python文件读取

文件读取的模式

模式	介绍
r (read 的缩写)	读取文件（返回的是字符串类型）
rb	二进制形式读取文件

文件对象的读取方法

方法名	参数	介绍	举例
read	无	将文件内容一次性全部独取出来，返回整个整个文件的字符串	f.read()
readlines	无	将文件内容的每一行内容切割成列表读取，返回文件列表	f.readlines()
readline	无	将文件每一行内容进行读取，一次返回一行，返回文件中的一行	f.readline()
mode	无	open()函数的 mode属性，调用可返回当前文件模式	f.mode()
name	无	返回文件名称	f.name()
closed	无	返回一个 bool 类型，调用 closed() 函数可以知道文件是否关闭	f.closed()

```python
import os.path

current_path = os.getcwd()

file = open('read_test.py', 'r')
data = file.read()
file.close()

print(data)
print('*****************************')
print(type(data))
```

read_test.py 文件内容如下

```python
# coding:utf-8

import os.path


current_path = os.getcwd()

file_path = os.path.join(current_path, 'test.txt')
file_test = open(file_path, 'w+', encoding='utf-8')
file_test.write('人生苦短 我用Python')
file_test.read()
file_test.close()
```

```
In [1]: import os.path
   ...:
   ...: current_path = os.getcwd()
   ...:
   ...: file = open('read_test.py', 'r')
   ...: data = file.read()
   ...: file.close()
   ...:
   ...: print(data)
   ...: print('**************************')
   ...: print(type(data))
# coding:utf-8

import os.path



current_path = os.getcwd()

file_path = os.path.join(current_path, 'test.txt')
file_test = open(file_path, 'w+', encoding='utf-8')
file_test.write('人生苦短 我用Python')
file_test.read()
file_test.close()
**************************
<class 'str'>
```

读取文件内容赋值给 data

打印“read_test.py”文件内容

data 类型为 str

Structure

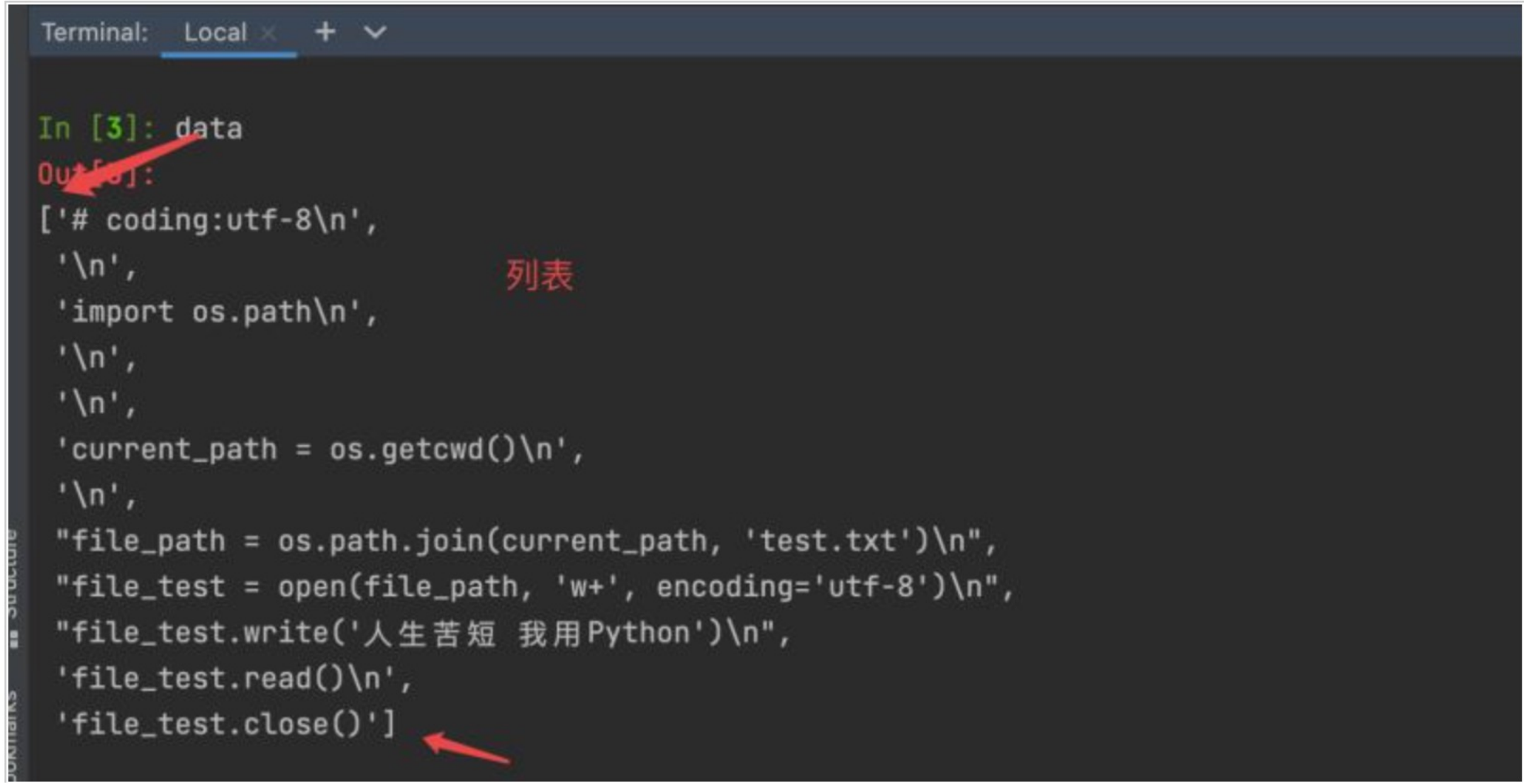
Bookmarks

使用 readlines() 函数 读取文件内容

继续上面的脚本我们使用 readlines() 函数

```
file = open('read_test.py', 'r')
data = file.readlines()
file.close()

print(data)
print('*****************************')
print(type(data))
```

Terminal 终端 执行的效果如下图：


Terminal: Local
In [3]: data
Out[3]:
['# coding:utf-8\n',
'\n',
'import os.path\n',
'\n',
'\n',
'current_path = os.getcwd()\n',
'\n',
"file_path = os.path.join(current_path, 'test.txt')\n",
"file_test = open(file_path, 'w+', encoding='utf-8')\n",
"file_test.write('人生苦短 我用Python')\n",
'file_test.read()\n',
'file_test.close()']
列表

这里我们发现每一行和空行都会有一个换行符，如果我们需要读取、处理每一行的内容，空行与换行符会给我们造成很大的困扰。这里我们就可以利用到字符串的 strip() 函数加上 for 循环 就可以处理了。

```
_data = []

for i in data:
    temp = i.strip()
    if temp != '':
        _data.append(temp)

_data

# >>> 执行结果如下图：
```

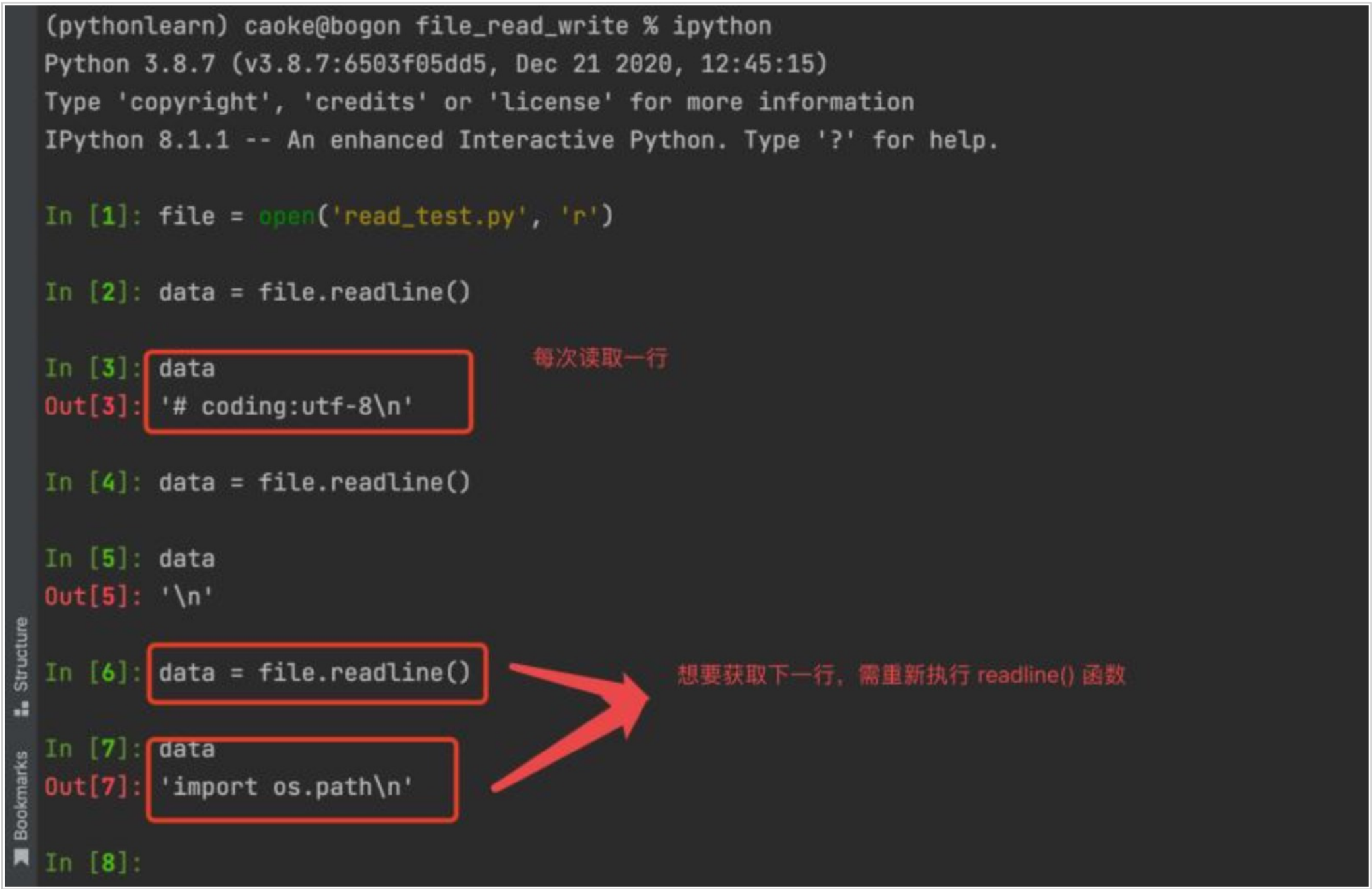
```
In [4]: _data = []  定义一个空列表，用来临时存储去掉空行与换行符的每一行内容
   ...:
   ...: for i in data:      循环遍历 data 的每一行内容
   ...:     temp = i.strip()   去掉每一行的空格
   ...:     if temp != '':     如果每一行不为空
   ...:         _data.append(temp)则将临时处理空格、换行符及空行的每一行内容添加到 _data 列表
   ...:

In [5]: _data   打印输出 _data 列表，换行符与空行已经被处理掉了
Out[5]:
['# coding:utf-8',
 'import os.path',
 'current_path = os.getcwd()',
 "file_path = os.path.join(current_path, 'test.txt')",
 "file_test = open(file_path, 'w+', encoding='utf-8')",
 "file_test.write('人生苦短 我用Python')",
 'file_test.read()',
 'file_test.close()']

In [6]:
```

使用 readline() 函数 逐行读取文件内容

上文我们提到 readline() 函数 会针对文件每一行内容进行读取，一次返回一行；如果想要读取下一行内容，就需要再一次执行 readline() 函数；下面我们来看一下 演示案例：

```
file = open('read_test.py', 'r')
data = file.readline()

data

# >>> 执行结果如下:
# >>> '# coding:utf-8\n'

data = file.readline()

data

# >>> 执行结果如下:
# >>> '\n'
```

如下图：


(pythonlearn) caoke@bogon file_read_write % ipython
Python 3.8.7 (v3.8.7:6503f05dd5, Dec 21 2020, 12:45:15)
Type 'copyright', 'credits' or 'license' for more information
IPython 8.1.1 -- An enhanced Interactive Python. Type '?' for help.
In [1]: file = open('read_test.py', 'r')
In [2]: data = file.readline()
In [3]: data
Out[3]: '# coding:utf-8\n'
每次读取一行
In [4]: data = file.readline()
In [5]: data
Out[5]: '\n'
In [6]: data = file.readline()
想要获取下一行，需重新执行 readline() 函数
In [7]: data
Out[7]: 'import os.path\n'
In [8]:
Structure
Bookmarks

每次如果都按照如上最终方案去写的话，实在太繁琐。Python引入了with语句来自动帮我们调用close()方法

重点：！！！ with 的作用就是自动调用close（）方法！！！

```
# 使用方法:
with open('/path/to/file', 'r') as f:
    print(f.read())

# >>> 相较于单独使用 open() 函数，是不是代码更佳简洁，并且不必调用f.close()方法了呢?
```