

# How can I reshape a Pandas DataFrame from a long format to a wide format?

Authored by  
**stats writer**

July 2, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I reshape a Pandas DataFrame from a long format to a wide format?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=165687>

Reshaping a Pandas DataFrame from a long format to a wide format involves rearranging the structure of the data to make it easier to analyze and understand. This is done by pivoting the data so that the rows become columns and vice versa, resulting in a wider table with a clearer representation of the data. This process is useful when dealing with large datasets and allows for easier manipulation and analysis of the data. It can be achieved using built-in functions in Pandas such as "pivot()" and "melt()", or through manual manipulation of the DataFrame's structure. Ultimately, reshaping a DataFrame from long to wide format can improve data organization and aid in data analysis.

## **Pandas: Reshape DataFrame from Long to Wide**

**You can use the following basic syntax to convert a pandas DataFrame from a long format to a wide format:**

```
df = pd.pivot(df, index='col1', columns='col2', values='col3')
```

**In this scenario, col1 will become the index, col2 will become the columns, and col3 will be used as the values inside the DataFrame.**

**The following example shows how to use this syntax in practice.**

**Example: Reshape Pandas DataFrame from Long to Wide**

**Suppose we have the following pandas DataFrame in a long format:**

```
import pandas as pd

#create DataFrame in long format
df = pd.DataFrame({'team': ,
'player': ,
'points': })
```

```
#view DataFrame
df
```

```
team player points
0 A 1 11
1 A 2 8
2 A 3 10
3 A 4 6
4 B 1 12
5 B 2 5
6 B 3 9
7 B 4 4
```

We can use the following syntax to reshape this DataFrame from a long format to a wide format:

```
#reshape DataFrame from long format to wide format
df = pd.pivot(df, index='team', columns='player',
```

```
values='points')
```

```
#view updated DataFrame
```

```
df
```

```
player 1 2 3 4
```

```
team
```

```
A 11 8 10 6
```

```
B 12 5 9 4
```

The DataFrame is now in a wide format.

We used 'team' as the index column, 'player' as the columns, and 'points' as the values inside of the DataFrame.

Note that we could instead use 'player' as the index column and 'team' as the columns if we'd like:

```
#reshape DataFrame from long format to wide format
```

```
df = pd.pivot(df, index='player', columns='team',  
values='points')
```

```
#view updated DataFrame
```

```
df
```

**team A B**

**player**

**1 11 12**

**2 8 5**

**3 10 9**

**4 6 4**

**This DataFrame is also in a wide format.**

**Note: You can find the complete documentation for the pandas pivot() function .**

**Additional Resources**

**The following tutorials explain how to perform other common operations in Python:**