

1. Which of the following expressions extracts the second element from a one-dimensional NumPy array `arr`?

- A) `arr[1]`
- B) `arr[2]`
- C) `arr[0:2]`
- D) `arr[1:2]`

2. What will be the output of the following slicing operation on a one-dimensional array?

```
```python
```

```
arr = np.array([1, 2, 3, 4, 5])
```

```
slice_ = arr[2:4]
```

```
print(slice_)
```

```
```
```

- A) `[3, 4]`
- B) `[2, 3]`
- C) `[4, 5]`
- D) `[3, 4, 5]`

3. Given a two-dimensional NumPy array `arr_2d`, which of the following expressions extracts the second row?

- A) `arr_2d[1]`
- B) `arr_2d[0:2]`
- C) `arr_2d[:, 1]`
- D) `arr_2d[1, :]

4. What does the slicing expression `arr[:, 1:3]` do for a two-dimensional array `arr`?

- A) Selects all rows and all columns from the second to the third column.
- B) Selects all columns and all rows from the second to the third row.
- C) Selects all rows and all columns from the second to the third element.
- D) Selects all rows and only the second and third columns.

5. Which of the following slicing expressions extracts a subarray from the upper-left quadrant of a two-dimensional array `arr_2d`?

- A) `arr_2d[0:2, 0:2]`
- B) `arr_2d[:-2, :2]`
- C) `arr_2d[0:2, :]`
- D) `arr_2d[:, :2]`

6. What will be the shape of the sliced array when applying `arr[1:4, np.newaxis, :]` to a three-dimensional array `arr`?

- A) `(4, 1, n)`
- B) `(3, 1, n)`
- C) `(3, n, 1)`
- D) `(4, n, 1)`

7. Which of the following expressions is equivalent to `arr[:-2]` for a one-dimensional array `arr`?

- A) `arr[2:]`
- B) `arr[:-3]`
- C) `arr[:len(arr)-2]`
- D) `arr[:-1]`

8. What does the slicing expression `arr[..., 1]` do for a three-dimensional array `arr`?

- A) Selects all elements from the first axis with the second index.
- B) Selects all elements from the second axis with the first index.
- C) Selects all elements from the last axis with the second index.
- D) Selects all elements from the second-to-last axis with the first index.

9. Given a two-dimensional NumPy array `arr_2d`, which slicing expression will select the last element from the last row?

- A) `arr_2d[-1, -1]`
- B) `arr_2d[-1:, -1]`
- C) `arr_2d[-1:, :]`
- D) `arr_2d[:, -1]`

10. What will be the output of the following slicing operation on a one-dimensional array?

```
```python
arr = np.array([1, 2, 3, 4, 5])
slice_ = arr[::-1]
print(slice_)
```
```

- A) `[5, 4, 3, 2, 1]`
- B) `[1, 2, 3, 4, 5]`
- C) `[5, 4, 3, 2]`
- D) `[4, 3, 2, 1]`

11. What would be the output of the following slicing operation on a two-dimensional array `arr_2d`?

```

` `` python
arr_2d = np.array([[1, 2, 3],
                   [4, 5, 6],
                   [7, 8, 9]])

slice_2d = arr_2d[1:, 1:]

print(slice_2d)
` ``

```

- A) ` [[5, 6], [8, 9]] `
- B) ` [[4, 5], [7, 8]] `
- C) ` [[2, 3], [5, 6], [8, 9]] `
- D) ` [[5, 6, 7], [8, 9]] `

12. Given a three-dimensional array `arr_3d`, what will be the shape of the sliced array when applying `arr_3d[:, :, ::2]`?

- A) The same as `arr_3d`
- B) The same along the first and second axis, but reduced by half along the third axis
- C) Half the size along all axes
- D) Double the size along all axes

13. What does the following slicing expression do to a two-dimensional array `arr_2d`?

```

` `` python
arr_2d = np.array([[1, 2, 3],
                   [4, 5, 6],
                   [7, 8, 9]])

slice_2d = arr_2d[:, ::2]

` ``

```

- A) Selects every other row and reverses the columns
- B) Selects every other column and reverses the rows
- C) Reverses both rows and columns
- D) Selects every other element and reverses the order

14. What will be the output of the following slicing operation on a three-dimensional array?

```
```python
```

```
arr_3d = np.array([[[1, 2, 3],
 [4, 5, 6]],
 [[7, 8, 9],
 [10, 11, 12]]])
```

```
slice_3d = arr_3d[:, :, ::-1]
```

```
print(slice_3d)
```

```
```
```

- A) `[[[3, 2, 1], [6, 5, 4]], [[9, 8, 7], [12, 11, 10]]]`
- B) `[[[6, 5, 4], [3, 2, 1]], [[12, 11, 10], [9, 8, 7]]]`
- C) `[[[3, 4, 5], [2, 1, 6]], [[9, 10, 11], [8, 7, 12]]]`
- D) `[[[6, 5, 4], [3, 2, 1]], [[12, 11, 10], [9, 8, 7]]]`

15. What would be the output of the following slicing operation on a one-dimensional array?

```
```python
```

```
arr = np.array([1, 2, 3, 4, 5])
```

```
slice_ = arr[[0, 2, 4]]
```

```
print(slice_)
```

```
```
```

A) `[1, 3, 5]`

B) `[0, 2, 4]`

C) `[2, 4]`

D) `[1, 2, 3]`

16. Given a two-dimensional array `arr_2d`, what does the slicing expression `arr_2d[::-1, ::-1]` do?

A) Reverses the order of elements along each axis

B) Reverses the rows and columns

C) Selects every other element in reverse order

D) Reverses the entire array

17. What will be the output of the following slicing operation on a two-dimensional array?

```
```python
```

```
arr_2d = np.array([[1, 2, 3],
```

```
 [4, 5, 6],
```

```
 [7, 8, 9]])
```

```
slice_2d = arr_2d[-2:, -2:]
```

```
print(slice_2d)
```

```
```
```

A) `[[2, 3], [5, 6]]`

B) `[[4, 5], [7, 8]]`

C) `[[5, 6], [8, 9]]`

D) `[[6, 5], [9, 8]]`

18. Given a three-dimensional array `arr_3d`, what will be the shape of the sliced array when applying `arr_3d[1:, ::-1, ::2]`?

- A) `(2, n, n)`
- B) `(2, n/2, n)`
- C) `(n/2, n, 2)`
- D) `(n, n, n)`

19. What will be the output of the following slicing operation on a two-dimensional array?

```
```python
arr_2d = np.array([[1, 2, 3],
 [4, 5, 6],
 [7, 8, 9]])
slice_2d = arr_2d[:, ::2]
print(slice_2d)
```
```

- A) `[[1, 3], [4, 6], [7, 9]]`
- B) `[[1, 4, 7], [2, 5, 8], [3, 6, 9]]`
- C) `[[1, 2], [4, 5], [7, 8]]`
- D) `[[1, 3], [4, 6], [7, 9]]`

20. What would be the output of the following slicing operation on a three-dimensional array?

```
```python
arr_3d = np.array([[[1, 2, 3],
 [4, 5, 6]],
 [[7, 8, 9],
```

```
[10, 11, 12]]])
```

```
slice_3d = arr_3d[::-1,::-1,::-1]
```

```
print(slice_3d)
```

```
...
```

A) `[[[12, 11, 10], [9, 8, 7]], [[6, 5, 4], [3, 2, 1]]]`

B) `[[[6, 5, 4], [3, 2, 1]], [[12, 11, 10], [9, 8, 7]]]`

C) `[[[3, 2, 1], [6, 5, 4]], [[9, 8, 7], [12, 11, 10]]]`

D) `[[[10, 9, 8], [7, 6, 5]], [[4, 3, 2], [1, 0, -1]]]`