◆ 1. Indexing → `s[0]`, `s[-1]`

Indexing means accessing **individual characters** of a string.

```python
s = "Python"
print(s[0])    # P (1st character)
print(s[-1])   # n (last character)
```

Use case: Get initials, last letters, or check characters.

◆ 2. Slicing → `s[1:4]`, `s[:]`, `s[::-1]`

Slicing means getting a **substring**.

```python
s = "Programming"
print(s[1:4])    # 'rog' → index 1 to 3
print(s[:])      # Full string
print(s[::-1])   # Reverse → 'gnimmargorP'
```

Use case: Extract words, reverse strings, cut text, etc.

◆ 3. String Methods

📌 `lower()`, `upper()`, `title()`

Used for changing the case.

```python
s = "mUhAmMaD"
print(s.lower())   # muhammad
print(s.upper())   # MUHAMMAD
print(s.title())   # Muhammad
```

📌 `strip()`, `replace()`, `find()`, `count()`

- `strip()` → Removes **spaces** from start & end
- `replace()` → Replace parts of a string
- `find()` → Finds the **first index** of a character
- `count()` → Counts how many times a character appears

python

```
s = " Hello Python  "
print(s.strip())              # 'Hello Python'

s = "Banana"
print(s.replace("a", "*"))    # B*n*n*

print(s.find("n"))            # 2 (index of 1st 'n')

print(s.count("a"))           # 3
```

📌 **`startswith()`** / **`endswith()`**

Check if the string starts or ends with specific text.

```
python

s = "muhammad@gmail.com"
print(s.startswith("muhammad"))      # True
print(s.endswith(".com"))            # True
```

Used in validation, like checking file extensions, email format, etc.

◆ 4. `split()` and `join()`

📌 **`split()`**

Break string into **list of words**.

```
python

text = "I love Python"
words = text.split()
print(words)   # ['I', 'love', 'Python']
```

📌 **`join()`**

Combine list items into a **single string**.

```
python

words = ['I', 'love', 'Python']
print(" ".join(words))   # I love Python
```

Very useful in **sentence processing** or **data cleanup**.

◆ 5. String Formatting

📌 f-string:

```
python

name = "Muhammad"
print(f"My name is {name}")
```

📌 .format():

```
python

print("My name is {}".format(name))
```

Use for **clean output** or printing variables.

◆ 6. Content Checking Methods

These check the **type of content** in the string.

```
python

s = "abc123"
print(s.isalnum())   # True (letters + numbers)

print(s.isdigit())   # False (not all digits)

print("123".isdigit())  # True

print("Hello".isalpha())  # True (only letters)

print("   ".isspace())    # True (only spaces)
```

🔁 Strings in Loops

You've already done amazing things with loops and strings! □

✅ Reversing:

```
python

s[::-1]
```

✅ Palindrome check:

```
python

if s == s[::-1]
```

✅ Vowel counting:

```
python
```

```
for ch in s:
    if ch in 'aeiou':
        count += 1
```

✅ Password or menu input:

```
python

while True:
    if password == "open123":
        break
```

💡 Pro Tip:

Strings in Python are **immutable**, which means you can’t change them directly (like `s[0] = 'x'` won’t work). Instead, you create new strings using slicing, joining, or replacement.