

# How to replace empty strings with NaN in Pandas?

Authored by  
**stats writer**

November 29, 2025

## RECOMMENDED CITATION

stats writer (2025). *How to replace empty strings with NaN in Pandas?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=101932>

In Pandas, you can replace empty strings with NaN values by using the `replace()` function. Pass in a dictionary containing the empty strings as keys and NaN as values. Then, the `replace()` function will replace all the empty strings with NaN values in the DataFrame.

You can use the following syntax to replace empty strings with NaN values in pandas:

```
df = df.replace(r'^s*$', np.nan, regex=True)
```

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

### Example: Replace Empty Strings with NaN

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

```
import pandas as pd
```

```
#create DataFrame
```

```
df = pd.DataFrame({'team': ,  
'position': ,  
'points': ,  
'rebounds': })
```

```
#view DataFrame
```

```
df
```

```
team position points rebounds
```

```
0 A 5 11
```

```
1 B G 7 8
```

```
2 G 7 10
```

```
3 D F 9 6
```

```
4 E F 12 6
```

```
5 9 5
```

```
6 G C 9 9
```

```
7 H C 4 12
```

Notice that there are several empty strings in both the **team** and **position** columns.

We can use the following syntax to replace these empty strings with NaN values:

```
import numpy as np
```

```
#replace empty values with NaN  
df = df.replace(r'^s*$', np.nan, regex=True)
```

```
#view updated DataFrame  
df
```

```
team position points rebounds
```

```
0 A NaN 5 11
```

```
1 B G 7 8
```

```
2 NaN G 7 10
```

```
3 D F 9 6
```

```
4 E F 12 6
```

```
5 NaN NaN 9 5
```

```
6 G C 9 9
```

```
7 H C 4 127
```

Notice that each of the empty strings have been replaced with NaN.

**Note:** You can find the complete documentation for the **replace** function in pandas .