**[正斜杠（/）与反斜杠（\）总结](http://blog.csdn.net/north_easter/article/details/7904865)**

　正斜杠/表示除法，分隔。在windows系统中通常用来分隔命令行参数，/表示选项等。不能作为文件名。

　　反斜杠\，在windows系统中用来表示目录。

　　而在unix系统中，/表示目录。由于web遵循unix命名，所以在网址（URL）中，/表示目录。

**C#中反斜杠\与斜杠/ 转移字符 路径的问题**

　　在C#中 \ 是转义字符,如下所示

|  |  |
| --- | --- |
| **转义序列** | **字符** |
| \’ | 单引号 |
| \” | 双引号 |
| \\ | 反斜杠 |
| \0 | 空 |
| \a | 警告 |
| \b | 退格 |
| \f | 换页 |
| \n | 换行 |
| \r | 回车 |
| \t | 水平制表符 |
| \v | 垂直制表符 |

　　就比如说要输出这一段字符串 “/\_\” 可能你会认为是这样写 @""/\_\""

　　但这样写编译器都不让你通过了,但看上表所示,你可以这么写 "\"/\_\\\""

　　另外在获取文件路径的时候 很多人喜欢用 d:\test.txt

　　然后就会发生所谓的转义字符错误了

　　解决的办法一般人是使用 @d:\test.txt 我不知道这样写是否会带来一些问题

　　但我一直的做法是d:/test.txt 也完全可以访问

　　在网上找了下, 似乎斜杠与反斜杠最大的区别就是

　　网络中不能以斜杠在windows系统中通常用来分隔命令行参数，/表示选项等。

　　算术运算符/用来取商，如5/2=2.5

引用说明：有正斜杠与反斜杠之分，正斜杠，一般就叫做斜杠，符号为“/”；反斜杠的符号为“\” 引用斜杠（/）在JAVA中没有什么特别的意义，就是代表一个字符‘/’;   
反斜杠（\）则不然，它和紧跟着它的那个字符构成转义字符，如“\n”（表示换行）、“\””（表示字符‘”’）等,所以在字符串中要表示字符'\'要用“\\”来表示，例：如果你这样定义一个字符串String s = “name\sex”是错误的，要这样定义String s = “name\\sex”;   
引用注意：在正则表达式中的“\\”表示和后面紧跟着的那个字符构成一个转义字符（姑且先这样命名），代表着特殊的意义；所以如果你要在正则表达式中表示一个反斜杠\，应当写成“\\\\”。如果你这样获得一个Matcher，Matcher m = Pattern.compile(“\\”).matcher(“\\”)将会报错,你应当这样写Matcher m = Pattern.compile(“\\\\”).matcher(“\\”)才是正确且匹配的引用接下来我们来看一下String类中replace(CharSequence target,CharSequence replacement)方法与replaceAll(String regex, String replacement)方法的区别：

public static void main(String[] arg) throws OgnlException {

String s ="sdf\\a\\aa";

//把s中的反斜杠\ 替换为\\

System.out.println(s);

System.out.println(s.replaceAll("\\\\", "\\\\\\\\"));

System.out.println(s.replace("\\", "\\\\"));

}

引用可以看出上面两种都返回相同的替换结果。   
这 里的关键是String.replaceAll()是用regular expression 来作为参数的。但是java本身的字符串对于转义符\也有类似的处理。首先，java会把“\\\\”解释成一个字符串(其中包含两个char)。接下来，由于replaceAll是以正则表达式作为参数，所以“\\”被解释成一个regex。对于一个regex来说这就代表着一个字符，就是“\”。对后面的那个8个\来说，最终会被解释成“\\”。   
换言之，假设String.replaceAll()是以普通字符串，而不是regex作为参数，那么这样写代码: String target = source.replaceAll('\\', '\\\\'); 就可以了。

　　正斜杠(左斜杠)，符号是"/"；反斜杠(右斜杠)，符号是"\"。

　　在Unix/Linux中，路径的分隔采用正斜杠"/"，比如"/home/hutaow"；而在Windows中，路径分隔采用反斜杠"\"，比如"C:\Windows\System"。

　　有时我们会看到这样的路径写法，"C:\\Windows\\System"，也就是用两个反斜杠来分隔路径，这种写法在网络应用或编程中经常看到，事实上，上面这个路径可以用"C:/Windows/System"来代替，不会出错。但是如果写成了"C:\Windows\System"，那就可能会出现各种奇怪的错误了。

　　至于上述问题出现的原因，要从字符串解析这方面来分析。

　　学过编程的人都应该知道，在C里面，输出字符串时，如果想输出一个换行，那就要加上'\n'这个标志，类似的，输出一个TAB，就加上'\t'，也就是说，反斜杠("\")这个符号会把跟在它后面的字符结合起来转义成其它字符。根据这个原理，如果想输出双引号('"')，就需要输入'\"'，这样才会将包含了双引号的字符串正确的写入内存中。那么如果想输入一个反斜杠呢？很简单，只要敲'\\'就可以了。

　　看到这里或许有些人已经看出眉目了，如果"C:\Windows\System"这个路径字符串交给C编译器编译，实际写入内存的字符串并没有包含反斜杠"\"，甚至紧跟在反斜杠后面的字母也一起被转义成了其它的字符，再次调用的话势必会出问题。

　　字符串解析不仅仅局限于C编译器，Java编译器、一些配置文件的解析、Web服务器等等，都会遇到对字符串进行解析的这个问题，由于传统的Windows采用的是单个斜杠的路径分隔形式，导致在对文件路径进行解析的时候可能发生不必要的错误，所以就出现了用双反斜杠"\\"分隔路径的形式。不管解析引擎是否将反斜杠解析成转义字符，最终在内存中得到的都是"\"，结果也就不会出问题了。

　　由此也可以看出Windows或者说DOS在设计初期考虑不够周全，为了和Unix一些特征区别开来，将Unix中的正斜杠"/"分隔路径方式改变成了反斜杠"\"。这样改变导致的一个问题就是在早期DOS命令行中，正常的文件名是不能包含空格的，如果包含了空格，会导致输入这样的文件名时，命令解析无法将其和参数区分开。例如，想要进入"hutaow yuan"这个目录（在此先忽略8.3命名规则），直接输入"cd hutaow yuan"，命令行会将其解析为进入"hutaow"目录，而后面的"yuan"做参数，这显然不是所期望的。

　　而在Unix中，文件名如果包含空格，可以直接在空格前加上反斜杠"\"进行转义，从而很好的和命令参数区别出来（参数之间一般使用空格分隔）。还是上面的例子，在Unix中，只要输入"cd hutaow\ yuan"(在yuan前面的空格前加上"\")，命令行会正确的辨认出"hutaow yuan"并进入这个目录。

　　当然，现在Windows的后续版本已经使用其它方法（比如文件名用双引号括住）解决了空格问题。

正斜杠，又称左斜杠，符号是"/";  
反斜杠，也称右斜杠，符号是"\"。

经常很迷惑正斜杠与反斜杠到底有何区别？以下是一些总结:

DOS路径：  
C:\WINDOWS\SETTING   ..这是反斜杠的作用后面跟着的是文件名  
C:\dir /P            ...这就是正斜杠后面是一个名令的参数dir 是命令**.**

C#中:

在C#中 \ 是转义字符,如下所示

|  |  |
| --- | --- |
| **转义序列** | **字符** |
| \’ | 单引号 |
| \” | 双引号 |
| \\ | 反斜杠 |
| \0 | 空 |
| \a | 警告 |
| \b | 退格 |
| \f | 换页 |
| \n | 换行 |
| \r | 加车 |
| \t | 水平制表符 |
| \v | 垂直制表符 |

另外在获取文件路径的时候 很多人喜欢用 "d:\test.txt"    ,然后就会发生所谓的转义字符错误了   
解决的办法一般人是使用@"d:\test.txt"

windows系统:

反斜杠\，在windows系统中用来表示目录。  
正斜杠/，网址、Url地址。

Asp.net中：  
"../ "   表示当前目录的上级目录,即当前的父目录;  
"/"      表示网站的根目录；