

# Uvod u programiranje



2023/24



# O kursu

- **Model 70+30**
  - Redovno prisustvo nastavi = 4 poena
  - 2 kolokvijuma = 66 poena
  - Završni deo ispita = 30 poena
- **Uslov za izlazak na završni ispit – više od polovine poena na predispitnim obavezama**

# Najvažnije teme

- Cilj predmeta: programerska pismenost
- Uvod u programiranje
- Osnove programskog jezika Python
  - aritmetičke operacije
  - standardni ulaz/izlaz
  - ugrađene funkcije
  - upravljačke komande, grananje, ponavljanja
  - funkcije, rekurzivne funkcije
  - stringovi
- Strukture - liste, skupovi, torke, rečnici

# Kompjuterski programi

Rešavanje problema korišćenjem kompjuterskih programa se može razložiti na više etapa:

1. Definisanje problema
2. Analiza problema
3. Definisanje algoritma
4. Projektovanje programa
5. Kodiranje
6. Testiranje
7. Analiza rezultata
8. Isporuka programa
9. Održavanje

# Šta je algoritam?

- Algoritmi su korišćeni davno pre nego što je iko pokušao da definiše šta je to algoritam.

## Primeri:

- Euklidov postupak za konstruisanje pomoću lenjira i šestara
- Euklidov postupak za određivanje NZD
- Vavilonski iterativni postupak za određivanje kvadratnog korena (1500 g.p.n.e.).

# Šta je algoritam?

## Postupak za određivanje kvadratnog korena

- Da biste odredili kvadratni koren bilo kog pozitivnog broja  $S$ :
  - 1. Napravite početnu pretpostavku.** Pretpostavite da je vrednost kvadratnog korena od  $S$  neki pozitivni broj  $x_0$ .
  - 2. Poboljšajte pretpostavljeni rezultat.** Primenite formulu  $x_1 = (x_0 + S / x_0) / 2$ . Broj  $x_1$  je bolja aproksimacija za  $\sqrt{S}$ .
  - 3. Ponavljajte korak 2 dok ne konvergirate ka rešenju.** Primenjujte formulu  $x_{n+1} = (x_n + S / x_n) / 2$  sve dok se cifre brojeva  $x_{n+1}$  i  $x_n$  ne budu poklapale do u željenu decimalu.

# Algoritmi

Reč algoritam, onako kako je mi danas poznajemo, prvi je upotrebio persijski matematičar Musa Al Horezmi pre više od 1200 godina.

- Konačan uređen niz precizno formulisanih pravila kojima se rešava jedan ili čitava klasa problema

Može se opisati tekstualno na prirodnom jeziku ili grafički



# Zadatak

- Opisati algoritam za zamenu probušene gume na autu





# Zadatak

## Algoritam

- Probušena guma na autu:

Potrebno: auto

Ako je guma probušena onda

Potrebno: rezervni točak, dizalica, ključ

Podići auto dizalicom

Ako ima neodšrafljenih šrafova

Odšrafiti šraf

Skinuti točak

Staviti rezervni točak

Ako ima nezašrafljenih šrafova

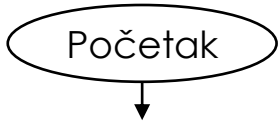
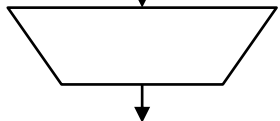
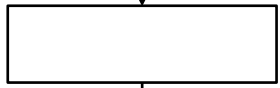
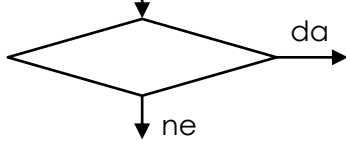
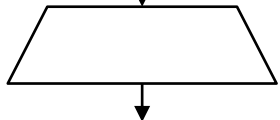
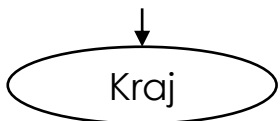
Zašrafiti šraf

Spustiti auto

Spakovati alat

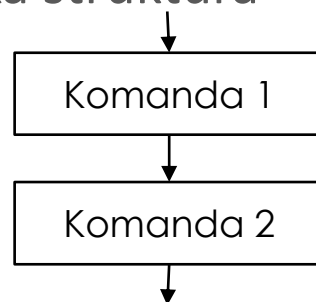
Srećan put

# Alogitmi – grafički simboli

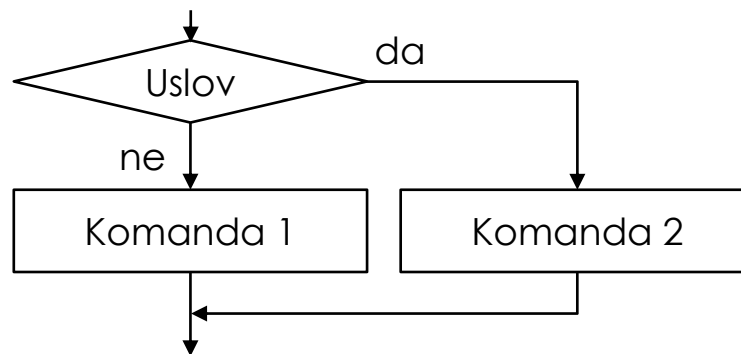
	Početak algoritma
	Ulazne veličine
	Izračunavanje
	Uslovno grananje
	Izlazne veličine
	Kraj algoritma

# Alogitmi – osnovne strukture

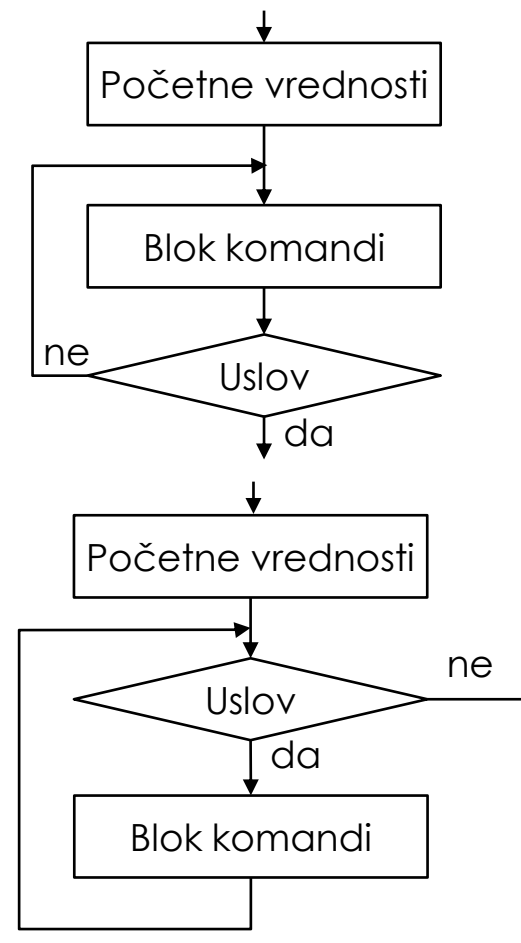
- Linijska struktura



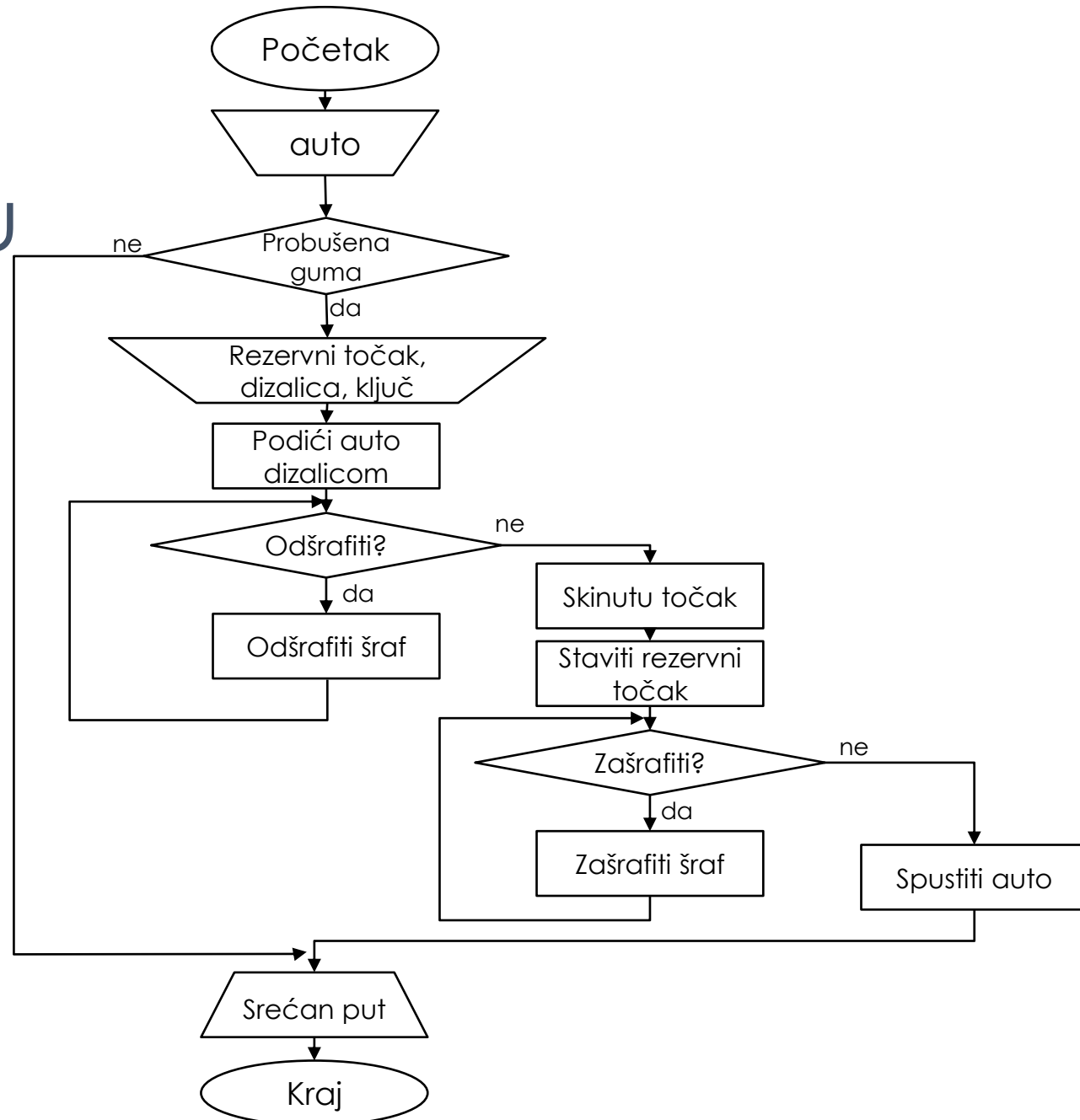
- Razgranata struktura



- Ciklička struktura



# Algoritmi: probušena guma na autu



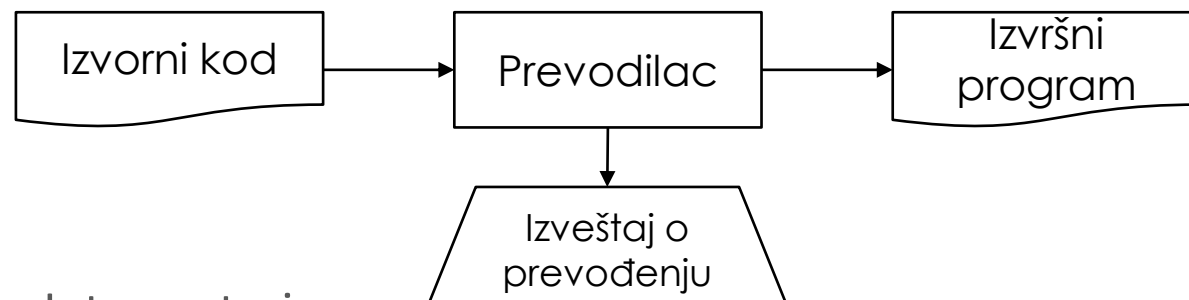
# Vrste programskih jezika

- Algoritmi se u računaru realizuju odgovarajćim **programom**
- Program može biti realizovan korišćenjem različitih simbola (jezika), kojima se reprezentuju pojedine operacije podržane od strane računarskog sistema
- Prema stepenu bliskoski sa arhitekturom računara
  - Jezici niskog nivoa (mašinski jezik, Assembler)
  - Jezici visokog nivoa (FORTRAN, Pascal, C, C++, C#, Java...)

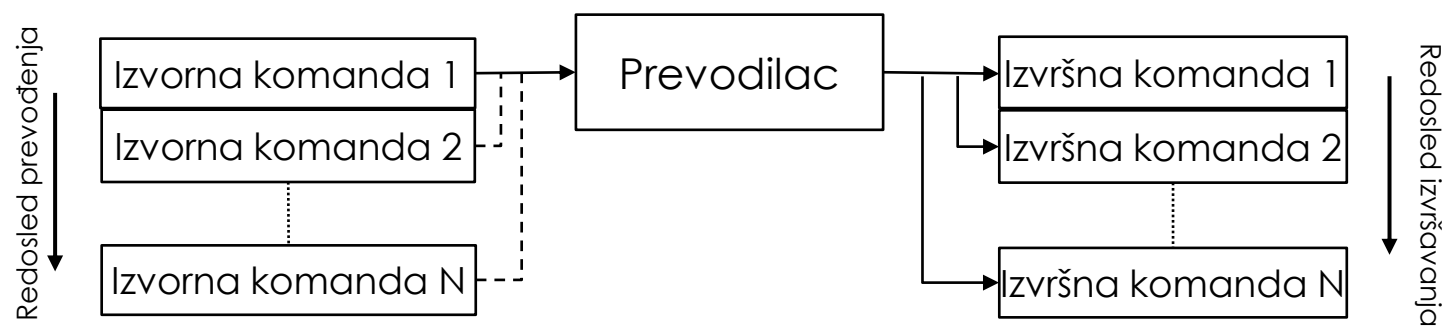
# Vrste programskih jezika

- Prema načinu prevođenja

- Kompajleri



- Interpreteri



# Vrste programskih jezika

- Viši programski jezici dele se prema načinu kako se program formira:
  - Imperativni (opisuju kako se nešto izračunava)
  - Deklarativni (opisuju šta treba izračunati) – Prolog i SQL
- Program prilikom izvršavanja prolazi kroz određena stanja koja su definisana promenljivim u memoriji. Programer u imperativnom jeziku određuje način na koji pojedine promenljive menjaju svoja stanja, dok se kod deklarativnih jezika ovim pitanjem bave prevodilac ili interpreter, na osnovu opisa željenog rezultata.
- Imperativni jezici se razlikuju prema organizaciji izvornog koda:
  - Proceduralni – Basic, Fortran, Pascal, C
  - Objektno orijentisani – C++, Java, C#
- Funkcionalni pristup programiranju je bliži deklarativnom, gradi se od funkcija - Haskell

# Python





# Python

- Savremeni jezici primenjuju različite principe koji omogućavaju pisanje i organizaciju programa na različite načine.
- Python omogućava:
  - **Proceduralno**
  - Objektno
  - Kombinacija proceduralnog i objektnog
- Python podržava i pojedine elemente funkcionalnog programiranja

# Zašto Python?

- Osmislio ga je holandski programer Guido van Rossum krajem osamdesetih godina prošlog veka
- Python je interpretirani, objektno-orijentisani jezik visokog nivoa, namenjen za pravljenje svih vrsta aplikacija – od inženjerskih i naučnih, do poslovnih i web primena
- Pravila jezika su jednostavna, a izvorni kod je čitak i često dosta kraći negu u slučaju Java ili C/C++ ekvivalenta.
- Dostupan je za različite operative sisteme: Windows, Linux, Mac
- Python stavlja na raspolaganje kvalitetne standardne biblioteke, koji nude gotova rešenja za mnoge probleme.

# Python

- Na adresi <https://www.python.org/downloads/> možete preuzeti instalacioni fajl programskog jezika Python za odgovarajući operativni sistem. Trenutno aktuelna verzija je 3.12.2 pa će postupak instalacije biti prikazan na osnovu nje. Pokrenuti instalacioni fajl, pojaviće se prozor sa slike:

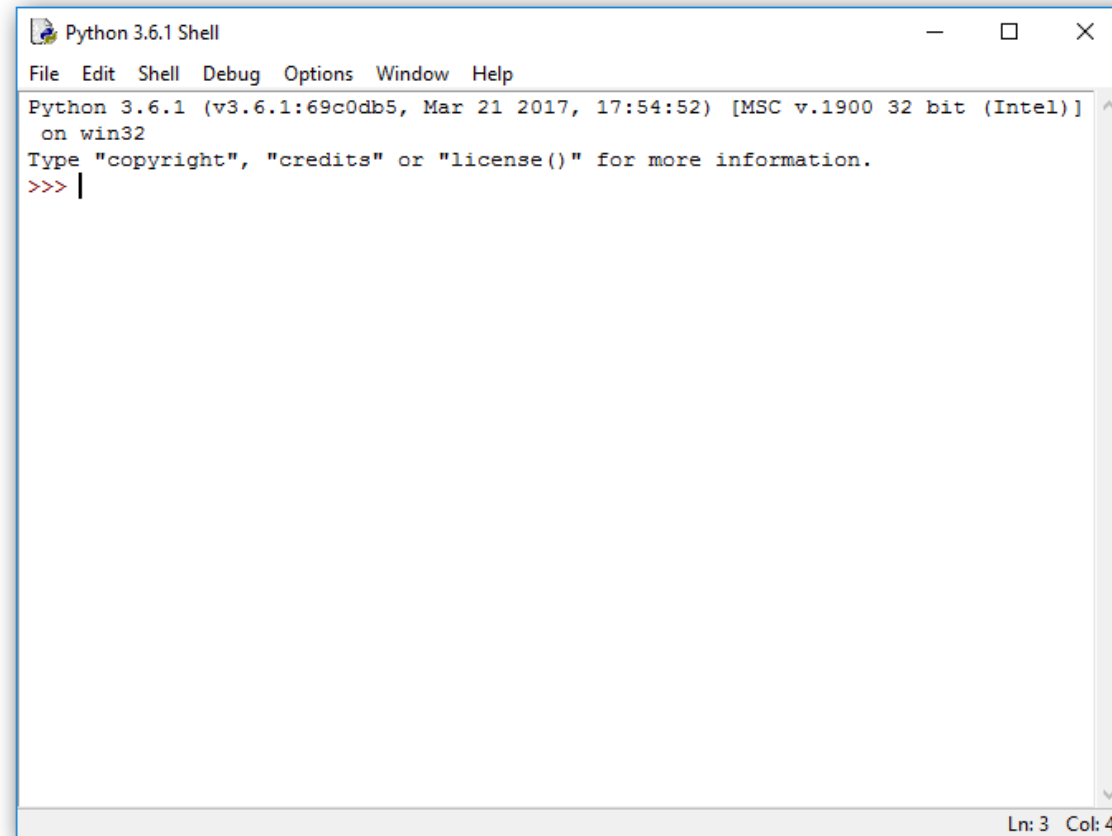


# Python

- Programi se pišu uz pomoć odgovarajućeg softverskog paketa pod imenom **razvojno okruženje**
- Razvojno okruženje čini skup softverskih alata pomoću kojih se kreira program. U obavezne alate spadaju editor, debager, interpreter ili prevodilac i sistem za pomoć.
- Editor – za unos teksta izvornog koda
- Debager – testira rad programa u toku izvršavanja
- Sistem za pomoć – omogućava pregled dokumenata koja pojašnjavaju korišćenje

# Python

- Uz osnovnu instalaciju dobija se besplatno razvojno okruženje IDLE, koje uključuje interaktivni interpreter, editor teksta i alate za testiranje koda
- Nakon pokretanja, otvara se sledeći prozor:



The screenshot shows a window titled "Python 3.6.1 Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area displays the following information:

```
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)]  
on win32  
Type "copyright", "credits" or "license()" for more information.  
>>> |
```

The status bar at the bottom right of the window shows "Ln: 3 Col: 4".



# Python

- Simbol `>>>` označava tekuću liniju za unos komande koja se, po pritisku na taster `<Enter>`, izvršava od strane interpretera, a njen rezultat ispisuje u sledećem redu.

```
>>> 2 + 3
```

```
5
```

```
>>> 2 - 3
```

```
-1
```

```
>>> 2 * 3
```

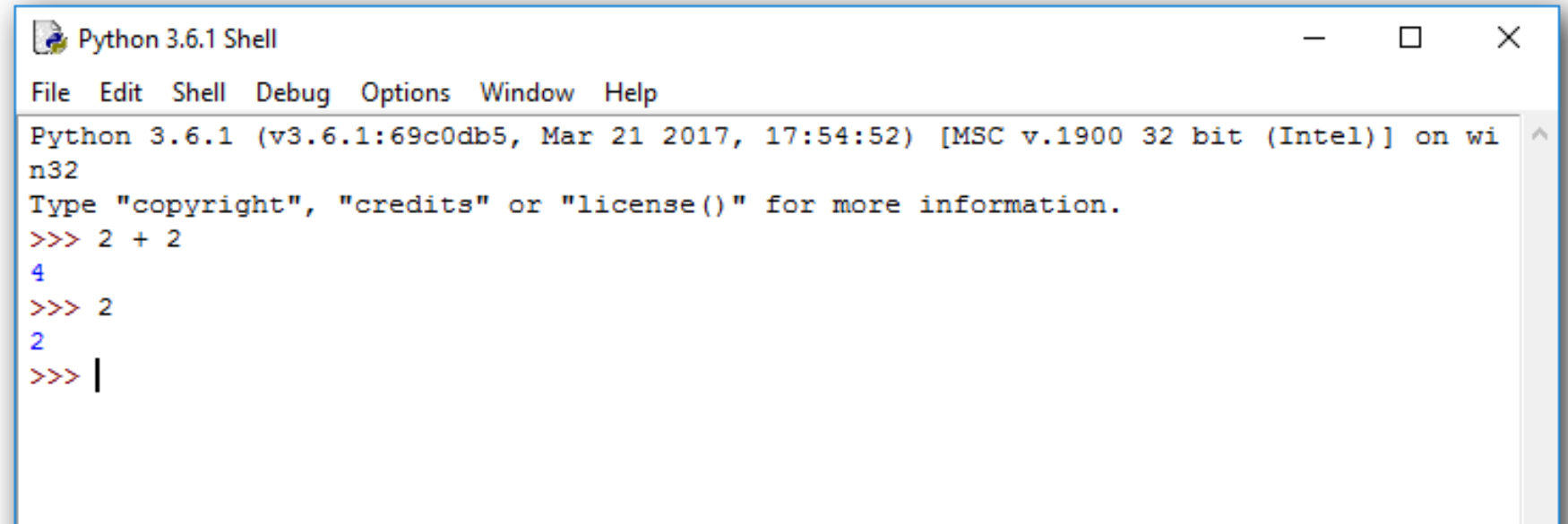
```
6
```

```
>>> 9 / 4
```

```
2.25
```

kao kalkulator

# Python



```
Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)] on wi
n32
Type "copyright", "credits" or "license()" for more information.
>>> 2 + 2
4
>>> 6 +
SyntaxError: invalid syntax
>>> |
```

- U slučaju da smo napravili grešku u kucanju, ispisaće nam se poruka o grešci:

```
>>> 6 +
```

```
SyntaxError: invalid syntax
```

```
>>>
```

- Kao što se može videti, pogrešan deo koda je automatski uklonjen i možemo nastaviti sa radom.

# Python

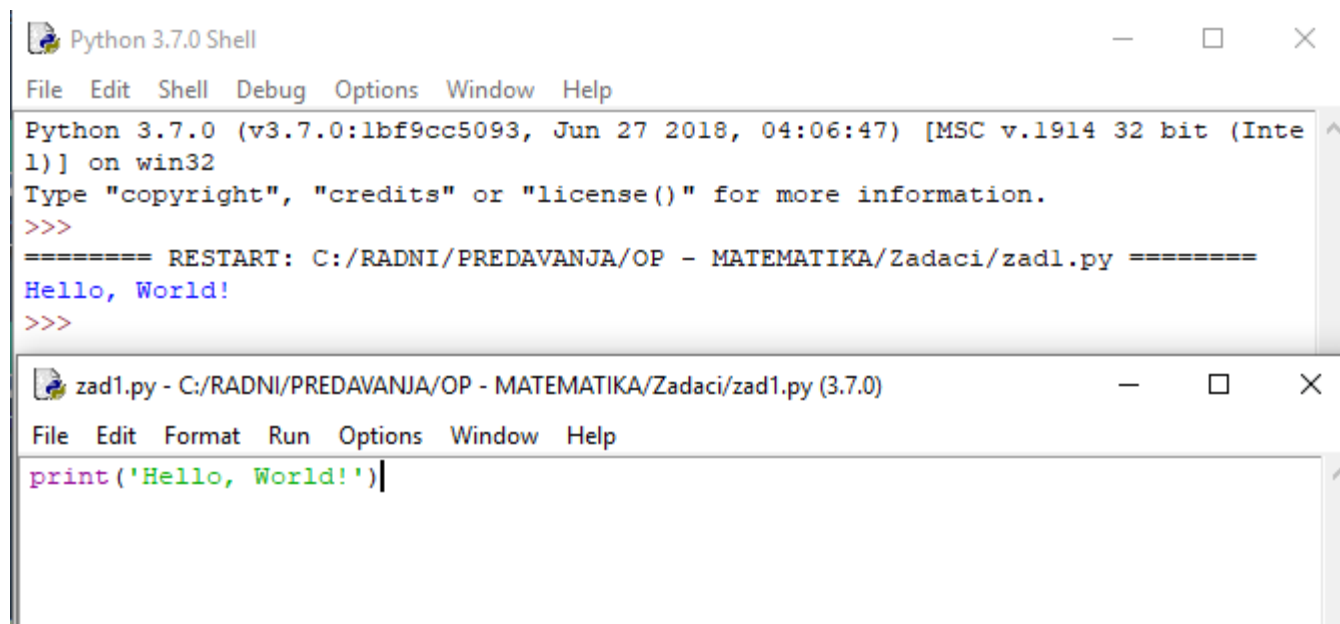
- Rad u interaktivnom okruženju nije pogodan za izvršavanje složenijeg postupka jer se naredbe koje ga realizuju, moraju zapamtiti i ponoviti uvek u istom redosledu.
- Zato se naredbe grupišu u program koji je zapisan u datoteci na disku.
- Program se može pokrenuti, kako iz IDLE-a tako i van njega, na za to predviđene načine.
  - Programski kod se zapisuje u fajl – skript
  - Ekstenzija fajla je `.py`
  - Interpreter izvršava kod



# Prvi program

- Testiranje rada programa podrazumeva da se, uz pomoć editora u IDLE-u, prvo formira programska datoteka.
- Editor se pokreće iz menija **File**, u okviru osnovnog prozora razvojnog okruženja, izborom opcije **New File** za novi, odnosno **Open...**, za otvaranje postojećeg programa

Po snimanju na disk, u datoteku sa ekstenzijom **.py** (u primeru **zad1.py**), program se pokrece izborom opcije **Run Module**, iz menija **Run**.



The image shows two overlapping windows from the Python 3.7.0 IDLE environment. The top window is titled "Python 3.7.0 Shell" and displays the following text:

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/RADNI/PREDAVANJA/OP - MATEMATIKA/Zadaci/zad1.py =====
Hello, World!
>>>
```

The bottom window is titled "zad1.py - C:/RADNI/PREDAVANJA/OP - MATEMATIKA/Zadaci/zad1.py (3.7.0)" and shows the source code for the script:

```
print('Hello, World!')
```

# Promenljive

- Ime promenljive sme sadržati samo
  - Brojeve
  - Velika i mala slova
  - Donju crticu \_
- Ne sme biti ključna reč
- Ime ne sme početi brojem
- Python razlikuje velika i mala slova

and def exec if not return  
assert del finally import or try  
break elif for in pass while  
class else from is print yield  
continue except global lambda raise

Neka od dozvoljenih imena:

`jabuka, slova123, A4, ime_psa`

Neka od nedozvoljenih imena:

`ime_psa, 123slova, jabuka#`

# Vrste grešaka

- Syntax Error

```
>>> 1a = 100
```

```
SyntaxError: invalid syntax
```

```
>>> a = 3 * 2 +
```

```
SyntaxError: invalid syntax
```

```
>>>
```

- Runtime Error – greška se javlja tek kada se pokrene program

- Semantic Error

```
>>> x = 5 + "Praktikum"
```

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

```
  File "<pyshell#15>", line 1, in <module>
```

```
    x = 5 + "Praktikum"
```

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

# Promenljive

```
>>> a = 1
>>> b = 2
>>> c = 2.5
>>> a
1
>>> c
2.5
>>> x
Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    x
NameError: name 'x' is not defined
>>>
```

# Promenljive

```
>>> poruka = "Programiranje je lepo"  
>>> pi = 3.14  
>>> n = 5  
>>> poruka  
'Programiranje je lepo'  
>>> pi  
3.14  
>>> n  
5
```