Chapter 7: Strings in Python

🎯 Learning Objectives:

By the end of this chapter, learners will be able to:

- Understand what strings are and how they are used in Python.
- Perform basic string operations like indexing and slicing.
- Use built-in string methods.
- Format strings effectively.
- Handle multiline and raw strings.

🧠 7.1 What is a String?

A **string** is a sequence of characters enclosed in single (`'`), double (`"`), or triple (`''' ` or `"""`) quotes.

✅ Examples:

```
name = "Alice"

greeting = 'Hello'

multiline = """This is

a multi-line string"""
```

🔍 7.2 String Indexing and Slicing

Python strings are **indexed**, starting from `0`.

✅ Indexing:

```
text = "Python"

print(text[0])  # Output: P

print(text[-1]) # Output: n
```

✅ Slicing:

```
text = "Programming"

print(text[0:6])    # Output: Progra

print(text[:4])     # Output: Prog

print(text[4:])     # Output: ramming
```

🔁 7.3 Common String Operations

Operation	Syntax / Example	Output
Concatenation	`"Hello" + "World"`	`'HelloWorld'`
Repetition	`"Hi" * 3`	`'HiHiHi'`
Length	`len("Python")`	`6`

Membership	`'a' in "apple"`	`True`

🧰 7.4 Useful String Methods

Method	Description	Example
`lower()`	Converts to lowercase	`"Hello".lower()` → `'hello'`
`upper()`	Converts to uppercase	`"hello".upper()` → `'HELLO'`
`strip()`	Removes leading/trailing spaces	`" Hello ".strip()` → `'Hello'`
`replace(a, b)`	Replaces `a` with `b`	`"Hi Sam".replace("Sam", "Tom")`
`split()`	Splits into list by space (default)	`"a,b,c".split(",")` → `['a','b','c']`
`join()`	Joins list into string	`" ".join(['I','love','Python'])`

`find()`	Returns index of first match	`"hello".find('e')` → `1`

🗒️ 7.5 String Formatting

🔹 **Using `f-strings` (Recommended – Python 3.6+)**

```
name = "Alice"

age = 25

print(f"My name is {name} and I am {age} years old.")
```

🔹 **Using `.format()`**

```
print("My name is {} and I am {} years old.".format("Bob", 30))
```

📜 7.6 Multiline and Raw Strings

🔹 **Multiline Strings**

```
note = """This is

a multiline

string."""
```

🔹 **Raw Strings (ignores escape sequences)**

```
path = r"C:\Users\Name"

print(path)  # Output: C:\Users\Name
```

🧪 7.7 Try It Yourself

1. Create a string and print the first, middle, and last characters.
2. Write a program that takes a user's name and prints a greeting using f-string.
3. Replace all spaces in `"Python is easy"` with `-`.
