

Python Basics (Part 1)

- SHREYAS S

Agenda

- Programming basics
- History of python
- Overview of python
- Installation and setup
- Introduction to jupyter notebook, colab, Kaggle and github
- Why python and R vs python
- Syntax and comparison with other programming languages
- Comments in python
- Variables in python
- Data types in python (Numbers, Boolean, String, list, tuple, set, dictionary)

What is Programming?

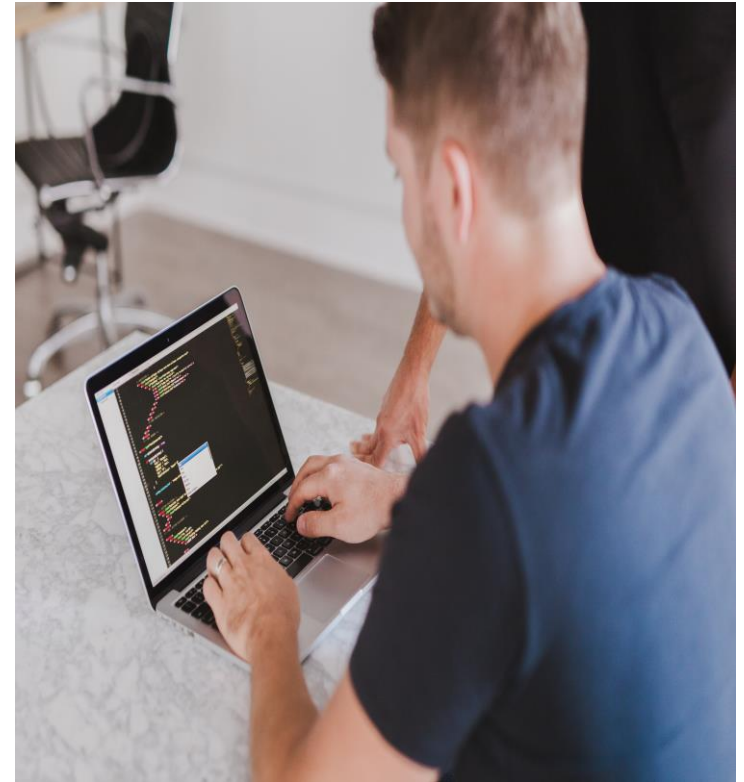
Computer programming is the process of designing and building an executable computer program to accomplish a specific computing result or to perform a specific task.

- *Wikipedia*

Let's simplify the above statement below:

When a series of instructions are provided to the computer to perform a series of task then that's what is called as programming. The series of instruction provided to the computer/machine is called program. The person who writes these instruction is called as programmer or coder.

Let's look at the diagram to understand this concept visually.





Human



Instructions
(Program)



Computer



Programmed
Computer



History of Python

- Python is a high-level programming language. It is widely used for multiple purpose.
- It was designed and developed by Guido Van Rossum in 1991.
- It is the successor of a programming language called ABC programming.
- The main objective behind developing it was to improve code readability and easy syntax.
- The two most used versions of python were 2.x and 3.x.
- The support for python 2.x is no longer provided by its creators from January 1, 2020.
- The current latest stable version of python is 3.8.5

Overview

Python as a programming language can be used in the following areas:

- Software/Application development
- Web development
- Scripting
- Mathematics/Scientific program development
- Handle big data related problems

Overview

Some of the features of python are:

- More readability
- Ease of learning
- Interactive Mode
- Vast availability of libraries
- Extendable
- Portable
- Database support
- Dynamic data types
- Compiled and interpreted internally (unlike other programming languages like java)

Installation and setup

There are two ways to install python:

- 1) By downloading python setup from python.org/downloads/ and following installation instructions based on the operating system.
- 2) Download anaconda from anaconda.com and install it. Python get installed along with anaconda. This way of installing python is recommended because anaconda comes with built-in libraries and IDE for data science. It can be considered as data science toolkit.
- 3) Verify the installation by doing `python --version` on command line.

Introduction to Jupyter notebook

Google colab and Kaggle Kernel

Kaggle and Github

Why Python

- Easy to understand and beginner friendly.
- Compatible with major platforms.
- Vast set of libraries, open source framework and tools.
- Flexibility
- Large community.
- Career opportunities.
- Growth of use in python in fields like AI, ML, Big data, Automation, Cloud computing.

R vs Python

R	Python
Used for data analysis and statistics	Used for writing end to end code for production.
Used by R&D and scholars	Used by programmers and developers
Bit difficult to use in the beginning due to unique syntax	Easy to use and best for beginners.
Need to run locally	Can be integrated with app and deployed
Less number of jobs than python	More number of jobs due to easy code integration feature
Average growth rate	High growth rate
Average salary in India: ₹504,396 (Data analysis using R)	Average salary in India: ₹624,029 (Software Engineer in python)

Syntax

- Syntax in python is dependent on indentation. Indentation are spaces in line of code that defines the code block.
- The syntax of declaring variable is much simpler as compared to other programming languages. In python you don't need to specify the type of variable. You just need to specify the variable name and the value assigned to it.

You will get the idea of syntax as we move further on to the upcoming slides.

Lets check the difference in syntax of python and some other programming languages in the next slide.

Java and C

- Syntax in java for printing hello world.

```
class A {  
    public static void main(String args[]){  
        System.out.println("Hello World");  
    }  
}
```

- Syntax in C for printing hello world.

```
#include <stdio.h>  
  
int main() {  
    printf("Hello, World!");  
    return 0;  
}
```

C++ and C

- Syntax in C++ for printing hello world.

```
// Your First C++ Program
#include <iostream>
int main() {
    std::cout << "Hello World!";
    return 0;
}
```

- Syntax in C for printing hello world.

```
object Test
{
    // Main method
    def main(args: Array[String])
    {
        println("hello World")
    }
}
```


Python syntax for hello world

- Now let's look at how simple and readable it is to write hello world program that prints "hello world" on the screen.

`print("hello world")`

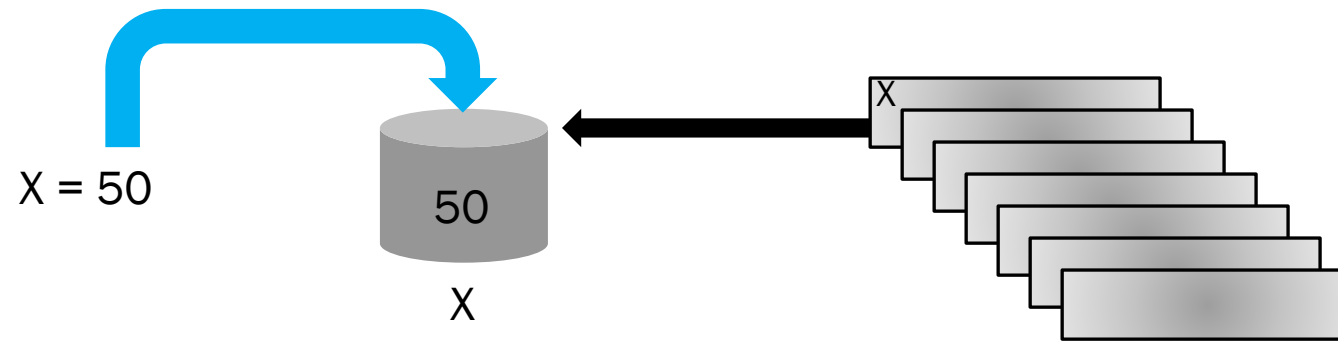
- Now here you can see the difference in the number of lines of code and complexity required to get the text hello world printed on the screen.
- In other programming languages as you saw in the previous slides, the number of lines were more than 3 and the number of lines of code required for printing hello world in python is 1.
- This is the beauty of python that with minimal code you can achieve equal or greater results.

Comments in python

- Comments are one of the important concept in any programming language still many programmers ignore it.
- Using comments you can describe your code and make it more readable and understandable.
- They can also be used to instantly ignore a line of code while execution.
- Comments are ignored by python interpreter.
- You can use single line comment by using # followed by the comment.
- You can also use multi line comment method wherein you can mention a comment that is more than 1 line.

Variables in python

- Variables can be considered as boxes or containers where we store some value like number.



- In python the we don't specify the data type of variables. Hence it is called dynamically typed language.
- The variable can store numbers, strings, list, etc.

Variables in python

- Variable names should be descriptive. It increases the readability of the code. For eg. if you want a variable to store the salary of an employee then you should declare the variable as `salary = 5000`.
- There are some rules and variable naming conventions that you need to follow:
 - a. Variable names must start with a lowercase letter and underscore.
 - b. Variable should not start with a number.
 - c. They are case sensitive. Name, name, NAME all three of them can contain different values at the same time as all three are considered as different variable.
 - d. Variable names must contain only alpha-numeric character i.e (A-Z, a-z, 0-9) and underscores (_).
- Also you can declare multiple variables in a single line with different or same values.

Variables in python

- For printing variable values in python along with string we can use multiple methods.
- Variables also have scope. The variable you define inside a function cannot be used outside that particular function.
- If you want to access the variables that are declared inside a function (local variables) in any part of code then you need to make it a global variable by specifying global keyword.

Data types in Python

- Previously we described how variables store the data in it and we can access and use whenever we want.
- But what type of data can python store in its variables? To answer lets look what is data type and what are the different data type that python supports.
- First of all, lets understand that python is dynamically typed language. What that means is we don't need to specify the type of variable while declaring it. Python understand it on its own.
- Some data types and data structures are mutable (can be changed) and some are immutable.

Data types and data structures in Python

➤ Based on our requirement we can use different data types for different purposes. Following are the standard built-in data types that python has.

a. Numeric: int, float, complex

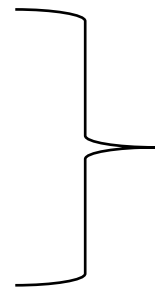
b. Text: str

c. Boolean

d. Sequence: list, tuple

e. Dictionary

f. Set



Data
Structures

Data types in Python (Numeric)

- In python we have three different data types that handle numeric values:
 - a. Int
 - b. Float
 - c. Complex
- Int can hold integer values in it. Eg: 1, 10, -7, 0 etc.
- Float can hold decimal point values 10.1, -34.98, 0.92742.
- Complex number can hold real number and an imaginary part 'j'. Eg: 4j, 10+9j, 2.9j
- You check the data type using type function like `type(x)`.

Data types in Python (Strings)

- String is the data type that can hold text, sequence of characters.
- String variable can be defined with single, double, or triple quotes.
- You can also define multiline strings in python using three single or double quotes both at the start and end of the string.
- Strings are arrays hence you can perform the slicing, indexing on strings as well.
- There are many built-in functions for strings.
- You can also check if a particular small string is present in larger string. Like 'hello' in 'hello world'.

Data types in Python (Boolean)

- Boolean type has two built in values that represents this data type, True and False.
- Boolean data types are very useful while you implement loops.
- When you compare two values then its resultant output is Boolean.
- Evaluating any expression will give you output of Boolean data type i.e. True or False.
- If any value with content is mostly evaluated as True. `bool(5)`, `bool('python')`.
- Some values are evaluated as false. Most of them are without content. `bool(False)`, `bool(())`.
- You can also return Boolean values in function. Some built in function returns Boolean values.
EG: `isupper()`

Data structures in Python (List)

- List is a mutable data structure.
- List comes to rescue when you want a group of multiple values assigned to a single variable.
- While declaring the variable you need to start and end with [and] respectively and values in between them.
- While declaring multiple values in list you need to separate them with comma.
- The main characteristic of list is that you need not store same type of values in single list. A list can contain int, float and string at the same time.
- List index starts from 0. So the first element in list will have index 0.
- Values can be repeated in list.
- You can perform various operations on list like slicing, appending value, deleting value, concatenating and so on.

Data structures in Python (Tuple)

- Tuples are similar to list. They can contain sequence of values like list but the main difference here is that tuple is immutable. Once created they can't be changed.
- Unlike list, while declaring tuple we use round brackets () instead of square brackets.
- You can also create tuple without using round brackets and directly assign multiple values to a variable.
- Just like list, even tuple's index starts from 0.
- Just like list values can be repeated in tuple.
- Since tuple is not mutable most of the operation like ones in list can't be used in tuple. Operations like appending value or deleting a value from tuple will give you error.

Data structures in Python (Set)

- Sets are unordered and unindexed.
- Sets are defined using curly brackets {}.
- Sets are of mutable type.
- You cannot access elements from set using the index since they are unindexed.
- You can add, remove elements from sets.
- There are many predefined functions of sets like there are in list, tuple.

Data structures in Python (Dictionary)

- Dictionary are mutable type of data structure.
- Dictionary contains keys and values in it. Dictionary can be imagined as key-value pairs.
`{key1:value1, key2:value2}`
- Values can be repeated and of mutable type but keys cannot be repeated and should be only of immutable type.
- You can access values of dictionary using key. Eg: `dict['key']`
- You can perform operations like deleting dictionary elements or add new elements.
- You can also use nested dictionary based on your requirement.

Agenda for Next Session

- ❑ Conditional loops (if/else, for, while)
- ❑ Operators
- ❑ Object Oriented Programming concepts (Oops)
- ❑ Basic Operations

QUESTIONS?

THANK YOU