

# Python Programming Basics

## Introduction:

- Python is a high level general purpose and very popular programming language
- It is being used in web development, machine learning, data science, game development, hacking etc.
- Python programs are generally smaller than other languages
- Programmers have to type relatively less code and indentation requirement of the language makes it easy for them.
- Very simple syntax
- General purpose (Simple and still very powerful)
- Multi paradigm support:
  - \* Procedural style programming like C

- ★ OOPs like JAVA
- ★ Functional programming like list.

- Portable and Platform independent  
Programs are typically compiled into an intermediate code, then the code is run by the interpreter.
- Dynamically Typed,
- Automatic Garbage Collection
- Popular applications like youtube, Netflix, Quora, Instagram, Drop Box etc use Python.

## First Program

As said before that Python has a simple syntax, the program written does not need to have been enclosed in a main() like C++ and does not need any semi-colon or return type.

Code:

print("Hello! World")

Output

Hello! World

## Comments In Python

### Single line comments:

In python we can write comments by writing them after "#" symbol like this:

# The following lines are null.

### Multi line comments:

For multi line comments in python we can put "#" in the begining of every line we want to to be a comment.

e.g

# Total number of products  
# used by women are 95.

OR

We can put comments enclosed  
in three quotes. e.g.

"""Total number of products  
used by women are 95."""

This is also called docstring.

# Input / Output in Python

Q1 Taking Input from users?

We / Developers often have the need to interact with the user, either to get data or to provide some sort of results. Python provides two inbuilt functions to read the input from the keyboard.

- 1 `input(prompt)`
- 2 `raw_input(prompt)`

\*`1 input()`: This function takes the input from the user and converts it into a string. When this function is called the program stops and wait for the user's response. When user presses enter key on keyboard, the program resumes.

Example:

```
# Python Program showing  
# use of input()
```

```
val = input("Enter your value:")
```

print (value)

Output:

Enter your value: --  
[value]

Working :

- 1 When `input()` executes program flow will be stopped until the user has given input.
  - 2 Text message displayed on the screen for user is called prompt.
  - 3 Whatever you enter as an input, the `input()` will convert it into a string until typecasting<sup>Q</sup> is used.
- 2 `raw_input()`: This works in older versions (like Python 2.x).

(Q) Taking multiple inputs from user?

In C/C++ we can take multiple inputs using scanf.  
But in Python we do take multiple inputs using two methods.

- Using `split()` method:
- Using list comprehension. Advanced

\* `split()` Method:

This helps in getting multiple inputs from users. It breaks the input by the specified separator. If no separator is specified any space is considered separator.

Syntax

`input().split(separator, maxsplit)`

Example:

# taking two inputs at a time

`x,y = input("Enter two values").split()`

```
print ("Number of boys:", x)
print ("Number of girls:", y)
```

Output:

Write this program and  
see for yourself.

Note: List Comprehension will  
be explained in advanced  
Python.

Print() Function:

Print() function shows/display  
the string/value/variable on screen  
or any other output device.  
We used it really frequently  
throughout previous topic.

Example:

```
name = "Abdullah"
```

```
print("Name:", name)
```

Output:

Name : Abdullah

## end parameter in print():

By default Python's `print()` ends with a newline. If programmers with C/C++ may wonder how to print without a newline. For this we use `end` parameter in Python's `print()`. By default it is `\n`, i.e. the new line character.

Now we will end the print statement with a space.

### Example #1:

```
#ends the output with a space  
print ("Welcome to", end = " ")  
print ("Geeks for Geeks.")
```

### Output:

Welcome to Geeks for  
Geeks

## sep Parameter in print():

When we have multiple arguments that has to be printed out on screen, we use "sep" parameter.

It is used to separate these arguments. By default it is a whitespace but we can use other characters, symbol or digits.

Example:

```
print("10" + "04", "2023", sep = "")
```

Output:

10-04-2023