# Chapter 3 – Strings in Python

## 1️⃣ What is a String?

A **string** is a data type in Python that stores a sequence of characters — letters, numbers, or symbols — enclosed in **single (' ')**, **double (" ")**, or **triple (''' ''')** quotes.

**Examples:**

```
str1 = 'Hello'
str2 = "Saumya Singh"
str3 = '''Welcome to Python!'''
```

🧠 **Note:** Strings are **immutable**, meaning once created, their content cannot be changed directly.

## 2️⃣ Creating Strings

You can create strings in different ways:

```
name = "Samosa"
greet = 'Hello'
msg = """Python is fun!"""
```

### ✅ String Concatenation:

```
print("Hello " + "World")   # Output: Hello World
```

### ✅ Length of String:

```
len("GulabJamun")   # Output: 10
```

## 3️⃣ Indexing

Each character in a string has a **position (index)** starting from **0**.

```
str = "SaumyaSingh"
```

```
Index: 0 1 2 3 4 5 6 7 8 9 10
Chars: S a u m y a S i n g h
```

**Examples:**

```
str = "Samosa"
print(str[0])    # S
print(str[3])    # o
```

❌ Strings are immutable:

```
str[0] = 'B'   # Error: Strings cannot be changed directly
```

## Practice Question 1

Write a Python program that takes a user’s name as input and prints:

1. The first character
2. The last character
3. The total length of the name

## 4️⃣ Slicing

Slicing lets you access a **part of a string**.

**Syntax:**

```
string[start : end]   # end index is excluded
```

**Examples:**

```
str = "GulabJamun"
print(str[0:5])   # Gulab
print(str[:6])    # GulabJ
print(str[5:])    # Jamun
```

**Negative Indexing**

```
 G  u  l  a  b  J  a  m  u  n
-10 -9 -8 -7 -6 -5 -4 -3 -2 -1

str = "GulabJamun"
print(str[-5:-1])   # Jamu
```

## Practice Question 2

Write a program that takes your favorite food name as input and prints:

- The middle 3 characters
- The last 2 characters

## 5️⃣ Common String Methods

| Method | Description | Example |
|---|---|---|
| `.upper()` | Converts all characters to uppercase | `"samosa".upper()` → `'SAMOSA'` |
| `.lower()` | Converts all characters to lowercase | `"Saumya".lower()` → `'saumya'` |
| `.title()` | Capitalizes the first letter of each word | `"hello world".title()` → `'Hello World'` |
| `.find(sub)` | Returns index of first occurrence | `"banana".find("na")` → `2` |
| `.replace(old, new)` | Replaces all occurrences | `"Python is cool".replace("cool", "fun")` → `'Python is fun'` |
| `.count(sub)` | Counts occurrences | `"mango".count("a")` → `1` |
| `.endswith(suffix)` | Checks if string ends with given substring | `"coder.".endswith(".")` → `True` |

|  |  |  |
|---|---|---|
| `.capitalize()` | Capitalizes first letter only | `"python".capitalize()` → `'Python'` |

### Practice Question 3

Write a program that:

- Takes a sentence as input
- Converts it to lowercase
- Replaces all spaces `" "` with underscores `"_"`
- Prints the new string

## 6️⃣ Formatted Strings (f-Strings)

f-Strings make it easy to include variables inside strings.

**Example:**

```
name = "Saumya Singh"
age = 21
print(f"My name is {name} and I am {age} years old.")
```

Output:

```
My name is Saumya Singh and I am 21 years old.
```

## 7️⃣ Escape Sequences

Escape sequences let you use special formatting in strings.

| **Escape Sequence** | **Description** | **Example** |
|---|---|---|
| `\n` | New line | `"Hello\nWorld"` → prints on two lines |

| | | |
|---|---|---|
| `\t` | Tab space | `"A\tB"` → adds a tab between A and B |
| `\\` | Backslash | `"C:\\newfolder"` → `C:\newfolder` |
| `\'` | Single quote | `'It\'s great'` → `It's great` |
| `\"` | Double quote | `"He said \"Hi\""` → `He said "Hi"` |

## 8️⃣ Mini Project: Emoji Converter 😊

Convert text-based emotions into emojis.

**Code Example:**

```
# Emoji Converter - Basic Version (No if, no loop)

msg = input("Enter your message: ")

msg = msg.replace(":)", "😊")
msg = msg.replace(":(", "☹️")
msg = msg.replace(":D", "😃")
msg = msg.replace(";)", "😉")

print(msg)
```

**Example Run:**

```
Enter your message: Hello :) I am learning Python from Saumya :D
Output: Hello 😊 I am learning Python 😃
```

## 9️⃣ Extra String Operations

1. **Concatenation:**
   `"Hello" + "Samosa"` → `'HelloSamosa'`

2. **Repetition:**
   `"Yum! " * 3` → `'Yum! Yum! Yum! '`

3. **Membership:**
   `"a" in "banana"` → `True`
   `"z" not in "mango"` → `True`

4. **len() Function:**
   `len("SaumyaSingh")` → `11`

# Assignment Set 3 [ Chapter 3 ]

1. Write a program that takes a sentence and prints:

- Total characters (`len()`)
- Uppercase version
- Lowercase version

**Example:**

```
Input: GulabJamun is Sweet
Output:
Total characters: 19
Uppercase: GULABJAMUN IS SWEET
Lowercase: gulabjamun is sweet
```

2. Write a Python program that takes any word or sentence as input and prints:

- The **first character**
- The **last character**
- The **total number of characters**

**Example:**

```
Input:  Python
Output:
First character: P
Last character: n
Total characters: 6
```

## ✅ Summary

- Strings are **immutable** and store text.
- Use **indexing** and **slicing** to access parts of a string.
- **String methods** help modify or analyze text.
- **f-Strings** simplify variable formatting.
- **Escape sequences** let you print quotes, tabs, and new lines.
- Practice small **string programs** to master text manipulation!