

Problem-Based Learning

Python 3

Lecture: 12-dictionary

Online Python

- <https://www.programiz.com/python-programming/online-compiler/>

Lecture Notes

- <https://web.phy.ntnu.edu.tw/~hongyi/?url=notes>



Python list and Python dictionary

✓ Python list:

```
[[96, 65, 73], [88, 76, 82], [92, 84, 89]]
```

✓ Python dict:

```
{'a':[96, 65, 73], 'b':[88, 76, 82], 'c':[92, 84, 89]}
```

Python list → Pandas DataFrame

- `import pandas as pd`
- `a = [[1, 2, 3],
 [6, 7, 8],
 [2, 3, 4]]`
- `ind = ['a', 'b', 'c']`
- `col = ['A', 'B', 'C']`
- `d = pd.DataFrame(a, index=ind, columns=col)`
- `print(d)`

	A	B	C
a	1	6	2
b	2	7	3
c	3	8	4

Python dict (column-like) → Pandas DataFrame

- `import pandas as pd`
- `a = {
 'A': [1, 2, 3],
 'B': [6, 7, 8],
 'C': [2, 3, 4]
}`
- `print(a)`
- `ind = ['a', 'b', 'c']`
- `d = pd.DataFrame(a, index=ind)`
- `print(d)`

keys() values()
↓ ↓
{ 'A': [1, 2, 3],
 'B': [6, 7, 8],
 'C': [2, 3, 4] }

	A	B	C
a	1	6	2
b	2	7	3
c	3	8	4

Python dict (row-like) → Pandas DataFrame

- `import pandas as pd`
- `a = {`

`'a': [1, 6, 2],`

`'b': [2, 7, 3],`

`'c': [3, 8, 4]`

`}`

- `col = ['A', 'B', 'C']`

- `d = pd.DataFrame(a.values(), index=a.keys(), columns=col)`

- `print(d)`

keys() values()

↓ ↓
{ 'a': [1, 6, 2], 'b': [2, 7, 3], 'c': [3, 8, 4] }

	A	B	C
a	1	6	2
b	2	7	3
c	3	8	4

Pandas DataFrame：增加直行單筆資料

- `b = d`
- `c = [7, 8, 9]`
- `b['D'] = c`
- `print(b)`

	A	B	C	D
a	1	6	2	7
b	2	7	3	8
c	3	8	4	9

Pandas DataFrame：增加橫列單筆資料

- `b = d`
- `c = [4, 9, 5]`
- `b.loc['d'] = c`
- `print(b)`

	A	B	C
a	1	6	2
b	2	7	3
c	3	8	4
d	4	9	5

Pandas DataFrame：增加直行多筆資料

- `b = d`
- `a2 = {
 'D':[7,8,9],
 'E':[3,4,5]
}`
- `ind = ['a', 'b', 'c']`
- `d2 = pd.DataFrame(a2, index=ind)`
- `b = pd.concat([b,d2], axis=1)`
- `print(b)`

	A	B	C	D	E	← axis=1
a	1	6	2	7	3	
b	2	7	3	8	4	
c	3	8	4	9	5	

↑
axis=0

Pandas DataFrame：增加直行多筆資料

- `b = d`
- `a2 = {`
 `'a': [7, 3],`
 `'b': [8, 4],`
 `'c': [9, 5],`
 `}`
- `col2 = ['D', 'E']`
- `d2 = pd.DataFrame(a2.values(), index=c.keys(), columns=col2)`
- `b = pd.concat([b, d2], axis=1)`
- `print(b)`

	A	B	C	D	E	← axis=1
a	1	6	2	7	3	
b	2	7	3	8	4	
c	3	8	4	9	5	

↑
axis=0

Pandas DataFrame：增加橫列多筆資料

- `b = d`
- `a2 = {
 'A': [4, 5],
 'B': [9, 1],
 'C': [5, 6]
}`
- `ind2 = ['d', 'e']`
- `d2 = pd.DataFrame(a2, index=ind2)`
- `b = pd.concat([b, d2], axis=0)`
- `print(b)`

Pandas DataFrame：增加橫列多筆資料

- `b = d`
- `a2 = {
 'd': [4, 9, 5],
 'e': [5, 1, 6],
}`
- `d2 = pd.DataFrame(a2.values(), index=c.keys(), columns=col)`
- `b = pd.concat([b, d2], axis=0)`
- `print(b)`

Problem

- 計算每個學生的平均成績，並附加到原表格。
- 找出每一個學生的最高分和最低分 and 其所對應的學科，並附加到原表格。

	國文	英文	數學
學生1	96	65	73
學生2	88	76	82
學生3	92	84	89
學生4	82	73	64
學生5	70	83	68

+

平均

+

最高分

+

最低分

提示：
b.min(axis=?)
b.max(axis=?)
b.mean(axis=?)

b.idxmin(axis=?)
b.idxmax(axis=?)