

Chapter 3: Strings and Methods in Python

Introduction to Strings

A string is a sequence of characters enclosed in either single quotes (' '), double quotes (" "), or triple quotes (''' ' or "" "" """). Strings are widely used in Python for handling text-based data, making them essential in AI and Automation tasks.

Declaring Strings

Single-quoted string

```
greeting = 'Hello, World! '
```

Double-quoted string

```
name = "John Doe"
```

Triple-quoted string (used for multi-line strings)

```
message = '''This is a
multi-line string.'''
```

Real-Life Example: User Greetings

```
user_name = input("Enter your name: ")
print("Hello, " + user_name + "! Welcome to AI and Automation.")
```

Accessing Characters in a String

Strings are indexed, meaning we can access characters using their positions (starting from 0).

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

String Slicing

```
text = "Automation"
print(text[0:4]) # Output: Auto
print(text[:5]) # Output: Autom
print(text[3:]) # Output: omation
```

Common String Methods

Python provides several built-in string methods that make string manipulation easier.

1. `upper()` and `lower()`

Convert strings to uppercase or lowercase.

```
word = "Machine Learning"
print(word.upper()) # Output: MACHINE LEARNING
print(word.lower()) # Output: machine learning
```

Real-Life Example: Checking User Input Case

```
password = "SecRet123"
user_input = input("Enter your password: ")
if user_input.lower() == password.lower():
    print("Access Granted!")
else:
    print("Access Denied!")
```

2. `strip()`, `lstrip()`, and `rstrip()`

Remove whitespace from strings.

```
text = "  AI Automation  "
print(text.strip()) # Output: "AI Automation"
print(text.lstrip()) # Output: "AI Automation  "
print(text.rstrip()) # Output: "  AI Automation"
```

Real-Life Example: Cleaning User Input

```
email = "  user@example.com  "
clean_email = email.strip()
print("Processed Email:", clean_email)
```

3. `replace()`

Replace a part of a string with another.

```
sentence = "I love coding in Python."
print(sentence.replace("Python", "Java")) # Output: I love coding in Java.
```

4. `split()` and `join()`

Splitting a string into a list and joining a list into a string.

```
text = "AI,Machine Learning,Deep Learning"
words = text.split(",")
print(words) # Output: ['AI', 'Machine Learning', 'Deep Learning']
```

```
joined_text = " | ".join(words)
print(joined_text) # Output: AI | Machine Learning | Deep Learning
```

Real-Life Example: Extracting First and Last Name

```
full_name = "Elon Musk"
names = full_name.split()
first_name = names[0]
last_name = names[1]
print("First Name:", first_name)
print("Last Name:", last_name)
```

5. `find()` and `count()`

Find the position of a substring and count occurrences.

```
phrase = "Automation makes life easier."  
print(phrase.find("life")) # Output: 17  
print(phrase.count("e"))   # Output: 3
```

Real-Life Example: Finding a Hashtag in a Tweet

```
tweet = "Learning AI is fun! #AI #Automation"  
if tweet.find("#AI") != -1:  
    print("This tweet talks about AI!")
```

Exercises

Try these exercises to test your understanding:

1. Write a Python program that takes a string as input and prints it in uppercase and lowercase.
2. Extract the domain name from an email address (e.g., input: "user@example.com", output: "example.com").
3. Replace all occurrences of "bad" with "good" in a given string.
4. Count how many times the letter "a" appears in the sentence: "AI and automation are amazing."
5. Write a program that checks if a string contains the word "Python".
6. Write a Python script to extract the first and last word from a sentence entered by the user.

Challenge Task

Problem: Create a Python script that asks the user for their full name, removes any extra spaces, and then displays the following:

- The full name in uppercase.
- The full name in lowercase.
- The number of characters (excluding spaces).
- The first and last name separately.

Example Output:

```
Enter your full name: John Doe
Uppercase: JOHN DOE
Lowercase: john doe
Character Count (excluding spaces): 7
First Name: John
Last Name: Doe
```

Advanced Challenge: Reverse and Palindrome Checker

Task: Write a program that:

1. Takes a string input from the user.
2. Reverses the string.
3. Checks if the original and reversed strings are the same (Palindrome check).

```
Example Output:
Enter a word: radar
Reversed Word: radar
It's a palindrome!
```

Summary

- Strings are sequences of characters enclosed in quotes.
- Python provides various string methods for manipulation (e.g., `upper()`, `lower()`, `strip()`, `replace()`, `split()`, etc.).
- String slicing helps extract portions of a string.
- String indexing allows accessing individual characters.
- Real-world applications include processing user input, extracting details from text, and formatting data for AI and Automation.

By mastering strings, you can process text data efficiently, which is crucial for AI and Automation tasks!