INFORMATIČKI KLUB

FUTURA

LIGA PROGRAMIRANJA #2

python

LIGA PROGRAMIRANJA U PYTHONU ZA OSNOVNE ŠKOLE – 5. RADIONICA

Mario Miličević, Informatički klub FUTURA

Dubrovnik, 12. ožujka 2016.

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 davatelj licence izravno podupire Vas ili Vaše korištenje djela.
  - **Nekomercijalno** — Ne smijete koristiti materijal u **komercijalne svrhe**.
  - **Dijeli pod istim uvjetima** — Ako remiksirate, mijenjate ili prerađujete materijal, Vaše prerade morate distribuirati pod **istom licencom** pod kojom je bio izvornik.

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

- 12.03.2016. – **5. radionica**
- 02.04.2016. – **Finale Lige programiranja**

- Web stranica Lige programiranja:

www.futura.com.hr/liga-programiranja-u-pythonu-2015-2016/

Ponavljanje: Moduli – zbirke funkcija

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

```
>>> 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
```

Ponavljanje: 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
```

Ponavljanje: 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
```

Ponavljanje: Modul `time`

- Modul **time** sadrži dosta funkcija vezanih za vrijeme, ali nama je trenutno potrebna samo funkcija **time()**:

```
>>> from time import time
>>> print(time())
1455056710.04582
>>> print(time())
1455056717.3822396
```

koliko sekundi je proteklo od početka "epohe" (tj. od 1.1.1970.)

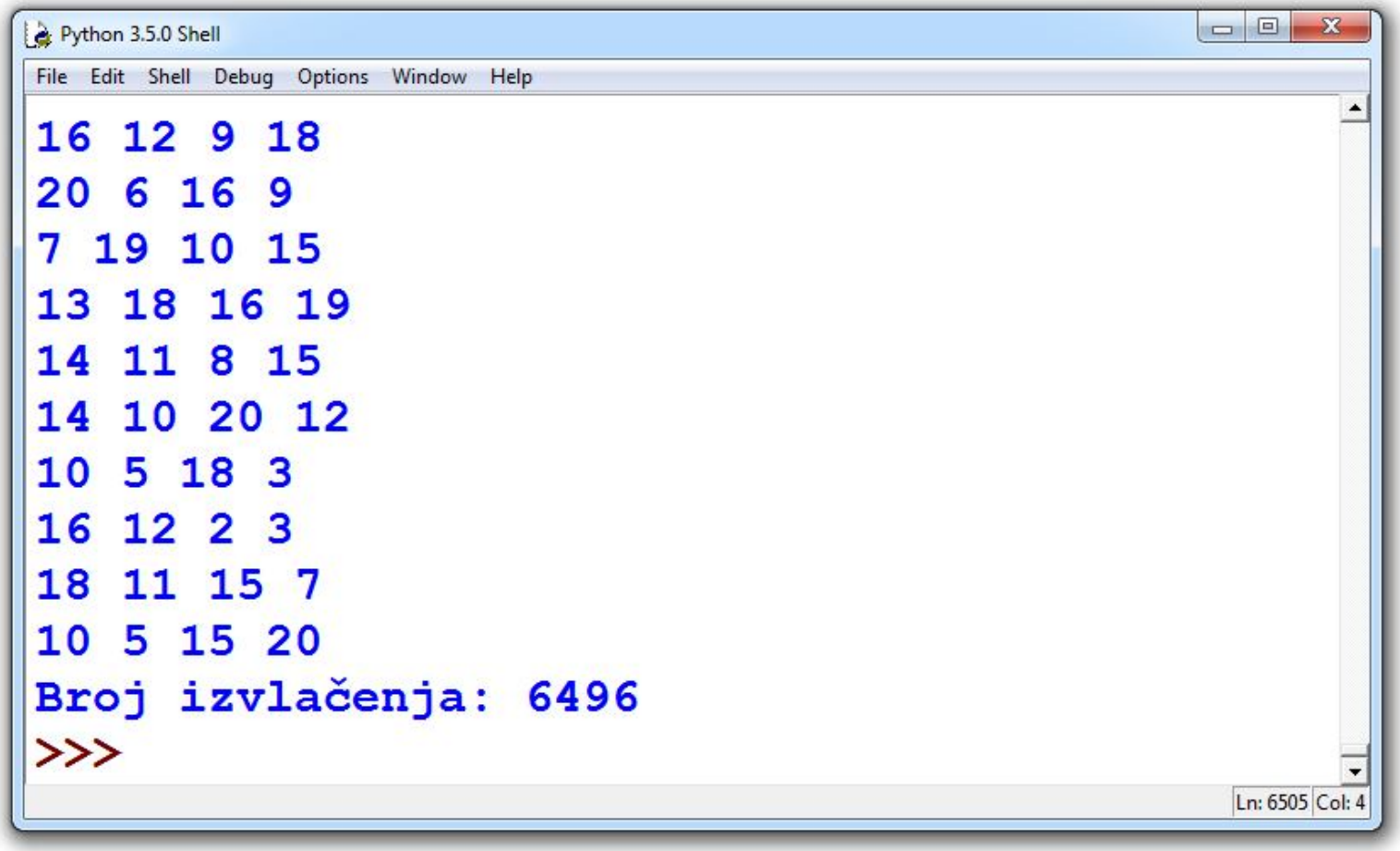
- Funkcija **time()** najviše se koristi za mjerenje proteklog vremena.
- https://docs.python.org/3/library/time.html

Zadatak: Loto

- Lora igra Loto **4 od 20**. Zanima je koliko je izvlačenja potrebno kako bi bila izvučena njezina kombinacija. Izvlačenja se simuliraju pomoću generatora slučajnih brojeva.
- Ulazni podatci:
  - Lorina kombinacija - **4** broja između **1** i **20**, u istom redu odvojeni prazninama
- Izlazni podatci:
  - Izvučene kombinacije
  - Koliko je izvlačenja bilo potrebno kako bi se izvukla Lorina kombinacija

Zadatak: Loto

- Primjer pokretanja programa, za Lorine brojeve **5 10 15 20**:



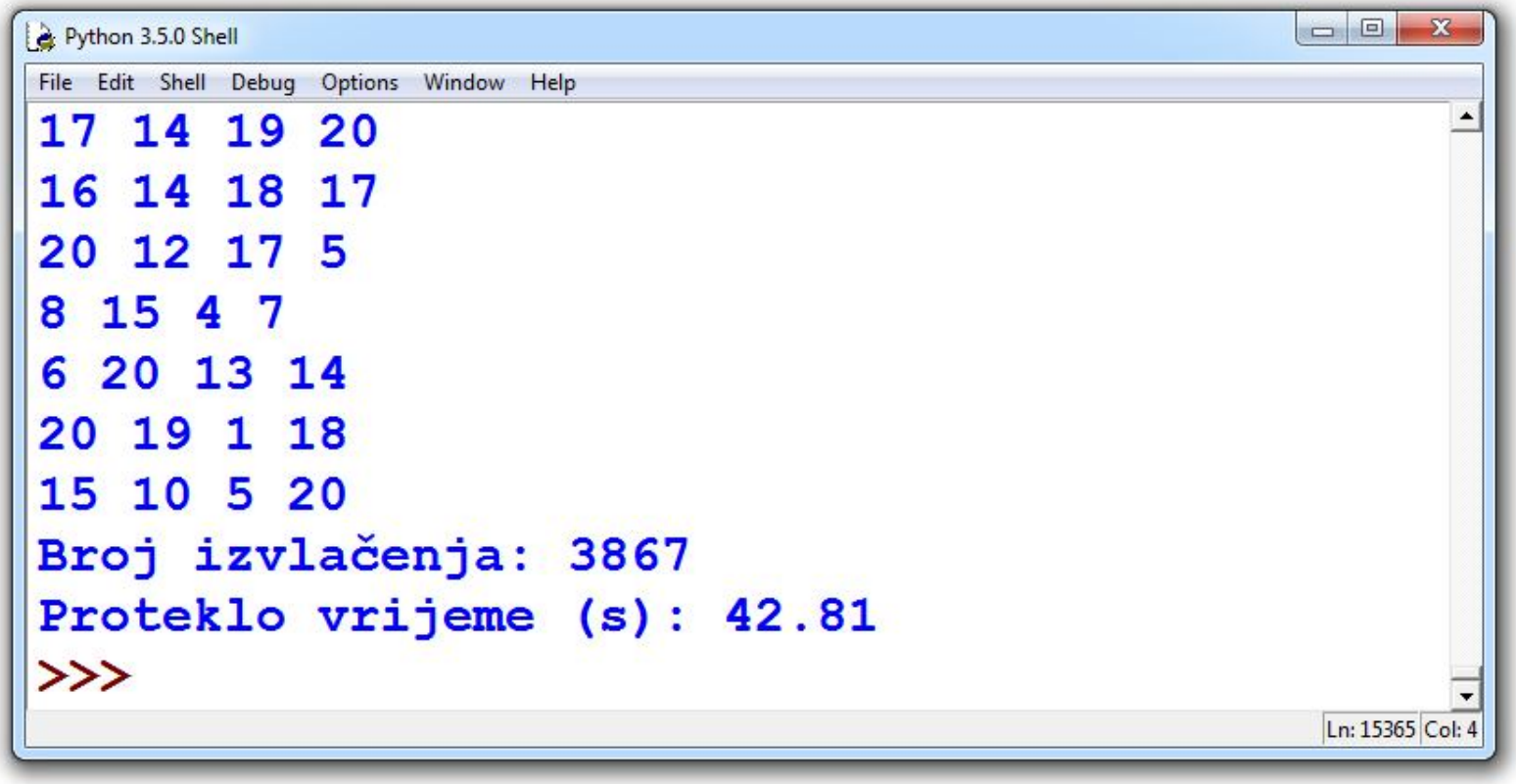
Loto - rješenje

```
from random import sample
brojevi = [i for i in range(1, 21)]
br = 0
ul = input('Lorina kombinacija - 4 broja (1-20): ')
lora = ul.split()
for i in range(len(lora)):
    lora[i]=int(lora[i])
while True:
    komb = sample(brojevi, 4)
    print()
    for i in komb:
        print(i, end=' ')
    br += 1
    lora.sort()
    komb.sort()
    if lora == komb:
        print('\nBroj izvlačenja:', br)
        break
```

loto.py

Zadatak: Loto - trajanje progr.

- Za prethodni zadatak još ispisati i koliko je vremena bilo potrebno za ispis svih generiranih izvlačenja:



Loto - trajanje progr. - rješenje

```
from random import sample
from time import time
brojevi = [i for i in range(1, 21)]
br = 0
ul = input('Lorina kombinacija - 4 broja (1-20): ')
lora = ul.split()
for i in range(len(lora)):
    lora[i]=int(lora[i])
start = time()
while True:
    komb = sample(brojevi, 4)
    print()
    for i in komb:
        print(i, end=' ')
    br += 1
    lora.sort()
    komb.sort()
    if lora == komb:
        print('\nBroj izvlačenja:', br)
        break
stop = time()
print('Proteklo vrijeme (s):', round(stop-start, 2))
```

Loto-time.py

početak i kraj mjerenja vremena

Zadatak: Križić - kružić

- Marica i Ivica igraju ***križić-kružić*** na ploči dimenzija 4x4. Marica je kružić (**O**) i uvijek igra posljednja. Međutim, nikad nije sigurna hoće li tim zadnjim potezom pobijediti.
- Treba napraviti program koji će pomoći Marici odrediti da li je odigravanjem zadnjeg kružića pobijedila!

Napomena: ne treba provjeravati dijagonale

O	X	O	X
O	O		O
X	O	X	X
X	X	O	X

hoće li Marica pobijediti?

Zadatak: Križić - kružić

- Ulazni podatci:
  - Raspored na ploči - 4 znaka u 4 reda. Koriste se velika slova '**O**' i '**X**', kao i znak '_' za prazno mjesto.
- Izlazni podatak:
  - Ispisati **DA** ili **NE** (pobjeđuje li Marica)

Ulaz	Izlaz
_XOO OXXO OXXX OOXX	DA

Ulaz	Izlaz
OOXX XXOO O_OX OXXX	NE

Ulaz	Izlaz
OXOX XX_X OOOX OXXO	NE

Ulaz	Izlaz
XXOX OXXX OO_O OXOX	DA

Vrijeme

Križić - kružić - rješenje


```
odg = 'NE'
ul = []
for i in range(4):
    ul.append(input('Unesi 4 znaka: '))
    if '_' in ul[i]:
        r = i
        s = ul[i].index('_')
if ul[r].count('O') == 3:
    odg = 'DA'
if (ul[0][s]+ul[1][s]+ul[2][s]+ul[3][s]).count('O')==3:
    odg = 'DA'

print(odg)
```

krizic-kruzic.py

Zadatak: Prosti brojevi

- Učitati prirodni broj **N**
- Provjeriti i ispisati da li su prosti sljedeći brojevi:
  - **N - 1**
  - **N**
  - **N + 1**

Ulaz	Izlaz
6	5 je prost 6 nije prost 7 je prost
150	149 je prost 150 nije prost 151 je prost
50	49 je prost 50 nije prost 51 nije prost

Prosti brojevi - rješenje


```
from math import sqrt, ceil
N = int(input('Unesi N: '))

prost = 'je'
for i in range(2, ceil(sqrt(N-1))+1):
    if (N-1) % i == 0:
        prost = 'nije'
        break
print(N-1, prost, 'prost')

... nastavak
```

prosti.py

Prosti brojevi - rješenje


```
... nastavak

prost = 'je'
for i in range(2, ceil(sqrt(N))+1):
    if N % i == 0:
        prost = 'nije'
        break
print(N, prost, 'prost')

prost = 'je'
for i in range(2, ceil(sqrt(N+1))+1):
    if (N+1) % i == 0:
        prost = 'nije'
        break
print(N+1, prost, 'prost')
```

Problem: ponavljanje koda


```
from math import sqrt, ceil
N = int(input('Unesi N: '))

prost = 'je'
for i in range(2, ceil(sqrt(N-1))+1):
    if (N-1) % i == 0:
        prost = 'nije'
        break
print(N-1, prost, 'prost')

prost = 'je'
for i in range(2, ceil(sqrt(N))+1):
    if N % i == 0:
        prost = 'nije'
        break
print(N, prost, 'prost')

prost = 'je'
for i in range(2, ceil(sqrt(N+1))+1):
    if (N+1) % i == 0:
        prost = 'nije'
        break
print(N+1, prost, 'prost')
```

Definiranje vlastitih funkcija

- Programski kod koji se ponavlja najbolje je napisati kao funkciju.
- U Pythonu, kao i u drugim programskim jezicima, moguće je definirati vlastite funkcije:

```
def ime_funkcije(popis parametara)
    naredba1_1
    ...
    naredba1_n
    return vrijednost
```

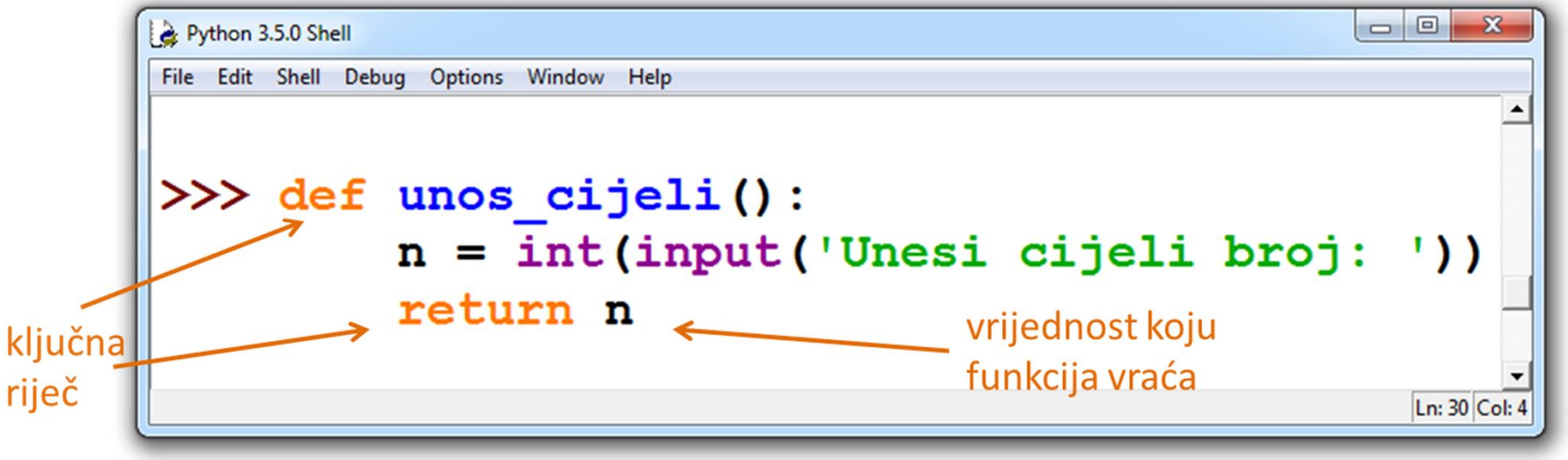
Definiranje vlastitih funkcija

```
def ime_funkcije(popis parametara)
    naredba1_1
    ...
    naredba1_n
    return vrijednost
```

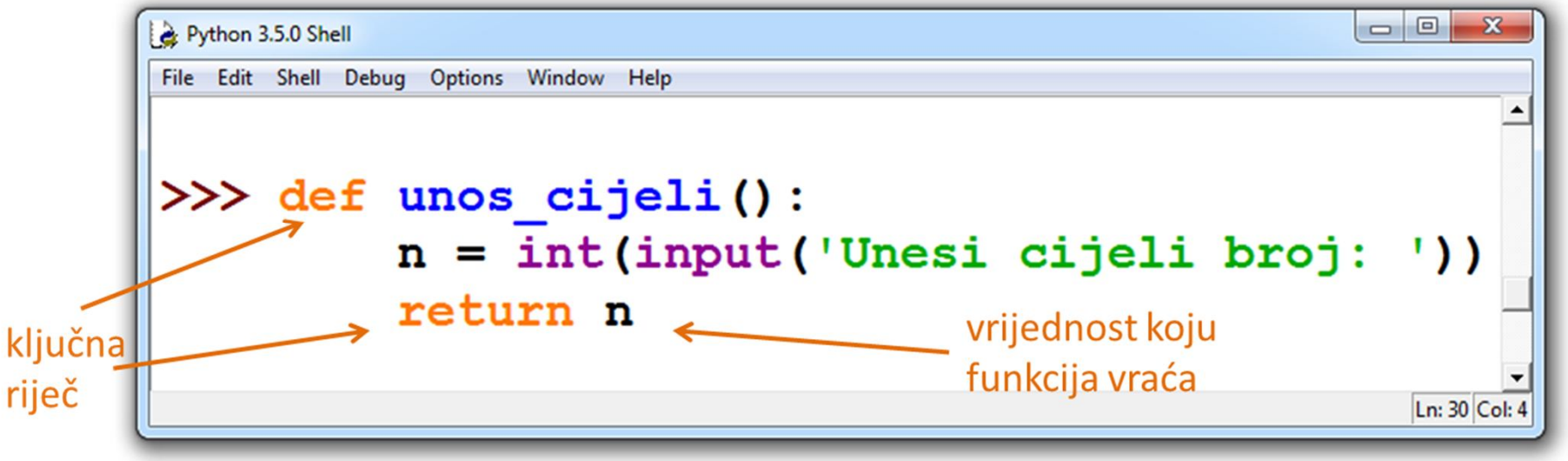
- Parametri su vrijednosti koje se predaju kod poziva funkcije. Funkcija ne mora imati ulazne parametre - u tom slučaju se nakon imena funkcije pišu prazne zagrade ().
- Naredbom return funkcija vraća jednu ili više vrijednosti. Ako funkcija ne vraća vrijednost, onda se iza naredbe return ništa ne piše.

Definiranje vlastitih funkcija

- Primjer: kako se često ponavlja kod za unos cijelih brojeva, onda ćemo napraviti odgovarajuću funkciju.
- Funkciju je moguće definirati i u ***Idle*** sučelju:



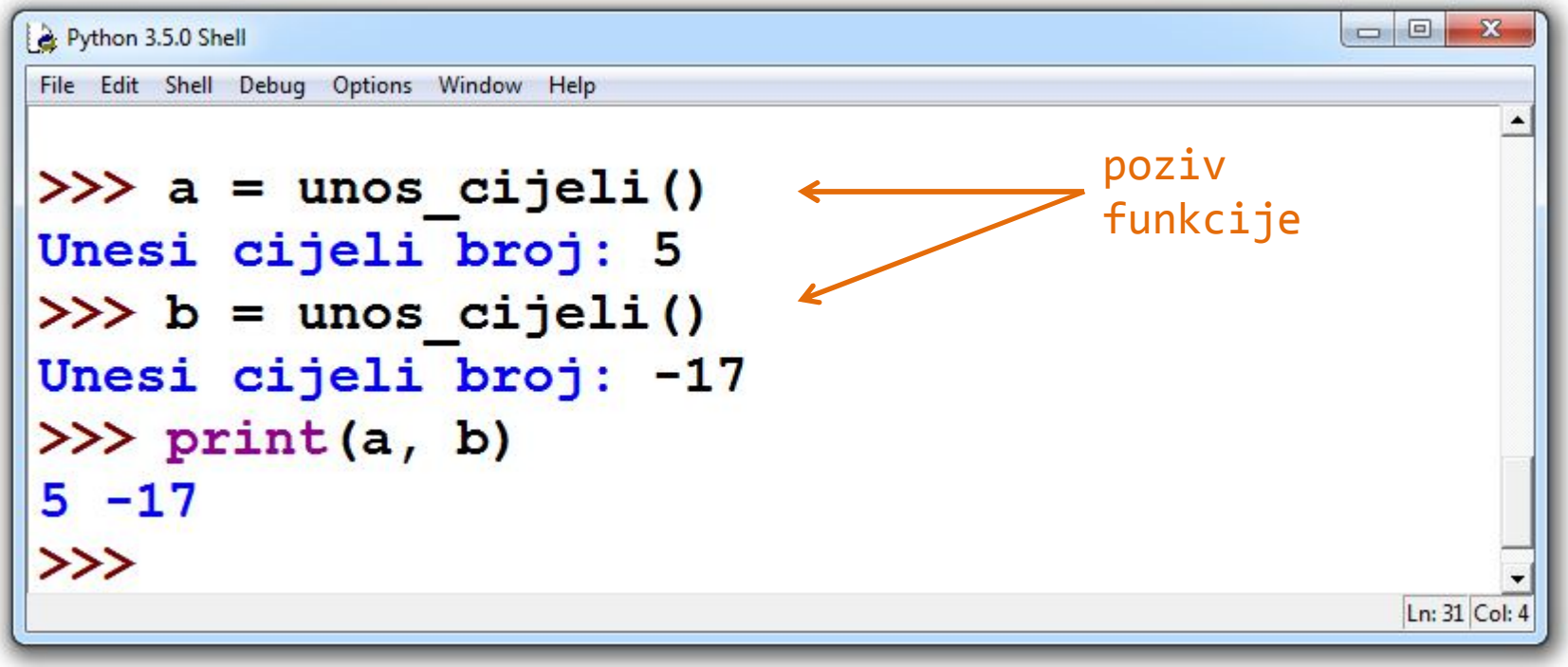
Definiranje vlastitih funkcija



- Funkcija se zove **unos_cijeli** i nema ulaznih parametara.
- Funkcija **unos_cijeli vraća** cijeli broj koji je unešen pomoću tipkovnice.

Definiranje vlastitih funkcija

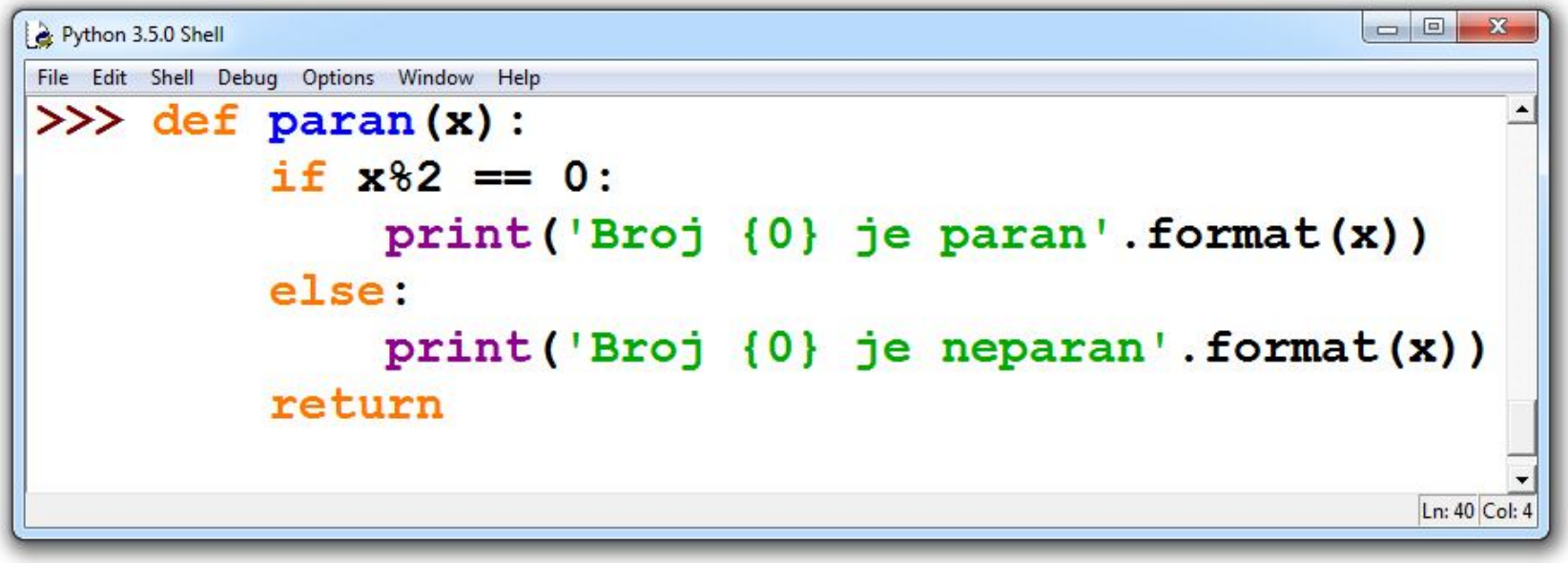
- Poziv funkcije **unos_cijeli**:



- Varijabla **a** pokazuje na prvu vrijednost koju vraća funkcija, a varijabla **b** na drugu vrijednost.
- Uočiti da se u definiciji funkcije koristila varijabla **n**.

Definiranje vlastitih funkcija

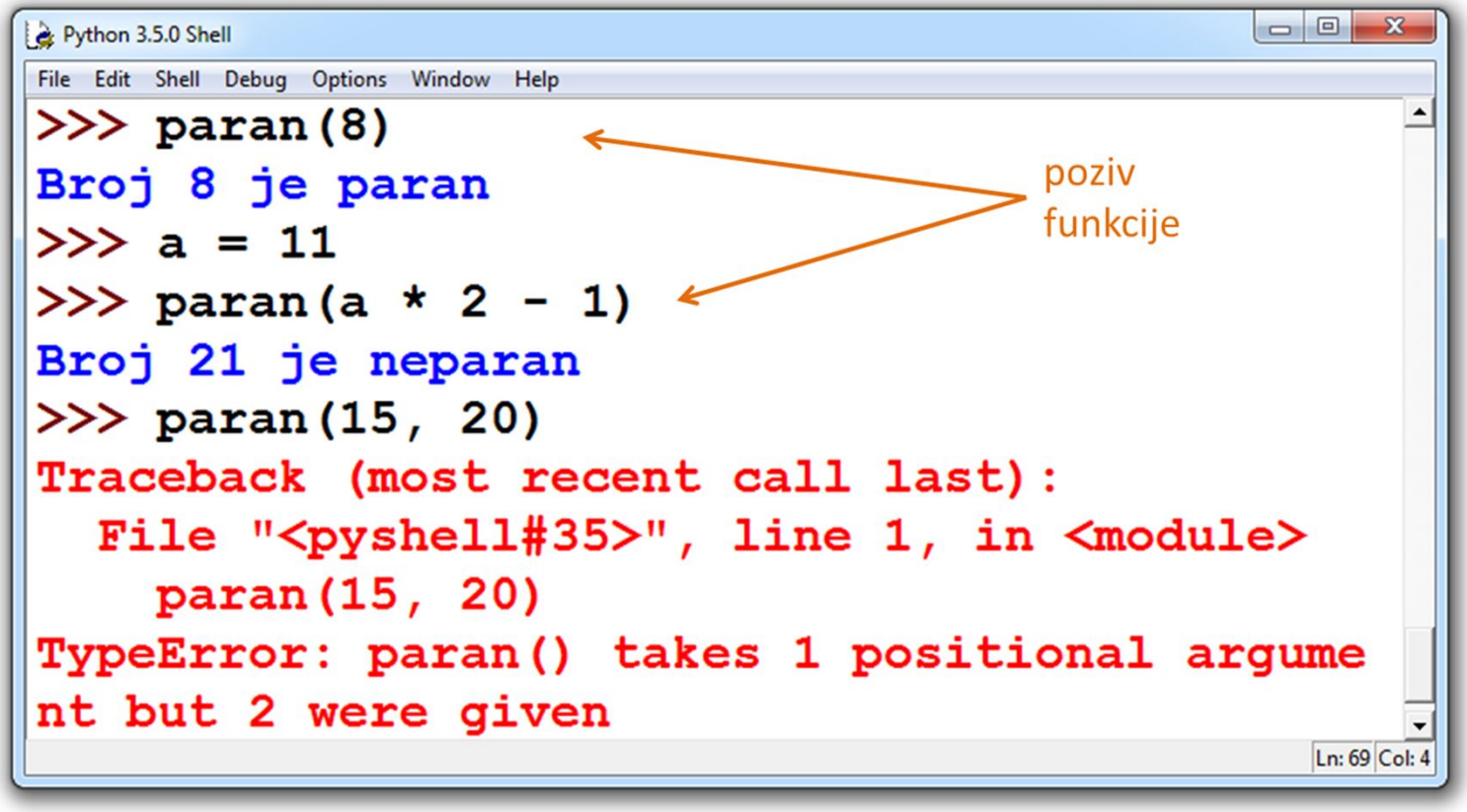
- Sljedeća funkcija provjeravat će je li broj **paran**:



- Funkcija se zove paran i ima jedan ulazni parametar **x** - broj za kojeg će se provjeriti da li paran.
- Funkcija paran ne vraća vrijednost u program iz kojeg je pozvana, već samo ispiše poruku.

Definiranje vlastitih funkcija

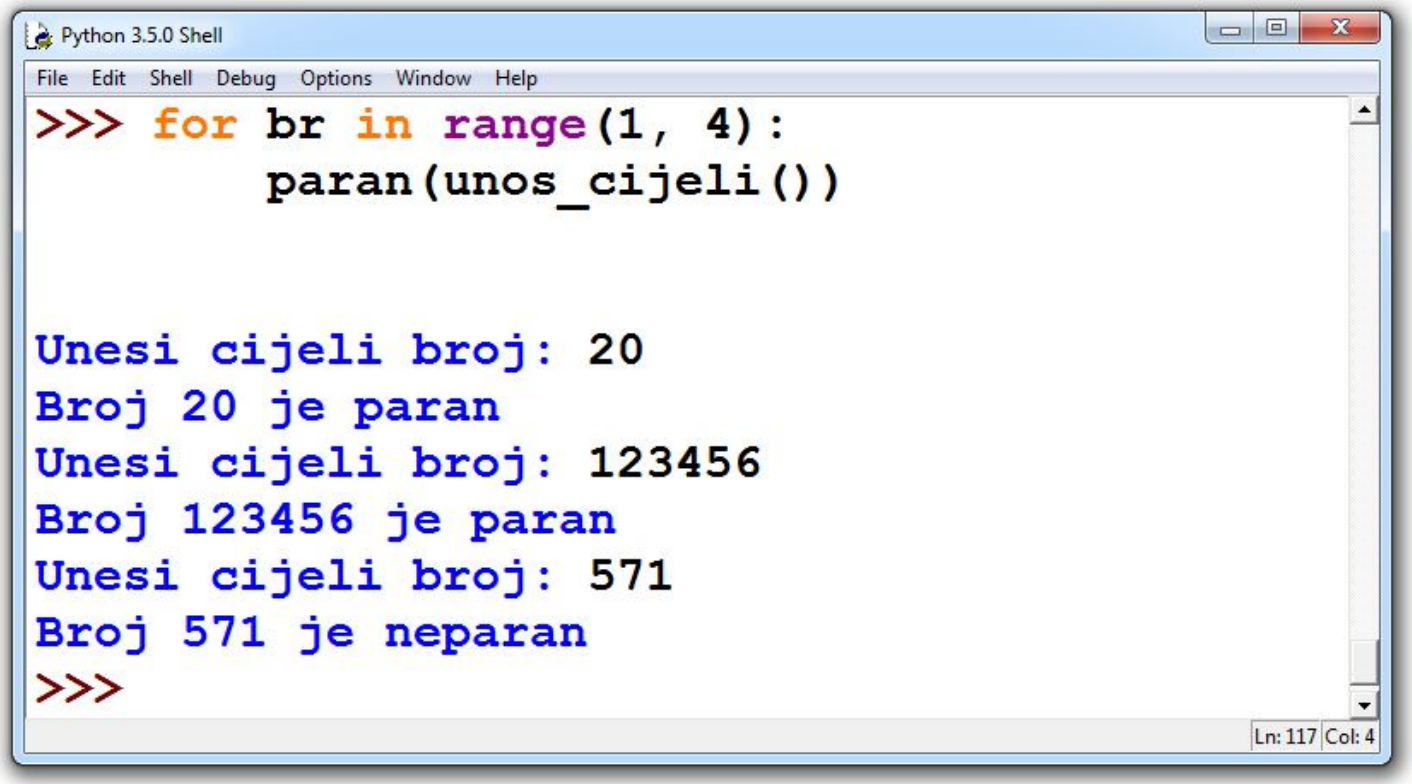
- Poziv funkcije **paran**:



- U slučaju neispravnog broja parametara dolazi do **pogreške**.

Definiranje vlastitih funkcija

- Jedna funkcija može biti parametar u pozivu druge funkcije (naravno ako odgovara tip podataka koji funkcija vraća):



Zadatak: Prosti brojevi

- Učitati prirodni broj **N**
- Provjeriti i ispisati da li su prosti sljedeći brojevi:
  - **N - 1**
  - **N**
  - **N + 1**
- Za provjeru da li je broj prost napisati funkciju **prost(x)**!

Ulaz	Izlaz
6	5 je prost 6 nije prost 7 je prost
Ulaz 150	Izlaz 149 je prost 150 nije prost 151 je prost
Ulaz 50	Izlaz 49 je prost 50 nije prost 51 nije prost

Prosti brojevi - rješenje


```
from math import sqrt, ceil

def prost(x):
    prost = 'je'
    for i in range(2, ceil(sqrt(x))+1):
        if x % i == 0:
            prost = 'nije'
            break
    return prost

N = int(input('Unesi N: '))
print(N-1, prost(N-1), 'prost')
print(N, prost(N), 'prost')
print(N+1, prost(N+1), 'prost')
```

funkcija

glavni program

Prosti-fun.py

Glavni program kao funkcija

Prosti-main.py

```
from math import sqrt, ceil

def prost(x):
    prost = 'je'
    for i in range(2, ceil(sqrt(x))+1):
        if x % i == 0:
            prost = 'nije'
            break
    return prost

def main():
    N = int(input('Unesi N: '))
    print(N-1, prost(N-1), 'prost')
    print(N, prost(N), 'prost')
    print(N+1, prost(N+1), 'prost')

main()
```

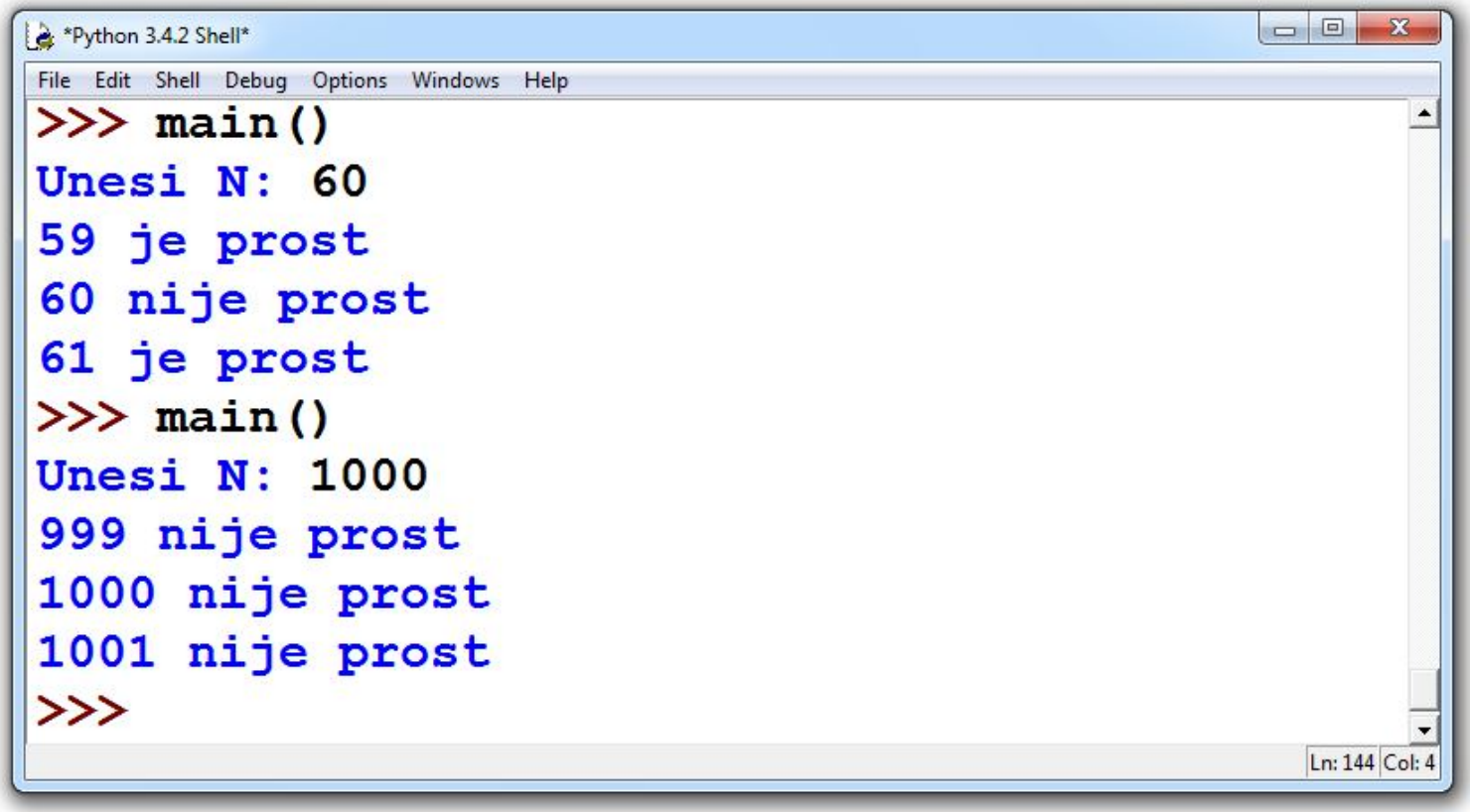
funkcija

glavni program kao funkcija

pokretanje glavnog programa

Glavni program kao funkcija

- Dodatna prednost ovakvog pristupa je što se sada u ***Idle*** sučelju program može više puta pokretati pozivom funkcije **main()**:



Zadatak: Inicijali

- Napisati **funkciju inic(a)** koja će za bilo koji broj imena/prezimena vraćati **inicijale**.
- U glavnom programu ispisati inicijale za tri osobe s različitim brojem imena/prezimena – u skladu s testnim podatcima:

Ulaz	Izlaz
Marko Karatomislavić Ana Iva Lovrić Tea Ita Anić Perić	M.K. A.I.L. T.I.A.P.
Ulaz	Izlaz
Luka Modrić Pave Župan Rusković Anna Maria Smith Wesson	L.M. P.Ž.R. A.M.S.W.

Inicijali - rješenje

inicijali.py

```
def inic(a):
    rez = ''
    lst = a.split()
    for i in lst:
        rez = rez + i[0] + '.'
    return rez

def main():
    ul = input('Ime i prezime: ')
    print(inic(ul))
    ul = input('Ime i dva prezimena: ')
    print(inic(ul))
    ul = input('Dva imena i dva prezimena: ')
    print(inic(ul))

main()
```

funkcija

glavni program kao funkcija

pokretanje glavnog programa

Zadatak: Dužina riječi

- Napisati **funkciju len_hr(s)** koja ispravno broji dužinu riječi na hrvatskom jeziku!
- U glavnom programu u petlji unositi riječi i ispisivati dužinu unešene riječi - dok se ne unese prazni string.

Ulaz	Izlaz
Dživo	4
Programiranje	12
Ljubav	5
Njegovanje	8
Polje	4
Ljuljanje	6

Dužina riječi - rješenje


```
def len_hr(s):
    lj = s.upper().count('LJ')
    nj = s.upper().count('NJ')
    dz = s.upper().count('DŽ')
    return len(s) - lj - nj - dz

def main():
    ul = '?'
    while ul != '':
        ul = input('Unesi riječ: ')
        print('Dužina riječi:', len_hr(ul))

main()
```

funkcija

glavni program kao funkcija

pokretanje glavnog programa

len_hr.py

Ne zaboravite!

- Za 21 dan – **u subotu** **02.04.2016.**
  Finale Lige programiranja
- **5./6. razredi** - početak **10:00**
- **7./8. razredi** - početak **10:00**
- **3** zadatka rješavate **60** minuta
- nemojte kasniti!