerade morate distribuirati pod istom licencom pod kojom je bio izvornik.

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

- 05.12.2015. – 2. radionica
- 19.12.2015. – 2. kolo Lige programiranja
- 16.01.2016. – **3. radionica**
- 30.01.2016. – **3. kolo Lige programiranja**
- 13.02.2016. – 4. radionica
- 27.02.2016. – 4. kolo Lige programiranja
- ...
- Web stranica Lige programiranja:
www.futura.com.hr/liga-programiranja-u-pythonu-2015-2016/



Ponavljanje: Metoda **split**

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

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

- Rezultat primjene metode **split** može se pohraniti u varijable

```
>>> vrijeme = '17:30:45'  
>>> hh, mm, ss = vrijeme.split(':')  
>>> print(hh, 'sati', mm, 'minuta', ss, 'sekundi')  
17 sati 30 minuta 45 sekundi
```

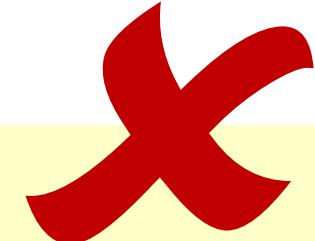
izračunati koliko
je sekundi prošlo
od početka dana



Ponavljanje: Metoda split

- Je li ovo dobar način?

```
>>> vrijeme = '17:30:45'  
>>> hh, mm, ss = vrijeme.split(':')  
>>> sekunde = (hh*60 + mm)*60 + ss
```



- Metoda split uvijek vraća nizove znakova (string)
- Brojčane varijable je potrebno pretvoriti u cijele brojeve (int) ili brojeve s pomičnom točkom (float)

```
>>> hh = int(hh)  
>>> mm = int(mm)  
>>> ss = int(ss)  
>>> sekunde = (hh*60 + mm)*60 + ss  
>>> print(sekunde)  
63045
```





Ponavljanje: String – niz znakova

□ Spajanje nizova znakova

```
>>> ime = 'Pero'  
>>> prez = 'Perić'  
>>> ucenik = ime + ' ' + prez  
>>> print(ucenik)  
Pero Perić
```

koristi se standardni operator za zbrajanje: +

svi operandi su stringovi!

□ Uvišestručenje niza znakova

```
>>> fut = 'Futura'  
>>> fut3 = fut * 3  
>>> print(fut3)  
FuturaFuturaFutura  
>>> print(fut3 * 2)  
FuturaFuturaFuturaFutura
```

koristi se standardni operator za množenje: *
-> Drugi operand je cijeli broj!



Ponavljanje: String – niz znakova

□ Operatori **in** i **not in**

```
>>> niz1 = 'Radionica program.'    >>> 'Rad' in niz1  
>>> niz2 = 'program'  
>>> 'rad' not in niz1  
True  
>>> niz2 not in niz1  
False  
>>> niz2 in niz1  
True
```

□ Dohvaćanje pojedinih znakova indeksiranjem

```
>>> fut = 'Informatički klub FUTURA'  
>>> print(fut[2])      >>> print(fut[0], fut[2:6])  
f                    I form  
>>> print(fut[2:6])    >>> print(fut[0] + fut[2:6])  
form                 Iform
```



Ponavljanje: Funkcije za stringove

- Za rad sa stringovima definirano je više funkcija - primjerice:

Funkcija	Opis djelovanja
len(s)	vraća duljinu stringa (broj znakova u stringu)
min(s)	vraća znak iz stringa koji ima najmanju kodnu vrijednost
max(s)	vraća znak iz stringa koji ima najveću kodnu vrijednost
ord(c)	vraća dekadski kod jednog znaka
chr(n)	vraća znak određen zadanim dekadskim kodom n
str(n)	vraća znakovni prikaz broja n



Ponavljanje: Metode za stringove

- Za rad sa stringovima definirano je i više metoda
- a neke kao što je primjerice **split** već smo koristili:

Metoda	Opis djelovanja
s.lower()	vraća kopiju stringa s sa svim malim slovima
s.upper()	vraća kopiju stringa s sa svim velikim slovima
s.replace(s1, s2)	vraća kopiju stringa s u kojem je svaki podniz s1 zamijenjen podnizom s2
s.find(s1)	vraća poziciju pojavljivanja podniza s1 u stringu s , ili -1 ako podniz nije pronađen
s.count(s1)	vraća broj koliko se puta podniz s1 pojavljuje u stringu s

Zadatak: Vremenska prognoza



- Unosi se grad i vrijeme u jednom retku **Vrijeme** odvojeni zarezom, a zatim u drugom retku rečenica. U rečenici zamijeniti svako pojavljivanje oznake **[grad]** s prethodno unesenim nazivom grada, a oznaku **[vrijeme]** s prethodno unesenim vremenom.
- Ispisati novu rečenicu.



ULAZ

Dubrovnik, pretežno vedro

Vrijeme za grad [grad] je [vrijeme].

IZLAZ

Vrijeme za grad Dubrovnik je pretežno vedro

ULAZ

Zagreb, kišovito

Grad [grad]: vrijeme je [vrijeme].

IZLAZ

Grad Zagreb: vrijeme je kišovito.



Vremenska prognoza – rješenje

```
ulaz = input('Unesi grad i vrijeme: ')
grad, vrijeme = ulaz.split(',')
recenica = input('Unesi rečenicu: ')
recenica = recenica.replace('[vrijeme]', 
                           vrijeme)
recenica = recenica.replace('[grad]', grad)
print(recenica)
```



vremenskaprognoza.py

```
Unesi grad i vrijeme: Dubrovnik, pretežno vedro
Unesi rečenicu: Vrijeme za grad [grad] je [vrijeme].
Vrijeme za grad Dubrovnik je pretežno vedro.
>>>
RESTART: D:\Google disk\futura\radionice\Liga_programiranja_2015\03radionica
\03-primjeri-zadaci\vremenskaprognoza.py
Unesi grad i vrijeme: Zagreb, kišovito
Unesi rečenicu: Grad [grad]: vrijeme je [vrijeme].
Grad Zagreb: vrijeme je kišovito.
```



Liste

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

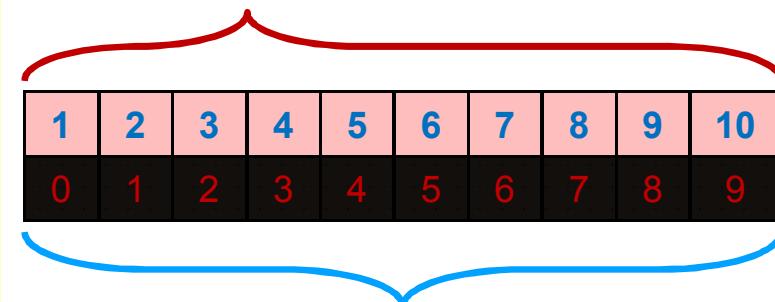
```
>>> br1, br2, br3, br4, br5, br6, br7, br8,  
br9, br10 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
```

- 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).



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  
>>> print(brojevi[4:7])  
[5, 6, 7]  
>>> 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
for i in range(5):
    lista[i] = float(input('Unesite broj: '))
for i in range(5):
    print(i, 'element liste:', lista[i])
```

deklaracija liste koja sadrži 5 elemenata i inicijalizacija svih elemenata na vrijednost 0.



lista.py

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

The screenshot shows the Python 3.4.1 Shell window. The user has entered five floating-point numbers via the input() function. These values are then indexed into the list 'lista' using a for loop, and each index-value pair is printed to the screen. The output is as follows:

```
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: Visina učenika



- Za 8 učenika se prvo unose njihova imena i visine (u cm). Nakon toga se unosi zadana visina između 140 i 200 cm.
- Ispisati u dva retka ime i visinu učenika čija je visina najbliža zadanoj visini.

Primjeri testnih podataka

ULAZ	ULAZ	ULAZ
Ana	Matej	Lorena
145	158	168
Marko	Đive	Mia
188	167	149
Pero	Stjepan	Franko
176	191	167
Ivana	Iva	Lovro
166	157	199
Petra	Pero	Nikoleta
159	178	159
Ivan	Božo	Mateo
161	188	169
Matea	Ljubica	Dora
175	168	183
Roko	Ivanka	Antun
179	167	187
167	178	176
IZLAZ	IZLAZ	IZLAZ
Ivana	Pero	Mateo
166	178	169

Vrijeme





Visina učenika – rješenje

```
ime = [''] * 8
vis = [0] * 8
razlika = 200
for i in range(8):
    ime[i]=input('Unesi ime učenika/ce: ')
    vis[i]=int(input('Unesi visinu (cm): '))
zad_vis=int(input('Unesi zadanu visinu: '))
for i in range(8):
    if abs(vis[i] - zad_vis) < razlika:
        razlika = abs(vis[i] - zad_vis)
        indeks = i
print(ime[indeks])
print(vis[indeks])
```



visina.py

Operatori za liste

- Za rad s listama koriste se isti operatori kao i za stringove:

Operator	Opis djelovanja
<code>+</code>	spajanje
<code>*</code>	uvišestručenje - jedan operand je tipa int
<code>in</code>	element je sadržan u listi
<code>not in</code>	element nije sadržan u listi

Operatori za liste

koristi se
standardni operator
za zbrajanje: +

□ Spajanje lista

```
>>> l1 = [1, 2, 3, 4, 5, 6]
>>> l2 = [-1, -2, -3]
>>> l3 = l1 + l2
>>> print(l3)
[1, 2, 3, 4, 5, 6, -1, -2, -3]
```

svi operandi su
liste!

□ Uvišestručenje lista

```
>>> l2 = [-1, -2]
>>> l4 = l2 * 2
>>> print(l4)
[-1, -2, -1, -2]
>>> print(l4 * 2)
[-1, -2, -1, -2, -1, -2, -1, -2]
```

koristi se
standardni operator
za množenje: *
-> Drugi operand je
cijeli broj!

Operatori za liste

□ Operatori **in** i **not in**

```
>>> l1 = [1, 2, 3, 4, 5, 6]
>>> l2 = [-1, -2, -3]
>>> 3 in l1
True
>>> 3 in l2
False
>>> 5 not in l1
False
>>> 5 not in l2
True
>>> [1, 2] in l1
False
```

direktan pristup
elementima liste
pomoću operatora **in**

```
>>> for broj in l1:
    print(broj, end=' ')
```

1 2 3 4 5 6

Ugrađene funkcije za liste

- Za rad s listama postoji više ugrađenih funkcija:

Funkcija	Opis djelovanja
len(a)	vraća duljinu liste a
min(a)	vraća najmanju vrijednost elementa liste a
max(a)	vraća najveću vrijednost elementa liste a
sum(a)	vraća sumu svih elemenata liste a
del(a[i])	iz liste a uklanja element s indeksom i
del(a[i:j])	iz liste a uklanja isječak koji započinje indeksom i , a završava indeksom j-1

Ugrađene funkcije za liste

```
>>> lista = [1, 2, 3, 4, 5, 6]
>>>
>>> len(lista)           >>> del(lista[2])
6                         >>> lista
>>>
>>> min(lista)          [1, 2, 4, 5, 6]
1                         >>> len(lista)
>>>
>>> max(lista)          5
6                         >>>
>>> sum(lista)           >>> del(lista[1:3])
15                        >>> print(lista)
                           [1, 5, 6]
                           >>> len(lista)
21                        3
```

Zadatak: Natjecanje



Primjeri testnih podataka

ULAZ	ULAZ
5	9
Ana 140	Matej 190
Marko 200	Đive 20
Pero 90	Franko 90
Ivana 150	Iva 160
Petra 85	Pero 120
100	Lovro 30
	Mateo 50
	Dora 160
	Božo 180
	90
IZLAZ	IZLAZ
3	6
163.333	150.0

- Unijeti broj učenika koji se natječu na školskom natjecanju iz programiranja. Za svakog učenika u istom redu unijeti ime i broj ostvarenih bodova (0-200).
- Unijeti broj bodova potreban za plasman na županijsko natjecanje.
- Maknuti s popisa sve učenike koji se nisu plasirali na županijsko natjecanje i ispisati:
 - koliko učenika se plasiralo na županijsko natjecanje
 - koliki su prosječan broj bodova ostvarili ti učenici

Vrijeme





Natjecanje – rješenje

```
n = int(input('Unesi broj učenika: '))
ime = [''] * n
bod = [0] * n
for i in range(n):
    ulaz = input('Unesi ime i broj bodova: ')
    ime[i], bod[i] = ulaz.split()
    bod[i] = int(bod[i])
prag = int(input('Prag za županijsko: '))
j = 0
while j < len(bod):
    if bod[j] < prag:
        del(ime[j])
        del(bod[j])
    else:
        j += 1
print('Broj učenika:', len(ime))
print('Prosjek bodova:', sum(bod)/len(bod))
```



natjecanje.py

Ugrađene metode za liste

- Postoji i više ugrađenih metoda za liste:

Metoda	Opis djelovanja
L1.append(x)	dodaje element x na kraju liste L1
L1.insert(i, x)	umeće element x prije i-tog elementa liste L1
L1.remove(x)	izbacuje element x (x je vrijednost elementa)
L1.pop(i)	vraća i izbacuje i-ti element iz liste L1 . Ako parametar i nije naveden vraća i izbacuje zadnji element
L1.reverse()	okreće redoslijed elemenata liste L1
L1.sort()	sortira listu L1 - od najmanje do najveće vrijednosti elementa

Ugrađene metode za liste

```
>>> L1 = [2, 4, 6, 8]
>>> L1.append(10)
>>> L1.append(12)
>>> print(L1)
[2, 4, 6, 8, 10, 12]
>>>
>>> L2 = []
>>> L2.append(1)
>>> L2.append(3)
>>> L2.append(5)
>>> print(L2)
[1, 3, 5]
>>>
```

vrijednosti 10 i 12
se dodaju na kraj
liste L1

stvaranje prazne
liste L1

vrijednosti 1, 3 i
5 se dodaju na kraj
liste L2

Ugrađene metode za liste

```
>>> L1 = [2, 4, 6, 8]
>>> L1.insert(2, 5)
>>> print(L1)
[2, 4, 5, 6, 8]
>>> L1.insert(10, 1)
>>> print(L1)
[2, 4, 5, 6, 8, 1]
>>> L1.remove(5)
>>> print(L1)
[2, 4, 6, 8, 1]
>>> L1.pop(2)
6
>>> L1.pop()
1
>>> print(L1)
[2, 4, 8]
```

vrijednost 5 se dodaje prije 3. elementa (koji ima indeks 2)

ako je indeks veći od indeksa zadnjeg elementa, dodaje se na kraj liste

izbacuje se element koji ima vrijednost 5

vraća se i izbacuje element s indeksom 2

ako nije naznačen indeks vraća se i izbacuje zadnji element

Ugrađene metode za liste

```
>>> L3 = [4, 2, 8, 6]
>>> L4 = ['b', 'e', 'u', 'a', 'p']
>>> L3.reverse()
>>> print(L3)
[6, 8, 2, 4]
>>> L4.reverse()
>>> print(L4)
['p', 'a', 'u', 'e', 'b']
>>> L3.sort()
>>> print(L3)
[2, 4, 6, 8]
>>> L4.sort()
>>> print(L4)
['a', 'b', 'e', 'p', 'u']
```

okreće se redoslijed elemenata u listama L3 i L4

listе L3 i L4 se sortiraju (od manjih prema većim vrijednostima elemenata)

Zadatak: Pozitivni brojevi



- Ana se voli igrati s pozitivnim brojevima, pa joj je potreban program u kojem se unose brojevi sve dok se ne unese negativni broj.
- Unesene pozitivne brojeve treba sortirati od većeg prema manjem.
- Od unesenih brojeva stvoriti dvije nove liste. U jednu je potrebno spremiti sve parne brojeve, a u drugu sve neparne brojeve.
- Ispisati sve tri liste.

Vrijeme



Primjer izvođenja programa

```
Unesi broj: 229
Unesi broj: 104
Unesi broj: 263
Unesi broj: 88
Unesi broj: 395
Unesi broj: 215
Unesi broj: 123
Unesi broj: -1
Lista: [395, 263, 229, 215, 123, 104, 88]
Parni brojevi: [104, 88]
Neparni brojevi: [395, 263, 229, 215, 123]
```



Pozitivni brojevi – rješenje

```
lista = []
broj = int(input('Unesi broj: '))
while broj > 0:
    lista.append(broj)
    broj = int(input('Unesi broj: '))
lista.sort()
lista.reverse()

parni, neparni = [], []
for br in lista:
    if br%2 == 0:
        parni.append(br)
    else:
        neparni.append(br)
print('Lista:', lista)
print('Parni brojevi:', parni)
print('Neparni brojevi:', neparni)
```



pozitivnibrojevi.py

Ne zaboravite!

- Za 15 dana – **u subotu 30.11.2016.** –
3. kolo Lige programiranja
- 5./6. razredi** - početak **10:00**
- 7./8. razredi** - početak **10:00**
- 3 zadatka rješavate 75 minuta**
- nemojte kasniti!**



Za kraj – Infokup 2016



<http://www.infokup.hr/>

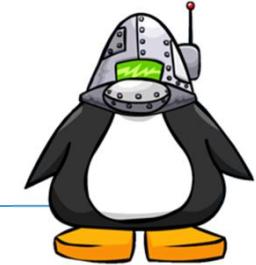
O Infokup natjecanju 1/4



- 3 razine natjecanja



- Županijsko natjecanje: 12. veljače 2016.
- Državno natjecanje: 15.-18. ožujka 2016.



O Infokup natjecanju 2/4

- Vrste natjecanja
 - Osnove informatike (teorija)
 - **Algoritmi** (to je ono što učimo na radionicama)
 - Razvoj softvera
- Školska razina natjecanja je **19.01.2016.**





O Infokup natjecanju 3/4

- Formalni mentor učenicima na natjecanju mora biti nastavnik iz iste škole
 - Trebate se svojim nastavnicima informatike javiti što prije jer su prijave otvorene do 19. siječnja 2016.
 - Ako još niste sigurni – prijavite se (uvijek stignete prespavati ili odustati ☺)
 - Znači šifra je "**Infokup – Algoritmi**"
- Web stranica natjecanja <http://www.infokup.hr/>
 - Detaljne informacije, propozicije natjecanja i prijava
 - Zadaci, rješenja i testni primjeri s prethodnih natjecanja



O Infokup natjecanju 4/4

- Za natjecanje u kategoriji **razvoj softvera** je za ovu godinu vjerojatno prekasno...
 - ... osim ako već nemate gotovo isprogramirano rješenje!
 - Za iduću godinu (2016/2017. ima dovoljno vremena za kvalitetnu pripremu)
- Ako je netko zainteresiran za sudjelovanje u ovoj vrsti natjecanja neka nam se javi osobno na kraju radionice, ili naknadno e-mailom na **futura.dubrovnik@gmail.com**
 - Kod ove vrste natjecanja imate **potpunu slobodu** izbora aplikacije koju želite napraviti te programskog jezika i razvojnih alata koje želite koristiti