**Topic 2 - Variables for Strings**

**Introduction**  
In Python, variables allow us to store information that we can reference and manipulate throughout our program. This lesson will focus on using variables specifically for storing text values, or *strings*, and understanding how to work with and change those values.

**1. What are Variables for Strings in Python?**

A *variable* in Python acts as a label or reference to a particular piece of data. When we assign a value to a variable, we’re creating a way to refer to that value without repeating it.

For example, let’s say:

* My name is Vishak.
* My nationality is Indian.

We can create variables for these facts so that we don’t need to repeat them explicitly. If I say, "Do you know my name?" you’ll know I’m referring to "Vishak." Similarly, "Is your nationality the same as mine?" refers to "Indian" without needing to mention it directly.

In Python, we assign values to variables in this way:

*name = "Vishak"*

*nationality = "Indian"*

The variables name and nationality now refer to the strings "Vishak" and "Indian", respectively. When Python encounters name, it understands that it refers to "Vishak".

**2. Why Use Variables for Strings?**

Using variables for strings has several advantages:

* **Code Reusability**: Variables allow us to reference the same value multiple times without retyping it.
* **Readability**: Descriptive variable names make it easier to understand what the code is doing.
* **Flexibility**: The value of a variable can be updated easily, making it adaptable if information changes.

**3. How to Create and Use Variables for Strings in Python**

**Basic Usage**  
When assigning a string to a variable, type the variable name, an equal sign =, and the string in quotes.

*name = "Vishak" # Assigns the value "Vishak" to the variable `name`*

*print(name) # Output: Vishak*

**Updating Variable Values**  
The value a variable refers to can change. For example, if I change my name to Rohan, we can reassign the variable name:

*name = "Rohan"*

*print(name) # Output: Rohan*

**Examples of Using Variables for Strings**

**1. Assigning and Printing Variables**

*name = "Vishak"*

*nationality = "Indian"*

*print(name) # Output: Vishak*

*print(nationality) # Output: Indian*

**2. Updating Variable Values**

*name = "Vishak"*

*print(name) # Output: Vishak*

# Changing the value of name

*name = "Rohan"*

*print(name) # Output: Rohan*

**3. Using Variables in print() Statements**

Instead of writing a string directly in print(), we can assign it to a variable and then print the variable.

*thank\_you = "Thank you for your help!"*

*print(thank\_you) # Output: Thank you for your help!*

**4. Example of a Variable Name without Quotes**

When creating a variable, **do not enclose its name in quotes**. Quotes are for strings only.

*name = "Vishak" # Correct*

*"country" = "India" # Incorrect*

**5. No Spaces in Variable Names**  
Variable names cannot contain spaces, so instead of country of origin, use country\_of\_origin or countryOfOrigin for readability.

**Naming Rules for Variables**

1. **No Quotes Around Variable Names**
   * Correct: name = "Vishak"
   * Incorrect: "name" = "Vishak"
2. **No Spaces in Variable Names**
   * Correct: country\_of\_origin = "India"
   * Incorrect: country of origin = "India"
3. **Choose Descriptive Variable Names**
   * It’s helpful to use names like first\_name or home\_city rather than random letters, which make the code hard to understand.

**Example Code with Multiple Variables**

first\_name = "Rahul"

city\_of\_origin = "Mumbai"

greeting = "Namaste"

print(first\_name) # Output: Rahul

print(city\_of\_origin) # Output: Mumbai

print(greeting) # Output: Namaste

**Conclusion**

Python variables provide a powerful way to store and reuse information, especially with strings. Understanding variables makes code more efficient, adaptable, and readable, which is essential for all Python developers.