2. Container types in Python

- Containers are objects that contain inner elements. We saw 2 such objects so far: str and list.
- There are other useful containers in Python. We can classify them by order and by mutability. Here are the common ones:

	ordered (sequences)	unordered
mutable	list [1,2,3]	set {1,2,3} dict {1:"a", 2:"b", 3:"c"}
immutable	str "abc" tuple (1,2,3)	

Tuples vs. Lists

- Tuples are much like lists, but syntactically they are enclosed in regular brackets, while lists are enclosed in square brackets.

```
>>> a = (2,3,4)
>>> b = [2,3,4]
>>> type(a)
<class 'tuple'>
>>> type(b)
<class 'list'>
>>> a[1], b[1]
(3, 3)
>>> [a[i]==b[i] for i in range(3)]
[True, True, True]
>>> a==b
False
```

- Tuples are much like lists, only they are immutable.

```
>>> b[0] = 0 # mutating the list
>>> a[0] = 0 # trying to mutate the tuple
Traceback (most recent call last):
    File "<pyshell#30>", line 1, in <module>
    a[0]=0
TypeError:  'tuple' object does not support item assignment
```

Using tuples for function return Values

A function can return more than a single value. For example

```
>>> def mydivmod(a,b):
    ''' integer quotient and remainder of a divided by b '''
    return a//b, a%b
```

When executing this function, we get back two (integer) values, "packed" in a tuple.

```
>>> mydivmod(21,5)
(4, 1)
>>> mydivmod(21,5)[0]
4
>>> type(mydivmod(21,5))
<class 'tuple'>
```

Incidentally, the returned values can simultaneously assigned:

```
>>> d,r = mydivmod(100,7)
>>> d
14
>>> r
2
>>> 7*14+2
100
```

Dictionaries

- Dictionaries contain pairs of elements key:value. They are used as mapping between a set of keys and a set of elements.
- Keys cannot repeat (i.e. they are unique), and must be immutable

```
>>> d = {"France":"Europe", "Genrmay":"Europe", "Japan":"Asia"}
>>> type(d)
<class 'dict'>
>>> d #order of elements not necessarily as defined in initialization
{'Germany':'Europe', 'France':'Europe', 'Japan':'Asia'}
>>> d["Japan"]
'Asia'
>>> d["Israel"]
Traceback (most recent call last):
File "<pyshell#1>", line 1, in <module>
d["Israel"]
KeyError:  'Israel'
>>> d["Egypt"] = "Africa"
>>> d
{'Germany':'Europe', 'France':'Europe', 'Egypt':'Africa', 'Japan':'Asia'}
```

Sets

- Sets are like dictionaries, but contain elements rather than pairs of key:val.
- In fact they resemble the mathematical notion of a set
- Some examples - in class

```
>>> s = {1,2,3,"a"}
```

Each type in Python supports various operations. For example, str has an operation called title (str.title), list supports count (list.count), etc.
Dictionaries, sets and tuples also support their own operations - we will introduce them when we need them. But don't wait for us!