INFORMATIČKI KLUB

FUTURA

LIGA PROGRAMIRANJA

python #3

LIGA PROGRAMIRANJA U PYTHONU ZA OSNOVNE ŠKOLE – 5. RADIONICA

Tomo Sjekavica, Mario Miličević, *Informatički klub FUTURA*

Dubrovnik, 18. veljače 2017.

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

- …
- 18.02.2017. – **5. radionica**
- 04.03.2017. – **4. kolo Lige programiranja**
- 18.03.2017. – 6. radionica
- 01.04.2017. – 5. kolo Lige programiranja
- …

- Web stranica Lige programiranja:

www.futura.com.hr/liga-programiranja-u-pythonu-2016-2017/

Ponavljanje: metoda split

- Metoda **`split`** vraća listu riječi iz zadanog znakovnog niza.
- Standardni razdjelnik je praznina `' '`.

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

- Korisnik može kod poziva metode **`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

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

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

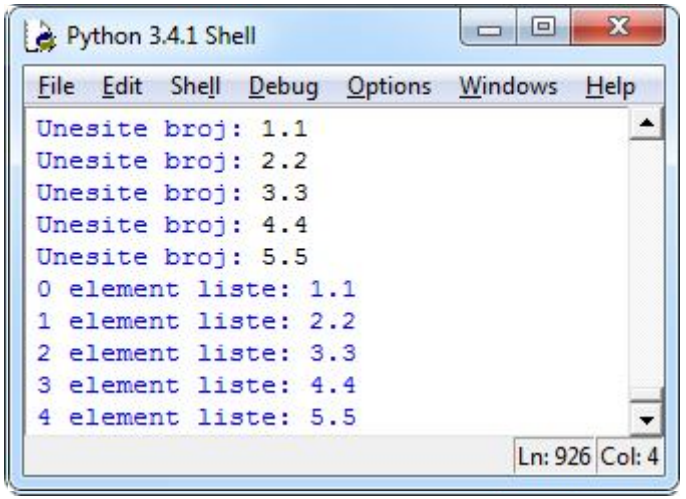
Deklaracija liste koja sadrži 5 elemenata i inicijalizacija svih elemenata na vrijednost 0.

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

i = 5
kraj petlje

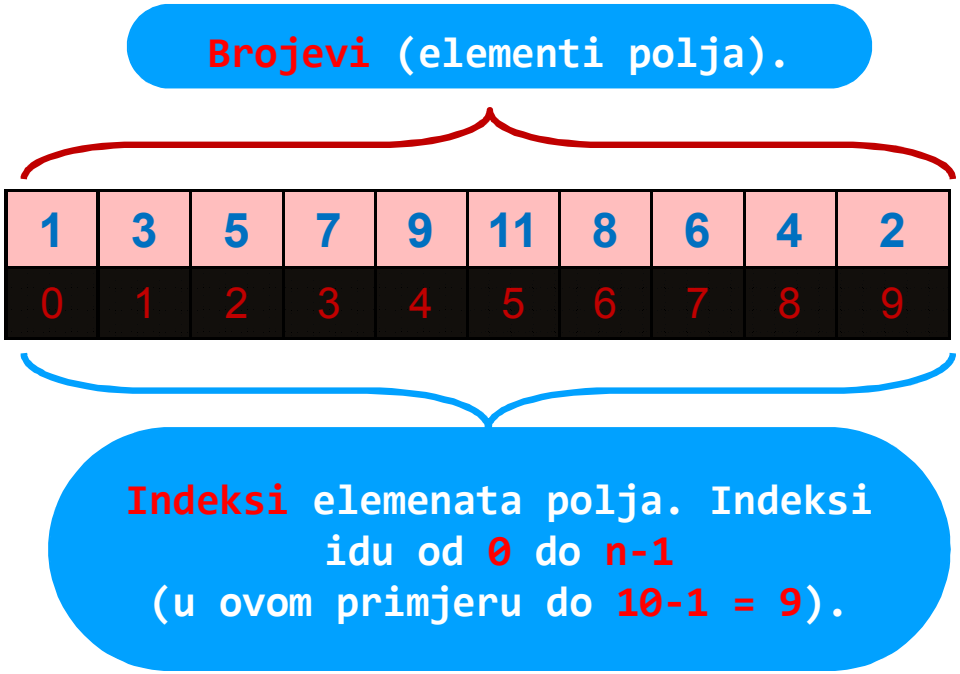
Pristup elementima liste pomoću for petlje i ispis vrijednosti na ekran.

1.1	2.2	3.3	4.4	5.5
0	1	2	3	4



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]
```



Još o listama

```
>>> brojevi = [1, 3, 5, 7, 9, 11, 8, 6, 4, 2]
>>>
>>>
>>>
>>> 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]
>>>
>>>
>>>
>>> len(brojevi)
10
>>> 10 in brojevi
False
>>> 11 in brojevi
True
>>> max(brojevi)
11
>>> min(brojevi)
1
```

Dužina liste

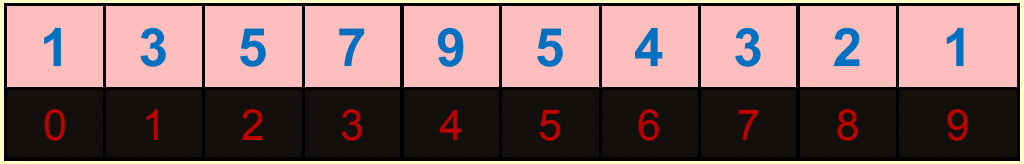
Vrijednost je element liste?

Najveća vrijednost u listi

Najmanja vrijednost u listi

Još o listama

```
>>> brojevi = [1, 3, 5, 7, 9, 5, 4, 3, 2, 1]
>>>
>>>
>>>
>>>
>>>
>>> brojevi.index(7)
3
>>>
>>> brojevi.index(3)
1
>>>
>>> brojevi.count(1)
2
```



Na kojem mjestu u listi prvi put javlja zadana vrijednost

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

Ulaz	Ulaz	Ulaz
5 3 1 4 2 6 6 12 5 10 9 2 7 7 0 0	11 5 10 5 9 2 10 9 11 5 5 0	3 3 4 4 2 2 3 3 2 8 4 6 1 0
Izlaz Ukupno: 44 Najviše: 13 Kat: 5	**Izlaz** Ukupno: 26 Najviše: 14 Kat: 10	**Izlaz** Ukupno: 26 Najviše: 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

Unos elemenata liste

- Što ako na početku programa ne znamo koliko će lista imati elemenata?
- Metoda append

```
>>> 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)
>>> print(L2)
[1, 3]
```

L1.append(x) → dodaje element x na kraju liste L1

vrijednosti 10 i 12 se dodaju na kraj liste L1

stvaranje prazne liste L2

vrijednosti 1 i 3 se dodaju na kraj liste L2

Zadatak: Brojevi

- Napisati program u kojem se s tipkovnice unose cijeli brojevi i spremaju u dvije liste. Parni brojevi se spremaju u jednu listu, a neparni u drugu listu.
- Unos se prekida kada se unese broj 0.
- Ispisati obje liste svaku u svom retku, s tim da se prvo ispisuje lista koja ima više brojeva.

- Testni podaci:

Ulaz	**Ulaz**
5 7 8 9 33 55 22 12 0	6 1 4 2 3 0
Izlaz	**Izlaz**
[5, 7, 9, 33, 55] [8, 22, 12]	[6, 4, 2] [1, 3]

Vrijeme

Brojevi – rješenje

```
parni = []
neparni = []
broj = int(input('Unesite broj: '))
while broj != 0:
    if broj%2 == 0:
        parni.append(broj)
    else:
        neparni.append(broj)
    broj=int(input('Unesite broj: '))
if len(parni) > len(neparni):
    print(parni)
    print(neparni)
else:
    print(neparni)
    print(parni)
```

brojevi.py

Osnovni tipovi podataka u Pythonu

- **`int`** – cijeli broj
- **`float`** – broj s pomičnom točkom
- **`str`** – znakovni nizz (string)
- **`bool`** – logički tip podatka

Ovo smo spomenuli na prvoj radionici

String – znakovni niz

- Jednostruki ili dvostruki navodnici

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

- Ispis dvostrukih navodnika u znakovnom nizu

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

- Ispis jednostrukih navodnika u znakovnom nizu

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

String – znakovni niz

- Spajanje znakovnih nizova (stringova)

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

Koristi se standardni operator za zbrajanje: +

Svi operandi su stringovi!

String – znakovni niz

- Uvišestručenje znakovnog niza

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

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

String – znakovni niz

- Duljina znakovnog niza

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

String – znakovni niz

- 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 – znakovni niz

- Dohvaćanje pojednih 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

Ulaz ERdDos Edo	Ulaz Marrrei Mare	Ulaz KLaKt2e Kate	Ulaz RRiosmeioo Romeo
Izlaz Edo DA	Izlaz Mrri NE	Izlaz Kate DA	Izlaz Riseo NE

Vrijeme

Mobitel – rješenje

```
ulaz = input('Što je utipkano: ')
ime = input('Ime prijatelja: ')
izlaz=''

for i in range (0, len(ulaz), 2):
    izlaz = izlaz + ulaz[i]

print(izlaz)
if izlaz == ime:
    print('DA')
else:
    print('NE')
```

mobitel.py

String – znakovni niz

- Neke od raspoloživih metoda za stringove

```
>>> fut = 'Informatički klub FUTURA'
>>>
>>> print(fut.upper())
INFORMATIČKI KLUB FUTURA
>>>
>>> print(fut.lower())
informatički klub futura
>>>
>>> print(fut.count('i'))
2
>>> print(fut.upper().count('I'))
3
```

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

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

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

String – znakovni niz

- 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 če ispraviti česte greške.

Izlaz
Broj grešaka: 1
Program će ispraviti česte greške.

Ulaz
Iva če prvo učiti, a kasnije če 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: ')

broj = ulaz.count(' će ')

if broj > 0:
    print('Broj grešaka: ', broj)
    print(ulaz.replace(' će ', ' ću '))
else:
    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

Ulaz 20z210	**Ulaz** 72o32	**Ulaz** 2z1002	**Ulaz** 120z111	**Ulaz** 2112o11
Izlaz 230	**Izlaz** 40	**Izlaz** 1004	**Izlaz** 231	**Izlaz** 2101

Vrijeme

Kalkulator – rješenje

```
ulaz = input('Unesi izraz koji treba izračunati: ')

z = ulaz.find('z')
o = ulaz.find('o')

if z > 0:
    op1 = int(ulaz[0:z])
    op2 = int(ulaz[(z+1):len(ulaz)])
    rez = op1 + op2
else:
    op1 = int(ulaz[0:o])
    op2 = int(ulaz[(o+1):len(ulaz)])
    rez = op1 - op2

print('Rezultat:', rez)
```

kalkulator.py

Funkcija eval

- Funkcija eval primljeni znakovni niz tretira kao Python izraz (izvršivi kod):

```
>>> eval('2 + 2')
4
>>> eval('2 + 2 * 2')
6
>>> x = 5
>>> eval('x % 2')
1
>>>
```

Zadatak: Kalkulator v2

- 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 to tako da će se koristiti zamjenski znak: '**z**' umjesto '**+**'.

  Npr.: **12z15** znači **12+15**
- Program treba ima minimalan broj linija koda.

```
>>> print(eval(input('Izraz: ').replace('z', '+')))
Izraz: 12z15
27
>>>
```

Ne zaboravite!

- Za 15 dana – **u subotu 04.03.2017.** –
  4. kolo Lige programiranja
- **5./6. razredi** - početak **10:00**
- **7./8. razredi** - početak **10:00**
- **3** zadatka rješavate **75** minuta
- nemojte kasniti!