Python: solutions to access the last element in a list

In Python, a list is an ordered sequence of items, where each item has an index that identifies it in the list. The index of the first element of a list is 0 and the index of the last element is length of the list minus 1. To access the last element of a Python list, there are several solutions.

The first solution is to use the index -1 to access the last element of the list. For example, if we have a list named `my_list`, we can access the last item with the following statement:

```
last_element = my_list[-1]
```

This statement returns the last element of the `my_list` list, regardless of its length.

Another solution is to use the `pop()` method of the list, which removes and returns the last element of the list. For example:

```
my_list = [1, 2, 3, 4, 5]
last_element = my_list.pop()
```

In this case, the variable `last_element` will contain the value of the last element of the list `my_list`, which will also be removed from the list itself.

Finally, if we just want to access the last element without removing it from the list, we can use the `len()` function to calculate the length of the list and then use the index of the last element. For example:

```
my_list = [1, 2, 3, 4, 5]
last_element = my_list[len(my_list)-1]
```

This statement returns the last element of the list `my_list`, using the index calculated as the length of the list minus 1.

Another solution to access the last element of a Python list is to use slicing. Slicing allows us to select a portion of the list using a range of indices.

To select only the last element of the list, we can use slicing with index -1 as in this example:

```
my_list = [1, 2, 3, 4, 5]
last_element = my_list[-1:]
```

In this case, the `last_element` variable will contain a new list containing only the last element of the `my_list` list. However, since this method returns a list, we need to access the actual element using index 0 as follows:

```
last_element = my_list[-1:][0]
```

Alternatively, we can use slicing with the range [length of list - 1: length of list], like in this example:

```
my_list = [1, 2, 3, 4, 5]
last_element = my_list[len(my_list)-1:]
```

In this case, the `last_element` variable will contain a new list containing only the last element of the `my_list` list. Again, we need to access the actual element using index 0 as follows:

```
last_element = my_list[len(my_list)-1:][0]
```

Slicing is another solution for accessing the last element of a Python list. However, we have to consider that this solution returns a new list containing only the last element, and we have to access the actual element using index 0. Therefore many developers prefer the solutions seen above.