INFORMATIČKI KLUB

FUTURA

LIGA PROGRAMIRANJA #3

python

LIGA PROGRAMIRANJA U PYTHONU ZA OSNOVNE ŠKOLE – 6. RADIONICA

Tomo Sjekavica, Mario Miličević, *Informatički klub FUTURA*

Dubrovnik, 18. ožujka 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

- 18.03.2017. – **6. radionica**
- 01.04.2017. – **5. kolo Lige programiranja**

  10:00 – 5./6. i 7./8. razredi

  11:30 – proglašenje najboljih!

- Web stranica Lige programiranja:

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

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)
FuturaFuturaFuturaFuturaFuturaFutura
```

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

Ponavljanje: String – niz znakova

- Duljina niza znakova

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

```
>>> prazno = ' '
>>> len(prazno)
1
```

- 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
```

Ponavljanje: String – niz znakova

- 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

Ponavljanje: 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('UTURA', 'utura'))
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: Brojevi

- Napiši program u kojem se unose cijeli brojevi tako da se svaka znamenka napiše slovima - bez razmaka. Treba ispisati koji je to broj.
- Ulazni podatak:
  - broj napisan slovima
- Izlazni podaci:
  - odgovarajući cijeli broj

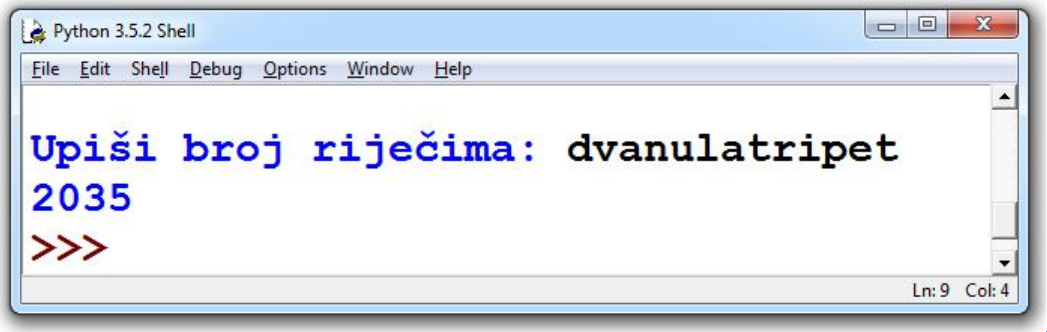
Vrijeme

- Testni podaci:

Ulaz **tridvanulanulatri**
Izlaz **32003**
Ulaz **osamosamtridva**
Izlaz **8832**

Brojevi - rješenje

```
znam = ['nula', 'jedan', 'dva', 'tri', 'četiri',
        'pet', 'šest', 'sedam', 'osam', 'devet']

broj = input('Upiši broj riječima: ')
for i in range(len(znam)):
    if znam[i] in broj:
        broj = broj.replace(znam[i], str(i))

print(broj)
```



brojevi.py

Moduli – zbirke funkcija

- Modul se prije korištenja mora uvesti s naredbom **`import naziv_modula`**
- Funkcija **`sqrt`** (korijen) iz **`math`** modula

$$\sqrt{9} = 3 \;\rightarrow\; 3 * 3 = 3^2 = 9$$

```
>>> import math
>>> sqrt(9)
Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    sqrt(9)
NameError: name 'sqrt' is not defined
>>> math.sqrt(9)
3.0
>>> math.sqrt(2)
1.4142135623730951
```

Moduli – zbirke funkcija

- Drugi način uvoza funkcija iz modula s naredbom **`from naziv_modula import funkcija1, funkcija2, …`**

```
>>> from math import sqrt, fabs
>>> sqrt(9)
3.0
>>> fabs(-3)
3.0
```

funkcija **fabs** vraća apsolutnu vrijednost broja

$$|3| = 3, \quad |-3| = 3$$

- Ako se žele uvesti sve funkcije iz nekog modula koristi se naredba:

```
>>> from math import *
>>> sqrt(9)
3.0
```

Modul `math`

- Najčešće korištene funkcije modula math:

`sqrt(x)`	korijen broja x
`fabs(x)`	apsolutna vrijednost broja x
`ceil(x)`	zaokruživanje na najmanji cijeli broj veći ili jednak broju x
`floor(x)`	zaokruživanje na najveći cijeli broj manji ili jednak broju x

- https://docs.python.org/3/library/math.html

```
>>> from math import *
>>> ceil(5.2)
6
>>> ceil(6.9)
7
```

```
>>> floor(5.2)
5
>>> floor(6.9)
6
```

Modul `math`

- Izračunati korijen sljedećeg izraza: $\sqrt{\frac{2+1}{2} + 2(3+1) + \frac{7+6}{2}}$

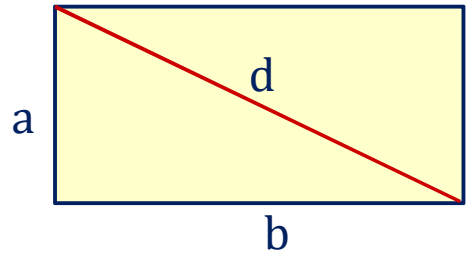
```
>>> from math import *
>>> sqrt((2+1)/2 + 2*(3+1) + (7+6)/2)
4.0
```

- Zaokružiti na veći i na manji cijeli broj sljedeći izraz: $\frac{2+1}{2} + 2(3+1)$

```
>>> ceil((2+1)/2 + 2*(3+1))
10
>>> floor((2+1)/2 + 2*(3+1))
9
```

Zadatak: Dijagonala

- Napiši program u kojem se unose duljine stranica **a** i **b** pravokutnika, te se računa duljina dijagonale pravokutnika.



$$d = \sqrt{a^2 + b^2}$$

- Ulazni podaci:
  - duljina stranica **a** i **b** u jednom retku
- Izlazni podaci:
  - duljina dijagonale

Vrijeme

- Testni podaci:

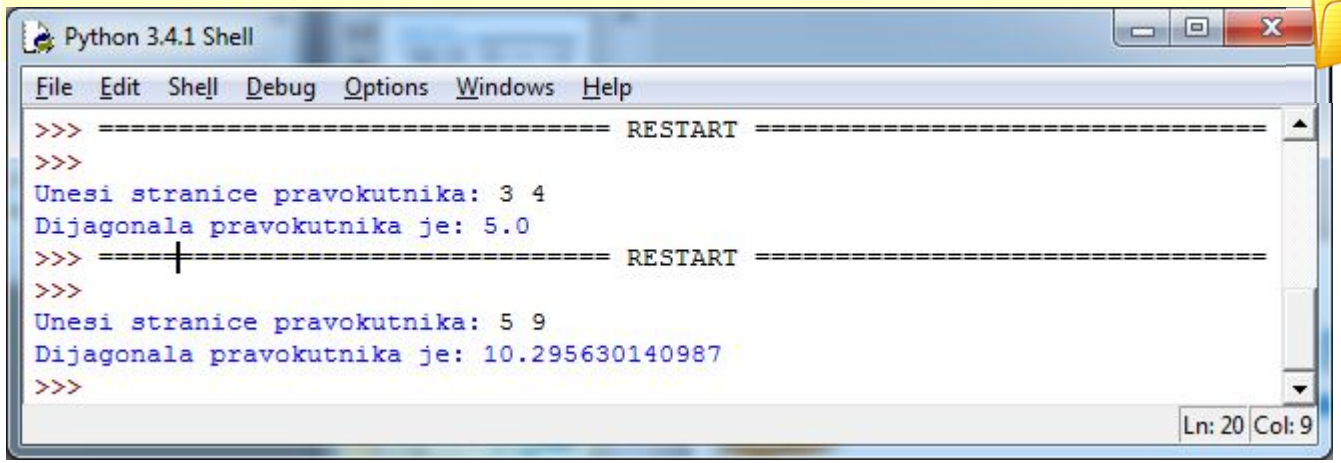
Ulaz 3 4
Izlaz 5.0
Ulaz **5 9**
Izlaz 10.295630140987

Dijagonala - rješenje

uvoz funkcije **sqrt** iz modula **math**

```
from math import sqrt
stranice = input('Unesi stranice pravokutnika: ')
a, b = stranice.split()
a = int(a)
b = int(b)
d = sqrt(a*a + b*b)
print('Dijagonala pravokutnika je:', d)
```

dijagonala.py



Modul `random`

- Funkcije za generiranje slučajnih brojeva:

`randint(a, b)`	vraća slučajni cijeli broj n koji je a <= n <= b
`random()`	vraća slučajni realni broj n koji je 0.0 <= n < 1.0
`uniform(a, b)`	vraća slučajni realni broj n koji je a <= n <= b ako je a <= b ili je b <= n <= a ako je b < a
`sample(N, k)`	vraća listu od k jedinstvenih elemenata iz seta N

- https://docs.python.org/3/library/random.html

```
>>> from random import *
>>> randint(0, 10)
5
>>> randint(0, 10)
9
```

slučajni cijeli broj u intervalu [0, 10]

Modul random

```
>>> from random import *
>>> random()
0.8460300294602602
>>> random()
0.9592937131735048
>>>
>>> uniform(0, 10)
1.594305867774457
>>> uniform(0, 10)
3.394179944212329
>>>
>>> brojevi = [i for i in range(20)]
>>> sample(brojevi, 5)
[14, 7, 16, 11, 17]
```

slučajni realni broj u intervalu [0, 1)

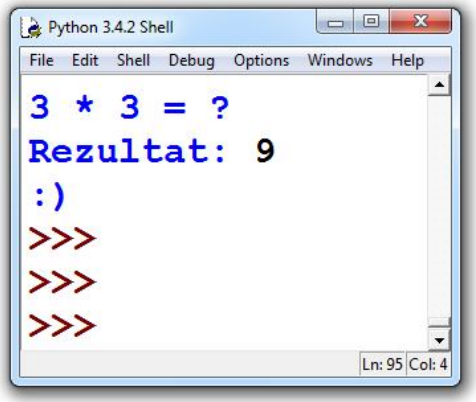
slučajni realni broj u intervalu [0, 10]

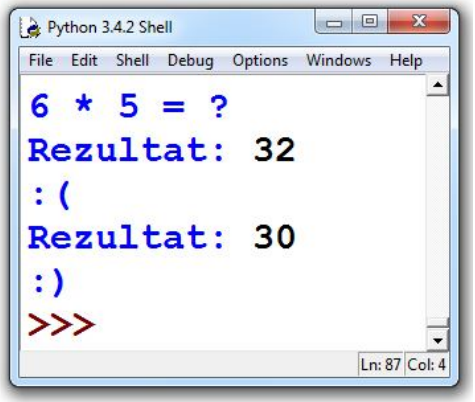
definicija liste koja sadrži 20 brojeva od 0 do 19

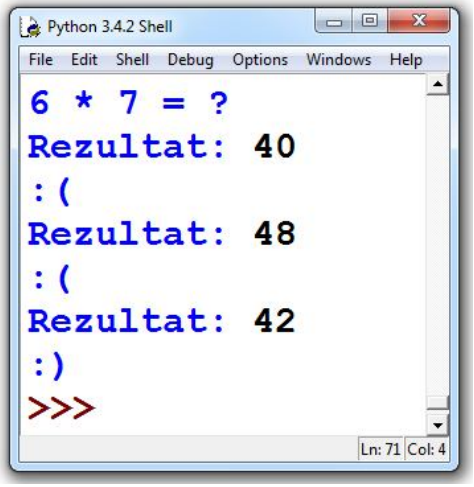
lista od 5 jedinstvenih brojeva iz liste *brojevi*

Zadatak: Tablica množenja

- Karmen mora vježbati tablicu množenja za brojeve od 1 do 10, pa joj treba napisati program koji će joj u tome pomoći.
- Program će korištenjem generatora slučajnih brojeva zadati zadatak, a Karmen treba upisati rezultat - dok taj rezultat ne bude točan.







Vrijeme

Tablica množenja

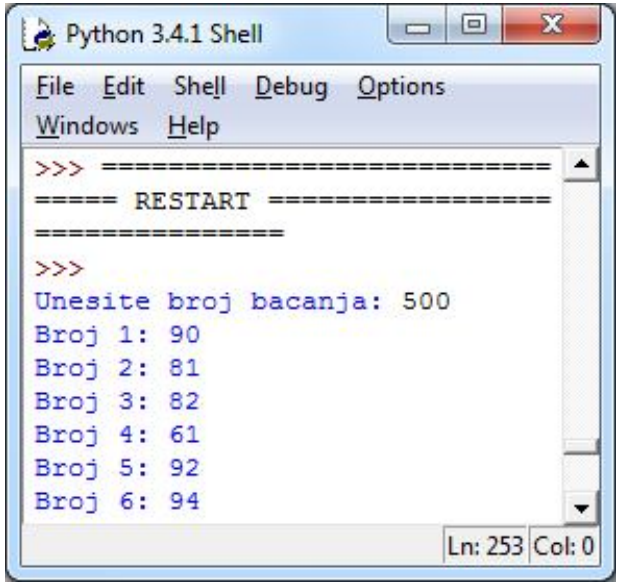
```
from random import randint
rez = False
br1 = randint(1,10)
br2 = randint(1,10)

print(br1, '*', br2, '= ?')

while not rez:
    odg = int(input('Rezultat: '))
    if odg == br1 * br2:
        print(':)')
        rez = True
    else:
        print(':(')
```

mnozenje.py

Zadatak: Kockica

- Ani je dosadilo bacati kockicu, pa joj treba program koji bi simulirao bacanje kockice. Kockica ima standardne vrijednosti od 1 do 6. Ana bi unijela broj bacanja, a program bi simulirao toliko bacanja i ispisao koliko je puta izašao koji broj.
- Ulazni podaci:
  - broj bacanja kockice
- Izlazni podaci:
  - koliko je puta izašao neki broj

Vrijeme



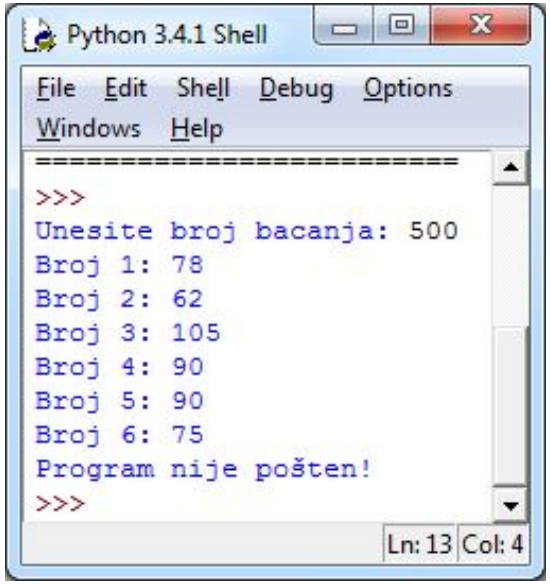
Kockica - rješenje

```
from random import randint
kockica = [0] * (6+1)

n = int(input('Unesite broj bacanja: '))

for i in range(n):
    broj = randint(1, 6)
    kockica[broj] = kockica[broj] + 1

for i in range(1, 6+1):
    print('Broj', str(i)+':', kockica[i])
```

kockica.py

Zadatak: Kockica 2

- Nadograditi prethodni zadatak tako da se ispisuje i poruka je li program pošten ili nije. Prema Ani program je pošten ako broj koji je izašao najviše puta ne odstupa više od 10% od idealne vrijednosti bacanja za neki broj.
- Ulazni podaci:
  - broj bacanja kockice
- Izlazni podaci:
  - koliko je puta izašao neki broj
  - poruka je li program pošten



Kockica 2 - rješenje

```
from random import randint
kockica = [0] * (6+1)
n = int(input('Unesite broj bacanja: '))
for i in range(n):
    broj = randint(1, 6)
    kockica[broj] = kockica[broj] + 1
for i in range(1, 6+1):
    print('Broj', str(i)+':', kockica[i])
prosjek = n / 6
najveci = max(kockica)
if najveci <= (prosjek * 1.1):
    print('Program je pošten!')
else:
    print('Program nije pošten!')
```

kockica2.py

Ne zaboravite!

- za 14 dana – **u subotu 01.04.2017.** –
  5. kolo Lige programiranja
- **5./6. i 7./8. razredi** - početak **10:00**
- **3** zadatka rješavate **60** minuta
- nemojte kasniti!
- **11:30** dodjela nagrada i priznanja najboljima