Variables and Data Types in Python

1. What is a Variable in Python?

A **variable** is like a **container** used to store data. You can assign a value to a variable and use it later in your program.

Syntax:

```
variable_name = value
```

Example:

```
name = "Raj"

age = 24

is_student = True
```

No need to declare the data type — Python figures it out **automatically**.

---

2. Rules for Naming Variables

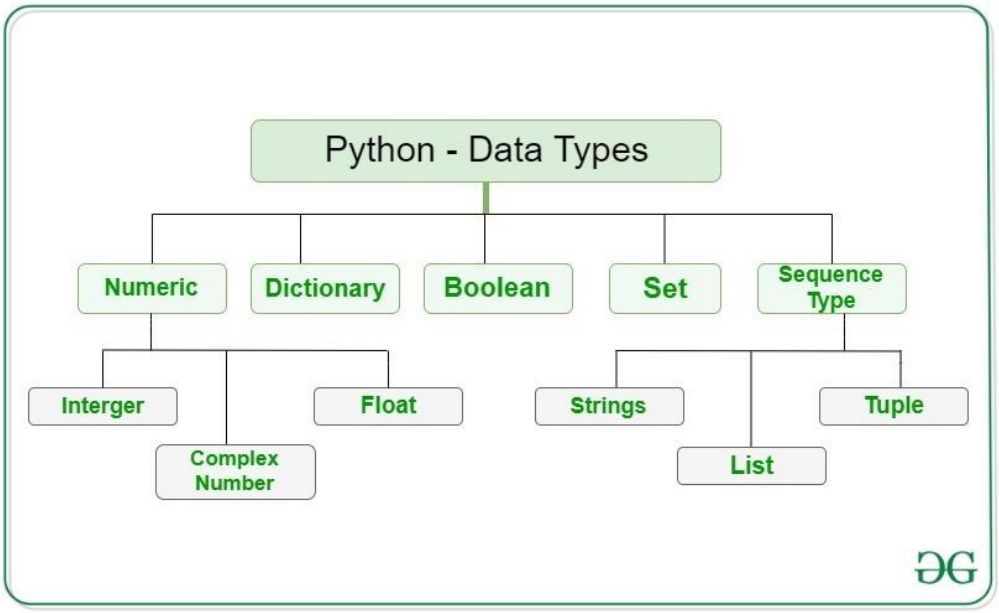
- Must begin with a **letter** (a–z, A–Z) or an **underscore (_)**
- Cannot begin with a number
- Cannot use Python **keywords** like `if`, `for`, `class`, etc.
- Cannot contain any special characters.
- Variable names are **case-sensitive** (`Age` and `age` are different)

```
name = 'raj'              # valid
_id = 10                  # valid
user1 = "hello"           # valid
_name_2 = "python"        # valid
Name = "Raj"              # valid (case-sensitive, different from 'name'
username3 = "test"        # valid

1name = "abc"             # invalid (starts with a number)
class = "BCA"             # invalid (reserved keyword)
total$value = 500         # invalid (contains special character '$')
data@ = 22                # invalid (contains special character '@')
user-name = "raj"         # invalid (contains special character '-')
var# = 90                 # invalid (ends with special character)
user_ = "final"           # valid (ends with underscore is allowed)
_data* = "error"          # invalid (ends with special character '*')
```

3. Python Data Types

Python has many built-in data types.



1 Numeric

Used for numeric calculations and values. It includes:

- **Integer (int)**: Whole numbers
  Example: a = 10
- **Float (float)**: Numbers with decimals
  Example: b = 3.14
- **Complex Number (complex)**: Used in scientific calculations
  Example: c = 2 + 3j

---

2 Dictionary

- A collection of **key-value pairs**.
- Keys must be unique and immutable (e.g., strings, numbers).

```
student = {
   "name": "Raj",
   "age": 24,
   "gpa": 3.8
}
```

Access: student["name"] → 'Raj'

---

3 Boolean

- Only two values: True or False
- Used for **decision-making** and **conditions**

```
is_active = True
```

```
has_passed = False
```

You can also get Boolean results from comparisons:

```
x = 5 > 3  # True
```

---

4 Set

- **Unordered** collection of **unique elements**
- Mutable (you can add or remove items)

```
unique_nums = {1, 2, 3}
```

- You can perform set operations like union, intersection, difference and systematic_difference.

---

5 Sequence Type

These types hold multiple items in a **sequence**. Includes:

String (str)

- Text enclosed in quotes.

```
name = "Python"
```

List

- Ordered, mutable (changeable)

```
fruits = ["apple", "banana", "mango"]
```

Tuple

- Ordered but **immutable** (cannot change)

```
dimensions = (20, 40)
```

None Type

Used to define a **null value** or **no value at all**.

```
x = None
```

---

4. Check the Type of a Variable

Use the `type()` function to find the data type:

```
x = 10
print(type(x))  # <class 'int'>

y = "Hello"
print(type(y))  # <class 'str'>
```