Python Online Free Ramzan Course 2025
Taught by: Shaida Muhammad

Strings

Day 12 - Python Basics

Shaida Muhammad

Tuesday, March 18, 2025

ShaidaSherpao@gmail.com

Agenda

1 What are strings?

2 Creating and accessing strings

3 String slicing

4 String methods

5 String formatting (f-strings, format(), % operator)

6 Escape sequences

7 Hands-on practice

What are Strings?

- **Definition:** A string is a sequence of characters enclosed in single (' '), double (" "), or triple (''' ''' or "" "") quotes.
- **Features:**
  - **Immutable:** Cannot be changed after creation.
  - **Ordered:** Characters have a specific position (index).
- **Example:**

```
greeting = "Starry Mashy!"
```

Creating and Accessing Strings

- **Creating Strings:**

```
single_quotes = 'Starry'
double_quotes = "Mashy!"
triple_quotes = '''This is a multi-line string.'''
```

- **Accessing Characters:**
    - Use indexing to access individual characters.

```
text = "Python"
print(text[0])  # Output: P
print(text[-1]) # Output: n (negative indexing)
```

- **Key Points:**
    - Indexing starts at 0.
    - Negative indexing starts at -1 (from the end).

String Slicing

- **Definition:** Extracting a portion of a string using start:stop:step.
- **Example:**

```
text = "Hello, World!"
print(text[0:5])   # Output: Hello
print(text[7:12])  # Output: World
print(text[::2])   # Output: Hlo ol! (step of 2)
```

- **Key Points:**
  - start is inclusive, stop is exclusive.
  - Omitting start or stop slices from the beginning or to the end.

String Methods

Common Methods:

- upper(): Converts to uppercase.

```
text = "hello"
print(text.upper())
# Output: HELLO
```

- lower(): Converts to lowercase.

```
text = "HELLO"
print(text.lower())
# Output: hello
```

- strip(): Removes leading and trailing whitespace.

```
text = "  hello  "
print(text.strip())  # Output: hello
```

- replace(): Replaces a substring.

```
text = "hello"
print(text.replace("h", "j"))
# Output: jello
```

- split(): Splits a string into a list.

```
text = "hello world"
print(text.split())
# Output: ['hello', 'world']
```

- join(): Joins a list into a string.

```
words = ["hello", "world"]
print(" ".join(words))
# Output: hello world
```

String Formatting

- **f-Strings (Python 3.6+):**

```
name = "Ali"
age = 17
print(f"My name is {name} and I am {age} years old.")
```

- **`format()` Method:**

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

- **% Operator (Older C Style):**

```
print("My name is %s and I am %d years old." % (name, age))
```

- **Key Points:**
  - f-strings are the most modern and readable.
  - `format()` is flexible and works in older Python versions.
  - % operator is less common in modern code.

Escape Sequences

- **Definition:** Special characters starting with a backslash (\).
- **Common Escape Sequences:**
  - \n: Newline
  - \t: Tab
  - \\: Backslash
  - \": Double quote
  - \': Single quote
- **Example:**

```
print("Hello\nWorld!")  # Output: Hello (newline) World!
print("This is a \"quote\".")  # Output: This is a "quote".
```

Hands-On Practice

- **Task 1:** Create a string and print it.

```python
text = "Python is fun!"
print(text)
```

- **Task 2:** Access the first and last character of a string.

```python
print(text[0])   # Output: P
print(text[-1])  # Output: !
```

- **Task 3:** Slice the string to extract a substring.

```python
print(text[0:6])
# Output: Python
```

- **Task 4:** Use string methods to manipulate a string.

```python
print(text.upper())
# Output: PYTHON IS FUN!
print(text.replace("fun", "awesome"))
# Output: Python is awesome!
```

- **Task 5:** Format a string using f-strings.

```python
name = "Ali"
age = 17
print(f"My name is {name} and I am {age} years old.")
```

Recap

- Strings are sequences of characters and are immutable.
- Use indexing and slicing to access parts of a string.
- String methods like `upper()`, `lower()`, `strip()`, `replace()`, `split()`, and `join()` are powerful tools.
- String formatting can be done using f-strings, format(), or the % operator.
- Escape sequences allow special characters in strings.

Homework

1. Write a function `reverse_string(s)` that returns the reverse of a string.
2. Use f-strings to create a formatted sentence with variables.Write a program that takes a sentence and counts the number of words.
3. Write a function `is_palindrome(s)` that checks if a string is a palindrome.

Python Online Free Ramzan Course 2025
Taught by: Shaida Muhammad

Q&A

- Do you have any questions?
- Share your thoughts.

Closing

Next class: Modules and Libraries