—

- description : _
- author :
- email : hylee@repia.com
- lastupdate : 2020-06-22

—

```python
# Section04-4
#                  (      )
#        ,

#                 (    X,    X,    O,     O)
# Key, Value (Json) -> MongoDB

#
print('#= = = =          = = = = #')
a = {'name': 'Kim', 'phone': '01012345678', 'birth': 870124}
b = {0: 'Hello python!'}
c = {'arr': [1, 2, 3, 4]}

print('#=== type,        = = = #')
print('a - ', type(a), a)
print('b - ', type(b), b)
print('c - ', type(c), c)

print()

#
print('#= = =            = = = #')
print('a - ', a['name'])  #    X ->
print('a - ', a.get('name'))  #     X -> None
print('b - ', b[0])
print('b - ', b.get(0))
print('c - ', c['arr'])
print('c - ', c['arr'][3])
print('c - ', c.get('arr'))

print()

#
print('#= = =              = = = #')
a['address'] = 'seoul'
```

```python
print('a - ', a)
a['rank'] = [1, 2, 3]
print('a - ', a)

print()

# dict_keys, dict_values, dict_items :        (iterate)
# dict_items -> dict
print('#===  dict_keys, dict_values, dict_items         = = = #')
print('a - ', a.keys())
print('b - ', b.keys())
print('c - ', c.keys())

print()

print('a - ', list(a.keys()))
print('b - ', list(b.keys()))
print('c - ', list(c.keys()))

print()

print('a - ', a.values())
print('b - ', b.values())
print('c - ', c.values())

print()

print('a - ', list(a.values()))
print('b - ', list(b.values()))
print('c - ', list(c.values()))

print()

print('a - ', a.items())
print('b - ', b.items())
print('c - ', c.items())

print()

print('a - ', list(a.items()))
print('b - ', list(b.items()))
print('c - ', list(c.items()))

print()

print('a - ', 'name' in a)
print('a - ', 'addr' in a)

print()
# Key        index        .
```

```python
# index
print('#== dict_keys   values           index              = = #')
temp = list(a.keys())
print(temp[1:3])
# Key           index           .

print()
print()


print('#= = = =      = = = = #')
#    (Sets)        (    X,      X)

#
a = set()
b = set([1, 2, 3, 4])
c = set([1, 4, 5, 6])
d = set([1, 2, 'Pen', 'Cap', 'Plate'])

print('#=== type,          = = = #')
print('a - ', type(a), a)
print('b - ', type(b), b)
print('c - ', type(c), c)
print('d - ', type(d), d)

print()

#
print('#= = =              = = = #')
t = tuple(b)
print('t - ', type(t), t)
print('t - ', t[0], t[1:3])

print()

#
print('#= = =               = = = #')
l = list(c)
print('l - ', type(l), l)
print('l - ', l[0], l[1:3])

print()

print("#= = =                 = = =")
#
s1 = set([1, 2, 3, 4, 5, 6])
s2 = set([4, 5, 6, 7, 8, 9])


print("#= =        = =")
print('l - ', s1 & s2)
print('l - ', s1.intersection(s2))
```

```
print("#==        ==")
print('l - ', s1 | s2)
print('l - ', s1.union(s2))

print("#==        ==")
print('l - ', s1 - s2)
print('l - ', s1.difference(s2))

print()

print("#===       /    ===")
#        &
s1 = set([1, 2, 3, 4])
s1.add(5)
print('s1 - ', s1)

s1.remove(2)
print('s1 - ', s1)
```

```
# ====        ==== #
#=== type,      === #
a -  <class 'dict'> {'name': 'Kim', 'phone': '01012345678', 'birth': 870124}
b -  <class 'dict'> {0: 'Hello python!'}
c -  <class 'dict'> {'arr': [1, 2, 3, 4]}

# ===         === #
a -  Kim
a -  Kim
b -  Hello python!
b -  Hello python!
c -  [1, 2, 3, 4]
c -  4
c -  [1, 2, 3, 4]

# ===         === #
a -  {'name': 'Kim', 'phone': '01012345678', 'birth': 870124, 'address':
'seoul'}
a -  {'name': 'Kim', 'phone': '01012345678', 'birth': 870124, 'address':
'seoul', 'rank': [1, 2, 3]}

#===  dict_keys, dict_values, dict_items       === #
a -  dict_keys(['name', 'phone', 'birth', 'address', 'rank'])
b -  dict_keys([0])
c -  dict_keys(['arr'])

a -  ['name', 'phone', 'birth', 'address', 'rank']
b -  [0]
```

```
c -  ['arr']

a -  dict_values(['Kim', '01012345678', 870124, 'seoul', [1, 2, 3]])
b -  dict_values(['Hello python!'])
c -  dict_values([[1, 2, 3, 4]])

a -  ['Kim', '01012345678', 870124, 'seoul', [1, 2, 3]]
b -  ['Hello python!']
c -  [[1, 2, 3, 4]]

a -  dict_items([('name', 'Kim'), ('phone', '01012345678'), ('birth',
870124), ('address', 'seoul'), ('rank', [1, 2, 3])])
b -  dict_items([(0, 'Hello python!')])
c -  dict_items([('arr', [1, 2, 3, 4])])

a -  [('name', 'Kim'), ('phone', '01012345678'), ('birth', 870124),
('address', 'seoul'), ('rank', [1, 2, 3])]
b -  [(0, 'Hello python!')]
c -  [('arr', [1, 2, 3, 4])]

a -  True
a -  False

#== dict_keys   values                index                   = = #
['phone', 'birth']


# = = = =      = = = = #
#=== type,          = = = #
a -  <class 'set'> set()
b -  <class 'set'> {1, 2, 3, 4}
c -  <class 'set'> {1, 4, 5, 6}
d -  <class 'set'> {1, 2, 'Plate', 'Cap', 'Pen'}

# = = =             = = = #
t -  <class 'tuple'> (1, 2, 3, 4)
t -  1 (2, 3)

# = = =               = = = #
l -  <class 'list'> [1, 4, 5, 6]
l -  1 [4, 5]

# = = =                  = = =
# = =        = =
l -  {4, 5, 6}
l -  {4, 5, 6}
# = =        = =
l -  {1, 2, 3, 4, 5, 6, 7, 8, 9}
l -  {1, 2, 3, 4, 5, 6, 7, 8, 9}
# = =        = =
l -  {1, 2, 3}
l -  {1, 2, 3}
```

```
# = = =              /       = = =
s1 -  {1, 2, 3, 4, 5}
s1 -  {1, 3, 4, 5}
```

Tip

, [python](), ,