Python

Introduction of Python

History & Introduction Of Python

- Python was developed by **Guido Van Rossum**
- Python is a general purpose high level programming language.
- Implementation of python was started in December 1989 at CWI in Netherland
- But officially Python was made available to public in 1991. The official Date of Python is Feb 20th 1991.
- It was started firstly as a hobby project because he was looking for an interesting project to keep him occupied during Christmas.
- The programming language which Python is said to have succeeded is ABC Programming Language

History & Introduction Of Python

Python

Guido Van Rossum

Christmas

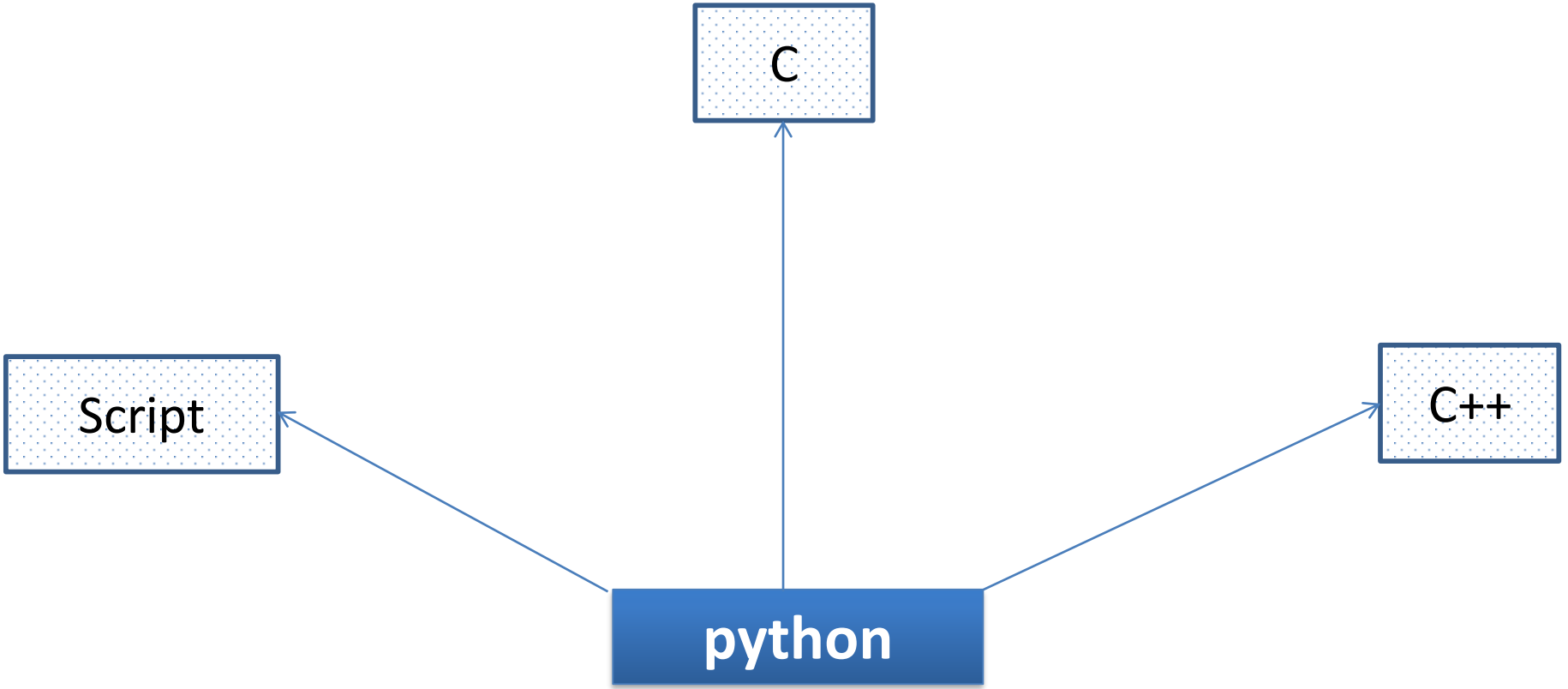
Implementation started

1989 at CWI in Netherland

Publicly available

Feb 20th 1991.

Inherit the features



(How Guido was decided the name python ?)

Programming language

BBC's TV Show
Monty Python's Flying Circus

Python language

inspiration

- He has taken syntax from ABC Language
- Guido Van Rossum also a member of creation of ABC language
- (How Guido was decided the name python ?)
- BBC's TV Show – 'Monty Python's Flying Circus, he got inspiration from this TV show

During the development of python Guido had taken most of the features from different different programming language.

1. Functional & structured Programming Features from C
2. Object Oriented Programming Features from C++
3. Scripting Language Features from Perl and Shell Script

Guido van Rossum

Python version release date

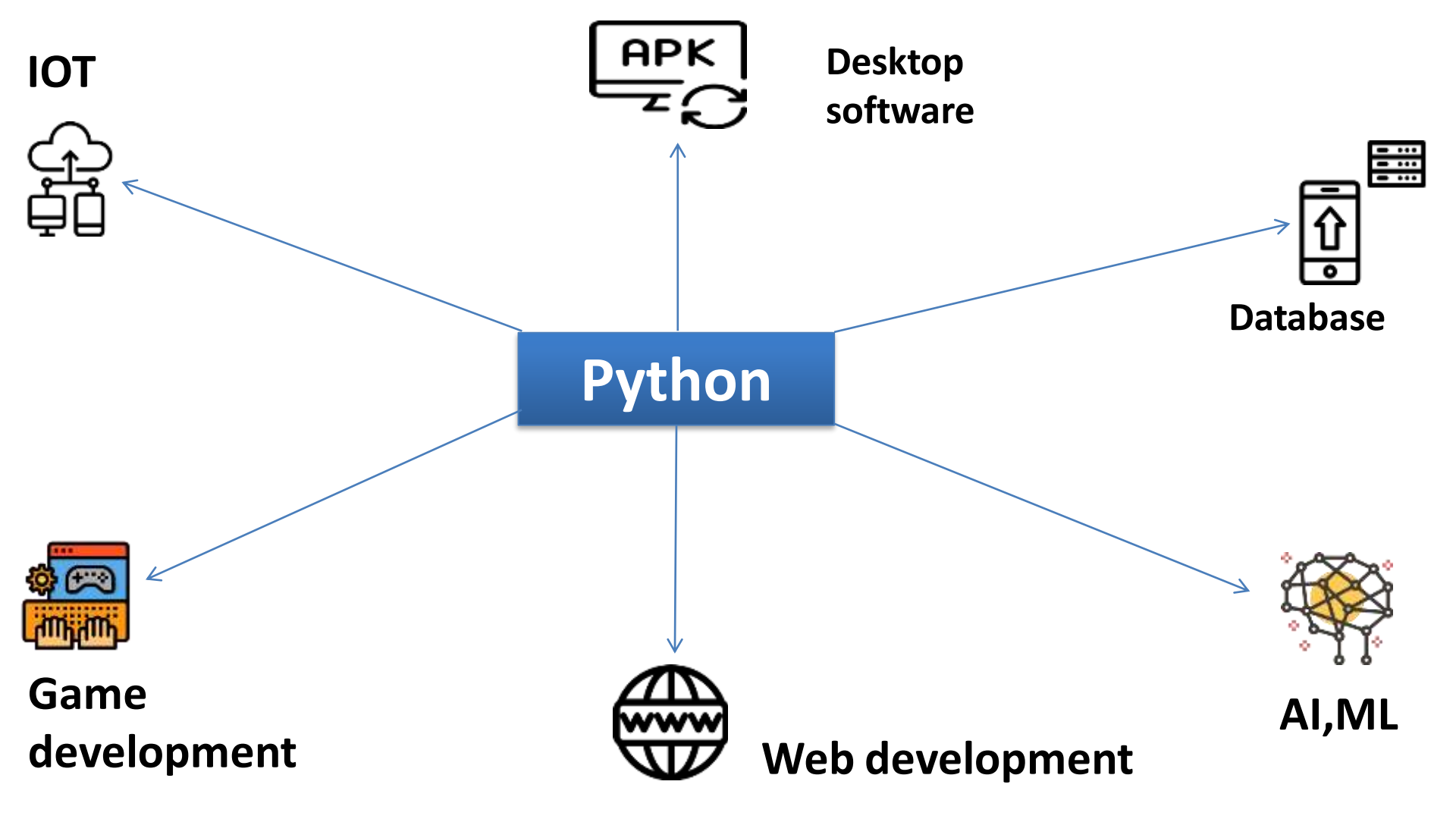
Python Version		Release Year
Python 1.0	-	Jan 1994
Python 1.5	-	31 Dec 1997
Python 1.5.2	-	April 1999
Python 1.6	-	05 Sep 2000
Python 2.0	-	16 Oct 2000
Python 2.0.1	-	22 Jun 2001
Python 2.1	-	17 April 2001
Python 2.2	-	21 Dec 2001
Python 2.3	-	29 Jul 2003
Python 2.4	-	30 Nov 2004
Python 2.5	-	19 Sep 2006
Python 2.6	-	01 Oct 2008
Python 2.7	-	03 Jul 2010

Python 3.1	-	27 Jun 2009
Python 3.2	-	20 Feb 2011
Python 3.3	-	29 Sep 2012
Python 3.4	-	16 Mar 2014
Python 3.5	-	13 Sep 2015
Python 3.6	-	23 Dec 2016
Python 3.7	-	27 Jun 2018
Python 3.8	-	14 Oct 2019

==========

Python 2.4.6	-	19 Dec 2008
Python 2.5.6	-	26 May 2011
Python 2.6.9	-	29 Oct 2013
Python 2.7.10	-	23 May 2015
Python 2.7.13	-	17 Dec 2016
Python 3.5.3	-	17 Jan 2017
Python 3.6.10	-	18 Dec 2019
python 3.8.2	-	24 feb 2020

Application of Python


IOT
APK
Desktop software
Database
Python
Game development
Web development
AI,ML

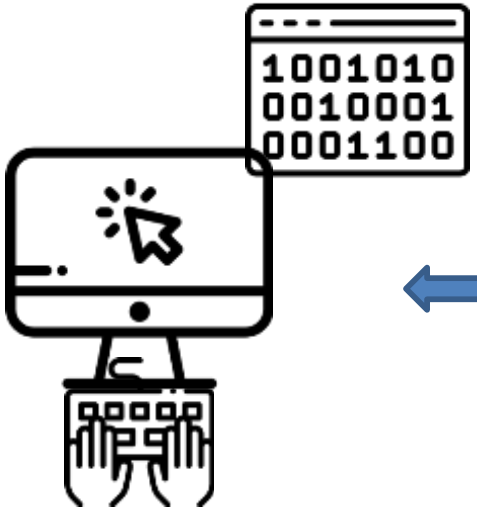
What Are The Application Of The Python? Which Kind Of Development We Can Do By Using Python?

- Python used for development of **standalone desktop** based application software
- Python used for development of web application like static and dynamic complex web application
- Python also used for creation of database application
- Python is highly recommended to used in Machine Learning , AI and in data analysis work
- In current time Top companies are focusing on Internet Of Things based product like home automation so for IOT development Python are highly used
- Python used for game application development
- If we go for ML,DL,AI then first choice is python

Python Features

- High Level Programming Language
- Portability
- Dynamically Typed
- Interpreted Programming Language
- Extensible
- Simple and Easy
- Freeware and Open source
- Embedded
- Extensive Library
- Object Oriented Concept
- GUI

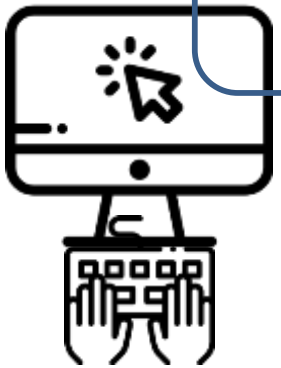
High Level Programming Language



compiler

process

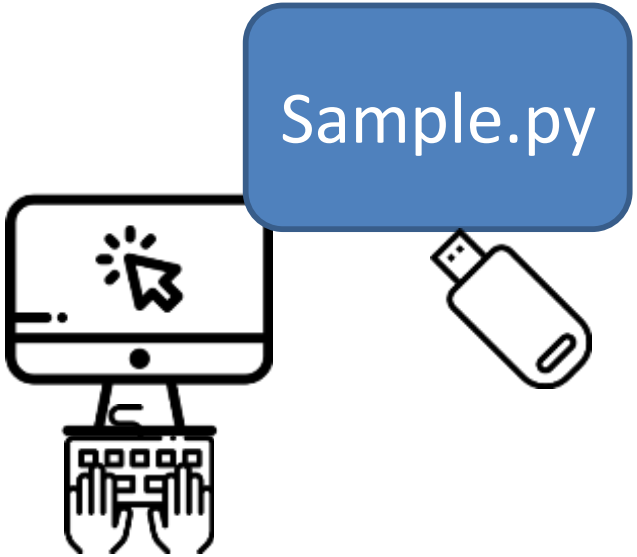
```
# include<>
If (condition)
{
}
```



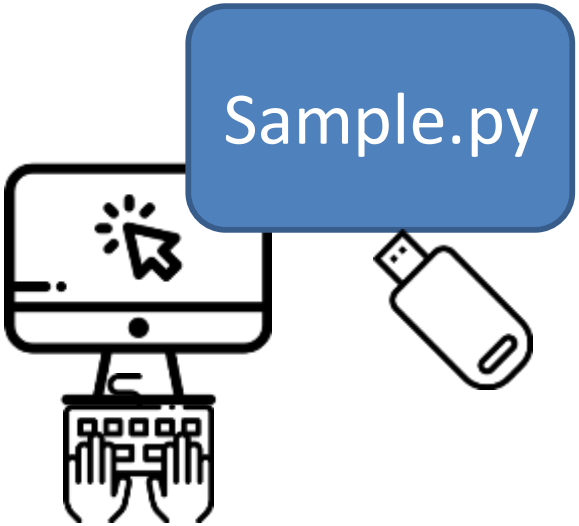
Low level language

High level language

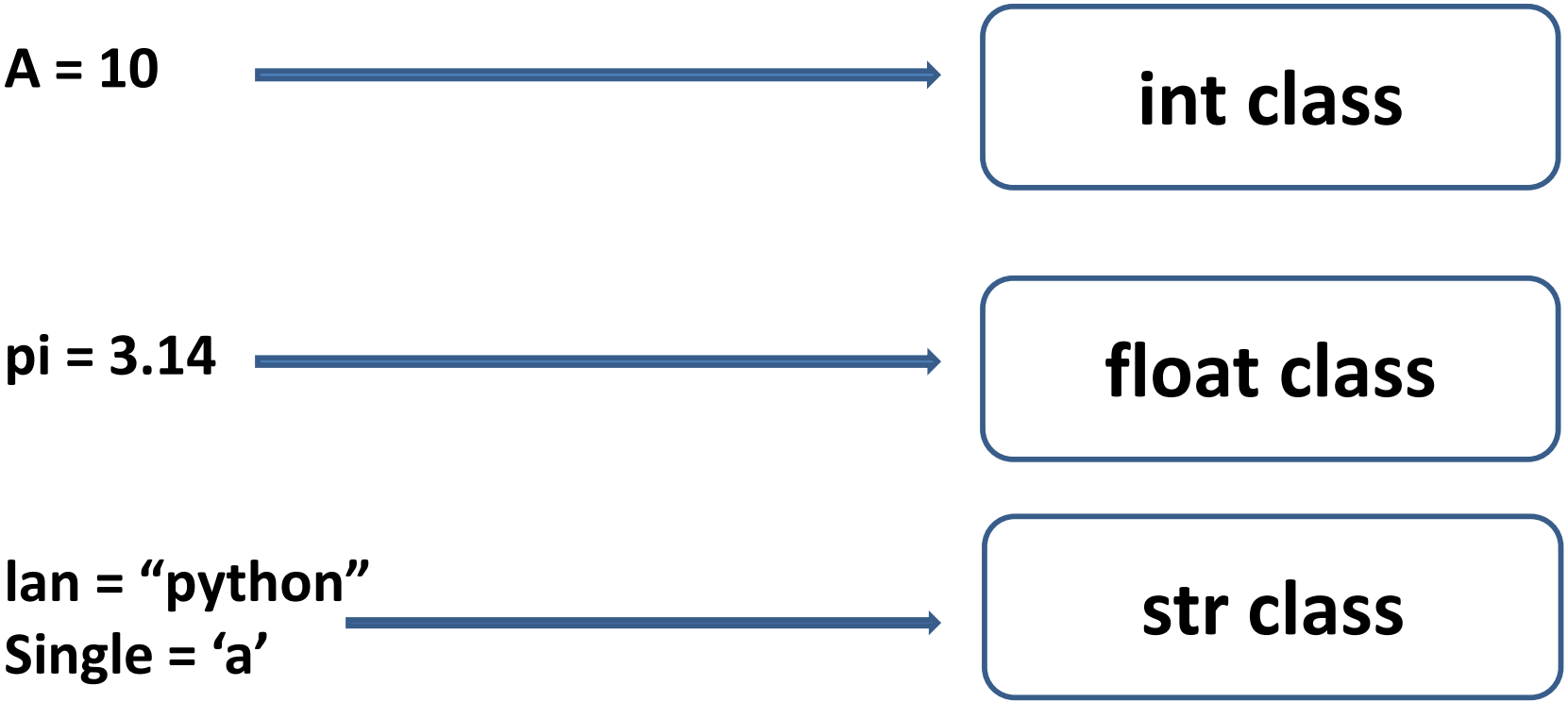
Portability

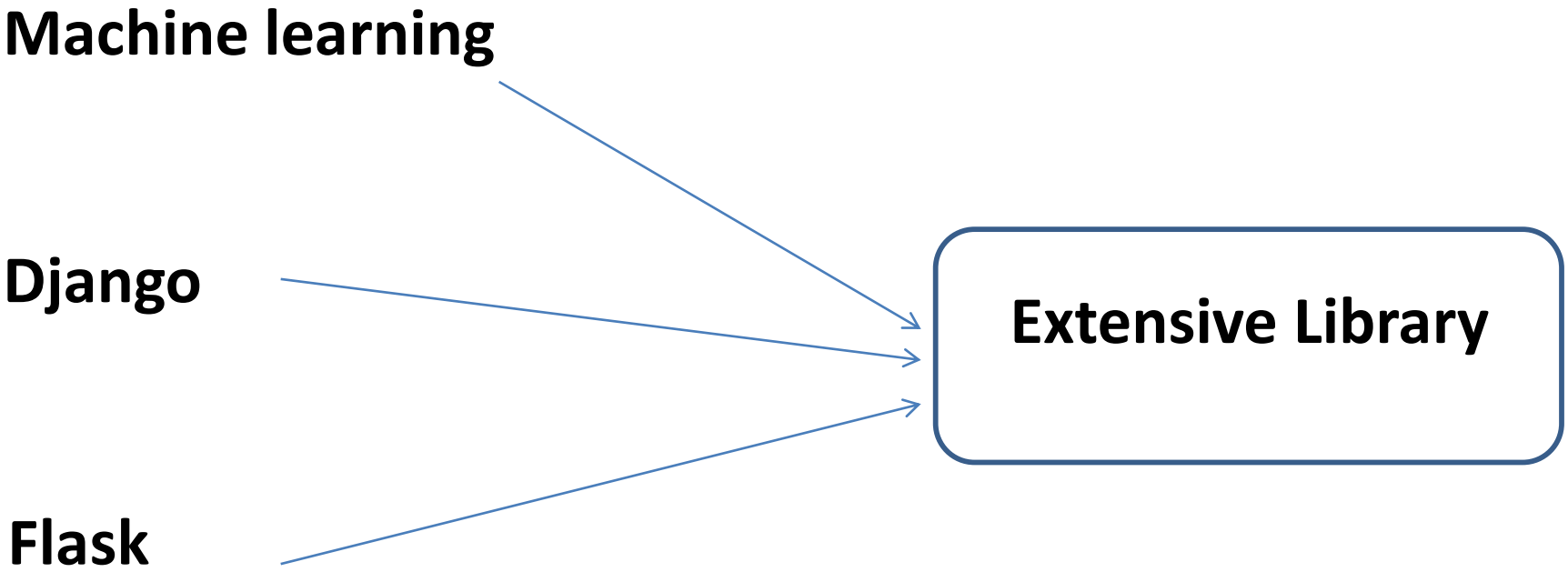


Windows machine



Mac machine


A = 10
int class
pi = 3.14
float class
lan = “python”
Single = ‘a’
str class


Machine learning
Django
Flask
Extensive Library

High Level Programming Language

High level language that are very close to human it means it is easy and understandable for human as comparison to machine level code

Simple And Easy

As comparison to other language in python we have to write less code, there is no curly bracket, there is no need to define data type explicitly

Dynamically Typed

When we assign any value to variable then according to that value internally data type class will allocate automatically

Interpreted Programming Language

There is python virtual machine concept in python (PVM) so portability are also possible and it interpret line by line

Object Oriented Concept

This provide real world concept so it provide reusability multiple forms and other concepts also

GUI

Python provides graphical user interface this is used for development of desktop based application using Tkinter , Kivy

Extensive Library

Python having large amount of extensive library for machine learning, django, flask framework etc.

Embedded

Other programming code we can use in python vice versa python code also can use in other language this is embedding

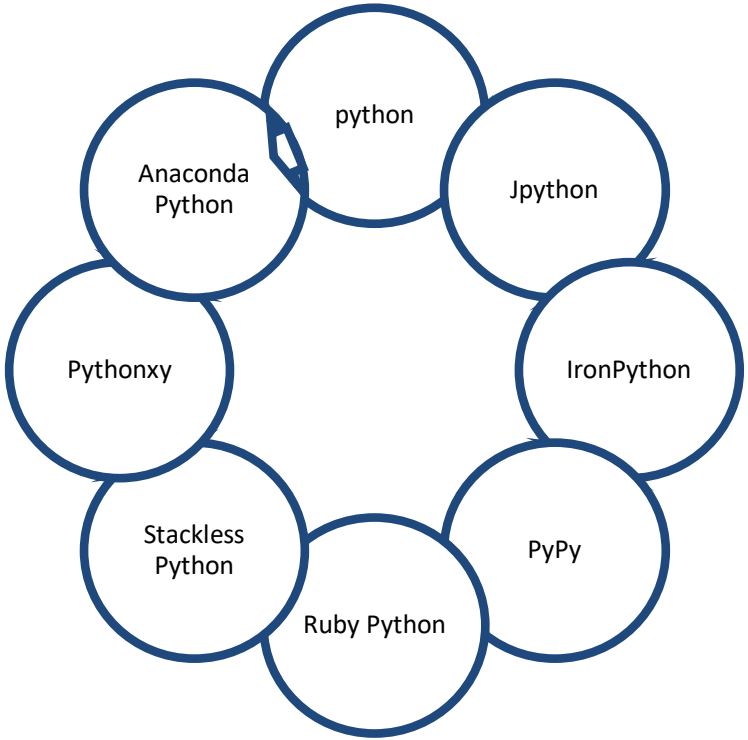
Portability

Python byte code can interpret according to platform like windows, Macintosh, Linux so we can say that python code are portable

Flavors Of Python means the different types of python compiler and these flavors help us to integrate different programming language into python

Flavors Of Python

- Cpython
- Jpython
- IronPython
- PyPy
- Ruby Python
- Stackless Python
- Pythonxy
- Anaconda Python


python
Jpython
IronPython
PyPy
Ruby Python
Stackless Python
Pythonxy
Anaconda Python

Freeware And OpenSource

No need to pay

Python source code is open

Specific requirement not full fill by standard python

As per our requirement we can customize python version

Ex ppython = prashant python

So Flavors of python is customize version of python

Flavors of python

To work with **C – language** application	⟹	**Cpython**
To work with **java** language application	⟹	**Jpython/jython**
To work with **C#. Net** application	⟹	**Iron python**
To work with **Ruby** language	⟹	**Ruby python**
To work with **ML,DL,DS**	⟹	**Anaconda python**

Python Are Very Easy And Simple To Use But How?

Java: -Lan

```
 public class Covid
 {
   public static void  main(String[] args)
    {
    SOP(“Hello Help4code");
    }
}
```

o/p = Hello Help4code

C: - Lan

```
#include<stdio.h>
 void main()
{
print(" Hello Help4code ");
}
```

o/p = Hello Help4code

==========

Python:- Lan

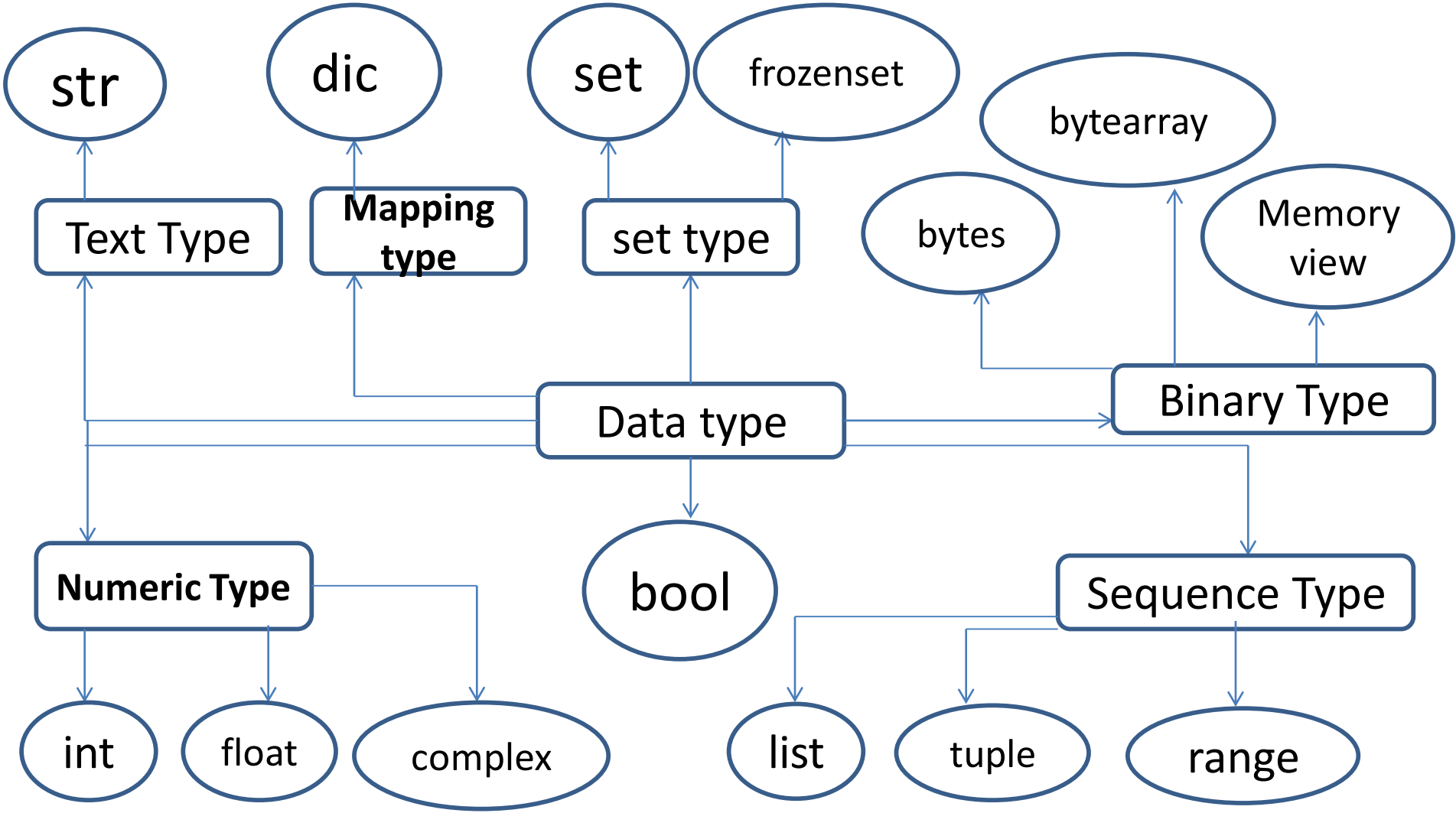
```
print(" Hello Help4code ")
```

o/p = Hello Help4code

Datatypes in Python

Data types in Python

- Data types tells that which type of data or value present inside of variable
- In python there is no need to define or declare data type explicitly or externally in program
- Data type basically depends on the value we provide because according to value internally the data type class will be assign automatically
- There for we can say that python is dynamically typed language
- By using type() function we can find the type of data type of variable


str
dic
set
frozenset
bytearray
Text Type
Mapping type
set type
bytes
Memory view
Binary Type
Data type
Numeric Type
bool
Sequence Type
int
float
complex
list
tuple
range

int

the range -2147483648 through 2147483647. (The range may be in Python 3, there is effectively no limit to how long an integer value can be. Of course, it is constrained by the amount of memory your system has but beyond the range it would be long integer

float

The float type in Python designates a floating-point number. float values are specified with a decimal point. Optionally, the character e or E followed by a positive or negative integer

Identifiers In Python

The name we use in python is called as python identifier or it can be name of variable, function, class etc.

Rules for identifier

- Identifier name can not be start with digit it should be start with alphabet, so digits are allowed but first must be alphabet.
- Identifier are case sensitive(so we can declare either lower case or in upper case)
- Reserve words are not allowed to declare as identifier
- Space are not allowed in identifier
- There is no special symbol are allowed except underscore
- If we start identifier name with underscore then it tells that it is private member
- If it start with single underscore then it private
- If it is start with two underscore then it is strongly private

Reserve Keywords

There are total 33 reserve keywords in python that comes along with python they are inbuilt.

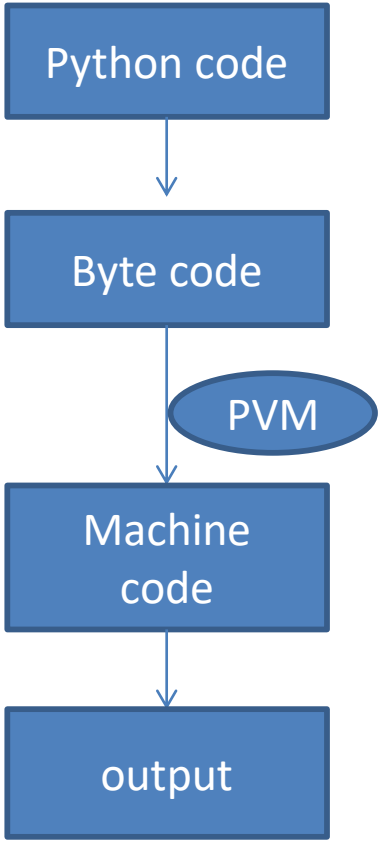
and	or	not	is	if	elif	else	import	from
as	class	def	pass	global	nonlocal	lambda	del	
with	try	except	finally	raise	assert	while	for	break
continue	return		in	yield	True	False	None	

All are alphabets & True, False, None except these three keyword all are in lower case

PVM(Python Virtual Machine)

Two stage system

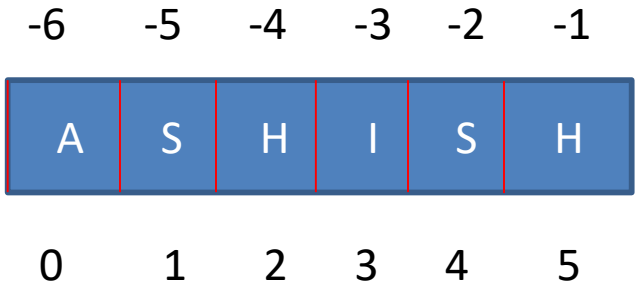
- When we run program The code get converted into byte code.



The byte code get converted by PVM into machine code so it is like interpreter there for PVM is also called as python interpreter.

String Slicing

- To extracting sub string from given string is basically known as slicing of string
- Here the string having positive(+) and negative(-) both indexing
- Positive indexing is start from left to right
- Negative indexing start from right to left
- These slicing also used in some other parts mostly in ML with pandas , numpy , array



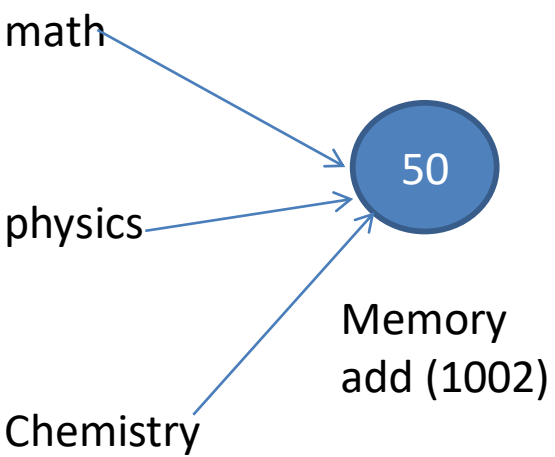
Fundamental Data types

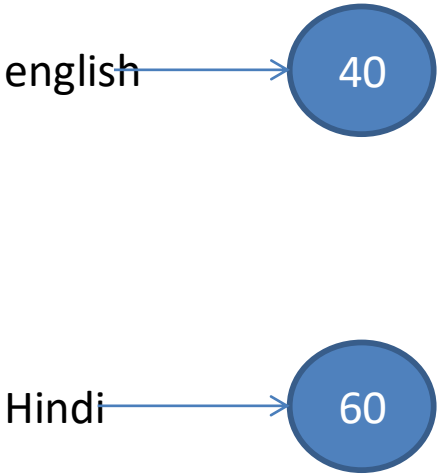
- int
- float
- complex
- bool
- str

- All fundamental data types are immutable
- Once we create a object we can’t do any changes
- If we want to do any changes then PVM first to check that is there any object available with same required data if it is available then ok otherwise new object will be created.
- Due to this memory management happen and this functionality really help to improve performance of program

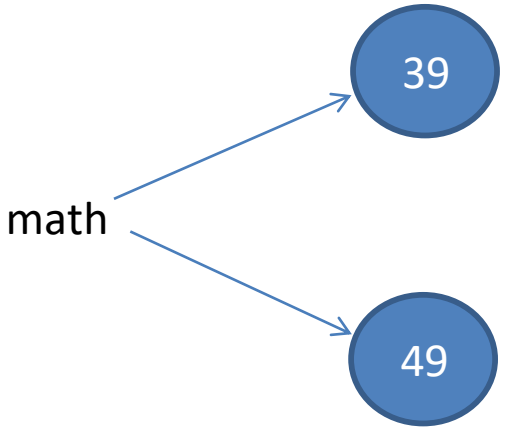
Memory Management

Math = 50 , physics = 50 , chemistry = 50 , English = 40 , Hindi = 60





Recheck paper and marks increase by 10



Type Casting

The conversion of value from one data type to another data type is called as typecasting

There are various predefine function are available for typecasting

- int()
- float()
- complex()
- bool()
- str()

Collection Datatype

Python Collection Data Type

There are four types of collection data type in python
As per program need programmer choose list data type

List:=

- Order wise data (we can say objects)
- Heterogeneous data are possible in list
- List represented by [ ] square bracket
- Duplicate data are also allowed
- List by nature it is growable
- Mutable

Tuple:=

Tuple is exactly as similar as list datatype only the difference is that list is mutable but tuple is immutable that means we can not change the value

- Order wise data (we can say objects)
- Heterogeneous data are possible in list
- Tuple represented by ( ) parenthesis
- Duplicate data are also allowed
- tuple by nature it is growable
- It is Immutable

Set Datatype:=

if we don’t want duplicate value and we want to represent random data that means there is no order wise data required then we should go for set datatype.

- No Order wise data (random)
- Heterogeneous data are possible in set
- set represented by { } parenthesis
- Duplicate data are not allowed
- set by nature it is growable
- It is mutable

frozenset Datatype:=

frozenset is exactly as similar as set datatype only the difference is that set is mutable but frozenset is immutable that means we can not use add() and remove()

- Heterogeneous data are possible in frozenset
- frozenset represented by { } parenthesis
- Duplicate data are not allowed
- frozenset by nature it is growable
- It is Immutable

dict Datatype:=

In dictionary we represent data as key and value pair

- dict represented by {key:value } parenthesis
- Duplicate keys are not allowed
- Duplicate values are allowed
- dict by nature it is growable
- It is mutable
- Unordered data

range datatype

If we want to represent sequence of number then we should go for range datatype
range datatype are immutable that means we can’t change

Difference between List and Tuple

List	Tuple
List is represented by square bracket, and square bracket are mandatory [ ]	Tuple is represented by parenthesis , and parenthesis are optional ( )
List objects are mutable	Tuple objects are immutable
Here if the requirement is not fixed then we should go for list	Here if the requirement is fixed then we should go for tuple
We can’t take list object as key for dictionary because key are immutable	We can take tuple object as key for dictionary because key are immutable

Operators

Operators

Operator is a special symbol which perform arithmetic and logical operation. support 8 types of operator they are given below:

No.	Category	Operator
1	Assignment Operator	=, +=, -=, *=, /=, %=, //=, **=
2	Arithmetic Operator	+, -, * , / , % , //
3	Relational Operator	<, >, <=, >=, ==
4	identity Operator	Is, is not
7	Logical Operator	and, or, not
5	Membership Operator	In, not in
6	Bitwise Operator	&, |, ~, <<, >>
8	Special Operator	Identity op, membership op

Special operators

There are two types of special operators

Identity operators

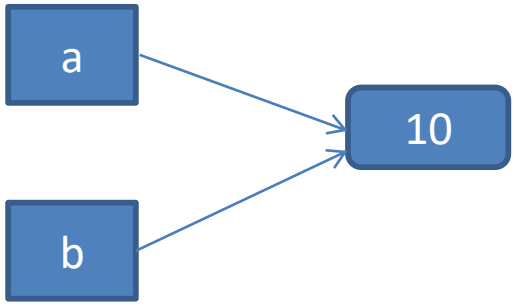
When we want do address comparison then we should go for Identity operators there are two types of identity operators

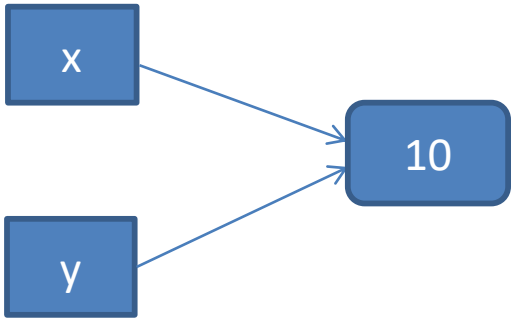
1. Is
2. Is not

For eg:=

a = 10

b= 10





x is y => this return true
if both x and y pointing to same object

x is not y => this return true
if x and y are not pointing to same object

Program:=

```
x= 10
y=10
print( x is y)
```

o/p=> True

Membership operators

By using membership operator we check that the object is present or not in collection data structure like (list, tuple, set, dict, string)

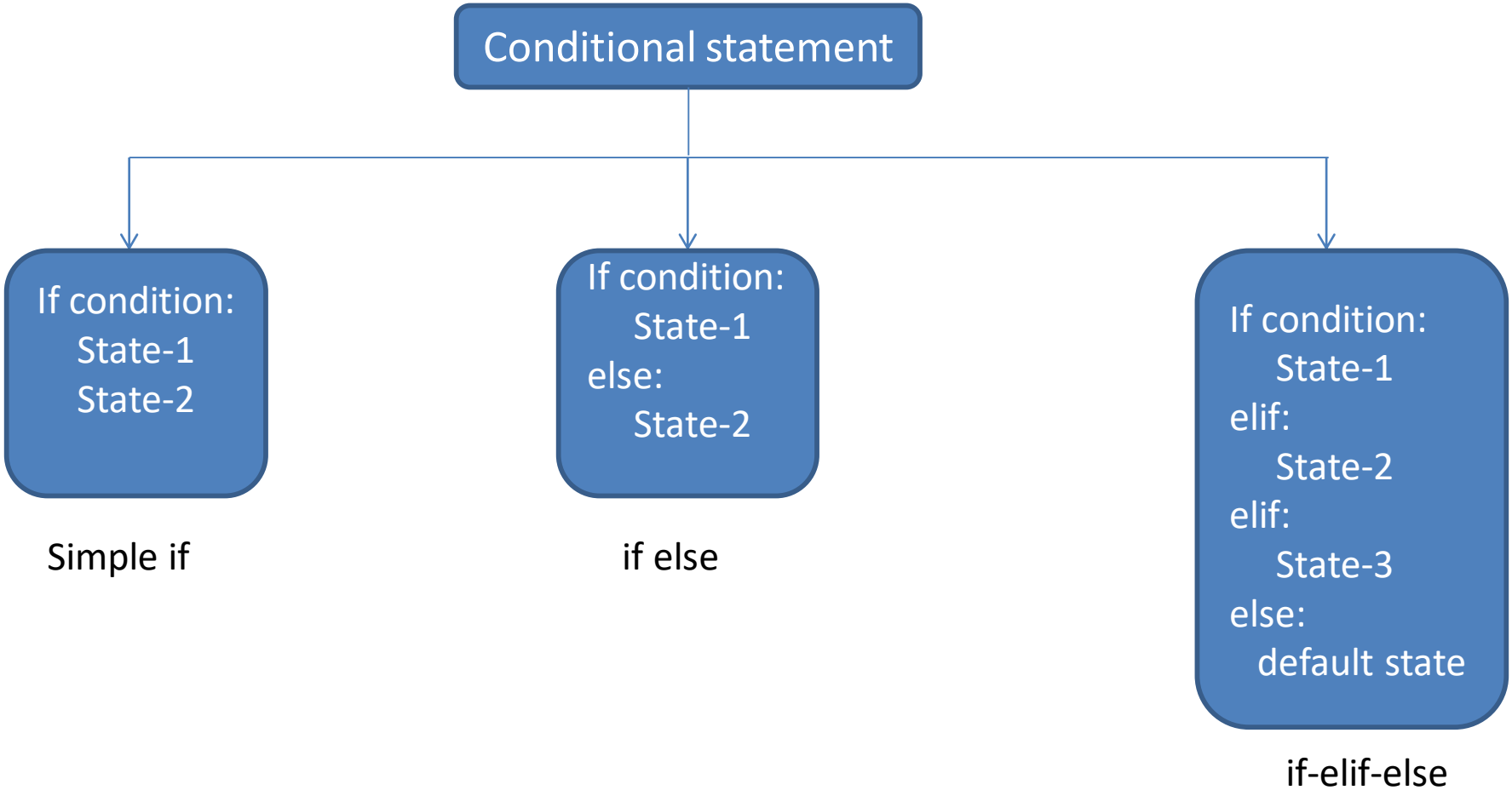
in ===➔ if the given object is present then it return true
not in ===➔ if the given object is not present then it return true

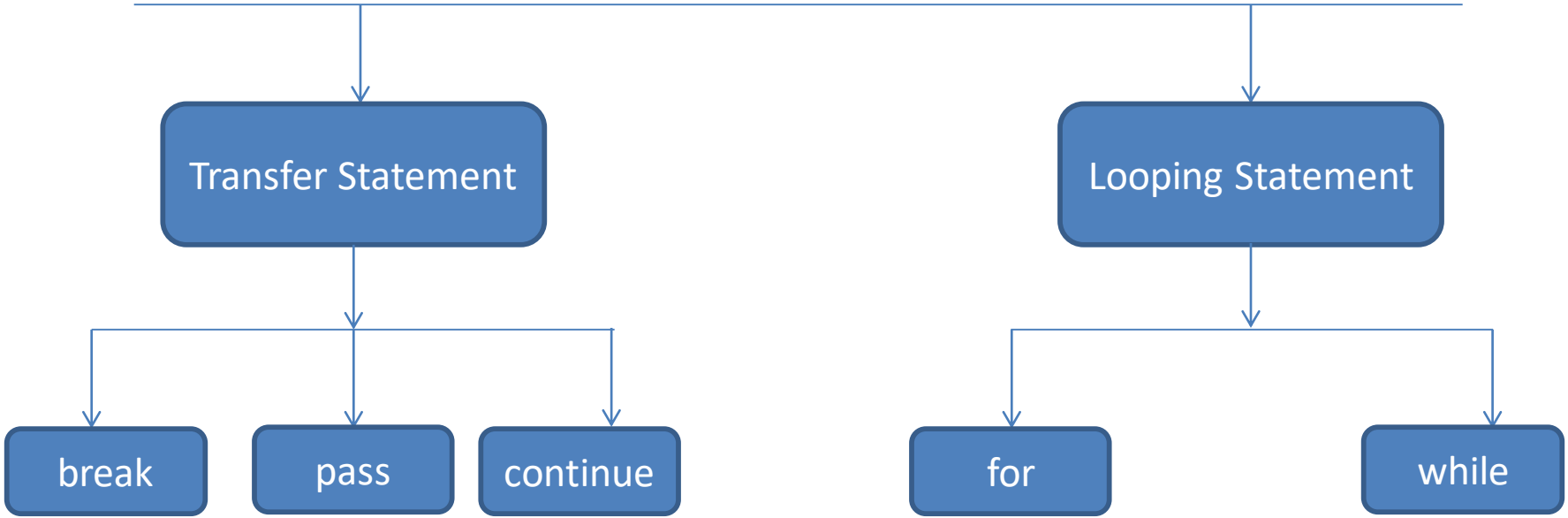
For eg => a= “help4code”
print(‘e’ in a) ==========➔ true
print(‘f’ not in) ==========➔ true

Conditional Statement

There come situations in real life when we need to make some decisions and based on these decisions, we decide what should we do next. Similar situations arise in programming also where we need to make some decisions and based on these decisions we will execute the next block of code.

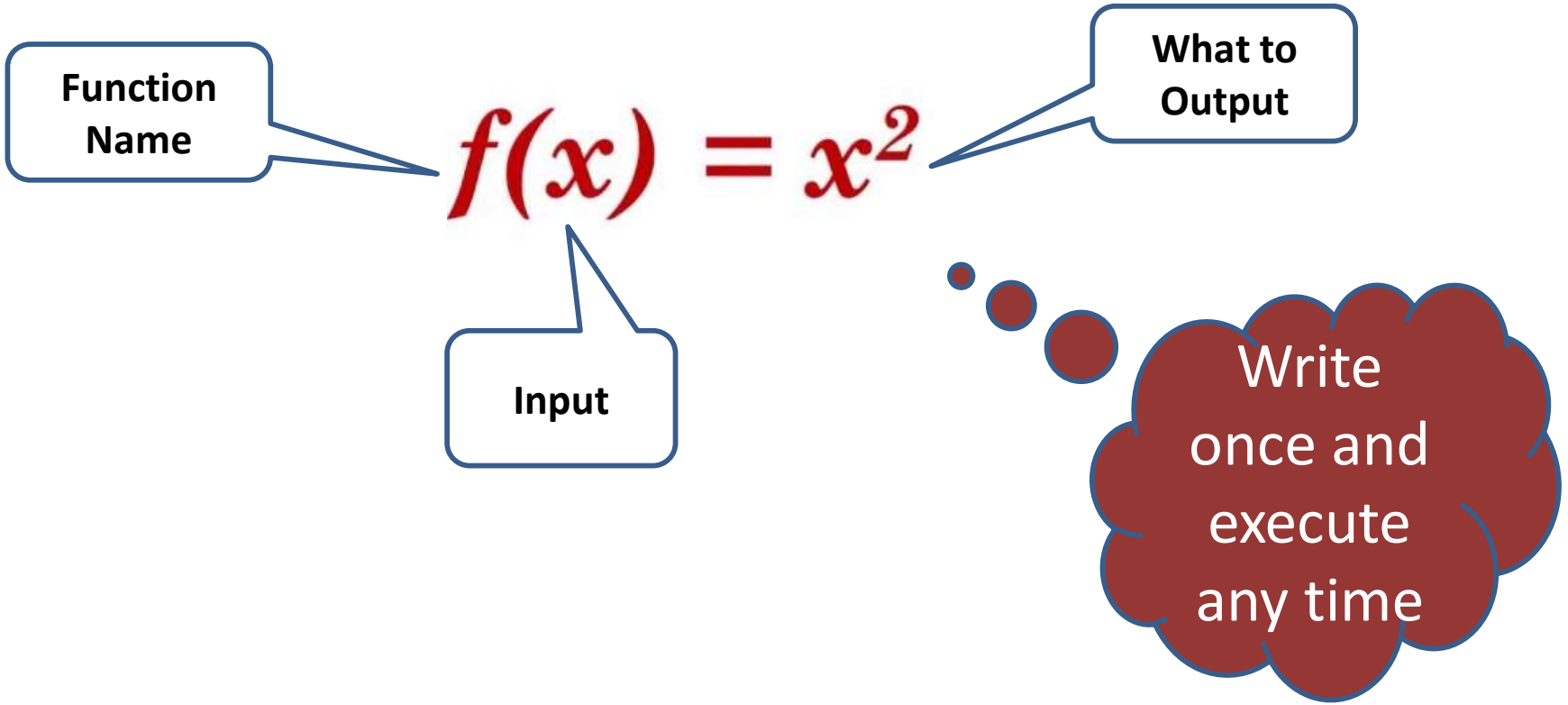
Conditional statement/Decision making statements in programming languages decides the direction of flow of program execution. Decision making statements available in C or C++ are:


Conditional statement
If condition:
State-1
State-2
Simple if
If condition:
State-1
else:
State-2
if else
If condition:
State-1
elif:
State-2
elif:
State-3
else:
default state
if-elif-else


Transfer Statement
Looping Statement
break
pass
continue
for
while

Function In Python

Modular Programming



What is function

Function is a self contained block which is executed separately
If suppose we want to execute a single statement or group of statement again and again then there is no need write this statement each time , so we can define this statement as single unit so as per our requirement we can call it any number of times So this unit is function

By using function program readability and reusability both will happen

Two Types of Function

1. Built in function
2. User define function

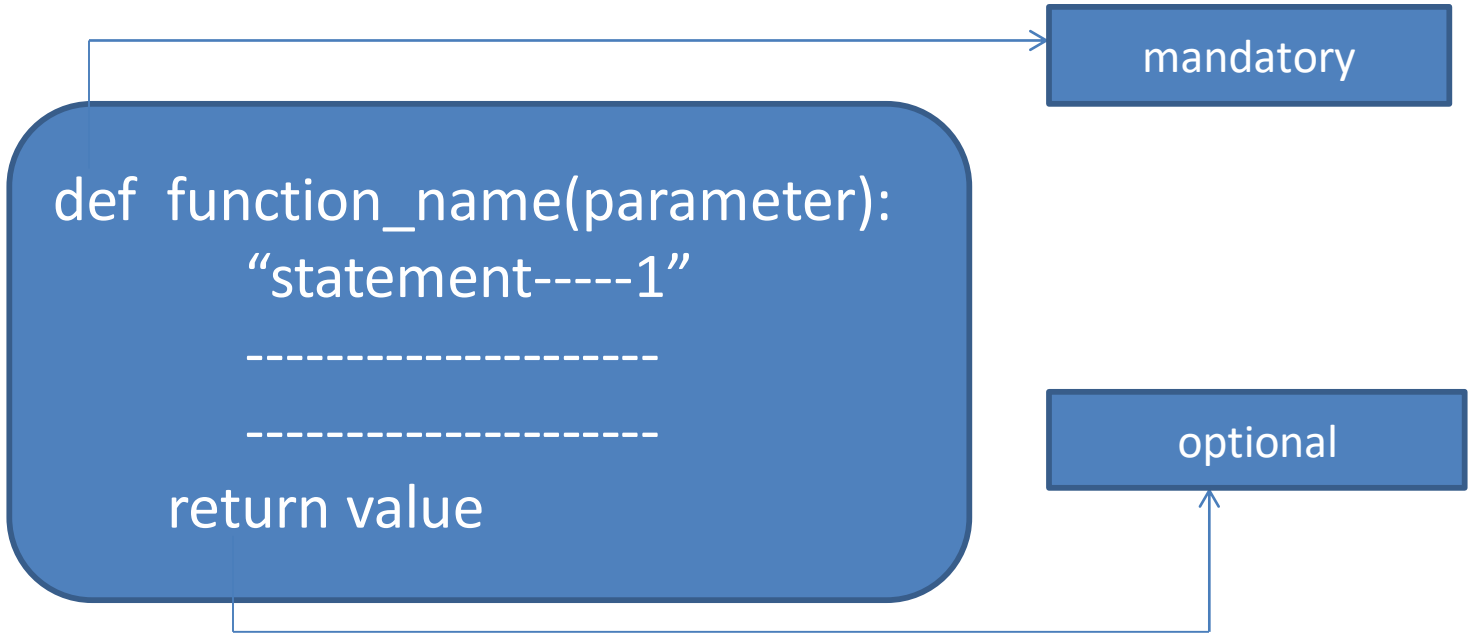
Built in function

Built in function is also known as predefine function or this function is already present inside of python library we can say, that comes along with python automatically so that this is called built in function

Eg:= type() , id(), input(), print(), eval()

Used Defined Function

According to program requirement programmer can developed the function this is called user defined function.



Types of argument in python

There are four types of argument are possible in python

1. Default argument
2. Positional argument
3. Keyword argument
4. Variable length argument

Types of Variable

There are two types of variable

1. Global variable
2. Local variable

Global variable

The variable which are declared outside of function are called as global variable and that are accessible through out the program

Local variable

The variable which are declared inside of function are called as local variable which is scope specific

global keyword

There are two uses of global keyword

If we want to declare global variable inside function

If it is available inside in function so we can do required operation

Recursive function

The function which call itself are called as recursion

Two main significance of recursive function

1. The length of the code will be reduce and easy to read
2. The complexity of problem can easily reduce

Anonymous function

The nameless function are called anonymous function or lambda function

Generators

This is function and which generate sequence wise value

Here the declaration is similar as like normal declaration of function but for returning value it uses yield keyword

Advantages

1. Generators are very easy
2. It improves memory utilization as well as performance
3. If we want to read data from large file then generator are best choice

Modules

Modules in python

The group of function, variable, classes saved in python as a single file then it is called as python module, an independent program in python are called as python module.

Every .py file act as a python module because it contain everything

If we want to access any function variable or class from one program to another program then we have to use import keyword

Syntax:-

import module_name

By using dir() function we can find member of module

Special variable __name__:

Every python program __name__ variable containing internally this variable tells that the program executed as individual program or as module

If that program executed as individual program then the value of that variable is __main__

If the program executed as a module from some other program then the value of this variable is the name of module where it is defined.

Here we can identify the program is executed as program or as module

Package In Python

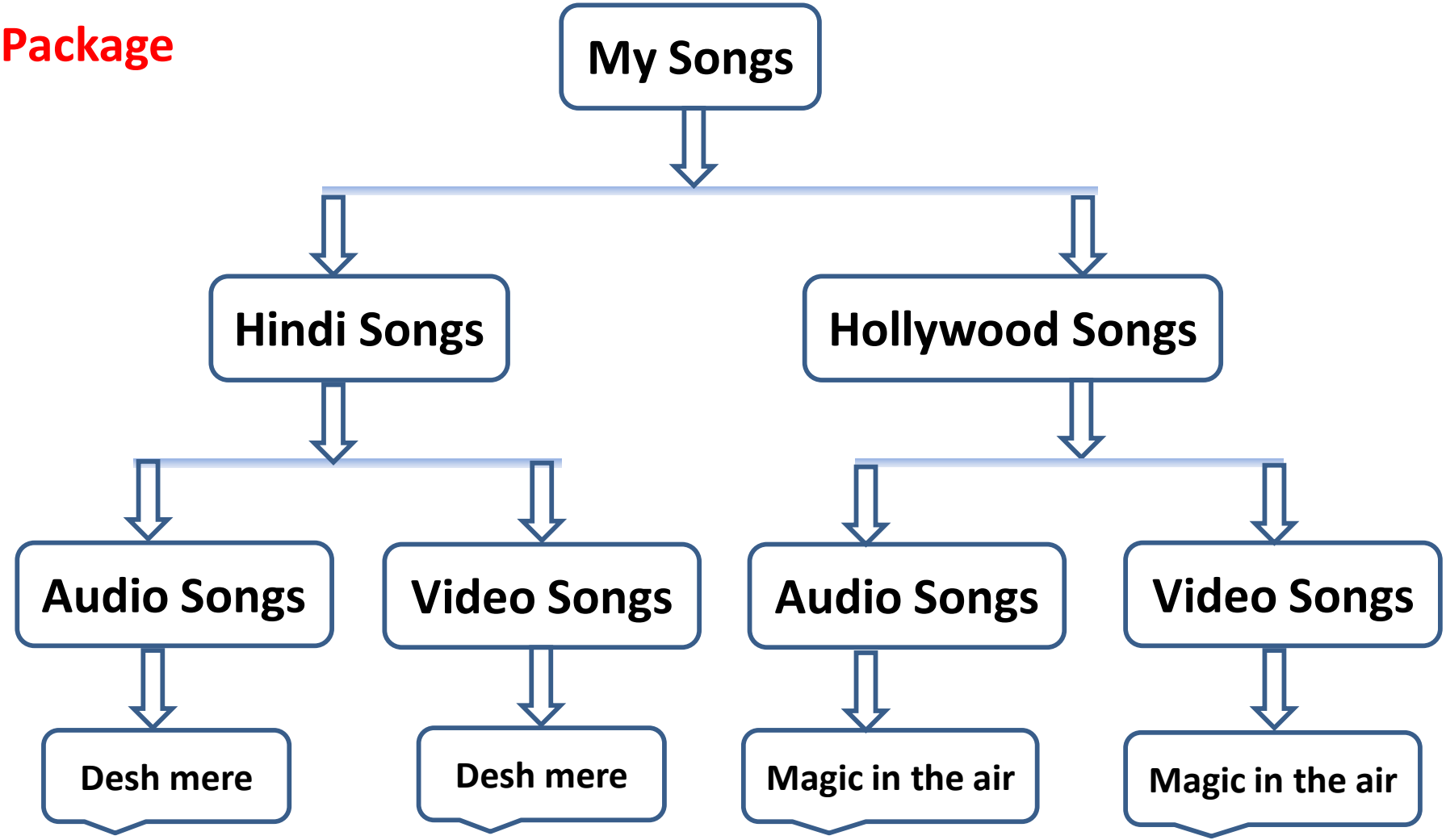
Package

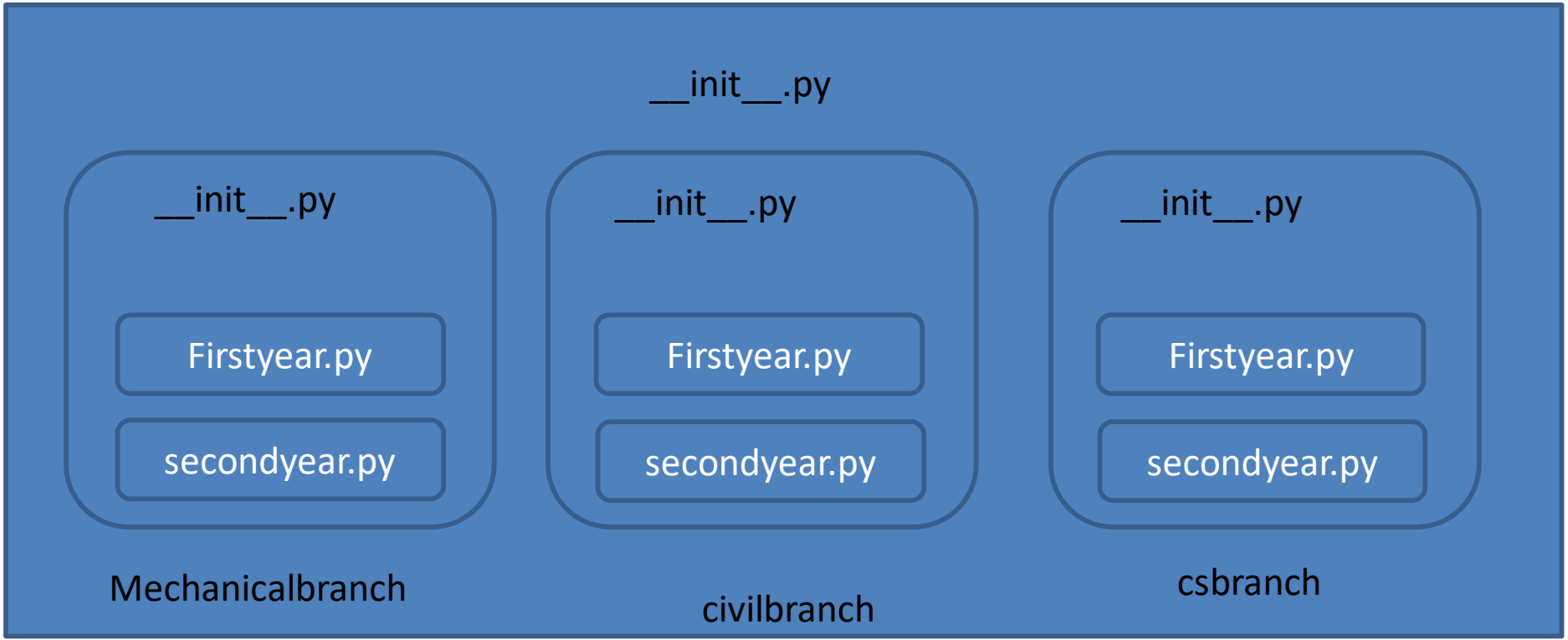
By using package we achieve encapsulation concept because here we bind or store group of modules in a single main package and as we know that module is a collection of variable, function, class.

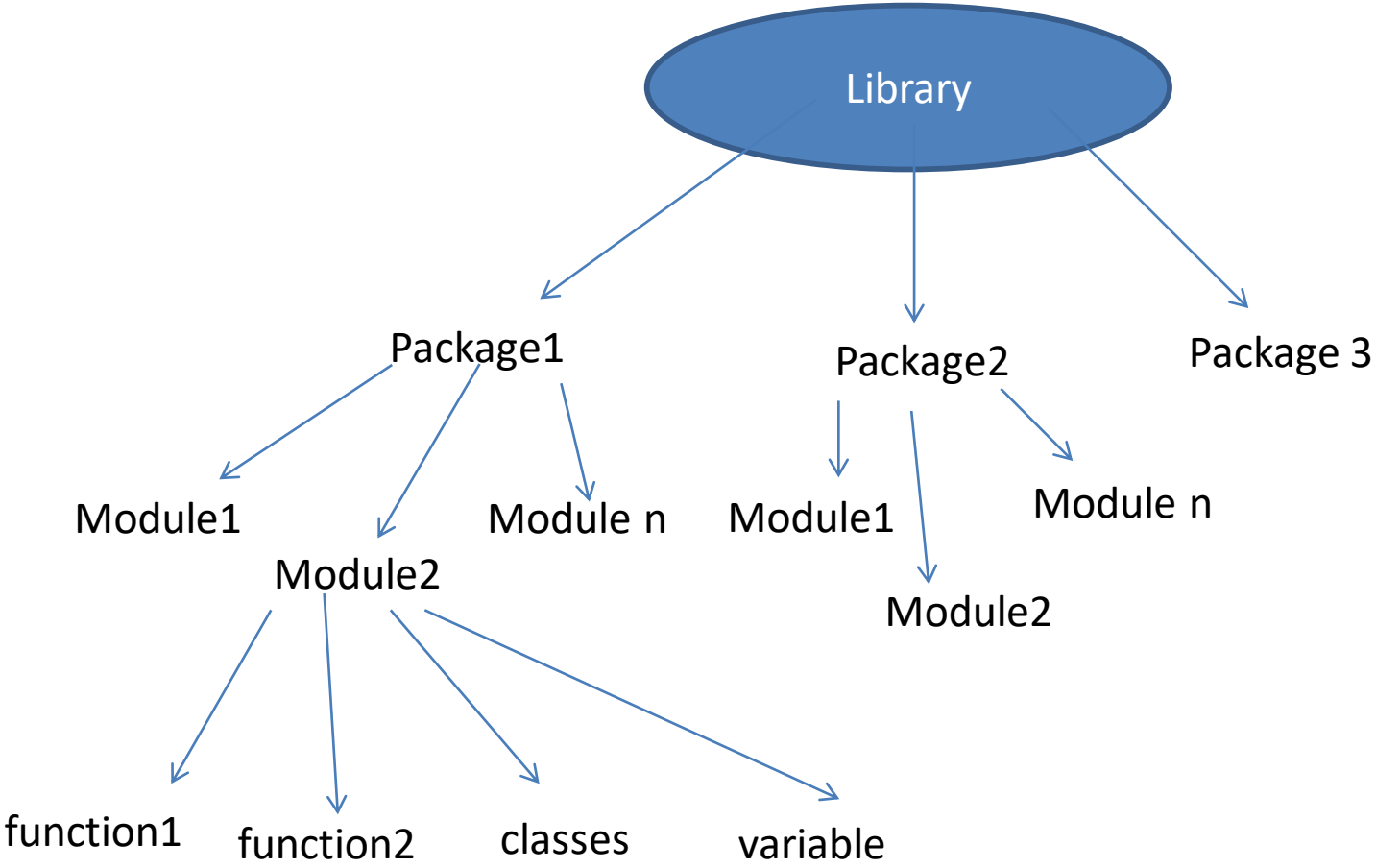
Package concept is as similar as like we create folder or directory in a computer or in drive.

Only those folder or directory which contain __init__.py empty file which will be considered as python package we can create package and sub package also.

Package


My Songs
Hindi Songs
Hollywood Songs
Audio Songs
Video Songs
Audio Songs
Video Songs
Desh mere
Desh mere
Magic in the air
Magic in the air


__init__.py
__init__.py
Firstyear.py
secondyear.py
Mechanicalbranch
__init__.py
Firstyear.py
secondyear.py
civilbranch
__init__.py
Firstyear.py
secondyear.py
csbranch
EngineeringCollege


Library
Package1
Package2
Package 3
Module1
Module2
Module n
Module1
Module2
Module n
function1
function2
classes
variable

Advantages of Package

1. By using package concept there is no naming conflicts because due to folder concept we can easily differentiate name of particular folder

2. Folder can be identify uniquely

3. Due to this we can implement modularity concept in our project

4. Package help us to achieve less time consuming efforts

File Handling In Python

Binary files

Whenever we have to store binary data like image file , video, audio files so for dealing this file we have to deal with binary files

Open()

This is a predefine function if we want to perform read or write operation so before starting read or write operation we have to open a file and for that purpose we use open().

But here also we have to specify mode of opening

F= open(filename , mode)

r := read mode

1. It open existing file for only read mode
2. File pointer will be at beginning of the file
3. If file is not exist the we will get error

w := write mode

1. It open existing file for only write mode
2. If the file already having some data then that will be overridden
3. If the file is not present that this mode will create new file

a :=

1. It open existing file for append mode
2. If the file already having some data then that will not be overridden here
3. If the file is not present that this mode will create new file

r+ :=

1. It open existing file for both read and write mode
2. If the file already having some data then that will not be deleted here
3. File pointer will be at beginning of the file

w+ :=

1. It open existing file for both write and read mode
2. If the file already having some data then that will be overridden here

a+ :=

1. It append file and read data from the file
2. If the file already having some data then that will not be overridden here

x :=

1. It open file in exclusive creation mode for performing write operation
2. If the file already

If we want to work with binary file then mode will be little bi different
rb, wb, ab, r+b, w+b, a+b, xb

e.g. := f = open(“myfile.txt”, “w”)

close()
After performing all operation in file we need close the file and for closing file we have to use command close()
f.close()

Properties of file object

After opening a file we will get file object so with file object and its properties we can access Data from file

with statement

1. When we opening file at that time we can use this statement
2. Inside of with statement block we can make group of operation
3. Here there is no need to close a file externally
4. With statement take the responsibility to close file
5. If any operation containing any error still it will close file

tell:=

From beginning of the file tell() function return the current cursor position that is file pointer Indexing in the file start from zero

seek():=

When we want cursor or file pointer at specified position inside file then we should use seek() function

Syntax:- filepointer.seek(offset, from)

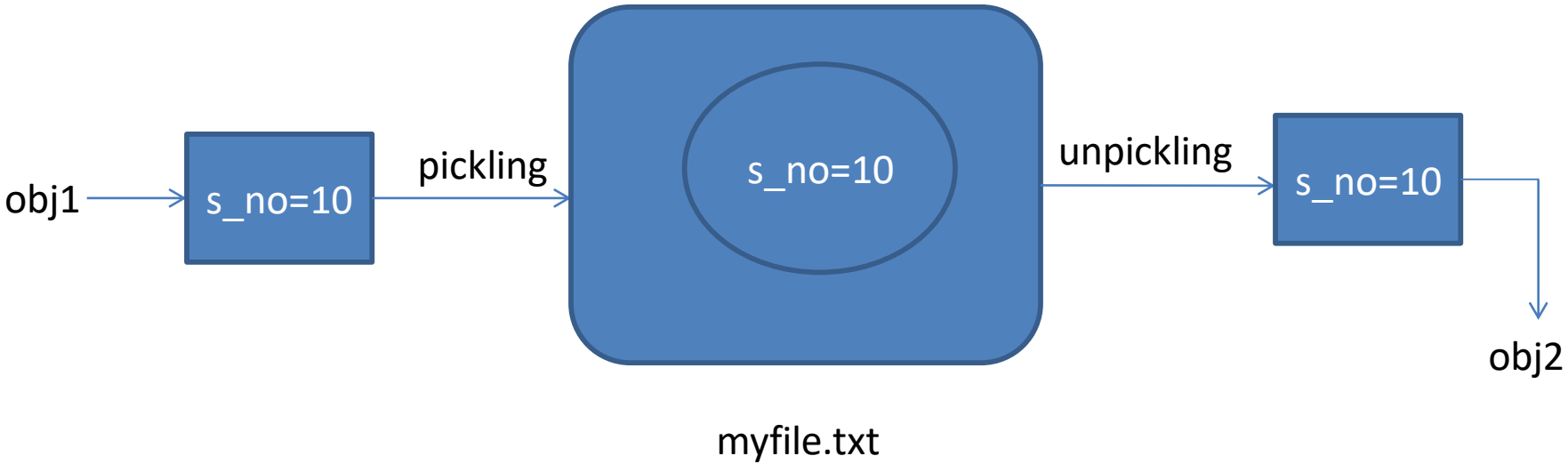
What is pickling and unpickling of objects

Pickling it is one of the way to convert the python object like list, dict , etc. into a character stream, actually the character stream contains all the information to reconstruct the python object in another python script

there is dump method which dump which dumps an object to a file object and second one is load which is load from file object.

The process where we dumping the object to the file is called pickling and loading the object from the file object is called unpickling

1. dump() function to perform pickling
2. load() function to perform unpickling



Exception Handling

Before starting exception handling we should know about types of error

1. Syntax error
2. Runtime error

Syntax error comes when we do not follow proper syntactical rule means predefine grammatical rules of python language

Runtime error comes due to incorrect logic or wrong input given by end user and run time error also known as exception

Exception handling this concept can be only implemented for runtime error but not for syntax error because syntax error can be solve by or correct by programmer

Syntax error comes at compile time so there is no meaning to write exception handling code here

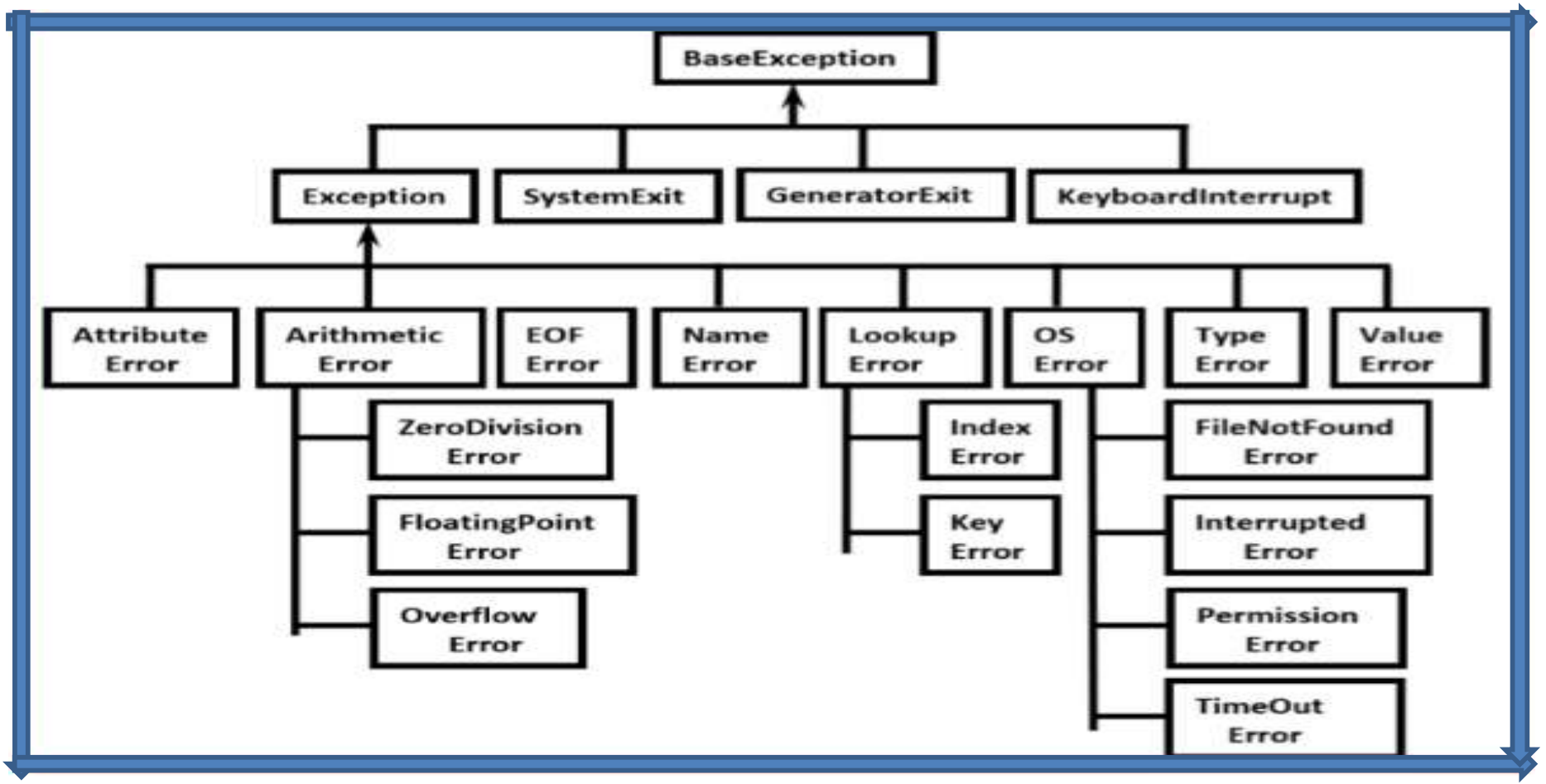
But runtime error comes at runtime so here there is max to max chance that to generate exception

What is exception

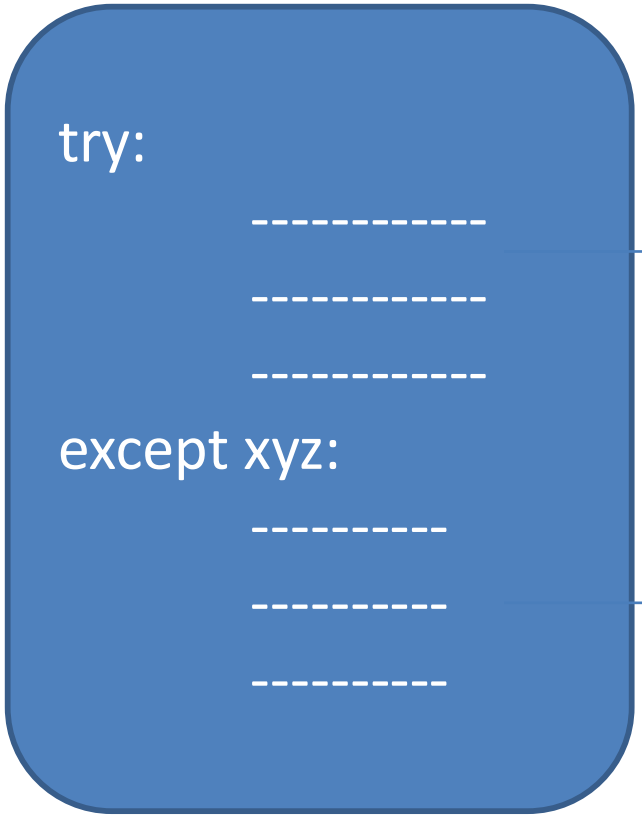
As we know that when we execute any program we can say that a normal flow of execution Has done so when due to any incorrect logic or a unexpected wrong input given by from user side which disturb a normal flow of execution so this is called exception

Exception handling means suppose if any runtime error generate in any part of the code so except that part rest of the code should be executed and error part will be handled by exception handling code so our normal flow of execution will be continue

Exception hierarchy



Exception handling by try and except



Here inside of try block we write those code where there will be probability that exception can raise

Here inside of except block we write exception handling code

Types of exception

Predefine exception

The exception which comes automatically by python virtual machine whenever any abnormal condition occur ,that are called predefine built in function

E.g) ZeroDivisionError, ValueError

User defined exception

Some time programmer want to indicates that something goes wrong And declare explicitly by raise keyword that are called as user defined exception or also we can say customize exception

How do we declare user define function

If we want define user define exception then we have to inherit Exception class because in python every exception are python class

Logging with python

If we want to store complete application flow and exception details in a file so we should go for python logging.

Logging help us to maintain statistics of a module , we can use log file
It help us in debugging to track the application suppose if any unwanted event occur so due to that log file we can find the problem very fast and if we don’t have log file then finding a problem is hard

Logging levels in python

As we know that when we are going to deal with log data so that we will not get same kind of information we will get types of information so the log data has divided into 6 level in python

Implementing logging

We have to create the file because we can store message in a file and also we have to specify the level of message because as we know that each level containing different information.

Level	Value	Details
CRITICAL	50	This indicates the serious problem so the program is not able to perform any function
ERROR	40	This is also indicating that software having serious problem and some part or functionality of software not work
WARNING	30	This indicates the warning message that means in future the unexpected event can occur in software
INFO	20	This indicates the some important message related to working
DEBUG	10	This will give the debugging information
NOTSET	0	This indicates the level has not set

Python Assertions

Python assertion is as debugging tool programmer put assertion in a code and as we know that programmer always wants to be true so suppose any of that is fail then it will not execute rest of the code.

By using assert keyword we can do this task

This will take input as Boolean condition if it return true then it doesn’t return anything but if it return false then it raise assertion error. Based on our requirement we can enable or disable assert statement

Types of assert statement

There are two types of statement

1. Simple version
2. Argument version

Simple version

Syntax :- assert conditional_statement

Argument version

Syntax :- assert conditional_statement, message

Python object oriented programming

Class in python

- Class is a way which binds data member and its associates function together within a single unit.
- Any variables declare within a class are called data members, any functions declare within a class are called members functions.
- Everything in a python a object
- Class properties can be represented by member variable
- In class action is done by member function

Syntax of class:-

Class classname:
    “statement”
        |
        |
    member variables
    member function

```
Class Student:

    rollno = 101

    def show(self):
        print(self.rollno)
```

In a class if we have to represent data so by using variable we can represent

There are three types of variable we can declare inside a class

Instance variable

instance variable declare at object level

Static variable

static variable declare at class level

Local variable

local variable declare at function level

There are three types of method we can declare at class

Class method
Static method
Instance method

Object:-
It is a instance of class. It is a basic runtime entity of an object oriented system
By using object we can communicate with data member and member function of the class

Syntax of object :-

referencevariablename = classname()

Ex:- obj = Student()

Self variable:-

1. by using self variable we pointing to the current object
2. self is a default argument
3. self is like this keyword in java
4. instead of writing self we can write any other name
5. in class level or in function first argument must be self

Constructor

1. Here in python the name of the constructor is __init__(self).
2. Constructor call automatically whenever we create object or we can say that at the time of creation of object.
3. By using constructor we initialize the object.
4. There is no mandatory to declare a constructor inside class but if we are not going to declare constructor so python provide default constructor.
5. Constructor will execute only one time when we create a object.
6. Always we have to pass one argument in constructor(self)

Types of variable

Instance variable:-

Instance is a synonymous name to object or we can say that any value that varied from object to object that are called as instance variable

Ex:- 50 is a instance of type int

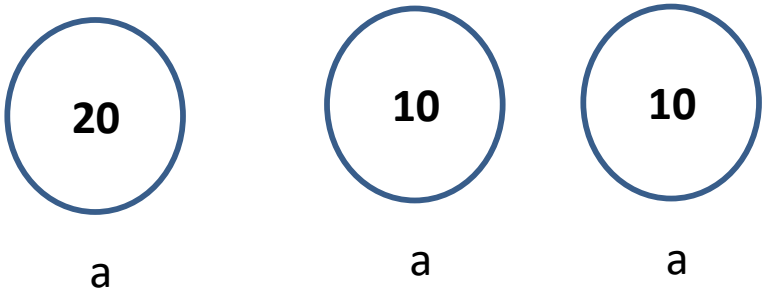
For every object a separate copy of instance variable created

We can declare instance variable using self keyword inside a constructor

Types of variable(instance var)

```
Class New:
        def __init__(self):
              self.a = 10

Obj1   = New()
Obj2   = New()
Obj3  =  New()
```

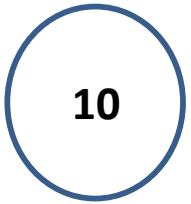


Types of variable static variable

```
Class New:
        self.a = 10

        def __init__(self):
                self.name="prashant"

Obj1   = New()
Obj2   = New()
Obj3  =  New()
```



In instance method we can also declare using self variable
By using object reference variable outside class

Static Variables:-

We can declare static variable in class level and outside of method
So we can say that it is in class level instead of instance level , static variable is initialize only one time in it’s life cycle, it is exist only single instance in every class.
Only one copy of static variable is created for total class and this will be accessible for all object of the class.

How to access static variable ? There are two ways to access static variable the first one is by class name or second way by object but programmer recommended to access by class name

By using object or self variable we can we can access or read static variable but we can not do modification or deletion.

If we try to delete then we will get error

Types of methods

There are three types of methods

1. Static method
2. Instance method
3. Class method

Instance method:-

If any instance variable we are implementing inside of any method then that method will be called as instance method.

Instance method should have self as parameter.

Static method:-

1. We can use static method when the code that belongs to class it do not use the object.

2. St they are not dependent on the state of the object

3. in a static method we will not use any instance or class variable.

4. Self and cls argument doesn’t required here to declare

5. We can declare static method using @staticmethod decorator

2. Class Methods:

Inside method implementation if we are using only class variables (static variables), then such type of methods we should declare as class method.

We can declare class method explicitly by using @classmethod decorator. For class method we should provide cls variable at the time of declaration
We can call classmethod by using classname or object reference variable.

Inner class:-

When we declare class inside another class that are called as inner class

Why we should go for inner class?

When without existing one type of object there will be no chance to exist another type of object that time we should go for inner class

Syntax*:-* *class classname:*

----

class classname:

Garbage Collection in python

As a programmer in C and C++ language we have responsibility to creating and deleting a objects.
In development time we know that where and when we have to create a object but at the deleting time we forget to delete the object and at runtime of application , so many unused objects take memory so memory related issues may come over there.

In python we have garbage collection concept which running in background continuously to finding unused object and delete that objects also.

By using garbage collector the probability of application failure due to memory issues is very less.

Destructors:-

Destructor is a special type of method, for destructor we use __del__
Internally garbage collector and destructor maintaining very good co-ordination, how? Let see…….

Before destroying object garbage collector call destructor to perform clean up operation like resources deallocation closing a database connection . Garbage collector waiting till the destructor completing its work because after that garbage collector destroy objects.

Polymorphism in python

Polymorphism

- Polymorphism means having many forms.
- We can define polymorphism as the ability of a message to be displayed in more than one form.
- Eg: A person in real world can perform many task at a time like at the same time can watch TV, can eat food, can receive a call

In C++ there are two types of polymorphism

- **Compile time Polymorphism:**

  eg: function overloading

- **Runtime Polymorphism:**

  eg: function overriding

How the polymorphism implemented here

Overloading in python

1. Operator overloading
2. Method overloading
3. Constructor overloading

Overriding in python

1. Method overriding
2. Constructor overriding

Function overloading

A function having same name but different arguments then such type of function is called as function overloading

Operator overloading:-

Here operator overloading is same operator for multiple purpose
Is known as operator overloading.
Here python supports operator overloading

E.g + operator can do addition and string concatenation in case of string data

Method Overriding

Method overriding co relating with inheritance because here parent and child relationship comes under the consideration. In case of inheritance parent class property by default available for child class also that means child can also use parent property.

So suppose child is not happy with parent class property then child can redefine its own property as per the requirement, This is called as method overriding.

Overriding supports both method and constructor overriding.

Inheritance in python

Extending property from one class to another class is called inheritance.

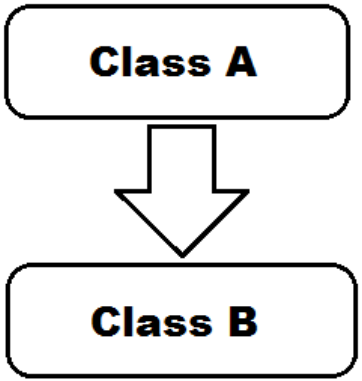
Directly we are getting here reusability concept

- **Base class:** A class which inherits its property to another is called base class or parent class.
- **Derived class:** A class in which properties are inherited called as derived class or child class.

types of inheritance

1. Single Inheritance
2. Multilevel Inheritance
4. Multiple Inheritance

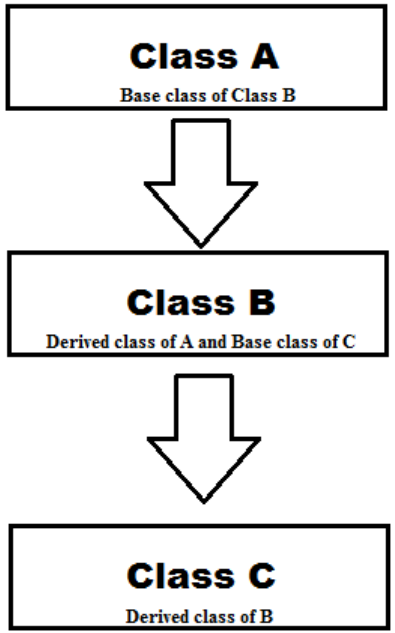
Single Inheritance



syntax

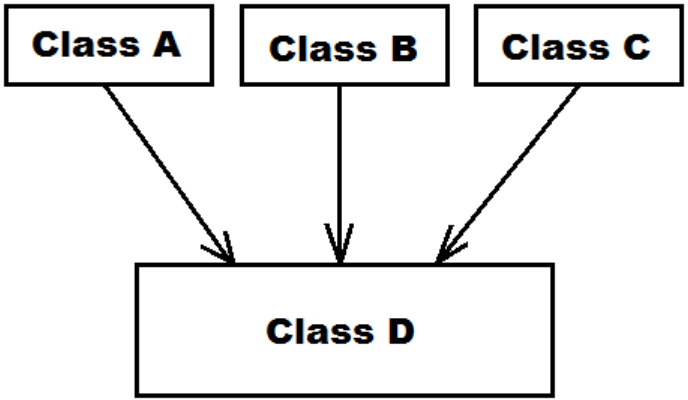
class derived-class(base-class):

Multilevel Inheritance




```
class GrandFather:
        ---------------------------
        ---------------------------
class Father(GrandFather):
        ---------------------------
        ---------------------------
class Child(Father):
        ---------------------------
        ---------------------------
```

Multiple Inheritance



Class Parent1:

Class Parent2:

|

|

Class Parent N:

Class Derived(Parent1,Parent2,...)

Regular expression in python

Some time as per the requirement according to some format and search pattern if we have to represent the string then obviously we have to go for regular expression

We can say that if string containing specific search pattern then we should use RegEx

E.g if we have to represent mobile number then we can use RegEx

Python re module provide support to work with regular expression.

Application of regular expression

1. Regular expression used to develop the digital circuit
2. Regular expression used to develop the compiler and interpreter
3. Regular expression used to develop the communication protocol like TCP/IP etc.
4. Regular expression used to develop the validation logic
5. Regular expression used to develop the pattern matching and searching application like ctrl-f

More about RegEx

Regular expression patterns are compiled in bytecode which is executed by matching engine
And the engine is written in C-language.

This engine execute and write the **re** module to produce bytecode which run faster.

For performing all the application **re** module having various function to execute different types and level of task

Compile() function

This function available inside **re** module and this function used to compile the patterns into regular expression object and which have various methods, which execute various operation like pattern matching

patternobj = re.compile(‘[a-z]’)

finditer() this return iterating object which match the object for every match from left to right direction
objmatch = patternobj.finditer(“pythonispython”)

There are three methods that we can use on match object
start() , end(), group(). This all method return something
Like start() method return starting index of the matching
Here end() method return end+1 index of the matching
group() this return the matching string of the group

Match character classes

By using this character classes we match the group of characters

- [abc]===>Either a or b or c
- [^abc] ===>Except a and b and c
- [a-z]==>Any Lower case alphabet symbol
- [A-Z]===>Any upper case alphabet symbol
- [a-zA-Z]==>Any alphabet symbol
- [0-9] Any digit from 0 to 9
- [a-zA-Z0-9]==>Any alphanumeric character
- [^a-zA-Z0-9]==>Except alphanumeric characters(Special Characters)

Inbuilt character classes

\s => Space character
\S => Any character except space character
\d => Any digit from 0 to 9
\D => Any character except digit
\w =>Any word character [a-zA-Z0-9]
\W=>Any character except word character (Special Characters)
. => Any character including special characters

Quantifiers

Quantifiers used to specify the number of occurrence to match

a => Exactly one 'a'
a+=> At least one 'a'
a*=> Any number of a's including zero number
a?=> At most one 'a' i.e either zero number or one number
a{m} => Exactly m number of a's
a{m,n} => Minimum m number of a's and Maximum n number of a's

match() function

For performing match operation we need string ,this match function used to match the given pattern to starting or beginning of the string. If match is done then we will get match object else we will get None.

fullmatch()

As a name suggest when we have to match full string with the given pattern then we have to use fullmatch() function. If match is done then we will get match object else we will get None.

Search() function

If the match found anywhere in the string then it return object else it will return None

findall() function

This function return a list which containing all matches

sub() function

This function perform substitution or replacement

re.sub(expression,replacement,string) here every match pattern will be replaced by provided replacement

subn() function

It is as similar as sub() function only one thing is different that it also return number of replacement. This return in tuple where first element is string and second one is number of replacement.

split() function

This function is used to split the given string as per the some pattern then we should use split() function