

INFORMATIČKI KLUB

FUTURA

LIGA PROGRAMIRANJA



LIGA PROGRAMIRANJA U PYTHONU ZA OSNOVNE ŠKOLE – 1. RADIONICA

Tomo Sjekavica, Informatički klub FUTURA
Dubrovnik, 18. listopada 2014.



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 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

- ☐ 18.10.2014. – 1. radionica
- ☐ 08.11.2014. – 2. radionica
- ☐ 22.11.2014. – 1. kolo Lige programiranja
- ☐ 06.12.2014. – 3. radionica
- ☐ 20.12.2014. – 2. kolo Lige programiranja
- ☐ termini u 2015. godini će biti naknadno određeni
- ☐ Web stranica Lige programiranja:
www.futura.com.hr/liga-programiranja-u-pythonu-2014-2015/

Pravila Lige programiranja

- ☐ Ekipno natjecanje škola **5./6. razredi** i **7./8. razredi**
- ☐ Ekipu škole **5./6. razredi** čine 2 ili 3 učenika iz 5. ili 6. razreda
- ☐ Ekipu škole **7./8. razredi** čine 2 ili 3 učenika iz 7. ili 8. razreda
- ☐ Za rezultate ekipa se uzimaju u obzir bodovi 2 najbolja učenika te ekipe za svako kolo lige
- ☐ Nagrade najboljim ekipama i najboljim pojedincima u kategorijama 5./6. i 7./8. razredi

Programski jezik Python

- ☐ www.python.org
- ☐ Open source program
- ☐ Besplatni program
- ☐ Jednostavna sintaksa
- ☐ Autor: Guido van Rossum - kraj 1989. godine
- ☐ **Python 1.0** – siječanj 1994. godine
- ☐ **Python 2.0** – listopad 2000. godine
- ☐ **Python 3.0** – prosinac 2008. godine
- ☐ Zadnje verzije: **Python 2.7.8** i **Python 3.4.1**



Primjena Pythona



- ☐ Web programiranje:
 - Django, Pyramid, Bottle, Tornado, Flask, web2py
- ☐ Razvoj samostojeće programske potpore:
 - wxPython, tkInter, PyGtk, PyQt
- ☐ Znanost i numeričke simulacije:
 - SciPy, Pandas, Ipython
- ☐ Razvoj softvera:
 - Buildbot, Trac, Roundup, Scons, Apache Gump
- ☐ Administracija sustava:
 - Ansible, Salt, OpenStack

Primjena Pythona

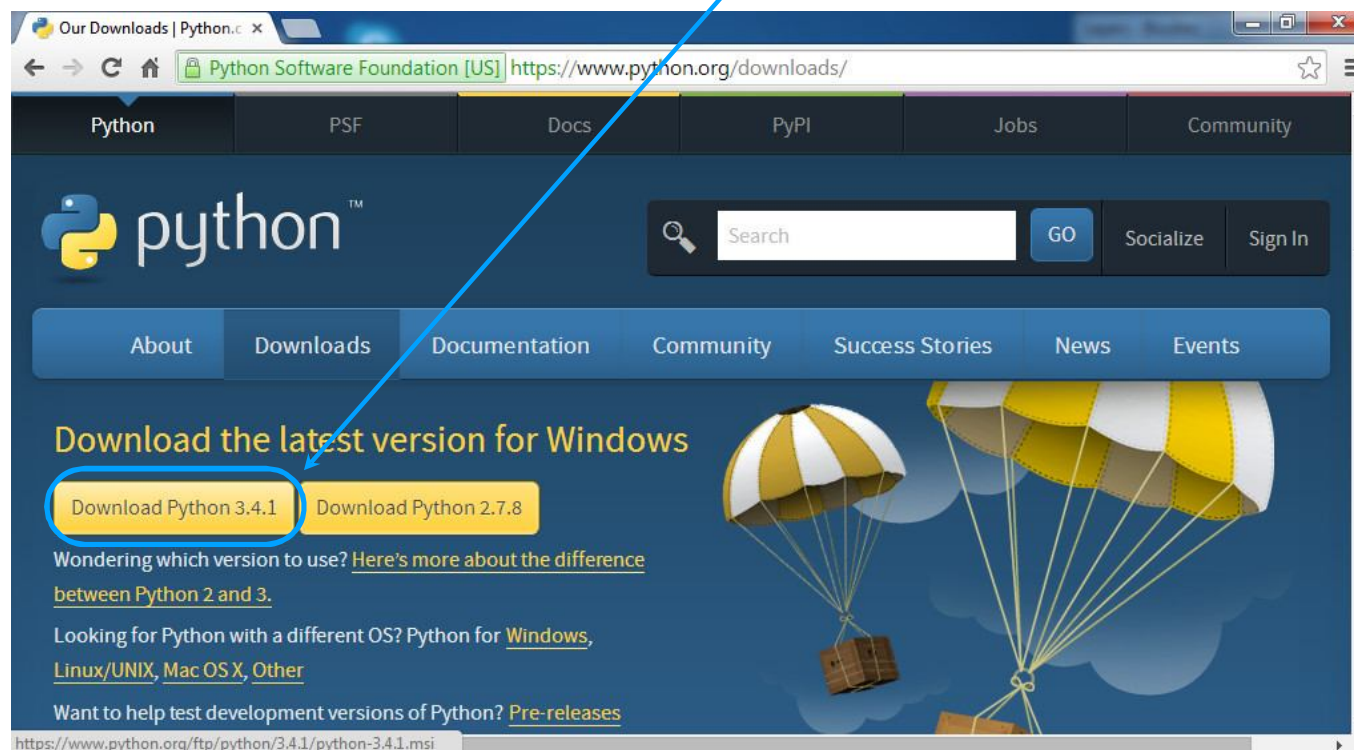


- **Disqus** – www.disqus.com
- **Lanyrd** – www.lanyrd.com
- **Pinterest** – ww.pinterest.com
- **Instagram** – www.instagram.com
- **Google Inc.** – www.google.com
- **YouTube** – www.youtube.com
- **Mozilla Support** – support.mozilla.org
- **NASA** – www.nasa.gov
- **New York Times** - www.nytimes.com
- **The Guardian** - www.theguardian.com/uk
- ...

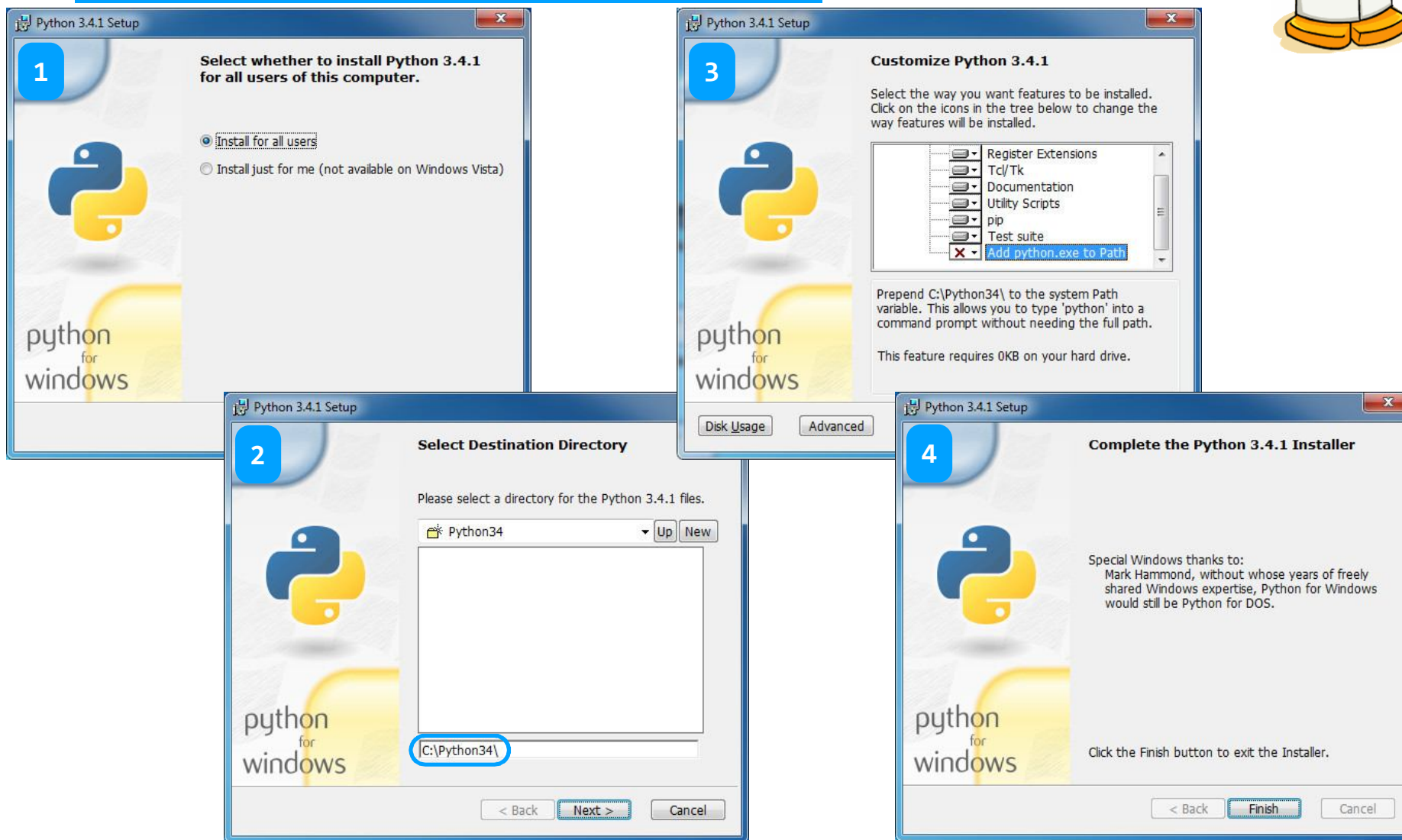
Instalacija Pythona



- ❑ www.python.org/downloads
- ❑ Koristiti ćemo zadnju verziju za Windows operacijski sustav – **Python 3.4.1**



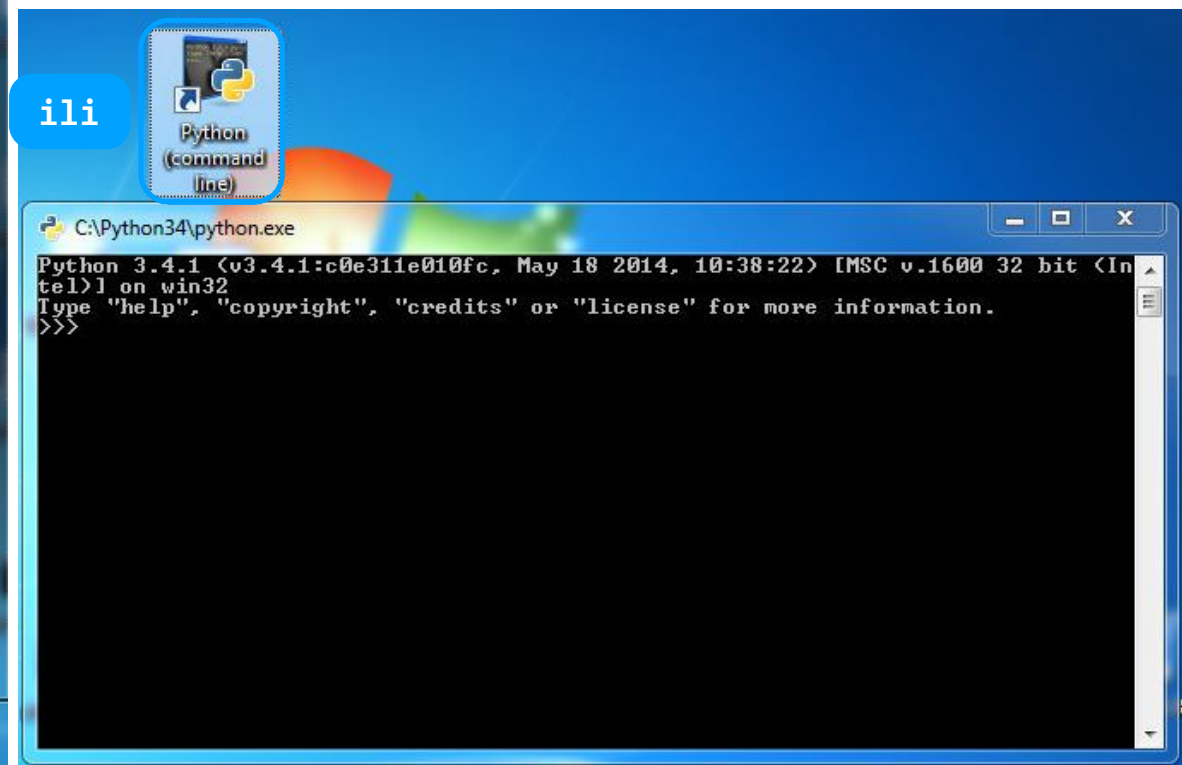
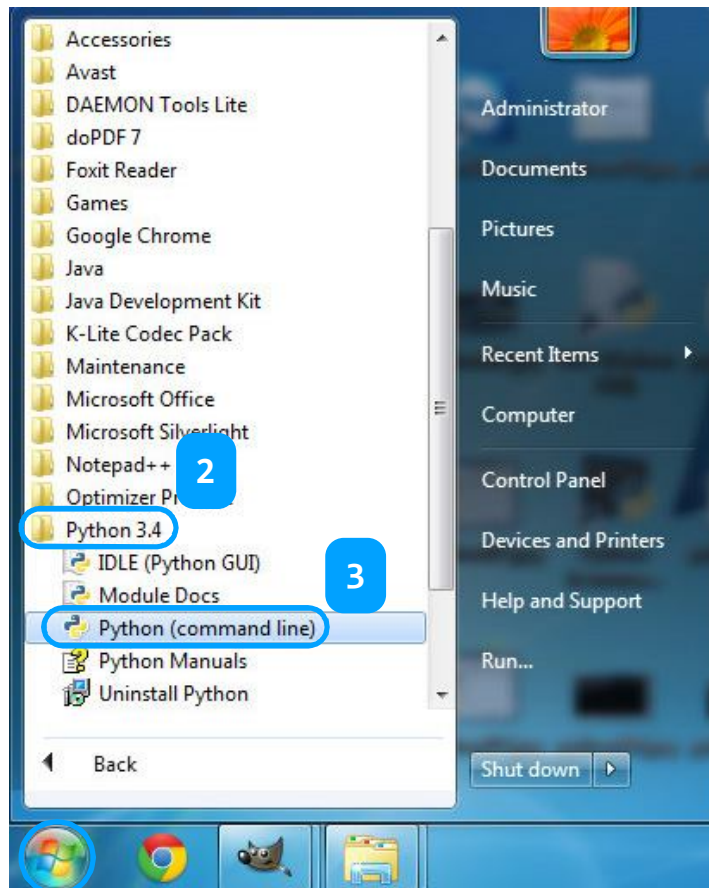
Instalacija Pythona



Pokretanje Python (command line)

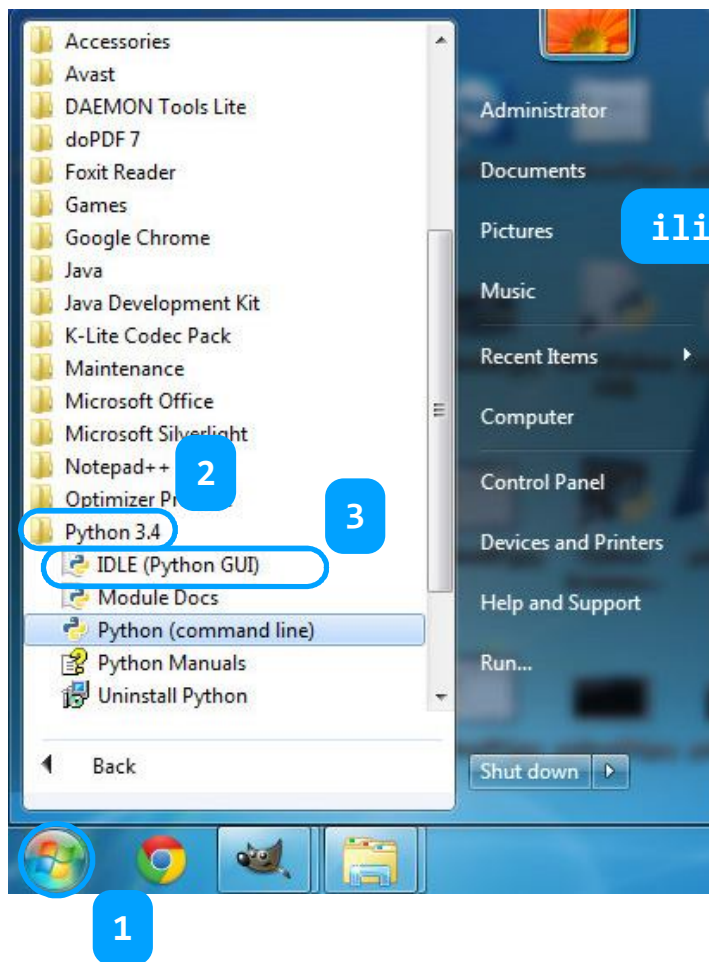


□ Python komandna linija (command line)

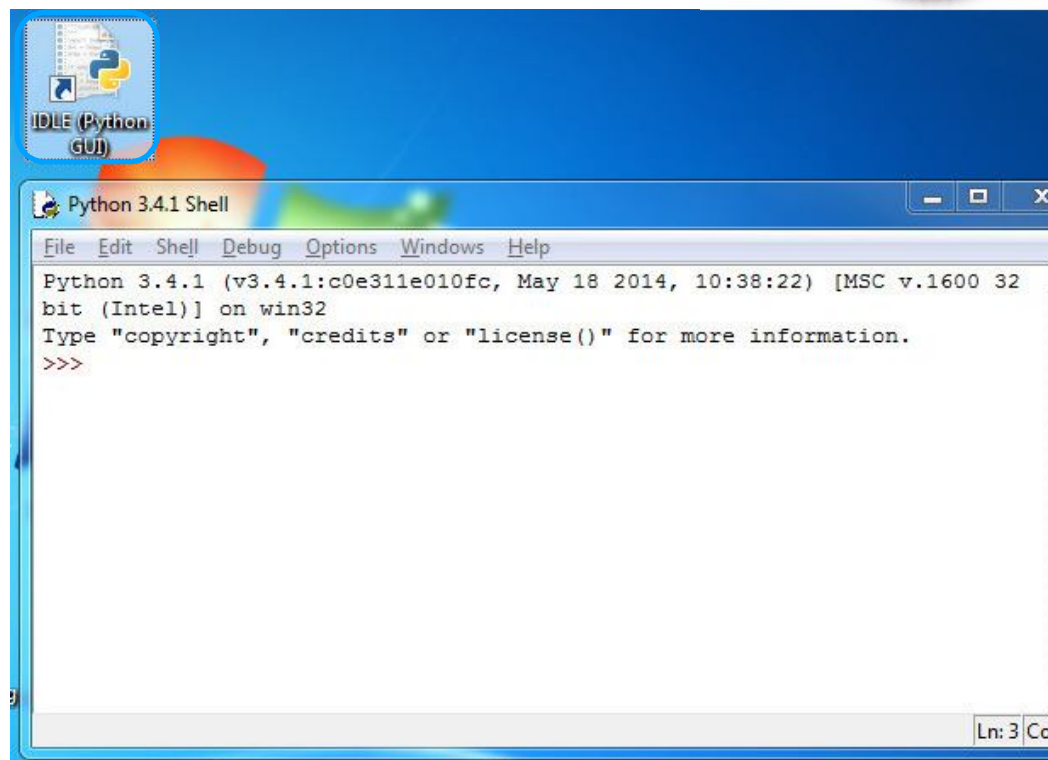


Pokretanje Python IDLE

□ Integrated DeveLopment Enviroment



Eric Idle - komičar
iz Monty Pythona



Osnovni tipovi podataka u Pythonu



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

GREŠKA: kod Pythona ne možemo unositi vodeće nule, crvenom bojom je ispisana greška, a crvenom bojom pozadine je označen dio koda gdje se dogodila greška

SyntaxError: invalid token

```
>>> 12345678901234567890123456789012345678901234
12345678901234567890123456789012345678901234
```

Brojevi s pomičnom točkom



□ Primjeri brojeva s pomičnom točkom

```
>>> 3.2          >>> -3.          >>> 0.32
3.2              -3.0          0.32
>>> 0.00032      >>> .000032    >>> 1e2
0.00032          3.2e-05      100.0
>>> 1e15         >>> 1e16
1000000000000000.0 1e+16
```

brojevi s pomičnom
točkom imaju granicu

□ Donja granica brojeva s pomičnom točkom

```
>>> 1.2345678901234567e-323  >>> 1.2345678901234567e-324
1e-323                        0.0
```

□ Gornja granica brojeva s pomičnom točkom

```
>>> 1.23456789012345678e308  >>> 1.23456789012345678e308
1.2345678901234567e+308      inf
```

Aritmetički operatori



zbrajanje	+
oduzimanje	-
množenje	*
dijeljenje	/
cjelobrojno dijeljenje	//
modulo (ostatak od dijeljenja)	%
potenciranje	**

- Prvenstvo pri izvođenju ima potenciranje, pa nakon toga množenje, dijeljenje, cjelobrojno dijeljenje i modulo, te na kraju zbrajanje i oduzimanje

Aritmetički izrazi



□ Jednostavni aritmetički izrazi

```
>>> 2 + 3          >>> 2 - 3          >>> 2 * 3
5                  -1                  6
>>> 2 / 3          >>> 2 % 3          >>> 2 ** 3
0.6666666666666666 2                  8
>>> 2 // 3        >>> 2+
0                  SyntaxError: invalid syntax
```

□ Složeni aritmetički izrazi

```
>>> 2 + 2 * 3      >>> (2 + 2) * 3 + 2 / 3
8                  12.666666666666666
```

□ Zadnja izračunata vrijednost

```
>>> 2 / 3          >>> _ * 2
0.6666666666666666 1.3333333333333333
```

GREŠKA: ako ne napišemo dobro aritmetički izraz prikazati će nam se greška

Zadatak: Razlomak



- Napišite jednu naredbu koja će izračunati vrijednost sljedećeg razlomka:

$$\frac{\frac{2+1}{2} + 2(3+1) - 4}{(2+3.1)\frac{2}{3} + 1}$$

Vrijeme



- Rješenje:



```
>>> ((2+1)/2 + 2*(3+1) - 4) / ((2+3.1)*2/3 + 1)  
1.25
```

brojnik razlomka
unutar zagrada

nazivnik razlomka
unutar zagrada

Nizovi znakova



☐ Jednostruki ili dvostruki navodnici

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

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

nizovi znakova su
označeni fontom
zelene boje

☐ 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'"
```

Nizovi znakova – funkcija print



- ❑ Funkcija je definirani skup naredbi
- ❑ Opći oblik funkcije u Pythonu

```
naziv_funkcije(parametar1, parametar2, ... , parametarN)
```

- ❑ Funkcija može primiti 0, 1 ili više parametara
- ❑ Funkcija **print**

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

```
>>> print('Python')
```

Python

```
>>> print('Radionica', 'Python', 2014)
```

Radionica Python 2014

funkcije print kao
parametre može primiti
različite tipove podataka

standardne Python
funkcije su označene
fontom ljubičaste boje

Nizovi znakova – funkcija print



☐ Aritmetički izrazi u ispisu

```
>>> print('Zbroj brojeva', 2, 'i', 3, 'je:', 2 + 3)
Zbroj brojeva 2 i 3 je: 5
```

☐ Ispis lijevo nakošene crte \

```
>>> print('Nakošena crta - \\.')
Nakošena crta - \.
```

☐ Tabulator - \t

```
>>> print('Korištenje\ttabulatora\tu\tPythonu.')
Korištenje      tabulatora      u      Pythonu.
```

☐ Prelazak u novi red pri ispisu - \n

```
>>> print('Prelazak\nu novi red u Pythonu.')
Prelazak
u novi red u Pythonu.
```

Variable

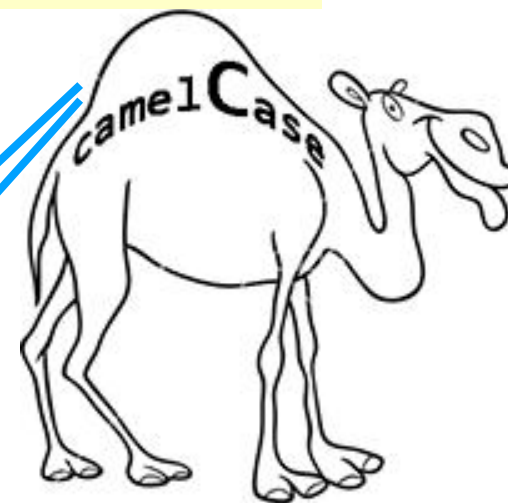


- ❑ Varijabla je memorijska lokacija kojoj pristupamo preko njenog naziva, a na njoj je zapisana vrijednost koja se može mijenjati
- ❑ Varijabla ima naziv i vrijednost

```
>>> varijabla = 10  
>>>
```

Diagram illustrating the variable assignment: `varijabla` is the **naziv variable** (variable name) and `10` is the **vrijednost variable** (variable value).

loši nazivi	dobri nazivi
aaaaa	brojac
abcdefgh	ime_prezime
ahauifhasfuhaiu	godinaRodjenja
hfjhds3u4444	imeNajPrijatelja



Variable



□ Pravila za imenovanje varijabli:

- Naziv varijable može sadržavati slova, brojeve i podvlake _
- Naziv varijable ne smije počinjati s brojem
- Naziv varijable ne smiju biti ključne riječi za koje su rezervirani nazivi, kao što **bool**, **True**, **False**, ...
- Naziv varijable smije sadržavati naše znakove (čćžšđČĆŽŠĐ), ali se to **nikako ne preporuča**
- Python razlikuje velika i mala slova, pa su **x** i **X**

Pridruživanje vrijednosti varijablama

□ Znak pridruživanja =

```
>>> x = 20
>>> print(x)
20
>>> x = x + 10
>>> print('x =', x)
x = 30

>>> y = -3.2
>>> print(y)
-3.2
>>> y = y * 2
>>> print('y =', y)
y = -6.4
```

□ U varijable se mogu spremiti i nizovi znakova

```
>>> python = 'Predavanje Python'
>>> print(python)
Predavanje Python
```

Zadatak: Sekunde



- Napišite niz naredbi koje će 10 000 sekundi pretvoriti u sate, minute i sekunde. Dobivene vrijednosti sata, minuta i sekundi spremite u varijable. Ispišite dobiveni broj sati, minuta i sekundi s jednom **print** funkcijom.

iz broja sekundi mogu dobiti minute tako da ih cjelobrojno podijelim s ukupnim brojem sekundi u 1 minuti

ostatak sekundi iz broja sekundi mogu dobiti tako da izračunam ostatak dijeljenja s ukupnim brojem sekundi u 1 minuti

Vrijeme





Zadatak: Sekunde - rješenje

```
>>> sekunde = 10000
>>> minute = sekunde // 60
>>> sekunde = sekunde % 60
>>> sati = minute // 60
>>> minute = minute % 60
>>> print(sati, 'sata,',
          minute, 'minuta i',
          sekunde, 'sekundi.')
2 sata, 46 minuta i 40 sekundi.
```

deklaracija
varijable sekundi i
pridruživanje
vrijednosti 10000

izračun broja minuta

izračun broja sekundi

izračun broja sati

izračun broja
preostalih minuta

ispis broja sati, minuta i sekundi

Korištenje pomoći



□ Funkcija **help**

naredba za izlazak iz pomoći

```
>>> help()
```

Welcome to Python 3.4's help utility!

If this is your first time using Python, you should definitely check

OUT za popis svih modula, ključnih riječi, simbola i tema python.org/3.4/tutorial/
En treba unijeti riječ modules, keywords, symbols ili topics

get help on writing Python programs and using Python modules. To quit this help utility and return to the interpreter, just type "quit".

To get a list of available modules, keywords, symbols, or topics, type "modules", "keywords", "symbols", or "topics". Each module also comes with a one-line summary of what it does; to list the modules whose name or summary contain a given string such as "spam", type "modules spam".

```
help> 
```

unos naziva modula, ključne riječi, simbola ili naziva funkcije za koju nam treba pomoć

Korištenje pomoći



□ Popis svih ključnih riječi Pythona

```
help> keywords
```

Here is a list of the Python keywords. Enter any keyword to get more help.

False	def	if	raise
None	del	import	return
True	elif	in	try
and	else	is	while
as	except	lambda	with
assert	finally	nonlocal	yield
break	for	not	
class	from	or	
continue	global	pass	

```
help>quit
```

izlazak iz pomoći

Korištenje pomoći



□ Drugi način: `help(naziv_funkcije)`

```
>>> help(print) ← pomoć za funkciju print
Help on built-in function print in module builtins:
```

```
print(...)
print(value, ..., sep=' ', end='\n', file=sys.stdout,
flush=False)
```

Prints the values to a stream, or to s

Optional keyword arguments:

file: a file-like object (s

sys.stdout.

sep: string inserted between values, default a space.

end: string appended after the last value, default a newline.

flush: whether to forcibly flush the stream.

```
>>>
```

vrijednosti koje ispisuje print
funkcija odvojene zarezom

standardno na kraju je prelazak
u novi red, ali korisnik može
postaviti neki niz znakova

standardno između dvije vrijednosti je razmak
ili korisnik može postaviti neki niz znakova

Funkcija print



□ Ispis niza vrijednosti

```
>>> godina = 2014
>>> radionica = 'Python'
>>> print('Radionica', radionica, godina, 'oš')
Radionica Python 2014 oš
```

□ Promjena standardnog separatora

```
>>> print('Radionica', 'Python', 'osnovne škole', sep='#')
Radionica#Python#osnovne škole
```

□ Promjena standardnog kraja ispisa

```
>>> print('Radionica', 'Python', 'osnovne škole', end='#')
Radionica Python osnovne škole#
```

Unos s tipkovnice



- ❑ Funkcija **input**

- ❑ Proučite pomoć za funkciju **input**

```
>>> help(input)
```

Help on built-in function input in module builtins:

```
input(...)
```

```
input([prompt]) -> string
```

funkcija input sve što se unese s tipkovnice sprema kao niz znakova

Read a string from standard input. The trailing newline is stripped.

If the user hits EOF (Unix: Ctl-D, Windows: Ctl-Z+Return), raise EOFError.

On Unix, GNU readline is used if enabled. The prompt string, if given,

is printed without a trailing newline before reading.

```
>>>
```

Unos s tipkovnice



- Pomoću funkcije **input** unesite vaše ime s tipkovnice, spremite ga u varijablu **ime**, te nakon toga ispišite vrijednost varijable **ime**.

```
>>> ime = input('Unesi svoje ime: ')\nUnesite vaše ime: Tomo\n>>> print('Uneseno ime je: ', ime)\nUneseno ime je: Tomo
```

Unos s tipkovnice



□ Primjer funkcije `input` s cijelim brojem

```
>>> broj = input('Unesi cijeli broj: ')
```

```
Unesi cijeli broj: 10
```

```
>>> broj + 10
```

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

```
  File "<pyshell#6>", line 1, in <module>
```

```
    broj+10
```

```
TypeError: Can't convert 'int' object to str implicitly
```

GREŠKA: broj 10 unesen s tipkovnice je spremljen kao niz znakova

□ Funkcija `int` – pretvara u cijeli broj

```
>>> broj = input('Unesi cijeli broj: ')
```

```
Unesi cijeli broj: 10
```

```
>>> broj = int(broj)
```

```
>>> broj + 10
```

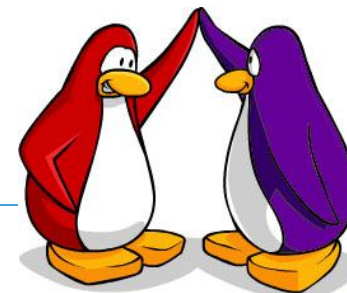
```
20
```

Program



- ☐ Naredbe smo dosad unosili i odmah pokretali u Python IDLE-u
- ☐ Što će se dogoditi ako zatvorimo Python IDLE?
- ☐ Izgubili smo sve naredbe koje smo unosili
- ☐ Program je skup naredbi čijim se izvršenjem obavlja neki posao
- ☐ Naredbe možemo spremiti kao poseban program, pa taj program možemo naknadno ažurirati i pokretati

Prvi Python program



Python IDLE

Python IDLE editor

komentari

kod programa

ekstenzija .py

unos naziva programa

spremanje programa

```
#Jednolinijski komentari
#Ovo je moj prvi Python program.

"""
Komentar u više linija.
Mogu se koristiti i jednostruki navodnici.
Ovo je moj prvi Python progr.
"""

'Ovo je također jednolinijski komentar.'
'Ovo je moj prvi Python program.'

print('Hello World!')
```

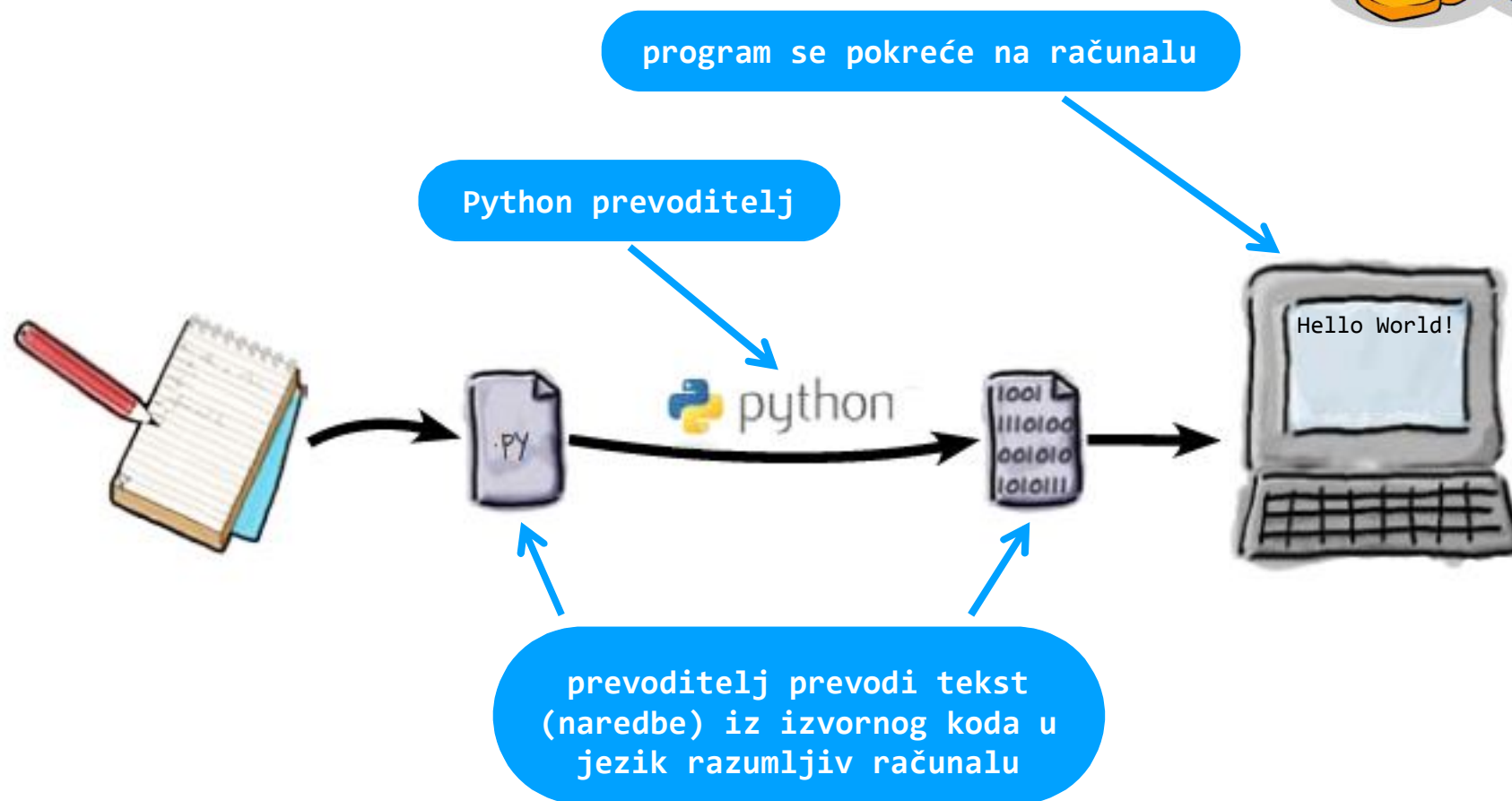
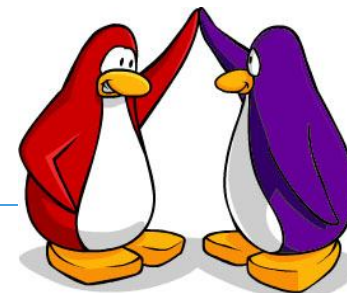
Save in: python_primeri

Name	Date modified	Type
No items match your search.		

File name: helloworld.py

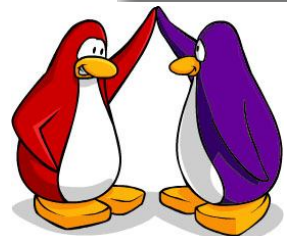
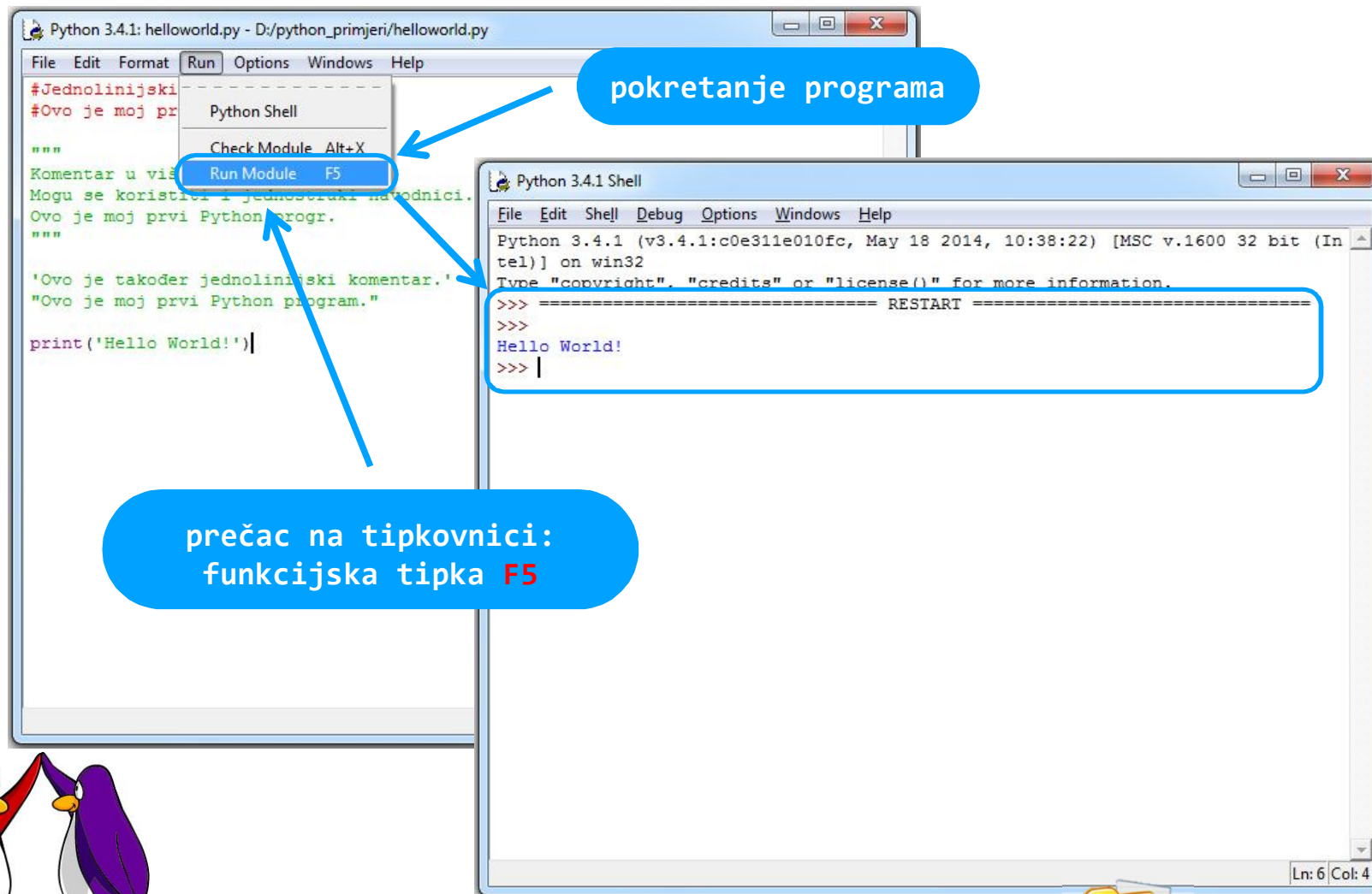
Save as type: Python files (*.py;*.pyw)

Prevođenje programa



Prilagođeno iz: P. Barry & D. Griffiths, Head First Programming, O'Reilly, 2009

Pokretanje programa u Python IDLE-u



helloworld.py

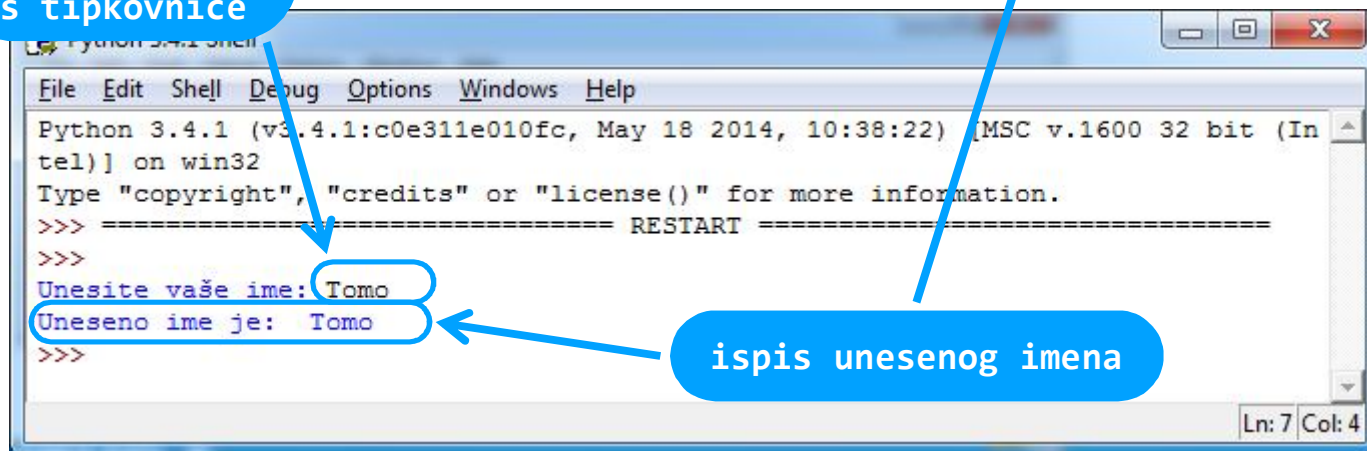
Program: Unos imena



- Naredbe za unos imena i ispis imena na ekran iz primjera spremite u program **unos_imena.py**, te pokrenite program u Python IDLE-u.

```
ime = input('Unesite vaše ime: ')\nprint('Uneseno ime je: ', ime)
```

unosi se ime
s tipkovnice



ispis unesenog imena



unos_imena.py

Zadatak: Prosjek



- Napišite program koji računa prosječnu ocjenu iz četiri predmeta. Ocjene se unose s tipkovnice. Ispišite dobivenu prosječnu ocjenu ispišite. Pokrenite i testirajte vaš program.

$$\frac{ocjena1 + ocjena2 + ocjena3 + ocjena4}{ukupan_broj_ocjena}$$

Vrijeme





Zadatak: Prosjek - rješenje

```
ocjena1 = int(input('Unesi prvu ocjenu: '))
ocjena2 = int(input('Unesi drugu ocjenu: '))
ocjena3 = int(input('Unesi treću ocjenu: '))
ocjena4 = int(input('Unesi četvrtu ocjenu: '))
prosjek = (ocjena1 + ocjena2 + ocjena3 + ocjena4)/4
print('Prosjek ocjena:', prosjek)
```

```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:38:22) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Unesi prvu ocjenu: 3
Unesi drugu ocjenu: 5
Unesi treću ocjenu: 4
Unesi četvrtu ocjenu: 5
Prosjek ocjena je: 4.25
>>> |
```



prosjek.py