

# How can I reindex rows in Pandas starting from 1 instead of 0?

Authored by  
**stats writer**

June 24, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I reindex rows in Pandas starting from 1 instead of 0?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=150821>

Reindexing rows in Pandas refers to the process of rearranging the index labels of a DataFrame in a specific order. By default, the index labels start from 0. However, if you wish to change the starting index from 0 to 1, you can use the "reindex" function and specify the parameter "index" as a list of numbers starting from 1. This will result in the DataFrame's rows being reindexed starting from 1 instead of 0, allowing for a more organized and intuitive representation of the data.

## Pandas: Reindex Rows Starting From 1

You can use the following basic syntax to reindex the rows of a pandas DataFrame starting from 1 instead of 0:

```
import pandas as pd
import numpy as np
df.index = np.arange(1, len(df) + 1)
```

The NumPy `arange()` function creates an array starting from 1 that increases by increments of 1 until the length of the entire DataFrame plus 1.

This array is then used as the index of the DataFrame.

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

### Example: Reindex Rows of Pandas DataFrame Starting From 1

Suppose we have the following pandas DataFrame that contains information about various basketball players:

```
import pandas as pd

#create DataFrame
df = pd.DataFrame({'team': ,
'points': ,
'assists': ,
'rebounds': })

#view DataFrame
print(df)

team points assists rebounds
0 A 18 5 11
1 B 22 7 8
2 C 19 7 10
3 D 14 9 6
4 E 14 12 6
5 F 11 9 5
6 G 20 9 9
7 H 28 4 12
```

Notice that the index currently ranges from 0 to 7.

To reindex the values in the index to column to instead start from 1, we can use the following syntax:

```
import numpy as np
#reindex values in index to start from 1
df.index = np.arange(1, len(df) + 1)

#view updated DataFrameprint(df)
```

**team points assists rebounds**

**1 A 18 5 11**

**2 B 22 7 8**

**3 C 19 7 10**

**4 D 14 9 6**

**5 E 14 12 6**

**6 F 11 9 5**

**7 G 20 9 9**

**8 H 28 4 12**

**Notice that the values in the index now start from 1.**

**Note #1:** The benefit of using the `len()` function to find the number of rows in the DataFrame is that we don't need to know how many rows are in the DataFrame before creating the new array of index values.

**Note #2:** You can find the complete documentation for the NumPy `arange()` function .

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

ARABPSYCHOLOGY.COM