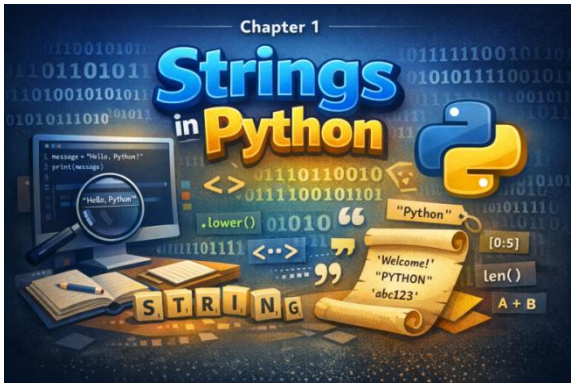
Chapter 1: Strings in Python (Detailed Course with Examples)



General Introduction:

In programming, a **string** is a sequence of characters used to represent text: words, sentences, symbols, numbers, etc.

Examples of strings:

- "Hello"
- "Python"
- "12345"
- "Djilali Bounaama University"

Strings are essential because they allow:

- Displaying messages
- Reading user input
- Text manipulation
- Storing information (names, addresses, codes, etc.)

Python provides many powerful and simple tools to work with strings efficiently.

1. Definition of a String:

In Python, a string can be defined using:

- Single quotes `' '`
- Double quotes `" "`
- Triple quotes `''' '''` or `""" """`

Example:

- name = "Seddik"
- city = 'Khemis Miliana'
- message = '"Welcome to the university"'

Remark: All three syntaxes create a string of type `str`.

Execution Result: No output (variables are stored).

2. Displaying a String (print):

The function `print()` displays text on the screen.

```
print("Hello")
print(name)
```

Execution Result:

Hello
Seddik
Remark: `print()` is used to show values to the user.
3. String Type
x = "Python"
print(type(x))
Execution Result:
<class 'str'>
Remark: `str` means string.
4. Length of a String
The function `len()` returns the number of characters.
word = "Python"
print(len(word))
Execution Result:
6
Remark: Spaces are also counted.
sentence = "Hello world"
print(len(sentence))
Execution Result:
11
5. Accessing Characters (Indexing)
Each character has a position (index):
P y t h o n
0 1 2 3 4 5
word = "Python"
print(word[0])
print(word[3])
Execution Result:
P
h
Remark: Indexing starts from 0.
Negative Index
print(word[-1])
print(word[-2])
Execution Result:
n
o
Remark: Negative indexes start from the end.
6. Slicing (Substrings)
word = "Programming"
print(word[0:5])

Execution Result:
Progr
Remark: The end index is excluded.
More examples:
print(word[:4])
print(word[4:])
print(word[::2])
Execution Result:
Prog
ramming
Pormig
7. Concatenation (Joining Strings)
first_name = "Ali"
last_name = "Ahmed"
print(first_name + " " + last_name)
Execution Result:
Ali Ahmed
Remark: + joins strings.
8. Repetition
print("Ha" * 3)
Execution Result:
HaHaHa
Remark: * repeats the string.
9. Converting to String
age = 25
print("I am " + str(age) + " years old")
Execution Result:
I am 25 years old
Remark: Numbers must be converted using `str()` before concatenation.
10. Important String Methods
a) upper() / lower()
text = "Python"
print(text.upper())
print(text.lower())
Execution Result:
PYTHON
python
b) capitalize()
print("python".capitalize())
Execution Result:
Python

c) strip()

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

Execution Result:

hello

Remark: Removes spaces at beginning and end.

d) replace()

```
print("hello".replace("lo", "p"))
```

Execution Result:

help

e) find()

```
print("python".find("t"))
```

Execution Result:

2

11. Checking Content

```
print("123".isdigit())
print("abc".isalpha())
print("abc123".isalnum())
```

Execution Result:

True
True
True

12. Looping Through a String

```
word = "Python"
for letter in word:
print(letter)
```

Execution Result:

P
y
t
h
o
n

13. String Comparison

```
print("abc" == "abc")
print("abc" == "ABC")
```

Execution Result:

True
False

14. String Formatting (f-strings)

```
name = "Ali"
```

```
age = 20
print(f"My name is {name} and I am {age} years old")
```
Execution Result:
```
My name is Ali and I am 20 years old
```
15. Multiline Strings
```
text = """
Hello
Welcome
Python
"""
print(text)
```
Execution Result:
```
Hello
Welcome
Python
```
16. Exercises with Solutions

Exercise 1

Display the length of the word "Informatics".

Solution:
```
print(len("Informatics"))
```
Result:
```
11
```
Exercise 2

Display the first and last letter of "Python".

Solution:
```
word = "Python"
print(word[0])
print(word[-1])
```
Result:
```
P
n
```
Exercise 3

Convert "hello" to uppercase.

Solution:
```
print("hello".upper())
```
Result:
```
HELLO
```
Conclusion

Strings are a fundamental element of Python. Mastering them is essential for building effective programs.

In the next chapters, we will learn how to combine strings with lists, files, and user input.