

String Basics

Accessing, Slicing, Iterating, Concatenating



Table of contents

01

Introduction

02

Accessing

03

Slicing

04

Iterating

05

Concatenating

01

Introduction



Introduction

- A **string** in Python is a **sequence** of characters enclosed in single ('), double ("") or triple (''' or ''''') quotes.
- Thus:
`'Hello' ~ "Hello" ~ """Hello"" ~ """Hello"""`
- Strings are **immutable**, meaning their contents **cannot** be changed after creation.

02

Accessing String

Accessing String

Python treats a string as a **sequence** of characters, each of which can be accessed individually using **indexing**.

Positive Indexing (starts with 0):

P	Y	T	H	O	N
0	1	2	3	4	5

Last Character: `len(string) - 1`

Accessing String

Python treats a string as a **sequence** of characters, each of which can be accessed individually using **indexing**.

Negative Indexing (starts with -1):

P	Y	T	H	O	N
-6	-5	-4	-3	-2	-1

Last Character: -1

Accessing String

ERROR!

IndexError: string index out of range

If you try to access an index out of range, Python raises an IndexError.

03

Slicing String



Slicing String

Python allows extracting substrings using **slicing** with the syntax:

Python

```
string[start:end:step]
```

start: The starting index (inclusive).

end: The stopping index (exclusive).

step: The step value (default is 1).

Slicing With Positive Indexes

P	Y	T	H	O	N
0	1	2	3	4	5

s[0:3] → Extracts "PYT" (index 0 to 2).

s[:4] → Extracts "PYTH" (from index 0 to 3).

s[2:] → Extracts "THON" (from index 2 to end).

s[:] → Extracts "PYTHON" (entire string).

Slicing With Negative Indexes

P	Y	T	H	O	N
-6	-5	-4	-3	-2	-1

`s[-4:-1]` → Extracts "**THO**" (index -4 to -2).

`s[-5:]` → Extracts "**YTHON**" (from index -5 to end).

`s[::-2]` → Extracts "**PYTH**" (excludes last two characters).

Slicing With Step (Skipping Characters)

P	Y	T	H	O	N
0	1	2	3	4	5

s[0:6:2] → Extracts "PTO" (every 2nd character).

s[::-1] → Extracts "NOHTYP" (reversed).

s[::-2] → Extracts "NHY" (every 2nd character in reverse).

04

Iterating Over String

Iterating Over String

Since a string is an iterable sequence, you can loop over it to access characters.

<u>Python</u>	<u>Output</u>
<pre>s = "PYTHON" for char in s: print(char)</pre>	P Y T H O N

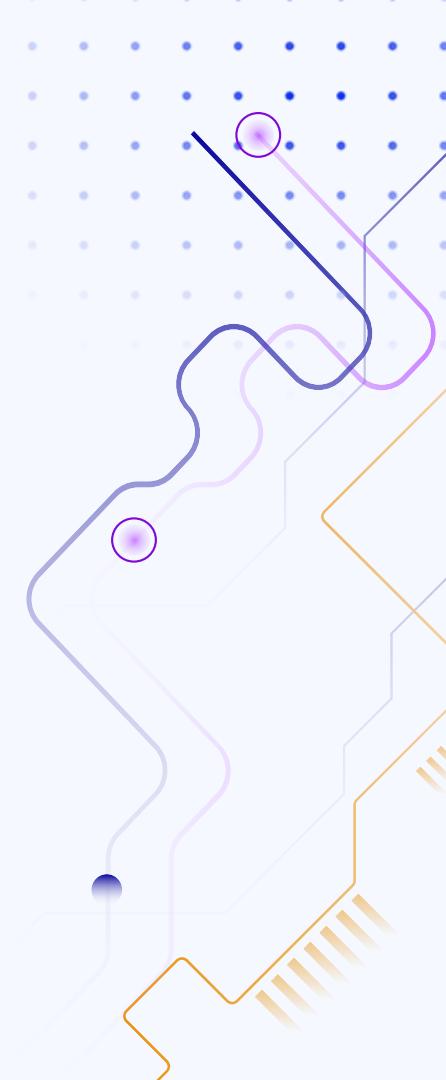
Iterating Over String

Since a string is an iterable sequence, you can loop over it to access characters.

<u>Python</u>	<u>Output</u>
<pre>s = "PYTHON" i = 0 while i < len(s): print(s[i]) i += 1</pre>	P Y T H O N

05

Concatenation and Repetition



Concatenation (+)

- Concatenation means **joining** two or more strings together.
- You can use the **(+)** operator for concatenation.

<u>Python</u>	<u>Output</u>
s1 = "Hello" s2 = "World" print(s1 + " " + s2)	Hello World

Concatenation (+)

Python does not allow concatenation between **strings** and **non-strings** (e.g., integers).

<u>Python</u>	<u>Output</u>
<pre>s1 = "Hello" s2 = 3 print(s1 + s2)</pre>	TypeError: can only concatenate str (not "int") to str

Repetition (*)

You can use the (*) operator to repeat a string multiple times.

<u>Python</u>	<u>Output</u>
s = "Hi" print(s * 3)	HiHiHi

Knowledge Reinforcement



1

Question

What does the following code print?

Python

```
s = "PYTHON"  
print(s[2])
```

Answer

- A. P
- B. Y
- C. T
- D. H

P	Y	T	H	O	N
0	1	2	3	4	5

1

Question

What does the following code print?

Python

```
s = "PYTHON"  
print(s[2])
```

Answer

A. P

B. Y

C. T

D. H

P	Y	T	H	O	N
0	1	2	3	4	5

2

Question

What does the following code print?

Python

```
s = "HELLO"  
print(s[10])
```

H	E	L	L	O
0	1	2	3	4

Answer

- (A) Prints a blank space
- (B) Returns None
- (C) Raises an IndexError
- (D) Prints the last character of the string

2

Question

What does the following code print?

Python

```
s = "HELLO"  
print(s[10])
```

H	E	L	L	O
0	1	2	3	4

Answer

- (A) Prints a blank space
- (B) Returns None
- (C) Raises an IndexError**
- (D) Prints the last character of the string

3

Question

What does the following code print?

Python

```
s = "HELLO"  
print(s[:3])
```

Answer

- (A) HELL
- (B) HEL
- (C) HE
- (D) LLO

H	E	L	L	O
0	1	2	3	4

3

Question

What does the following code print?

Python

```
s = "HELLO"  
print(s[:3])
```

Answer

- (A) HELL
- (B) HEL**
- (C) HE
- (D) LLO

H	E	L	L	O
0	1	2	3	4

4

Question

What does the following code print?

Python

```
s = "PYTHON"  
print(s[-4])
```

Answer

- (A) P
- (B) T
- (C) Y
- (D) N

P	Y	T	H	O	N
-6	-5	-4	-3	-2	-1

4

Question

What does the following code print?

Python

```
s = "PYTHON"  
print(s[-4])
```

Answer

(A) P

(B) T

(C) Y

(D) N

P	Y	T	H	O	N
-6	-5	-4	-3	-2	-1

5

Question

What does the following code print?

Python

```
s = "PYTHON"  
print(s[2:5])
```

Answer

- (A) THO
- (B) YTH
- (C) T
- (D) TH

P	Y	T	H	O	N
0	1	2	3	4	5

5

Question

What does the following code print?

Python

```
s = "PYTHON"  
print(s[2:5])
```

Answer

(A) THO

(B) YTH

(C) T

(D) TH

P	Y	T	H	O	N
0	1	2	3	4	5

6

Question

What does the following code print?

Python

```
s = "PYTHON"  
print(s[1:5:2])
```

Answer

- (A) YH
- (B) YT
- (C) YN
- (D) YO

P	Y	T	H	O	N
0	1	2	3	4	5

6

Question

What does the following code print?

Python

```
s = "PYTHON"  
print(s[1:5:2])
```

Answer

- (A) YH
- (B) YT
- (C) YN
- (D) YO

P	Y	T	H	O	N
0	1	2	3	4	5

7

Question

What does the following code print?

Python

```
s = "PROGRAMMING"  
print(s[::-1])
```

Answer

- (A) NOIMMARGORP
- (B) PROGRAMMING
- (C) ROGRAMMING
- (D) GNIMMARGORP

P	R	O	G	R	A	M	M	I	N	G
0	1	2	3	4	5	6	7	8	9	10

7

Question

What does the following code print?

Python

```
s = "PROGRAMMING"  
print(s[::-1])
```

Answer

- (A) NOIMMARGORP
- (B) PROGRAMMING
- (C) ROGRAMMING
- (D) GNIMMARGORP**

P	R	O	G	R	A	M	M	I	N	G
0	1	2	3	4	5	6	7	8	9	10

8

Question

How can you access **every second character** in a string?

Python

```
s = "PYTHON"
```

P	Y	T	H	O	N
0	1	2	3	4	5

Answer

- (A) s[::-2]
- (B) s[1::2]
- (C) s[::-2]
- (D) All of the above

8

Question

How can you access **every second character** in a string?

Python

```
s = "PYTHON"
```

P	Y	T	H	O	N
0	1	2	3	4	5

Answer

- (A) s[::-2]
- (B) s[1::2]
- (C) s[::-2]
- (D) All of the above

9

Question

What is the output of the following?

Python

```
s = "HELLO"  
print(s[-3:-1])
```

Answer

- (A) LL
- (B) LLO
- (C) EL
- (D) LO

H	E	L	L	O
-5	-4	-3	-2	-1

9

Question

What is the output of the following?

Python

```
s = "HELLO"  
print(s[-3:-1])
```

Answer

(A) LL

(B) LLO

(C) EL

(D) LO

H	E	L	L	O
0	1	2	3	4

10

Question

What is the output of the following?

Python

```
s = "abcdefgh"  
print(s[1:-1:2])
```

Answer

- (A) aceg
- (B) bdfh
- (C) bdf
- (D) None of the above

a	b	c	d	e	f	g	h
0	1	2	3	4	5	6	7

10

Question

What is the output of the following?

Python

```
s = "abcdefgh"  
print(s[1:-1:2])
```

Answer

- (A) aceg
- (B) bdfh
- (C) bdf**
- (D) None of the above

a	b	c	d	e	f	g	h
0	1	2	3	4	5	6	7

11

Question

What is the output of the following?

Python

```
s = "HELLO"  
s[0] = 'J'
```

H	E	L	L	O
0	1	2	3	4

Answer

- (A) Changes the first character to 'J'
- (B) Raises a TypeError
- (C) Nothing Happens
- (D) None of the above

11

Question

What is the output of the following?

Python

```
s = "HELLO"  
s[0] = 'J'
```

H	E	L	L	O
0	1	2	3	4

Answer

- (A) Changes the first character to 'J'
- (B) Raises a TypeError**
- (C) Nothing Happens
- (D) None of the above

12

Question

What is the output of the following?

Python

```
s = "ABCDEFGHIJ"  
print(s[-8:-2:2])
```

Answer

- (A) CEG
- (B) BDFH
- (C) BDF
- (D) None of the above

A	B	C	D	E	F	G	H	I	J
-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

12

Question

What is the output of the following?

Python

```
s = "ABCDEFGHIJ"  
print(s[-8:-2:2])
```

Answer

(A) CEG

(B) BDFH

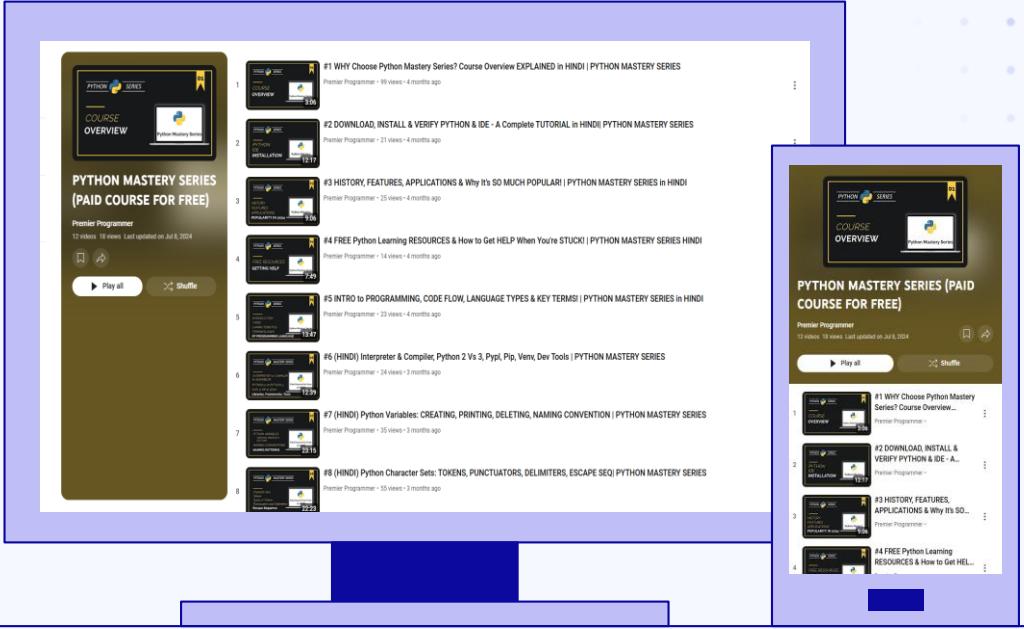
(C) BDF

(D) None of the above

A	B	C	D	E	F	G	H	I	J
-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

WATCH

Level up your coding with each episode in this focused Python series.



Next Video!

String Methods

