**PYTHON LEARNING**

INTRODUCTION TO PYTHON

**1. History of Python**

1. High level programming language

2. designed by Guido von Rossum in 1991

3. emphasizes more on code readability

4. has been an inspiration to other languages as well

5. succeeded ABC which had an interfacing with Amoeba Operating System

6. Guido von Rossum began working on applications of Python in 1989 and released the language by 1991.

7. Named after a famous Tv series Monty Python's Flying Circus.

8. Python Version 1.0 came out 1n 1994 with features like Exception Handling, Map, Reduce, Filter and Lambda.

9. Version 2.0 came out in 2000 with features like Garbage Collection and List Comprehensions

2. Python Features

1. dynamic, high - level, free open source and interpreted language

2. supports object oriented programming concepts as well as procedural programming concepts

3. easy - to- code and is developer friendly language

4. Portable language: Python code can run across all platforms like Mac, Unix, windows, etc.

5. Interpreted language: Instead of compiling the code and running it, Python code gets converted into immediate byte code and the code gets executed

6. Supports object oriented programming

7. Dynamically typed language

8. Large Standard Library: Python supports multiple libraries and gives us rich set modules. Helps with code reusability and provides libraries for unit testing, web browsers and regular expressions

9. Integrated Language: Python code can be integrated with other C/C++ codes.

10. provides GUI programming: libraries like PyQT supports GUI designing, which helps in making GUI apps using Python

11. Extensible language: programs written in Python can be compiled in Java, or C/C++.

3. Interpreter

1. It is a program installed on the system, it looks at the python code and runs it.

2. it reads what you type, evaluates it and prints the output known as read-eval-print loop

3. Python interpreter consists of two parts: Python Virtual Machine and Compiler

4. Our source code gets compiled into a byte code and the byte code is interpreted by PVM and then converted into machine language

5. Interpreters can be helpful to try out code ideas. It gives a quick and fast idea of what a particular syntax can do

6. It has an >>> prompt

7. It is also helpful in looking for syntax directories

8. Lexing, Parsing, Compiling, Generating Bytecode, Interpreting, Generating Machine Code are the stages of Python Code

9. Lexer: breaks the line of code into tokens

10. Parser: Uses these tokens to create tree syntax aka AST

11. Compiler turns AST into code objects

12. Interpreter: executes the code objects and generate a machine code

4. Applications

1. Web Development with help of Django, Flask, Bottle, etc. and some libraries that offer amazing visualizations, security and fast development process.

2. AI and ML: With help of some amazing libraries and functions. It offers simple, concise and readable code. Examples of libraries are Numpy, Keras, SciPy, Seaborn

3. Gaming: With help of libraries such as pygame, panda3D games like Pirates of Caribbean, Bridge Commander have been made.

4. AV applications: Python is equipped with a lot of tools and libraries which help in building Netflix, Spotify, etc. This has been handled by libraries like Dejavu, Pyo, Mingus,etc.

5. Software Development because of factors like code reusability, platform independence, built in libraries, high compatibility

6. Desktop GUI: With the help of libraries like PyQT, amazing GUI can be developed

7. Business applications: provides excellent security and scalability features

**PYTHON BASICS- KEYWORDS,COMMENTS VARIABLES**

Keywords

1. These are special reserved words in Python with specific meaning and restrictions based on how they can be used
2. Fundamental building blocks of Python Programming
3. Cannot assign a value to these keywords, if tried to do so, it would return a Syntax error.
4. There are 35 keywords in Python
5. kwlist and iskeyword() are two most useful ways to determine keywords

Identifiers

1. These are special types of words that are user -defined.
2. We use this to represent variables, functions,classes,modules, etc.

Comments

1. A good way to increase code’s readability.
2. They are integral part of the program
3. Syntax for comment: #This is a comment
4. Comments should be short, sweet, and to the point
5. There are also multiline comments: ””” This is a multiline comment”””
6. Avoid smelly, rude comments
7. Commenting can also help in debugging code and using a piece of code for temporary use.
8. Commenting is a valuable tool
9. Commenting makes life of developers easy.

Variables

1. Variables are reserved memory locations
2. When you create a variable to save some memory space
3. In Python you don't need to declare variables before using them, you can directly assign a value to it and start using it
4. Variables can be broadly classified as global variables and local variables.
5. Global Variables are the variables that can be accessed even out of the current scope or throughout the program
6. Local variables are the variables that can only be accessed within a loop, function or a class
7. Variable are names attached to a particular object
8. Syntax: i = 100
9. Variables can be also thought of as symbolic name that is reference to an object
10. The assignment operator creates an object with a specific value attached to it.
11. There are also certain restrictions on how to name a variable. The first character cannot be a special character or a digit.
12. Some common conventions are CamelCase, PascalCase and Snake Case
13. In Python, snake\_case is used for functions,variables and PascalCase is used for class names.