

Copying data All Versions

Examples

Copy a dictionary

A dictionary object has the method `copy`. It performs a shallow copy of the dictionary.

```
>>> d1 = {'1': []}
>>> d2 = d1.copy()
>>> d1 is d2
False
>>> d1[1] is d2[1]
True
```

Copy a set

Sets also have a `copy` method. You can use this method to perform a shallow copy.

```
>>> s1 = {}
>>> s2 = s1.copy()
>>> s1 is s2
False
>>> s2.add(3)
>>> s1
{}
>>> s2
{3, []}
```

Performing a deep copy

If you have nested lists, it is desirable to clone the nested lists as well. This action is called deep copy.

```
>>> import copy
>>> c = [[1,2]]
>>> d = copy.deepcopy(c)
>>> c is d
False
>>> c[0] is d[0]
False
```

Performing a shallow copy

A shallow copy is a copy of a collection without performing a copy of its elements.

```
>>> import copy
>>> c = [[1,2]]
>>> d = copy.copy(c)
>>> c is d
False
>>> c[0] is d[0]
True
```

Performing a shallow copy of a list

You can create shallow copies of lists using slices.

```
>>> l1 = [1,2,3]
>>> l2 = l1[:] # Perform the shallow copy.
>>> l2
[1,2,3]
>>> l1 is l2
False
```

Syntax

Parameters

Remarks