1. Variables and Data Types

Variables are essentially **names given to memory locations in a program**. They act as containers to store data. In Python, the assignment rule dictates that the value on the **right side of the equals sign (`=`) is stored into the variable on the left side**. This `EQUAL TO` sign is called the **assignment operator**. For example, `age = 23` means the value `23` is stored in a variable named `age`.

Rules for Naming Variables (Identifiers):

- Variable names (identifiers) can contain **uppercase letters (A-Z), lowercase letters (a-z), digits (0-9), and underscores (`_`)**.

- They **cannot start with a digit**. For example, `variable1` is valid, but `1variable` is not.

- **Special symbols** like `%`, `@`, `#`, `$`, `!` are **not allowed** in variable names.

- **Keywords** (reserved words in Python) **cannot be used as identifiers**. For instance, you cannot name a variable `True` or `if`.

Best Practices for Variable Names:

- Variable names should be **simple, short, and meaningful**.

- They should convey the **purpose or content of the variable**. For example, `name`, `age`, `count`, `sum` are good names, whereas `x`, `y`, `bd` are ambiguous.

Data Types:

Python automatically detects the type of value stored in a variable. This is known as an **implicitly typed language**, meaning you don't explicitly declare the data type when defining a variable, unlike languages like Java or C++.

Python supports several built-in data types:

- **Integer (`int`)**: Represents whole numbers (positive, negative, or zero).

- **String (`str`)**: Represents sequences of characters.

- **Float (`float`)**: Represents decimal numbers.

- **Boolean (`bool`)**: Represents truth values (`True`, `False`).

- **None (`NoneType`)**: Represents the absence of a value.

Keywords:

Reserved words in Python that **cannot be used as identifiers**. Examples: `False`, `None`, `True`, `and`, `or`, `if`, `elif`, `else`.

---

2. Strings

A **string** is a **sequence of characters**. It's a built-in data type that stores a collection of characters.

Creating Strings:

- Strings can be defined using `'Hello'`, `"World"`, or triple quotes for multi-line strings.

Escape Sequence Characters:

- `\n`: New line
- `\t`: Tab space

String Operations:

- Concatenation (`+`)
- Length (`len()`)
- Indexing
- Immutability
- Slicing

String Methods:

- `string.endswith("substring")`
- `string.capitalize()`
- `string.replace("old", "new")`
- `string.find("word")`
- `string.count("substring")`

---

3. Conditional Statements

Keywords:

- `if`
- `elif`
- `else`

Syntax and Usage:

Example:

if condition:

statement

elif another_condition:

statement

else:

statement

Logical Operators:

- `and`, `or`, `not`

Other Concepts:
- Indentation is mandatory
- Nesting conditionals is allowed
- Ternary operator: `x if condition else y`

---

4. Lists

A **list** is a **built-in data type that stores a set of values**.

Characteristics:
- Ordered
- Mutable
- Can store multiple data types

List Methods:
- `append()`
- `sort()`
- `reverse()`
- `insert()`
- `remove()`
- `pop()`
- `copy()`

---

5. Tuples

A **tuple** is an ordered collection of values.

Characteristics:
- Ordered
- Immutable
- Can store multiple data types

Tuple Methods:
- `index()`
- `count()`