

Ödev1) Boş Copy Nedir? ".Copy" Nedir, yazdıktan sonra neleri değiştirir, neleri değiştirmez ?

DeepCopy Nedir? Shallow Copy Nedir? Bunlar oluşturduğumuz df'de neleri değiştirir?

Boş Copy

Yeni bir liste üretmez .copy() ile çağrılır hedef ile nesne arasında bağ kurar.

A bir liste olduğunu farz edelim a.copy() işlenen listenin bir kopyasını üretmek yerine hedef ile nesne arasında bir bağ oluşturur.

Deep Copy

Yeni bir nesne oluşturur ve orijinal öğelerde bulunan iç içe geçmiş nesnelerin kopyalarını yinelemeli olarak ekler.

copy.deepcopy(x)

Bunun nedeni, eski_liste tüm iç içe nesneleri için geçerli olan yinelemeli olarak kopyalandı.

```
import copy

old_list = [[1, 1, 1], [2, 2, 2], [3, 3, 3]]
new_list = copy.deepcopy(old_list)

old_list[1][0] = 'BB'

print("Old list:", old_list)
print("New list:", new_list)
```

Çalıştırma Kodu >>

Eski liste: [[1, 1, 1], ['BB', 2, 2], [3, 3, 3]]

Yeni liste: [[1, 1, 1], [2, 2, 2], [3, 3, 3]]

DataFrame.copy(deep=True)-> Formatında da kullanılabilir.

Shallow copy

Orijinal öğenin referansını saklayan bir nesne oluşturur. İç içe geçmiş nesneler yerine iç içe geçmiş nesnelerin referansını kopyalar. Bu, bir kopyalama işleminin yinelenmediği veya iç içe geçmiş nesnelerin kopyalarını oluşturmadığı anlamına gelir. Aynı içeriğe sahip yeni ve bağımsız bir liste oluşturulur kısaca. Sığ bir kopya kullanarak yeni yuvalanmış bir nesne ekleme:

copy.copy(x)

Programı çalıştırdığımızda şu çıktıyı verecektir:

Her iki listenin de aynı yuvalanmış nesnelerin referansını paylaşması sebebiyle ikisinede eklendi.

```
import copy

old_list = [[1, 1, 1], [2, 2, 2], [3, 3, 3]]
new_list = copy.copy(old_list)

old_list[1][1] = 'AA'

print("Old list:", old_list)
print("New list:", new_list)
```

Çalıştırma Kodu >>

Eski liste: [[1, 1, 1], [2, 'AA', 2], [3, 3, 3]]

Yeni liste: [[1, 1, 1], [2, 'AA', 2], [3, 3, 3]]

Örnekler:

```
>>> s = pd.Series([1, 2], index=["a", "b"])
>>> s
a    1
b    2
dtype: int64
```

```
>>> s_copy = s.copy()
>>> s_copy
a    1
b    2
dtype: int64
```

```
>>> s = pd.Series([1, 2], index=["a", "b"])
>>> deep = s.copy()
>>> shallow = s.copy(deep=False)
```

```
>>> s is shallow
False
>>> s.values is shallow.values and s.index is shallow.index
True
```

```
>>> s is deep
False
>>> s.values is deep.values or s.index is deep.index
False
```

```
>>> s[0] = 3
>>> shallow[1] = 4
>>> s
a    3
b    4
dtype: int64
>>> shallow
a    3
b    4
dtype: int64
>>> deep
a    1
b    2
dtype: int64
```

```
>>> s = pd.Series([[1, 2], [3, 4]])
>>> deep = s.copy()
>>> s[0][0] = 10
>>> s
0    [10, 2]
1     [3, 4]
dtype: object
>>> deep
0    [10, 2]
1     [3, 4]
dtype: object
```

Ödev2)e-01 yerine 0.78'i nasıl görebiliriz?

Formatı float olarak değiştirebiliriz.

Ödev3)"np.Log()" doğal logaritma dediğimiz şekilde mi gelir yoksa log 2 tabanında mı gelir?

np.log e tabanında log aldırır.