

# How can I remove specific characters from strings in Pandas?

Authored by  
**stats writer**

June 25, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I remove specific characters from strings in Pandas?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=152530>

The process of removing specific characters from strings in Pandas involves using the "str.replace()" function, which allows for the replacement of certain characters with either a new character or an empty string. This function can be applied to a specific column or row in a Pandas dataframe, allowing for the removal of unwanted characters and the creation of cleaner and more organized data. By specifying the target character or characters to be replaced, the function allows for a quick and efficient way to manipulate strings within a Pandas dataframe.

## **Pandas: Remove Specific Characters from Strings**

**You can use the following methods to remove specific characters from strings in a column in a pandas DataFrame:**

### **Method 1: Remove Specific Characters from Strings**

```
df = df.str.replace('this_string', '')
```

### **Method 2: Remove All Letters from Strings**

```
df = df.str.replace('D', '', regex=True)
```

### **Method 3: Remove All Numbers from Strings**

```
df = df.str.replace('d+', '', regex=True)
```

**The following examples show how to use each method in practice with the following pandas DataFrame:**

```
import pandas as pd

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

#view DataFrame
print(df)

team points
0 Mavs2 12
1 Nets44 15
2 Kings33 22
3 Cavs90 29
4 Heat576 24
```

#### Example 1: Remove Specific Characters from Strings

We can use the following syntax to remove 'avs' from each string in the team column:

```
#remove 'avs' from strings in team column
df = df.str.replace('avs', "")

#view updated DataFrame
print(df)
```

**team points**

**0 M2 12**

**1 Nets44 15**

**2 Kings33 22**

**3 C90 29**

**4 Heat576 24**

**Notice that 'avs' was removed from the rows that contained 'Mavs' and 'Cavs' in the team column.**

**Example 2: Remove All Letters from Strings**

**We can use the following syntax to remove all letters from each string in the team column:**

```
#remove letters from strings in team column
```

```
df = df.str.replace('D', "", regex=True)
```

```
#view updated DataFrame
```

```
print(df)
```

**team points**

**0 2 12**

**1 44 15**

**2 33 22**

**3 90 29**

**4 576 24**

**Notice that all letters have been removed from each string in the team column.**

**Example 3: Remove All Numbers from Strings**

**We can use the following syntax to remove all numbers from each string in the team column:**

```
#remove numbers from strings in team column
```

```
df = df.str.replace('d+', "", regex=True)
```

```
#view updated DataFrame
```

```
print(df)
```

```
team points
```

```
0 Mavs 12
```

```
1 Nets 15
```

```
2 Kings 22
```

```
3 Cavs 29
```

```
4 Heat 24
```

**Notice that all numbers have been removed from each string in the team column.**

**Only the letters remain.**

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

**[How to Replace NaN Values with Zeros in Pandas](#)**

ARABPSYCHOLOGY.COM