

# StringIO和BytesIO

---

## StringIO

- io模块中的类
  - from io import StringIO
- 内存中，开辟的一个文本模式的buffer，可以像文件对象一样操作它
- 当close方法被调用的时候，这个buffer会被释放

getvalue() 获取全部内容。跟文件指针没有关系

```
from io import StringIO

# 内存中构建
sio = StringIO() # 像文件对象一样操作
print(sio.readable(), sio.writable(), sio.seekable())
sio.write("magedu\nPython")
sio.seek(0)
print(sio.readline())
print(sio.getvalue()) # 无视指针，输出全部内容
sio.close()
```

好处

一般来说，磁盘的操作比内存的操作要慢得多，内存足够的情况下，一般的优化思路是少落地，减少磁盘IO的过程，可以大大提高程序的运行效率

---

## BytesIO

- io模块中的类
  - from io import BytesIO
- 内存中，开辟的一个二进制模式的buffer，可以像文件对象一样操作它
- 当close方法被调用的时候，这个buffer会被释放

```
from io import BytesIO # 内存中构建
bio = BytesIO()
print(bio.readable(), bio.writable(), bio.seekable())
bio.write(b"magedu\nPython")
bio.seek(0)
print(bio.readline())
print(bio.getvalue()) # 无视指针，输出全部内容
bio.close()
```

---

## file-like对象

- 类文件对象，可以像文件对象一样操作
- socket对象、输入输出对象（stdin、stdout）都是类文件对象

```
from sys import stdout, stderr
f = stdout
print(type(f))
f.write('magedu.com')
```