

Slicing Questions

Created By [@Mohamed Hamed](#)

Artificial Intelligence student at KFS university, you can find me down:

- [LinkedIn](#)
 - [GitHub](#)
 - [YouTube](#)
 - ✓ To get the [file notes](#)
 - ✓ To get the [questions](#)
 - ✓ To get the [answers](#)
-

0. Which of the following correctly describes the end index in Python slicing [start:end:step]?
 - Inclusive, meaning the element at end is included in the slice.
 - Exclusive, meaning the element at end is NOT included in the slice.
 - It represents the number of elements to slice.
 - It is always ignored if step is negative.
1. If step = -1 in a slice operation, what is the direction of traversal?
 - Right to left (backward)
 - Left to right (forward)
 - It depends on the start and end indices.
 - It means no traversal occurs.
2. Consider the string s = "Python". What will s[-3:-1] return?

a) "tho"	c) "th"
b) "ho"	d) "on"
3. Which of the following statements is TRUE regarding negative indices in Python slicing?
 - Negative indices are only used for backward slicing.
 - Python converts negative indices to their positive equivalent before slicing.
 - A negative start index always results in an empty slice.
 - Negative indices are not allowed in the step parameter.
4. Given my_list = [1, 2, 3, 4, 5], what is the output of my_list[4:1:-1]?
 - [5, 4, 3]
 - [3, 4, 5]
 - [5, 4, 3, 2]
 - [2, 3, 4, 5]

5. If step > 0 and start >= end (after resolving negative indices), what will be the result of the slice?

- a) A slice containing only the start element.
- b) A slice containing only the end element.
- c) A slice containing all elements from start to end.
- d) An empty slice.

6. For a slice operation with step < 0, when will the slice be empty?

- a) When start > end.
- b) When start < end.
- c) When start = end.
- d) When start <= end.

7. What is the purpose of my_sequence[::-1]?

- a) To reverse the entire sequence.
- b) To get the last element of the sequence.
- c) To get the first element of the sequence.
- d) To get every other element in reverse order.

8. Consider data = "hello world". What will data[::2] return?

- a) "hlowrd"
- b) "el ol"
- c) "hlool"
- d) "hlwrd"

9. Which of the following slicing operations would extract the last three characters of a string s?

- a) s[len(s)-3:]
- b) s[-3:]
- c) s[:-3]
- d) Both a and b

10. While slicing a string if start is a negative number end also like start with a positive step so to get the output, start should be?

- a) Less than end in value.
- b) Greater than end in index position.
- c) Greater than end in value.
- d) Less than end in index position.

11. Answer the following questions:

1. Consider s = "abcdefg". What will s[-4:6] with one step from R to L return?

- a) "def"
- b) empty string
- c) "gfe"
- d) "defg"

2. x = x + 5 while x = s[-1:2] with one step forward what is the output?

- a) "gfed"
- b) TypeError
- c) ValueError
- d) "defg"

3. print('x' if x else 'y'):

- a) "x"
- b) "y"
- c) TypeError
- d) SyntaxError

4. before converting any negative indices if x = s[-1:2:-1]:

- a) always empty string
- b) "gfed"
- c) maybe gets output

5. as a python behavior the output of x?

- a) always empty string
- b) "gfed"
- c) maybe gets output

12. Consider s = "hello world". What will s[len(s)-7:len(s)-1:-2] return?

- a) SyntaxError
- b) empty string
- c) "drw"

13. Given x= [[1, 2], [5, 7], [[4, 9]]], what is the output of x[1][1]?

- e) [7]
- f) 7
- g) [5, 7]
- h) 5

14. Answer the following questions:

```
matrix = [
    [11, 12, 13, 14],
    [21, 22, 23, 24],
    [31, 32, 33, 34],
    [41, 42, 43, 44]
]
```

1. matrix[3][1:]

- a) [42, 43, 44]
- b) 42, 43, 44
- c) [32, 33, 34]
- d) 41, 42, 43

2. matrix[::-1][:2]

- a) [[31, 32, 33, 34], [41, 42, 43, 44]]
- b) ValueError
- c) 41, 42, 43, 44, [31, 32, 33, 34]
- d) [[41, 42, 43, 44], [31, 32, 33, 34]]

3. matrix[::-2][::-1]

- a) IndexError
- b) [21, 22, 23, 24], [31, 32, 33, 34]
- c) [31, 32, 33, 34], [11, 12, 13, 14]
- d) 21, 22, 23, 24, [31, 32, 33, 34]

4. `matrix[1:][::-1][0][1:3]`

- a) 42, 43 b) [42, 43] c) 42, 43, 44 d) 43, 44

5. `x = matrix[-1][-4] * 5`

- a) [42, 42, 42, 42, 42] b) 205 c) 210 d) 42, 42, 42, 42, 42