

第三周 《数据获取与表示》

In-video quiz

- ◆ 如果需要在文本文件 `test.txt` 的中间部分添加一行字符串，以下 `open` 语句的 `mode` 位置应该填什么？ 答案 `r+`

```
fp = open('test.txt', '_____')
```

解析：w 和 w+ 都需要清空原文件内容，a 和 a+ 都只能追加，也就是在文件尾部添加内容

- ◆ 如果文件 `data.txt` 中的内容是 “hello, world”，则 `p2` 的输出结果是什么？ 答案 B

```
>>> f = open('data.txt')
>>> p1 = f.read()
>>> p2 = f.read()
>>> f.close()
```

- (A) hello, world
- (B) 输出为空，因为文件指针已经读到文件尾部
- (C) hello
- (D) world

解析：注意文件指针，可以用 `f.tell()` 查看，执行完 `f.read()` 后文件指针已到文件的末尾，所以再次执行 `f.read()` 则无输出

- ◆ 已知有序列 `week = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']`，以下切片操作中可以获得序列 `['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday']` 的有哪些？ 答案 AC

- (A) `week[0:5]`
- (B) `week[0:4]`
- (C) `week[-7:-2]`
- (D) `week[-7:-3]`

解析：B 只包含下标从 0 到 3 的元素，D 只包含下标从 -7 到 -4 的元素

- ◆ 对于一个列表 `list1`，请利用 `help` 函数确定若要根据列表各元素长度对列表进行排序则横线位置的参数应该如何填写。 答案 `key`

```
list.sort(list1, _____ = len)
```

解析：在遇到函数参数和使用方法等不确定的情况下，可以多使用 `help` 函数，例如这里可以使用 `help(list.sort)` 查看 `sort` 方法的参数和使用方法。

Quiz:

- ◆ 判断如下陈述是否正确？ 答案 T

使用 `open()` 函数打开文件并调用 `write()` 函数写文件时，由于默认采用缓冲区，操作系统不会立刻把待写入的数据写入磁盘，而是先存储到内存缓冲区，如果使用不当，有可能导致部分数据未写入到磁盘而丢失。

解析：要使用 `close()` 函数或用文件异常处理语句 “`with open(file, mode) as fp: ...`” 处理

◆ 判断如下陈述是否正确？ 答案 F

对于一个用 “`r+`” 模式打开的文件，利用 `f.seek(0, 0)` 可以让文件指针回到文件头部，此时如果执行 `f.write('I am Yeliangchen')` 语句并正常关闭文件可以将字符串 “I am Yeliangchen” 插入到文件头部。

解析：这样的执行效果是覆盖而不是插入

◆ 以下对于序列的相关操作中正确的输出结果是哪一项？ 答案 C

(A)

<pre>>>> [5] * 2</pre>
<pre>[10]</pre>

(B)

<pre>>>> word = 'cloud'; word[5]</pre>
<pre>'d'</pre>

(C)

<pre>>>> word = 'cloud'; print min(word)</pre>
<pre>c</pre>

(D)

<pre>>>> print 'Merry Xmas ' + 12.25</pre>
<pre>Merry Xmas 12.25</pre>

D 解析：“`+`” 运算对象必须是字符串，所以 12.25 需要先转换成字符串然后进行连接运算

程序参考代码

◆ 请完成以下文件综合编程迷你项目。

```
def insert_line(lines):
```

```
    lines.insert(0, "Blowin' in the wind\n")
```

```
    lines.insert(1, "Bob Dylan\n")
```

```
    lines.append("1962 by Warner Bros. Inc.")
```

```
    return ".join(lines)
```

```
with open(r'Blowing in the wind.txt', 'r+') as f:
```

```

lines = list(f.readlines())

string = insert_line(lines)

print(string)

f.seek(0)

f.write(string)

```

◆ 抓取百度贴吧系列帖子

```

import urllib.request

def baidu_tieba(url, begin_page, end_page):

    for i in range(begin_page, end_page + 1):

        f_name = 'd:\\test\\'+str(i)+'.html'

        print('正在下载'+str(i)+'页面')

        data = urllib.request.urlopen(url+str(i)).read()

        with open(f_name, 'wb') as f: #一种更简洁有效的异常处理方式

            f.write(data)

if __name__ == "__main__":

    url = "http://tieba.baidu.com/p/"

    begin_page = 1000000000

    end_page = 1000000009

    baidu_tieba(url, begin_page, end_page) # 可使用列表保存页面名称

```

◆ 统计字符串中不同字符的个数（不区分大小写）。

```

def countchar(str):

    list1 = [0]*26

    for i in range(0,len(str)):

        if (str[i] >= 'a' and str[i] <='z'):

            list1[ord(str[i])-97] += 1

    return list1

if __name__ == "__main__":

```

```
str=raw_input()
str = str.lower()
print countchar(str)
```