**Introduction to Python**

1. Python Basics:
   1. What is Python:

* Python is a set of instructions that we give in the form of a Program to our computer to perform any specific task. It is a Programming language having properties like it is interpreted, object-oriented and it is high-level too. Due to its beginner-friendly syntax, it became a clear choice for beginners to start their programming journey. The major focus behind creating it is making it easier for developers to read and understand, also reducing the lines of code.
  1. What are key features of python:
* Easy to read and understand
* Interpreted language
* Object-oriented programming language
* Free and open-source
* Versatile and Extensible
* Multi-platform
* Hundreds of libraries and frameworks
* Flexible, supports GUI
* Dynamically typed
* Huge and active community

1. Installing Python:
   1. Describe the steps to install python on your operating system:

Step 1: Check Your System

* Before you begin, check if Python is already installed on your operating system. Open command prompt (python –version). If python is installed, it will display the version number.

Step 2: Download Python

* Visit the official Python website at python.org. and download the latest python version available for the operating system

Step 3: Choose the right version

* Python has two major versions: Python 2 and Python 3. Python 2 is no longer supported, so it's recommended to download Python 3. Choose the latest version of Python 3, which is displayed prominently on the website.

Step 4: Download the installer

* Python has two major versions: Python 2 and Python 3. Python 2 is no longer supported, so it's recommended to download Python 3. Choose the latest version of Python 3, which is displayed prominently on the website.

Step 5: Run the installer

* After downloading, locate the installer file and run it. Follow the installation wizard’s instructions. Be sure to check the box that says “Add Python X.X to PATH” during installation (replace “X.X” with the version number you downloaded). This is essential for easy command-line access.

Step 6: Verify the installation

* Once the installation is complete, open command prompt or terminal and run.

python –version

You should see the installed python version.

Step 7: Install a Code Editor (Optional)

* While Python includes the IDLE development environment, many developers prefer using code editors like Visual Studio Code, PyCharm, or Jupyter Notebook for a more robust coding experience. Download and install a code editor of your choice.

Step 8: Write your first python code

* Open your code editor and create a new Python file with the ".py" extension. For example, "hello.py."

1. Python Syntax and Semantics:
   1. Simple python program that print “hello, World!”

* //hello world program
* #include <stdio.h>
* Int main() {
* Printf(“ ! ! Hello World ! !);
* Return 0;
* }
  1. Explain the basic syntax elements used in the program
* Basic syntax represents the fundamental rules of a programming language. It includes the appropriate use of keywords, data structures, functions and loops.

1. Data types and Variables:
   1. List and describe the basic data type in Python
2. Numeric Data Type: Represents the data that has a numeric value. A numeric value can be an integer, a floating number, or even a complex number. The values are defined as Python int, Python float, and Python complex classes in Python.
3. Sequence Data Type: Is the ordered collection of similar or different Python data types. Sequences allow storing of multiple value in an organized and efficient fashion.

• Python String

• Python List

• Python Tuple

1. Boolean Data Type: Python Data type with one of the two built-in values, True or False. Boolean objects that are equal to True are truth (true), and those equal to False are false (false). However non-Boolean objects can be evaluated in a Boolean context as well and determined to be true or false. It is denoted by the class bool.
2. Set Data Type: A set is an unordered collection of data types that is iterable, mutable, and has no duplicate elements. The order of elements in a set is undefined though it may consist of various elements.
3. Dictionary Data Type: is an unordered collection of data values, used to store data values like a map, unlike other Python Data Types that hold only a single value as an element, a Dictionary holds a key: value pair. Key-value is provided in the dictionary to make it more optimized. Each key-value pair in a Dictionary is separated by a colon: whereas each key is separated by a ‘comma’.
   1. Write a short script that demonstrates how to create and use variables of different data types

* A data type is a classification of data which tells the compiler or interpreter how the programmer intends to use the data. Most programming languages support various types of data, including integer, real, character or string, and Boolean.

Discussion:

Our interactions (inputs and outputs) with a program are treated in many languages as a stream of bytes. These bytes represent data that can be interpreted as representing values that we understand. Additionally, within a program, we process this data in various ways such as adding them up or sorting them. This data comes in different forms. Examples include:

* your name – a string of characters
* your age – usually an integer
* the amount of money in your pocket – usually a value measured in dollars and cents (something with a fractional part)

|  |  |  |
| --- | --- | --- |
| **Data Type** | **Represents** | **Examples** |
| integer | whole numbers | -5, 0, 123 |
| floating point (real) | fractional numbers | -87.5, 0.0, 3.14159 |
| string | A sequence of characters | "Hello world!" |
| Boolean | logical true or false | true, false |
| nothing | no data | null |

common data types include:

1. Control Structure
   1. Explain the use of conditional statements and loops in Python.

* Loops and conditional statements are powerful constructs that allow programmers to automate repetitive tasks and control the flow of their programs based on certain conditions. In this article, we explored for loops, while loops, if statements, and if-else statements in Python with examples.
  1. Provide examples of an `if-else` statement and a `for` loop.
* The following example displays Number is positive if the value of number is greater than or equal to 0 . If the value of number is less than 0 , it displays Number is negative . if (number >= 0) printf("Number is positive\n"); else printf("Number is negative\n");
* age = 18 if age >= 18: print('You are eligible to vote. ') else: print('You are not eligible to vote. ')

1. Functions in Python
   1. What are functions in Python

* are essential building blocks that allow us to organize and reuse code efficiently. Functions provide a way to encapsulate a set of instructions, perform specific tasks, and return results.
  1. Why are they useful?
* to bundle a set of instructions that you want to use repeatedly or that, because of their complexity, are better self-contained in a sub-program and called when needed.
  1. Write a Python function that takes two arguments and returns their sum. Include an example of how to call this function.
* def add\_two\_num(a,b):
* sum=a+b;
* num1=int(input("Input the first number : "))
* num2=int(input("Input the second number :"))

1. Lists and Dictionaries:
   1. Describe the differences between lists and dictionaries in Python.

|  |  |
| --- | --- |
| Lists | Dictionaries |
| * Is an ordered collection of items | * Is an unordered data collection in a key |
| * Is a built-in data structure in Python that represents a collection of Unique elements | * Is a collection of key-value pairs, where each key is unique and associated with a value |
| * Is an array | * Is a hash |

* 1. Write a script that create a list of numbers and a dictionary with some key-value pairs, then demonstrates basic operations on both.
* Initialize an empty list res.
* Iterate over the items in the test\_dict dictionary using a for loop.
* For each item, create a list that contains the key of the item followed by the value of the item.
* Append the list created in step 3 to the res list.
* Print the res list.

1. Exception Handling:
   1. What is exception handling in Python?

* Is a mechanism that separates code that detects and handles exceptional circumstances from the rest of your program.
  1. How to handle an exception using try except block in Python?
* The finally clause will always execute, whether there is an error or not. x = 0 try: print(5 / x) except ZeroDivisionError: print("I am the except clause!") finally: print("I am the finally clause!") print("I am executing after the try clause!") # I am the except clause! # I am the finally clause!

1. Modules and Packages:
   1. Explain the concepts of modules and packages in Python.

* A module is a single file containing Python code, whereas a package is a collection of modules that are organized in a directory hierarchy.
  1. How can you import and use a module in your script?
* You need to use the import keyword along with the desired module name. When interpreter comes across an import statement, it imports the module to your current program. You can use the functions inside a module by using a dot(.) operator along with the module name.
  1. Provide an example using the `math` module.
* math is a built-in module in the Python 3 standard library that provides standard mathematical constants and functions.

1. Files I/O:
   * 1. How do you read from and write to files in Python?

* Read Only ('r’): This mode opens the text files for reading only. The start of the file is where the handle is located. It raises the I/O error if the file does not exist. This is the default mode for opening files as well.
* Read and Write ('r+’): This method opens the file for both reading and writing. The start of the file is where the handle is located. If the file does not exist, an I/O error gets raised.
* Write Only ('w’): This mode opens the file for writing only. The data in existing files are modified and overwritten. The start of the file is where the handle is located. If the file does not already exist in the folder, a new one gets created.
* Write and Read ('w+’): This mode opens the file for both reading and writing. The text is overwritten and deleted from an existing file. The start of the file is where the handle is located.
* Append Only ('a’): This mode allows the file to be opened for writing. If the file doesn't yet exist, a new one gets created. The handle is set at the end of the file. The newly written data will be added at the end, following the previously written data.
* Append and Read (‘a+’): Using this method, you can read and write in the file. If the file doesn't already exist, one gets created. The handle is set at the end of the file. The newly written text will be added at the end, following the previously written data.
  + 1. How to read and print the contents of a file in Python?
* we employ the built-in open() function, followed by the read() method. In this instance, 'example. txt' is the name of the file we wish to read, and 'r' is the mode in which we open the file. The 'r' mode signifies 'read', indicating that we're opening the file intending to read its content.