Data Structures
-- Dict and Flow Control

江琳

Dictionary

- A dictionary is like an address-book where you can find the address or contact details of a person by knowing only his/her name.

```
addr = {
          'William': 'william@hotmail.com',
          'Tracy': 'tracy123@163.com',
          'Lisa': 'lisa1990@qq.com',
        }
```

Dictionary

- **An empty dictionary without any items is written with just two curly braces, like this: {}.**

Dictionary

- **Keys are unique within a dictionary while values may not be. The values of a dictionary can be of any type, but the keys must be of an immutable data type such as strings, numbers, or tuples.**

It that correct?

```
info = {
          'name': 'William',
          17: 'age',
          True: True,
}
```

It that correct?

```
info = {
          'name': 'William',
          17: 'age',
          True: True,
          (1, 2): 'Tuple',
}
```

It that correct?

```
info = {
          'name': 'William',
          17: 'age',
          True: True,
          (1, 2): 'Tuple',
          [1, 2]: 'List',
}
```

It that correct?

```
info = {
          'name': 'William',
          17: 'age',
          True: True,
          (1, 2): 'Tuple',
          [1, 2]: 'List',
          (1, [1, 2]): 'Tup',
}
```

Accessing Values in Dictionary

- To access dictionary elements, you can use the familiar square brackets along with the key to obtain its value.

```
info = {'name': 'William', 'age': 17}

print('info["name"]: ', info['name'])

print('info["age"]: ', info['age'])
```

Accessing Values in Dictionary

- If we attempt to access a data item with a key, which is not part of the dictionary:

```
dict['address']

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

KeyError: 'address'
```

Update and add Values in Dictionary

```
info = {'name': 'William', 'age': 17}

# update value

info['age'] = 19

# add new item

info['address'] = 'Spring Garden Road'

print(info)
```

Build-in methods

dict.get(key, default=None)	For key key, returns value or default if key not in dictionary
dict.has_key(key)	Returns true if key in dictionary dict, false otherwise
dict.items()	Returns a list of dict's (key, value) tuple pairs
dict.keys()	Returns list of dictionary dict's keys

Build-in methods

```
info = {'name': 'William', 'age': 17}

print (info.get('name'))

print (info.items())

print (info.keys())
```

Data Structures
-- Dict and Flow Control

江琳

Flow Control Statement

- Decision making
  - if … else ..
- Loop
  - for loop
  - while loop

Decision making

Statement	Description
if statement	An if statement consists of a boolean expression followed by one or more statements.
if…else statement	An if statement can be followed by an optional else statement, which executes when the boolean expression is FALSE.
nested if statement	You can use one if or else if statement inside another if or else if statement(s).

Decision making

```
score = 89
if score > 90:
    print('A')
elif score > 75:
    print('B')
elif score > 60:
    print('C')
else:
    print('fail')
```

While loop

```
sum = 0

x = 1

while x <= 100:

    print(x)

    x += 1
```

For loop

```
list1 = ['physics', 'chemistry', 'history' , 'biology']
for subject in list1:
    print(subject)


tuple1 = ('physics', 'chemistry', 'history' , 'biology')
for subject in tuple1:
    print(subject)
```

For loop with dictionary

- Try the code below:

```
addr = {
'William': 'william@hotmail.com',
'Tracy': 'tracy123@163.com',
'Lisa': 'lisa1990@qq.com',
}
for item in addr:
    print(item)
```

For loop with dictionary

- **The result is:**

```
William
Tracy
Lisa
```

Only print the key of dictionary, where is the value?

For loop with dictionary

- Dictionary.items()

```
addr = {
'William': 'william@hotmail.com',
'Tracy': 'tracy123@163.com',
'Lisa': 'lisa1990@qq.com',
}
for item in addr.items():
    print(item)
```

For loop with dictionary

- **The result is:**

```
('William', 'william@hotmail.com')
('Tracy', 'tracy123@163.com')
('Lisa', 'lisa1990@qq.com')
```

What is output‘s data type?

For loop with dictionary

```
addr = {
'William': 'william@hotmail.com',
'Tracy': 'tracy123@163.com',
'Lisa': 'lisa1990@qq.com',
}
for k, v in addr.items():
    print(k, v)
```

range()

- range() is used to generates a list of number

```
list(range(5))
# result:
# [0,1,2,3,4]


list(range(1,6))
# result:
# [1,2,3,4,5]
```

range()

```
for i in range(5):
    print(i)
# result
# 0
# 1s
# 2
# 3
# 4
```

Questions?

Tutorial 2

Exercise 1

- **Ask the user for a number. Depending on whether the number is even or odd, print out an appropriate message to the user.**

1. **If the number is a multiple of 4, print out a different message.**
2. **Ask the user for two numbers: one number to check (call it num) and one number to divide by (check). If check divides evenly into num, tell that to the user. If not, print a different appropriate message.**

Exercise 2

- **Take a list: a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]**

1. **write a program that prints out all the elements of the list that are less than 5.**
2. **Instead of printing the elements one by one, make a new list that has all the elements less than 5 from this list in it and print out this new list.**

Exercise 3

- **Create a program that asks the user for a number and then prints out a list of all the divisors of that number. (If you don't know what a divisor is, it is a number that divides evenly into another number. For example, 13 is a divisor of 26 because 26 / 13 has no remainder.)**

Exercise 4

- **Take two lists, say for example these two:**
  - **a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]**
  - **b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]**
- **write a program that returns a list that contains only the elements that are common between the lists (without duplicates). Make sure your program works on two lists of different sizes.**