Containers

python共有lists,dictionaries,sets和tuples

列表Lists

列表就是Python中的数组，但是列表长度可变，且能包含不同类型元素。

```
xs = [3, 1, 2]    # Create a list
print(xs, xs[2])  # Prints "[3, 1, 2] 2"
print(xs[-1])     # Negative indices count from the end of the list; prints "2"
xs[2] = 'foo'     # Lists can contain elements of different types
print(xs)         # Prints "[3, 1, 'foo']"
xs.append('bar')  # Add a new element to the end of the list
print(xs)         # Prints "[3, 1, 'foo', 'bar']"
x = xs.pop()      # Remove and return the last element of the list
print(x, xs)      # Prints "bar [3, 1, 'foo']"
```

切片slicing

为了一次性地获取列表中的元素，Python提供了一种简洁的语法，这就是切片。

```
nums = range(5)    # range is a built-in function that creates a list of integers
print(nums)        # Prints "[0, 1, 2, 3, 4]"
print(nums[2:4])   # Get a slice from index 2 to 4 (exclusive); prints "[2, 3]"
print(nums[2:])    # Get a slice from index 2 to the end; prints "[2, 3, 4]"
print(nums[:2])    # Get a slice from the start to index 2 (exclusive); prints "[0, 1]"
print(nums[:])     # Get a slice of the whole list; prints ["0, 1, 2, 3, 4]"
print(nums[:-1])   # Slice indices can be negative; prints ["0, 1, 2, 3]"
nums[2:4] = [8, 9] # Assign a new sublist to a slice
print(nums)        # Prints "[0, 1, 8, 9, 4]"
```

> 切片中的:不包括后面的数字

循环Loops

```
animals = ['cat', 'dog', 'monkey']
for animal in animals:
    print(animal)
# Prints "cat", "dog", "monkey", each on its own line.
```

如果想要在循环体内访问每个元素的指针，可以使用内置的**enumerate**函数

```
animals = ['cat', 'dog', 'monkey']
for idx, animal in enumerate(animals):
```

```
    print('#%d: %s' % (idx + 1, animal)
# Prints "#1: cat", "#2: dog", "#3: monkey", each on its own line
```

列表推导List comprehensions

在编程的时候，我们常常想要将一种数据类型转换为另一种。

```
nums = [0, 1, 2, 3, 4]
squares = [x ** 2 for x in nums]
print(squares)   # Prints [0, 1, 4, 9, 16]
```

列表推导还可以包含条件：

```
nums = [0, 1, 2, 3, 4]
even_squares = [x ** 2 for x in nums if x % 2 == 0]
print(even_squares)  # Prints "[0, 4, 16]"
```

字典

字典用来储存（键, 值）对，这和Java中的Map差不多。

```
d = {'cat': 'cute', 'dog': 'furry'}  # Create a new dictionary with some data
print(d['cat'])       # Get an entry from a dictionary; prints "cute"
print('cat' in d)     # Check if a dictionary has a given key; prints "True"
d['fish'] = 'wet'    # Set an entry in a dictionary
print(d['fish'])      # Prints "wet"
# print d['monkey']  # KeyError: 'monkey' not a key of d
print(d.get('monkey', 'N/A'))  # Get an element with a default; prints "N/A"
print(d.get('fish', 'N/A'))    # Get an element with a default; prints "wet"
del d['fish']        # Remove an element from a dictionary
print(d.get('fish', 'N/A')) # "fish" is no longer a key; prints "N/A"
```

循环Loops

```
d = {'person': 2, 'cat': 4, 'spider': 8}
for animal in d:
    legs = d[animal]
    print('A %s has %d legs' % (animal, legs))
# Prints "A person has 2 legs", "A spider has 8 legs", "A cat has 4 legs"
```

如果你想要访问键和对应的值，那就使用iteritems方法：

```
d = {'person': 2, 'cat': 4, 'spider': 8}
for animal, legs in d.iteritems():
    print('A %s has %d legs' % (animal, legs))
```

```
# Prints "A person has 2 legs", "A spider has 8 legs", "A cat has 4 legs"
```

字典推导Dictionary comprehensions

和列表推导类似，但是允许你方便地构建字典。

```
nums = [0, 1, 2, 3, 4]
even_num_to_square = {x: x ** 2 for x in nums if x % 2 == 0}
print(even_num_to_square)  # Prints "{0: 0, 2: 4, 4: 16}"
```

集合Sets

集合是独立不同个体的无序集合。

```
animals = {'cat', 'dog'}
print('cat' in animals)   # Check if an element is in a set; prints "True"
print('fish' in animals)  # prints "False"
animals.add('fish')       # Add an element to a set
print('fish' in animals)  # Prints "True"
print(len(animals))       # Number of elements in a set; prints "3"
animals.add('cat')        # Adding an element that is already in the set does nothing
print(len(animals))       # Prints "3"
animals.remove('cat')     # Remove an element from a set
print(len(animals))       # Prints "2"
```

循环Loops

在集合中循环的语法和在列表中一样，但是集合是无序的，所以你在访问集合的元素的时候，不能做关于顺序的假设。

```
animals = {'cat', 'dog', 'fish'}
for idx, animal in enumerate(animals):
    print('#%d: %s' % (idx + 1, animal))
# Prints "#1: fish", "#2: dog", "#3: cat"
```

集合推导Set comprehensions

和字典推导一样，可以很方便地构建集合。

```
from math import sqrt
nums = {int(sqrt(x)) for x in range(30)}
print nums  # Prints "set([0, 1, 2, 3, 4, 5])"
```

元组Tuples

元组是一个值的有序列表（不可改变）。从很多方面来说，元组和列表都很相似。和列表最重要的不同在于，元组可以在字典中用作键，还可以作为集合的元素，而列表不行。

```
d = {(x, x + 1): x for x in range(10)}  # Create a dictionary with tuple keys
print(d)
t = (5, 6)        # Create a tuple
print(type(t))    # Prints "<type 'tuple'>"
print(d[t])       # Prints "5"
print(d[(1, 2)])  # Prints "1"
```