

Operators

Python'daki `and` operatörü, solundaki ve sağındaki ifadelerin **tek birer** `True` veya `False` değeri olmasını bekler. Ancak, `result > 73` ifadesi size tek bir `True / False` yerine, `result` dizisindeki her elemanın 73'ten büyük olup olmadığını gösteren bir **boolean dizi** (örneğin `[False, True, True, False]`) döndürür. Aynı şekilde `result <= 77` de başka bir boolean dizi döndürür.

Python'un `and` operatörü, iki boolean diziyi alıp ne yapacağını bilemez (Bütün elemanlar mı True olmalı? Herhangi biri mi True olmalı?), bu yüzden "The truth value of an array with more than one element is ambiguous" (Birden fazla eleman içeren bir dizinin doğruluk değeri belirsizdir) hatasını verir.

Doğru Yöntem:

NumPy dizilerinde eleman bazında (element-wise) mantıksal işlemler yapmak için standart `and`, `or`, `not` yerine özel operatörler kullanmalısınız:

- `&` : Eleman bazında mantıksal **AND** (ve)
- `|` : Eleman bazında mantıksal **OR** (veya)
- `~` : Eleman bazında mantıksal **NOT** (değil)

Ayrıca, operatör önceliği nedeniyle, her bir karşılaştırma ifadesini **parantez** `()` içine almanız **çok önemlidir**. Çünkü `&` operatörünün önceliği `>` ve `<=` gibi karşılaştırma operatörlerinden daha yüksektir. Parantez kullanmazsanız, NumPy işlemi yanlış yorumlayabilir.

Doğru kod şu şekilde olmalıdır:

Python

```
import numpy as np
```

```
# Örnek bir 'result' dizisi oluşturalım
result = np.array([70, 74, 75, 77, 78, 73, 80])
```

```
# Filtreleme işlemi: 73'ten büyük VE 77'ye eşit veya küçük olanlar
# Her koşul parantez içinde ve aralarında & operatörü var
ideal_results = result[(result > 73) & (result <= 77)]
```

```
print(f"Orijinal Dizi: {result}")
print(f"Filtrelenmiş Dizi (73 < eleman <= 77): {ideal_results}")
# Beklenen Çıktı: Filtrelenmiş Dizi (73 < eleman <= 77): [74 75 77]
```

Özetle: NumPy'de birden fazla koşulla filtreleme yaparken:

1. Python'un `and`, `or` yerine NumPy'nin `&`, `|` operatörlerini kullanın.
2. Her bir karşılaştırma koşulunu `()` parantez içine alın.