WEEK-1

Fundamental Concepts: Brief history; features; applications of python; python distributions; versions; python IDEs; Python interpreter; Execution of python programs, debugging python code; Indentation, Comments; best practices for python programming; Character set; tokens; keywords, variables, naming rules for

variables, Assignment.

History of Python

- Python laid its foundation in the late 1980s.
- The implementation of Python was started in December 1989 by Guido Van Rossum at CWI in Netherland.
- In February 1991, Guido Van Rossum published the code (labeled version 0.9.0) to alt.sources.
- In 1994, Python 1.0 was released with new features like lambda, map, filter, and reduce.
- Python 2.0 added new features such as list comprehensions, garbage collection systems.
- On December 3, 2008, Python 3.0 (also called "Py3K") was released. It was designed to rectify the fundamental flaw of the language

What is Python?

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.

Features of Python:

1. Easy to code:

Python is a high-level programming language. Python is very easy tolearn the language as compared to other languages like C, C#, Javascript,Java, etc. It is very easy to code in python language and anybody can learn python basics in a few hours or days. It is also a developer-friendly language**.**

2. Free and Open Source:

Python language is freely available at the official website.Since it is open-source, this means that source code is also available to the public. So youcan download it as, use it as well as share it.

3. Object-Oriented Language:

One of the key features of python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, objectsencapsulation, etc.

4. GUI Programming Support:

Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in python. PyQt5 is the most popular option for creating graphical apps with Python.

5. High-Level Language:

Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

6. Extensible feature:

Python is a Extensible language. We can write some Python code into C or C++ language and also we can compile that code in C/C++ language.

7. Python is Portable language:

Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platforms such as Linux, Unix, and Mac then we do not need to change it,we can run this code on any platform.

8. Python is Integrated language:

Python is also an Integrated language because we can easily integrated python with other languages like c, c++, etc.

9. Interpreted Language:

Python is an Interpreted Language because Python code is executedline by line at a time, like other languages C, C++, Java, etc. there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called bytecode.

10. Large Standard Library

Python has a large standard library which provides a rich set of module and functions so you do not have to write your own code for everysingle thing. There are many libraries present in python for such as regularexpressions, unit-testing, web browsers, etc.

11. Dynamically Typed Language:

Python is a dynamically-typed language. That means the type (for example- int, double,

long, etc.) for a variable is decided at run time not inadvance because of this feature we don't need to specify the type of variable.

Applications of Python:

- AI and machine learning.
- Data analytics.
- Data visualisation.
- Programming applications.
- Web development.
- Game development
- Language development
- Finance.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can bevery quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.

Python Distributions:

Here is a brief tour of Python distributions, from the standard implementation (CPython) to versions optimized for speed (PyPy), for special use cases (Anaconda, ActivePython), for different language runtimes (Jython, IronPython), and even for cutting-edge experimentation (PyCopy, MesaPy).

Python Versions:

Python programming language is being updated regularly with new features and supports. There are lots of update in Python versions, started from 1994 to current release. A list of Python versions with its released date is given below.

Python Version	Released Date
Python 1.0	January 1994
Python 2.0	October 16, 2000
Python 3.0	December 3, 2008
Python 3.6	December 23, 2016
Python 3.7	June 27, 2018
Python 3.8	October 14, 2019
Python 3.9	Oct. 5, 2020
Python 3.10.0	Oct. 4, 2021

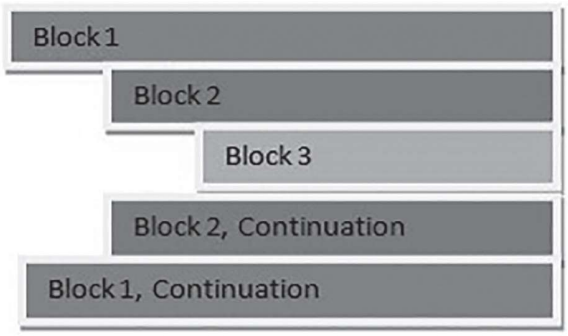
Python IDEs:

- An IDE (Integrated Development Environment) understand the code much better than a text editor.
- It usually provides features such as build automation, code lining, testing and debugging.
- This can significantly speed up the work.
- There are different IDEs which are as follows.

1. PyCharm
2. 2. Spyder
3. Eclipse PyDev
4. IDLE
5. Sublime Text 3
6. Atom
7. Thonny
8. Visual Studio Code
9. Vim

Python Indentation

- Indentation refers to the spaces at the beginning of a code line.
- Where in other programming languages the indentation in code isfor readability only, the indentation in Python is very important.
- Python uses indentation to indicate a block of code.



Python Comments

- Comments are non executable statements
- Comments can be used to explain Python code.
- Comments can be used to make the code more readable.
- Comments can be used to prevent execution when testing code.
- # is used to create single line comment
- Example: # This is my first line of code
- Python doesn't have syntax for multi line comment. you can use amultiline string
- multiline string (triple quotes) in your code, and place yourcomment inside it:
- Example: “”” This is multiline comment

  Enclosed within triple quotes”””

Character set

A character set is a set of valid characters acceptable by a programming language in scripting. In this case, we are talking about the Python programming language. So, the Python character set is a valid set of characters recognized by the Python language. These are the characters we can use during writing a script in Python. Python supports all ASCII / Unicode characters that include:

Alphabets: All capital (A-Z) and small (a-z) alphabets.

Digits: All digits 0-9.

Special Symbols: Python supports all kind of special symbols like, ” ‘ l ; : ! ~ @ # $ % ^ ` & * ( ) _ + – = { } [ ] \ .

White Spaces: White spaces like tab space, blank space, newline, and carriage return.

Other: All ASCII and UNICODE characters are supported by Python that constitutes the Python character set.

Tokens

A token is the smallest individual unit in a python program. All statements and instructions in a program are built with tokens.

The various tokens in python are :

1) Identifiers
2) Keywords
3) Literals
4) Operators
5) Punctuators

Identifiers

An identifier is a name given to a variable, function, class or module.Identifiers can be a combination of letters in lowercase (a to z) oruppercase (A to Z) or digits (0 to 9) or an underscore (_).

Examples: myCountry, other_1 , good_ morning all are valid examples.

Rules for naming Identifiers

- Python identifier must begin with an alphabet (A – Z and a – z) or an underscore
- Keywords cannot be used as identifiers.
- One cannot use spaces and special symbols like !, @, #, $, % etc. as identifiers.
- Identifier can be of any length.
- Identifiers are case sensitive

Keywords

- Keywords are a list of reserved words that have predefined meaning.
- Keywords have special vocabulary and cannot be used by programmers as identifiers for variables, functions, constantsor with any identifier name.
- Attempting to use a keyword as an identifier name will causean error.
- The following table shows the list of keywords of python

if	elif	else	and
while	for	assert	as
return	or	True	False
yield	try	catch	finally
is	in	del	import
except	lambda	with	pass
break	continue	from	class
nonlocal	global	raise	not

Python Variables:

- Variables are containers for storing data values.
- Python has no command for declaring a variable.
- A variable is created the moment you first assign a value to it.
- Variables do not need to be declared with any particular *type*, andcan even change type after they have been set.
- Example:x= 5
  y = "John"
  z = 6.8

Rules for Python variables:

- 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 andunderscores (A-z, 0-9, and _)
- Variable names are case-sensitive (age, Age and AGE are three different variables)

Assigning Values to Variables

The general format for assigning values to variables is as follows:

variable_name = expression

The equal sign (=) also known as simple assignment operator is used to assign values to variables. In the general format, the operand to the left of the = operatoris the name of the variable and the operand to the right of the = operator is the expression which can be a value or any code snippet that results in a value. That valueis stored in the variable on the execution of the assignment statement.

How to assign Variables Multiple Values

- Many Values to Multiple Variables

  Python allows you to assign values to multiple variables in one line:
  Example: x, y, z = "Orange", "Banana", "Cherry"
- One Value to Multiple Variables

  Python allows you to assign the same value to multiple variables in oneline:

  Example: x = y = z = "Orange"