🐍Python Variables & Data Types (Deep Dive)

---

1. What is a Variable?

A variable is simply a **name** that is used to store data in memory. Think of it like a **label on a container** – the label (variable name) points to the data inside.

- Python variables are **dynamically typed** → You don't need to declare the type explicitly.
- Variables are just **references** (they point to objects in memory).

Example:

```
x = 10      # int
x = "Hello" # now string
```

---

2. Rules for Naming Variables

✅Allowed: - Letters (a–z, A–Z), digits (0–9), underscore (`_`) - Must start with a letter or `_`

❌Not Allowed: - Special characters (`@, $, %, #`) - Keywords (`if`, `while`, `class`, etc.)

Example:

```
name = "Alice"   # valid
_name = "Bob"    # valid
2name = "Eve"    # ❌ invalid
```

---

3. Variable Assignment

- **Single assignment**:

```
a = 5
b = "Python"
```

- **Multiple assignment**:

```
x, y, z = 10, 20, 30
```

- **Same value to multiple variables**:

```
x = y = z = 100
```

- **Unpacking**:

```
numbers = [1, 2, 3]
a, b, c = numbers  # a=1, b=2, c=3
```

---

4. Memory & Variables

- Variables are just **names pointing to objects in memory**.
- Example:

```
a = 10
b = a
print(id(a), id(b))   # same memory id (points to same object)
```

- Python uses **reference counting + garbage collector** to manage memory.

---

5. Data Types in Python

Everything in Python is an **object**.

Numeric Types

- **int** → integers (unlimited size)
- **float** → decimal numbers
- **complex** → real + imaginary part

Boolean (`bool`)

- Only two values: `True`, `False`
- Internally `True = 1`, `False = 0`

String (`str`)

- Sequence of characters inside `' '` or `" "`

Collection / Sequence Types

- **List** (ordered, mutable)
- **Tuple** (ordered, immutable)
- **Set** (unordered, unique values)
- **Dictionary** (key-value pairs, mutable)

None Type

Represents **no value** / null equivalent.

---

6. Type Casting

Converting one data type to another.

- **Implicit Casting (automatic)**:

```
x = 5      # int
y = 2.5    # float
z = x + y  # z becomes float (7.5)
```

- **Explicit Casting (manual)**:

```
a = int("10")
b = float("3.14")
c = str(100)
```

---

7. Check Data Type

- Using `type()`:

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

- Using `isinstance()`:

```
x = 10
print(isinstance(x, int))  # True
```

---

8. Mutable vs Immutable Types

- **Mutable** → can be changed (list, dict, set)
- **Immutable** → cannot be changed (int, float, bool, string, tuple)

Example:

```
s = "hello"
# s[0] = "H" ❌ error (string immutable)

lst = [1, 2, 3]
lst[0] = 10  # ✅ allowed
```

---

9. Python is Dynamically Typed

- You don't declare data type while creating a variable.
- But Python is **strongly typed** → won't allow illegal operations.

Example:

```
x = "10"
y = 5
print(x + y)  # ❌ Error (cannot add str and int)
```