

Python任务练习

1 Python变量和数据类型

- 1. 表达式 `type(a) == type(b)` 和 `type(a) is type(b)` 区别是什么?
- 2. Python3中表达式 `1.1 + 2.2 == 3.3` 执行结果是 `True` 还是 `False` ? 为什么?
- 3. Python3中表达式 `0.1 + 0.1 + 0.1 - 0.3` 的结果是多少?
- 4. 给出一种方法, 针对第3题中的表达式执行精确计算。

2 Python字符串和列表

实现一个函数, 对于给定的一个字符串, 比如'aaabeeeezzzxxxxw', 将其中连续出现的字符转换成该字符和它重复出现的次数的组合并将结果返回, 比如对于上面字符串, 返回结果为'a3b1e4z3x5w1'。假定要编码的字符串只包含26个字母, 没有数字, 同时编写解码算法代码, 假定要解码的字符串是合法编码过的。

注意: 须附上完整代码, 并给出测试数据和代码运行结果。

3 Python列表

- 1. 实现一个函数`get_list_elements()`, 对于给定的列表, 可以根据需要返回列表中指定位置的多个元素。比如, 对于列表 `fruits = ['apple', 'orange', 'watermelon', 'banana', 'pear']`, 可以如下获取其中的多个元素:

```
>>> retrieved_fruits = get_list_elements(fruits, (0, 3, 4))
>>> retrieved_fruits
['apple', 'banana', 'pear']
```

获取列表元素时注意处理`IndexError`错误, 当获取的列表中指定位置的元素不存在时返回`None`, 比如:

```
>>> retrieved_fruits = get_list_elements(fruits, (0, 3, 5))
>>> retrieved_fruits
['apple', 'banana', None]
```

- 2. 对于下表中的省份和对应的省会, 编写一个省会问答小游戏, 程序运行后会提示用户回答某个省份的省会在哪里, 当用户回答正确时, 继续下一个提问。如果用户回答错误, 须告知用户, 并给出正确答案。每次提问选择的省份是任意的, 但要求相邻两次提问选择的省份不能重复。当用户输入q或Q的时候, 结束问答, 给出用户回答情况统计 (用户回答正确、错误的数量), 然后退出程序。

省份	省会/首府
河北省	石家庄
山西省	太原
辽宁省	沈阳
吉林省	长春
黑龙江省	哈尔滨
江苏省	南京
浙江省	杭州
安徽省	合肥
福建省	福州

省份	省会/首府
江西省	南昌
山东省	济南
河南省	郑州
广东省	广州
湖南省	长沙
湖北省	武汉
海南省	海口
四川省	成都
贵州省	贵阳
云南省	昆明
陕西省	西安
甘肃省	兰州
青海省	西宁
台湾省	台北
内蒙古自治区	呼和浩特
广西壮族自治区	南宁
西藏自治区	拉萨
宁夏回族自治区	银川
新疆维吾尔自治区	乌鲁木齐

4 编程练习

1. 建立一个用户类（User）和系统类(System)，来表示一个系统当中注册的用户。用户注册时需提供用户名（username）、密码（password）、邮箱地址（email）3种信息，当用户完成注册（register）后，会将用户信息添加到系统当中。当注册用户时，会检查其提供的用户名和邮箱地址是否已经被其他用户使用过，如果已经被使用，则提示用户并告知其注册失败。一个用户的密码和邮箱地址无法被直接查看，但是可以被修改（modify）和验证（verify）。另外，用户信息中还有一个状态（status）字段来表示用户是否已登录。当用户通过用户名和密码（或者邮箱地址和密码）完成登录验证（login）后，会改变用户的状态为已登录。当用户退出（logout）时，会将用户的状态设置为未登录。

根据上面的描述，建立2个类——系统类（System）和用户类（User），实现用户注册、登录、修改用户信息，退出登录等功能，并提供测试示例，模拟上面描述的一系列操作。

2. 结合本次任务中的第1题及上次任务中编写的省会问答小游戏，请编写代码，实现用户登录之后就能玩游戏的命令行程序，当用户处于未登录状态时，则不能玩，并提示用户登录。（可以将问答游戏作为系统提供的一种服务，要求用户必须要注册并登录之后才能使用这项服务。）

5 有道翻译小工具

编写一个命令行工具，利用有道词典（<http://dict.youdao.com>）完成中英翻译功能。

实现思路：发送请求并从获取到的数据中解析翻译结果。

使用方法：

```
./youdao.py "Hello, world!"  
./youdao.py "你好，世界！"
```

6 Python练习题

1. 下面的代码输出内容是什么？请解释你的答案。

```
def extend_list(val, list=[]):
    list.append(val)
    return list

list1 = extend_list(10)
list2 = extend_list(123, [])
list3 = extend_list('a')

print("list1 = %s" % list1)
print("list2 = %s" % list2)
print("list3 = %s" % list3)
```

如何修改函数 `extend_list` 的定义，使其产生复合预期的行为？

2. 下面的代码在Python3中输出是什么？请解释你的答案。

```
def div1(x, y):
    print("%s / %s = %s" % (x, y, x / y))

def div2(x, y):
    print("%s // %s = %s" % (x, y, x // y))

div1(5, 2)
div1(5., 2)
div2(5, 2)
div2(5., 2.)
```

3. 思考下面的代码片段：

```
1. list = [ [ ] ] * 5
2. list # output?
3. list[0].append(10)
4. list # output?
5. list[1].append(20)
6. list # output?
7. list.append(30)
8. list # output?
```

第2、4、6、8行代码的输出是什么？请解释你的答案。

4. 下面的代码输出内容是什么？请解释你的答案。

```
def multipliers():
    return [lambda x: i * x for i in range(4)]

print([m(2) for m in multipliers()])
```

7 Python练习题2

1. 填空

```
>>> a, b = ???
>>> a + b == b + a
False
```

??? 是什么，可以让这段代码成立？

2. 编写函数，根据单词的长度给一个列表排序（从小到大排），列表全由字符串元素构成，比如 `fruits = ['apple', 'banana', 'pear', 'raspberry', 'strawberry']`，返回排序后的结果。
3. 实现快速排序算法，对给定的由整数构成的列表进行排序，自己给出测试结果。

8 Python练习题3

1. 写出下面代码中 X0, X1, ...Xn 的最终的值：

```
X0 = dict(zip(('a', 'b', 'c', 'd', 'e'), (1, 2, 3, 4, 5)))
X1 = range(10)
X2 = sorted([i for i in X1 if i in X0])
X3 = sorted([X0[s] for s in X0])
X4 = [i for i in X1 if i in X3]
X5 = {i: i * i for i in X1}
X6 = [[i, i * i] for i in X1]
```

2. 实现一个函数，生成包含大写字母和数字的随机字符串，接收一个参数 N 来指定字符串的长度，每次调用生成指定长度的随机字符串，比如当 N = 6 时，输出示例如下：

- 6U1S75
- 4Z4UKK
- U911K4

3. 实现一个函数，把列表分割成同样大小的块，接受2个参数，第1个参数为想要分割的列表或可迭代对象，第2个参数指定分割的块的大小。比如：

```
l = range(1, 1000)
print(chunks(l, 10)) -> [[1..10], [11..20], ..., [991..999]]
```

4. 实现一个函数，根据字典中值来排序字典。
5. 实现一个函数，合并两个有序列表。（不使用标准库模块）

9 Python练习题4

1. 实现函数，返回字符串中各单词出现的频率（以字典形式，不使用标准库）
2. 实现函数，接受一个字典参数，返回将字典的值和键调换之后的新字典，比如对于下面的字典

```
cities = {
    'shijiazhuang': 'hebei',
    'tangshan': 'hebei',
    'zhengzhou': 'henan',
    'jinan': 'shandong',
    'zibo': 'shandong',
    'qingdao': 'shandong',
}
```

返回结果：

```
{
    'hebei': ['shijiazhuang', 'tangshan'],
    'henan': 'zhengzhou',
    'shandong': ['jinan', 'zibo', 'qingdao']
}
```

3. 实现函数，使用列表初始化字典，比如对于下面的列表：

```
L = [3, 7, 4, 8, 23, 41, 78, 166, 312, 604, 1218, 2156, 51112, 103524, 3122768, 6552236, 4334294967296]
```

返回一个字典，其中字典的键为列表中数字的长度，对应的键的值为列表中相应长度的数字。比如，对于上面的列表返回结果如下：

```
{
  1: [3, 7, 4, 8],
  2: [23, 41, 78],
  3: [166, 312, 604],
  4: [1218, 2156],
  5: [51112],
  6: [103524],
  7: [3122768, 6552236],
  13: [4334294967296]
}
```

4. 写一个装饰器函数，将所装饰的函数的参数的值和类型打印出来，如果函数的参数是关键字参数，将关键字参数的名字一起打印。

5. 写一个函数，接受一个字符串，并检查字符串是否满足下列条件：

1. 至少包含一个小写字母
2. 至少包含一个数字
3. 至少包含一个大写字母
4. 至少包含\$、#、@中的一个字符
5. 字符串最小长度为6个字符
6. 字符串最大长度为16个字符
7. 字符串中不得包含空白

如果字符串满足上述所有条件，返回 True ， 否则返回 False 。