WHERE()

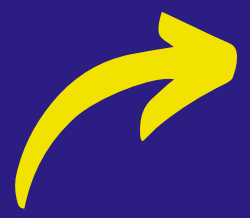
SUSHANT
KUSHWAHA

1

Today, let's look into the powerful **np.where()** function in **NumPy** and explore its versatility in data manipulation.

It is used to perform conditional operations or element wise selection on a Dataframe, let's have a look how it can be done

>>>



2

Here's a simple example to demonstrate it's usage:

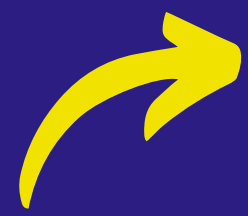
```
import pandas as pd
import numpy as np

# Create a DataFrame
data = {'Name': ['John', 'Alice', 'Bob', 'Emily', 'Jack'],
        'Age': [25, 30, 35, 40, 45]}

df = pd.DataFrame(data)

# Use np.where() to create a new column based on a condition
df['Status'] = np.where(df['Age'] > 30, 'Senior', 'Junior')

print(df)
```



3

OUTPUT;

```
   Name  Age  Status
0  John   25  Junior
1  Alice  30  Junior
2  Bob    35  Senior
3  Emily  40  Senior
4  Jack   45  Senior
```

In this example we have a DataFrame "**df**" with columns "**Name**" and "**Age**. We want to add one more column "**Status**" based on condition "**Age>30**". If condition is **True** the value in "**Status**" column should be set to "**Senior**" else "**Junior**".

4

```
np.where(df['Age'] > 30, 'Senior', 'Junior')
```

We can do this by using **np.where()** function as shown in the example. The first argument "df['Age'] > 30" is the condition. Second argument "**Senior**" is the value to be assigned if the condition is met and the third argument "**Junior**" is the value to be assigned if the given condition is not satisfied.

5

SUMMARY

np.where() can be used within pandas DataFrame operations to perform conditional operations, create new columns, update existing columns, or apply transformations based on specific conditions.

WAS THIS HELPFUL?

Share with a friend who needs it!

SUSHANT KUSHWAHA