WELCOME

Python Series

AGENDA

Python Variables
Creating Variables
Printing Variables
Deleting Variables
Few More Points on Variables
Naming Conventions
Variable Naming Pattern

PYTHON VARIABLES

- Python variables are the reserved memory locations used to store values.
- This means creating a variable reserves space in memory.
- The Python interpreter allocates memory based on a variable's data type.
- Data types determine what can be stored in the reserved memory.
- Variables can store various data types, including integers, decimals, and characters.

PYTHON VARIABLES

- Data items of different types are stored in computer memory, each having a unique address.
- Memory addresses are internally represented in binary form, and data is stored as binary because computers operate on binary principles.
- Assembly language can be used to convert data items and memory addresses into machine language instructions, but this is complex for most people.
- Accessing data directly using its memory ID is impractical; high-level languages like Python allow assigning an alias or label to memory locations for easier reference.
- For example, the string "Monday" is labeled as day and the number "2024" is labeled as year using the assignment operator (=) to link objects with labels.
- Python's **id()** function can return the memory address (ID) where an object is stored.

PYTHON VARIABLES

1000	1001	2024	1003
1004	1005	1006	1007
1008	MON	1010	1011
1012	1013	1014	1015

CREATING VARIABLES

- **Automatic Variable Creation**: Python variables do not require explicit declaration to reserve memory space. A variable in Python is created the moment you assign a value to it.

- **Assignment Operator (=):** The equal sign **('=')** is used to assign values to variables. The operand to the left of the **'='** operator is the variable name, and the operand to the right is the value to be stored in that variable.

- **Example of Creating Variables:**
  - Integer Variable: **age = 30**
  - Float Variable: **height = 5.9**
  - String Variable: **name = "Mogli"**

PRINTING VARIABLES

- **Printing Variables**: After creating variables and assigning values to them, you can print their values using the **'print()'** function.

- Example of Printing Variables:
  - To print the integer variable: **print(age)**
  - To print the float variable: **print(height)**
  - To print the string variable: **print(name)**

DELETING VARIABLES

- The **del** statement in Python is used to delete references to objects in memory.
- **Syntax**: del variable_name, where variable_name is the variable you want to delete.
- **Syntax**: del variable_name1, variable_name2 , for multiple deletion
- **Example of Deleting Single Variables:**
  - del age
- **Example of Deleting Multiple Variables:**
  - del height, name
- **Error on Accessing Deleted Variables**:
  - NameError: name 'B' is not defined

FEW MORE POINTS ON VARIABLES

- Use **type()** to get the type of the variable.
  - **Syntax**: type(variable_name)
  - **Output**: < class 'str' | 'int' | 'float'> depending on variable

- Python Variables are **case-sensitive**. It means that age, Age, AGE, aGe are all different variables.

- For string variables, we declare it using either single quote (' ') or double quote (" "). Using backticks (` `) will throw syntax error.

- **Multiple Variable Assignment:**
  - Syntax 1: variable_name1, variable_name2 = value1, value2
  - Example 1: a, b, c = 10, 'banana', 30
  - Syntax 2: variable_name1 = variable_name2 = value
  - Example 2: a = b = c = 10,

- **Casting Variable:** We can specify the data type of a variable.
  - Example:
    - X = str(10)
    - y = int(10)
    - Z = float(10)

Python Series

NAMING CONVENTION (RULES)

- **Start with Letter or Underscore**: A variable name must begin with a letter (either uppercase or lowercase) or an underscore (_).

- **No Numbers or Special Characters at the Start**: A variable name cannot start with a number or any special character (such as $, (, *, %, etc.).

- **Alpha-numeric and Underscores Only**: A variable name can consist of alpha-numeric characters (A-Z, a-z, 0-9) and underscores (_). No other characters are allowed.

- **Case Sensitivity**: Variable names are case-sensitive in Python. This means that Name, NAME, and name would be treated as three distinct variables.

- **Avoid Reserved Keywords**: Reserved keywords in Python cannot be used as variable names. These keywords include words like if, for, while, class, return, global, etc., as they have special meanings in Python syntax.

- **Unique Name:** Every variables should be unique.

NAMING PATTERNS (RULES)

- **Camel Case:** Eg: amountDues, successfulTaskExecution**.**
- **Pascal Case**: AmountDues, SuccessfulTaskExecution
- **Snake Case**: amount_dues, successful_task_execution
- **Screaming Snake Case**: TOTAL_MONTHS

UPCOMING

Character Sets
Tokens
Types of Tokens
Punctuators and Delimiters
Escape Sequence