Lecture 2　Online Quiz (Key Points)

金融科技协会　2018 年 11 月 14 日

一　List

1.1　第 10.7 题

random 模块中的 **shuffle()** 方法将序列的所有元素随机排序。注意，需要 **import random** 后才能使用 **random.shuffle**。

1.2　第 10.13 题

如果两个列表包含**同样类型**的元素，则可以使用比较运算符 (>, >=, <, <=, ==, **!**=) 对两个列表进行比较，比较的方法是：

首先分别比较第一个元素，如果它们不相同，就决定了比较的结果；如果相同，那就继续比较接下来的两个元素。一直重复此过程，直到比较完所有元素。

1.3　第 10.20 题

列表的 **sort()** 方法默认**升序**排列。

1.4　第 10.25 题

字符串的 **split()** 方法有默认参数，默认以**空格**为分隔符对字符串进行分割，返回列表。

1.5　第 10.32 题

不要将**可变对象** (mutable types) 设置为默认参数——

”Default values are computed once, then re-used.”

”Default parameter values are evaluated when the function definition is executed. This means that the expression is evaluated once, when the function is defined, and that the same “pre-computed” value is used for each call.”

函数也是对象。

每当函数定义完毕的时候 (即 python 执行 def 语句的时候)，python 就会创建一个函数对象，而函数的默认参数，就是这个函数对象的一个属性，即——默认参数相当于函数对象的一个属性。

因此，如果默认参数是可变对象，那么对默认参数进行的修改，都会被该函数对象“捕获”到，并“保存”起来，其实质是：函数对象的一个属性发生了改变，原因是该属性值 (默认参数的值) 是可变对象，可以就地修改。

将以下代码执行，并比较结果的差异：

```python
def f(i, values = []):
    values.append(i)
    return values

f(1)
f(2)
v = f(3)
print(v)

# [1, 2, 3]
```

图 1: 返回 [1,2,3]

```python
def f(i, values = []):
    values.append(i)
    return values

f(1)

def f(i, values = []):
    values.append(i)
    return values

f(2)
v = f(3)
print(v)

# [2, 3]
```

图 2: 返回 [2,3]

```python
def f(i, values = []):
    values.append(i)
    return values

f(1)

def f(i, values = []):
    values.append(i)
    return values

f(2)

def f(i, values = []):
    values.append(i)
    return values

v = f(3)
print(v)

# [3]
```

图 3: 返回 [3]

二 Tuples, Sets, and Dictionaries

2.1 第 14.8 题

{} 的类型是**字典**而非**集合**。

```
>>> type({})
<class 'dict'>
>>>
>>>
```

图 4: {} 是字典而非集合

2.2 第 14.9 题

对集合使用**比较运算符**的规则：

第一，== 和 != 检测两个集合是否包含相同的元素，即集合是否相等；

第二，如果 **s1** 是 **s2** 的一个真子集，则 **s1<s2** 和 **s2>s1** 都将返回 **True**；

第三，如果 **s1** 是 **s2** 的一个子集，则 **s1<=s2** 和 **s2>=s1** 都将返回 **True**。

2.3 第 14.23 题

乘号不能作用于集合。

```
>>> a = {1,2}
>>> a * 2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for *: 'set' and 'int'
>>>
>>>
```

图 5: 乘号不能作用于 set

2.4 第 14.24 题

加号也不能作用于集合。

```
>>> s1 = {1,2,3}
>>> s2 = {4,5,6}
>>> s1 + s2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for +: 'set' and 'set'
>>>
>>>
```

图 6: 加号不能作用于 set

2.5 第 14.28 题

可以使用 == 或 != 对两个字典进行相等性检测，但不能使用其他比较运算符对字典进行比较，没有意义。

2.6 第 14.31 题

使用 **del** 删除字典中的某个 **key-value**，语法是 **del dictName[keyName]**。

2.7 第 14.36 题

A Python tuple is immutable if every element in the tuple is immutable. 下例：

```
>>> list1 = [1,2,3]
>>> list2 = [4,5,6]
>>> T = (list1, list2)
>>> T
([1, 2, 3], [4, 5, 6])
>>> list1.append(999)
>>>
>>> T
([1, 2, 3, 999], [4, 5, 6])
>>>
```

图 7: tuple 中的元素为可变对象

三 Selections

3.1 第 4.4 题

random.randrange(a,b) 产生一个在 **a** 到 **b-1** 之间的随机整数，等价于 **randint(a, b-1)**；而 **random.random()** 返回一个 **0.0**到 **1.0**之间 (**不包括 1.0**) 的随机浮点数。

3.2 第 4.18 题

sys.exit()，运行到此行代码时，程序退出。但可以设置一个参数——默认为 **0**，即 **sys.exit()** 等价于 **sys.exit(0)**，此时程序正常退出，不抛出任何异常；设置为其他数值，

如 **sys.exit(9)**，则该语句会抛出一个 **SystemExit** 异常，然后终止——当然，可以使用 **try/except** 语句对该异常进行捕获，使得程序能够继续正常运行。

3.3 第 4.19 题

Python 允许这种写法 $-10 < x < 10$，等价于 $x > -10$ and $x < 10$。

3.4 第 4.33 题

关于运算顺序，**and** 先运算，**or** 后运算。

四 Loops

4.1 第 5.17 题

在**循环继续条件**中使用**浮点数**可能会导致**数值错误**。这是由于 python 在存储浮点数的时候会有**精度损失**问题，如下：

图 8: 精度损失

因此避免使用浮点数进行循环继续条件的计算，尽量使用整数。

4.2 第 5.18 题

大小相仿的浮点数相加，精度保护地好一些。

By Draymond