

# How can I merge two Pandas DataFrames that have different column names?

Authored by  
**stats writer**

June 28, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I merge two Pandas DataFrames that have different column names?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=156437>

To merge two Pandas DataFrames with different column names, you can use the "merge" function and specify the columns to be used as the merging keys. This allows you to combine the data from both DataFrames based on a common column or multiple columns. Additionally, you can also use the "concat" function to simply append one DataFrame to another, regardless of the column names. It is important to carefully consider the column names and their data types to ensure a successful merge and avoid any data loss.

## Pandas: Merge Two DataFrames with Different Column Names

You can use the following basic syntax to merge two pandas DataFrames with different column names:

```
pd.merge(df1, df2, left_on='left_column_name', right_on='right_column_name')
```

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

Example: Merge Two Pandas DataFrames with Different Column Names

Suppose we have the following two pandas DataFrames:

```
import pandas as pd
```

```
#create first DataFrame
```

```
df1 = pd.DataFrame({'team': ,
```

```
'points': })
```

```
#view DataFrame
```

```
print(df1)
```

```
team points
```

```
0 A 4
```

```
1 B 4
```

```
2 C 6
```

```
3 D 8
```

```
4 E 9
```

```
5 F 5
```

```
#create second DataFrame
```

```
df2 = pd.DataFrame({'team_name': ,  
'rebounds': })
```

```
#view DataFrame
```

```
print(df2)
```

```
team_name rebounds
```

```
0 A 12
```

```
1 B 7
```

```
2 C 8
```

```
3 D 8
```

4 E 5

5 F 11

We can use the following syntax to perform an inner join, using the team column in the first DataFrame and the team\_name column in the second DataFrame:

```
#merge DataFrames
```

```
df3 = pd.merge(df1, df2, left_on='team',  
right_on='team_name')
```

```
#view result
```

```
print(df3)
```

```
team points team_name rebounds
```

```
0 A 4 A 12
```

```
1 B 4 B 7
```

```
2 C 6 C 8
```

```
3 D 8 D 8
```

```
4 E 9 E 5
```

```
5 F 5 F 11
```

Notice that we're able to successfully perform an inner join even though the two column names that we used for the join were different in each DataFrame.

Note that we can also use the following code to drop the `team_name` column from the final merged DataFrame since the values in this column match those in the `team` column:

```
#drop team_name column
```

```
df3.drop('team_name', axis=1, inplace=True)
```

```
#view updated DataFrame
```

```
print(df3)
```

```
team points rebounds
```

```
0 A 4 12
```

```
1 B 4 7
```

```
2 C 6 8
```

```
3 D 8 8
```

```
4 E 9 5
```

```
5 F 5 11
```

Notice that the `team_name` column has been dropped from the DataFrame.

**Related:**

**Additional Resources**

**The following tutorials explain how to perform other common tasks in pandas:**

ARABPSYCHOLOGY.COM