**Introduction to Python**

Python is general purpose, high level and interpreted and dynamically typed programming language. Which is used to implemented any kind of programs.

* Python is used in web development, data analysis, artificial intelligence, automation.
* Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

**Why learn Python**

* Easy to read & write
* Easy to understand
* Less in syntax
* Time consummations is less
* Python works on different platforms

**Interpreted**

Line by line converts into machine understandable and executes

**Compiler**

A block of code will be executed at a time

**General purpose**

Python can be used any where in software development

**High level programming language**

The language which is human readable

**Low level programming language**

Low level programming language is directly interact with the computers.

**Python Comments**

Comments in python are used to explain code and improve readability. They are ignored by the interpreter

1. **Single line comment**

Use the # symbol to write a comment on a single line

1. **Multi-line comments**

Triple quotes(“”” or ‘’’ ‘’’)can be used for multi-line coments.

**Variables in Python**In Python, Variables are used to store data values. Python does not require you to declare a variable type. It determines the type based on assigned values.EX: **Declaring Variables**x = 10 # Integery = 3.14 # Floatname = “Madhuri" # Stringactive= True #Boolean

**Variable Naming Rules**Must start with a letter (A-Z or a-z) or an underscore\_Cannot start with a numberCan contain letters, numbers, and underscores (\_)Cannot use Python keywords (e.g., if, else, while, def)**EX:**valid\_name = “Hello” \_invalid = 5 # Starts with an underscore 9number = 100 # (cannot start with a number)**Assigning Multiple Variables**

Python allows multiple variables to be assigned at once.EX:a, b, c = 1, 2, 3print (a, b, c) #Output: 1,2,3Can also assign the same value to multiple variables**EX:**x = y = z = "Same Value"**Variable Types**Python variables can hold different types of data.EX:Num = 10 # Integer pi = 3.14 # Float text = "Python” # Stringflag = True # Booleanitems = [1, 2, 3] # List**Type ()**: To check the data type, use type( )print (type (Num)) # Output: <class 'int’>print(type(pi)) # Output: <class 'float'>**Changing Variable Type**Changing one type of data into another type of data dynamically is know as Type Casting.**EX:**x = 5 # Integer x = str(x) # Now it's a string y = int("10") # Converts string "10" to integer 10 z = float (3) # Converts integer 3 to float 3.0**Global and Local Variables**Two types of Variables**Local Variables**The variable which can be declared inside the function is known as local variableEX:def greet (): message = “Hello, World!” # Local variable print(message)greet () # Output: Hello, World! # print(message) # This would cause an error because 'message' is not accessible outside the function.

**Global Variables**The variable which can be declared outside of the function is known as global variable**EX:**greeting = "Hello, World!" def say\_hello(): print(greeting) # Accessing the global variablesay\_hello() # Output: Hello, World!print(greeting) # Output: Hello, World!

* Another way to declare a global variable inside a function by using **global keyword.**

EX:count = 0 def increment(): global count # Declare 'count' as a global variable count=count+1increment() print(count) # Output: 1**Data Types**

Python has several build-in data types.

**1. Numeric Types**

* **int** → Integer (e.g., 10, -5, 1000)
* **float** → Floating-point number (e.g., 10.5, -3.14, 2.0)
* **complex** → Complex numbers (e.g., 3+4j, 2-1j)

**2. Sequence Types**

* **list** → Ordered, mutable collection (e.g., [1, 2, 3])
* **tuple** → Ordered, immutable collection (e.g., (1, 2, 3))
* **range** → Sequence of numbers (e.g., range(1, 10))

**3. Text Type**

* **str** → String (e.g., "hello", 'Python')

**4. Set Types**

* **set** → Unordered, mutable collection of unique elements (e.g., {1, 2, 3})
* **frozen set** → Immutable set (e.g., frozen set ({1, 2, 3}))

**5. Mapping Type**

* **dict** → Key-value pairs (e.g., {"name": "John", "age": 30})

**6. Boolean Type**

* **bool** → Represents True or False (e.g., True, False)

**7. Binary Types**

* **bytes** → Immutable sequence of bytes (e.g., b"hello")
* **bytearray** → Mutable sequence of bytes
* **memoryview** → View of memory buffer

**8. None Type**

* **NoneType** → Represents a null value (None)

**Operators in Python**

Operators in Python are special symbols that perform operations on variables and values. Python has several types of operators

Python divides the operators in the following groups:

* Arithmetic operators
* Assignment operators
* Comparison operators
* Logical operators
* Identity operators
* Membership operators
* Bitwise operators

1. **Arithmetic operators**

Arithmetic operators are used with numeric values to perform common mathematical operations

Addition ------> a=10, b=3--------->a+b=13

Subtraction-------->a-b=7

Multiplication--------->a\*b=30

Division------------>a/b=3.333

Floor Division------------>a//b=3

Modulus--------------->a%b=1

Exponentiation---------->a\*\*b=1000

1. **Assignment operators**

Assignment operators are used to assign values to variables

Assign (=)

Add & Assign (+=)

Subtract & Assign (-=)

Multiply & Assign (\*=)

Divide & Assign (/=)

Floor Divide & Assign (//=)

Modulus & Assign (%=)

Exponentiate & Assign (\*\*=)

1. **Comparison operators**

Comparison operators are used to compare two values. It returns True or False

EX: a=10, b=3

Equal to (==) -------->a==b---------->False

Not equal to (! =) ----------> a! = b----------->True

Greater than (a>b) ---------->True

Less than (a<b) ------------> False

Greater than or equal to (a >=b) ---------->True

Less than or Equal to (a<=b) -------------->False

1. **Logical operators**

Logical operators are used to combine conditional statements

And---->Return True if both conditions are True

Or------->Return True if at least one condition is True

Not-------->Reverse the boolean values

1. **Identity operators**

Identity operators are used to compare the objects, not if they are equal, but if they are actually the same object, with the same memory location

is ------->Returns True if both variables refer to the same object

is not ------>Returns True if variable refer to different objects

1. **Membership operators**

Membership operators are used to test if a sequence is presented in an object

in -------->Returns True if a value exists in the sequence

not in ------->Return True if a value does not exist in the sequence

1. **Bitwise operators**

Membership operators are used to test if a sequence is presented in an object

AND (&)

‘(‘) ----->OR

XOR (^)

NOT (bitwise complement) (~)

Left Shift (<<)

Right Shift (>>)

**Conditional Statements in Python**

To check the conditions based on the requirements. Python uses if, elif, and else

**1. if Statement**

Executes a block of code **only if** a condition is True.

EX:

x = 10

if x > 5:

print("x is greater than 5")

**2. if-else Statement**

Executes one block if the condition is True, otherwise executes the else block.

EX:

x = 3

if x > 5:

print("x is greater than 5")

else:

print("x is not greater than 5")

**3. if-elif-else Statement**

Checks multiple conditions in sequence.

EX:

x = 0

if x > 0:

print("Positive number")

elif x < 0:

print("Negative number")

else:

print("Zero")

**4. Nested if Statements**

if statements inside another if statement.

EX:

x = 10

if x > 0:

print("Positive number")

if x % 2 == 0:

print("Even number")

**Conditional Statements**Conditional Statements are used to execute code based on a condition.**Break**The break statement is used to exit a loop immediately when a condition is met.EX:for Num in range(1, 6): if Num == 3: break # Stops the loop when Num is 3 print(Num)**Continue**The continue statement skip the current iteration and moves to next iteration.EX:for Num in range(1, 6): if Num == 3: continue # Skips printing when Num is 3 print(Num)