

Python List Slice

If this Python Tutorial saves you
hours of work, please **whitelist it in**
your ad blocker 🙏 and

Donate Now

(<https://www.pythontutorial.net/donation/>)

to help us ❤️ pay for the web
hosting fee and CDN to keep the

website running.

Summary: in this tutorial, you'll learn about Python list slice and how to use it to manipulate lists effectively.

Introduction to Python List slice notation

Lists (<https://www.pythontutorial.net/python-basics/python-list/>) support the slice notation that allows you to get a sublist from a list:

```
sub_list = list[begin: end: step]
```

In this syntax, the **begin** , **end** , and **step** arguments must be valid indexes. And they're all optional.

The **begin** index defaults to zero. The **end** index defaults to the length of the list. And the **step** index defaults to 1.

The slice will start from the **begin** up to the **end** in the step of **step** .

The **begin** , **end** , and **step** can be positive or negative. Positive values slice the list from the first element to the last element while negative values slice the list from the last element to the first element.

In addition to extracting a sublist, you can use the list slice to change the list such as updating, resizing, and deleting a part of the list.

Python List slice examples

Let's take some examples of using the list slice.

1) Basic Python list slice example

Suppose that you have the following list of strings:

```
colors = ['red', 'orange', 'yellow', 'green', 'blue', 'indigo', 'violet']
```

The following example uses the list slice to get a sublist from the `colors` list:

```
colors = ['red', 'orange', 'yellow', 'green', 'blue', 'indigo', 'violet']
sub_colors = colors[1:4]

print(sub_colors)
```

Output:

```
['orange', 'yellow', 'green']
```

The `begin` index is 1, so the slice starts from the `'orange'` color. The `end` index is 4, therefore, the last element of the slice is `'green'`.

As a result, the slice creates a new list with three colors: `['orange', 'yellow', 'green']`.

This example doesn't use the `step`, so the slice gets all values within the range without skipping any elements.

2) Using Python List slice to get the n-first elements from a list

To get the n-first elements from a list, you omit the `begin` argument:

```
list[:n]
```

The following example returns a list that includes the first three elements from the colors list:

```
colors = ['red', 'orange', 'yellow', 'green', 'blue', 'indigo', 'violet']
sub_colors = colors[:3]

print(sub_colors)
```

Output:

```
['red', 'orange', 'yellow']
```

Notice that the `colors[:3]` is equivalent to the `color[0:3]` .

3) Using Python List slice to get the n-last elements from a list

To get the n-last elements of a list, you use the negative indexes.

For example, the following returns a list that includes the last 3 elements of the `colors` list:

```
colors = ['red', 'orange', 'yellow', 'green', 'blue', 'indigo', 'violet']
sub_colors = colors[-3:]

print(sub_colors)
```

Output:

```
['blue', 'indigo', 'violet']
```

4) Using Python List slice to get every nth-element from a list

The following example uses the `step` to return a sublist that includes every 2nd element of the colors list:

```
colors = ['red', 'orange', 'yellow', 'green', 'blue', 'indigo', 'violet']
sub_colors = colors[::2]

print(sub_colors)
```

Output:

```
['red', 'yellow', 'blue', 'violet']
```

5) Using Python List slice to reverse a list

When you use a negative step, the slice includes the list elements starting from the last element to the first element. In other words, it reverses the list. See the following example:

```
colors = ['red', 'orange', 'yellow', 'green', 'blue', 'indigo', 'violet']
reversed_colors = colors[::-1]

print(reversed_colors)
```

Ouptut:

```
['violet', 'indigo', 'blue', 'green', 'yellow', 'orange', 'red']
```

6) Using Python List slice to substitute part a list

Besides extracting a part of a list, the list slice allows you to change the list element.

The following example changes the first two elements in the colors list to the new values:

```
colors = ['red', 'orange', 'yellow', 'green', 'blue', 'indigo', 'violet']
colors[0:2] = ['black', 'white']

print(colors)
```

Output:

```
['black', 'white', 'yellow', 'green', 'blue', 'indigo', 'violet']
```

7) Using Python List slice to partially replace and resize a list

The following example use the list slice to replace the first and second elements by the new ones and also add a new element to the list:

```
colors = ['red', 'orange', 'yellow', 'green', 'blue', 'indigo', 'violet']
print(f"The list has {len(colors)} elements")

colors[0:2] = ['black', 'white', 'gray']
print(colors)
print(f"The list now has {len(colors)} elements")
```

Output:

```
The list has 7 elements
['black', 'white', 'gray', 'yellow', 'green', 'blue', 'indigo', 'violet']
The list now has 8 elements
```

8) Using Python list slice to delete elements

The following shows how to use the list slice to delete the 3rd, 4th, and 5th elements from the `colors` list:

```
colors = ['red', 'orange', 'yellow', 'green', 'blue', 'indigo', 'violet']  
del colors[2:5]  
  
print(colors)
```

Output:

```
['red', 'orange', 'indigo', 'violet']
```

Summary

- Use list slice to extract a sublist from a list and modify the list.