Variables in Python

What is a Variable?

A **variable** in Python is like a container used to store information (data) that can be referenced and manipulated later in your program.
It helps you label data with a descriptive name so your programs can be understood more clearly.

Example:

```
name = "Amit"
age = 25
is_student = True
```

Rules for Naming Variables

1. Variable names can contain letters, numbers, and underscores.
2. They must begin with a letter or underscore (_), not a number.
3. Python is **case-sensitive** (`name` and `Name` are different).
4. You **cannot** use reserved Python keywords (like `if`, `while`, `for`, etc.) as variable names.

❌ Invalid variable names:

```
1name = "Raj"       # starts with number
for = 10            # uses a reserved keyword
my-name = "Ajay"    # hyphen is not allowed
```

✅ Valid variable names:

```
my_name = "Priya"
_age = 23
total3 = 450
```

```
1 name = 'Raj'
2 print(name)
3 Name = 'shyam'
4 print(Name)
5 print(id(name))
6 print(id(Name))
```

Naming Conventions (Best Practices)

1. Use lowercase letters and underscores for variable names.
   Example: `student_name`, `total_marks`
2. For multi-word variable names, you can follow:
   - **snake_case**: `first_name` (most common in Python)
   - **camelCase**: `firstName`
   - **PascalCase**: `FirstName` (usually for class names)

Note: Stick with `snake_case` for variable names in Python.

Real-life Example

Think of a variable as a labelled container in your kitchen.

```
sugar_kg = 2
rice_kg = 5
```

Here, `sugar_kg` and `rice_kg` are containers (variables) holding quantity of items.

🧠 Practice Exercise: Try it Yourself!

1. Create variables for your name, age, and city.

2. Swap two variables without using a temporary variable.
3. Try dynamic typing: assign a string to a variable, then assign an integer.
4. Use `type()` to print the type of your variables.
5. Use valid and invalid variable names and observe what errors appear.