**Day-01**

**07-02-2025**

**===============**

**Python Fundamentals:**

**===============**

**Syntax:**

**======**

-> set of rules

**High level programming languages:**

**=======================**

-> three types on the basis of data:

 1) Statically Typed

 =============

 Ex: C

 -> The assigned value can be get rounded according to the type of the variable.

 Ex: int a = 12.2;// 12.2 (float) can convert into an integer

 2) Dynamically Typed

 ===============

 -> For Dynamically typed languages, we need not required the type before the variable name.

 -> Based on the assignment of the value, the type of the variable can be changed.

a = 12.3;

b = 32.1;

print(type(a))

print(type(b))

 3) Strongly Typed

 ============

 Ex: Java

 -> To the variable we need to assign a value based on the specified type

 Ex: int a = 12; // correct

 int p = 0.001l// incorrect

-> High level programming languages are classified into three types:

 based on the usage

 1) Functional programming languages

 =========================

 Ex: C

 2) Object based Programming languages ==> VB Script

 3) Object oriented Programming languages ==> Python, Java, C++, C# etc.

OOPs (Object Oriented Programming System)

 Concepts:

 class, Object, method, constructor, Destructor, Garbage Collection etc.,

 Principles:

 Encapsulation, Inheritance, Polymorphism, Abstraction

**What is Python?**

**===========**

-> It is one the high-level programming language

-> Easy

-> General Purpose programming language

**Features**

**======**

1) Dynamically typed programming language

 based on the assigned value the type of the variable can be automatically detected by the python.

2) Object Oriented Programming Language

3) Open source

4) Free

5) Platform independent language

**Python Version History**

**===============**

-> Guido Van Rossum

Python-0.0 🡺 1991

1.0 🡪 1994

1.1 🡪 1994

2.0 🡪 2000

2.1 🡪 2001

3.0 🡪 2008

3.1 🡪 2009

3.10 🡪 2021 etc.,

Current version of python ==> 3.13.1

**Day-02**

**10-02-2025**

**===============**

**Python software setup**

**===============**

1) Python is one of the open source

2) free software

-> Before going to install the python software, first we should check whether the python software is already existed or not.

 i) open command prompt(cmd in search bar)

 ii) type a command:

 python --version

 if "python version" is displayed, then we should understand that python is already existed.

 otherwise:

 we should install.

**Installation process:**

**==============**

1) Based on your system configuration,

 you should download the python software (.exe file)

2) To install the python software in our computer:

 click on the downloaded exe file to make it run.

3) In the new wizard:

 select an option "Add python.exe to PATH"

4) In same wizard, we have two options:

 1) Install now ==> can make install the software in "c-drive of your computer"

 2) Customize Installation => can make install the software into required folder

5) Click on "Customize Installation"

 then click on "next"

 click on browse

 select your location

 click on install

How to write python program

6) Click on close.

Check whether the python software has installed successfully or not:

========================================

i) open command prompt(cmd in search bar)

ii) type a command:

 python --version

**How to write program using python?**

**========================**

Program ==> collection of statements

statements ==> instructions to the computer.

Ex: Finding the sum of two numbers

========================

1) taking two numbers as per the requirement

2) performing the sum on those two numbers.

3) printing the result

-> To write any program, the programmer need five types of statements:

 1) Input statement

 2) Output Statement

 3) Memory Statement

 4) Control Statement

 5) Arithmetic and logical statement

-> we need 5 types of tokens:

 1) Keywords

 2) Identifiers

 3) Variables

 4) Literals

 5) Strings

-> By using above requirements, we can develop two types of programs:

 1) Static program

 2) Dynamic Program

-> Static program:

 fixed input

 fixed output

for any number of executions.

-> Dynamic program:

 based on the input, the output of the program can differ.

-> To write python programs:

 we need either:

 1) Text Editors or

 2) IDE (Integrated Development Environment)

**Static program**

**==========**

Printing of "Hello World!"

================

-> Python is interpreter dependent language.

-> The interpreter can read only one statement at a time.

-> if the defined statement is correct, then only it can allow to check the next statement.

**using Text Editors**

**=============**

1) Open the notepad file

2) Write your program

3) Save that program file into your desired folder with any name but must be with ".py" extension.

4) To execute the python program:

 open the command prompt

 navigate to your python file location:

 Syntax:

 cd file\_location

 run the file:

 Syntax:

 python your\_name.py

Program:

======

print("Hello")

print("Good Evening")

print("Welcome to Ashok IT")

print("We have daily python development class at 6pm")

print("By Ravi")

**Day-03**

**11-02-2025**

**============**

**Python program structure:**

**=================**

Documentation

------------------

 ==> also called as "commenting section"

 ==> can define in three ways:

 1) Single-line commenting

 ---------------------------

 -> can always start with '#'

 2) Multi-line commenting

 -------------------------

 -> can allow to define with triple-quotes ('''abcd''')

 3) Documentation String (Doc string)

 ==> Suggestable

package section

------------------

 Ex: Ecommerce Application

 p1, p2, p3.....,p10

-> optional

-> packaging section can use to integrate all independent and unique modules into single unit.

import section

----------------

-> to take the information from one module to another module, we can use "import section".

Ex: import random

-> Optional

class definition

-----------------

-> class ==> reference datatype

-> class is a collection of data members and methods.

Syntax:

 class ClassName:

 data members

 methods

-> Optional

Note:

=====

Python is like C: it is a structured programming language

and like java: it is an object oriented programming language.

-> Hence, the python program without class and with class able to execute.

user-defined functions

-------------------------

Syntax:

 def function-name(parameters):

 statement-1

 statement-2

**Dynamic Program in Python**

**==================**

-> Based on the input, the output of the program can vary

-> such type of programs are called as "dynamic programs".

-> To write dynamic programs, we need three things:

 1) Memory statement

 2) Input Statement

 3) Output Statement

**1) Memory statement**

**==============**

-> Memory statement describe that how the memory can be created and how the value can be store.

-> To define the memory statement, we need:

 1) Variable

 2) Identifier

 3) Datatypes

Variable:

=======

-> is a name

which can use to store a value of any type.

-> also called as "container" or "named memory"

How to define the variable:

=================

Syntax:

 variable-name = value

-> The variable can define in two ways:

 1) Compile time

 2) Run time

Note

=====

Other languages like c:

 data definition:

 datatype variable-name = value;

-> But python is dynamically typed programming language

 so, the data type can able to understand based on the assigned value.

# How the python is dynamically typed

'''

**type():**

**======**

-> inbuilt function

-> can use to know the type of data

Syntax:

 type(variable-name)

Note:

====

type() can return the type of data in terms of class.

**Compile time definition:**

**================**

a = 10

b = 0.001

c = True

d = 'python'

e = 1-2j

print(type(a))

print(type(b))

print(type(c))

print(type(d))

print(type(e))

Note:

====

The static program allows the variable in compile time definition.

Dynamic program allows the variable in run time definition.

**How variable can work:**

**===============**

In python the memory hierarchy consisting of two parts:

 1) Stack memory

 2) Heap memory

-> The variable definition has two parts:

 1) Variable name

 2) Value

-> Variable name of the variable definition can store in "stack memory" and the value can store in heap memory

and then from stack memory to heap memory, there is a dynamic linking can be created.



Ex: print(a)

-> PVM (Python Virtual Machine) can search about 'a' in the stack memory

-> Once it will be identified, the with the help of dynamic linking, the corresponding value in heap memory can be identified. Then that value can print.

**Day-04**

**17-02-2025**

**=================**

**How to name the variable?**

**==================**

Using an identifier, we can name the variable.

**Identifier:**

**=======**

-> Identifier is a name

 which we can use to name any entity like: variables, functions, classes, methods, objects, packages, modules etc.

**Identifier Rules/Naming Conventions:**

**=========================**

1) For identifier, we can use the character set:

 i) Alphabets (Lower case/Upper case)

 ii) Digits (0 to 9)

 iii) Underscore sign (\_)

 only (no other chars allowed)

2) Identifier never start with digit (first character of an identifier never be a digit)

Ex: 9abc = 12 ==> Syntax Error

But identifier can start with either alphabet or \_

Ex: \_9abc = 12 ==> Valid

abc9 = 12 ==> Valid

3) No keyword we can use as an identifier.

Ex: import = 100 ==> Syntax Error

4) Identifiers are case sensitive.

-> Identifiers can allowed to define in 5 different cases:

 pythonprogramsareeasy ==> Lower case ==> all chars in lower case format

 PYTHONPROGRAMSAREEASY ==> Upper case ==> all chars in upper case format

 PythonProgramsAreEasy ==> Title case ==> Each word of the identifier can start with capital and remaining all are in lower case

 Pythonprogramsareeasy ==> Capitalize case ==> first letter of the entire name in capital and remaining all in lower case

 pythonProgramsAreEasy ==> Camel case ==> from second word within the identifier start with capital letter and remaining all in lower case.

**What is an Error?**

**============**

-> Error is an unexpected behavior in the program which cause the interruption of the program.

-> Errors are classified into three types:

 1) Syntax Errors

 ==> the errors can be occurred because of mistakes in syntax.

 2) Run time Errors/Exceptions

 3) Logical Errors

Note:

====

To run the python file using command prompt, we need to use a command:

 python fileName.py

 (or)

 py filename.py

**Keywords:**

**=======**

-> Any programming language with Libraries.

-> Library is a collection of packages.

-> Package is the collection of Modules.

-> Module is the collection of classes, functions etc.

Any python file is behave as module.

-> Module is always with '.py' extension.

-> Keywords of python library called as "pre-defined words" also called as "reserved words"/

-> Every keyword has a definite meaning

 can use to define the specific functionality.

Ex: def ==> use to define the functions/methods

import ==> when we want import any data from external sources. etc.

-> In python total 35 keywords are available.

Q: WAP to display all python reserved words?

=============================

-> python keywords are defined in the module of python library named as "keyword"

-> when we want to display all the keywords of python, we must be import this module into our python program.

Syntax:

 import keyword

-> In the keyword module, all keywords were grouped and identified with a variable is called as "kwlist".

-> To get all keywords, we can use this syntax:

 keyword.kwlist

import keyword

allKeywords = keyword.kwlist

print(allKeywords)

Output:

======

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

Note:

====

-> Python keywords are categorized into two types:

 1) Reserved values/Literals==> 3

 2) Reserved words ==> 32

-> out of 35 keywords, 32 keywords are reserved words must be define/write in lower case only. And remaining three are called as "Reserved values/literals", those are allowed to define in capitalize case only.