Dictionaries and their usage

In this lecture, we discuss a more advance data structure, *dictionary*, in Python.
The **key concept** in using dictionary is that each item in a distionary is a pair.
Another important thing to remember is that dictionary **does not** store these key values paris in any paritcular order.

Created by John C.S. Lui, May 29, 2018

General Syntax of dictionary

dictionary_name = {key_1: value_1, key_2: value_2, key_3: value_3}

Let illustrate it using an example.

In [ ]:
```
# Define a disctionary which stores professor's name and his/her research interests

professors_interests = {
    'john c. s. lui': [ 'applied machine learning', 'network science', 'network economics', 'mobile and IoT
    'patrick p. c. lee': ['file and storage systems', 'network monitoring' ],
    'james cheng': ['distributed systems', 'big data analytic systems', 'distributed machine learning syste
    }
```

In [ ]:
```
# Let see how we can access items in the disctionary

# Accessing dictionary via keys
print("For %s, research interests are: %s" %('john c. s. lui'.title(), professors_interests['john c. s. lui
print("For %s, research interests are: %s" %('patrick p. c. lee'.title(), professors_interests['patrick p.
print("For %s, research interests are: %s" %('james cheng'.title(), professors_interests['james cheng']))
```

In [ ]:

```
# Define a disctionary which stores professor's name and his/her research interests

professors_interests = {
    'john c. s. lui': [ 'applied machine learning', 'network science', 'network economics', 'mobile and IoT
    'patrick p. c. lee': ['file and storage systems', 'network monitoring' ],
    'james cheng': ['distributed systems', 'big data analytic systems', 'distributed machine learning syste
    }

# Let's access items within the research interests
name = ''
name = input('Please input name: ')   # Ask user to enter the name (or key)

while name != 'q':
    if name in professors_interests:             # check whether  the "key" is in the disctionary !!!!
        print("For %s, research interests are: " %name.title())
        interests = professors_interests[name]   # given the key name, access value (or the list)
        for item in interests:                   # access each item within a list
            print("\t" + str(interests.index(item)+1) +  ". " + item.title())
    else:
        print("Sorry, the name is not in the list.")

    name = input('Please input name: ')          # ask user to input again
```

In [ ]:
```
# Let see how we can loop through all elements in the dictionary
# Define a disctionary which stores professor's name and his/her research interests

professors_interests = {
    'john c. s. lui': [ 'applied machine learning', 'network science', 'network economics', 'mobile and IoT
    'patrick p. c. lee': ['file and storage systems', 'network monitoring' ],
    'james cheng': ['distributed systems', 'big data analytic systems', 'distributed machine learning syste
    }

for name, interests in professors_interests.items(): # extract the key:value of each item in the dictionary
    print("For Professor " + name.title() + ", interests are: ", end='')
    for interest in interests:
        if interest != interests[-1]:
            print(interest +", ", end='')
        else:
            print(interest +".")
```

Some common functions for dictionary

Let's illustrate some useful dictionary functions

In [ ]:
```
# Create a new and empty dictionary
professors = {}

# Fill it with some key:value pairs
# In here, key is the name (string) of a professor
#          value is a list, which contains research interests (list), rank (string) and age (integer)
professors['john c. s. lui'] =[['applied machine learning', 'network science', 'network economics', 'mobile
       'full professor', 82]
professors['patrick p. c. lee'] = [['file and storage systems', 'network monitoring'], 'associate professor
professors['james cheng'] = [['distributed systems', 'big data analytic systems', 'distributed machine lear
        'assistant professor', 18]
professors['eric lo'] = [['block chain', 'database systems', 'big data computing'], 'associate professor',

# Display them out
for name, a_list in professors.items():
    research_interest = a_list[0]  # get research interest
    rank = a_list[1]               # get rank
    age  = a_list[2]               # get age
    print("\nName: " + name.title() + ", age = " + str(age) +  ", rank = " + rank.title() + ".")
    print("  "+"Research interests are: ", end='')
    for research_area in research_interest:
        if research_area != research_interest[-1]:
            print(research_area + ", ", end='')
        else:
            print(research_area + ".")
```

How can we modify the values?

Let say some professors have some new research interests, or need to delete some old interest. What can we do?
Let's illustrate.

In [ ]:

```
# Create a new and empty dictionary
professors = {}

# Fill it with some key:value pairs
# In here, key is the name (string) of a professor
#          value is a list, which contains research interests (list), rank (string) and age (integer)
professors['john c. s. lui'] =[['applied machine learning', 'network science', 'network economics', 'mobile
       'full professor', 82]
professors['patrick p. c. lee'] = [['file and storage systems', 'network monitoring'], 'associate professor
professors['james cheng'] = [['distributed systems', 'big data analytic systems', 'distributed machine lear
        'assistant professor', 18]
professors['eric lo'] = [['block chain', 'database systems', 'big data computing'], 'associate professor',


# Let say all professors have a new research interests of "A.I."
for name, a_list in professors.items():
    a_list[0].append('A.I.') # adding a new research topic to all professors in the dictionary
    research_interest = a_list[0]  # get research interest
    rank = a_list[1]               # get rank
    age  = a_list[2]               # get age
    print("\nName: " + name.title() + ", age = " + str(age) +  ", rank = " + rank.title() + ".")
    print("  "+"Research interests are: ", end='')
    for research_area in research_interest:
        if research_area != research_interest[-1]:
            print(research_area + ", ", end='')
        else:
            print(research_area + ".")

print('\n-----------------------------\n')

# Let say that AI is not hot anymore, and 'john c. s. lui' wants to drop it from his research interests.
professors['john c. s. lui'][0].remove('A.I.') # access the value with index 0 being list of research inter
print( professors['john c. s. lui'][0])    # print out the research interest of john c. s. lui
```

How can we modify keys?

To modify keys, it is a bit more complicated. Let's illustrate.

Assume we want to change "john c. s. lui" to "John C.S. Lui" (or changing the key).

In [ ]:
```
# Create a dictionary
professors = {}
professors['john c. s. lui'] =[['applied machine learning', 'network science', 'network economics', 'mobile
       'full professor', 82]
professors['patrick p. c. lee'] = [['file and storage systems', 'network monitoring'], 'associate professor
professors['james cheng'] = [['distributed systems', 'big data analytic systems', 'distributed machine lear
        'assistant professor', 18]
professors['eric lo'] = [['block chain', 'database systems', 'big data computing'], 'associate professor',

# Let's copy the "value" of the key 'john c. s. lui' to a new key 'John C.S. Lui'. Then delete the old key:

professors['John C.S. Lui'] = professors['john c. s. lui']  # create a new key:value pair
del professors['john c. s. lui']

# Let's print out the dictionary
for name, a_list in professors.items():
    a_list[0].append('A.I') # adding a new research topic to all professors in the dictionary
    research_interest = a_list[0]  # get research interest
    rank = a_list[1]               # get rank
    age  = a_list[2]               # get age
    print("\nName: " + name.title() + ", age = " + str(age) +  ", rank = " + rank.title() + ".")
    print("  "+"Research interests are: ", end='')
    for research_area in research_interest:
        if research_area != research_interest[-1]:
            print(research_area + ", ", end='')
        else:
            print(research_area + ".")
```

Looping through all items in a dictionary

We have seen this before. But let's use a "simple" example.
The **main idea** is to use *items()* so to obtain **all key-value pairs** from a dictionary into a **list**

In [ ]:
```
my_dictionary = {'key1':'key1_value', 'key2': 'key2_value', 'key3':'key3_value', 'key4':'key4_value'}

for key, value in my_dictionary.items():
    print('k:v par is ' + key + " " + value)
```

Looping through all keys in a dictionary

The **main idea** is to use *keys()* to get all keys from a disctionary into a **list**

In [ ]:
```
my_dictionary = {'key1':'key1_value', 'key2': 'key2_value', 'key3':'key3_value', 'key4':'key4_value'}

for key in my_dictionary.keys():
    print(key)
```

This is actually the **default behavior** of looping through the dictionary itself.
So you can leave out the *.keys()* part, and get the exact same behavior:

In [ ]:
```
my_dictionary = {'key1':'key1_value', 'key2': 'key2_value', 'key3':'key3_value', 'key4':'key4_value'}

for key in my_dictionary:
    print('for key='+ key +', its value is ' + my_dictionary[key])
```

In [ ]:

```
# Let's consider more examples
subjects = {'computer science': 'A study of the science and technology of computing.',
            'medicine': 'A study of biology, disease and its cure.',
            'BBA': 'A study of,... well, a not very serious subject.'
            }

# Show each subject in our dictionary.
print("The following subjects are in our dictionary:")
for subject in subjects:
    print("- %s" % subject)

# Allow the user to type in a word, and then display the meaning for that word.
requested_subject = input("\nWhat subject you are looking for? ")
if requested_subject in subjects.keys():
   print("\n%s: %s" % (requested_subject, subjects[requested_subject]))
else:
  print ("\nThe requested subject is NOT in the dictionary")
```

In [ ]:
```
# Let's modify the above program so that user can keep asking information

subjects = {'computer science': 'A study of the science and technology of computing.',
            'medicine': 'A study of biology, disease and its cure.',
            'BBA': 'A study of,... well, a not very serious subject.'
                }

# Show each subject in our dictionary.
print("The following subjects are in our dictionary:")
for subject in subjects:
    print("- %s" % subject)

# Allow the user to type in a word, and then display the meaning for that word.
requested_subject = input("\nWhat subject you are looking for? ") # ask input from user
while requested_subject != 'quit':
    if requested_subject in subjects.keys():
       print("\n%s: %s" % (requested_subject, subjects[requested_subject]))
    else:
       print ("\nThe requested subject is NOT in the dictionary")
    requested_subject = input("\nWhat subject you are looking for? ") # ask input from user
print('You have selected to quit.')
```

Looping through all values in a dictionary

The **main idea** is to use *values()* to get all keys from a disctionary into a **list**

In [ ]:
```
# define a dictionary

subjects = {'computer science': 'A study of the science and technology of computing.',
            'medicine': 'A study of biology, disease and its cure.',
            'economics': 'A study of the production, consumption, and transfer of wealth.',
            'theology': 'A study of the nature of God and religious belief.',
            'BBA': 'A study of,... well, a not very serious subject.'
                }

for value in subjects.values():
    print('values are:', value)
```

In [ ]:
```
# Let's process each word in each of the value

# define a dictionary
subjects = {'computer science': 'A study of the science and technology of computing.',
            'medicine': 'A study of biology, disease and its cure.',
            'economics': 'A study of the production, consumption, and transfer of wealth.',
            'theology': 'A study of the nature of God and religious belief.',
            'BBA': 'A study of,... well, a not very serious subject.'
                }
for value in subjects.values():       # access the sting in each value
    words_in_value = value.split( )  # separate each word in the string via the 'space' separator (now we h
    print("\nFor VALUE:", words_in_value, ", ", "\neach words are:")
    for word in words_in_value:
        print(word + ", ",  end='')
```

In [ ]:
```
# Let's write a program to allow user to "search" for keyword in each value

# define a dictionary
subjects = {'computer science': 'A study of the science and technology of computing.',
            'medicine': 'A study of biology, disease and its cure.',
            'economics': 'A study of the production, consumption, and transfer of wealth.',
            'theology': 'A study of the nature of God and religious belief.',
            'BBA': 'A study of,... well, a not very serious subject.'
                }

# Ask user to input keyword to search
keyword = input("\nPlease type in a keyword: ") # ask input from user
while keyword.upper() != 'QUIT':                # convert string to upper case
    found_flag = False
    for key, value in subjects.items():         # loop through all key/value pair
       value = value.replace(',', ' ')          # replace all comma and period so as to facilitate search
       value = value.replace('.', ' ')
       words_in_value = value.split()           # get a list of words in the value
       # print(words_in_value)
       if keyword in words_in_value:
           print("The subject you are looking for is ", key)
           found_flag = True

    if found_flag == False:
        print("Can't find the keyword in the dictionary")
    keyword = input("\nPlease type in a keyword: ")
```

How can we "add" items in a dictionary?

Let's illustrate

In [ ]:
```
# define a dictionary
subjects = {'computer science': 'A study of the science and technology of computing.',
            'medicine': 'A study of biology, disease and its cure.',
            'economics': 'A study of the production, consumption, and transfer of wealth.',
            'theology': 'A study of the nature of God and religious belief.',
            'BBA': 'A study of,... well, a not very serious subject.'
                }

subjects['chemistry'] = 'A study of the properties and characteristics of substances.'
subjects['philosophy'] = 'A study of the fundamental nature of knowledge, reality, and existence.'
for key, value in subjects.items():
    print('Key = ' + key +', value = ' + value )

print('Easy as pie !!!')

print ("Note that things may not be stored in our expected order")  # Illustrate the storage ordering
subjects
```

Lists in a dictionary

In dictionary, the key and value do not need to be of the same type. Let's consider a list as value

In [4]:
```
# define a score sheet for all students. Note that the lists don't have to have the same dimension !!!
score_sheet = {'john': [100, 100, 100, 100, 100],
               'peter': [99, 99, 99],
               'jack': [50,0, 50, 0]}

# Display their scores
print('For John, his scores are:', score_sheet['john'])
print('For Peter, his scores are:', score_sheet['peter'])
print('For Jack, his scores are:', score_sheet['jack'])
```

```
For John, his scores are: [100, 100, 100, 100, 100]
For Peter, his scores are: [99, 99, 99]
For Jack, his scores are: [50, 0, 50, 0]
```

In [7]:
```
# Let's put these print statements in a loop
# define a score sheet for all students
score_sheet = {'john': [100, 100, 100, 100, 100],
               'peter': [99, 99, 99],
               'jack': [50,0, 50, 0]}

# Display their scores
#for name in score_sheet.keys():        # Note that this also works !!!!!!
for name in score_sheet:
    print('For ' + name.title() + ', his scores are: ', score_sheet[name])
```

```
For John, his scores are:  [100, 100, 100, 100, 100]
For Peter, his scores are:  [99, 99, 99]
For Jack, his scores are:  [50, 0, 50, 0]
```

In [8]:
```
# Let's access each number within the list
# define a score sheet for all students
score_sheet = {'john': [100, 100, 100, 100, 100],
               'peter': [99, 99, 99],
               'jack': [50,0, 50, 0]}

# Display their scores
#for name in score_sheet.keys():        # Note that this also works !!!!!!
for name in score_sheet:
    print ('For ' + name.title() + ', his scores are: ', end='')
    for score in score_sheet[name]:
        print(str(score) + ', ',  end='')
    print("\n")
```

```
For John, his scores are: 100, 100, 100, 100, 100,

For Peter, his scores are: 99, 99, 99,

For Jack, his scores are: 50, 0, 50, 0,
```

Exercise

Modify the above program so that you can remove the annoyming "," and replace it with ".".

Dictionaries inside a dictionary

Let's consider how to nest a dictionary inside of a dictionary.

In [14]:
```
# This program stores information about pets. For each pet,
#   we store the kind of animal, the owner's name, and
#   the breed.
pets = {'willie': {'kind': 'dog', 'owner': 'eric', 'vaccinated': True},
        'walter': {'kind': 'cockroach', 'owner': 'jake', 'vaccinated': False},
        'peso': {'kind': 'dog', 'owner': 'mary', 'vaccinated': True},
        'john': {'kind': 'dog', 'owner': 'cuhk', 'vaccinated': False}
        }

# Let's show all the information for each pet.
print("Here is what I know about Willie:")
print("kind: " + pets['willie']['kind'])
print("owner: " + pets['willie']['owner'])
print("vaccinated: " + str(pets['willie']['vaccinated']))

print("\nHere is what I know about Walter:")
print("kind: " + pets['walter']['kind'])
print("owner: " + pets['walter']['owner'])
print("vaccinated: " + str(pets['walter']['vaccinated']))

print("\nHere is what I know about Peso:")
print("kind: " + pets['peso']['kind'])
print("owner: " + pets['peso']['owner'])
print("vaccinated: " + str(pets['peso']['vaccinated']))

print("\nHere is what I know about John:")
print("kind: " + pets['john']['kind'])
```

```
print("owner: " + pets['john']['owner'])
print("vaccinated: " + str(pets['john']['vaccinated']))
```

```
Here is what I know about Willie:
kind: dog
owner: eric
vaccinated: True

Here is what I know about Walter:
kind: cockroach
owner: jake
vaccinated: False

Here is what I know about Peso:
kind: dog
owner: mary
vaccinated: True

Here is what I know about John:
kind: dog
owner: cuhk
vaccinated: False
```

In [15]:
```
# Let's make the above code consice.
pets = {'willie': {'kind': 'dog', 'owner': 'eric', 'vaccinated': True},
        'walter': {'kind': 'cockroach', 'owner': 'eric', 'vaccinated': False},
        'peso': {'kind': 'dog', 'owner': 'chloe', 'vaccinated': True},
        'john': {'kind': 'dog', 'owner': 'cuhk', 'vaccinated': False}
        }

# Let's show all the information for each pet.
for pet_name, pet_information in pets.items():
    print("\nHere is what I know about %s:" % pet_name.title())
    print("kind: " + pet_information['kind'])
    print("owner: " + pet_information['owner'])
    print("vaccinated: " + str(pet_information['vaccinated']))
```

```
Here is what I know about Willie:
kind: dog
owner: eric
vaccinated: True

Here is what I know about Walter:
kind: cockroach
owner: eric
vaccinated: False

Here is what I know about Peso:
kind: dog
owner: chloe
vaccinated: True

Here is what I know about John:
kind: dog
owner: cuhk
vaccinated: False
```

In [16]:
```
# The above code won't work if we add more key/value into the dictionary (or the values of pets).
# Let's modify our code.

pets = {'willie': {'kind': 'dog', 'owner': 'eric', 'vaccinated': True},
        'walter': {'kind': 'cockroach', 'owner': 'eric', 'vaccinated': False},
        'peso': {'kind': 'dog', 'owner': 'chloe', 'vaccinated': True},
        'john': {'kind': 'dog', 'owner': 'cuhk', 'vaccinated': False}
        }

# Let's show all the information for each pet.
for pet_name, pet_information in pets.items():
    print("\nHere is what I know about %s:" % pet_name.title())
    # Each animal's dictionary is in 'information'
    for key in pet_information:
        print(key + ": " + str(pet_information[key]))
```

```
Here is what I know about Willie:
kind: dog
owner: eric
vaccinated: True

Here is what I know about Walter:
kind: cockroach
owner: eric
vaccinated: False

Here is what I know about Peso:
kind: dog
owner: chloe
vaccinated: True

Here is what I know about John:
kind: dog
owner: cuhk
vaccinated: False
```

In [ ]: