# [linux中批量替换文本中字符串--转载](https://www.cnblogs.com/nkwy2012/p/6365714.html)

（一）通过vi编辑器来替换。  
vi/vim 中可以使用 :s 命令来替换字符串。  
:s/well/good/ 替换当前行第一个 well 为 good  
:s/well/good/g 替换当前行所有 well 为 good  
:n,$s/well/good/ 替换第 n 行开始到最后一行中每一行的第一个 well 为 good  
:n,$s/well/good/g 替换第 n 行开始到最后一行中每一行所有 well 为 good  
n 为数字，若 n 为 .，表示从当前行开始到最后一行  
:%s/well/good/（等同于 :g/well/s//good/） 替换每一行的第一个 well 为 good  
:%s/well/good/g（等同于 :g/well/s//good/g） 替换每一行中所有 well 为 good  
可以使用 # 作为分隔符，此时中间出现的 / 不会作为分隔符  
:s#well/#good/# 替换当前行第一个 well/ 为 good/  
:%s#/usr/bin#/bin#g 可以把文件中所有路径/usr/bin换成/bin

（二）直接替换文件中的字符串。(此法不用打开文件即可替换字符串，而且可以批量替换多个文件。)  
1.perl命令替换，参数含义如下：  
-a    自动分隔模式，用空格分隔$\_并保存到@F中。相当于@F = split ”。分隔符可以使用-F参数指定  
-F    指定-a的分隔符，可以使用正则表达式  
-e    执行指定的脚本。  
-i<扩展名>   原地替换文件，并将旧文件用指定的扩展名备份。不指定扩展名则不备份。  
-l    对输入内容自动chomp，对输出内容自动添加换行  
-n    自动循环，相当于 while(<>) { 脚本; }  
-p    自动循环+自动输出，相当于 while(<>) { 脚本; print; }  
用法示例：  
perl -p -i.bak -e 's/\bfoo\b/bar/g' \*.c  
将所有C程序中的foo替换成bar，旧文件备份成.bak

perl -p -i -e "s/shan/hua/g" ./lishan.txt ./lishan.txt.bak   
将当前文件夹下lishan.txt和lishan.txt.bak中的“shan”都替换为“hua”

perl -i.bak -pe 's/(\d+)/ 1 + $1 /ge' file1 file2   
将每个文件中出现的数值都加一

2.sed命令下批量替换文件内容

格式: sed -i "s/查找字段/替换字段/g" `grep 查找字段 -rl 路径` 文件名

-i 表示inplace edit，就地修改文件

 -r 表示搜索子目录

 -l 表示输出匹配的文件名  
s表示替换，d表示删除

示例：sed -i "s/shan/hua/g"  lishan.txt

          把当前目录下lishan.txt里的shan都替换为hua

sed的其他用法如下：

1、删除行首空格  
   sed 's/^[ ]\*//g' filename  
   sed 's/^ \*//g' filename  
   sed 's/^[[:space:]]\*//g' filename

2、行后和行前添加新行  
   行后：sed 's/pattern/&\n/g' filename  
   行前：sed 's/pattern/\n&/g' filename  
   &代表pattern

3、使用变量替换(使用双引号)  
    sed -e "s/$var1/$var2/g" filename

4、在第一行前插入文本  
    sed -i '1 i\插入字符串' filename

5、在最后一行插入  
    sed -i '$ a\插入字符串' filename

6、在匹配行前插入  
    sed -i '/pattern/ i "插入字符串"' filename

7、在匹配行后插入  
   sed -i '/pattern/ a "插入字符串"' filename

8、删除文本中空行和空格组成的行以及#号注释的行  
   grep -v ^# filename | sed /^[[:space:]]\*$/d | sed /^$/d