Segment 4 - Concatenating and transforming data

```
import numpy as np
import pandas as pd

from pandas import Series, DataFrame


DF_obj = pd.DataFrame(np.arange(36).reshape(6,6))
DF_obj
```

	0	1	2	3	4	5
0	0	1	2	3	4	5
1	6	7	8	9	10	11
2	12	13	14	15	16	17
3	18	19	20	21	22	23
4	24	25	26	27	28	29
5	30	31	32	33	34	35

```
DF_obj_2 = pd.DataFrame(np.arange(15).reshape(5,3))
DF_obj_2
```

Show hidden output

Concatenating data

```
pd.concat([DF_obj, DF_obj_2], axis=1)  #merging of data
```

Show hidden output

```
pd.concat([DF_obj, DF_obj_2]) #appending
```

Show hidden output

Transforming data

Dropping data

```
d1= DF_obj.drop([0, 2])   #inplace=True
d1
```

Show hidden output

```
DF_obj.drop([0, 2],inplace=False)
DF_obj
```

	0	1	2	3	4	5
0	0	1	2	3	4	5
1	6	7	8	9	10	11
2	12	13	14	15	16	17
3	18	19	20	21	22	23
4	24	25	26	27	28	29
5	30	31	32	33	34	35

```
DF_obj.drop([0, 2], axis=1)
```

	1	3	4	5
0	1	3	4	5
1	7	9	10	11
2	13	15	16	17
3	19	21	22	23
4	25	27	28	29
5	31	33	34	35

Adding data

```
series_obj = Series(np.arange(6))
series_obj.name = "added_variable"
series_obj
```

```
0    0
1    1
2    2
3    3
4    4
5    5
Name: added_variable, dtype: int64
```

```
variable_added = DataFrame.join(DF_obj, series_obj)
variable_added
```

	0	1	2	3	4	5	added_variable
0	0	1	2	3	4	5	0
1	6	7	8	9	10	11	1
2	12	13	14	15	16	17	2
3	18	19	20	21	22	23	3
4	24	25	26	27	28	29	4
5	30	31	32	33	34	35	5

```
added_datatable = variable_added.append(variable_added, ignore_index=False)
added_datatable
```

```
<ipython-input-36-9bea37e0ee2a>:1: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future vers
  added_datatable = variable_added.append(variable_added, ignore_index=False)
```

	0	1	2	3	4	5	added_variable
0	0	1	2	3	4	5	0
1	6	7	8	9	10	11	1
2	12	13	14	15	16	17	2
3	18	19	20	21	22	23	3
4	24	25	26	27	28	29	4
5	30	31	32	33	34	35	5
0	0	1	2	3	4	5	0
1	6	7	8	9	10	11	1
2	12	13	14	15	16	17	2
3	18	19	20	21	22	23	3
4	24	25	26	27	28	29	4
5	30	31	32	33	34	35	5

```
added_datatable = variable_added.append(variable_added, ignore_index=True)
added_datatable
```

```
<ipython-input-37-b2d1bccdcfb9>:1: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future vers
  added_datatable = variable_added.append(variable_added, ignore_index=True)
```

	0	1	2	3	4	5	added_variable
0	0	1	2	3	4	5	0
1	6	7	8	9	10	11	1
2	12	13	14	15	16	17	2
3	18	19	20	21	22	23	3
4	24	25	26	27	28	29	4
5	30	31	32	33	34	35	5
6	0	1	2	3	4	5	0
7	6	7	8	9	10	11	1
8	12	13	14	15	16	17	2
9	18	19	20	21	22	23	3
10	24	25	26	27	28	29	4
11	30	31	32	33	34	35	5

Sorting data

```
DF_sorted = DF_obj.sort_values(by=(5), ascending=[False])
DF_sorted
```

	0	1	2	3	4	5
5	30	31	32	33	34	35
4	24	25	26	27	28	29
3	18	19	20	21	22	23
2	12	13	14	15	16	17
1	6	7	8	9	10	11
0	0	1	2	3	4	5

Task 1: Concatenate DataFrames

You are given two DataFrames representing students in two different classes. Use Pandas to concatenate them into a single DataFrame.

Class A:

name,age,grade

Alice,20,A

Bob,21,B

Charlie,22,C

Class B:

plaintext

name,age,grade

David,20,B

Eve,22,A

Frank,21,C

1. Create two DataFrames using the data above.
2. Concatenate the two DataFrames along rows.
3. Add a new column called "class" indicating "Class A" or "Class B" for each student.
4. Print the final DataFrame.

Task 2: Merge Datasets Based on a Common Column

You are given two DataFrames representing student scores and student email addresses. Use merge to join these DataFrames on the "name" column.

Student Scores: name,score

Alice,85

Bob,78

Charlie,90

Eve,88

Student Emails: name,email

Alice,alice@example.com

Charlie,charlie@example.com

Eve,eve@example.com

Frank,frank@example.com

1. Merge the two DataFrames to create a new DataFrame with all columns.
2. Use an inner join to keep only the students who appear in both DataFrames.
3. Print the final merged DataFrame.

Task 3: Transform Columns in a DataFrame

Use the merged DataFrame from Task 2 and perform the following transformations:

1. Add 5 points to each student's score.
2. Create a new column called "pass" that indicates whether the student's score is greater than or equal to 80 (True/False).
3. Print the transformed DataFrame.

```
Start coding or generate with AI.
```