py4kids (https://github.com/wgong/py4kids)

Python Data Types - Set, Dictionary

In this lesson, we learn 2 important data types:

- Set : a sequence of unique items
- Dictionary : set of key:value pairs, aka, Map, Associative Array, Hash Table

In [1]:
```
from jyquickhelper import add_notebook_menu
add_notebook_menu()
```

Out[1]:

Set (https://docs.python.org/3/library/stdtypes.html?highlight=set#set)

- an unordered collection of distinct items
- set collection **delimitor** is curly brackets: {, }

In [2]:
```
color_set = {'Red','Green','Blue'}
```

In [3]:
```
type(color_set)
```

Out[3]:
```
set
```

In [4]:

```
color_set.add('White')
```

In [5]:

```
# cannot add duplicate item
color_set.add('Red')
print(color_set)
```

```
{'Blue', 'White', 'Green', 'Red'}
```

In [6]:

```
len(color_set)
```

Out[6]:

```
4
```

In [7]:

```
# check existence
print('Black' in color_set)
```

```
False
```

In [8]:

```
# check existence
print('Red' in color_set)
```

```
True
```

Distinct items in a list

In [9]:

```
num_list = [3, 5, 3, 13, 7, 9, 13, 13]
num_list
```

Out[9]:

```
[3, 5, 3, 13, 7, 9, 13, 13]
```

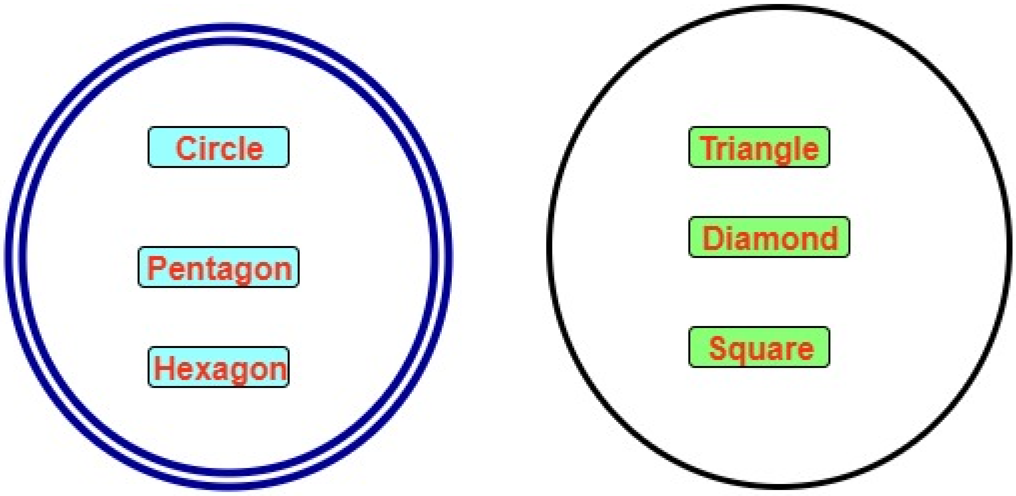
In [10]:

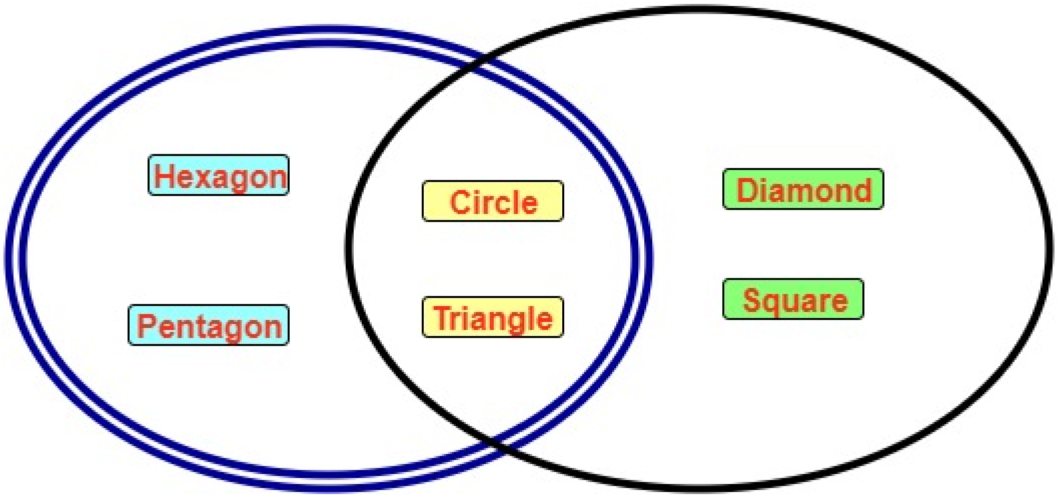
```
print(set(num_list))
```

```
{9, 13, 3, 5, 7}
```

Set operations

Venn diagram





In [11]:
```
shapes_1 = {'Circle','Triangle','Pentagon', 'Hexagon'}
print(shapes_1)
```

```
{'Triangle', 'Pentagon', 'Circle', 'Hexagon'}
```

In [12]:
```
type(shapes_1)
```

Out[12]:
```
set
```

In [13]:
```
shapes_2 = {'Circle','Triangle','Diamond', 'Square'}
print(shapes_2)
```

```
{'Diamond', 'Triangle', 'Square', 'Circle'}
```

In [14]:
```
# Intersection -- find the common item between two sets
shapes_1.intersection(shapes_2)
```

Out[14]:
```
{'Circle', 'Triangle'}
```

In [15]:
```
# Union -- find total items among all the sets
shapes_1.union(shapes_2)
```

Out[15]:
```
{'Circle', 'Diamond', 'Hexagon', 'Pentagon', 'Square', 'Triangle'}
```

In [16]:
```
# Difference
shapes_1.difference(shapes_2)
```

Out[16]:
```
{'Hexagon', 'Pentagon'}
```

In [17]:
```
# order matters
shapes_2.difference(shapes_1)
```

Out[17]:
```
{'Diamond', 'Square'}
```

Dictionary (https://docs.python.org/3/library/stdtypes.html?highlight=set#dict) / Map

- Dictionary **delimitor** is curly brackets: {, } ,
  - key/value pair **delimitor** is ":"
- Keys are distinct, that is why it uses same delimitors as set
- The value store information associated with a key
- an unordered collection of key:value pair (unlike regular English dictionary, keys are NOT sorted alphabetically)
- a very efficient/useful data structure
- Three nicknames
  - Map (2-dimentional)
  - Associative Array
  - Hash table

Start with an example

In [18]:
```
favorite_sports = {'Ralph Williams' : 'Football',
    'Michael Tippett' : 'Basketball',
    'Edward Elgar' : 'Baseball',
    'Rebecca Clarke' : 'Netball',
    'Ethel Smyth' : 'Badminton',
    'Frank Bridge' : 'Rugby'
    }
```

In [19]:
```
type(favorite_sports)
```

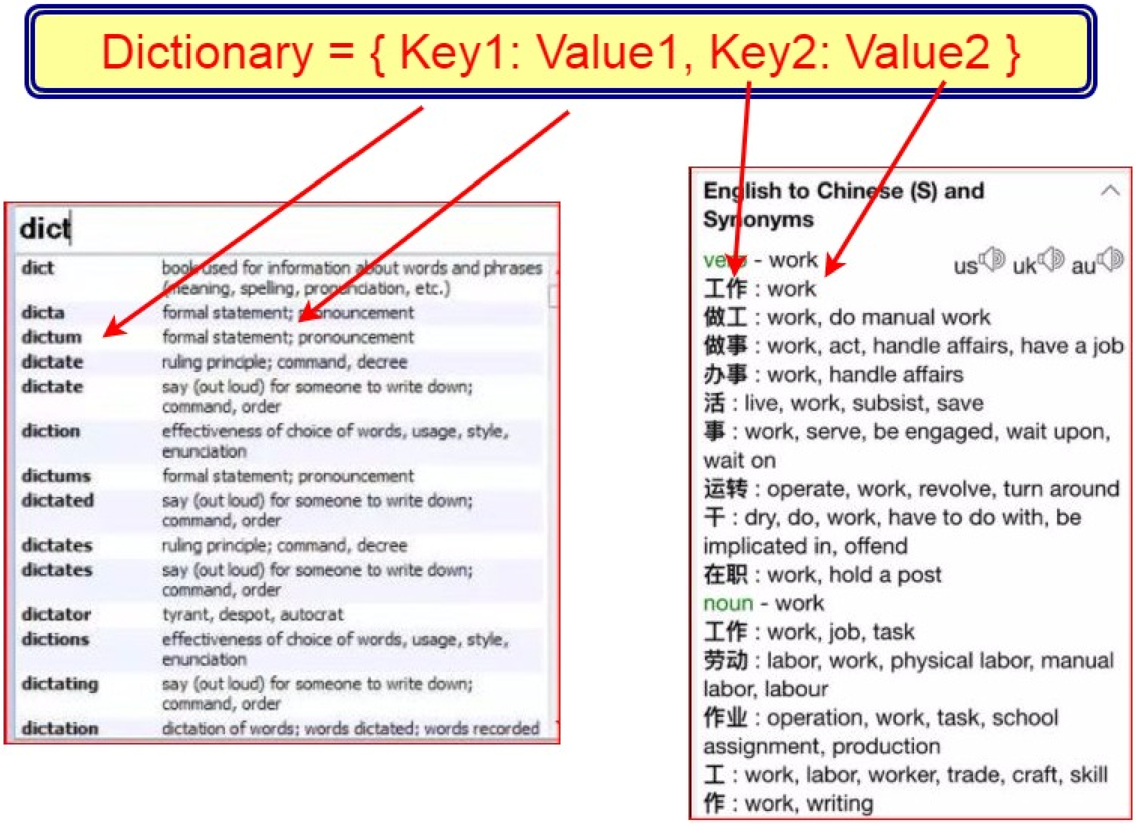
Out[19]:
```
dict
```

In [20]:
```
len(favorite_sports)
```

Out[20]:
```
6
```

In [21]:
```
print(favorite_sports)
```

```
{'Ethel Smyth': 'Badminton', 'Rebecca Clarke': 'Netball', 'Michael Tippett': 'B
asketball', 'Frank Bridge': 'Rugby', 'Ralph Williams': 'Football', 'Edward Elga
r': 'Baseball'}
```



Dictionary = { Key1: Value1, Key2: Value2 }

dict

dict	book used for information about words and phrases (meaning, spelling, pronunciation, etc.)
dicta	formal statement; pronouncement
dictum	formal statement; pronouncement
dictate	ruling principle; command, decree
dictate	say (out loud) for someone to write down; command, order
diction	effectiveness of choice of words, usage, style, enunciation
dictums	formal statement; pronouncement
dictated	say (out loud) for someone to write down; command, order
dictates	ruling principle; command, decree
dictates	say (out loud) for someone to write down; command, order
dictator	tyrant, despot, autocrat
dictions	effectiveness of choice of words, usage, style, enunciation
dictating	say (out loud) for someone to write down; command, order
dictation	dictation of words; words dictated; words recorded

English to Chinese (S) and Synonyms

verb - work us uk au

工作 : work
做工 : work, do manual work
做事 : work, act, handle affairs, have a job
办事 : work, handle affairs
活 : live, work, subsist, save
事 : work, serve, be engaged, wait upon, wait on
运转 : operate, work, revolve, turn around
干 : dry, do, work, have to do with, be implicated in, offend
在职 : work, hold a post

noun - work

工作 : work, job, task
劳动 : labor, work, physical labor, manual labor, labour
作业 : operation, work, task, school assignment, production
工 : work, labor, worker, trade, craft, skill
作 : work, writing

How to create dictionary

In [22]:
```
dict2 = { '工作' : 'work', '学习':'study, learn' , '玩':'play'}
```

In [23]:
```
print(dict2)
```

```
{'学习': 'study, learn', '工作': 'work', '玩': 'play'}
```

In [24]:
```
dict2['学习']
```

Out[24]:
```
'study, learn'
```

In [25]:
```
key = '工作'
print("Meaning of %s is %s" % (key,  dict2[key]))
```

```
Meaning of 工作 is work
```

In [26]:
```
dict3 = dict(name='John', age=10, height=54.5, weight= 70)
```

In [27]:
```
dict3
```

Out[27]:
```
{'age': 10, 'height': 54.5, 'name': 'John', 'weight': 70}
```

In [28]:
```
type(dict3)
```

Out[28]:
```
dict
```

In [29]:
```
len(dict3)
```

Out[29]:
```
4
```

Common operations

get all the keys

In [30]:
```
key_list = dict2.keys()
```

In [31]:
```
print(key_list)
```

```
dict_keys(['学习', '工作', '玩'])
```

get all the values

In [32]:
```
value_list = dict2.values()
```

In [33]:
```
print(value_list)
```

```
dict_values(['study, learn', 'work', 'play'])
```

get all the items

key:value pair is called an item

In [34]:
```
item_list = dict2.items()
```

In [35]:
```
print(item_list)
```

```
dict_items([('学习', 'study, learn'), ('工作', 'work'), ('玩', 'play')])
```

In [36]:
```
# count number of items
print(len(dict2))
```

```
3
```

in - existence check

```
In [37]: print('玩' in dict2)

True
```

```
In [38]: print('游戏' in dict2)

False
```

add an item

```
In [39]: dict2['游戏'] = 'game'
```

```
In [40]: dict2

Out[40]: {'学习': 'study, learn', '工作': 'work', '游戏': 'game', '玩': 'play'}
```

```
In [41]: # count number of items
         print(len(dict2))

4
```

update an item

```
In [42]: dict2['游戏'] = 'computer game'.upper()
```

```
In [43]: dict2

Out[43]: {'学习': 'study, learn', '工作': 'work', '游戏': 'COMPUTER GAME', '玩': 'play'}
```

remove an item

```
In [44]: dict2['work'] = '工作'
```

```
In [45]: dict2

Out[45]: {'work': '工作',
          '学习': 'study, learn',
          '工作': 'work',
          '游戏': 'COMPUTER GAME',
          '玩': 'play'}
```

```
In [46]: del dict2['work']
```

```
In [47]: dict2

Out[47]: {'学习': 'study, learn', '工作': 'work', '游戏': 'COMPUTER GAME', '玩': 'play'}
```

```
In [48]: dict2['work'] = '工作'
In [49]: dict2

Out[49]: {'work': '工作',
          '学习': 'study, learn',
          '工作': 'work',
          '游戏': 'COMPUTER GAME',
          '玩': 'play'}
```

```
In [50]: dict2.pop('work')

Out[50]: '工作'
```

```
In [51]: dict2

Out[51]: {'学习': 'study, learn', '工作': 'work', '游戏': 'COMPUTER GAME', '玩': 'play'}
```

clear a dictionary

```
In [52]: dict3 = {1: 'one', 2: 'two', 3: 'three'}
```

```
In [53]: dict3

Out[53]: {1: 'one', 2: 'two', 3: 'three'}
```

```
In [54]: dict3.clear()
```

```
In [55]: dict3

Out[55]: {}
```

```
In [56]: len(dict3)

Out[56]: 0
```

reset to empty

```
In [57]: dict3 = {1: 'one', 2: 'two', 3: 'three'}
```

```
In [58]: dict3

Out[58]: {1: 'one', 2: 'two', 3: 'three'}
```

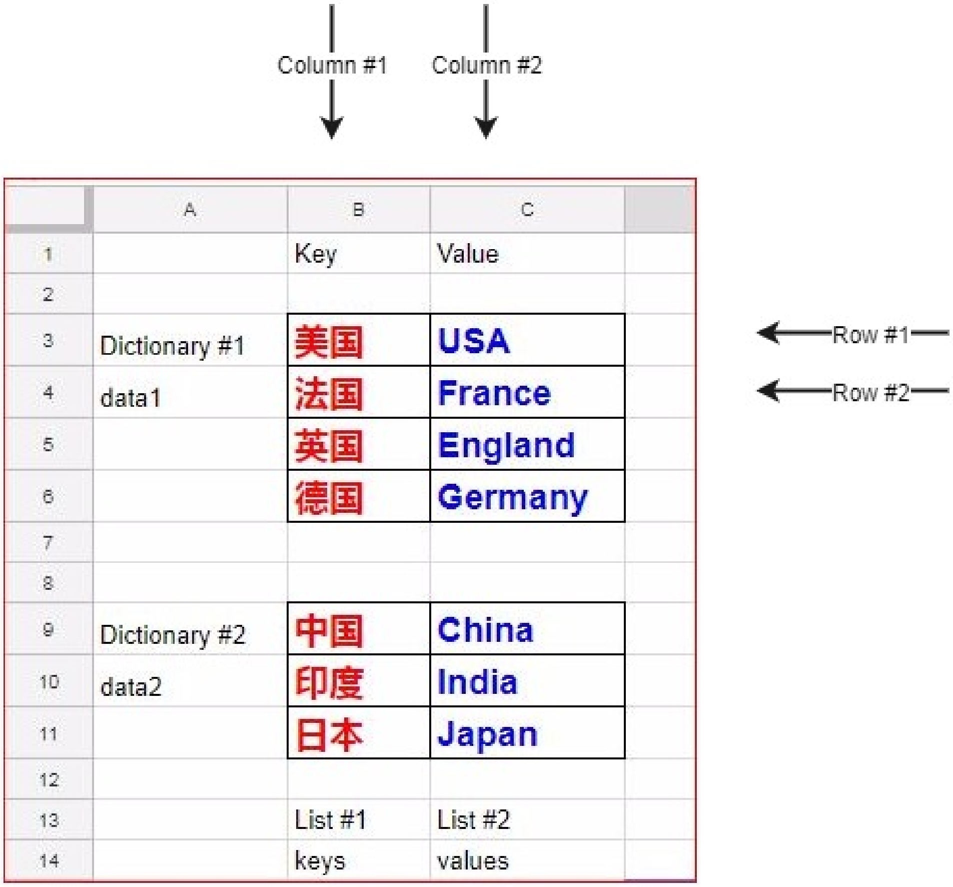
```
In [59]: dict3 = {}
```

```
In [60]: dict3

Out[60]: {}
```

merge two dictionaries into one

row-wise



	A	B	C
1		Key	Value
2			
3	Dictionary #1	美国	USA
4	data1	法国	France
5		英国	England
6		德国	Germany
7			
8			
9	Dictionary #2	中国	China
10	data2	印度	India
11		日本	Japan
12			
13		List #1	List #2
14		keys	values

In [61]:
```
# western countries
dict4_a = {'美国':'USA', '英国':'England', '法国':'France', '德国':'Germany'
           #, '俄国' : 'Russia'
          }
```

In [62]:
```
dict4_a
```

Out[62]:
```
{'德国': 'Germany', '法国': 'France', '美国': 'USA', '英国': 'England'}
```

In [63]:
```
# eastern countries
dict4_b = {'中国':'China', '印度':'India', '日本':'Japan'}
```

In [64]:
```
dict4_b
```

Out[64]:
```
{'中国': 'China', '印度': 'India', '日本': 'Japan'}
```

In [65]:
```
dict4 = dict(list(dict4_a.items()) + list(dict4_b.items()))
```

```
In [66]: dict4
```

```
Out[66]: {'中国': 'China',
          '印度': 'India',
          '德国': 'Germany',
          '日本': 'Japan',
          '法国': 'France',
          '美国': 'USA',
          '英国': 'England'}
```

zip two lists into a dictionary

column-wise

```
In [67]: key_list = dict4.keys()
```

```
In [68]: value_list = dict4.values()
```

```
In [69]: key_list, value_list
```

```
Out[69]: (dict_keys(['日本', '中国', '英国', '印度', '法国', '美国', '德国']),
          dict_values(['Japan', 'China', 'England', 'India', 'France', 'USA', 'German
         y']))
```

```
In [70]: dict5 = dict(zip(key_list, value_list))
```

```
In [ ]:
```

```
In [71]: dict5
```

```
Out[71]: {'中国': 'China',
          '印度': 'India',
          '德国': 'Germany',
          '日本': 'Japan',
          '法国': 'France',
          '美国': 'USA',
          '英国': 'England'}
```

```
In [72]: # switch key/value
         dict6 = dict(zip(value_list, key_list))
```

```
In [73]: dict6
```

```
Out[73]: {'China': '中国',
          'England': '英国',
          'France': '法国',
          'Germany': '德国',
          'India': '印度',
          'Japan': '日本',
          'USA': '美国'}
```

Complex dictionary

```
In [74]: # key is string, value is a list
```

```
In [75]: dict7 = dict(one=[0], two=[0,1], three=[0,1,2], four=[0,1,2,4])
```

```
In [76]: dict7

Out[76]: {'four': [0, 1, 2, 4], 'one': [0], 'three': [0, 1, 2], 'two': [0, 1]}
```

```
In [77]: # nested dictionary:  key is number, value is a dictionary
```

```
In [78]: dict8 = {1: {'name':'John Wang', 'sex':'Male', 'grade':7, 'age':14} ,
                  2: {'name':'Jane Li', 'sex':'Female', 'grade':8, 'age':15} ,
                  3: {'name':'Kevin Chen', 'sex':'Male', 'grade':6, 'age':12}
                 }
```

```
In [79]: dict8

Out[79]: {1: {'age': 14, 'grade': 7, 'name': 'John Wang', 'sex': 'Male'},
          2: {'age': 15, 'grade': 8, 'name': 'Jane Li', 'sex': 'Female'},
          3: {'age': 12, 'grade': 6, 'name': 'Kevin Chen', 'sex': 'Male'}}
```

```
In [80]: dict8[1]

Out[80]: {'age': 14, 'grade': 7, 'name': 'John Wang', 'sex': 'Male'}
```

```
In [81]: #dict8[5]
```

first encounter with loop

- **for** is a keyword for looping,
- repeat within the loop
- code within the loop must be indented because python uses indentation to group codes together.

NOTE: this is unique and critical for python language

```
In [82]: for item in dict6:
             print(item)

         Germany
         India
         Japan
         USA
         China
         England
         France
```

In [83]:

```
for key,value in dict6.items():
    print('key=', key,' \t: ', 'value=',value)
```

```
key= Germany    :  value= 德国
key= India      :  value= 印度
key= Japan      :  value= 日本
key= USA        :  value= 美国
key= China      :  value= 中国
key= England    :  value= 英国
key= France     :  value= 法国
```

In [84]:

```
print(dict6.items())
```

```
dict_items([('Germany', '德国'), ('India', '印度'), ('Japan', '日本'), ('USA',
 '美国'), ('China', '中国'), ('England', '英国'), ('France', '法国')])
```

In [85]:

```
# how to track loop number - use a counter

# initialize the counter before loop starts
n = 0
for item in dict6:
    n = n + 1  # increment counter by 1
    print('loop counter = %d' % n)
    print('\t\tkey=', item)
```

```
loop counter = 1
                key= Germany
loop counter = 2
                key= India
loop counter = 3
                key= Japan
loop counter = 4
                key= USA
loop counter = 5
                key= China
loop counter = 6
                key= England
loop counter = 7
                key= France
```

In [86]:

```
# how to loop thru a dictionary

# initialize the counter before loop starts
n = 0
for item in dict6:
    n = n + 1  # increment counter by 1
    print('loop counter = %d' % n)
    print('\t\tKey  =', item)
    print('\t\tValue=', dict6[item])
```

```
loop counter = 1
                Key  = Germany
                Value= 德国
loop counter = 2
                Key  = India
                Value= 印度
loop counter = 3
                Key  = Japan
                Value= 日本
loop counter = 4
                Key  = USA
                Value= 美国
loop counter = 5
                Key  = China
                Value= 中国
loop counter = 6
                Key  = England
                Value= 英国
loop counter = 7
                Key  = France
                Value= 法国
```

In [ ]: