

# How to Create New Column Using Multiple If Else Conditions in Pandas

Authored by  
**stats writer**

November 22, 2025

## RECOMMENDED CITATION

stats writer (2025). *How to Create New Column Using Multiple If Else Conditions in Pandas*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=100085>

Pandas can be used to create new columns using multiple if else conditions by using the 'np.where()' function. This function takes three arguments: a boolean condition to check, the value if the condition is True, and the value if the condition is False. The output of this function will be the new column that is created. This function can be used to create multiple columns by adding more conditions and values in the arguments. With the help of this function, we can create new columns in Pandas dataframe with multiple if else conditions.

You can use the following syntax to create a new column in a pandas DataFrame using multiple if else conditions:

```
#define conditions
```

```
conditions = == 'A' & (df < 20),
```

```
(df == 'A') & (df >= 20),
```

```
(df == 'B') & (df < 20),
```

```
(df == 'B') & (df >= 20)
```

```
]
```

```
#define results
```

```
results =
```

```
#create new column based on conditions in column1 and column2
```

```
df = np.select(conditions, results)
```

This particular example creates a column called **new\_column** whose values are based on the values in **column1** and **column2** in the DataFrame.

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

## Example: Create New Column Using Multiple If Else Conditions in Pandas

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': })
```

```
#view DataFrame
```

```
print(df)
```

```
team points
```

```
0 A 15
```

```
1 A 18
```

```
2 A 22
```

```
3 A 24
```

```
4 B 12
```

```
5 B 17
```

```
6 B 20
```

```
7 B 28
```

Now suppose we would like to create a new column called **class** that classifies each player into one of the following four groups:

**Bad\_A** if team is A and points < 20

**Good\_A** if team is A and points ≥ 20

**Bad\_B** if team is B and points < 20

**Good\_B** if team is B and points ≥ 20

We can use the following syntax to do so:

```
import numpy as np
```

```
#define conditions
```

```
conditions = == 'A' & (df < 20),
```

```
(df == 'A' & (df >= 20),
```

```
(df == 'B' & (df < 20),
```

```
(df == 'B' & (df >= 20)
```

```
]
```

```
#define results
```

```
results =
```

```
#create new column based on conditions in column1 and column2
```

```
df = np.select(conditions, results)
```

```
#view updated DataFrame
```

```
print(df)
```

```
team points class
```

```
0 A 15 Bad_A
```

1 A 18 Bad\_A  
2 A 22 Good\_A  
3 A 24 Good\_A  
4 B 12 Bad\_B  
5 B 17 Bad\_B  
6 B 20 Good\_B  
7 B 28 Good\_B

The new column called **class** displays the classification of each player based on the values in the **team** and **points** columns.

**Note:** You can find the complete documentation for the NumPy **select()** function .

ARABPSYCHOLOGY.COM