



# PYTHON

Osnove i primena na Grid-u

# Agenda

- Kompajler vs Interpreter
- Osnovni tipovi odataka i operatori
- Napredni nivo (liste, recnici, tuple, fajlovi)
- OOP
- Grid

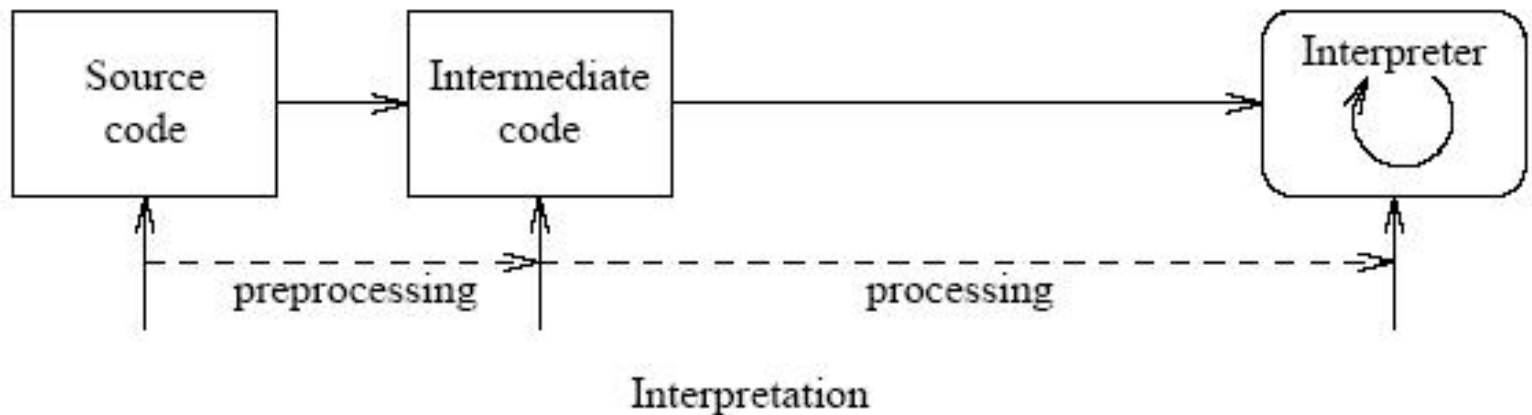
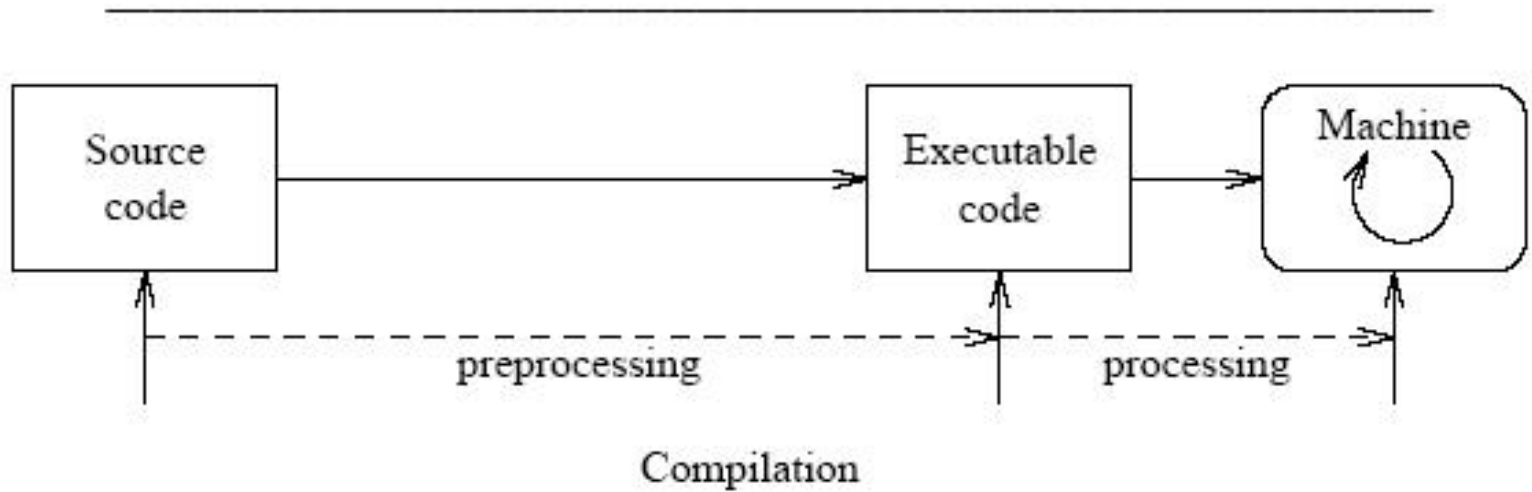
## Kompajler vs Interpreter

Osnovni tipovi odataka i operatori

Napredni nivo

OOP

Grid



# Razlike

Kompajler vs Interpreter

Osnovni tipovi odataka i operatori

Napredni nivo

OOP

Grid

- Komanda se završava na kraju reda, nema **;**
- Blokovi nisu omeđeni zagradama **{ }**
- Svaki blok je uvučen istim brojem **space**
- Ne postoji **main()**

# Tipovi podataka

- Logički
- Numerički (int, long, float, complex)
- Sekvencijalni tipovi (string, list, tuple ...)
- Iterator

Promenljivoj se **NE MORA** dodeliti tip  
podataka

slovo = "a"    i = 5    c = 3+j5    f = 5.8

# Operacije

Kompajler vs Interpreter

**Osnovni tipovi odataka i operatori**

Napredni nivo

OOP

Grid

- $x // y$
- $x ** y$
- is x, is not x
- $x << n$ ,  $x >> n$

# Sekvencijalne operacije

Operation	Result	Notes
<code>x in s</code>	<code>True</code> if an item of <code>s</code> is equal to <code>x</code> , else <code>False</code>	(1)
<code>x not in s</code>	<code>False</code> if an item of <code>s</code> is equal to <code>x</code> , else <code>True</code>	(1)
<code>s + t</code>	the concatenation of <code>s</code> and <code>t</code>	(6)
<code>s * n, n * s</code>	<code>n</code> shallow copies of <code>s</code> concatenated	(2)
<code>s[i]</code>	<code>i</code> th item of <code>s</code> , origin 0	(3)
<code>s[i:j]</code>	slice of <code>s</code> from <code>i</code> to <code>j</code>	(3)(4)
<code>s[i:j:k]</code>	slice of <code>s</code> from <code>i</code> to <code>j</code> with step <code>k</code>	(3)(5)
<code>len(s)</code>	length of <code>s</code>	
<code>min(s)</code>	smallest item of <code>s</code>	
<code>max(s)</code>	largest item of <code>s</code>	
<code>s.index(i)</code>	index of the first occurrence of <code>i</code> in <code>s</code>	
<code>s.count(i)</code>	total number of occurrences of <code>i</code> in <code>s</code>	

# if iskaz

Kompajler vs Interpreter

**Osnovni tipovi odataka i operatori**

Napredni nivo

OOP

Grid

If <uslov 1>:

<blok 1>

elif <uslov 2>:

<blok 2>

...

elif <uslov N>:

<blok N>

else:

<blok N+1>



# while petlja

Kompajler vs Interpreter

**Osnovni tipovi odataka i operatori**

Napredni nivo

OOP

Grid

```
while <uslov>:  
    <blok>  
<blok 2>
```

# for petlja

Kompajler vs Interpreter

**Osnovni tipovi odataka i operatori**

Napredni nivo

OOP

Grid

for *šta* in *gde*:

<blok>

<blok>

# Primer

Kompajler vs Interpreter

**Osnovni tipovi odataka i operatori**

Napredni nivo

OOP

Grid

# Stringovi

Kompajler vs Interpreter

Osnovni tipovi odataka i operatori

**Napredni nivo**

OOP

Grid

- Nizovi karaktera u određenom poretku.

# Liste i tuple

Kompajler vs Interpreter

Osnovni tipovi odataka i operatori

**Napredni nivo**

OOP

Grid

- Lista je niz elemenata u određenom poretku.
- Tuple je lista koja se ne može menjati nakon kreiranja

# Dictionaries

Kompajler vs Interpreter

Osnovni tipovi odataka i operatori

**Napredni nivo**

OOP

Grid

- Predstavljaju kolekciju neuređenih vrednosti kojima se pristupa pomoću ključa
- Jedan element čine par

**{ ključ : vrednost }**

# Funkcije

Kompajler vs Interpreter

Osnovni tipovi odataka i operatori

**Napredni nivo**

OOP

Grid

```
def ime_funkcije(argumenti):  
    <telo funkcije>  
return izraz,vrednost...
```

# Primer

Kompajler vs Interpreter

Osnovni tipovi odataka i operatori

**Napredni nivo**

OOP

Grid



# OOP

Kompajler vs Interpreter

Osnovni tipovi odataka i operatori

Napredni nivo

**OOP**

Grid

- Klase i objekti
- Atributi (polja) i metode
- Private, public i static
- Specijalne metode
- self i other

# Klase i objekti

Kompajler vs Interpreter

Osnovni tipovi odataka i operatori

Napredni nivo

**OOP**

Grid

```
class ime_klase (ime_osnovne_klase):  
    <telo klase>
```

```
objekat = ime_klase()
```

# Polja i metode

Kompajler vs Interpreter

Osnovni tipovi odataka i operatori

Napredni nivo

**OOP**

Grid

- Polja čuvaju informacije koje su karakteristične za pojedinačne objekte
- Metode predstavljaju operacije koje se mogu izvršiti nad objektom, definišu se kao i funkcije

# Pravo pristupa

- Polja kojima korisnik može pristupiti su tipa **public**. To su sva polja i metode koja nisu secijalno naznačena
- Polja kojima se ne može pristuiti su tipa **private**. Polja i metode su tipa private ako njihova imena poćinju sa **\_\_**
- Polja tipa static su zajednička za sve objekte jedne klase. Da bi se koristila nije neophodno imati objekat klase.

**staticmethod()**

# Specijalne metode

Kompajler vs Interpreter

Osnovni tipovi odataka i operatori

Napredni nivo

**OOP**

Grid

- `__init__()` – konstruktor
- `__str__()` – konverzija u string
- `__cmp__()` - komparacija

# self i other

Kompajler vs Interpreter

Osnovni tipovi odataka i operatori

Napredni nivo

**OOP**

Grid

- **self** označava da se polje ili metoda iz date klase.
- **other** označava da se polje ili metoda nasleđene u datu klasu.

# Primer

Kompajler vs Interpreter

Osnovni tipovi odataka i operatori

Napredni nivo

**OOP**

Grid

# Primer

Kompajler vs Interpreter

Osnovni tipovi odataka i operatori

Napredni nivo

OOP

**Grid**



# LINKOVI

<http://www.greenteapress.com/thinkpython/thinkCSpy/thinkCSpy.pdf>

<http://docs.python.org/tutorial/index.html>

[http://en.wikibooks.org/wiki/Python\\_Programming](http://en.wikibooks.org/wiki/Python_Programming)

<http://docs.python.org/library/>

<http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-00-introduction-to-computer-science-and-programming-fall-2008/index.htm>

<http://www.python.org/>