Chapter 03 : Strings

Python Notes by Wahab

Python Strings: Everything You Need to Know

In Python, strings are sequences of characters enclosed within quotes (either single `''` or double `""`). Strings are **immutable**, meaning once created, they cannot be changed.

1. Creating a String

You can create a string using single, double, or triple quotes. Triple quotes allow for multiline strings.

- Example:

```
name = 'Wahab'         # Single quotes
greeting = "Hello!"  # Double quotes
paragraph = '''This is
a multiline
string.'''
```

String Slicing

Instead of accessing individual characters directly, we often use slicing to get parts of a string. Slicing allows us to extract substrings by specifying a range of indices.
Syntax: string[start:end] (Note: end is excluded)

- Example

```
text = "Python"
print(text[0:4])  # Output: Pyth (characters from index 0 to 3)
print(text[:4])   # Output: Pyth (same as above, start defaults to 0)
print(text[2:])   # Output: thon (from index 2 to the end)
```

You can also use negative indexing to slice from the end of the string:
`print(text[-3:]) # Output: hon (last three characters)`

String Slicing with Skipping Values

In Python, you can slice strings by specifying the start and end indices, but you can also **skip values** by using a third parameter called the **step**.

Syntax

The general syntax for slicing a string with a step value is:
`string[start:end:step]`

- Example

```
text = "Python is awesome"
# Slice the string from index 0 to 10, skipping every second character
print(text[0:10:2])  # Output: 'Pto s'
```

Common String Methods in Python

1. `lower()`

Converts all characters in the string to lowercase.

2. `upper()`

Converts all characters in the string to uppercase.

3. `strip()`

Removes leading and trailing spaces from the string.

4. `replace()`

Replaces a specified substring with another substring.

5. `split()`

Splits the string into a list using a specified delimiter.

6. `join()`

Joins elements of a list into a string with a specified separator.

7. `find()`

Returns the index of the first occurrence of a substring.

8. `count()`

Counts the number of times a substring appears in the string.

9. `startswith()`

Checks if the string starts with a specified prefix.

10. `endswith()`

Checks if the string ends with a specified suffix.

11. `capitalize()`

Capitalizes the first character of the string.

12. `title()`

Converts the string to title case (first letter of each word capitalized).

13. `isalpha()`

Checks if the string contains only alphabetic characters.

14. `isdigit()`

Checks if the string contains only digits.

15. `islower()`

Checks if all characters in the string are lowercase.

16. `isupper()`

Checks if all characters in the string are uppercase.

17. `len()`

Returns the length of the string (number of characters).

Commonly Used Escape Sequences in Python

Here are the escape sequences that are used most frequently:

- `\'` - Single quote
- `\"` - Double quote
- `\\` - Backslash
- `\n` - Newline (starts a new line)
- `\t` - Tab (adds a tab space)

These five escape sequences cover 90% of common string manipulation scenarios in Python.