字典的操作

黄天羽

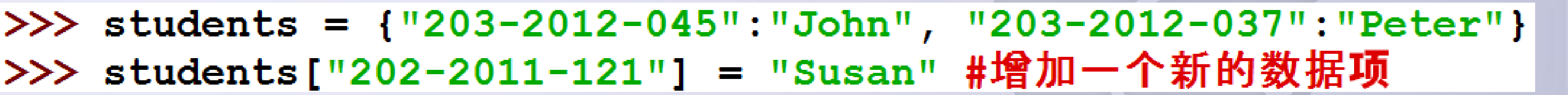
北京理工大学

python 语言程序设计

为字典增加一项

- dictionaryName[key] = value

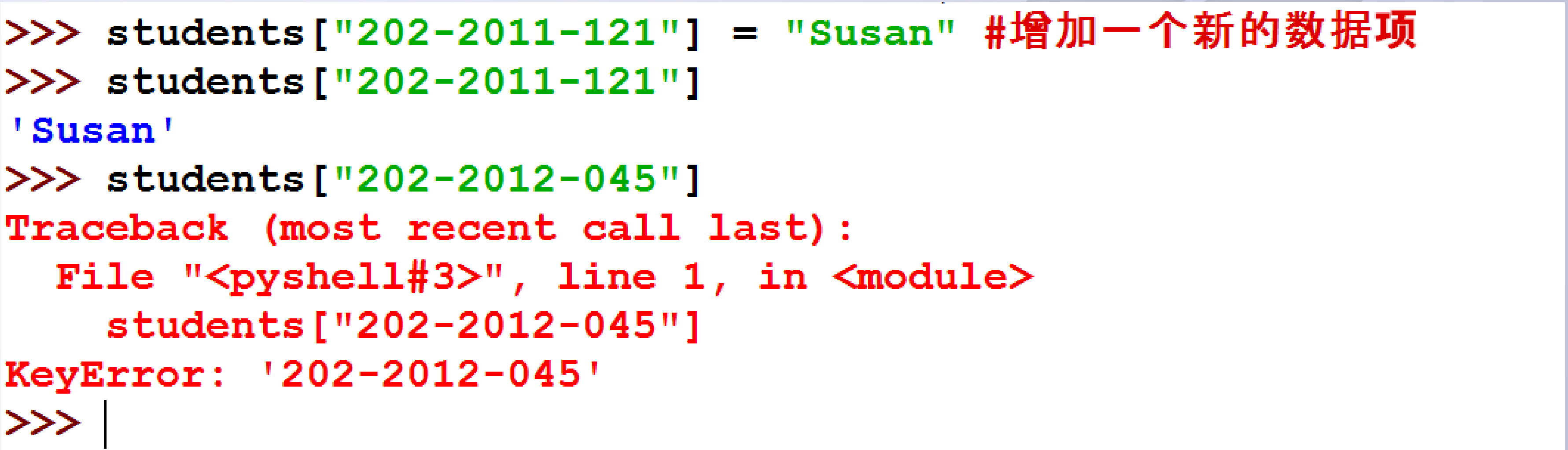
- 举例：



```
>>> students = {"203-2012-045":"John", "203-2012-037":"Peter"}
>>> students["202-2011-121"] = "Susan" #增加一个新的数据项
```

访问字典中的值

- dictionaryName[key] 返回键key对应的值value
- 举例：

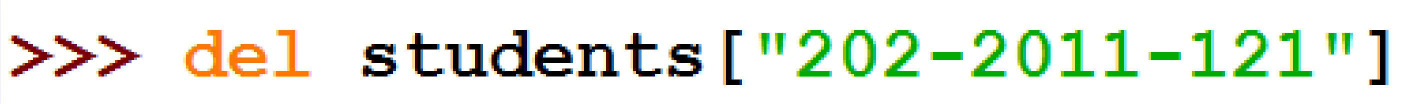


```
>>> students["202-2011-121"] = "Susan" #增加一个新的数据项
>>> students["202-2011-121"]
'Susan'
>>> students["202-2012-045"]
Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    students["202-2012-045"]
KeyError: '202-2012-045'
>>> 
```

删除字典中的一项

- del dictionaryName[key]

- 举例：



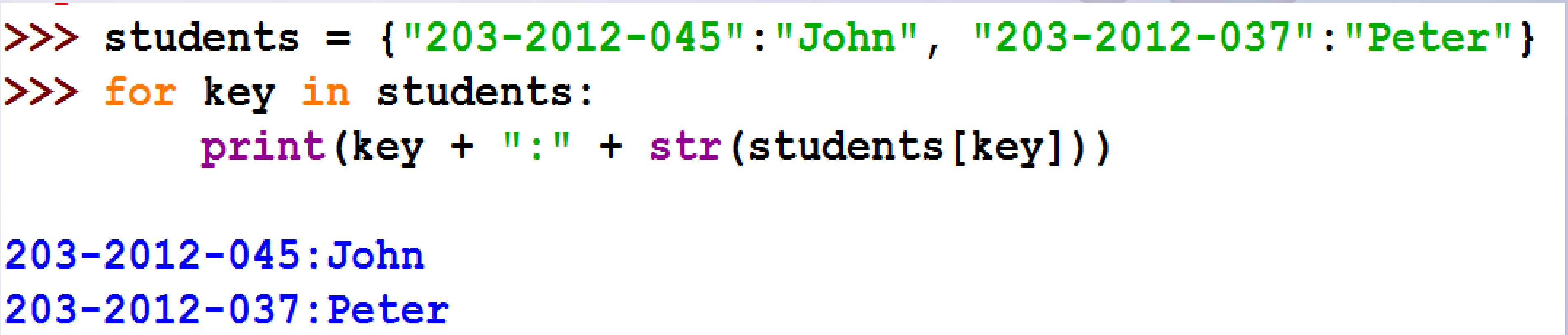
```
>>> del students["202-2011-121"]
```

字典的遍历

- for key in students:

  print (key + “:” + str(stuendents[key]))

- 举例



```
>>> students = {"203-2012-045":"John", "203-2012-037":"Peter"}
>>> for key in students:
	print(key + ":" + str(students[key]))

203-2012-045:John
203-2012-037:Peter
```

字典的遍历

- 遍历字典的键key

  for key in dictionaryName.keys(): print.(key)

- 遍历字典的值value

  for value in dictionaryName.values(): print.(value)

- 遍历字典的项

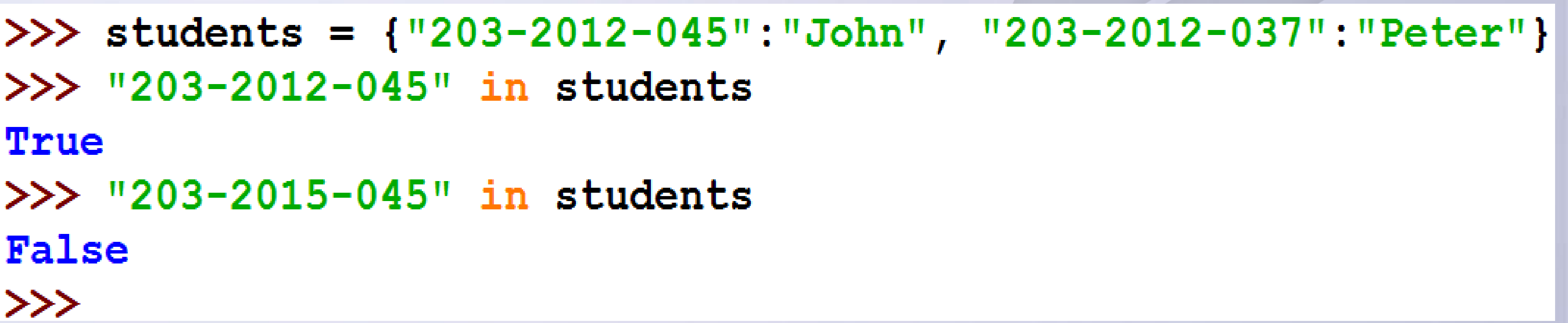
  for item in dicitonaryName.items(): print.(item)

- 遍历字典的key-value

  for item，value in adict.items(): print(item, value)

是否一个键在字典中

- in 或者 not in
- 举例

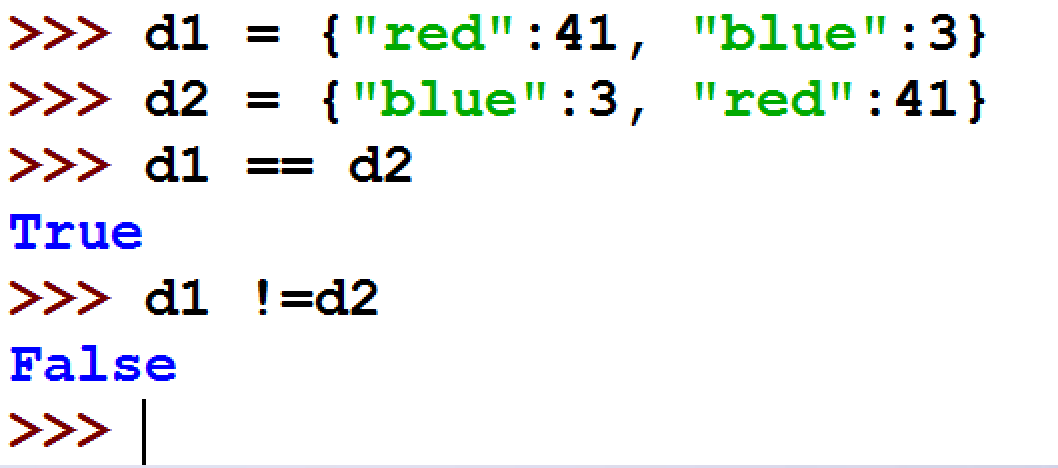


```
>>> students = {"203-2012-045":"John", "203-2012-037":"Peter"}
>>> "203-2012-045" in students
True
>>> "203-2015-045" in students
False
>>>
```

字典的标准操作符

- - , <,>,<=,>=,==,!=,and, or, not

- 字典相等测试

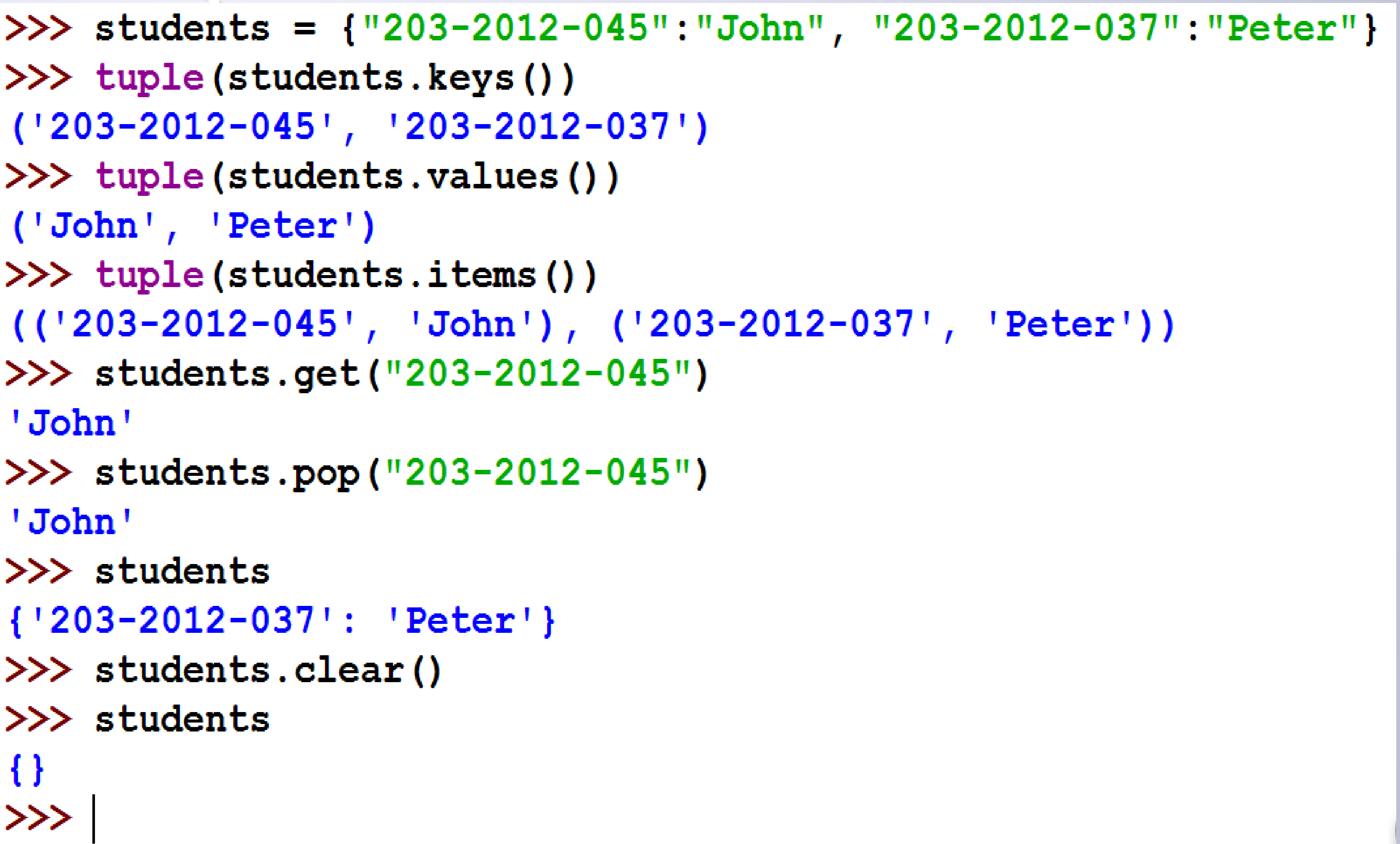


```
>>> d1 = {"red":41, "blue":3}
>>> d2 = {"blue":3, "red":41}
>>> d1 == d2
True
>>> d1 !=d2
False
>>> 
```

Python还提供了丰富的字典方法,其中 ：

keys():tuple	返回一个包含字典所有Key的列表
values():tuple	返回一个包含字典所有value的列表
Items():tuple	返回一一个包含所有键值的列表
clear():None	删除字典中的所有项目
get(key):value	返回字典中key对应的值
pop(key):val	删除并返回字典中key对应的值
update(字典)	将字典中的键值添加到字典中

字典的方法



```
>>> students = {"203-2012-045":"John", "203-2012-037":"Peter"}
>>> tuple(students.keys())
('203-2012-045', '203-2012-037')
>>> tuple(students.values())
('John', 'Peter')
>>> tuple(students.items())
(('203-2012-045', 'John'), ('203-2012-037', 'Peter'))
>>> students.get("203-2012-045")
'John'
>>> students.pop("203-2012-045")
'John'
>>> students
{'203-2012-037': 'Peter'}
>>> students.clear()
>>> students
{}
>>> 
```