

★ STRINGS IN PYTHON ★

Date : _____

Page No. : _____

A string is a sequence of characters enclosed in single quotes (' '), double quotes (" ") or triple quotes (' ' ' ').

① INTRODUCTION

- Strings are immutable (cannot be changed).
- Elements of a string are characters.
- Indexing in python starts from 0.
- Example:

```
s = "CodeStep Academy"
Index: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Char: C o d e S t e p   A c a d e m y
```

② STRING OPERATIONS

(A) Concatenation (+)

Joins two or more strings.

Example:

```
s1 = "CodeStep"
s2 = "Academy"
s = s1 + " " + s2
print(s) # Output: CodeStep Academy
```

(B) Repetition (*)

Repeats a string n times.

Example:

```
s = "Hi! "
print(s * 3) # Output: Hi! Hi! Hi!
```

(C) Membership (in, not in)

Checks if a substring exists in a string.

Example:

```
s = "Python is fun"
print("Python" in s) # True
print("java" not in s) # True
```

(D) Slicing

Extracts a part of string.

Syntax: s[start:stop:step]

Example:

```
s = "CodeStepAcademy"
print(s[0:4]) # Code
print(s[4:8]) # Step
print(s[:4]) # Code
print(s[4:]) # StepAcademy
print(s[::2]) # CdeSpcdmy
```

Note:

- start is inclusive
- stop is exclusive
- step is optional
- default step is 1

(E) Traversing a String using Loops

We can access each character using loops.

Example:

```
s = "Python"
for ch in s:
    print(ch, end = " ")
# Output: P y t h o n
```

③ BUILT-IN FUNCTIONS / METHODS OF STRINGS

No.	Method	Description	Example	Output
1	len()	Returns length of string	len("CodeStep")	8
2	capitalize()	Converts first character to upper case	"python".capitalize()	'Python'
3	title()	Converts first character of each word to upper case	"hello world".title()	'Hello World'
4	lower()	Converts all characters to lower case	"PYTHON".lower()	'python'
5	upper()	Converts all characters to upper case	"python".upper()	'PYTHON'
6	count(sub)	Returns number of occurrences of substring	"banana".count("a")	3
7	find(sub)	Returns lowest index of substring, -1 if not found	"banana".find("na")	2
8	index(sub)	Returns lowest index of substring, raises error if not found	"banana".index("na")	2
9	endswith(suffix)	Checks if string ends with suffix (True/False)	"file.txt".endswith(".txt")	True
10	startswith(prefix)	Checks if string starts with prefix (True/False)	"Python123".startswith("Py")	True
11	isalnum()	Returns True if all characters are alphanumeric	"abc123".isalnum()	True
12	isalpha()	Returns True if all characters are alphabetic	"Python".isalpha()	True
13	isdigit()	Returns True if all characters are digits	"12345".isdigit()	True
14	islower()	Returns True if all cased characters are lower case	"python".islower()	True
15	isupper()	Returns True if all cased characters are upper case	"PYTHON".isupper()	True
16	isspace()	Returns True if all characters are whitespace	"\t\n".isspace()	True
17	lstrip()	Removes leading whitespace	"hello".lstrip()	'hello'
18	rstrip()	Removes trailing whitespace	"hello ".rstrip()	'hello'
19	strip()	Removes leading and trailing whitespace	"hello ".strip()	'hello'
20	replace(old, new)	Replaces all occurrences of old with new	"one one".replace("one", "1")	'1 1'
21	join(iterable)	Joins elements of iterable with string separator	"-".join(["a", "b", "c"])	'a-b-c'

partition(sep)

Splits string at first occurrence of sep. Returns a tuple: (before, sep, after)

Example:

```
s = "key-value-pair"
print(s.partition("-"))
# Output: ('key', '-', 'value-pair')
```

Remember

Strings are immutable. Any operation creates a new string.

split(sep=None, maxsplit=-1)

Splits string into a list.

Example:

```
s = "apple, banana, grape"
print(s.split(",")) # ['apple', 'banana', 'grape']
print(s.split()) # ['apple', 'banana', 'grape']
print("a b c".split()) # ['a', 'b', 'c']
```

Exam Tip

Practice these methods with examples. They are very important in programs!

Stay Curious,
Keep Coding! ★

Practice more at www.codestepacademy.in

Scan to Visit
Website →

