

pandas.DataFrame

module 4

DataFrame = Table

The diagram illustrates a DataFrame as a table with the following components:

- INDEX**: A blue arrow points to the first column of the table.
- POSITIONAL INDEX ON THE ROWS**: A green arrow points to the row indices (0 to 10) on the left side of the table.
- POSITIONAL INDEX ON THE COLUMNS**: A green arrow points to the column indices (0, 1, 2) above the table.
- COLUMNS (It's an Index object)**: A blue arrow points to the column headers (hw1, hw2, program) at the top of the table.
- VALUES**: An orange arrow points to the data cells within the table.

		0	1	2
		hw1	hw2	program
0	Demetria	2.0	4.0	MSIS
1	Dorian	10.0	10.0	MSIS
2	Garland	9.0	1.0	MSIS
3	Illuminada	2.0	NaN	MBA
4	Jeannine	6.0	7.0	MSIS
5	Jenny	8.0	NaN	NaN
6	John	NaN	10.0	MSIS
7	Luci	7.0	7.0	MSIS
8	Mercy	5.0	6.0	MSIS
9	Michael	6.0	10.0	MBA
10	Shelby	1.0	10.0	MSIS

Index, columns, values

Return the index (as an index object), the columns (as index object) and the values (as 2-dimensional ndarray)

Example:

```
df.index
```

```
Index([u'Demetria', u'Dorian', u'Garland', u'Illuminada', u'Jeannine', u'Jenny',  
      u'John', u'Luci', u'Mercy', u'Michael', u'Shelby'],  
      dtype='object')
```

```
df.columns
```

```
Index([u'hw1', u'hw2', u'program'], dtype='object')
```

```
df.values
```

```
array([[2.0, 4.0, 'MSIS'],  
       [10.0, 10.0, 'MSIS'],  
       [9.0, 1.0, 'MSIS'],  
       [2.0, nan, 'MBA'],  
       [6.0, 7.0, 'MSIS'],  
       [8.0, nan, nan],  
       [nan, 10.0, 'MSIS'],  
       [7.0, 7.0, 'MSIS'],  
       [5.0, 6.0, 'MSIS'],  
       [6.0, 10.0, 'MBA'],  
       [1.0, 10.0, 'MSIS']], dtype=object)
```

POSITIONAL INDEX
ON THE COLUMNS

		0 1 2			COLUMNS
		hw1	hw2	program	
INDEX	0	Demetria	2.0	4.0	MSIS
	1	Dorian	10.0	10.0	MSIS
	2	Garland	9.0	1.0	MSIS
	3	Illuminada	2.0	NaN	MBA
	4	Jeannine	6.0	7.0	MSIS
	5	Jenny	8.0	NaN	NaN
	6	John	NaN	10.0	MSIS
	7	Luci	7.0	7.0	MSIS
	8	Mercy	5.0	6.0	MSIS
	9	Michael	6.0	10.0	MBA
	10	Shelby	1.0	10.0	MSIS

POS. INDEX
ON ROWS

VALUES

df.iloc[x,y]

Access using the **positional index**..

- **x** is the information needed to select the rows: positional index or range of integers
- **y (optional)** is the information needed to select the columns: positional index or range of integers

POSITIONAL INDEX ON THE COLUMNS

INDEX

POS. INDEX ON ROWS

COLUMNS

VALUES

		0	1	2
		hw1	hw2	program
0	Demetria	2.0	4.0	MSIS
1	Dorian	10.0	10.0	MSIS
2	Garland	9.0	1.0	MSIS
3	Illuminada	2.0	NaN	MBA
4	Jeannine	6.0	7.0	MSIS
5	Jenny	8.0	NaN	NaN
6	John	NaN	10.0	MSIS
7	Luci	7.0	7.0	MSIS
8	Mercy	5.0	6.0	MSIS
9	Michael	6.0	10.0	MBA
10	Shelby	1.0	10.0	MSIS

df.iloc[x,y] – one row

```
df.iloc[2,:]
```

```
df.iloc[2]
```

RESULT:

```
hw1      9
hw2      1
program  MSIS
Name: Garland, dtype: object
```

← A Series!

POSITIONAL INDEX ON THE COLUMNS

INDEX

COLUMNS

POS. INDEX ON ROWS

		0	1	2
		hw1	hw2	program
0	Demetria	2.0	4.0	MSIS
1	Dorian	10.0	10.0	MSIS
2	Garland	9.0	1.0	MSIS
3	Illuminada	2.0	NaN	MBA
4	Jeannine	6.0	7.0	MSIS
5	Jenny	8.0	NaN	NaN
6	John	NaN	10.0	MSIS
7	Luci	7.0	7.0	MSIS
8	Mercy	5.0	6.0	MSIS
9	Michael	6.0	10.0	MBA
10	Shelby	1.0	10.0	MSIS

df.iloc[x,y] – one column

```
df.iloc[:,1]
```

RESULT:

Demetria	4.0
Dorian	10.0
Garland	1.0
Illuminada	NaN
Jeannine	7.0
Jenny	NaN
John	10.0
Luci	7.0
Mercy	6.0
Michael	10.0
Shelby	10.0

Name: hw2, dtype: float64

POSITIONAL INDEX ON THE COLUMNS

INDEX

COLUMNS

POS. INDEX ON ROWS

		0	1	2
		hw1	hw2	program
0	Demetria	2.0	4.0	MSIS
1	Dorian	10.0	10.0	MSIS
2	Garland	9.0	1.0	MSIS
3	Illuminada	2.0	NaN	MBA
4	Jeannine	6.0	7.0	MSIS
5	Jenny	8.0	NaN	NaN
6	John	NaN	10.0	MSIS
7	Luci	7.0	7.0	MSIS
8	Mercy	5.0	6.0	MSIS
9	Michael	6.0	10.0	MBA
10	Shelby	1.0	10.0	MSIS

df.iloc[x,y] – one specific value

```
df.iloc[2,1]
```

1.0

POSITIONAL INDEX ON THE COLUMNS

INDEX

POS. INDEX ON ROWS

COLUMNS

		0	1	2
		hw1	hw2	program
0	Demetria	2.0	4.0	MSIS
1	Dorian	10.0	10.0	MSIS
2	Garland	9.0	1.0	MSIS
3	Illuminada	2.0	NaN	MBA
4	Jeannine	6.0	7.0	MSIS
5	Jenny	8.0	NaN	NaN
6	John	NaN	10.0	MSIS
7	Luci	7.0	7.0	MSIS
8	Mercy	5.0	6.0	MSIS
9	Michael	6.0	10.0	MBA
10	Shelby	1.0	10.0	MSIS

df.iloc[x,y] – a subset of rows/columns

```
df.iloc[:5, -2:]
```

	hw2	program
Demetria	4.0	MSIS
Dorian	10.0	MSIS
Garland	1.0	MSIS
Illuminada	NaN	MBA
Jeannine	7.0	MSIS

POSITIONAL INDEX ON THE COLUMNS

INDEX

POS. INDEX ON ROWS

COLUMNS

		0	1	2
		hw1	hw2	program
0	Demetria	2.0	4.0	MSIS
1	Dorian	10.0	10.0	MSIS
2	Garland	9.0	1.0	MSIS
3	Illuminada	2.0	NaN	MBA
4	Jeannine	6.0	7.0	MSIS
5	Jenny	8.0	NaN	NaN
6	John	NaN	10.0	MSIS
7	Luci	7.0	7.0	MSIS
8	Mercy	5.0	6.0	MSIS
9	Michael	6.0	10.0	MBA
10	Shelby	1.0	10.0	MSIS

df.loc[x,y]

Access using the **index labels**.

- **x** is the information needed to select the rows: label index, range of index labels, or **boolean masks**

- **y (optional)** is the information needed to select the columns: label index, range of index labels, or **boolean masks**

POSITIONAL INDEX ON THE COLUMNS

INDEX

COLUMNS

POS. INDEX ON ROWS

VALUES

		0	1	2
		hw1	hw2	program
0	Demetria	2.0	4.0	MSIS
1	Dorian	10.0	10.0	MSIS
2	Garland	9.0	1.0	MSIS
3	Illuminada	2.0	NaN	MBA
4	Jeannine	6.0	7.0	MSIS
5	Jenny	8.0	NaN	NaN
6	John	NaN	10.0	MSIS
7	Luci	7.0	7.0	MSIS
8	Mercy	5.0	6.0	MSIS
9	Michael	6.0	10.0	MBA
10	Shelby	1.0	10.0	MSIS

df.loc[x,y] – one specific value

```
df.loc['Garland', 'hw2']
```

1.0

POSITIONAL INDEX ON THE COLUMNS

INDEX

COLUMNS

POS. INDEX ON ROWS

		0	1	2
		hw1	hw2	program
0	Demetria	2.0	4.0	MSIS
1	Dorian	10.0	10.0	MSIS
2	Garland	9.0	1.0	MSIS
3	Illuminada	2.0	NaN	MBA
4	Jeannine	6.0	7.0	MSIS
5	Jenny	8.0	NaN	NaN
6	John	NaN	10.0	MSIS
7	Luci	7.0	7.0	MSIS
8	Mercy	5.0	6.0	MSIS
9	Michael	6.0	10.0	MBA
10	Shelby	1.0	10.0	MSIS

df.loc[x,y] – one row

```
df.loc['Garland',:]
```

```
df.loc['Garland']
```

RESULT:

```
hw1      9
hw2      1
program  MSIS
Name: Garland, dtype: object
```

← A Series!

POSITIONAL INDEX ON THE COLUMNS

INDEX

COLUMNS

POS. INDEX ON ROWS

		0	1	2
		hw1	hw2	program
0	Demetria	2.0	4.0	MSIS
1	Dorian	10.0	10.0	MSIS
2	Garland	9.0	1.0	MSIS
3	Illuminada	2.0	NaN	MBA
4	Jeannine	6.0	7.0	MSIS
5	Jenny	8.0	NaN	NaN
6	John	NaN	10.0	MSIS
7	Luci	7.0	7.0	MSIS
8	Mercy	5.0	6.0	MSIS
9	Michael	6.0	10.0	MBA
10	Shelby	1.0	10.0	MSIS

df.loc[x,y] – one column

```
df.loc[:, 'hw1']
```

```
df['hw1']
```

```
df.hw1
```



RESULT:

Name	
Demetria	2.0
Dorian	10.0
Garland	9.0
Iluminada	2.0
Jeannine	6.0
Jenny	8.0
John	NaN
Luci	7.0
Mercy	5.0
Michael	6.0
Shelby	1.0
Name: hw1, dtype: float64	

POSITIONAL INDEX
ON THE COLUMNS

INDEX

		0	1	2
		hw1	hw2	program
0	Demetria	2.0	4.0	MSIS
1	Dorian	10.0	10.0	MSIS
2	Garland	9.0	1.0	MSIS
3	Iluminada	2.0	NaN	MBA
4	Jeannine	6.0	7.0	MSIS
5	Jenny	8.0	NaN	NaN
6	John	NaN	10.0	MSIS
7	Luci	7.0	7.0	MSIS
8	Mercy	5.0	6.0	MSIS
9	Michael	6.0	10.0	MBA
10	Shelby	1.0	10.0	MSIS

COLUMNS

POS. INDEX
ON ROWS

df.loc[x,y] – using Boolean masks

Select those students whose name starts with 'J'

```
mask = (df.index >= 'J') & (df.index < 'K')
df.loc[mask,:]
```

	hw1	hw2	program
Name			
Jeannine	6.0	7.0	MSIS
Jenny	8.0	NaN	NaN
John	NaN	10.0	MSIS

POSITIONAL INDEX ON THE COLUMNS

INDEX

COLUMNS

POS. INDEX ON ROWS

		0	1	2
		hw1	hw2	program
0	Demetria	2.0	4.0	MSIS
1	Dorian	10.0	10.0	MSIS
2	Garland	9.0	1.0	MSIS
3	Illuminada	2.0	NaN	MBA
4	Jeannine	6.0	7.0	MSIS
5	Jenny	8.0	NaN	NaN
6	John	NaN	10.0	MSIS
7	Luci	7.0	7.0	MSIS
8	Mercy	5.0	6.0	MSIS
9	Michael	6.0	10.0	MBA
10	Shelby	1.0	10.0	MSIS

Problems

1. Retrieve Shelby's hw1 grade
2. Retrieve Shelby's information
3. Who obtained the highest grade in hw2? Note that there are ties
4. Find those students who obtained the same score in hw1 and in hw2.
5. Find the average hw1 score of those students who got a hw2 score greater than 5.

sort_values

Sort the rows based on the value of a column

```
df.sort_values(by='hw1',ascending=False)
```

	hw1	hw2	program
Dorian	10.0	10.0	MSIS
Garland	9.0	1.0	MSIS
Jenny	8.0	NaN	NaN
Luci	7.0	7.0	MSIS
Jeannine	6.0	7.0	MSIS
Michael	6.0	10.0	MBA
Mercy	5.0	6.0	MSIS
Demetria	2.0	4.0	MSIS
Illuminada	2.0	NaN	MBA
Shelby	1.0	10.0	MSIS
John	NaN	10.0	MSIS

```
df.sort_values(by=['hw1','hw2'],ascending=[False, True])
```

	hw1	hw2	program
Dorian	10.0	10.0	MSIS
Garland	9.0	1.0	MSIS
Jenny	8.0	NaN	NaN
Luci	7.0	7.0	MSIS
Jeannine	6.0	7.0	MSIS
Michael	6.0	10.0	MBA
Mercy	5.0	6.0	MSIS
Demetria	2.0	4.0	MSIS
Illuminada	2.0	NaN	MBA
Shelby	1.0	10.0	MSIS
John	NaN	10.0	MSIS

POSITIONAL INDEX
ON THE COLUMNS

				0 1 2		
				hw1 hw2 program		
0	Demetria	2.0	4.0	MSIS		
1	Dorian	10.0	10.0	MSIS		
2	Garland	9.0	1.0	MSIS		
3	Illuminada	2.0	NaN	MBA		
4	Jeannine	6.0	7.0	MSIS		
5	Jenny	8.0	NaN	NaN		
6	John	NaN	10.0	MSIS		
7	Luci	7.0	7.0	MSIS		
8	Mercy	5.0	6.0	MSIS		
9	Michael	6.0	10.0	MBA		
10	Shelby	1.0	10.0	MSIS		

INDEX

COLUMNS

POS. INDEX
ON ROWS

sort_index

Sort by the index labels

```
df.sort_index()
```

	hw1	hw2	program
Demetria	2.0	4.0	MSIS
Dorian	10.0	10.0	MSIS
Garland	9.0	1.0	MSIS
Illuminada	2.0	NaN	MBA
Jeannine	6.0	7.0	MSIS
Jenny	8.0	NaN	NaN
John	NaN	10.0	MSIS
Luci	7.0	7.0	MSIS
Mercy	5.0	6.0	MSIS
Michael	6.0	10.0	MBA
Shelby	1.0	10.0	MSIS

POSITIONAL INDEX ON THE COLUMNS

COLUMNS

		0	1	2
INDEX		hw1	hw2	program
0	Demetria	2.0	4.0	MSIS
1	Dorian	10.0	10.0	MSIS
2	Garland	9.0	1.0	MSIS
3	Illuminada	2.0	NaN	MBA
4	Jeannine	6.0	7.0	MSIS
5	Jenny	8.0	NaN	NaN
6	John	NaN	10.0	MSIS
7	Luci	7.0	7.0	MSIS
8	Mercy	5.0	6.0	MSIS
9	Michael	6.0	10.0	MBA
10	Shelby	1.0	10.0	MSIS

POS. INDEX ON ROWS

head and tail

Return the first or last rows

```
df.head(4)
```

	hw1	hw2	program
Demetria	2.0	4.0	MSIS
Dorian	10.0	10.0	MSIS
Garland	9.0	1.0	MSIS
Illuminada	2.0	NaN	MBA

```
df.tail(3)
```

	hw1	hw2	program
Mercy	5.0	6.0	MSIS
Michael	6.0	10.0	MBA
Shelby	1.0	10.0	MSIS

POSITIONAL INDEX ON THE COLUMNS

INDEX

POS. INDEX ON ROWS

COLUMNS

		0	1	2
		hw1	hw2	program
0	Demetria	2.0	4.0	MSIS
1	Dorian	10.0	10.0	MSIS
2	Garland	9.0	1.0	MSIS
3	Illuminada	2.0	NaN	MBA
4	Jeannine	6.0	7.0	MSIS
5	Jenny	8.0	NaN	NaN
6	John	NaN	10.0	MSIS
7	Luci	7.0	7.0	MSIS
8	Mercy	5.0	6.0	MSIS
9	Michael	6.0	10.0	MBA
10	Shelby	1.0	10.0	MSIS

Problems

1. Sort the MSIS students by hw2 descending.
2. Show **only** the field *hw1* of the four students with the largest hw2 grade (do not use nlargest on the dataframe... it has bugs)

mean, max, min, etc

Aggregate functions will be broadcasted to all columns (axis = 0, default) or rows

```
df.mean()
```

```
hw1    5.600000  
hw2    7.222222  
dtype: float64
```

```
df.mean(axis=1)
```

```
Demetria    3.0  
Dorian     10.0  
Garland     5.0  
Iluminada   2.0  
Jeannine    6.5  
Jenny       8.0  
John       10.0  
Luci        7.0  
Mercy       5.5  
Michael     8.0  
Shelby      5.5  
dtype: float64
```

POSITIONAL INDEX ON THE COLUMNS

INDEX

COLUMNS

		0	1	2
		hw1	hw2	program
0	Demetria	2.0	4.0	MSIS
1	Dorian	10.0	10.0	MSIS
2	Garland	9.0	1.0	MSIS
3	Iluminada	2.0	NaN	MBA
4	Jeannine	6.0	7.0	MSIS
5	Jenny	8.0	NaN	NaN
6	John	NaN	10.0	MSIS
7	Luci	7.0	7.0	MSIS
8	Mercy	5.0	6.0	MSIS
9	Michael	6.0	10.0	MBA
10	Shelby	1.0	10.0	MSIS

POS. INDEX ON ROWS

Problems

1. Compute the spread (i.e., highest minus lowest hw grade) of each student. Consider only the students who submitted both homeworks
2. Who has the largest spread?

Adding rows

A new student has joined. His name is Oliver and he is the MSIS program; his hw1 is missing and his hw2 score is 8.

```
df2 = df.copy()
```

```
import numpy as np
df2.loc['Oliver'] = [np.nan, 8, 'MSIS']
df2
```

POSITIONAL INDEX
ON THE COLUMNS

012COLUMNS

INDEX

POS. INDEX
ON ROWS

		hw1	hw2	program
	Name			
0	Demetria	2.0	4.0	MSIS
1	Dorian	10.0	10.0	MSIS
2	Garland	9.0	1.0	MSIS
3	Illuminada	2.0	NaN	MBA
4	Jeannine	6.0	7.0	MSIS
5	Jenny	8.0	NaN	NaN
6	John	NaN	10.0	MSIS
7	Luci	7.0	7.0	MSIS
8	Mercy	5.0	6.0	MSIS
9	Michael	6.0	10.0	MBA
10	Shelby	1.0	10.0	MSIS
11	Oliver	NaN	8.0	MSIS

Adding rows

A new student has joined. Her name is Caroline and she got 4 in hw2. She is not in any program yet.

```
df2.loc['Caroline', 'hw2'] = 4  
df2
```

POSITIONAL INDEX
ON THE COLUMNS

INDEX

POS. INDEX
ON ROWS

COLUMNS

	0	1	2	
	hw1	hw2	program	
Name				
0	Demetria	2.0	4.0	MSIS
1	Dorian	10.0	10.0	MSIS
2	Garland	9.0	1.0	MSIS
3	Illuminada	2.0	NaN	MBA
4	Jeannine	6.0	7.0	MSIS
5	Jenny	8.0	NaN	NaN
6	John	NaN	10.0	MSIS
7	Luci	7.0	7.0	MSIS
8	Mercy	5.0	6.0	MSIS
9	Michael	6.0	10.0	MBA
10	Shelby	1.0	10.0	MSIS
11	Oliver	NaN	8.0	MSIS
12	Caroline	NaN	4.0	NaN

Adding columns

Add an "empty" column hw3

```
df2 = df.copy()
```

```
df2['hw3'] = np.nan  
df2
```


POSITIONAL INDEX
ON THE COLUMNS

INDEX

POS. INDEX
ON ROWS

COLUMNS

		0	1	2	
		hw1	hw2	program	hw3
	Name				
0	Demetria	2.0	4.0	MSIS	NaN
1	Dorian	10.0	10.0	MSIS	NaN
2	Garland	9.0	1.0	MSIS	NaN
3	Illuminada	2.0	NaN	MBA	NaN
4	Jeannine	6.0	7.0	MSIS	NaN
5	Jenny	8.0	NaN	NaN	NaN
6	John	NaN	10.0	MSIS	NaN
7	Luci	7.0	7.0	MSIS	NaN
8	Mercy	5.0	6.0	MSIS	NaN
9	Michael	6.0	10.0	MBA	NaN
10	Shelby	1.0	10.0	MSIS	NaN



Adding calculated columns

Let's add a column with the final grade. It is computed as $0.2 \cdot \text{hw1} + 0.8 \cdot \text{hw2}$.

```
df2 = df.copy()
```

```
df2['finalGrade'] = 0.2 * df2['hw1'] + 0.8 * df2['hw2']  
df2
```

POS. INDEX
ON ROWS

POSITIONAL INDEX
ON THE COLUMNS

INDEX		0 1 2			COLUMNS
		hw1	hw2	program	finalGrade
	Name				
0	Demetria	2.0	4.0	MSIS	3.6
1	Dorian	10.0	10.0	MSIS	10.0
2	Garland	9.0	1.0	MSIS	2.6
3	Illuminada	2.0	NaN	MBA	NaN
4	Jeannine	6.0	7.0	MSIS	6.8
5	Jenny	8.0	NaN	NaN	NaN
6	John	NaN	10.0	MSIS	NaN
7	Luci	7.0	7.0	MSIS	7.0
8	Mercy	5.0	6.0	MSIS	5.8
9	Michael	6.0	10.0	MBA	9.2
10	Shelby	1.0	10.0	MSIS	8.2

