

lec1_step7

October 22, 2020

```
In [ ]: # Practice 3-1 (page 13/29)
        # https://www.slideshare.net/tadahirotaniguchi0624/3-46861684
```

```
In [ ]: # https://note.nkmk.me/python-dict-list-sort/
```

```
In [13]: import pprint
```

```
l = [{'Name': 'Australia', 'Population': 25680158, 'Capital City': 'Canberra', 'Points': [-35.28, 149.13]},
     {'Name': 'Bangladesh', 'Population': 169468990, 'Capital City': 'Dhaka', 'Points': [23.71, 90.41]},
     {'Name': 'Chile', 'Population': 17373831, 'Capital City': 'Santiago', 'Points': [-27.37, -70.33]}]
```

```
In [19]: pprint.pprint(sorted(l, key=lambda x: x['Name']))
```

```
[{'Capital City': 'Canberra',
  'Name': 'Australia',
  'Points': [-35.28, 149.13],
  'Population': 25680158},
 {'Capital City': 'Dhaka',
  'Name': 'Bangladesh',
  'Points': [23.71, 90.41],
  'Population': 169468990},
 {'Capital City': 'Santiago',
  'Name': 'Chile',
  'Points': [-27.37, -70.33],
  'Population': 17373831}]
```

```
In [20]: pprint.pprint(sorted(l, key=lambda x: x['Population']))
```

```
[{'Capital City': 'Santiago',
  'Name': 'Chile',
  'Points': [-27.37, -70.33],
  'Population': 17373831},
 {'Capital City': 'Canberra',
  'Name': 'Australia',
  'Points': [-35.28, 149.13],
  'Population': 25680158},
 {'Capital City': 'Dhaka',
  'Name': 'Bangladesh',
  'Points': [23.71, 90.41],
  'Population': 169468990}]
```

```
'Name': 'Bangladesh',  
'Points': [23.71, 90.41],  
'Population': 169468990}]
```

```
In [21]: pprint.pprint(sorted(l, key=lambda x: x['Population'], reverse=True))
```

```
[{'Capital City': 'Dhaka',  
  'Name': 'Bangladesh',  
  'Points': [23.71, 90.41],  
  'Population': 169468990},  
{ 'Capital City': 'Canberra',  
  'Name': 'Australia',  
  'Points': [-35.28, 149.13],  
  'Population': 25680158},  
{ 'Capital City': 'Santiago',  
  'Name': 'Chile',  
  'Points': [-27.37, -70.33],  
  'Population': 17373831}]
```

```
In [25]: # https://note.nkmk.me/python-dict-create/  
keys = ['k1', 'k2', 'k3']  
values = [1, 2, 3]  
d = {k: v for k, v in zip(keys, values)}  
print(d)
```

```
{'k1': 1, 'k2': 2, 'k3': 3}
```

```
In [ ]: Node=[]  
keys = ['cost', 'h', 'f']  
values = [1, 2, 3]  
for i in  
d = {k: v for k, v in zip(keys, values)}  
print(d)
```

```
In [28]: # A: ascii code 65  
chr(65)
```

```
Out[28]: 'A'
```

```
In [32]: # for i in range(1,10,1):  
for i in range(1,10):  
    s=chr(i+65-1)  
    print(s)
```

```
A  
B
```

C
D
E
F
G
H
I

```
In [30]: # for i in range(1,10,1):
         for i in range(65,65+10):
             s=chr(i)
             print(s)
```

A
B
C
D
E
F
G
H
I
J

```
In [2]: Node=[chr(i) for i in range(65,65+10)]
         print(Node)
```

['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']

```
In [5]: H=list(range(1,len(Node)))
         print(H)
         H=list(range(1,len(Node)))
         print(H)
         F=3*list(range(1,len(Node)))
         print(F)
```

[1, 2, 3, 4, 5, 6, 7, 8, 9]

[1, 2, 3, 4, 5, 6, 7, 8, 9]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9]

```
In [6]: print(Cost)
         H=list(map(lambda x: x * 2, Cost))
         print(H)
         F=list(map(lambda x: x * 3, Cost))
         print(F)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
[2, 4, 6, 8, 10, 12, 14, 16, 18]
[3, 6, 9, 12, 15, 18, 21, 24, 27]
```

```
In [56]: data1 = [1, 3, 6, 50, 5]
        data2 = list(map(lambda x: x * 2, data1))
        print(data1)
        print(data2)
```

```
[1, 3, 6, 50, 5]
[2, 6, 12, 100, 10]
```

```
In [64]: keys = ['node', 'cost', 'h', 'f']
        values = [1, 2, 3]
        d = {k: v for k, v in zip(keys, values)}
        print(d)
```

```
{'node': 1, 'cost': 2, 'h': 3}
```

```
In [66]: keys = ['node', 'cost', 'h', 'f']
        values = [1, 2, 3]
        d_all=[]
        for i in range(0, len(Node)-1):
            values=[Node[i], Cost[i], H[i], F[i]]
            d = {k: v for k, v in zip(keys, values)}
            d_all.append(d)
        print(d_all)
```

```
[{'node': 'A', 'cost': 1, 'h': 2, 'f': 3}, {'node': 'B', 'cost': 2, 'h': 4, 'f': 6}, {'node':
```

```
In [70]: pprint.pprint(sorted(d_all, key=lambda x: x['node']))
```

```
[{'cost': 1, 'f': 3, 'h': 2, 'node': 'A'},
 {'cost': 2, 'f': 6, 'h': 4, 'node': 'B'},
 {'cost': 3, 'f': 9, 'h': 6, 'node': 'C'},
 {'cost': 4, 'f': 12, 'h': 8, 'node': 'D'},
 {'cost': 5, 'f': 15, 'h': 10, 'node': 'E'},
 {'cost': 6, 'f': 18, 'h': 12, 'node': 'F'},
 {'cost': 7, 'f': 21, 'h': 14, 'node': 'G'},
 {'cost': 8, 'f': 24, 'h': 16, 'node': 'H'},
 {'cost': 9, 'f': 27, 'h': 18, 'node': 'I'}]
```

```
In [71]: pprint.pprint(sorted(d_all, key=lambda x: x['cost']))
```

```
[{'cost': 1, 'f': 3, 'h': 2, 'node': 'A'},
 {'cost': 2, 'f': 6, 'h': 4, 'node': 'B'},
 {'cost': 3, 'f': 9, 'h': 6, 'node': 'C'},
 {'cost': 4, 'f': 12, 'h': 8, 'node': 'D'},
 {'cost': 5, 'f': 15, 'h': 10, 'node': 'E'},
 {'cost': 6, 'f': 18, 'h': 12, 'node': 'F'},
 {'cost': 7, 'f': 21, 'h': 14, 'node': 'G'},
 {'cost': 8, 'f': 24, 'h': 16, 'node': 'H'},
 {'cost': 9, 'f': 27, 'h': 18, 'node': 'I'}]
```

```
In [72]: pprint.pprint(sorted(d_all, key=lambda x: x['h']))
```

```
[{'cost': 1, 'f': 3, 'h': 2, 'node': 'A'},
 {'cost': 2, 'f': 6, 'h': 4, 'node': 'B'},
 {'cost': 3, 'f': 9, 'h': 6, 'node': 'C'},
 {'cost': 4, 'f': 12, 'h': 8, 'node': 'D'},
 {'cost': 5, 'f': 15, 'h': 10, 'node': 'E'},
 {'cost': 6, 'f': 18, 'h': 12, 'node': 'F'},
 {'cost': 7, 'f': 21, 'h': 14, 'node': 'G'},
 {'cost': 8, 'f': 24, 'h': 16, 'node': 'H'},
 {'cost': 9, 'f': 27, 'h': 18, 'node': 'I'}]
```

```
In [73]: pprint.pprint(sorted(d_all, key=lambda x: x['f']))
```

```
[{'cost': 1, 'f': 3, 'h': 2, 'node': 'A'},
 {'cost': 2, 'f': 6, 'h': 4, 'node': 'B'},
 {'cost': 3, 'f': 9, 'h': 6, 'node': 'C'},
 {'cost': 4, 'f': 12, 'h': 8, 'node': 'D'},
 {'cost': 5, 'f': 15, 'h': 10, 'node': 'E'},
 {'cost': 6, 'f': 18, 'h': 12, 'node': 'F'},
 {'cost': 7, 'f': 21, 'h': 14, 'node': 'G'},
 {'cost': 8, 'f': 24, 'h': 16, 'node': 'H'},
 {'cost': 9, 'f': 27, 'h': 18, 'node': 'I'}]
```

```
In [ ]:
```