



## 9.3. List Element Deletion

Using slices to delete list elements can be awkward and therefore error-prone. Python provides an alternative that is more readable. The `del` statement removes an element from a list by using its position.

Save & Run

5/13/2021, 1:52:44 PM - 2 of 2

Show in CodeLens

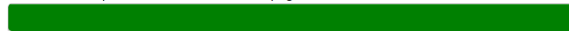
```
1 a = ['one', 'two', 'three']
2 del a[1]
3 print(a)
4
5 alist = ['a', 'b', 'c', 'd', 'e', 'f']
6 del alist[1:5]
7 print(alist)
8
```

```
['one', 'three']
['a', 'f']
```

Activity: 1 -- ActiveCode (ac8\_2\_1)

As you might expect, `del` handles negative indices and causes a runtime error if the index is out of range. In addition, you can use a slice as an index for `del`. As usual, slices select all the elements up to, but not including, the second index.

You have attempted 2 of 1 activities on this page

[9.2. Mutability">](#)[9.4. Objects and References">](#)[9.2. Mutability">](#)

✓ Completed. Well Done!

[9.4. Objects and References">Next Section - 9.4. Objects and References](#)