

# PANDAS

DATA:Series

1. Write a Pandas program to compare the elements of the two Pandas Series.

```
import pandas as pd
ds1 = pd.Series([2, 4, 6, 8, 10])
ds2 = pd.Series([1, 3, 5, 7, 10])
print("Series1:")
print(ds1)
print("Series2:")
print(ds2)
print("Compare the elements of the said Series:")
print("Equals:")
print(ds1 == ds2)
print("Greater than:")
print(ds1 > ds2)
print("Less than:")
print(ds1 < ds2)
```

2. Write a Pandas program to convert a dictionary to a Pandas series.

Sample dictionary: d1 = {'a': 100, 'b': 200, 'c':300, 'd':400, 'e':800}

```
import pandas as pd
d1 = {'a': 100, 'b': 200, 'c':300, 'd':400, 'e':800}
print("Original dictionary:")
print(d1)
new_series = pd.Series(d1)
print("Converted series:")
print(new_series)
```

3. Write a Python Pandas program to convert the first column of a DataFrame as a Series.

```
import pandas as pd
d = {'col1': [1, 2, 3, 4, 7, 11], 'col2': [4, 5, 6, 9, 5, 0], 'col3': [7, 5, 8, 12, 1, 11]}
df = pd.DataFrame(data=d)
print("Original DataFrame")
print(df)
s1 = df.ix[:,0]
print("\n1st column as a Series:")
print(s1)
print(type(s1))
```

4. Write a Pandas program to sort a given Series

```
import pandas as pd
s = pd.Series(['100', '200', 'python', '300.12', '400'])
print("Original Data Series:")
print(s)
new_s = pd.Series(s).sort_values()
print(new_s)
```

5. Write a Pandas program to add some data to an existing Series.

```
import pandas as pd
s = pd.Series(['100', '200', 'python', '300.12', '400'])
print("Original Data Series:")
print(s)
print("\nData Series after adding some data:")
new_s = s.append(pd.Series(['500', 'php']))
print(new_s)
```

6. Write a Pandas program to change the order of index of a given series.

```
import pandas as pd
s = pd.Series(data = [1,2,3,4,5], index = ['A', 'B', 'C', 'D', 'E'])
print("Original Data Series:")
print(s)
s = s.reindex(index = ['B', 'A', 'C', 'D', 'E'])
print("Data Series after changing the order of index:")
print(s)
```

7. Write a Pandas program to get the items of a given series not present in another given series

```
import pandas as pd
sr1 = pd.Series([1, 2, 3, 4, 5])
sr2 = pd.Series([2, 4, 6, 8, 10])
print("Original Series:")
print("sr1:")
print(sr1)
print("sr2:")
print(sr2)
print("\nItems of sr1 not present in sr2:")
result = sr1[~sr1.isin(sr2)]
print(result)
```

8. Write a Pandas program to get the items which are not common of two given series.

```
import pandas as pd
import numpy as np
sr1 = pd.Series([1, 2, 3, 4, 5])
sr2 = pd.Series([2, 4, 6, 8, 10])
print("Original Series:")
print("sr1:")
print(sr1)
print("sr2:")
print(sr2)
print("\nItems of a given series not present in another given series:")
sr11 = pd.Series(np.union1d(sr1, sr2))
sr22 = pd.Series(np.intersect1d(sr1, sr2))
result = sr11[~sr11.isin(sr22)]
print(result)
```

9. Write a Pandas program to calculate the frequency counts of each unique value of a given series.

```
import pandas as pd
import numpy as np
num_series = pd.Series(np.take(list('0123456789'),
np.random.randint(10, size=40)))
print("Original Series:")
print(num_series)
print("Frequency of each unique value of the said series.")
result = num_series.value_counts()
print(result)
```

10. Write a Pandas program to get the positions of items of a given series in another given series.

```
import pandas as pd
series1 = pd.Series([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
series2 = pd.Series([1, 3, 5, 7, 10])
print("Original Series:")
print(series1)
print(series2)
result = [pd.Index(series1).get_loc(i) for i in series2]
print("Positions of items of series2 in series1:")
print(result)
```

11. Write a Pandas program to calculate the number of characters in each word in a given series.

```
import pandas as pd
series1 = pd.Series(['Php', 'Python', 'Java', 'C#'])
print("Original Series:")
print(series1)
result = series1.map(lambda x: len(x))
print("\nNumber of characters in each word in the said series:")
print(result)
```

12. Write a Pandas program to convert given series into a dataframe with its index as another column on the dataframe.

```
import numpy as np
import pandas as pd
char_list = list('ABCDEFGHJKLMNOP')
num_arr = np.arange(8)
num_dict = dict(zip(char_list, num_arr))
num_ser = pd.Series(num_dict)
df = num_ser.to_frame().reset_index()
print(df.head())
```

## Pandas Joining and merging DataFrame

---

1. Write a Pandas program to join the two given dataframes along rows and assign all data.

### Test Data:

```
student_data1:
  student_id      name  marks
0          S1  Danniella Fenton    200
1          S2    Ryder Storey    210
2          S3    Bryce Jensen    190
3          S4      Ed Bernal    222
4          S5    Kwame Morin    199
student_data2:
  student_id      name  marks
0          S4  Scarlett Fisher    201
1          S5   Carla Williamson    200
2          S6    Dante Morse    198
```

3	S7	Kaiser William	219
4	S8	Madeeha Preston	201

```
import pandas as pd

student_data1 = pd.DataFrame({
    'student_id': ['S1', 'S2', 'S3', 'S4', 'S5'],
    'name': ['Danniella Fenton', 'Ryder Storey', 'Bryce Jensen',
            'Ed Bernal', 'Kwame Morin'],
    'marks': [200, 210, 190, 222, 199]})

student_data2 = pd.DataFrame({
    'student_id': ['S4', 'S5', 'S6', 'S7', 'S8'],
    'name': ['Scarlette Fisher', 'Carla Williamson', 'Dante
Morse', 'Kaiser William', 'Madeeha Preston'],
    'marks': [201, 200, 198, 219, 201]})

print("Original DataFrames:")
print(student_data1)
print("-----")
print(student_data2)
print("\nJoin the said two dataframes along rows:")
result_data = pd.concat([student_data1, student_data2])
print(result_data)
```

2. Write a Pandas program to join the two given dataframes along columns and assign all data

```
result_data = pd.concat([student_data1, student_data2], axis = 1)
print(result_data)
```

3. Write a Pandas program to append rows to an existing DataFrame and display the combined data.

### Test Data:

	student_id	name	marks
0	S1	Danniella Fenton	200
1	S2	Ryder Storey	210
2	S3	Bryce Jensen	190

```

3          S4          Ed Bernal      222
4          S5          Kwame Morin    199
New Row(s)
student_id          S6
name          Scarlett Fisher
marks          205
dtype: object

```

```

import pandas as pd
student_data1 = pd.DataFrame({
    'student_id': ['S1', 'S2', 'S3', 'S4', 'S5'],
    'name': ['Danniella Fenton', 'Ryder Storey', 'Bryce Jensen',
    'Ed Bernal', 'Kwame Morin'],
    'marks': [200, 210, 190, 222, 199]})

s6 = pd.Series(['S6', 'Scarlette Fisher', 205], index=['student_id',
    'name', 'marks'])
print("Original DataFrames:")
print(student_data1)
print("\nNew Row(s)")
print(s6)
combined_data = student_data1.append(s6, ignore_index = True)
print("\nCombined Data:")
print(combined_data)

```

4. Write a Pandas program to append a list of dictionaries or series to a existing DataFrame and display the combined data.

### Test Data:

```

student_id          name      marks
0          S1  Danniella Fenton    200
1          S2      Ryder Storey    210
2          S3      Bryce Jensen    190
3          S4          Ed Bernal    222
4          S5      Kwame Morin    199
Dictionary:
student_id          S6
name          Scarlett Fisher
marks          205
dtype: object

```

```

import pandas as pd

```

```

student_data1 = pd.DataFrame({
    'student_id': ['S1', 'S2', 'S3', 'S4', 'S5'],
    'name': ['Danniella Fenton', 'Ryder Storey', 'Bryce Jensen',
'Ed Bernal', 'Kwame Morin'],
    'marks': [200, 210, 190, 222, 199]})

s6 = pd.Series(['S6', 'Scarlette Fisher', 205], index=['student_id',
'name', 'marks'])

dicts = [{'student_id': 'S6', 'name': 'Scarlette Fisher', 'marks':
203},
        {'student_id': 'S7', 'name': 'Bryce Jensen', 'marks': 207}]

print("Original DataFrames:")
print(student_data1)
print("\nDictionary:")
print(s6)
combined_data = student_data1.append(dicts, ignore_index=True,
sort=False)
print("\nCombined Data:")
print(combined_data)

```

5. Write a Pandas program to join the two given dataframes along rows and merge with another dataframe along the common column id.

### Test Data:

```

student_data1:
  student_id      name  marks
0         S1  Danniella Fenton   200
1         S2    Ryder Storey   210
2         S3    Bryce Jensen   190
3         S4      Ed Bernal   222
4         S5    Kwame Morin   199
student_data2:
  student_id      name  marks
0         S4  Scarlette Fisher   201
1         S5   Carla Williamson   200
2         S6    Dante Morse   198
3         S7   Kaiser William   219
4         S8   Madeeha Preston   201
exam_data:
  student_id  exam_id

```

0	S1	23
1	S2	45
2	S3	12
3	S4	67
4	S5	21
5	S7	55
6	S8	33
7	S9	14
8	S10	56
9	S11	83
10	S12	88
11	S13	12

```
import pandas as pd
student_data1 = pd.DataFrame({
    'student_id': ['S1', 'S2', 'S3', 'S4', 'S5'],
    'name': ['Danniella Fenton', 'Ryder Storey', 'Bryce Jensen',
'Ed Bernal', 'Kwame Morin'],
    'marks': [200, 210, 190, 222, 199]})

student_data2 = pd.DataFrame({
    'student_id': ['S4', 'S5', 'S6', 'S7', 'S8'],
    'name': ['Scarlette Fisher', 'Carla Williamson', 'Dante
Morse', 'Kaiser William', 'Madeeha Preston'],
    'marks': [201, 200, 198, 219, 201]})

exam_data = pd.DataFrame({
    'student_id': ['S1', 'S2', 'S3', 'S4', 'S5', 'S7', 'S8', 'S9',
'S10', 'S11', 'S12', 'S13'],
    'exam_id': [23, 45, 12, 67, 21, 55, 33, 14, 56, 83, 88, 12]})

print("Original DataFrames:")
print(student_data1)
print(student_data2)
print(exam_data)

print("\nJoin first two said dataframes along rows:")
result_data = pd.concat([student_data1, student_data2])
print(result_data)

print("\nNow join the said result_data and df_exam_data along
student_id:")
final_merged_data = pd.merge(result_data, exam_data, on='student_id')
print(final_merged_data)
```



6. Write a Pandas program to join the two dataframes using the common column of both dataframes.

### Test Data:

```
student_data1:
  student_id      name  marks
0         S1  Danniella Fenton    200
1         S2    Ryder Storey    210
2         S3    Bryce Jensen    190
3         S4      Ed Bernal    222
4         S5    Kwame Morin    199
student_data2:
  student_id      name  marks
0         S4  Scarlettte Fisher    201
1         S5  Carla Williamson    200
2         S6    Dante Morse    198
3         S7  Kaiser William    219
4         S8  Madeeha Preston    201
```

```
import pandas as pd
student_data1 = pd.DataFrame({
    'student_id': ['S1', 'S2', 'S3', 'S4', 'S5'],
    'name': ['Danniella Fenton', 'Ryder Storey', 'Bryce Jensen',
    'Ed Bernal', 'Kwame Morin'],
    'marks': [200, 210, 190, 222, 199]})

student_data2 = pd.DataFrame({
    'student_id': ['S4', 'S5', 'S6', 'S7', 'S8'],
    'name': ['Scarlettte Fisher', 'Carla Williamson', 'Dante
Morse', 'Kaiser William', 'Madeeha Preston'],
    'marks': [201, 200, 198, 219, 201]})

print("Original DataFrames:")
print(student_data1)
print(student_data2)
merged_data = pd.merge(student_data1, student_data2, on='student_id',
how='inner')
print("Merged data (inner join):")
print(merged_data)
```

7. Write a Pandas program to join the two dataframes with matching records from both sides where available.

## Test Data:

```
student_data1:
  student_id      name  marks
0          S1  Danniella Fenton    200
1          S2    Ryder Storey    210
2          S3    Bryce Jensen    190
3          S4      Ed Bernal    222
4          S5    Kwame Morin    199
```

```
student_data2:
  student_id      name  marks
0          S4  Scarlettte Fisher    201
1          S5    Carla Williamson    200
2          S6      Dante Morse    198
3          S7    Kaiser William    219
4          S8    Madeeha Preston    201
```

```
import pandas as pd
student_data1 = pd.DataFrame({
    'student_id': ['S1', 'S2', 'S3', 'S4', 'S5'],
    'name': ['Danniella Fenton', 'Ryder Storey', 'Bryce Jensen',
            'Ed Bernal', 'Kwame Morin'],
    'marks': [200, 210, 190, 222, 199]})

student_data2 = pd.DataFrame({
    'student_id': ['S4', 'S5', 'S6', 'S7', 'S8'],
    'name': ['Scarlettte Fisher', 'Carla Williamson', 'Dante
Morse', 'Kaiser William', 'Madeeha Preston'],
    'marks': [201, 200, 198, 219, 201]})

print("Original DataFrames:")
print(student_data1)
print(student_data2)
merged_data = pd.merge(student_data1, student_data2, on='student_id',
                        how='outer')
print("Merged data (outer join):")
print(merged_data)
```

8. Write a Pandas program to join (left join) the two dataframes using keys from left dataframe only.

## Test Data:

```
data1:
```

	key1	key2	P	Q
0	K0	K0	P0	Q0
1	K0	K1	P1	Q1
2	K1	K0	P2	Q2
3	K2	K1	P3	Q3

data2:

	key1	key2	R	S
0	K0	K0	R0	S0
1	K1	K0	R1	S1
2	K1	K0	R2	S2
3	K2	K0	R3	S3

```
import pandas as pd
data1 = pd.DataFrame({'key1': ['K0', 'K0', 'K1', 'K2'],
                      'key2': ['K0', 'K1', 'K0', 'K1'],
                      'P': ['P0', 'P1', 'P2', 'P3'],
                      'Q': ['Q0', 'Q1', 'Q2', 'Q3']})
data2 = pd.DataFrame({'key1': ['K0', 'K1', 'K1', 'K2'],
                      'key2': ['K0', 'K0', 'K0', 'K0'],
                      'R': ['R0', 'R1', 'R2', 'R3'],
                      'S': ['S0', 'S1', 'S2', 'S3']})
print("Original DataFrames:")
print(data1)
print("-----")
print(data2)
print("\nMerged Data (keys from data1):")
merged_data = pd.merge(data1, data2, how='left', on=['key1', 'key2'])
print(merged_data)
print("\nMerged Data (keys from data2):")
merged_data = pd.merge(data2, data1, how='left', on=['key1', 'key2'])
print(merged_data)
```

9. Write a Pandas program to join two dataframes using keys from right dataframe only.

### Test Data:

data1:

	key1	key2	P	Q
0	K0	K0	P0	Q0
1	K0	K1	P1	Q1
2	K1	K0	P2	Q2
3	K2	K1	P3	Q3

data2:

	key1	key2	R	S
--	------	------	---	---

0	K0	K0	R0	S0
1	K1	K0	R1	S1
2	K1	K0	R2	S2
3	K2	K0	R3	S3

```
import pandas as pd
data1 = pd.DataFrame({'key1': ['K0', 'K0', 'K1', 'K2'],
                      'key2': ['K0', 'K1', 'K0', 'K1'],
                      'P': ['P0', 'P1', 'P2', 'P3'],
                      'Q': ['Q0', 'Q1', 'Q2', 'Q3']})
data2 = pd.DataFrame({'key1': ['K0', 'K1', 'K1', 'K2'],
                      'key2': ['K0', 'K0', 'K0', 'K0'],
                      'R': ['R0', 'R1', 'R2', 'R3'],
                      'S': ['S0', 'S1', 'S2', 'S3']})
print("Original DataFrames:")
print(data1)
print("-----")
print(data2)
print("\nMerged Data (keys from data2):")
merged_data = pd.merge(data1, data2, how='right', on=['key1', 'key2'])
print(merged_data)
print("\nMerged Data (keys from data1):")
merged_data = pd.merge(data2, data1, how='right', on=['key1', 'key2'])
print(merged_data)
```

10. Write a Pandas program to merge two given datasets using multiple join keys.

### Test Data:

```
data1:
  key1 key2  P  Q
0   K0   K0 P0 Q0
1   K0   K1 P1 Q1
2   K1   K0 P2 Q2
3   K2   K1 P3 Q3
data2:
  key1 key2  R  S
0   K0   K0 R0 S0
1   K1   K0 R1 S1
2   K1   K0 R2 S2
3   K2   K0 R3 S3
```

```
import pandas as pd
```

```

data1 = pd.DataFrame({'key1': ['K0', 'K0', 'K1', 'K2'],
                      'key2': ['K0', 'K1', 'K0', 'K1'],
                      'P': ['P0', 'P1', 'P2', 'P3'],
                      'Q': ['Q0', 'Q1', 'Q2', 'Q3']})
data2 = pd.DataFrame({'key1': ['K0', 'K1', 'K1', 'K2'],
                      'key2': ['K0', 'K0', 'K0', 'K0'],
                      'R': ['R0', 'R1', 'R2', 'R3'],
                      'S': ['S0', 'S1', 'S2', 'S3']})
print("Original DataFrames:")
print(data1)
print("-----")
print(data2)
print("\nMerged Data:")
merged_data = pd.merge(data1, data2, on=['key1', 'key2'])
print(merged_data)

```

11. Write a Pandas program to create a new DataFrame based on existing series, using specified argument and override the existing columns names.

```

import pandas as pd
s1 = pd.Series([0, 1, 2, 3], name='col1')
s2 = pd.Series([0, 1, 2, 3])
s3 = pd.Series([0, 1, 4, 5], name='col3')
df = pd.concat([s1, s2, s3], axis=1, keys=['column1', 'column2',
'column3'])
print(df)

```

12. Write a Pandas program to create a combination from two dataframes where a column id combination appears more than once in both dataframes.

### Test Data:

```

data1:
  key1 key2  P  Q
0  K0   K0  P0 Q0
1  K0   K1  P1 Q1
2  K1   K0  P2 Q2
3  K2   K1  P3 Q3
data2:
  key1 key2  R  S
0  K0   K0  R0 S0
1  K1   K0  R1 S1
2  K1   K0  R2 S2

```

```
3      K2      K0      R3      S3
```

```
import pandas as pd
data1 = pd.DataFrame({'key1': ['K0', 'K0', 'K1', 'K2'],
                      'key2': ['K0', 'K1', 'K0', 'K1'],
                      'P': ['P0', 'P1', 'P2', 'P3'],
                      'Q': ['Q0', 'Q1', 'Q2', 'Q3']})
data2 = pd.DataFrame({'key1': ['K0', 'K1', 'K1', 'K2'],
                      'key2': ['K0', 'K0', 'K0', 'K0'],
                      'R': ['R0', 'R1', 'R2', 'R3'],
                      'S': ['S0', 'S1', 'S2', 'S3']})
print("Original DataFrames:")
print(data1)
print("-----")
print(data2)
print("\nMerged Data (many-to-many join case):")
result = pd.merge(data1, data2, on='key1')
print(result)
```

13. Write a Pandas program to combine the columns of two potentially differently-indexed DataFrames into a single result DataFrame.

### Test Data:

```
data1:
   A  B
K0 A0 B0
K1 A1 B1
K2 A2 B2
data2:
   C  D
K0 C0 D0
K2 C2 D2
K3 C3 D3
```

```
import pandas as pd
data1 = pd.DataFrame({'A': ['A0', 'A1', 'A2'],
                      'B': ['B0', 'B1', 'B2']},
                      index=['K0', 'K1', 'K2'])

data2 = pd.DataFrame({'C': ['C0', 'C2', 'C3'],
                      'D': ['D0', 'D2', 'D3']},
                      index=['K0', 'K2', 'K3'])

print("Original DataFrames:")
```

```

print(data1)
print("-----")
print(data2)
print("\nMerged Data (Joining on index):")
result = data1.join(data2)
print(result)

```

14. Write a Pandas program to merge two given dataframes with different columns.

### Test Data:

```

data1:
  key1 key2  P  Q
0   K0   K0 P0  Q0
1   K0   K1 P1  Q1
2   K1   K0 P2  Q2
3   K2   K1 P3  Q3
data2:
  key1 key2  R  S
0   K0   K0 R0  S0
1   K1   K0 R1  S1
2   K1   K0 R2  S2
3   K2   K0 R3  S3

```

```

import pandas as pd
data1 = pd.DataFrame({'key1': ['K0', 'K0', 'K1', 'K2'],
                      'key2': ['K0', 'K1', 'K0', 'K1'],
                      'P': ['P0', 'P1', 'P2', 'P3'],
                      'Q': ['Q0', 'Q1', 'Q2', 'Q3']})
data2 = pd.DataFrame({'key1': ['K0', 'K1', 'K1', 'K2'],
                      'key2': ['K0', 'K0', 'K0', 'K0'],
                      'R': ['R0', 'R1', 'R2', 'R3'],
                      'S': ['S0', 'S1', 'S2', 'S3']})
print("Original DataFrames:")
print(data1)
print("-----")
print(data2)
print("\nMerge two dataframes with different columns:")
result = pd.concat([data1, data2], axis=0, ignore_index=True)
print(result)

```

15. Write a Pandas program to Combine two DataFrame objects by filling null values in one DataFrame with non-null values from other DataFrame.

### Test Data:

Original DataFrames:

	A	B
0	NaN	3
1	0.0	4
2	NaN	5

  

	A	B
0	1	3.0
1	1	NaN
2	3	3.0

```
import pandas as pd
df1 = pd.DataFrame({'A': [None, 0, None], 'B': [3, 4, 5]})
df2 = pd.DataFrame({'A': [1, 1, 3], 'B': [3, None, 3]})
df1.combine_first(df2)
print("Original DataFrames:")
print(df1)
print("-----")
print(df2)
print("\nMerge two dataframes with different columns:")
result = df1.combine_first(df2)
print(result)
```