In this lesson, we will append two Pandas series using the **append()** method. This will append two series. With that, the **ignore_index** parameter of the append() will allow you to ignore or consider the index. If **ignore_index** is set to True original indexes are ignored and rep by 0, 1, 2, etc. in the output. The default is **False**.

Note: The append() method deprecated since version Pandas 1.4.0.

We will see two examples:

1. Append two Pandas series considering the original index
2. Append two Pandas series ignoring the original index

Append two Pandas series considering the original index

Discover related topics

Arrays

Apply

To append two series, use the **append()** method. The **ignore_index** parameter is by default set to **False.** This will keep both series indexes alive even after the append. Let us see an example:


```
import pandas as pd

# Data to be stored in the Pandas Series
data1 = [10, 20, 40, 80, 100]
data2 = [150, 200]

# Create two Series using the Series() me
series1 = pd.Series(data1, index = ["RowA
series2 = pd.Series(data2, index = ["RowF

# Display the Series
print("Series1 (with custom index labels)
print("\nSeries2 (with custom index label

# Append
# The ignore_index parameter is by defaul
result = series1.append(series2, ignore_i

# Print the result
print("\nResult after appending (consider
```

Output

```
Series1 (with custom index labels):
RowA     10
RowB     20
RowC     40
RowD     80
RowE    100
dtype: int64

Series2 (with custom index labels):
RowF    150
RowG    200
dtype: int64

Result after appending (considering origi
```

Append two Pandas series ignoring the original index

To append two series, use the **append()** method. For ignoring the original indexes and replacing them with 0, 1, 2, etc, set the **ignore_index** to **True** as discuss above.

Menu

Let us see an example:

```
import pandas as pd

# Data to be stored in the Pandas Series
data1 = [10, 20, 40, 80, 100]
data2 = [150, 200]

# Create two Series using the Series() me
series1 = pd.Series(data1, index = ["RowA
series2 = pd.Series(data2, index = ["RowF

# Display the Series
print("Series1 (with custom index labels)
print("\nSeries2 (with custom index label

# Append
# The ignore_index parameter is set to Tr
result = series1.append(series2, ignore_i

# Print the result
print("\nResult after appending (ignoring
```

To combine two Pandas series into one in Python, use the **combine()** method. It uses a specific function for the decision, mentioned by the user as a parameter of the combine() method.

We will see an example that will fetch the largest values from both series. Each element of both the series will be compared one by one. Let us see the example:

```
import pandas as pd

# Data to be stored in the Pandas Series
data1 = [10, 20, 40, 80, 100]
data2 = [25, 5, 75, 95, 45]
```

```
import pandas as pd

# Data to be stored in the Pandas Series
data1 = [10, 20, 40, 80, 100]
data2 = [25, 5, 75, 95, 45]

# Create a Series using the Series() meth
series1 = pd.Series(data1)
series2 = pd.Series(data2)

# Display the Series
print("Before combining the series:\
print("Series1: \n",series1)
print("Series2: \n",series2)

def demo(x1, x2) :
  if (x1 > x2):
    return x1
  else:
    return x2

# Combine two series into one
# The function returns the largest value
res = series1.combine(series2, demo)

# Display the result
print("\nAfter combining into one:\n",res
```


Output

```
Before combining the series:

Series1:
0     10
1     20
2     40
3     80
4    100
dtype: int64
Series2:
0    25
1     5
2    75
3    95
4    45
dtype: int64

After combining into one:
 0     25
```