Variables in Python

What are Variables?

Variables in Python are names given to memory locations to store data. Essentially, a variable acts as a label or a tag that points to an object (a piece of data) in your computer's memory. When you create a variable, you're not putting data *into* the variable itself, but rather making the variable *point to* that data. This allows you to refer to and manipulate data using a simple, descriptive name instead of directly dealing with complex memory addresses. Python is dynamically typed, meaning you don't need to explicitly declare the type of data a variable will hold; it infers the type at the time of assignment.

Visualization

Imagine a name tag or a sticky note. On the sticky note, you write a name (e.g., 'user_age'). You then stick this note onto a specific item, like a box containing the number 30. The sticky note ('user_age') now points to the box with '30' inside. If you later want to change the value, you simply move the 'user_age' sticky note to a new box containing, say, '31'. The sticky note itself doesn't contain the number; it just indicates where to find it.

Code Example

```
# --- Variable Assignment --- # 1. Assigning an integer value my_integer = 10
print(f"my_integer: {my_integer}") # Output: my_integer: 10 # 2. Assigning a string value
my_string = "Hello Python!" print(f"my_string: {my_string}") # Output: my_string: Hello Python!
# 3. Assigning a float value my_float = 3.14 print(f"my_float: {my_float}") # Output: my_float:
3.14 # 4. Assigning a boolean value is_active = True print(f"is_active: {is_active}") # Output:
is_active: True # --- Reassignment --- # Variables can be reassigned to new values my_integer =
20 # 'my_integer' now points to a new integer object print(f"my_integer after reassignment:
{my_integer}") # Output: my_integer after reassignment: 20 # Variables can even be reassigned to
values of a different type (dynamic typing) my_string = 12345 # 'my_string' now points to an
integer object print(f"my_string after type reassignment: {my_string}") # Output: my_string
after type reassignment: 12345 # --- Using Variables --- # Variables can be used in expressions
and operations num1 = 5 num2 = 7 sum_result = num1 + num2 print(f"Sum of num1 and num2:
{sum_result}") # Output: Sum of num1 and num2: 12 greeting = "Hi" name = "Alice" full_message =
greeting + ", " + name + "!" print(f"Full message: {full_message}") # Output: Full message: Hi,
Alice!
```

Real-World Analogy

Think of variables like labels on storage containers. You have many containers (data) in a warehouse (computer memory), and each contains some item (a value like a number, text, etc.). Instead of remembering the exact location of a container, you put a unique label (the variable name, e.g., 'apples_count') on the container. When you need to access the apples, you just look for the 'apples_count' label. If you get new apples and put them in a different container, you simply move the 'apples_count' label to the new container. The label itself isn't the apples; it just tells you *where* to find them.