The double brackets in the expression `df[['ID']]` are used to create a new DataFrame containing only the "ID" column from the original DataFrame `df`. It is a way to select a single column from a DataFrame and store it as a new DataFrame rather than as a Series.

Let's understand the difference between single and double brackets when accessing a column in a DataFrame:

1. Single Brackets (`df['ID']`): When you use single brackets to access a column in a DataFrame, it returns the column as a pandas Series. The Series is a one-dimensional labeled array and represents a single column of data with an index.

```python

import pandas as pd

# Sample DataFrame

data = {

'ID': [1, 2, 3],

'Name': ['Alice', 'Bob', 'Charlie'],

'Age': [25, 30, 22]

}

df = pd.DataFrame(data)

# Accessing 'ID' column as a Series

x = df['ID']

print(x)

```

Output:

```

0 1

1 2

2 3

Name: ID, dtype: int64

```

2. Double Brackets (`df[['ID']]`): When you use double brackets to access a column in a DataFrame, it returns the column as a new DataFrame. The resulting DataFrame will have a single column, just like a Series, but it preserves the column name as well.

```python

# Accessing 'ID' column as a new DataFrame

x = df[['ID']]

print(x)

```

Output:

```

ID

0 1

1 2

2 3

```

Using double brackets with the column name inside creates a new DataFrame with that single column, whereas using single brackets returns the column as a Series. The choice between single and double brackets depends on whether you want the result as a Series or a DataFrame.