

INFORMATIČKI KLUB

FUTURA

LIGA PROGRAMIRANJA



python

LIGA PROGRAMIRANJA U PYTHONU ZA OSNOVNE ŠKOLE – 5. RADIONICA

Mario Miličević, Informatički klub FUTURA
Dubrovnik, 14. veljače 2015.



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 sljedećim 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

- ...
- 14.02.2015 – **5. radionica**
- 28.02.2015 – **4. kolo Lige programiranja**
- 14.03.2015 – 6. radionica
- 28.03.2015 – 5. kolo Lige programiranja
- ...

- Web stranica Lige programiranja:
www.futura.com.hr/liga-programiranja-u-pythonu-2014-2015/



Ponavljanje: 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

```
vrijeme = input('Unesite vrijeme: ')
minute, sekunde = vrijeme.split(':')
minute = int(minute)
sekunde = int(sekunde)
```

Ponavljanje: Unos elemenata liste

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

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

```
for i in range(5):  
    lista[i] = float(input('Unesite broj: '))
```

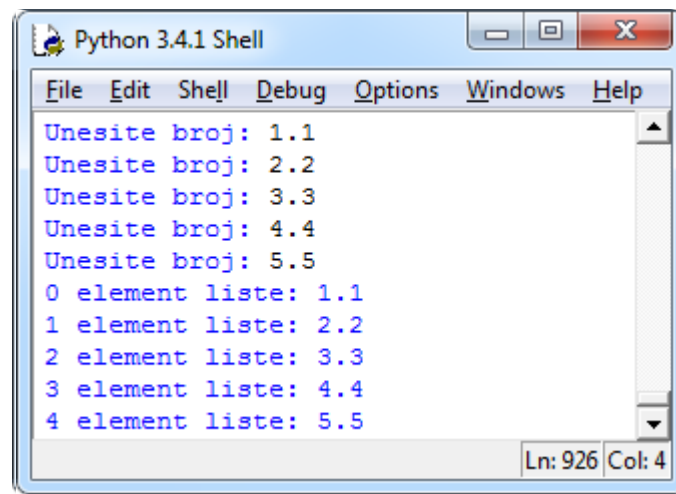
```
for i in range(5):  
    print(i, 'element liste:', lista[i])
```

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

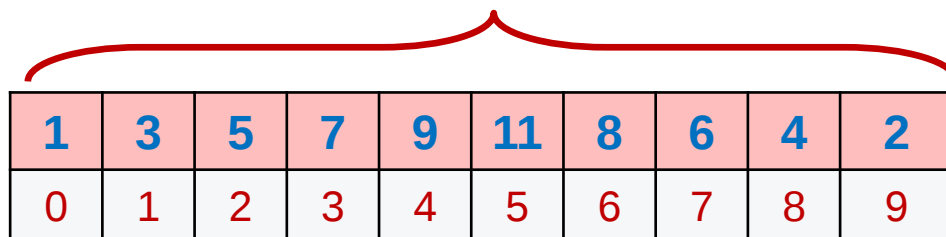


```
Python 3.4.1 Shell  
File Edit Shell Debug Options Windows Help  
Unesite broj: 1.1  
Unesite broj: 2.2  
Unesite broj: 3.3  
Unesite broj: 4.4  
Unesite broj: 5.5  
0 element liste: 1.1  
1 element liste: 2.2  
2 element liste: 3.3  
3 element liste: 4.4  
4 element liste: 5.5  
Ln: 926 Col: 4
```

Još o listama

```
>>> brojevi = [1, 3, 5, 7, 9, 11, 8, 6, 4, 2]
>>> print(brojevi)
[1, 3, 5, 7, 9, 11, 8, 6, 4, 2]
```

Brojevi (elementi polja).



The diagram shows a list structure with two rows. The top row contains the elements of the list: 1, 3, 5, 7, 9, 11, 8, 6, 4, 2. The bottom row contains the corresponding indices: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. A red bracket above the elements indicates they are the 'Brojevi' (values) of the list. A blue bracket below the indices indicates they are the 'Indeksi' (indices) of the elements.

1	3	5	7	9	11	8	6	4	2
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**).

Još o listama

```
>>> brojevi = [1, 3, 5, 7, 9, 11, 8, 6, 4, 2]
```

1	3	5	7	9	11	8	6	4	2
0	1	2	3	4	5	6	7	8	9

```
>>> print(brojevi[2])
```

```
5
```

```
>>> print(brojevi[10])
```

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

```
File "<pyshell#11>", line 1, in <module>
```

```
    print(brojevi[10])
```

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

```
>>> print(brojevi[3:6])
```

```
[7, 9, 11]
```

Još o listama

```
>>> brojevi = [1, 3, 5, 7, 9, 11, 8, 6, 4, 2]
```

1	3	5	7	9	11	8	6	4	2
0	1	2	3	4	5	6	7	8	9

```
>>> len(brojevi)
```

Dužina liste

```
10
```

```
>>> 10 in brojevi
```

```
False
```

```
>>> 11 in brojevi
```

```
True
```

```
>>> max(brojevi)
```

```
11
```

```
>>> min(brojevi)
```

```
1
```

Najveća vrijednost
u listi

Najmanja vrijednost
u listi

Još o listama

```
>>> brojevi = [1, 3, 5, 7, 9, 5, 4, 3, 2, 1]
```

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

```
>>> brojevi.index(7)  
3
```

Na kojem mjestu u
listi prvi put
javlja zadana
vrijednost

```
>>> brojevi.index(3)  
1
```

```
>>> brojevi.count(1)  
2
```

Koliko se puta
zadana vrijednost
javlja u listi

Zadatak: Hotel



- ☐ Hotel ima 12 katova, koji su označeni brojevima od 0 (prizemlje) do 11 (posljednji kat). Napisati program kojim će se bilježiti trenutni broj gostiju na nekom katu.
- ☐ Kod unosa podataka treba upisati oznaku kata, kao i broj gostiju koji su došli (prirodni broj). Na početku hotel je prazan. Podaci se unose dok se ne upiše da je na neki kat došlo 0 gostiju.
- ☐ Ispisati
 - ukupni broj gostiju u hotelu,
 - najveći broj gostiju na nekom katu,
 - na kojem se katu prvi put javlja taj broj

Zadatak: Hotel



□ Testni podaci

<u>Ulaz</u>	<u>Ulaz</u>	<u>Ulaz</u>
5 3	11 5	3 3
1 4	10 5	4 4
2 6	9 2	2 2
6 12	10 9	3 3
5 10	11 5	2 8
9 2	5 0	4 6
7 7		1 0
0 0		
<u>Izlaz</u>	<u>Izlaz</u>	<u>Izlaz</u>
Uk: 44	Uk: 26	Uk: 26
Max: 13	Max: 14	Max: 10
Kat: 5	Kat: 10	Kat: 2

Vrijeme





Hotel – rješenje

```
hotel = [0] * (11+1)
gost = 1
while gost != 0:
    unos = input('Unesite kat i broj gostiju: ')
    kat, gost = unos.split()
    kat = int(kat)
    gost = int(gost)
    hotel[kat] = hotel[kat] + gost
ukupno = 0
for i in range(0, 11+1):
    ukupno = ukupno + hotel[i]
najvise = max(hotel)
katNajvise = hotel.index(najvise)
print('Ukupno:', ukupno)
print('Najviše:', najvise)
print('Kat:', katNajvise)
```



hotel.py

Osnovni tipovi podataka u Pythonu



- **int** – cijeli broj
- **float** – broj s pomičnom točkom
- **str** – niz znakova (string)
- **bool** – logički tip podatka

Ovo smo spomenuli
na prvoj
radionici

String – niz znakova



□ Jednostruki ili dvostruki navodnici

```
>>> 'Python'  
'Python'
```

```
>>> "Python"  
'Python'
```

□ Ispis dvostrukih navodnika u nizu znakova

```
>>> 'Radionica "Python" za \'osnovne škole\''  
'Radionica "Python" za "osnovne škole"'
```

□ Ispis jednostrukih navodnika u nizu znakova

```
>>> "Radionica 'Python' za \'osnovne škole\'"  
"Radionica 'Python' za 'osnovne škole'"
```

String – niz znakova



□ Spajanje stringova

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

Koristi se
standardni operator
za zbrajanje: +

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

Svi operandi su
stringovi!

String – niz znakova



□ Uvišestručenje niza znakova

```
>>> fut = 'Futura'
```

```
>>> fut3 = fut * 3
```

```
>>> print(fut3)
```

```
FuturaFuturaFutura
```

```
>>>
```

```
>>> print(fut3*2)
```

```
FuturaFuturaFuturaFuturaFuturaFutura
```

```
>>>
```

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



String – niz znakova



□ Duljina niza

```
>>> fut = 'Informatički klub FUTURA'
>>>
>>> len(fut)
24
>>> prazno = ''
>>> len(prazno)
0
>>> prazno = ' '
>>> len(prazno)
1
>>>
```

String – niz znakova



□ Dohvaćanje pojedinih znakova indeksiranjem

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

String – niz znakova



□ Dohvaćanje pojedinih znakova indeksiranjem

```
>>> fut = 'Informatički klub FUTURA'
>>> print(fut[len(fut)])
Traceback (most recent call last):
  IndexError: string index out of range
>>> print(fut[len(fut)-1])
A
>>> print(fut[-1])
A
>>> print(fut[-1:-5])

>>> print(fut[-5:-1])
UTUR
```

Negativni indeks:
dohvat znakova od
kraja niza

Zadatak: Mobitel



- Paulin stari mobitel uzima u obzir svaki drugi utipkani znak. Promjerice:

PRaTuileae => **PRaTuileae** => **Paula**

- Hoće li Paula uspjeti nazvati željenog prijatelja?
- Ulazni podaci:
 - niz znakova
 - ime prijatelja
- Izlazni podaci:
 - ime koje će mobitel pokušati nazvati
 - DA ili NE – hoće li će Paula nazvati željenog prijatelja

Zadatak: Mobitel



□ Testni podaci

<u>Ulaz</u> ERdDos Edo	<u>Ulaz</u> Marrrei Mare	<u>Ulaz</u> KLaKt2e Kate	<u>Ulaz</u> RRiosmeioo Romeo
<u>Izlaz</u> Edo DA	<u>Izlaz</u> Mrri NE	<u>Izlaz</u> Kate DA	<u>Izlaz</u> Riseo NE

Vrijeme





Mobitel – rješenje

```
ulaz = input('Što je utipkano: ')\nime = input('Ime prijatelja: ')\nizlaz = ''\n\nfor i in range(0, len(ulaz), 2):\n    izlaz = izlaz + ulaz[i]\n\nprint(izlaz)\nif izlaz == ime:\n    print('DA')\nelse:\n    print('NE')
```



mobitel.py

String – niz znakova



□ Neke od raspoloživih metoda za stringove

```
>>> fut = 'Informatički klub FUTURA'
```

```
>>>
```

```
>>> print(fut.upper())  
INFORMATIČKI KLUB FUTURA
```

`ime_stringa.upper()`
vraća kopiju stringa sa
svim velikim slovima

```
>>>
```

```
>>> print(fut.lower())  
informatički klub futura
```

`ime_stringa.lower()`
vraća kopiju stringa sa
svim malim slovima

```
>>>
```

```
>>> print(fut.count('i'))  
2
```

`ime_stringa.count()`
broji koliko se puta
javlja zadani podniz

```
>>> print(fut.upper().count('I'))  
3
```

String – niz znakova



□ Neke od raspoloživih metoda za stringove

```
>>> fut = 'Informatički klub FUTURA'
```

```
>>>
```

```
>>> print(fut.find('mat'))
```

```
>>> 5
```

```
>>> print(fut.find('MAT'))
```

```
>>> -1
```

```
>>>
```

```
>>> print(fut.replace('UT', 'ut'))
```

```
Informatički klub FutURA
```

```
>>>
```

ime_stringa.find()
vraća poziciju prvog
pojavljivanja
zadanog podniza, ili
-1 ako podniz nije
pronađen

ime_stringa.replace()
vraća kopiju stringa
sa zamijenjenim
podnizom

Zadatak: Pravopis



- ☐ Ivo stalno ratuje s pravopisom, pa često pomoćni glagol "će" napiše kao "če".
- ☐ Treba napisati program koji će Iva ispraviti svaki put kad napiše pogrešno napiše "če".
- ☐ Ulazni podaci:
 - Ivova rečenica
- ☐ Izlazni podaci:
 - SVE OK – ako Ivo nije pogriješio
 - ... ili ...
 - broj grešaka
 - ispravno napisana rečenica

Zadatak: Pravopis



Ulaz

Program će ispraviti česte greške.

Izlaz

Broj grešaka: 1

Program će ispraviti česte greške.

Ulaz

Iva će prvo učiti, a kasnije će se igrati.

Izlaz

Broj grešaka: 2

Iva će prvo učiti, a kasnije će se igrati.

Ulaz

Na početku programi neće biti teški.

Izlaz

SVE OK

Vrijeme





Pravopis – rješenje

```
ulaz = input('Ivova rečenica: ')\n\nbroj = ulaz.count(' če ')\n\nif broj > 0:\n    print('Broj grešaka: ', broj)\n    print(ulaz.replace(' če ', ' će '))\nelse:\n    print('SVE OK')
```



pravopis.py

Zadatak: Kalkulator



- ❑ Renati se pokvarila tipkovnica, pa više ne može unijeti znakove **+**, **-**, ***** i **/**. Zato joj ne radi kalkulator.
- ❑ Treba pomoći Renati tako što će se napraviti novi program za kalkulator gdje će barem moći zbrajati i oduzimati, i to tako da će se koristiti zamjenski znakovi: **z** umjesto **+** i **o** umjesto **-**.

Npr.: **12z15** znači **12+15**, a **123o7** znači **123-7**

- ❑ Ulazni podaci:
 - izraz koji treba izračunati
- ❑ Izlazni podaci:
 - rezultat zbrajanja ili oduzimanja

Zadatak: Kalkulator



□ Testni podaci

<u>Ulaz</u> 20z210	<u>Ulaz</u> 72o32	<u>Ulaz</u> 2z1002	<u>Ulaz</u> 120z111	<u>Ulaz</u> 2112o11
<u>Izlaz</u> 230	<u>Izlaz</u> 40	<u>Izlaz</u> 1004	<u>Izlaz</u> 231	<u>Izlaz</u> 2101

Vrijeme





Kalkulator – rješenje

```
ulaz = input('Unesi izraz koji treba izračunati: ')\n\nz = ulaz.find('z')\no = ulaz.find('o')\n\nif z > 0:\n    op1 = int(ulaz[0:z])\n    op2 = int(ulaz[(z+1):len(ulaz)])\n    rez = op1 + op2\nelse:\n    op1 = int(ulaz[0:o])\n    op2 = int(ulaz[(o+1):len(ulaz)])\n    rez = op1 - op2\n\nprint('Rezultat:', rez)
```



kalkulator.py

Ne zaboravite!

- ☐ Za 15 dana – u subotu 28.02.2015 –
3. kolo Lige programiranja
- ☐ **5./6. razredi** ekipe od **3** učenika (ili barem 2)
početak **9:00**
- ☐ **7./8. razredi** ekipe od **3** učenika (ili barem 2)
početak **10:30**
- ☐ **3** zadatka rješavate **75** minuta
- ☐ nemojte kasniti!

