String Formatting (f-string)

Problem Statement

Python's String and Number Combination Issue:

- Python only knows how to combine strings with other strings.
- It doesn’t know how to combine a string and a number directly.

Example of the Problem:

name = "Alice"

age = 30

print("My name is " + name + " and I am " + age + " years old.")

Error - This code throws an error: Type. Error: can only concatenate str (not "int") to str.

Solution Using str() for Conversion:

- To tell Python to treat age as a string, we use str(age), so that Python can combine two strings: "My name is " and "30".

Example - print("My name is " + name + " and I am " + str(age) + " years old.")

Output : My name is Alice and I am 30 years old.

This solves the problem by converting the number into a string using str().

Challenge with Code Management Using Concatenation:

As the string grows more complex, it becomes difficult to manage code with many variables, e.g number1

= 5 number2 = 10

print("The sum of " + str(number1) + " and " + str(number2) + " is " + str(number1 + number2) + ".")

In this example, manually converting each variable with str() and concatenating with + makes the code harder to read and maintain.

2. **Solution Using f-strings:** **# This is best approach for to concanate string and number**

- f-strings allow you to directly perform calculations (e.g., number1 + number2) inside the curly braces {} without worrying about converting them to strings.
- Python automatically converts non-string variables to strings when using f-strings. The f-string simplifies the code:

print(f"The sum of {number1} and {number2} is {number1 + number2}.")

Example - print(f“My Name is {name} and my age is {age}”)

Output – My Name is Alice and my age is 30

Comparison:

Without f-strings: You need to manually concatenate strings using + and use str() to convert non-string types. This can make the code harder to read and manage.

With f-strings: You can directly embed variables and expressions inside the string using {}. F-strings are more concise, readable, and Python handles type conversion automatically.