Python Workshop
Session One

Introduction to Python

Alireza Gerami University of Bojnord Esfandiar Pourtahmasb

Overview

- Syllabus
- What is Python?
- What can Python do?
- Why Python?

Syllabus

- Introduction
  - Interpreter vs. Compiler
  - Setup Python and Editor
  - Basic Syntax
  - Variable and Data Types
  - Operator
  - Input and Output
- Conditional Statements
  - If else & elif
- Looping
  - For & While
- Control Statements
  - Continue & Break & Pass
- String
  - Accessing Strings
  - Basic Operations
  - Slicing
  - Methods
- Data structures
  - List
  - Set
  - Tuple
  - Dictionaries
- Functions
  - Defining
  - Arguments
  - Global and local variables
  - Lambda
  - Built-in Functions
- Modules
  - Importing module
  - Packages
- Files
  - Read and Write
- Exception Handling
- Useful Library

What is Python?

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

Open source programming language

https://github.com/python

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

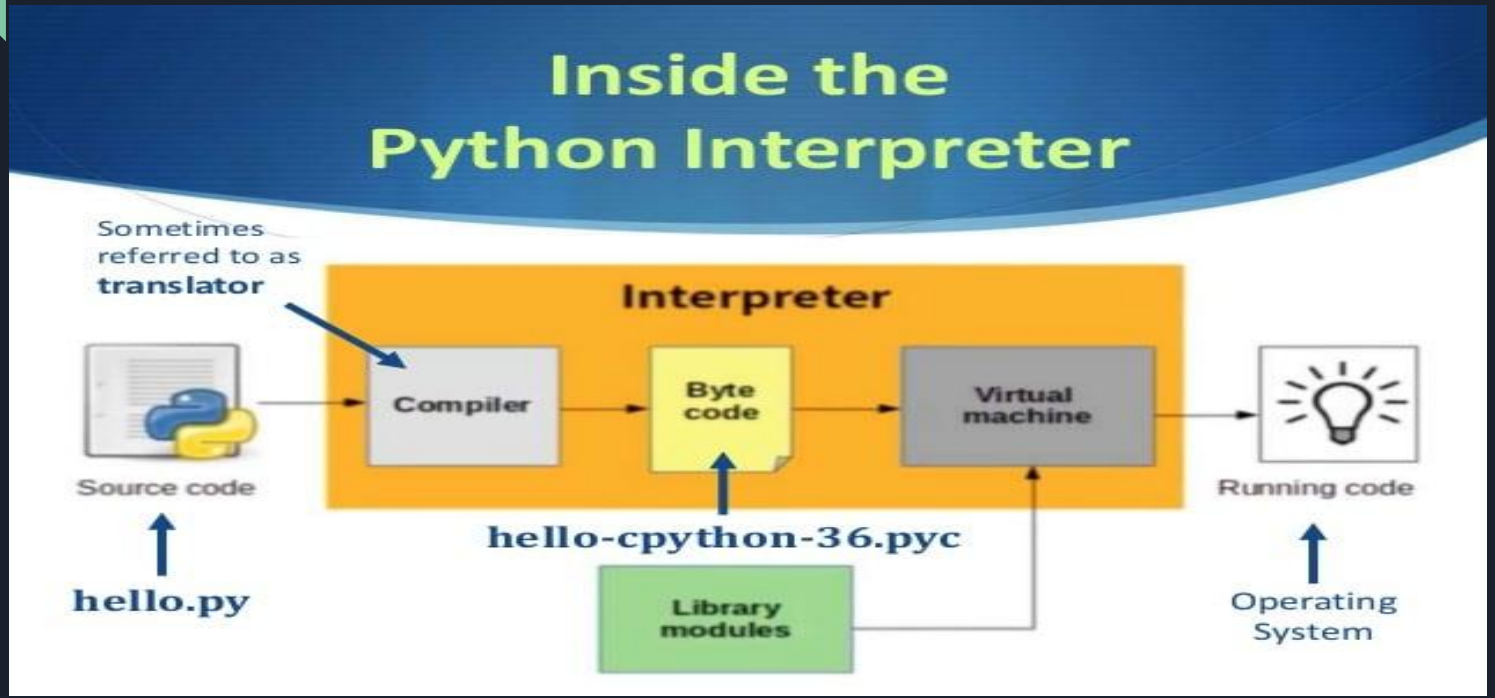
What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.

Interpreter VS Compiler



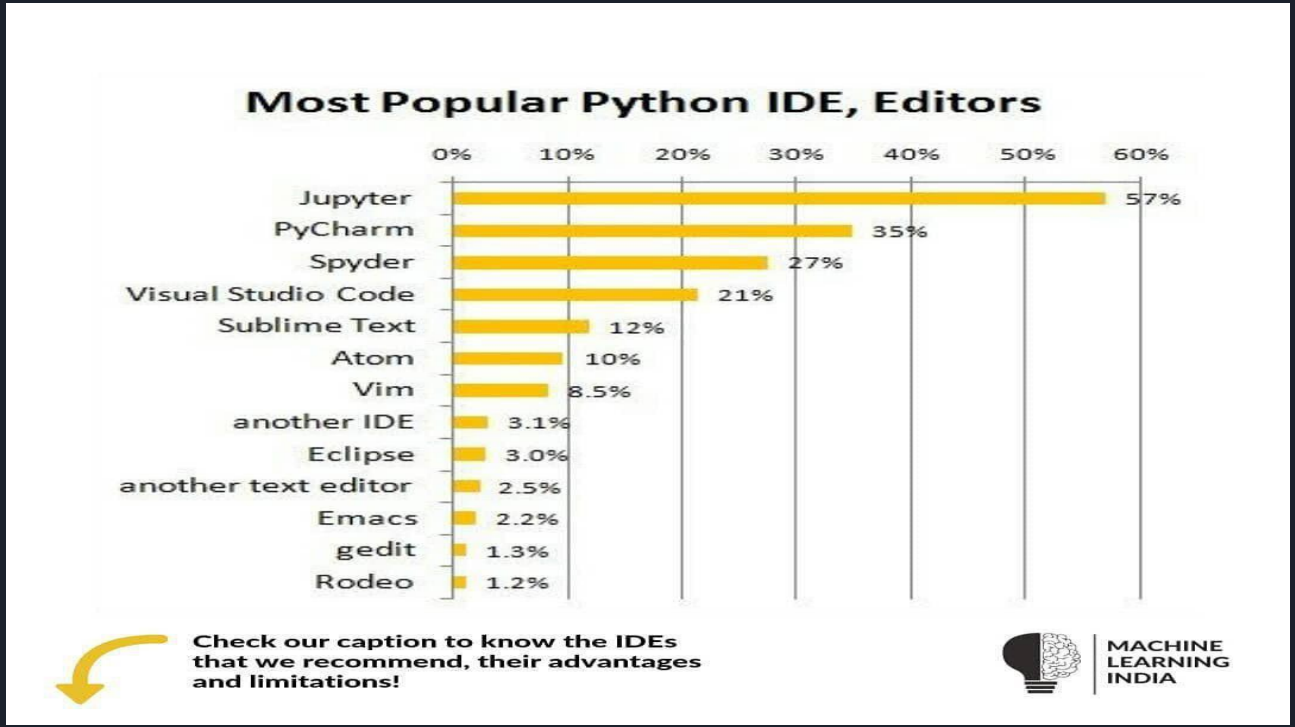
```
a=5

a+=1

print(a)

'b'd\x00Z\x00e\x00d\x017\x00Z\x00e\x01e\x00\x83\x01\x01\x00d\x02S\x00
```

Install Python and Editor


Most Popular Python IDE, Editors
0%
10%
20%
30%
40%
50%
60%
Jupyter
57%
PyCharm
35%
Spyder
27%
Visual Studio Code
21%
Sublime Text
12%
Atom
10%
Vim
8.5%
another IDE
3.1%
Eclipse
3.0%
another text editor
2.5%
Emacs
2.2%
gedit
1.3%
Rodeo
1.2%
Check our caption to know the IDEs that we recommend, their advantages and limitations!
MACHINE LEARNING INDIA

Basic Syntax

Python was designed for readability, and has some similarities to the English language with influence from mathematics.

- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Input & output

First Project

"Hello to Python"

Variables and Data Types

Keywords

Creating Variables

Variable Names

Keywords

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	

Creating Variables

Variables are containers for storing data values. Unlike other programming languages, Python has no command for declaring a variable.

A variable is created the moment you first assign a value to it.

Creating Variables

Variables do not need to be declared with any particular type and can even change type after they have been set.

```
x = 4 # x is of type int

x = "Sally" # x is now of type str
```

Variable Names

A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume). Rules for Python variables:

- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alphabet-numeric characters and underscores (Az, 0-9, and _)
- Variable names are case-sensitive (age, Age and AGE are three different variables)

```
#Legal variable names:
myvar = "John"
my_var = "John"
_my_var = "John"
myVar = "John"
MYVAR = "John"
myvar2 = "John"
```

```
#Illegal variable names:
2myvar = "John"
my-var = "John"
my var = "John"
```

Built-in Data Types

Text Type:	str
Numeric Types:	int, float, complex
Sequence Types:	list, tuple, range
Mapping Type:	dict
Set Types:	set, frozenset
Boolean Type:	bool
Binary Types:	bytes, bytearray, memoryview

Python Operators

Arithmetic operators (+ - * / % ** //)

Assignment operators (= += -= *= /= %= //= **= &= |= ^= >>= <<=)

Comparison operators(== != > < >= <=)

Logical operators (and or not)

Identity operators (is is not)

Membership operators (in not in)

Bitwise operators (& | ^ ~ << >>)