

## Table of Contents

-Installation:.....	2
-What is Python Language: .....	4
-Why Python? .....	4
-Features of Python .....	5
-Advantages of Python Language.....	6
-Disadvantages of Python Language .....	6
-print(): .....	7
-PIP: .....	7
-Modules: .....	8
-Comments: .....	8
1. Single line comments.....	8
2. Multi line comments.....	8
-Variables:.....	9
-String format():.....	10
#Multiple Values.....	10
#Index Numbers .....	10
#Named Indexes .....	10
-DataTypes:.....	11

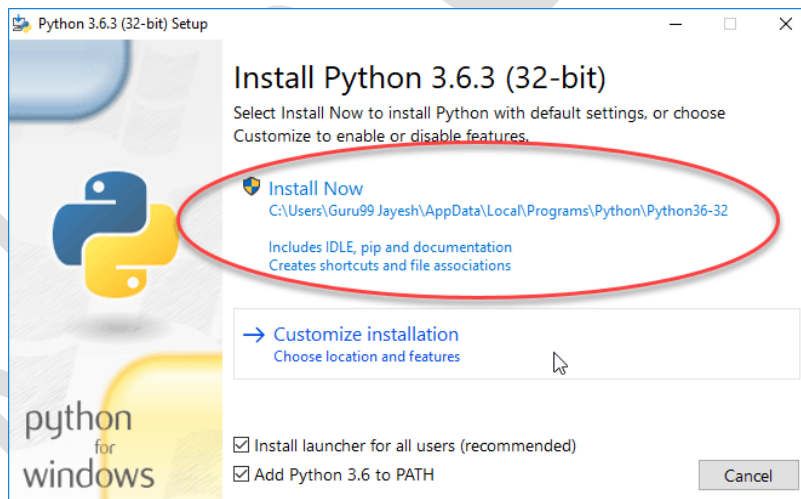
## Chapter 1: Python Introduction

### Installation:

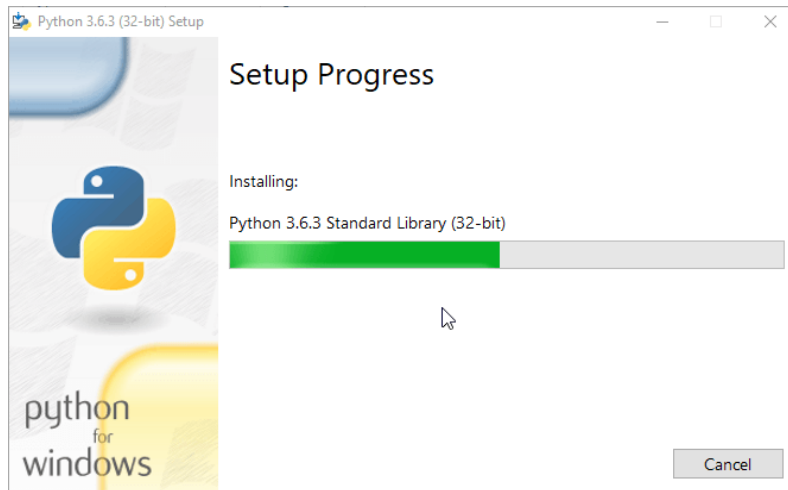
**Step 1)** To download and install Python, visit the official website of Python <https://www.python.org/downloads/> and choose your version.



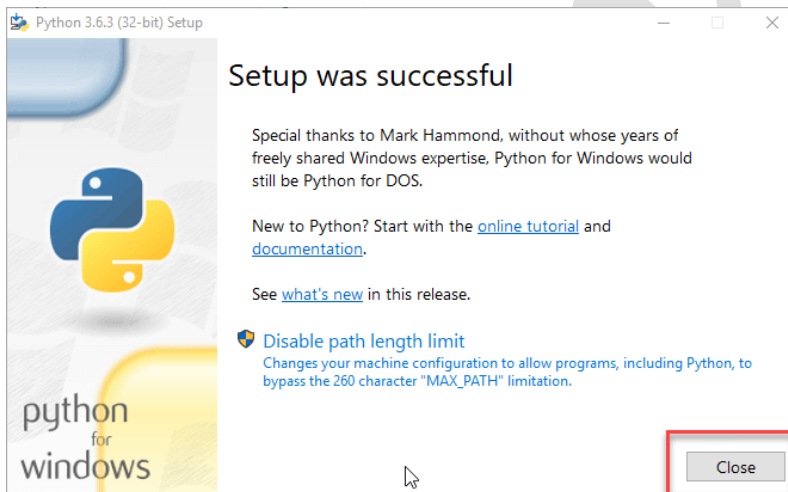
**Step 2)** Once the download is completed, run the .exe file to install Python. Now click on Install Now. And check on add python to path



**Step 3)** You can see Python installing at this point.



**Step 4)** When it finishes, you can see a screen that says the Setup was successful. Now click on “Close”.



# Fasten your seat belt, Let's have a journey to python.



## What is Python Language:

Python is a high-level, interpreted programming language that is widely used for developing a variety of applications, including web applications, scientific computing, data analysis, artificial intelligence, and automation. It was created by Guido van Rossum and first released in 1991.

Python is known for its simple and elegant syntax, which makes it easy to read and write, and its extensive set of libraries and frameworks that make it a powerful language for a wide range of applications.

## Why Python?

**General-purpose programming:** Python is a versatile language that can be used for a wide range of programming tasks. It can be used for web development, scientific computing, data analysis, artificial intelligence, automation, and much more.

- **Simple and easy to learn:** Python has a simple and easy-to-understand syntax, which makes it a popular choice for beginners. Its readability and clean code structure also make it easier to maintain and modify code in the long run.
- **Large standard library:** Python has a large standard library that includes modules for a wide range of tasks, such as string manipulation, file I/O, networking, and more. This makes it easier to write complex programs without having to write all the code from scratch.
- **Cross-platform compatibility:** Python can run on multiple platforms, including Windows, macOS, and Linux. This makes it easier to develop and deploy applications across different operating systems.

- Data science and machine learning: Python has become a popular choice for data science and machine learning because of its powerful libraries, such as NumPy, pandas, and scikit-learn, which provide efficient data processing, manipulation, and analysis capabilities.

Overall, Python's versatility, ease of use, and powerful libraries make it a popular choice for developers, scientists, and researchers across a wide range of fields.



## Features of Python:

Here are some of the key features of the Python programming language:

- Easy-to-read syntax: Python has a simple, easy-to-read syntax that emphasizes code readability and reduces the cost of program maintenance. This makes it easy to learn for beginners and reduces the time required to develop code.
- Dynamically typed: Python is dynamically typed, which means that you don't need to declare the data type of a variable before using it. The data type is inferred automatically at runtime, which reduces the time required to write code.
- Interpreted: Python is an interpreted language, which means that you don't need to compile your code before running it. This makes it easier to write and test code quickly.
- Object-oriented: Python supports object-oriented programming, which allows you to define classes and create objects that encapsulate data and behaviour. This makes it easier to organize and manage large codebases.
- Large standard library: Python has a large standard library that provides a wide range of modules and functions for tasks such as file I/O, networking, and data processing. This reduces the time required to write code and makes it easier to create complex programs.

- Cross-platform: Python can run on multiple platforms, including Windows, macOS, and Linux. This makes it easier to develop and deploy applications across different operating systems.
- High-level language: Python is a high-level language, which means that it provides abstractions that allow you to focus on the task you want to perform, rather than the low-level details of the computer's hardware.
- Memory management: Python has automatic memory management, which means that it automatically allocates and deallocates memory for your program. This reduces the risk of memory leaks and makes it easier to write memory-safe code.

Overall, Python's ease of use, readability, and large standard library make it a popular choice for a wide range of programming tasks.

### Advantages of Python Language:

- Easy to learn: Python has a simple syntax and a large standard library, making it easy for beginners to learn.
- Large community and support: Python has a large and active community that provides extensive support, documentation, and third-party libraries, making it easier to find solutions to common problems.
- Cross-platform compatibility: Python can run on a variety of platforms, including Windows, macOS, and Linux.
- Productivity: Python's simplicity and ease of use can increase productivity, allowing developers to write code faster and with fewer errors.
- Versatile: Python can be used for a wide variety of tasks, from web development to data analysis to machine learning.

### Disadvantages of Python Language:

- Performance: Python is an interpreted language and can be slower than compiled languages like C or C++. However, there are ways to optimize performance with tools like Cython or Numba.
- GIL limitation: Python's Global Interpreter Lock (GIL) can limit its ability to take advantage of multiple cores or processors for parallel processing.
- Mobile development: While Python can be used for mobile development, it is not as widely used in this area as other languages like Java or Swift.
- Runtime errors: Python's dynamically typed nature can make it more prone to runtime errors than statically typed languages.

Overall, Python's ease of use, versatility, and community support make it a popular choice for a wide range of programming tasks. While it may not be the best choice for every situation, it is often a good choice for many developers.



## print():

print() function is used to print specified text to the terminal.

Syntax: print(object(s), sep=separator, end=end, file=file, flush=flush)

Parameter	Description
<i>object(s)</i>	Any object, and as many as you like. Will be converted to string before printed
<i>sep='separator'</i>	Optional. Specify how to separate the objects, if there is more than one. Default is ' '
<i>end='end'</i>	Optional. Specify what to print at the end. Default is '\n' (line feed)
<i>file</i>	Optional. An object with a write method. Default is sys.stdout
<i>flush</i>	Optional. A Boolean, specifying if the output is flushed (True) or buffered (False). Default is False

Ex.

```
print("Hello Madi")
```

Output: Hello Madi

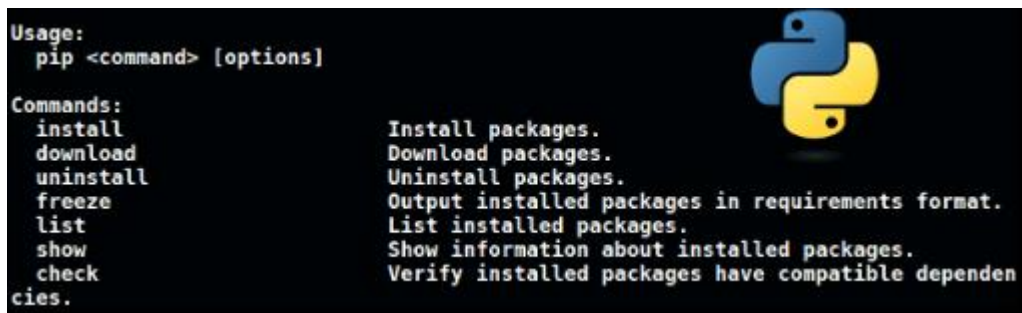
Python 3.6 uses the input() method.

Python 2.7 uses the raw\_input() method.

## PIP:

What is pip used for in Python?

What is pip ? pip is the standard package manager for Python. It allows you to install and manage additional packages that are not part of the Python standard library.



Ex.

```
pip install opencv-python
```

```
pip install opencv-contrib-python
```

## Modules:

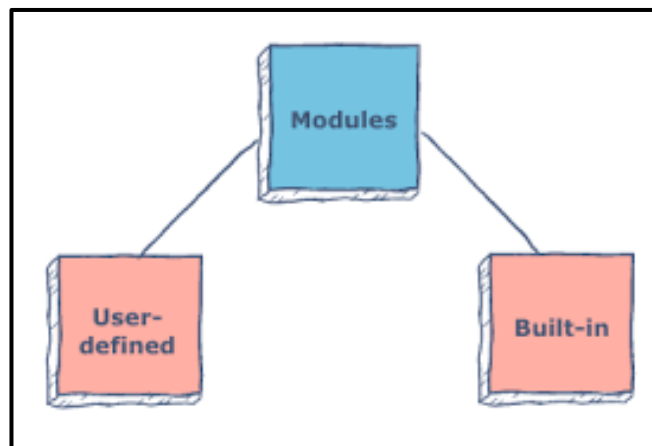
A Python module is a file containing Python definitions and statements. A module can define functions, classes, and variables. A module can also include runnable code. Grouping related code into a module makes the code easier to understand and use. It also makes the code logically organized.

### A FILE CONTAINING FUNCTIONS

Types of modules:

There are two types of modules:

1. Built-in Modules.
2. User-defined Modules.



## Comments:

Comments in Python is the inclusion of short descriptions along with the code to increase its readability. A developer uses them to write his or her thought process while writing the code. It explains the basic logic behind why a particular line of code was written.

1. **Single line comments:** use #

Ex. # This is single line comment

2. **Multi line comments:** use """

Ex. """If you don't want to write a comment using hashtags,  
you can put them in a multi-line comment block.

However, if you put hashtags in strings, they do not  
create comments.

"""

Ex.

"""

This is only comment

"""

```
print("Hello Madi")
```

```
# Hello Madi
```

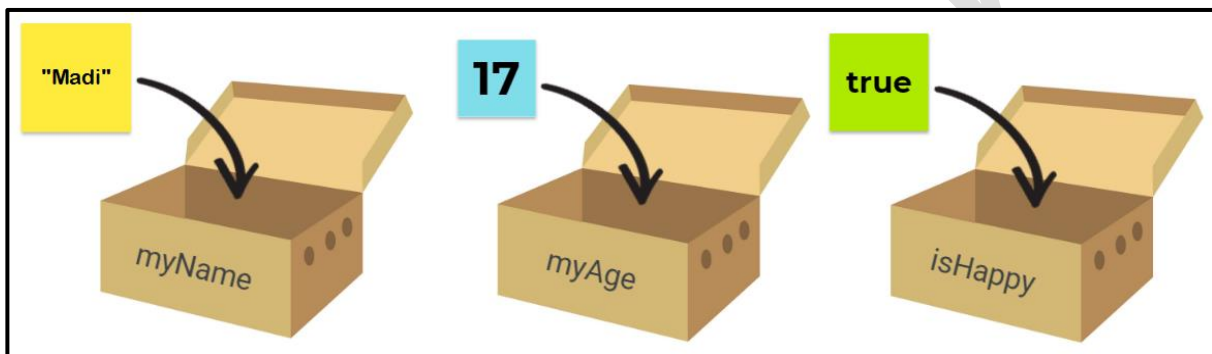


## Variables:

Variables are containers for storing data values.

Rules for creating variables in Python:

- 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 alpha-numeric characters and underscores (A-z, 0-9, and \_).
- Variable names are case-sensitive (name, Name and NAME are three different variables).
- The reserved words(keywords) cannot be used naming the variable.



## String format():

The format() method allows you to format selected parts of a string.

Sometimes there are parts of a text that you do not control, maybe they come from a database, or user input?

To control such values, add placeholders (curly brackets {}) in the text, and run the values through the format() method

Ex.:

```
name = "Madi"
```

```
txt = "{} is perfect"
```

```
print(txt.format(name)) # Madi is perfect
```

### Multiple Values

If you want to use more values, just add more values to the format() method:

Ex.2.

```
quantity = 3
```

```
itemno = 567
```

```
price = 499
```

```
myorder = "I want {} pieces of item number {} for {:.2f} rupees."
```

```
print(myorder.format(quantity, itemno, price))
```

# You can add parameters inside the curly brackets to specify how to convert the value

### Index Numbers

You can use index numbers (a number inside the curly brackets {0}) to be sure the values are placed in the correct placeholders

Ex.3.

```
quantity = 3
```

```
itemno = 567
```

```
price = 499
```

```
myorder = "I want {0} pieces of item number {1} for {2:.2f} rupees."
```

```
print(myorder.format(quantity, itemno, price))
```

### Named Indexes:

You can also use named indexes by entering a name inside the curly brackets {carname}, but then you must use names when you pass the parameter values

```
txt.format(carname = "Ford")
```

Ex.4.

```
myorder = "I have a {carname}, it is a {model}."  
print(myorder.format(carname = "Ford", model = "Mustang"))
```

## DataTypes:

- In programming, data type is an important concept.
- Variables can store data of different types, and different types can do different things.
- Python has the following data types built-in by default, in these categories:

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

None Type: NoneType

Ex.

```
x = 5
```

```
print(type(x)) # int
```

```
# type returns the data type of specified value
```

