

Introduction:

In programming, when working with arrays or lists, it's crucial to understand how copying works to avoid unintended side effects. In this article, we'll explore the behavior of the copy function in Python lists and NumPy arrays, highlighting the differences and best practices for copying data structures.

Copy Function with Lists:

- Copy Function with Lists:
 - In Python lists, assignment operations like `lst1 = lst` create a reference to the original list rather than a new copy.
 - Modifying either the original list or the copied list affects both variables.
 - To create an independent copy, we use the `copy()` method.

```
lst = [1,2,3,4,5]
```

```
print(lst)
```

```
lst1 = lst
```

```
print(lst1)
```

```
print('-'*10)
```

```
lst[0] = 5
```

```
print(lst)
```

```
print(lst1)
```

```
print('-'*10)
```

```
lst1[0] = 4
```

```
print(lst)
```

```
print(lst1)
```

```
print()
```

```
print('*'*20)
```

```
print()
```

```
lst = [1,2,3,4,5]
```

```
print(lst)
```

```
lst1 = lst.copy()
```

```
print(lst1)
print('-'*10)
```

```
lst[0] = 5
print(lst)
print(lst1)
print('-'*10)
lst[0] = 1
```

```
lst1[0] = 5
print(lst)
print(lst1)
```

Output

```
[1, 2, 3, 4, 5]
```

```
[1, 2, 3, 4, 5]
```

```
-----
```

```
[5, 2, 3, 4, 5]
```

```
[5, 2, 3, 4, 5]
```

```
-----
```

```
[4, 2, 3, 4, 5]
```

```
[4, 2, 3, 4, 5]
```

```
*****
```

```
[1, 2, 3, 4, 5]
```

```
[1, 2, 3, 4, 5]
```

```
-----
```

```
[5, 2, 3, 4, 5]
```

```
...
```

Copy Function with Arrays:

- Copy Function with NumPy Arrays:
 - NumPy arrays exhibit similar behavior with assignment and copying.
 - The `copy()` method is used to create a deep copy of the array, ensuring independence.

```
import numpy as np
```

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

```
arr_ = arr.copy()
```

```
print(arr)
```

```
print(arr_)
```

```
print('-'*10)
```

```
arr[0] = 5
```

```
print(arr)
```

```
print(arr_)
```

```
arr[0] = 1
```

```
print('-'*10)
```

```
arr_[0] = 5
```

```
print(arr)
```

```
print(arr_)
```

Output

```
[1 2 3 4 5]
```

```
[1 2 3 4 5]
```

```
-----
```

```
[5 2 3 4 5]
```

```
[1 2 3 4 5]
```

```
-----
```

```
[1 2 3 4 5]
```

```
[5 2 3 4 5]
```

Conclusion:

Understanding how copying works with Python lists and NumPy arrays is crucial for avoiding unintended side effects and maintaining data integrity. By using the `copy()` method, developers can create independent copies of data structures, facilitating safer and more predictable programming practices.