

How can I create an empty DataFrame in Pandas with specified column names?

Authored by
stats writer

July 2, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I create an empty DataFrame in Pandas with specified column names?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=165808>

To create an empty DataFrame in Pandas with specified column names, you can use the `pd.DataFrame()` function and pass in a list of the desired column names as the `columns` parameter. This will create a new DataFrame with no rows and the specified column names, allowing you to easily add data to it later.

Pandas: Create Empty DataFrame with Column Names

You can use the following basic syntax to create an empty pandas DataFrame with specific column names:

```
df = pd.DataFrame(columns=)
```

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

Example 1: Create DataFrame with Column Names & No Rows

The following code shows how to create a pandas DataFrame with specific column names and no rows:

```
import pandas as pd

#create DataFrame
df = pd.DataFrame(columns=)

#view DataFrame
df
```

ABCD E

We can use shape to get the size of the DataFrame:

```
#display shape of DataFrame
```

```
df.shape
```

```
(0, 5)
```

This tells us that the DataFrame has 0 rows and 5 columns.

We can also use list() to get a list of the column names:

```
#display list of column names
```

```
list(df)
```

Example 2: Create DataFrame with Column Names & Specific Number of Rows

The following code shows how to create a pandas DataFrame with specific column names and a specific number of rows:

```
import pandas as pd
```

```
#create DataFrame  
df = pd.DataFrame(columns=,  
index=range(1, 10))  
#view DataFrame  
df
```

```
A B C D E  
1 NaN NaN NaN NaN NaN  
2 NaN NaN NaN NaN NaN  
3 NaN NaN NaN NaN NaN  
4 NaN NaN NaN NaN NaN  
5 NaN NaN NaN NaN NaN  
6 NaN NaN NaN NaN NaN  
7 NaN NaN NaN NaN NaN  
8 NaN NaN NaN NaN NaN  
9 NaN NaN NaN NaN NaN
```

Notice that every value in the DataFrame is filled with a NaN value.

Once again, we can use shape to get the size of the DataFrame:

```
#display shape of DataFrame
```

df.shape

(9, 5)

Additional Resources

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

ARABPSYCHOLOGY.COM