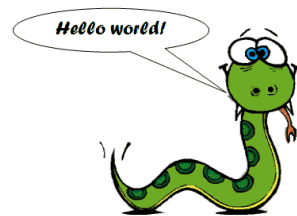
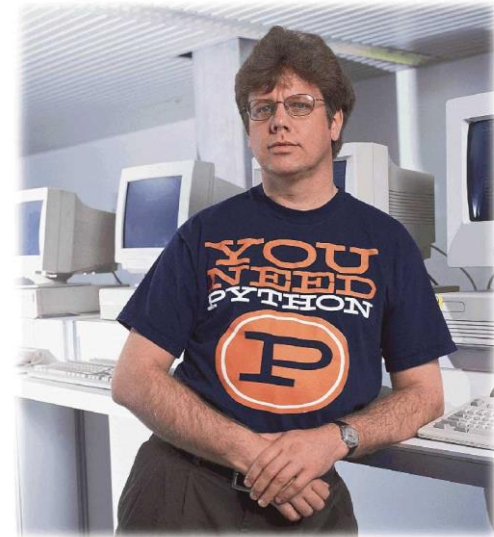


Uvod u programski jezik Python

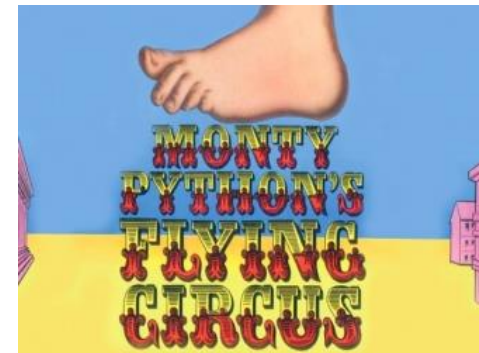
Zlatan Soldo dipl. ing.
Geodetska tehnička škola
Zagreb, ožujak 2014.

Zašto Python ?

- Nastao 1991. godine
- Autor (inicijator): Guido van Rossum (National Research Institute for Mathematics and Computer Science in the Netherlands)
- Radi na više OS (Windows, Linux/Unix, Mac OS X)
- Jednostavan i lak (za početnike) – interpreter
- Moćan za (profesionalce)
- Dobro dokumentiran
- Među Top10 po popularnosti
- Besplatan, dostupan na adresi: www.python.org



- Ime nema veze sa:
- Ime dobio po seriji "Monty Python's Flying Circus"



Ponovimo osnovne pojmove:

Algoritam - precizno opisan način rješenja nekog problema

Zadatak:

Napiši algoritam za program koji će iz učitane duljine stranice kvadrata izračunati njegovu površinu i opseg i ispisati ih na ekranu.

Algoritam

Prezentacija
iz udžbenika
Sysprint

početak:

učitaj duljinu stranice a

izračunaj površinu kvadrata

izračunaj opseg kvadrata

ispiši površinu kvadrata

ispiši opseg kvadrata

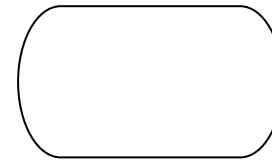
kraj

Ponovimo osnovne pojmove:

Dijagram toka – grafički prikaz algoritma

Simboli dijagrama toka su unaprijed dogovoreni

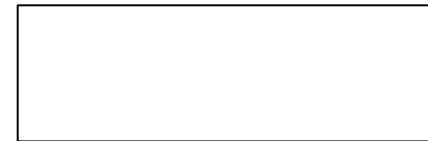
Ulaz-izlaz
Prezentacija
iz udžbenika
Sysprint



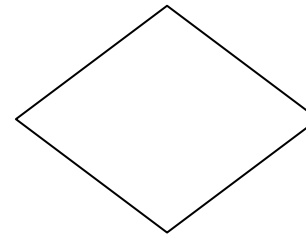
početak



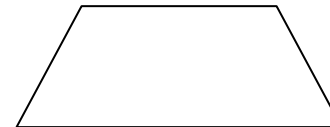
Ulaz



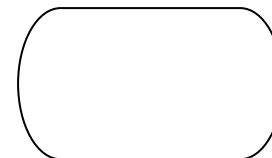
Obrada



odluka
(grananje)



Izlaz



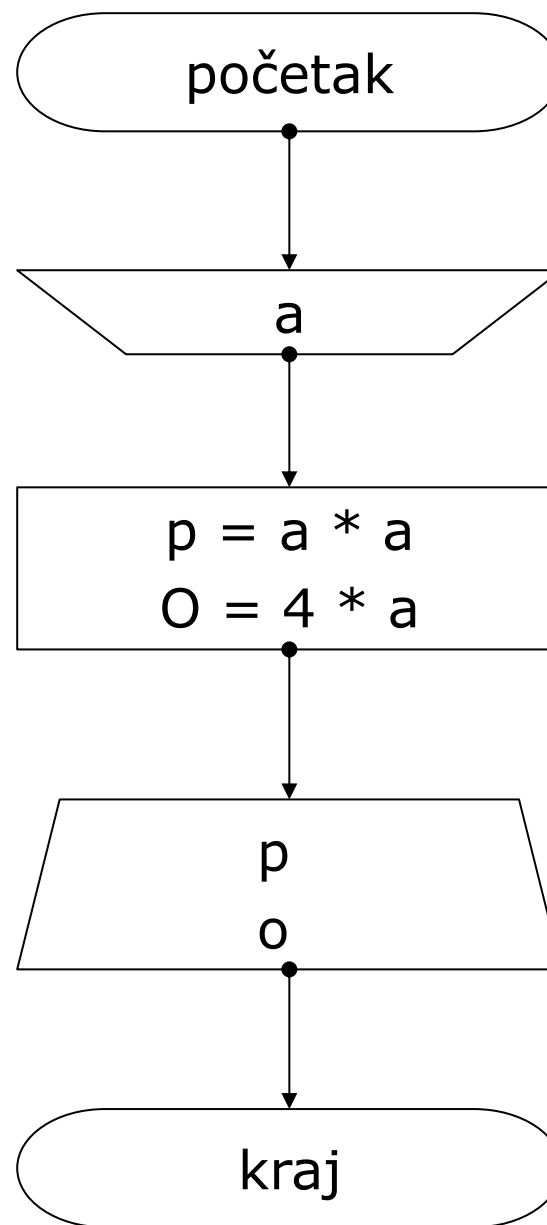
kraj

Ponovimo osnovne pojmove:

Dijagram toka – grafički prikaz algoritma

Zadatak:

Nacrtaj dijagram toka za program koji će iz učitane duljine stranice kvadrata izračunati njegovu površinu i opseg i ispisati ih na ekranu.



Ponovimo osnovne pojmove:

Program - točan niz naredbi koje računalo izvršava da bi se riješio određeni zadatak.

Zadatak:

Napiši program koji će iz učitane duljine stranice kvadrata izračunati njegovu površinu i opseg i ispisati ih na ekranu.

```
a = float(input("dužina = "))
```

```
povrsina = a * a
```

```
opseg = 4 * a
```

```
print('Površina je', povrsina)
```

```
print('Opseg je', opseg)
```

Ponovimo osnovne pojmove:

Programiranje – postupak izrade programa

Programer – osoba koja stvara program



Analiza problema:

Što i kako treba riješiti, što su ulazni podaci, a što trebamo dobiti – što su izlazni podaci

Planiranje:

Planiranje ljudi, vremena, opreme itd., odabir programskog jezika, sastavljanje algoritma.

Kodiranje:

Pisanje algoritma na nekom programskom jeziku

Testiranje programa i ispravljanje pogrešaka:

Program testiramo ulaznim podacima za koje unaprijed znamo rezultat

Dokumentacija:

Korisnička dokumentacija je neka vrsta „uputa za upotrebu“

Programerska dokumentacija sadrži detaljan opis programa koji omogućava naknadne prilagodbe i održavanje

Održavanje:

Gotov program treba prilagođavati zahtjevima korisnika (nove inačice) i otklanjati uočene pogreške.

Ponovimo osnovne pojmove:

Programski jezik – sredstvo za pisanje programa

Pseudo jezik – ili pseudokod je način zapisa algoritma kratkim izrazima nalik govornom jeziku

Zadatak:

Napiši program koji će iz učitane duljine stranice kvadrata izračunati njegovu površinu i opseg i ispisati ih na ekranu.

početak:

učitaj duljinu stranice a

površina = $a * a$

opseg = $4 * a$

ispiši površina

ispiši opseg

kraj

Programi prevoditelji

Interpreter

Program odmah spreman za izvođenje. Svaku naredbu izvornog koda pročita, prevede i izvrši.

Pročitaj naredbu

Prevedi naredbu

Izvrši naredbu

Potraži slijedeću naredbu



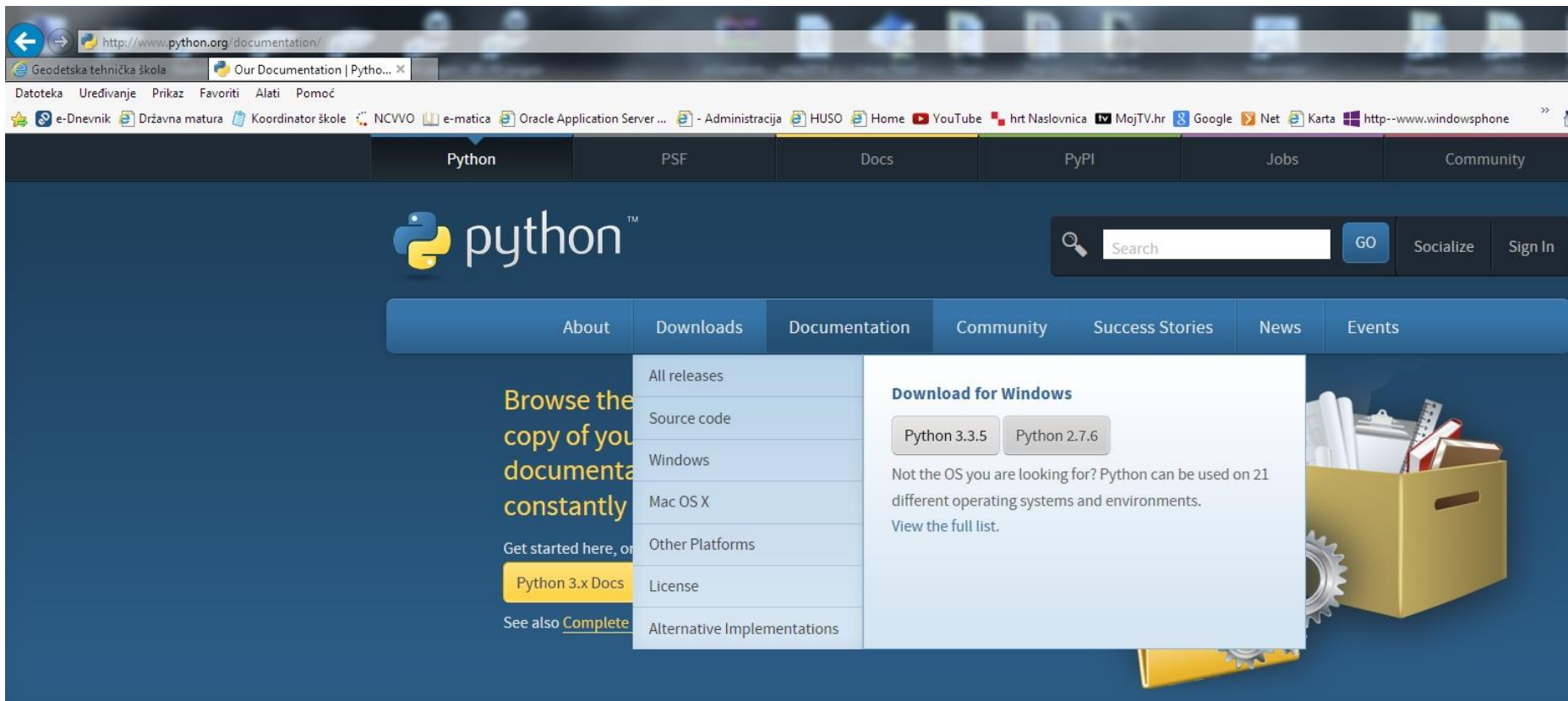
Kompajler

Izvorni kod (source code) je tekstualna datoteka razumljiva čovjeku koja se prije izvođenja prevede (kompajlira) i tako nastaje izvršni kod koji se izvršava (na računalu pod određenim OS-om.)

www.python.org



Python Programming Language – Official Website



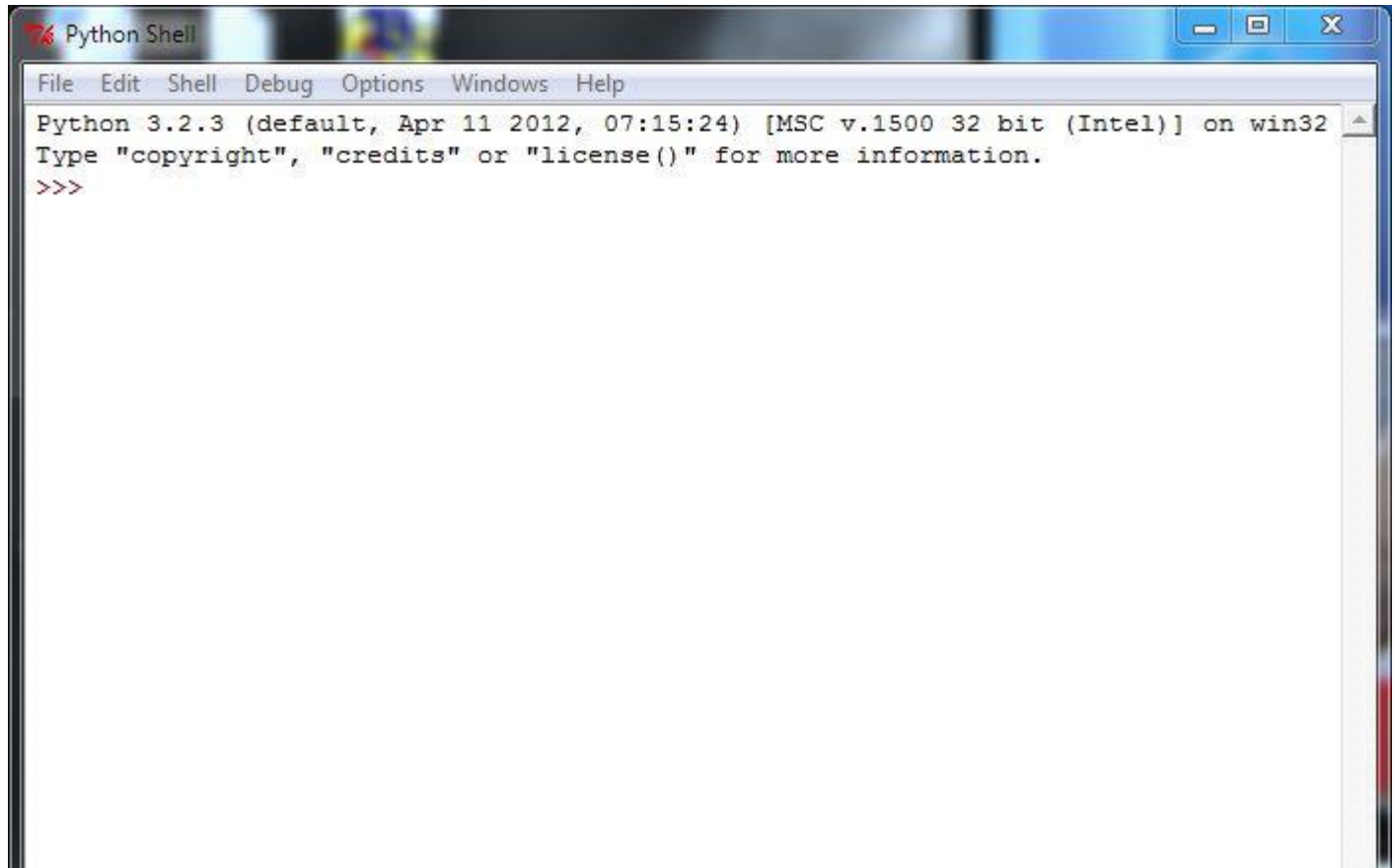
The current production versions are Python 3.3.5. and Python 2.7.6

Nakon instalacije

- Prvo pokretanje:



- IDLE (Python GUI) Integrated DeveLopment Environment



The screenshot shows a Windows-style window titled "Python Shell". The window has a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main content area displays the following text:

```
Python 3.2.3 (default, Apr 11 2012, 07:15:24) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
```

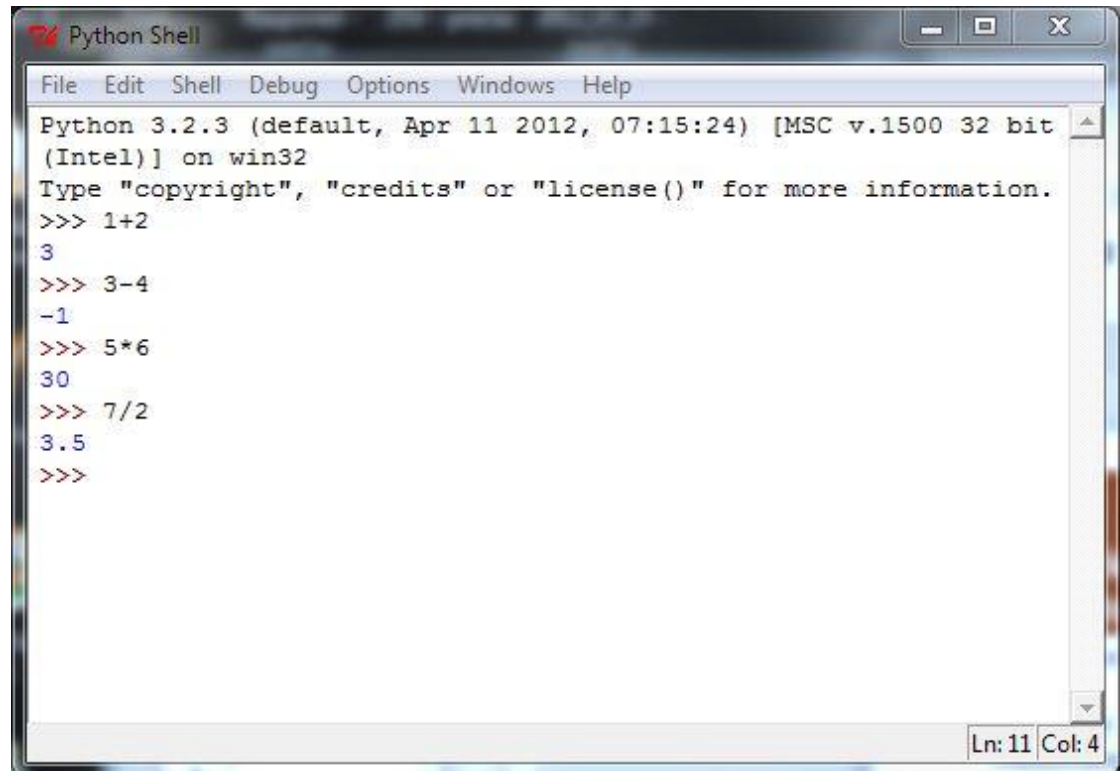


Python
(command line)

Python kao kalkulator

Osnovne matematičke operacije:

- + Zbrajanje
- - Oduzimanje
- * Množenje
- / Dijeljenje



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.2.3 (default, Apr 11 2012, 07:15:24) [MSC v.1500 32 bit
(Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 1+2
3
>>> 3-4
-1
>>> 5*6
30
>>> 7/2
3.5
>>>
```

Ln: 11 Col: 4

Uvodimo zagrade

Pri računanju prvo se izvode računске operacije višeg reda, a zatim one nižeg reda:

1. Potenciranje
2. Množenje i dijeljenje
3. Zbrajanje i oduzimanje

Primjer:

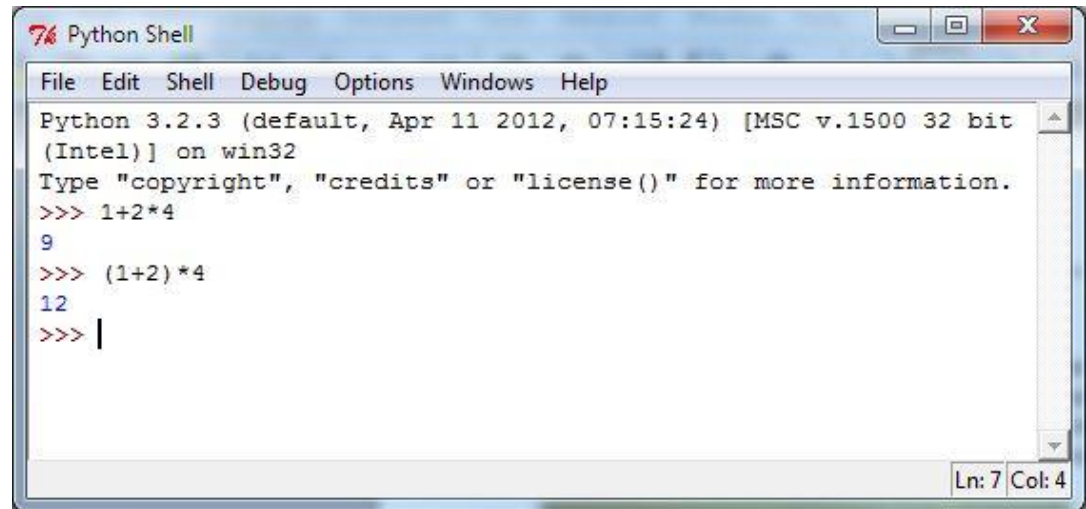
$1+2*4$

Napomena!

Uvođenjem zagrada mijenja se redoslijed izvođenja računskih operacija. Prvo se izvode one računске operacije koje su u zagradaama.

Primjer:

$(1+2)*4$



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.2.3 (default, Apr 11 2012, 07:15:24) [MSC v.1500 32 bit
(Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 1+2*4
9
>>> (1+2)*4
12
>>> |
```

(malo drugačije) Dijeljenje

- / (obično) dijeljenje
- // cjelobrojno dijeljenje
- % ostatak cjelobrojnog dijeljenja

> > > 8.5/2

4.25

> > > 8.5//2

4.0

> > > 8.5%2

0.5

(malo drugačije) Dijeljenje

Zadatak:

Ulaznica za kino košta 25 kuna. Ti imaš 72 kune.

1. Izračunaj koliko ulaznica možeš kupiti.
2. Izračunaj koliko će ti kuna ostati nakon kupovine ulaznica

> > > 72//25

2

> > > 72%25

22

> > >

Potenciranje

. ** Potenciranje

Napomena:

Potenciranje na $1/n$ je isto što i n -ti korijen

Primjer:

$$2^3 = 8$$

$$\sqrt[3]{8} = 2$$

$$8^{1/3} = 2$$

Izračunaj:

$$5^2$$

$$\sqrt{25}$$

$$>>> 5^{**}2$$

$$25$$

$$>>> 25^{**}1/2$$

$$12.5$$

$$>>> 25^{**}(1/2)$$

$$5.0$$

Matematičke funkcije

Većina funkcija se nalazi u modulima. **Modul** je svaka datoteka s Python naredbama, čije ime ima nastavak .py

Standardne matematičke funkcije su u modulu *math*

Prije uporabe funkcija potrebno je učitati pojedinu funkciju iz modula (ili sve funkcije iz modula)

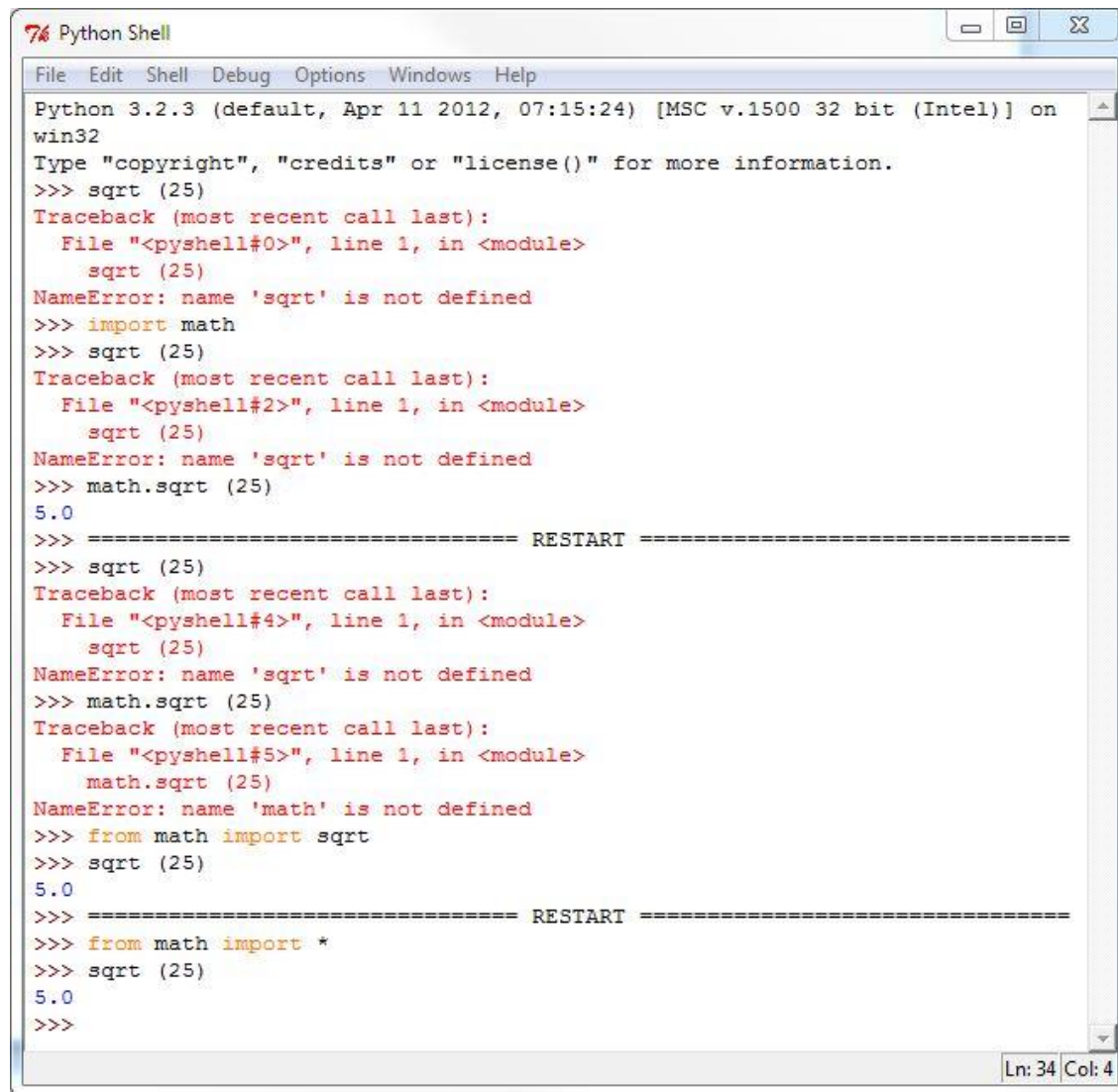
Primjer:

$$\sqrt{25} = 5$$

`sqrt (25)`

Napomena!

Restart shell (Ctrl+F6) briše sve učitanе module



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.2.3 (default, Apr 11 2012, 07:15:24) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> sqrt (25)
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    sqrt (25)
NameError: name 'sqrt' is not defined
>>> import math
>>> sqrt (25)
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    sqrt (25)
NameError: name 'sqrt' is not defined
>>> math.sqrt (25)
5.0
>>> ===== RESTART =====
>>> sqrt (25)
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    sqrt (25)
NameError: name 'sqrt' is not defined
>>> math.sqrt (25)
Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    math.sqrt (25)
NameError: name 'math' is not defined
>>> from math import sqrt
>>> sqrt (25)
5.0
>>> ===== RESTART =====
>>> from math import *
>>> sqrt (25)
5.0
>>>
```

Uvodimo varijable

Svi računalni programi na neki način obrađuju podatke.

Podaci se sakupljaju, pohranjuju, obrađuju i mijenjaju tijekom izvođenja programa.

Svi podaci koje program koristi moraju biti pohranjeni u memoriji

- **Memorijska lokacija** u kojoj je pohranjen podatak je jedinstveno određena svojom **adresom**. Da biste mogli koristiti pohranjene podatke, morate znati adresu memorijske lokacije na kojoj su pohranjeni.
- Da bi se olakšao pristup podacima, koriste se **varijable** koje služe kao spona između programa i memorijskih lokacija s podacima, jer se varijabla pridružuje adresi memorijske lokacije
- Svaka varijabla ima **ime, tip i vrijednost**
- Kada se nova vrijednost zapisuje u varijablu, ona zamijenjuje (briše) prethodnu vrijednost
- Čitanje varijabli iz memorije ih ne mijenja
- Varijabli se pristupa zvanjem njena imena

Svaka varijabla ima svoje ime

Ime varijable mora počinjati slovom ili podcrtom _

Ime varijable ne smije počinjati brojem

Ime varijable može sadržavati slova brojke i podcrtu

U imenu varijable razlikuju se velika i mala slova

Mogu se (ograničeno) koristiti hrvatski dijakritički znakovi: č,ć,đ,š,ž,Č,Ć,Đ,Š,Ž

Savjet:

Izbjegavajte ih u imenu varijable

Ispravno ime varijable:

a

A

X

DUZINA_1

duzina_1

Duzina_1

_Sirina

_sirina

vertikalni_kut

HorizontalniKut

Neispravno ime varijable:

1stranica

Radius*kruznice

print

Nedozvoljena imena

<u>abs()</u>	<u>dict()</u>	<u>help()</u>	<u>min()</u>	<u>setattr()</u>
<u>all()</u>	<u>dir()</u>	<u>hex()</u>	<u>next()</u>	<u>slice()</u>
<u>any()</u>	<u>divmod()</u>	<u>id()</u>	<u>object()</u>	<u>sorted()</u>
<u>ascii()</u>	<u>enumerate()</u>	<u>input()</u>	<u>oct()</u>	<u>staticmethod()</u>
<u>bin()</u>	<u>eval()</u>	<u>int()</u>	<u>open()</u>	<u>str()</u>
<u>bool()</u>	<u>exec()</u>	<u>isinstance()</u>	<u>ord()</u>	<u>sum()</u>
<u>bytearray()</u>	<u>filter()</u>	<u>issubclass()</u>	<u>pow()</u>	<u>super()</u>
<u>bytes()</u>	<u>float()</u>	<u>iter()</u>	<u>print()</u>	<u>tuple()</u>
<u>callable()</u>	<u>format()</u>	<u>len()</u>	<u>property()</u>	<u>type()</u>
<u>chr()</u>	<u>frozenset()</u>	<u>list()</u>	<u>range()</u>	<u>vars()</u>
<u>classmethod()</u>	<u>getattr()</u>	<u>locals()</u>	<u>repr()</u>	<u>zip()</u>
<u>compile()</u>	<u>globals()</u>	<u>map()</u>	<u>reversed()</u>	<u>__import__()</u>
<u>complex()</u>	<u>hasattr()</u>	<u>max()</u>	<u>round()</u>	
<u>delattr()</u>	<u>hash()</u>	<u>memoryview()</u>	<u>set()</u>	

Još malo o varijablama

Svaki podatak koji program obrađuje smješten je u memoriji računala na određenoj memorijskoj adresi (lokaciji).

Mjesto u memoriji rezervirano za pohranu podatka naziva se **varijabla**.

Primjer:

Jedno od rezerviranih memorijskih mjesta označimo kao varijablu **a** i pridružimo mu vrijednost **5**



Različiti tipovi varijabli

Varijable mogu biti različitog tipa.

numeričke (brojčane) varijable mogu biti:

Cjelobrojne (**integer**)

a = 3

Relni (**float**)

a = 3.8

string (nizovne) varijable su niz znakova (slova, brojke i posebni znakovi) napisan između navodnika ili tzv. jednostrukih navodnika

Ostale: liste, N-torke, rječnik

Primjer:

Naredba **print** ispisuje vrijednost ranije zadane varijable

```
>>> a=5
```

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

```
5
```

```
>>> A=25
```

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

```
25
```

```
>>> X="Pozdrav svima!"
```

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

```
Pozdrav svima!
```

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

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

```
  File "<pyshell#9>", line 1, in  
<module>
```

```
    print (x)
```

```
NameError: name 'x' is not defined
```

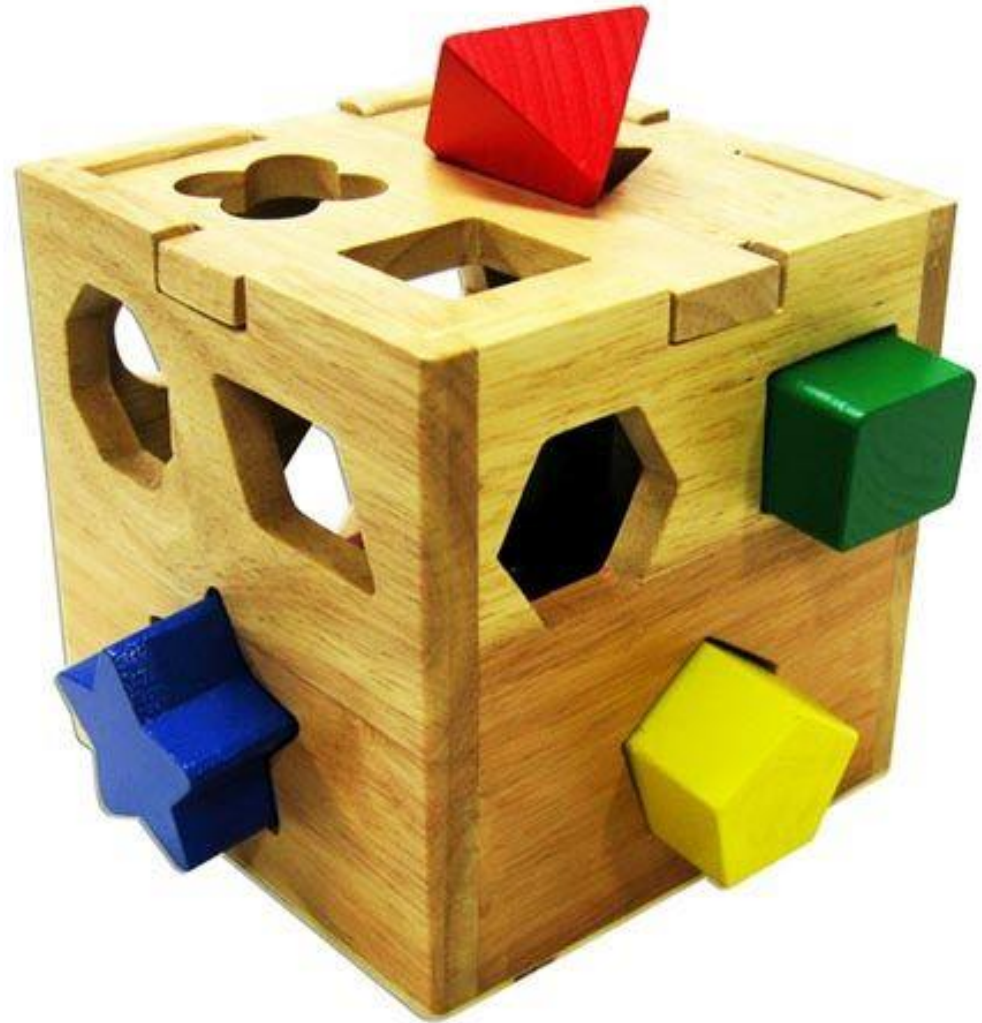
Ništa nevažno o tipovima podataka

Važno je znati da se prema različitim tipovima podataka treba različito odnositi.

– Na primjer, različiti tipovi brojeva zahtijevaju različitu količinu memorije za pohranu.

Također, nekim s tipovima podataka možemo raditi ono što s nekim drugima ne možemo.

– Na primjer, brojeve možemo zbrajati, oduzimati, množiti i dijeliti, dok stringove ne možemo. S druge strane, stringove možemo ulančavati.



Primjeri

```
>>> a = 5
>>> b = '12'
>>> print (a)
5
>>> print (b)
12
>>> print (a+a)
10
>>> print (b+b)
1212
>>>
```

```
>>> print (a+b)
```

Traceback (most recent call last):

```
File "<pyshell#12>", line 1, in <module>
    print (a+b)
```

TypeError: unsupported operand type(s) for +:
'int' and **'str'**

```
>>>
```

```
>>> print (4*b)
```

```
12121212
```

```
>>>
```

```
>>> print (b*3)
```

```
121212
```

```
>>>
```


Help

Naredba help prikazuje sve opcije pojedine naredbe, funkcije ...

Primjer:

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

Help on built-in function print in module builtins:

```
print(...)
```

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

Prints the values to a stream, or to sys.stdout by default.

Optional keyword arguments:

file: a file-like object (stream); defaults to the current sys.stdout.

sep: string inserted between values, default a space.

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

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

Traceback (most recent call last):

```
  File "<pyshell#18>", line 1, in <module>
```

```
    help (math)
```

NameError: name 'math' is not defined

```
>>> import math
```

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

Help on built-in module math:

NAME

math

DESCRIPTION

This module is always available. It provides access to the mathematical functions defined by the C standard.

FUNCTIONS

acos(...)

acos(x)

Return the arc cosine (measured in radians) of x.

acosh(...)

acosh(x)

Return the hyperbolic arc cosine (measured in radians) of x.

asin(...)

asin(x)

Return the arc sine (measured in radians) of x.

... itd ...

Help

Izbor nekih funkcija modula
math

Napomena!

Konstante **pi** i **e** su dio modula
math

cos(...)
cos(x)

Return the cosine of x (measured in radians).

degrees(...)
degrees(x)

Convert angle x from radians to degrees.

radians(...)
radians(x)

Convert angle x from degrees to radians.

sin(...)
sin(x)

Return the sine of x (measured in radians).

sqrt(...)
sqrt(x)

Return the square root of x.

tan(...)
tan(x)

DATA

e = 2.718281828459045
pi = 3.141592653589793

>>>



(konačno...)

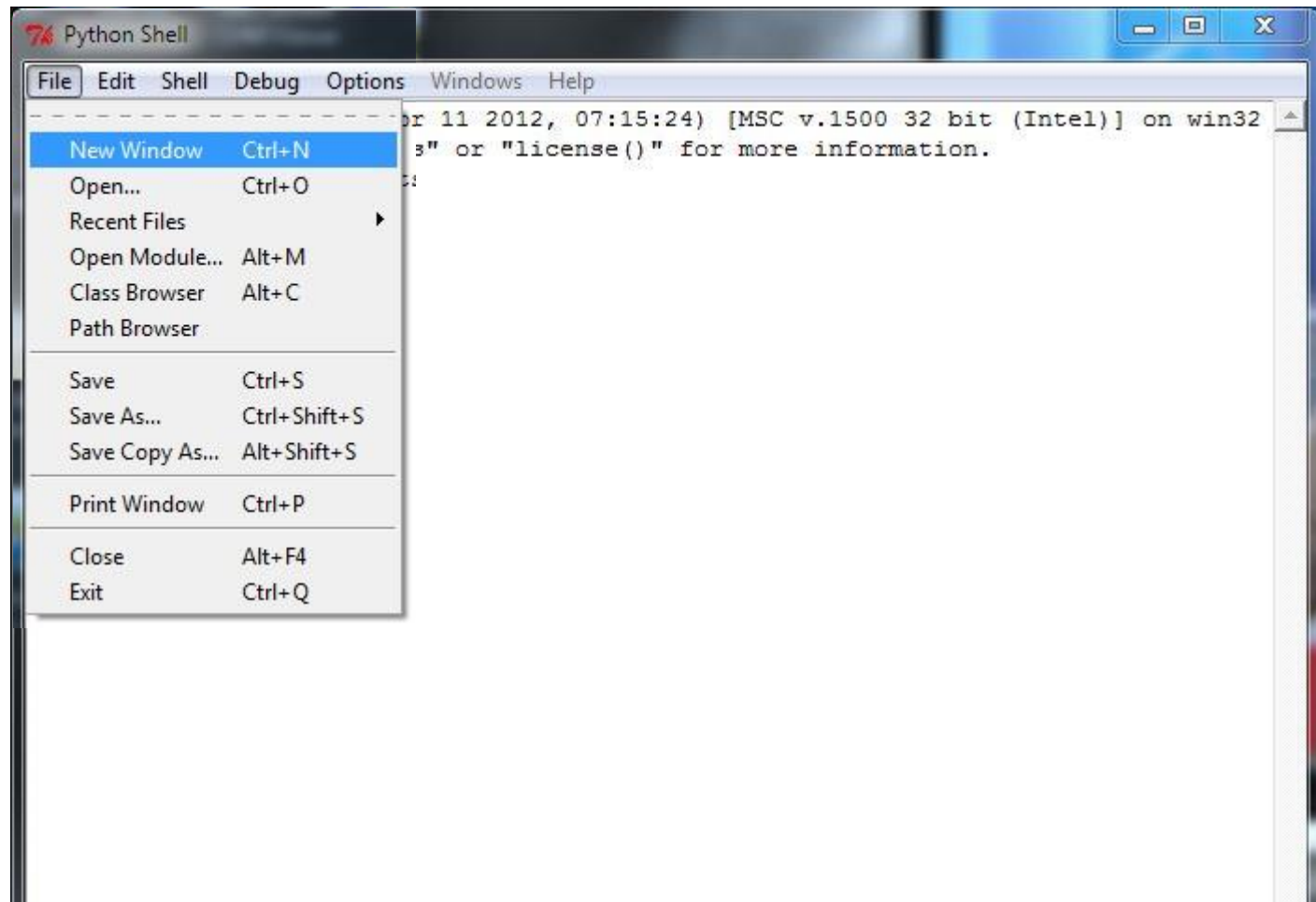
PROGRAMIRANJE

Moj prvi program

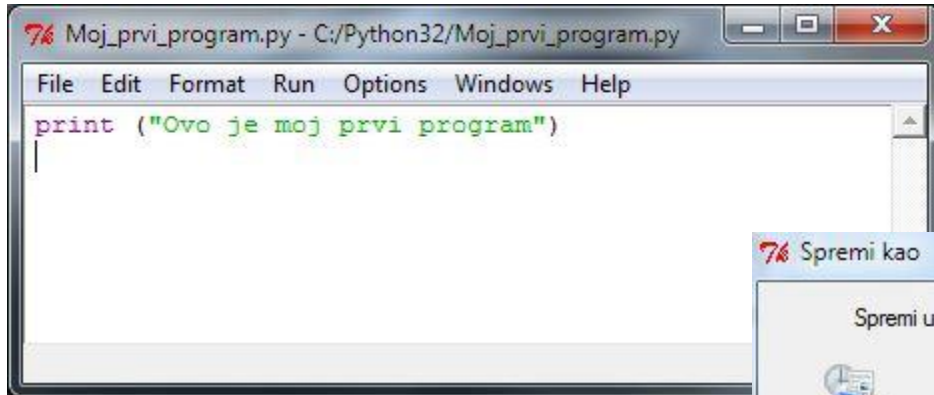
- Pokreni IDLE (Python GUI)



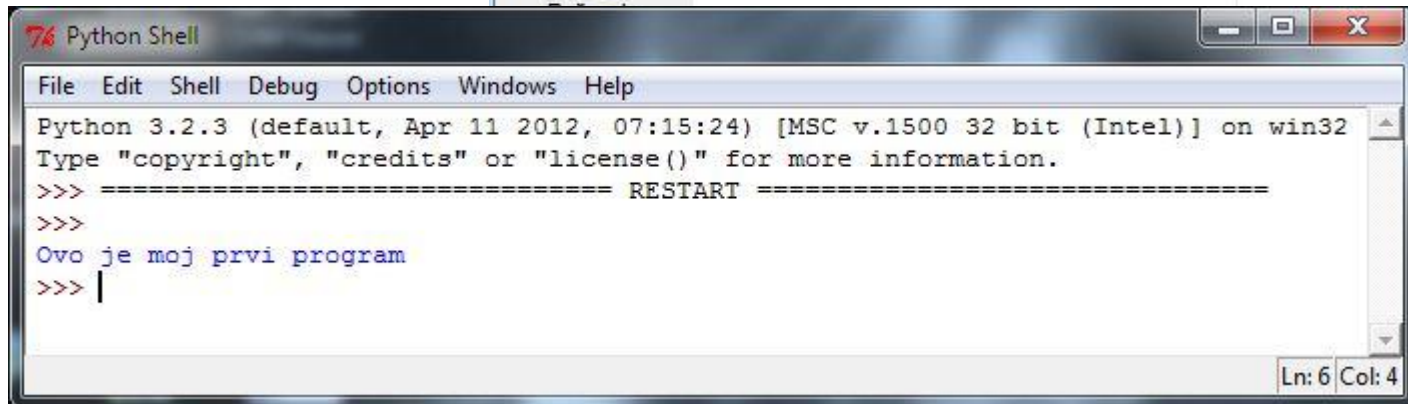
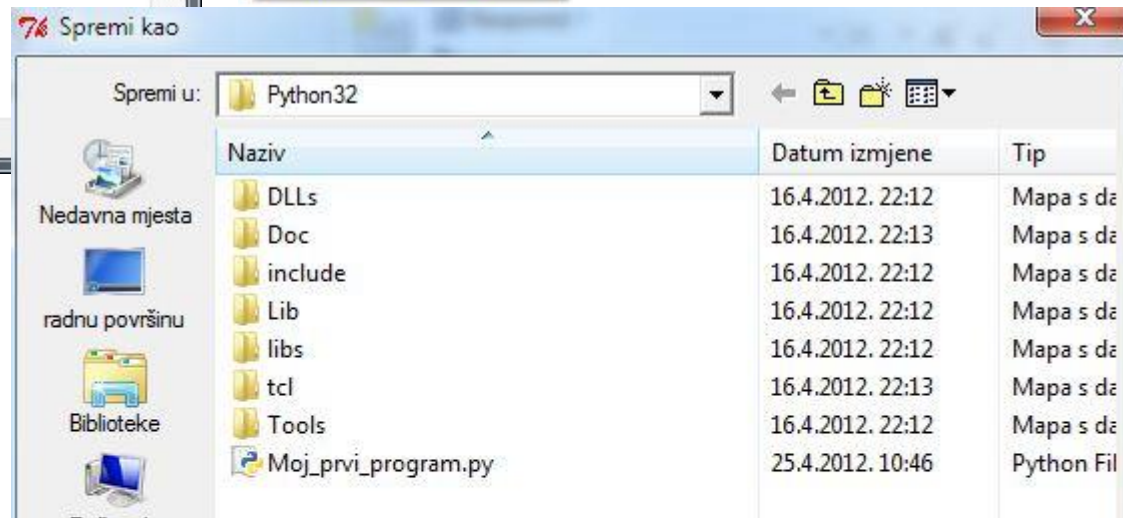
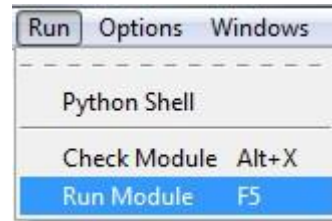
IDLE (Python GUI)



Moj prvi program

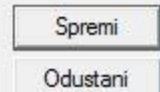


```
7% Moj_prvi_program.py - C:/Python32/Moj_prvi_program.py
File Edit Format Run Options Windows Help
print ("Ovo je moj prvi program")
```



```
7% Python Shell
File Edit Shell Debug Options Windows Help
Python 3.2.3 (default, Apr 11 2012, 07:15:24) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Ovo je moj prvi program
>>> |
```

Ln: 6 Col: 4



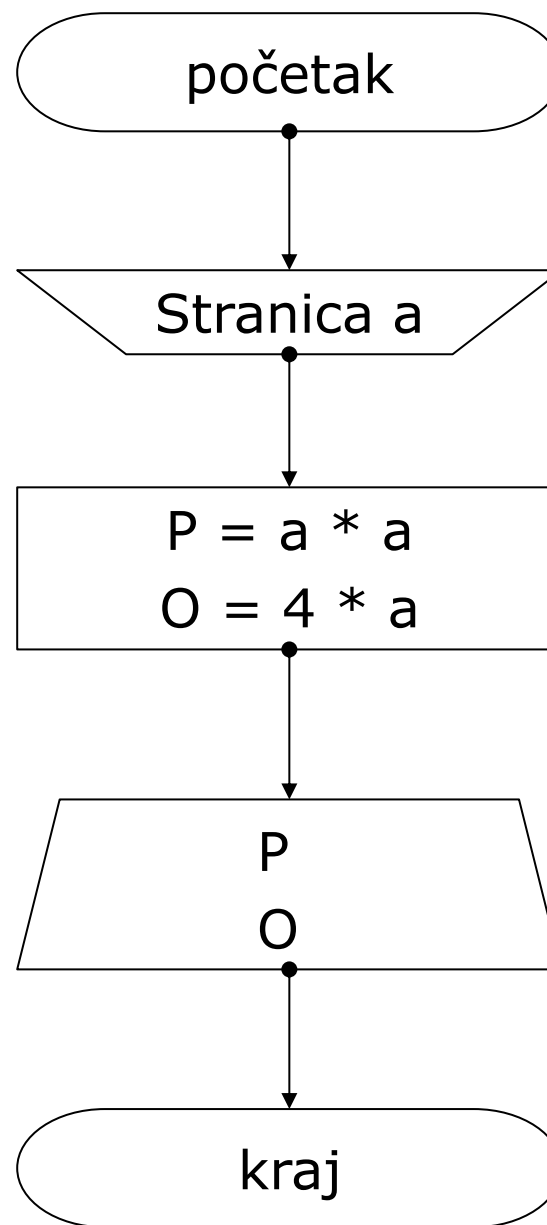
Zadatak:

Nacrtaj dijagram toka za program koji će na temelju zadane duljine stranice kvadrata a izračunati njegovu površinu P i opseg O . Vrijednosti P i O treba ispisati ih na ekranu.

Ulaz: duljina stranice a

Obrada: $P = a^2$
 $O = 4 * a$

Izlaz: površina kvadrata P
opseg kvadrata O



Zadatak:

Napiši program koji će na temelju zadane duljine stranice kvadrata a izračunati njegovu površinu P i opseg O . Vrijednosti P i O treba ispisati ih na ekranu.

Ulaz: duljina stranice a

Obrada: $P = a^2$
 $O = 4 * a$

Izlaz: površina kvadrata P
opseg kvadrata O

```
a = 2
```

```
P = a * a
```

```
O = 4 * a
```

```
print (P)
```

```
print (O)
```

```
>>>
```

```
Izlaz:
```

```
>>>
```

```
4
```

```
8
```

```
>>>
```

Zadatak:

Napiši program koji će na temelju zadane duljine stranice kvadrata a izračunati njegovu površinu P i opseg O. Vrijednosti P i O treba ispisati ih na ekranu.

Ulaz: duljina stranice a

Obrada: $P=a^2$
 $O=4*a$

Izlaz: površina kvadrata P
opseg kvadrata O

Poboljšanje:

Opišimo vrijednosti na izlazu. Koristit ćemo naredbu **print**

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

Help on built-in function print in module builtins:

```
print(...)
```

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

Prints the values to a stream, or to `sys.stdout` by default.

Optional keyword arguments:

`file`: a file-like object (stream); defaults to the current `sys.stdout`.

`sep`: string inserted between values, default a space.

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

Zadatak:

Napiši program koji će na temelju zadane duljine stranice kvadrata a izračunati njegovu površinu P i opseg O . Vrijednosti P i O treba ispisati ih na ekranu.

Ulaz: duljina stranice a

Obrada: $P = a^2$

$O = 4 * a$

Izlaz: površina kvadrata P
opseg kvadrata O

Poboljšanje:

Opišimo vrijednosti na izlazu. Koristit ćemo naredbu **print**

```
a = 2
```

```
P = a * a
```

```
O = 4 * a
```

```
print ('Površina kvadrata je', P)
```

```
print ("Opseg kvadrata je",O)
```

```
>>>
```

```
Površina kvadrata je 4
```

```
Opseg kvadrata je 8
```

```
>>>
```

Zadatak:

Napiši program koji će na temelju zadane duljine stranice kvadrata a izračunati njegovu površinu P i opseg O. Vrijednosti P i O treba ispisati ih na ekranu.

Ulaz: duljina stranice a

Obrada: $P=a^2$

$O=4*a$

Izlaz: površina kvadrata P
opseg kvadrata O

Poboljšanje:

Omogućimo unos vrijednosti duljine stranice a s tipkovnice pomoću naredbe **input**

Naredba **input** omogućava unos **stringa** sa standardnog ulaza

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

Help on built-in function input in module builtins:

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

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

Read a string from standard input.
itd...

Zadatak:

Naredba **input** omogućava unos **stringa** sa standardnog ulaza (tipkovnice)

Ulaz: duljina stranice a

Poboljšanje:

Omogućimo unos vrijednosti duljine stranice a s tipkovnice pomoću naredbe **input**

Napomena:

Duljina stranice nije string već realni broj!
(Dolazi do greške u programu)

```
a = input ()  
#kontrola unosa  
print (a)
```

```
P = a * a  
O = 4 * a
```

```
print ('Površina kvadrata je', P)  
print ("Opseg kvadrata je",O)
```

Zadatak:

```
>>>
```

```
3
```

```
3
```

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

```
File
```

```
"C:\Python32\Kvadrat_3.py",  
line 5, in <module>
```

```
    P = a * a
```

```
TypeError: can't multiply  
sequence by non-int of type  
'str'
```

```
>>>
```

```
a = input ()  
#kontrola unosa  
print (a)
```

```
P = a * a
```

```
O = 4 * a
```

```
print ('Površina kvadrata je', P)
```

```
print ("Opseg kvadrata je",O)
```

Zadatak:

```
>>>
```

```
Upiši duljinu stranice kvadrata  
=3.5
```

```
Duljina stranice kvadrata = 3.5
```

Traceback (most recent call
last):

```
File  
"C:/Python32/Kvadrat_3a.py",  
line 5, in <module>
```

```
P = a * a
```

```
TypeError: can't multiply  
sequence by non-int of type  
'str'
```

```
>>>
```

```
a = input ("Upiši duljinu stranice kvadrata =")  
#kontrola unosa  
print ("Duljina stranice kvadrata =",a)
```

```
P = a * a
```

```
O = 4 * a
```

```
print ('Površina kvadrata je', P)
```

```
print ("Opseg kvadrata je",O)
```

Zadatak:

>>>

Upiši duljinu stranice kvadrata
=3.5

Duljina stranice kvadrata = 3.5

Površina kvadrata je 12.25

Opseg kvadrata je 14.0

>>>

```
a = input("Upiši duljinu stranice kvadrata =")
```

```
a = float(a)
```

```
P = a * a
```

```
O = 4 * a
```

```
print('Površina kvadrata je', P)
```

```
print("Opseg kvadrata je",O)
```

Zadatak:

Napiši program koji će na temelju zadane duljine stranice kvadrata **a** izračunati njegovu površinu **P** i opseg **O**. Vrijednosti **P** i **O** treba ispisati ih na ekranu.

```
>>>
```

```
Upiši duljinu stranice kvadrata  
=3.5
```

```
Površina kvadrata je 12.25
```

```
Opseg kvadrata je 14.0
```

```
>>>
```

```
a = float(input("Upiši duljinu stranice kvadrata ="))
```

```
P = a * a
```

```
O = 4 * a
```

```
print ('Površina kvadrata je', P)
```

```
print ("Opseg kvadrata je",O)
```

Dodatna poboljšanja

```
a = float(input("Upiši duljinu stranice kvadrata u metrima "))
```

```
P = a * a
```

```
O = 4 * a
```

```
print('Površina kvadrata je', P, "m četvornih")
```

```
print("Opseg kvadrata je", O, "m")
```

```
>>>
```

```
Upiši duljinu stranice kvadrata u metrima 3.5
```

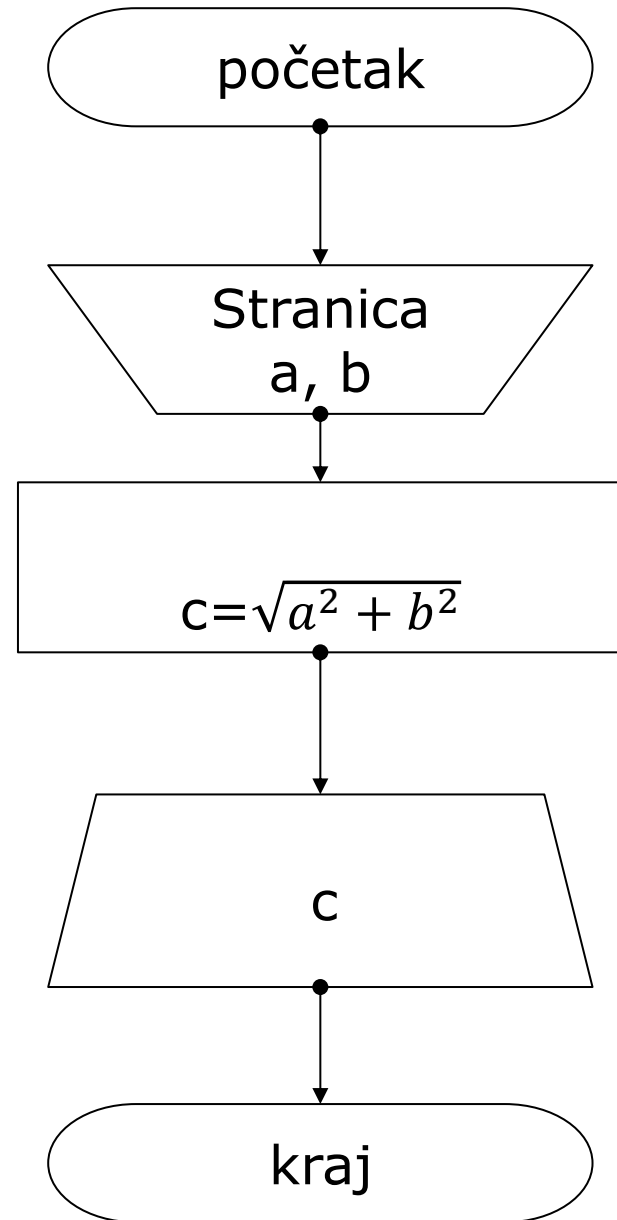
```
Površina kvadrata je 12.25 m četvornih
```

```
Opseg kvadrata je 14.0 m
```

```
>>>
```


Zadatak:

Nacrtaj dijagram toka za program koji će na temelju zadane duljine kateta **a** i **b** izračunati duljinu hipotenuze **c**. Vrijednosti **c** treba ispisati ih na ekranu.



Zadatak:

Napiši program koji će na temelju zadane duljine kateta **a** i **b** izračunati duljinu hipotenuze **c**. Vrijednosti **c** treba ispisati ih na ekranu.

```
>>>
```

```
Upiši duljinu katete a=3
```

```
Upiši duljinu katete b=4
```

```
duljina hipotenuze je= 5.0
```

```
>>>
```

```
# ulazni podaci
a=input('Upiši duljinu katete a=')
b=input('Upiši duljinu katete b=')

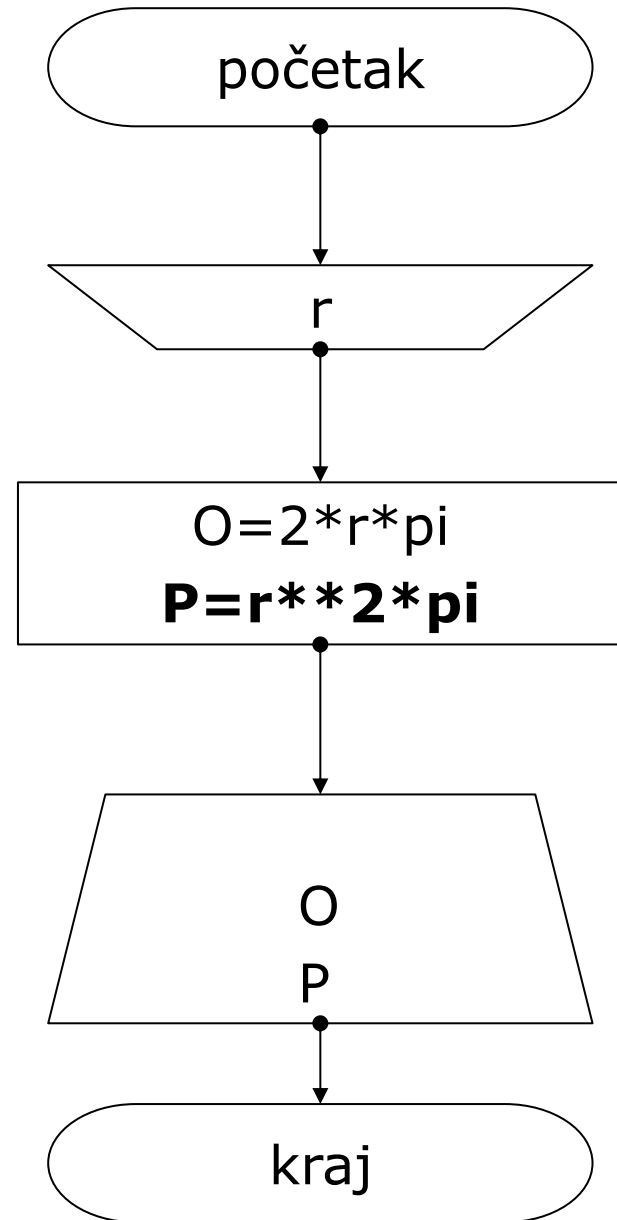
# pretvaranje stringa u realni broj
a=float(a)
b=float(b)

#izračunavanje hipotenuze
from math import *
c=sqrt(a**2+b**2)

#ispis (izlaz) podataka
print()
print ('Duljina hipotenuze je=', c)
```

Zadatak:

Nacrtaj dijagram toka za program koji će na temelju zadanog radijusa kružnice **r** izračunati njen opseg **O** i površinu kruga **P**



Zadatak:

Napiši program koji će upitati korisnika kako mu je ime i koliko ima godina. Nakon toga će ispisati poruku:
(Ime korisnika) drago mi je da imaš (broj godina) godina.

```
>>>
```

```
Kako se zoveš? Zlatan  
Koliko imaš godina? 57
```

```
Zlatan drago mi je da imaš  
57.0 godina
```

```
>>>
```

```
ime=input('Kako se zoveš? ')  
godine=input('Koliko imaš godina? ')  
  
#godine=float(godine)
```

```
print()  
print (ime, 'drago mi je da imaš',  
godine, 'godina.')
```

Zadatak:

Napiši program koji će na temelju zadanih duljina stranica pravokutnika **a** i **b** izračunati njegovu površinu **P**, opseg **O** i duljinu dijagonale **d**. Vrijednosti **a** i **b** treba unijeti s tipkovnice u metrima, a vrijednosti **P**, **O** i **d** ispisuju se na ekranu u odgovarajućim mjernim jedinicama.

```
a=input ("upiši duljinu stranice a u metrima ")  
b=input ("upiši duljinu stranice b u metrima ")
```

```
a=float(a)  
b=float(b)
```

```
P=a*b  
O=2*a+2*b  
d=(a**2+b**2)**(1/2)
```

```
print ()  
print ('Površina pravokutnika je=', P, 'metara kvadratnih')  
print ('Opseg pravokutnika je=', O, 'metara kvadratnih')  
print ('Dijagonala pravokutnika je=', d, 'metara')
```

RELACIJSKI OPERATORI :

- Relacijski operatori uspoređuju dva operanda
- Rezultat usporedbe ima vrijednost *True* ili *False*

OPERATOR	ZNAČENJE SIMBOLA
>	veće od
<	manje od
>=	veće ili jednako
<=	manje ili jednako
==	jednako
!=	nije jednako (različito od)

RELACIJSKI OPERATORI :

- Relacijski operatori uspoređuju dva operanda
- Rezultat usporedbe ima vrijednost *True* ili *False*

- $>>> 3>2$
- *True*
- $>>> 2>3$
- *False*
- $>>> 4<5$
- *True*
- $>>> 4>5$
- *False*
- $>>> 4>=4$
- *True*
- $>>> 4>=2$
- *True*
- $>>> 5<=5$
- *True*
- $>>> 5<=6$
- *True*
- $>>> 7==7$
- *True*
- $>>> 7==8$
- *False*
- $>>> 8!=8$
- *False*
- $>>> 8!=9$
- *True*
- $>>>$

Donošenje odluka u programima:

- *ako je, onda i inače* (engl. *if, then, else*)
- *ključne riječi u Pythonu:*
- **if, else i elif,**

```
print ('Program će ispitati je li broj koji unesete veći ili manji od 10')
```

```
broj=int(input('Unesi broj'))
```

```
if broj > 10:
```

```
    print ('Broj je veći od deset')
```

```
else:
```

```
    print ('Broj je manji od deset')
```

Pitanje: što će biti ako upišemo broj 10?

Donošenje odluka u programima:

- ako je, onda i inače
(engl. *if, then, else*)
- ključne riječi u Pythonu:
- **if, else i elif,**

```
print ('Proram će ispitati je li broj koji unesete veći ili manji od 10')
broj=int(input('Unesi broj '))
if broj > 10:
    print ('Broj je veći od deset')
#ispituje da li je broj jednak 10
elif broj==10:
    print('Broj je jednak 10')
else:
    print ('Broj je manji od deset')
```

Donošenje odluka u programima:

- ako je, onda i inače
(engl. *if, then, else*)
- ključne riječi u Pythonu:
- **if, else i elif,**

Napiši program koji će za neki broj (unesen s tipkovnice) ispitati da li je pozitivan ili negativan. Rezultate ispitivanja treba ispisati na ekranu.

Donošenje odluka u programima:

- ako je, onda i inače
(engl. *if, then, else*)
- ključne riječi u Pythonu:
- **if, else i elif,**

Napiši program koji će za neki broj (unesen s tipkovnice) ispitati da li je paran ili neparan. Rezultate ispitivanja treba ispisati na ekranu.

(Napomena: parni brojevi su oni koji su djeljivi s 2 bez ostatka)

Donošenje odluka u programima:

- ključne riječi u Pythonu:
- **if, else i elif,**

Napiši program koji će za prirodni broj upisan putem tipkovnice provjeriti parnost.

(Napomena: parni brojevi su oni koji su djeljivi s 2 bez ostatka)

```
n=int(input("Upiši prirodni broj "))
if n%2==0:
    print("Broj je paran!")
else:
    print("Broj je neparan")
```

Donošenje odluka u programima:

- ključne riječi u Pythonu:
- **if, else i elif,**

```
broj = 25
pretpostavka = int(input('unesite broj: '))
if pretpostavka == broj:
    print('Bravo, pogodili ste.')
    print('(ali to ne znaci da ste genije!')
elif pretpostavka < broj:
    print('Ne, broj je veci')
else:
    print('Ne, broj je manji')
print('Gotovo')
```

Petlje:

- ključne riječi u Pythonu:
- **for in range**

```
print ('Ispisujemo brojeve redom:')  
for n in range (1,15):  
    print (n)
```

Pitanje:

Zašto je niz stao na 14?

Ispisujemo brojeve redom:

1
2
3
4
5
6
7
8
9
10
11
12
13
14

Petlja:

- ključne riječi u Pythonu:
- **for in range**

Napiši program koji će ispisati sve cijele brojeve od 1 do 20

```
print ('Ispisujemo brojeve redom:')  
for kontrolni_broj in range (1,21):  
    print (kontrolni_broj)
```

Petlja:

- ključne riječi u Pythonu:
- **for in range**

Napiši program koji će ispisati sve cijele brojeve od donje do gornje granice koju zadajemo putem tipkovnice

```
dg=int(input('Upiši donju granicu ranga: '))
gg=int(input('Upiši gornju granicu ranga: '))
print ('Ispisujemo brojeve redom:')
for kontrolni_broj in range (dg,gg+1):
    print (kontrolni_broj)
print ('Kraj programa')
```


Petlje:

- ključne riječi u Pythonu:
- **for in range**

Ispis svih parnih brojeva u rasponu ...

```
for broj in range(1,10):
    print (broj*2)
else:
    print ('Prikazani su svi parni brojevi od 1 do 20')
```

```
for broj in range(1,11):
    print (broj*2)
else:
    print ('Prikazani su svi parni brojevi od 1 do 20')
```

Napomena:

Drage učenice i dragi učenici,
ova prezentacija nije rađena kao nastavni materijal za samoučenje
već kao pomoćno sredstvo koje smo koristili tijekom predavanja.
Onima koji su propustili dio predavanja ili se žele podsjetiti na neke
dijelove predavanja može poslužiti i kao materijal za samoučenje
odnosno ponavljanje gradiva.

Puno uspjeha i zadovoljstva u programiranju želi vam
Prof. Zlatan Soldo