Python核心数据类型-文件

文件：计算机中由操作系统管理的具有名字的存储区域.

文件打开方式：
r只读，r+ 读写若文件存在不创建.

w覆盖写，w+新建覆盖读写，覆盖写都会将文件内容清零

r+可读可写，若文件不存在，报错.
w+可读可写，若文件不存在，创建操作文件.

In [2]:
```
#示例代码1.只读方式打开文件，文件不可写.
f = open('123.log', 'r')
f.read()
f.write('123\n')
```

```
---------------------------------------------------------------------------
IOError                                   Traceback (most recent call last)
<ipython-input-2-3a908797bff7> in <module>()
      2 f = open('123.log', 'r')
      3 f.read()
----> 4 f.write('123\n')

IOError: File not open for writing
```

In [1]:
```
#示例代码2.以读写模式打开文件.
f = open('123.log', 'r+')
f.read()
f.write('123\n')
f.close()
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-1-e9192b8f8244> in <module>
      1 #示例代码2.以读写模式打开文件. 注意：写的方式为追加写.
----> 2 f = open('123.log', 'r+')
      3 f.read()
      4 f.write('123\n')
      5 f.close()

FileNotFoundError: [Errno 2] No such file or directory: '123.log'
```

In [8]:
```
#示例代码3.以覆盖写的模式打开文件，读操作失败
f = open('123.log', 'w')
f.write('xxx\n')
f.read()
f.close()
```

```
---------------------------------------------------------------------------
IOError                                   Traceback (most recent call last)
<ipython-input-8-ac1178616f86> in <module>()
      2 f = open('123.log', 'w')
      3 f.write('xxx\n')
----> 4 f.read()
      5 f.close()

IOError: File not open for reading
```

In [9]:
```
#示例代码4.以追加写的模式打开文件，读操作失败
f = open('123.log', 'a')
f.write('sdfsdf\n')
print f.read()
f.close()
```

```
---------------------------------------------------------------------------
IOError                                   Traceback (most recent call last)
<ipython-input-9-1fe730cd8daa> in <module>()
      2 f = open('123.log', 'a')
      3 f.write('sdfsdf\n')
----> 4 print f.read()
      5 f.close()

IOError: File not open for reading
```

In [10]:
```
#示例代码5. 以读写方式打开文件
f = open('123.log', 'r+') #读写
f = open('123.log', 'w+') #读写(覆盖写)
f = open('123.log', 'a+') #读写
```

In [11]:
```
#文件对象常用方法
dir(f)
```

Out[11]:
```
['__class__',
 '__delattr__',
 '__doc__',
 '__enter__',
 '__exit__',
 '__format__',
 '__getattribute__',
 '__hash__',
 '__init__',
 '__iter__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'close',
 'closed',
 'encoding',
 'errors',
 'fileno',
 'flush',
 'isatty',
 'mode',
 'name',
 'newlines',
 'next',
 'read',
 'readinto',
 'readline',
 'readlines',
 'seek',
 'softspace',
 'tell',
 'truncate',
 'write',
 'writelines',
 'xreadlines']
```

In [32]:
```
#1.read() 将文件中的指定size的内容, 生成一个字符串, 不指定size默认读取所有
f = open('123.log')
help(f.read)
t = f.read()
print t, type(t)
f.close()
```

```
Help on built-in function read:

read(...)
    read([size]) -> read at most size bytes, returned as a string.

    If the size argument is negative or omitted, read until EOF is reached.
    Notice that when in non-blocking mode, less data than what was requested
    may be returned, even if no size parameter was given.

sdfsdf
sdf
sdf
werwe
vxcvxcv
sdf123

<type 'str'>
```

In [33]:
```
#2.readline() 返回文件内容的下一行
f = open('123.log')
help(f.readline)
print f.readline()
print f.readline()
print f.readline()
print f.readline()
print f.readline()
print f.readline()
print f.readline()
print f.readline()
f.close()
```

```
Help on built-in function readline:

readline(...)
    readline([size]) -> next line from the file, as a string.

    Retain newline.  A non-negative size argument limits the maximum
    number of bytes to return (an incomplete line may be returned then).
    Return an empty string at EOF.

sdfsdf

sdf

sdf

werwe

vxcvxcv

sdf123
```

In [35]:
```
#3.readlines() 将文件中的所有内容读取到一个列表当中, 列表的元素为文件的每一行
f = open('123.log')
help(f.readlines)
f.readlines()
```

```
Help on built-in function readlines:

readlines(...)
    readlines([size]) -> list of strings, each a line from the file.

    Call readline() repeatedly and return a list of the lines so read.
    The optional size argument, if given, is an approximate bound on the
    total number of bytes in the lines returned.
```

Out[35]:
```
['sdfsdf\n', 'sdf\n', 'sdf\n', 'werwe\n', 'vxcvxcv\n', 'sdf123\n', '\n']
```

In [37]:
```
#4.tell() 获取当前读取的文件内容的位置游标
help(f.tell)
print f.tell()
```

```
Help on built-in function tell:

tell(...)
    tell() -> current file position, an integer (may be a long integer).

37
```

In [39]:
```
#5.seek() 跳转到指定的游标位置
help(f.seek)
f.seek(0)
print f.readlines()
```

```
Help on built-in function seek:

seek(...)
    seek(offset[, whence]) -> None.  Move to new file position.

    Argument offset is a byte count.  Optional argument whence defaults to
    0 (offset from start of file, offset should be >= 0); other values are 1
    (move relative to current position, positive or negative), and 2 (move
    relative to end of file, usually negative, although many platforms allow
    seeking beyond the end of a file).  If the file is opened in text mode,
    only offsets returned by tell() are legal.  Use of other offsets causes
    undefined behavior.
    Note that not all file objects are seekable.
['sdfsdf\n', 'sdf\n', 'sdf\n', 'werwe\n', 'vxcvxcv\n', 'sdf123\n', '\n']
```

In [40]:
```
#6.flush() 将缓冲区的内容写到磁盘
help(f.flush)
```

```
Help on built-in function flush:

flush(...)
    flush() -> None.  Flush the internal I/O buffer.
```

In [41]:
```
#7.close() 关闭文件对象
help(f.close)
```

```
Help on built-in function close:

close(...)
    close() -> None or (perhaps) an integer.  Close the file.

    Sets data attribute .closed to True.  A closed file cannot be used for
    further I/O operations.  close() may be called more than once without
    error.  Some kinds of file objects (for example, opened by popen())
    may return an exit status upon closing.
```

In [43]:
```
#8.closed 判断文件是否关闭
print f.closed
f.close()
print f.closed
```

```
False
True
```

In [45]:
```
#9.write() 将字符串写入到文件中
f = open('123.log', 'a+')
f.write('apple\n')
f.write('bananan\n')
f.close()
```

In [46]:
```
#10.writelines() 将列表中的所有元素组成的字符串写入到文件中.
f = open('123.log', 'a+')
f.writelines(["a", "dd", "football", "\n"])
f.flush()
f.seek(0)
print f.readlines()
f.close()
```

```
['sdfsdf\n', 'sdf\n', 'sdf\n', 'werwe\n', 'vxcvxcv\n', 'sdf123\n', '\n', 'apple\n', 'bananan\n', 'addfootball\n']
```

In [ ]:
```

```