在通过github或者其他远程托管服务器来和其他人进行协同开发代码的时候，确保换行符被正确处理是一项很重要的事。首先，不同的操作系统对换行符的定义会有所不同，Unix或类Unix操作系统的换行符叫做LF(Line Feed, 换行， 字符表示\n)，而windows系统的叫做CRLF(Carriage Return + Line Feed： 回车换行, 字符表示\r\n)，二者具有很大的区别：

Unix系统里，每行结尾只有 “<换行>”，即”\n”；Windows系统里面，每行结尾是“<回车><换行>, \r\n, CRLF”

如果我们跨平台开发，但是用同一个repo,可能会出现的换行问题为。我们在window下改变一个文件一行， 然后push到repo, 然后在linux下准备pull(fetch + merge) 下来，继续进行开发。会出现的问题是，在同一个文件里面有混杂的回车符号，有的采用了\r\n(Window)，有的采用了\n。

git其实可以自己处理换行符不统一的问题，但是可能会产生意想不到的结果。因此，有必要进行相关的配置，我们通常有两种方案：

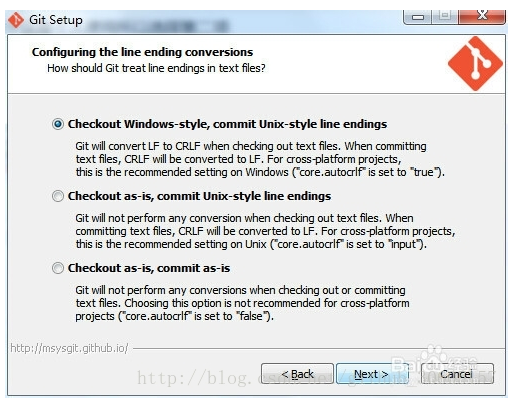
**全局配置换行符**

针对某个仓库的局部配置

换行符的全局配置

$ git config --global core.autocrlf true

其实，在你安装windows版本的git或者torgoiseGit时，可能已经进行过这样的配置，



Checkout Windows-style, commit Unix-style line endings

当检出文本文件时，Git会将LF转换为CRLF。当提交文本文件时，CRLF将会被转换为LF。对于跨平台的项目，这是对Windows系统的推荐设置。(“core.autocrlf” is set to “true”)

Checkout as-is, commit Unix-style line endings

当检出文本文件时，Git不会做任何转换。当提交文本文件时，CRLF将会被转换为LF。对于跨平台的项目，这是对Unix系统的推荐设置。(“core.autocrlf” is set to “input”)

Checkout as-is, commit as-is

不论是检出还是提交文本文件时，Git都不会做任何转换。对于跨平台的项目，不推荐采用该选项。

其实这三个选项对应着三个值的设计：

$ git config --global core.autocrlf true

（Checkout Windows-style, commit Unix-style line endings）

Git可以在你提交时自动地把行结束符CRLF转换成LF，而在签出代码时把LF转换成CRLF。

$ git config --global core.autocrlf input

(Checkout as-is, commit Unix-style line endings)

Linux或Mac系统使用LF作为行结束符，因此你不想Git在签出文件时进行自动的转换；当一个以CRLF为行结束符的文件不小心被引入时你肯定想进行修正， 把core.autocrlf设置成input来告诉Git在提交时把CRLF转换成LF，签出时不转换：

这样会在Windows系统上的签出文件中保留CRLF，会在Mac和Linux上，包括仓库中保留LF。

$ git config --global core.autocrlf false

(Checkout as-is, commit as-is

)

如果你是Windows程序员，且正在开发仅运行在Windows上的项目，可以设置false取消此功能，把回车符记录在库中. 在迁入迁出的时候不做任何转换。

**单一仓库的换行符局部配置**

当然，有更好的方法解决换行符不统一的问题——使用.gitattributes文件统一换行符。这种方法是针对某个仓库进行换行符的统一配置，即时你已经进行了全局配置。

另外，这个文件有点儿类似于.gitignore，不仅名字很类似，使用方式，编写语法也很像。这个文件必须位于仓库的根目录，可以像其他文件一样进行修改、提交。下面介绍如何编写这个文件：

文件内容看起来像一张表格，总共分为两列：左边一列是git要匹配的文件名；右边是git需要采用的换行符格式。下面我们来看一个栗子：

# Set the default behavior, in case people don't have core.autocrlf set.

\* text=auto

# Explicitly declare text files you want to always be normalized and converted

# to native line endings on checkout.

\*.c text

\*.h text

# Declare files that will always have CRLF line endings on checkout.

\*.sln text eol=crlf

# Denote all files that are truly binary and should not be modified.

\*.png binary

\*.jpg binary

如果你熟悉.gitignore的话，你会觉得上面这个文件的左边一列很熟悉，这里我们就不再赘述了，不熟悉的话，请自行查阅相关资料。唯一的不同就是.gitattributes文件多了右边一列，如text, text eol=crlf, binary，刚刚我们也说过这一列是用来设置左边对应文件使用的换行符格式的，左右两列用空格隔开。下面我们来详细介绍下右边一列的语法：

text=auto

让git自行处理左边匹配的文件使用何种换行符格式，这是默认选项。

text eol=crlf

对左边匹配的文件统一使用CRLF换行符格式，如果有文件中出现LF将会转换成CRLF。

text eol=lf

对左边匹配的文件统一使用LF换行符格式，如果有文件中出现CRLF将会转换成LF。

binary

告诉git这不是文本文件，不应该对其中的换行符进行改变。另外，binary和符号-text -diff是等价的。

上面这些语法应该已经足够了，如果有兴趣的，可以自行查阅相关资料，比如官方的资料：https://git-scm.com/book/en/v2/Customizing-Git-Git-Attributes。