

Chapter 9: Strings – In Depth (Upgraded)

"If lists are collections, strings are conversations. Every character matters."

What You Will Learn

- What strings are and how they work in Python
- How to use popular string methods
- How to slice, index, and loop through strings
- Why strings are immutable
- How to solve real problems using string logic
- A hands-on mini project

What is a String?

A string is text in Python, defined using quotes:

```
greeting = "Hello"  
name = 'Alice'
```

Strings are:

- Ordered (you can index them)
- Immutable (cannot be changed directly)
- Iterable (can loop through them)

Common String Methods

Method	Purpose
<code>.lower()</code>	Converts to lowercase
<code>.upper()</code>	Converts to uppercase
<code>.strip()</code>	Removes leading/trailing spaces
<code>.find()</code>	Returns index of substring or -1
<code>.replace()</code>	Replaces part of string
<code>.split()</code>	Splits string into a list
<code>.join()</code>	Joins list into string
<code>.isdigit()</code>	Checks if all characters are digits

Method	Purpose
<code>.isalpha()</code>	Checks if all characters are letters

Examples

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

print("banana".find("na")) # 2
print("I love Java".replace("Java", "Python")) # I love Python
```

Indexing and Negative Indexing

Each character has an index:

```
Index → 0  1  2  3  4  5
        P  y  t  h  o  n
Reverse → -6 -5 -4 -3 -2 -1
```

Examples:

```
word = "Python"
print(word[0]) # P
print(word[-1]) # n
print(word[-2]) # o
```

Slicing Strings

```
text = "Hello, world!"
```

```
print(text[0:5]) # Hello
print(text[:5]) # Hello
print(text[7:]) # world!
print(text[-6:-1]) # world
print(text[::-1]) # !dlrow ,olleH (reverse string)
```

Rule:

- `start` is inclusive
- `end` is exclusive
- Can skip values or use negative indexes

Looping Through Strings

```
for letter in "cat":  
    print(letter)
```

Output:

```
c  
a  
t
```

Strings are Immutable

```
text = "Hello"  
text[0] = "Y" # ❌ Error!
```

✅ Fix:

```
text = "Hello"  
new = "Y" + text[1:]  
print(new) # Hello
```

Mini Project: Email Formatter

Ask the user to enter their name and domain.

```
name = input("Enter name: ").strip().lower()  
domain = input("Enter domain: ").strip().lower()  
  
email = name + "@" + domain + ".com"  
print("Generated email:", email)
```

Practice Time

Mini Quiz (10 Qs)

1. What does `.strip()` do?
2. What is `word = "Python"; print(word[1:4])`?
3. Are strings mutable in Python?
4. How do you replace "dog" with "cat"?
5. What is the index of "world" in "Hello, world"?
6. What does `"123".isdigit()` return?

7. Write a slice to get "thon" from "Python"
 8. What happens with `.find("x")` if "x" isn't found?
 9. What is `len("abc")`?
 10. Fix the bug: `text = "Hi"; text[0] = "h"`
-

Basic Practice Problems (15)

- Print the first and last characters of a string
 - Ask user to input a name and greet them in all uppercase
 - Replace "Java" with "Python" in a string
 - Count how many times "a" appears in "banana"
 - Ask user for a sentence, and print the number of spaces
 - Take input and check if it's numeric (`.isdigit()`)
 - Get the last 3 characters of a string
 - Reverse a string using slicing
 - Extract the domain name from an email like "[user@gmail.com](#)"
 - Ask for a word, print its length and middle character
 - Remove leading/trailing spaces and make lowercase
 - Split a sentence into words and print each word
 - Join a list of words with `-`
 - Capitalize only the first letter of a word
 - Ask user for full name → print initials (e.g., John Doe → J.D.)
-

Intermediate Problems (5)

- Build a simple Caesar cipher (shift characters by 1)
 - Ask for a paragraph and count how many times "the" appears
 - Remove all vowels from a string
 - Create a word frequency counter using `.split()`
 - Ask for a password, validate: at least 8 chars, 1 digit, 1 uppercase
-

Debug Challenges (5)

What's wrong?

```
text = "Hello"  
text[0] = "Y"
```

This doesn't remove all spaces:

```
msg = " hello "  
print(msg.strip(" "))
```

Why is this giving error?

```
word = "test"  
print(word[4])
```







Fix:

```
name = "John"  
print(name.lower[0])
```

Output of:

```
"hello".find("z")
```

Quick Recap

-  Strings are sequences of characters
-  Indexing starts at 0; negative indexing starts at -1
-  Use slicing to extract or reverse parts
-  Strings are immutable
-  Use `.find()`, `.replace()`, `.strip()`, etc.
-  Loop through strings character-by-character

Explore More

Try:

```
help(str)  
dir(str)
```

And search for: **Python string methods list**

Python gives you over 40+ tools to handle text!