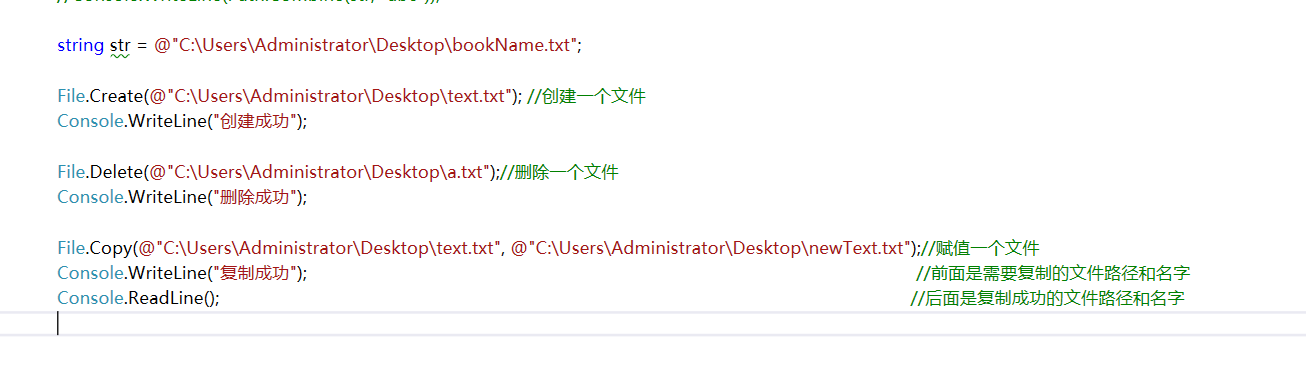
**File类** **静态类**

1. File这个类是直接操作文件的复制、删除、创建等等
2. 由于File类为静态类，我们就可以直接类名.方法名来调用
3. **File类只能改写和读取小数据，修改大数据的时候需要用到文件流**

**基本方法：**



1. **读取数据：**

**byte[] File.ReadAllBytes(文件路径)； 返回一个字节类型的数组**



**String Encoding.Default.GetString(byte[])； 把括号里的字节数组转化为字符串**



**可以简写成：**



**注意：**



正方形为编码形式，如果该文件的编码形式为UTF-8 但是写入的是ANSI（Default）那么输出str的时候就会乱码，除非我们改成 UTF-8的形式就不会，如图：



详情参考====>

1. **写入数据：**

**创建文件 File.WriteAllBytes（创建的文件路径名，需要创建的文件的byte数组）；**

**那么怎么去的获得需要创建的文件Byte字节数组？**

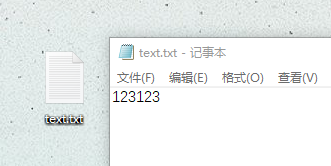
**通过以下方法把string字符串转化成字节数组：**

**byte[] Enconding.Defauly.GetBytes(字符串)；//Defauly为ANSI编码形式，我们写入的是什么形式，那么文件的编码形式就是什么**

**所以写入数据可以简写为：**



并且返回以下值：



=====================以上为基本方法========================

=====================以下为拓展方法========================

**特殊的方法：**

1. bool File.Exists（string path）；判断该文件是否存在，返回true、false
2. void File.Move(string scoreFileName,string destFileName);移动指定文件

**读取数据方法总结：**

1. string[] File.ReadAllLines（文件路径，Encoding编码形式)；把该路径中的txt文件按照Encoding的编码形式读取出来，并且一行一行读取，并返回一个数组
2. string File.ReadAllText（文件路径，Encoding编码形式）；把该路径中的txt文件按照Encoding的编码形式都取出来，并返回一个字符串
3. byte[] File.ReadAllBytes(文件路径)；把该路径中的文件都取出来，并返回一个字节数组，我们需要显示出来的时候还要通过：Encoding.编码名.GetString()；来把字节数组转换成字符串

**区别：**

1. **方法1 和方法2只能读取txt文件，方法3能读取任何类型，因为他能读取字节**
2. **方法1返回的是一个数组，方法2是返回的整个txt文档的内容，所以当我们需要对文件中的内容进行修改的时候可以用方法1**

**写入创建数据方法总结：**

1. File.WriteAllBytes（创建文件的路径和文件名，创建文件的byte数组）；创建一个文件，但是需要用到byte数组，所以创建的时候还需要用：Encoding.编码名.GetBytes()；来创建一个字节型数组
2. File.WriteAllLines（创建文件的路径和文件名，创建文件的string数组）；用数组中的元素一行一行的写入文件中并创建
3. File.WriteAllText（创建文件的路径和文件名，字符串）；按照字符串的内容写入该文件中并创建出来
4. File.Appen …… 在文件内容中追加新的内容，不会覆盖的原有内容

**区别：**

1. **方法1能够写入任何类型的文件，因为他是写入字节，而方法2、3只能写入文本文件，因为写入的都是string类型**