

PYTHON

编程基础

文件对象的readline、 readlines和seek方法

readline方法

使用文件对象的readline方法可以从文件中每次读取一行数据，其语法格式为：

```
f.readline()
```

其中，f是open函数返回的文件对象。
readline方法将从文件中读取的一行数据返回。

例：readline方法使用示例

```
ls=[]  
with open('D:\\Python\\test.txt','r') as f:  
    ls.append(f.readline())  
    ls.append(f.readline())  
print(ls)
```

```
['Python是一门流行的编程语言！\n', '我喜欢学习Python语言！']
```

readlines方法

使用文件对象的readlines方法可以从文件中按行读取所有数据，其语法格式为：

```
f.readlines()
```

其中，f是open函数返回的文件对象。

readlines方法将从文件中按行读取的所有数据以列表形式返回。

例：readlines方法使用示例

```
1 with open('D:\\Python\\test.txt','r') as f:  
2     ls=f.readlines()  
3 print(ls)
```

```
['Python是一门流行的编程语言！\n', '我喜欢学习Python语言！']
```

提示

使用list函数也可以得到readlines方法同样的结果。
例如，对于第2行代码，将其改为“ls=list(f)”，最后运行结果相同。

seek方法

使用seek方法可以移动文件指针，从而实现文件的随机读写，其语法格式为：

```
f.seek(pos, whence=0)
```

其中，f是open函数返回的文件对象；pos是要移动的字节数；whence是参照位置，默认值0表示以文件首作为参照位置，1和2分别表示以当前文件指针位置和文件尾作为参照位置。seek方法没有返回值。

例：seek方法使用示例

```
with open('D:\\Python\\test.txt','r') as f:  
    f.seek(6,0)  
    print(f.readline())
```

是一门流行的编程语言！

例：seek方法使用示例

提示

文件的顺序读写是指打开文件后，按照从前向后的顺序依次进行数据的读/写操作；而随机读写可以直接使文件指针指向某个位置，并对该位置的数据进行读/写操作，即读/写数据的位置不按固定顺序、可以随机指定。



当以文本方式打开文件后，只支持以文件首作为参照位置进行文件指针的移动；而以二进制方式打开文件后，可以支持全部的三种参照位置。通过seek方法实现的文件随机读写主要用于二进制文件，建议尽量不对文本文件进行随机读写。

与seek对应的还有一个tell方法，其用于获取当前文件指针的位置。