

Pandas - MultiLevel Indexing

```
In [1]: import pandas as pd
import numpy as np
```

```
In [2]: data = [['Alice', 'Alice', 'Bob', 'Bob', 'Charlie', 'Charlie', 'Dave', 'Dave'],
                ['cs1', 'cs2', 'cs1', 'cs2', 'cs1', 'cs2', 'cs1', 'cs2']]
```

```
In [3]: tuples = list(zip(*data))
tuples
```

```
Out[3]: [('Alice', 'cs1'),
         ('Alice', 'cs2'),
         ('Bob', 'cs1'),
         ('Bob', 'cs2'),
         ('Charlie', 'cs1'),
         ('Charlie', 'cs2'),
         ('Dave', 'cs1'),
         ('Dave', 'cs2')]
```

```
In [4]: [(student, course) for student in ['Alice', 'Bob', 'Charlie', 'Dave'] \
         for course in ['cs1', 'cs2']]
```

```
Out[4]: [('Alice', 'cs1'),
         ('Alice', 'cs2'),
         ('Bob', 'cs1'),
         ('Bob', 'cs2'),
         ('Charlie', 'cs1'),
         ('Charlie', 'cs2'),
         ('Dave', 'cs1'),
         ('Dave', 'cs2')]
```

```
In [5]: indices = pd.MultiIndex.from_tuples(tuples,
                                             names = ['Student', 'Class'])
indices
```

```
Out[5]: MultiIndex([( 'Alice', 'cs1'),
                    ( 'Alice', 'cs2'),
                    (  'Bob', 'cs1'),
                    (  'Bob', 'cs2'),
                    ('Charlie', 'cs1'),
                    ('Charlie', 'cs2'),
                    (  'Dave', 'cs1'),
                    (  'Dave', 'cs2')],
                    names=['Student', 'Class'])
```

```
In [6]: np.random.seed(123)
s = pd.Series(np.random.randint(60,80,8), index = indices)
s
```

```
Out[6]: Student  Class
Alice    cs1      73
         cs2      62
Bob      cs1      62
         cs2      66
Charlie  cs1      77
         cs2      79
Dave     cs1      70
         cs2      61
dtype: int64
```

```
In [7]: # Using tuples as atomic labels

np.random.seed(123)
pd.Series(np.random.randint(60,80,8), index = tuples)
```

```
Out[7]: (Alice, cs1)      73
(Alice, cs2)      62
(Bob, cs1)      62
(Bob, cs2)      66
(Charlie, cs1)    77
(Charlie, cs2)    79
(Dave, cs1)      70
(Dave, cs2)      61
dtype: int64
```

```
In [8]: # Another approach

data = [['Alice', 'Bob', 'Charlie', 'Dave'],
        ['cs1', 'cs2']]

indices = pd.MultiIndex.from_product(data,
                                     names=['Student', 'Class'])
indices
```

```
Out[8]: MultiIndex([( 'Alice', 'cs1'),
                    ( 'Alice', 'cs2'),
                    (  'Bob', 'cs1'),
                    (  'Bob', 'cs2'),
                    ('Charlie', 'cs1'),
                    ('Charlie', 'cs2'),
                    (  'Dave', 'cs1'),
                    (  'Dave', 'cs2')],
                  names=['Student', 'Class'])
```

```
In [9]: np.random.seed(123)
s = pd.Series(np.random.randint(60,80,8), index = indices)
s
```

```
Out[9]: Student  Class
Alice    cs1      73
         cs2      62
Bob      cs1      62
         cs2      66
Charlie  cs1      77
         cs2      79
Dave     cs1      70
         cs2      61
dtype: int64
```

```
In [10]: # Convenient

np.random.seed(123)
data = [['Alice', 'Alice', 'Bob', 'Bob', 'Charlie', 'Charlie', 'Dave', 'Dave'],
        ['cs1', 'cs2', 'cs1', 'cs2', 'cs1', 'cs2', 'cs1', 'cs2']]

s = pd.Series(np.random.randint(60,80,8), index = data)
s
```

```
Out[10]: Alice    cs1    73
          cs2    62
          Bob     cs1    62
          cs2    66
          Charlie cs1    77
          cs2    79
          Dave    cs1    70
          cs2    61
          dtype: int64
```

```
In [11]: np.random.seed(123)
data = [['Alice', 'Alice', 'Bob', 'Bob', 'Charlie', 'Charlie', 'Dave', 'Dave'],
        ['cs1', 'cs2', 'cs1', 'cs2', 'cs1', 'cs2', 'cs1', 'cs2']]

df = pd.DataFrame(np.random.randint(60,80,(8, 4)), index = data,
                  columns = ['Quiz1', 'Quiz2', 'Quiz3', 'Quiz4'])
df
```

```
Out[11]:
```

		Quiz1	Quiz2	Quiz3	Quiz4
Alice	cs1	73	62	62	66
	cs2	77	79	70	61
Bob	cs1	60	77	75	69
	cs2	60	74	60	75
Charlie	cs1	79	74	64	60
	cs2	76	64	77	63
Dave	cs1	62	67	62	75
	cs2	76	67	69	63

```
In [12]: df.index
```

```
Out[12]: MultiIndex([( 'Alice', 'cs1'),
                      ( 'Alice', 'cs2'),
                      (  'Bob', 'cs1'),
                      (  'Bob', 'cs2'),
                      ('Charlie', 'cs1'),
                      ('Charlie', 'cs2'),
                      (  'Dave', 'cs1'),
                      (  'Dave', 'cs2')],
                    )
```

```
In [13]: df.index.names = ['Student', 'Class']
df
```

Out[13]:

		Quiz1	Quiz2	Quiz3	Quiz4
Student	Class				
Alice	cs1	73	62	62	66
	cs2	77	79	70	61
Bob	cs1	60	77	75	69
	cs2	60	74	60	75
Charlie	cs1	79	74	64	60
	cs2	76	64	77	63
Dave	cs1	62	67	62	75
	cs2	76	67	69	63

```
In [14]: df.index.get_level_values(0)
```

Out[14]: Index(['Alice', 'Alice', 'Bob', 'Bob', 'Charlie', 'Charlie', 'Dave', 'Dave'], dtype='object', name='Student')

```
In [15]: df.index.get_level_values(1)
```

Out[15]: Index(['cs1', 'cs2', 'cs1', 'cs2', 'cs1', 'cs2', 'cs1', 'cs2'], dtype='object', name='Class')

```
In [16]: df.index.get_level_values('Class')
```

Out[16]: Index(['cs1', 'cs2', 'cs1', 'cs2', 'cs1', 'cs2', 'cs1', 'cs2'], dtype='object', name='Class')

Indexing with MultiIndex

```
In [17]: df
```

Out[17]:

		Quiz1	Quiz2	Quiz3	Quiz4
Student	Class				
Alice	cs1	73	62	62	66
	cs2	77	79	70	61
Bob	cs1	60	77	75	69
	cs2	60	74	60	75
Charlie	cs1	79	74	64	60
	cs2	76	64	77	63
Dave	cs1	62	67	62	75
	cs2	76	67	69	63

```
In [18]: df.loc['Bob']
```

```
Out[18]:
```

	Quiz1	Quiz2	Quiz3	Quiz4
Class				
cs1	60	77	75	69
cs2	60	74	60	75

```
In [19]: df.loc['Bob', 'cs1']
```

```
Out[19]: Quiz1    60
Quiz2    77
Quiz3    75
Quiz4    69
Name: (Bob, cs1), dtype: int64
```

```
In [20]: df.loc['Bob'].loc['cs1']
```

```
Out[20]: Quiz1    60
Quiz2    77
Quiz3    75
Quiz4    69
Name: cs1, dtype: int64
```

```
In [21]: df.loc[( 'Bob', 'cs1')]
```

```
Out[21]: Quiz1    60
Quiz2    77
Quiz3    75
Quiz4    69
Name: (Bob, cs1), dtype: int64
```

```
In [22]: df.loc[( 'Bob', 'cs1'), 'Quiz1']
```

```
Out[22]: 60
```

```
In [23]: df.loc['Bob': 'Dave']
```

```
Out[23]:
```

		Quiz1	Quiz2	Quiz3	Quiz4
Student	Class				
Bob	cs1	60	77	75	69
	cs2	60	74	60	75
Charlie	cs1	79	74	64	60
	cs2	76	64	77	63
Dave	cs1	62	67	62	75
	cs2	76	67	69	63

```
In [24]: df.loc[('Bob', 'cs2'),('Dave', 'cs1')]
```

Out[24]:

		Quiz1	Quiz2	Quiz3	Quiz4
Student	Class				
Bob	cs2	60	74	60	75
	cs1	79	74	64	60
	cs2	76	64	77	63
Dave	cs1	62	67	62	75

```
In [25]: # with list of labels or tuples
```

```
df.loc[ [('Bob', 'cs2'), ('Dave', 'cs1')] ]
```

Out[25]:

		Quiz1	Quiz2	Quiz3	Quiz4
Student	Class				
Bob	cs2	60	74	60	75
Dave	cs1	62	67	62	75

```
In [ ]:
```

```
In [26]: df
```

Out[26]:

		Quiz1	Quiz2	Quiz3	Quiz4
Student	Class				
Alice	cs1	73	62	62	66
	cs2	77	79	70	61
Bob	cs1	60	77	75	69
	cs2	60	74	60	75
Charlie	cs1	79	74	64	60
	cs2	76	64	77	63
Dave	cs1	62	67	62	75
	cs2	76	67	69	63

```
In [27]: df.columns
```

Out[27]: Index(['Quiz1', 'Quiz2', 'Quiz3', 'Quiz4'], dtype='object')

```
In [28]: df.unstack()
```

Out[28]:

	Quiz1		Quiz2		Quiz3		Quiz4	
Class	cs1	cs2	cs1	cs2	cs1	cs2	cs1	cs2
Student								
Alice	73	77	62	79	62	70	66	61
Bob	60	60	77	74	75	60	69	75
Charlie	79	76	74	64	64	77	60	63
Dave	62	76	67	67	62	69	75	63

```
In [29]: df.unstack()['Quiz1']
```

Out[29]:

Class	cs1	cs2
Student		
Alice	73	77
Bob	60	60
Charlie	79	76
Dave	62	76

```
In [30]: df.unstack()['Quiz1', 'cs1']
```

Out[30]: Student
Alice 73
Bob 60
Charlie 79
Dave 62
Name: (Quiz1, cs1), dtype: int64

```
In [31]: # With Series
```

```
In [32]: s
```

Out[32]: Alice cs1 73
cs2 62
Bob cs1 62
cs2 66
Charlie cs1 77
cs2 79
Dave cs1 70
cs2 61
dtype: int64

```
In [33]: s[:, 'cs1']
```

```
Out[33]: Alice      73  
        Bob        62  
        Charlie    77  
        Dave       70  
        dtype: int64
```

```
In [34]: s['Bob']
```

```
Out[34]: cs1      62  
        cs2      66  
        dtype: int64
```

```
In [35]: s['Bob']['cs1']
```

```
Out[35]: 62
```

```
In [36]: s[('Bob', 'cs1')]
```

```
Out[36]: 62
```

```
In [37]: s['Bob', 'cs1']
```

```
Out[37]: 62
```

```
In [38]: s
```

```
Out[38]: Alice      cs1      73  
        cs2      62  
        Bob        cs1      62  
        cs2      66  
        Charlie    cs1      77  
        cs2      79  
        Dave       cs1      70  
        cs2      61  
        dtype: int64
```

```
In [39]: s.unstack()
```

```
Out[39]:
```

	cs1	cs2
Alice	73	62
Bob	62	66
Charlie	77	79
Dave	70	61

```
In [40]: s.unstack()['cs1']
```

```
Out[40]: Alice      73  
        Bob        62  
        Charlie    77  
        Dave       70  
        Name: cs1, dtype: int64
```

```
In [ ]:
```

```
In [ ]:
```


Cross-section

- `xs()`

In [41]: `df`

Out[41]:

		Quiz1	Quiz2	Quiz3	Quiz4
Student	Class				
Alice	cs1	73	62	62	66
	cs2	77	79	70	61
Bob	cs1	60	77	75	69
	cs2	60	74	60	75
Charlie	cs1	79	74	64	60
	cs2	76	64	77	63
Dave	cs1	62	67	62	75
	cs2	76	67	69	63

In [42]: `df.xs('Bob')`

Out[42]:

		Quiz1	Quiz2	Quiz3	Quiz4
Class					
cs1		60	77	75	69
cs2		60	74	60	75

In [43]: `df.xs('cs1', level=1)`

Out[43]:

		Quiz1	Quiz2	Quiz3	Quiz4
Student					
Alice		73	62	62	66
Bob		60	77	75	69
Charlie		79	74	64	60
Dave		62	67	62	75

In [44]: `df.xs('cs1', level='Class')`

Out[44]:

		Quiz1	Quiz2	Quiz3	Quiz4
Student					
Alice		73	62	62	66
Bob		60	77	75	69
Charlie		79	74	64	60
Dave		62	67	62	75

```
In [45]: # Using slice
```

```
df.loc[(slice(None), 'cs1'), :]
```

Out[45]:

		Quiz1	Quiz2	Quiz3	Quiz4
Student	Class				
Alice	cs1	73	62	62	66
Bob	cs1	60	77	75	69
Charlie	cs1	79	74	64	60
Dave	cs1	62	67	62	75

```
In [46]: df.loc[(slice('Alice', 'Charlie'), slice(None)), :]
```

Out[46]:

		Quiz1	Quiz2	Quiz3	Quiz4
Student	Class				
Alice	cs1	73	62	62	66
	cs2	77	79	70	61
Bob	cs1	60	77	75	69
	cs2	60	74	60	75
Charlie	cs1	79	74	64	60
	cs2	76	64	77	63

MultilIndex for Columns

```
In [47]: # hierarchical indices and columns

np.random.seed(123)
index = pd.MultiIndex.from_product([[2017, 2018], [1, 2]],
                                   names=['year', 'visit'])
columns = pd.MultiIndex.from_product([['Alice', 'Bob', 'Charlie'], ['HR', 'Temp']],
                                   names=['subject', 'type'])

# mock some data
data = np.round(np.random.randn(4, 6), 1)
data[:, ::2] *= 10
data += 37

# create the DataFrame
health_data = pd.DataFrame(data, index=index, columns=columns)
health_data
```

```
Out[47]:
```

		subject		Alice		Bob		Charlie	
		type		HR	Temp	HR	Temp	HR	Temp
	year	visit							
2017		1	26.0	38.0	40.0	35.5	31.0	38.7	
		2	13.0	36.6	50.0	36.1	30.0	36.9	
2018		1	52.0	36.4	33.0	36.6	59.0	39.2	
		2	47.0	37.4	44.0	38.5	28.0	38.2	

```
In [48]: health_data['Bob']
```

```
Out[48]:
```

		type	HR	Temp
	year	visit		
2017		1	40.0	35.5
		2	50.0	36.1
2018		1	33.0	36.6
		2	44.0	38.5

```
In [49]: health_data['Bob', 'HR']
```

```
Out[49]: year  visit
2017    1      40.0
        2      50.0
2018    1      33.0
        2      44.0
Name: (Bob, HR), dtype: float64
```

```
In [50]: health_data.loc[:, ('Bob', 'HR')]
```

```
Out[50]: year  visit
2017    1      40.0
        2      50.0
2018    1      33.0
        2      44.0
Name: (Bob, HR), dtype: float64
```

```
In [51]: health_data.iloc[:2, :2]
```

Out[51]:

		subject Alice		
		type	HR	Temp
year	visit			
2017	1		26.0	38.0
	2		13.0	36.6

```
In [52]: health_data
```

Out[52]:

subject		Alice		Bob		Charlie	
type		HR	Temp	HR	Temp	HR	Temp
year	visit						
2017	1	26.0	38.0	40.0	35.5	31.0	38.7
	2	13.0	36.6	50.0	36.1	30.0	36.9
2018	1	52.0	36.4	33.0	36.6	59.0	39.2
	2	47.0	37.4	44.0	38.5	28.0	38.2

```
In [53]: health_data.xs('HR', level='type', axis=1)
```

Out[53]:

		subject	Alice	Bob	Charlie
year	visit				
2017	1		26.0	40.0	31.0
	2		13.0	50.0	30.0
2018	1		52.0	33.0	59.0
	2		47.0	44.0	28.0

```
In [54]: health_data
```

Out[54]:

subject		Alice		Bob		Charlie	
	type	HR	Temp	HR	Temp	HR	Temp
year	visit						
2017	1	26.0	38.0	40.0	35.5	31.0	38.7
	2	13.0	36.6	50.0	36.1	30.0	36.9
2018	1	52.0	36.4	33.0	36.6	59.0	39.2
	2	47.0	37.4	44.0	38.5	28.0	38.2

```
In [55]: # IndexSlice
```

```
idx = pd.IndexSlice  
idx
```

```
Out[55]: <pandas.core.indexing._IndexSlice at 0x1178cb278>
```

```
In [56]: health_data.loc[idx[:,2], idx[:, 'HR']]
```

```
Out[56]:
```

		subject	Alice	Bob	Charlie
		type	HR	HR	HR
year	visit				
2017	2	13.0	50.0	30.0	
2018	2	47.0	44.0	28.0	

```
In [57]: health_data.loc[:, idx[:, 'HR']]
```

```
Out[57]:
```

		subject	Alice	Bob	Charlie
		type	HR	HR	HR
year	visit				
	1	26.0	40.0	31.0	
2017	2	13.0	50.0	30.0	
	1	52.0	33.0	59.0	
2018	2	47.0	44.0	28.0	

```
In [58]: health_data.loc[:, idx['Alice':'Bob', 'HR']]
```

```
Out[58]:
```

		subject	Alice	Bob
		type	HR	HR
year	visit			
	1	26.0	40.0	
2017	2	13.0	50.0	
	1	52.0	33.0	
2018	2	47.0	44.0	

```
In [59]: health_data.index
```

```
Out[59]: MultiIndex([(2017, 1),  
                    (2017, 2),  
                    (2018, 1),  
                    (2018, 2)],  
                  names=['year', 'visit'])
```

```
In [60]: health_data.columns
```

```
Out[60]: MultiIndex([( 'Alice',  'HR'),
                      ( 'Alice',  'Temp'),
                      (  'Bob',    'HR'),
                      (  'Bob',    'Temp'),
                      ('Charlie',  'HR'),
                      ('Charlie',  'Temp')],
                    names=['subject', 'type'])
```

Swapping levels

```
In [61]: df
```

```
Out[61]:
```

		Quiz1	Quiz2	Quiz3	Quiz4
Student	Class				
Alice	cs1	73	62	62	66
	cs2	77	79	70	61
Bob	cs1	60	77	75	69
	cs2	60	74	60	75
Charlie	cs1	79	74	64	60
	cs2	76	64	77	63
Dave	cs1	62	67	62	75
	cs2	76	67	69	63

```
In [62]: df.swaplevel()
```

```
Out[62]:
```

		Quiz1	Quiz2	Quiz3	Quiz4
Class	Student				
cs1	Alice	73	62	62	66
cs2	Alice	77	79	70	61
cs1	Bob	60	77	75	69
cs2	Bob	60	74	60	75
cs1	Charlie	79	74	64	60
cs2	Charlie	76	64	77	63
cs1	Dave	62	67	62	75
cs2	Dave	76	67	69	63

```
In [63]: df.swaplevel().loc['cs1']
```

```
Out[63]:
```

	Quiz1	Quiz2	Quiz3	Quiz4
Student				
Alice	73	62	62	66
Bob	60	77	75	69
Charlie	79	74	64	60
Dave	62	67	62	75

```
In [64]: df.index
```

```
Out[64]: MultiIndex([( 'Alice', 'cs1'),
                    ( 'Alice', 'cs2'),
                    (  'Bob', 'cs1'),
                    (  'Bob', 'cs2'),
                    ('Charlie', 'cs1'),
                    ('Charlie', 'cs2'),
                    (  'Dave', 'cs1'),
                    (  'Dave', 'cs2')],
                  names=['Student', 'Class'])
```

```
In [65]: df2 = df.swaplevel('Student', 'Class')
df2
```

```
Out[65]:
```

		Quiz1	Quiz2	Quiz3	Quiz4
Class	Student				
cs1	Alice	73	62	62	66
cs2	Alice	77	79	70	61
cs1	Bob	60	77	75	69
cs2	Bob	60	74	60	75
cs1	Charlie	79	74	64	60
cs2	Charlie	76	64	77	63
cs1	Dave	62	67	62	75
cs2	Dave	76	67	69	63

```
In [66]: df2.mean()
```

```
Out[66]: Quiz1      70.375
Quiz2      70.500
Quiz3      67.375
Quiz4      66.500
dtype: float64
```

```
In [67]: df2.mean(level='Class')
```

```
Out[67]:
```

	Quiz1	Quiz2	Quiz3	Quiz4
Class				
cs1	68.50	70.0	65.75	67.5
cs2	72.25	71.0	69.00	65.5

```
In [68]: df2.mean(level='Student')
```

```
Out[68]:
```

	Quiz1	Quiz2	Quiz3	Quiz4
Student				
Alice	75.0	70.5	66.0	63.5
Bob	60.0	75.5	67.5	72.0
Charlie	77.5	69.0	70.5	61.5
Dave	69.0	67.0	65.5	69.0

```
In [69]: df2
```

```
Out[69]:
```

		Quiz1	Quiz2	Quiz3	Quiz4
Class	Student				
cs1	Alice	73	62	62	66
cs2	Alice	77	79	70	61
cs1	Bob	60	77	75	69
cs2	Bob	60	74	60	75
cs1	Charlie	79	74	64	60
cs2	Charlie	76	64	77	63
cs1	Dave	62	67	62	75
cs2	Dave	76	67	69	63

```
In [70]: df2.sort_index(level='Class')
```

```
Out[70]:
```

		Quiz1	Quiz2	Quiz3	Quiz4
Class	Student				
cs1	Alice	73	62	62	66
	Bob	60	77	75	69
	Charlie	79	74	64	60
	Dave	62	67	62	75
cs2	Alice	77	79	70	61
	Bob	60	74	60	75
	Charlie	76	64	77	63
	Dave	76	67	69	63

```
In [71]: print(df2.sort_index(level='Class'))
```

		Quiz1	Quiz2	Quiz3	Quiz4
Class	Student				
cs1	Alice	73	62	62	66
	Bob	60	77	75	69
	Charlie	79	74	64	60
	Dave	62	67	62	75
cs2	Alice	77	79	70	61
	Bob	60	74	60	75
	Charlie	76	64	77	63
	Dave	76	67	69	63


```
In [72]: df2.sort_index(level='Student')
```

Out[72]:

		Quiz1	Quiz2	Quiz3	Quiz4
Class	Student				
cs1	Alice	73	62	62	66
cs2	Alice	77	79	70	61
cs1	Bob	60	77	75	69
cs2	Bob	60	74	60	75
cs1	Charlie	79	74	64	60
cs2	Charlie	76	64	77	63
cs1	Dave	62	67	62	75
cs2	Dave	76	67	69	63

```
In [73]: df2.reset_index(inplace=True)  
df2
```

Out[73]:

	Class	Student	Quiz1	Quiz2	Quiz3	Quiz4
0	cs1	Alice	73	62	62	66
1	cs2	Alice	77	79	70	61
2	cs1	Bob	60	77	75	69
3	cs2	Bob	60	74	60	75
4	cs1	Charlie	79	74	64	60
5	cs2	Charlie	76	64	77	63
6	cs1	Dave	62	67	62	75
7	cs2	Dave	76	67	69	63

```
In [ ]:
```