PYTHON

</>

“

Python is an experiment in how much freedom programmers need. Too much freedom and nobody can read another's code; too little and expressiveness is endangered

”

– Guido van Rossum

Key Features

- Free and Open Source
- High-level programming language
- Object-oriented Scripting
- General-purpose Interpreted
- GUI Programming Support
- Extensible feature
- Portable and Interpreted
- Large Standard Library
- Dynamically Typed and Memory Allocating language
- Easy to Code, Read and Debug

Companies using Python

amazon

SPACEX

UBER

NETFLIX

IBM

intel®

NASA

ebay

Quora

Many more.....

Applications of Python

- Web Development
- Data Science
- Artificial Intelligence and Machine Learning
- Enterprise Applications
- Education Sector
- Web Scraping Applications
- Game Development
- Software Development (Frontend and backend development)

Install Python

Step 1: Download Python

https://www.python.org/downloads/

Step 2: Run the Executable Installer

Step 3: Add Python to environmental variables (Optional)

Step 4: Verify the Python Installation in command line / terminal

VARIABLES

Symbolic name that is a reference or pointer to an object.

Variables in Python

- Variable can have a **short name** or a **descriptive name**
- Must **start** with a **letter** or the **underscore** character
- **Cannot start** with a **number**
- Can only contain alphanumeric characters and underscores (**A-z, 0-9, and _)**
- Can be **case sensitive.**
- **Cannot** use **reserved keywords.**
- Object are **dynamically typed**

- Variable Assignment

```
n = 3
m = ”John”
```

- Casting variables

```
x = str(3)
y = int(3)
z = float(3)
```

- Variable Type

```
type(n)
```

- Object Identity

```
id(a)
```

Reserved Words (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	

Assignment - Variables

1. Create numerical object and assign a variable to the object

    a. Display the value and type

    b. Find the object id

2. Create string object and assign a variable to the object

    a. Display the value and type

    b. Find the object id

Operands & Expressions

Operators

- Arithmetic Operators
  a. Addition
  b. Subtraction
  c. Multiplication
  d. Division
  e. Modulo
  f. Floor Division(Integer Division)
  g. Exponentiation

- Logical Operators
  a. not
  b. and
  c. or

- Identity Operators
  a. is
  b. is not

- Comparison Operators
  a. Equal to (==)
  b. Not Equal to(!=)
  c. Less than(<)
  d. Less than or Equal to (<=)
  e. Greater than(>)
  f. Greater than or Equal to(>=)

- Bitwise Operators

a.	&	AND
b.	|	OR
c.	^	XOR
d.	~	NOT
e.	<<	Zero fill left shift
f.	>>	Signed right shift

Conditional Statements

It allows conditional execution of a statement or group of statements based on the value of an expression.

The conditions are evaluated and processed as true or false. If this is found to be true, the program is run as needed. If the condition is found to be false, the statement following the If condition is executed.

1. If the statement
2. If else statement
3. Nested if statement
4. If…Elif ladder
5. Short-Hand if statement
6. Short-Hand if-else statement

Loops

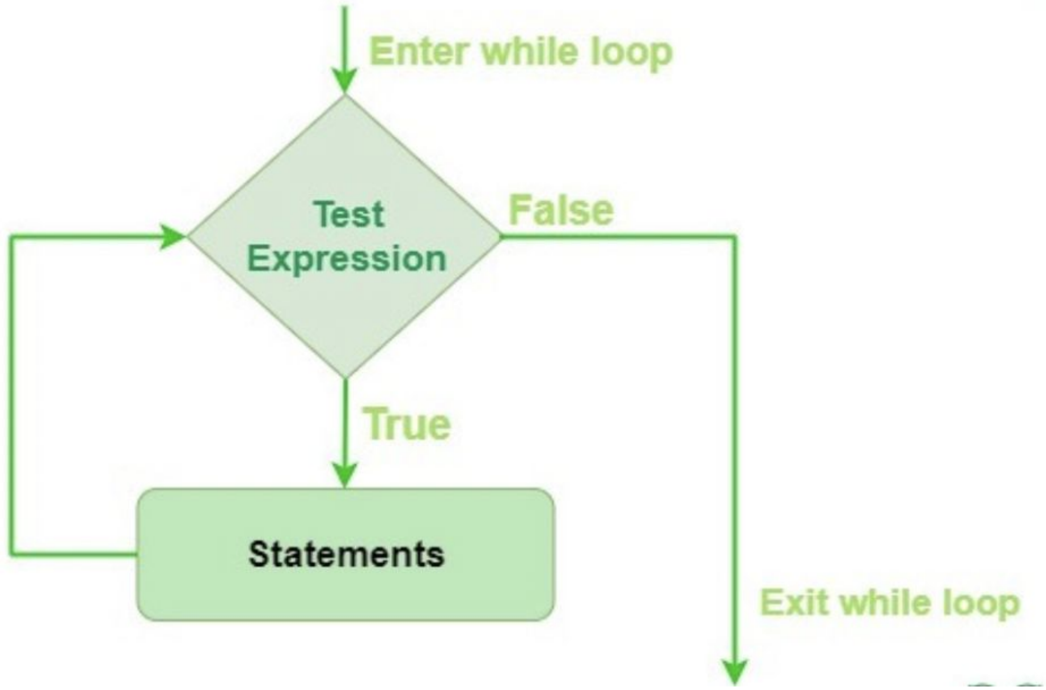
Python provides three ways for executing the loops. While all the ways provide similar basic functionality, they differ in their syntax and condition checking time.

While Loop

A while loop is used to execute a block of statements repeatedly until a given condition is satisfied. And when the condition becomes false, the line immediately after the loop in the program is executed.

Syntax:

```
while expression:
      statement(s)
```

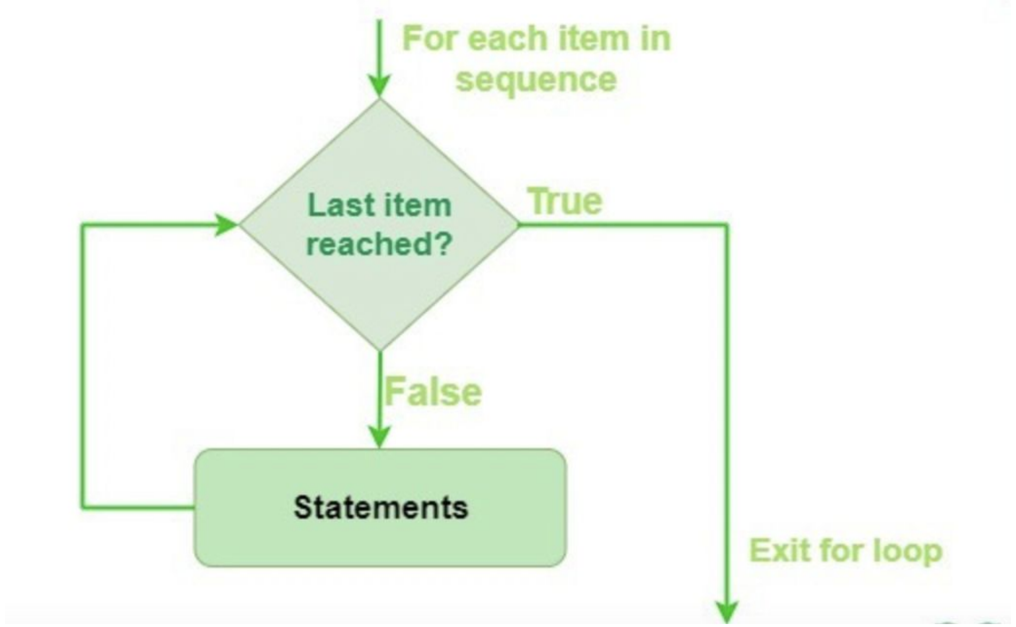


For Loop

For loop is used for sequential traversal i.e. it is used for iterating over an iterable like String, Tuple, List, Set or Dictionary.

Syntax:

```
for var in iterable:
  # statements
```



Command Line Arguments

The arguments that are given after the name of the program in the command line shell of the operating system are known as Command Line Arguments.

Sys Module

- Another way to accept input function other than input()
- Inputs will be passed to the program as an argument from the command prompt
- **Sys** module is used to process the inputs form the command prompt
- All the arguments passing to the command prompt will forms a List as **argv**

This will be accessed as sys.argv ----> treated as a string

Elements of argv can be accessed using index

sys.argv[0] --- File name

sys.argv[1] --- Arguments

sys.argv[2] --- Arguments

. .

. .

. .

sys.argv[n] --- Arguments

Sequential Structures & Operations

4 built-in data types in Python used to store collections of data are List,Tuple, Set, and Dictionary, all with different qualities and usage.

Number List

- Lists are used to store multiple items in a single variable.
- Variables are stored in []
- List items are **ordered**, **changeable or mutable**, and allow **duplicate values**.
- List items are indexed, index starts with `[0]`.

List Operations

append()	Adds an element at the end of the list
clear()	Removes all the elements from the list
copy()	Returns a copy of the list
count()	Returns the number of elements with the specified value
extend()	Add the elements of a list (or any iterable), to the end of the current list
index()	Returns the index of the first element with the specified value
insert()	Adds an element at the specified position
pop()	Removes the element at the specified position
remove()	Removes the item with the specified value
reverse()	Reverses the order of the list
sort()	Sorts the list

Tuples

- Tuples are used to store multiple items in a single variable.
- Variables are stored in ()
- A tuple is a collection which is **ordered** and **unchangeable or immutable.** Also allow **duplicate values**
- Tuples are written with round brackets.
- Tuples items are also indexed, index starts with `[0]`
- A tuple can contain different data types like string, integer, float, boolean etc.

Sets

- Sets are used to store multiple items in a single variable.
- Variables are stored in {}
- A set is a collection which is ***unordered***, ***unchangeable****, ***unindexed*** and **do not** allow **duplicate** values.
- Unordered means that the items in a set do not have a defined order. Set items can appear in a different order every time you use them, and cannot be referred to by index or key.
- Once a set is created, you cannot change its items, but you can remove items and add new items
- The data type of sets are 'set’

Dictionaries

- Dictionaries are used to store data values in **key:value** pairs
- Dictionaries are written with curly brackets
- A dictionary is a collection which is **ordered**, **changeable** and **do not allow duplicates**.
- The data type of dictionary are 'dict'
- The values in dictionary items can be of any data type

Dictionary Operations

Method	**Description**
clear()	Removes all the elements from the dictionary
copy()	Returns a copy of the dictionary
fromkeys()	Returns a dictionary with the specified keys and value
get()	Returns the value of the specified key
items()	Returns a list containing a tuple for each key value pair
keys()	Returns a list containing the dictionary's keys
pop()	Removes the element with the specified key
popitem()	Removes the last inserted key-value pair
setdefault()	Returns the value of the specified key. If the key does not exist: insert the key, with the specified value
update()	Updates the dictionary with the specified key-value pairs
values()	Returns a list of all the values in the dictionary

File Handling

- The key function for working with files in Python is the `open()` **function**.
- The `open()` function takes two parameters; ***filename*, and *mode*.**

There are 4 different modes (methods) for opening a file

"r" - Read - Default value. Opens a file for reading, error if the file does not exist

"a" - Append - Opens a file for appending, creates the file if it does not exist

"w" - Write - Opens a file for writing, creates the file if it does not exist

"x" - Create - Creates the specified file, returns an error if the file exists

Delete a file/ folder

- To delete a file, you must import the **os** module, and run its **os.remove()** function
- To delete a folder, import the **os** module, and run its **os.rmdir()** function

Functions

- Major use of function is its **Reusability.** Once we write a function, that function can be called number of times in any other programs
- In order to re-write a logic, we can write a single generalised function and can make use of it.
- **Debugging** the program will be easier for the user. If a program with more line of code or more complex logic has to be executed, it would be difficult to find the errors(especially the logical errors).

- ~~Function Declaration~~
- Function Call
- Function Definition

In python, first start write the function definition, then write the call function inside the program.

Function Definition

- Every function should start with **def** keyword
- Every function should have a **name** which is not any keyword
- **Parameters/ Arguments** are the inputs given to the user defined function, which is optional (This should be included between parentheses)
- Every function name with / without arguments **should end with colon**
- **Return** can be **empty or value.** Every function will return its control to its calling function
- **Multi-value return** can be accepted. In python, function can return multiple values to the calling function, this can be achieved in the form of tuples.

Function Call

- Function name
- Arguments / Parameters

Equal to the Function definition

Same function name and the same number of arguments/parameters should be returned in the Function definition.

Function definition will be executed after function call

Global Variables

- Any variable created outside of the function is a global variable.
- These variables can be accessed from both inside and outside of the function.
- Scope of local variable is only within the function, where the variables have been declared
- If local and global variables have same value, first preference is for local variables.

Variable scope and Return Values

Return function will give the value and store it in a variable and can be use in the later programs

On the other hand, the print can only display the values, it never store the values

Lambda Functions

- A lambda function is an **anonymous** function.
- These functions are not defined by **def keyword.**
- Return **expressions** but not values
- One line functions
- Can take any number of values
- This function cannot access global variables
- Defined by the keyword called **lambda**
- Function doesn’t hold any name, but the lambda function can be assigned to any variable.

Syntax:

```
lambda  arguments : expression
```

Standard Libraries

There are over 137,000 python libraries present today, and they play a vital role in developing machine learning, data science, data visualization, image and data manipulation applications, and more.

- Libraries are also called as Modules
- There are built-in functions or methods, which will reduce the code written by the user.
- Most of the libraries are free and open sourced.

❖ Operating System Interface (os)	: To interact with the operating system
❖ Command Line Arguments (sys)	: To process command line arguments
❖ Arrow (arrow)	: To work with date and time
❖ bokeh	: To interact with data visualisation for modern web browsers.
❖ BeautifulSoup	: Web scraping
❖ Django	: Data Analysis
❖ Keras	: ML / data mining

Standard Libraries

Matplotlib	To create charts , graphs, pie-charts,scatter plots, histograms etc
Numpy	Numerical Python, for array representations
Pandas	Data Analysis
OpenCV	Image processing(open source computer vision)
PyTorch	Machine Learning / Deep Learning
Pygames	Writing Video games, consist of computer graphics and sound waves
Requests	Creating http requests user friendly
Scikit	ML and statistical modeling for classification, regression, clustering via consist interfac
Seaborn	High level interactive and informative Data Visualisation
TextBlob	Textual Language Processing
Pillow	Python Imaging Library
WxPython	Developing GUI

Modules in Python

- A Python module is a file containing Python definitions and statements.
- Modules can be **Built-in Modules** and **User-defined Modules**.

Package Installation

Package Installation

Method 1 : If Anaconda is already installed on your System

To install a python package or module, enter the code below in Anaconda Prompt

```
pip  install package-name
```

Method 2 : NO Need of Anaconda

- Open **RUN** box using shortcut Windows Key + R
- Enter cmd in the RUN box
- Once you press OK, it will show command prompt screen.
- Search for folder named Scripts where pip applications are stored.
- In command prompt, change directory to the file location of Scripts folder
- Type *pip install package-name*

Package Installation

Method 3 : Install Python Package from IPython console

To install a python package or module, enter the code below in Anaconda Prompt

```
!pip install package-name          #some users face error "SyntaxError: invalid syntax"
```

OR

```
python -m pip install package-name
```

Useful

Description	Command
To uninstall a package	pip uninstall package
To upgrade a package	pip install --upgrade package
To search a package	pip search package-name
To check all the installed packages	pip list

Object Oriented Programming Concepts (OOPs)

Object Oriented Concepts

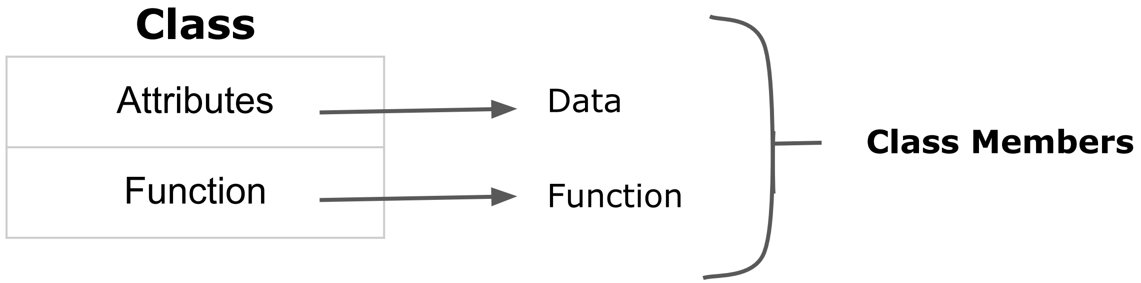
Object-oriented programming (OOPs) is a computer programming model that organizes software design around data, or objects, rather than functions and logic.

OOPs Terminologies

- Class
- Object
- Abstraction
- Encapsulation
- Inheritance
- Polymorphism

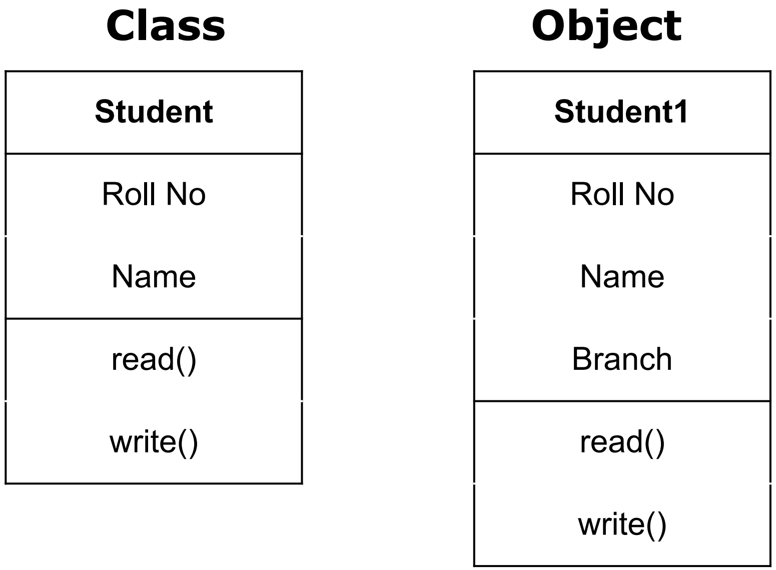
Classes

- A Class is a blueprint for creating objects
- Every class can have multiple objects/ instances, classes are always followed by object
- Classes are a **logical structure** with a behaviour
- These are a user-defined data type that is used to encapsulate data and associated functions together.
- Classes bind the data together into a single unit.
- Class can have attributes and function



Objects

- It is an instance/ reference of a class.
- With the help of objects, we can access the methods and attributes of a class.
- All the attributes and functions created for the class are available for the objects as well.
- Object is a physical entity



Abstraction

- Abstraction method is used to **hide the implementation** details
- It only **shows the essential** details.
- An abstract method is a method that has a declaration but does not have an implementation
- A class which contains one or more abstract methods is called an abstract class.

Encapsulation

- Binding the data and function in single entity is encapsulation.
- There are 3 access specifiers:
  - Public : Data & function from one class can be accessed by any function in any class
  - Protected : Data & function are only accessed by the class where it is declared and inherited
  - Private : Data & function are only accessed by the class where it is declared only.
- Hierarchy of data protection is **Public < Protected < Private**

Inheritance

- Major concept is base class and its derivatives. **A derived class can inherit the properties of a base class.**
    - Derived class -----> Child class
    - Base class -----> Parent class
- Derived class will be having same attributes and functions of the base class and also it can have another attributes and functions which are not present in the Parent Class.
- Whatever attributes and functions are declared in the parent class, those properties can be accessed and used by the child.
- Inheritance can be implemented in different ways, Single Inheritance, Multiple Inheritance, Multi-level Inheritance.

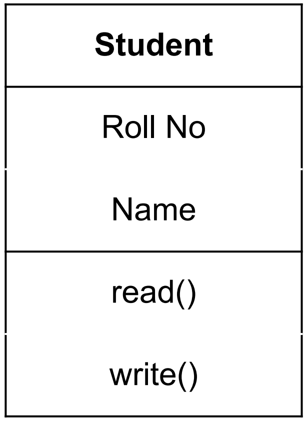
Polymorphism

- Polymorphism deals with methods used in a class.
- Implementing same method (having same name) in different context
    - add(x,y) -----> used in parent class
    - add()   ------> used in child class

    Here the function definition will be different

Class & Objects

Class



Syntax:

```
class  class_name:
    --------------------
    -------data--------
    --------------------
    Function1
    Function2
    Function3
```

All these functions should be defined.

- It is an instance/ reference of a class

Polymorphism

- Polymorphism deals with methods used in a class.
- Implementing same method (having same name) in different context
    - add(x,y) -----> used in parent class
    - add()   ------> used in child class

    Here the function definition will be different

-

Class & Objects

- To solve real world issues using a virtual world solutions

Error & Exception Handling

Error & Exception Handling

- Compile Time error
- Logical error
- Run Time error

Compile Time Error

- These are the syntax errors occurs while running a program
    - missing (:)
    - Spelling mistakes etc
- It is the most easiest error to find

Logical Error

- The code will be syntactically correct and the code compiles correctly, but the output will be wrong
- These errors can be figured out at least at the time of quality checking of the code.

Run Time Error

- The machine understands what you are saying, but runs into trouble when following the instructions.
    - Dividing a number by zero
- Syntactically and logically correct, but if the user is not following the instructions correctly

Thank you!!!