

## 关于 sed

测试文档: (people.txt)

```
Jack    is 18-year old, he comes from US.
Mike    is 16-year old, he comes from Canada.
Chen    is 21-year old, he comes from China.
Lau     is 18-year old, he comes from HongKong.
Michael is 20-year old, he comes from UK.
Phoebe  is 18-year old, she comes from Australie.
```

1, 替换:

例如, 将"-year"改成" years":

```
sed "s/-year/years/" people.txt
```

2, 指定某些行替换:

例如, 将第 2 行的"-year"改成" years":

```
sed "2s/-year/years/" people.txt
```

再如, 将第 2 至 5 行的"-year"改成" years":

```
sed "2,5s/-year/years/" people.txt
```

3, 直接通过 sed 修改原文, 加选项 -i:

```
sed -i "2s/-year/years/" people.txt
```

**注意: sed 缺省状态下不会修改原文**

4, 替换每一行中的所有的小写 s 成大写 S:

```
sed "s/s/S/g" people.txt
```

**注意: g 的意思是一行中所有的匹配项, 否则缺省只会匹配第一个 s**

5, 替换每一行中的第 2 个小写 s 成大写 S:

```
sed "s/s/S/2" people.txt
```

6, 替换每一行中的第 2 个以后的小写 s 成大写 S:

```
sed "s/s/S/2g" people.txt
```

## 多个匹配

7, 将"-year"改成" years", 并且将第 3 行以后的最后一个任意字符去掉:

```
sed 's/-year/ years/; 3,$s/.$//' people.txt
```

以上命令等价于:

```
sed -e 's/-year/ years/' -e '3,$s/.$//' people.txt
```

**注意: 在单引号里面, 元义字符可以直接使用, 如果要去掉元义则要在前面加\ ; 在双引号里面, sed 的命令要使用元义, 则需要加\ , 而命令的正则表达式要使用元义直接使用就行。**

8, 将&代替被匹配的变量:

```
sed "s/is/[&]/" people.txt
```

意思是==>: 将文本中每一行出现的第一个 is 的左右两边加上[ ]

9, 如果使用正则表达式匹配项的时候使用了圆括号括了起来, 那么可以用\1, \2, \3……等来表示这些项:

```
sed "s/^(.*)\tis.*from \(.*\). /\1\t:\2/g" people.txt
```

分解一下:

1, 其中加粗的`^(.*)\tis.*from \(.*\).`是正则表达式, 简化一下是`^(.*)\tis.*from (.*).`在这个表达式中, 制表符\t 前面的匹配项可以被记为\1, from 后面的匹配项可以被记为\2, 后面加下划线的`\1\t:\2`部分是替换的字符串, 也就是只打印匹配出来的名字和国籍, 中间用制表符和冒号隔开。

10, 提前预读多一行缓冲来进行匹配:

```
sed 'N;s/is/IS/' people.txt
```

**注意:** 由于替换只会针对第一个出现的单词 is, 而通过 N 又多读了一行, 因此这个命令的结果是只会替换奇数行。

11, 在指定行的前面插入(i)或者后面插入(a)一些信息:

```
sed '3i abcd' people.txt 意思是==>在第 3 行的前面插入 abcd
```

```
sed '2a abcd' people.txt 意思是==>在第 2 行的后面插入 abcd
```

```
sed '1,4a abcd' people.txt 意思是==>在第 1 至 4 行的后面分别插入 abcd
```

```
sed '/US/a abcd' people.txt 意思是==>在匹配 US 的行的后面插入 abcd
```

12, 将指定的行替换成其他信息:

```
sed "2c ok" people.txt 意思是==>将第 2 行替换成 ok
```

13, 将指定的行删除掉:

```
sed '2d' people.txt 意思是==>将第 2 行给删掉
```

```
sed '/US/d' people.txt 意思是==>将匹配/US/的所有行给删掉
```

```
sed '/\<he\>/d' people.txt 意思是==>将匹配 he 的所有行给删掉, 注意: 之所以要用<>将 he 给括起来, 是因为不想匹配 she, 当然, <>需要转义, 写成\<\>
```

14, 打印指定匹配的行, 用命令 p:

```
sed '/Chen/p' people.txt -n 意思是==>打印匹配 Chen 的行
```

```
sed '/Chen/, /Lau/p' people.txt -n 意思是==>打印匹配 Chen 或者 Lau 的行
```

```
sed '3, /UK/p' people.txt -n 意思是==>从第 3 行开始打印, 直到匹配 UK 为止
```

```
sed '/UK/, 6p' people.txt -n 意思是==>从匹配 UK 的行开始打印, 直到第 6 行为止
```

15, 使用相对位置:

```
sed '/US/, +2p' people.txt -n 意思是==>打印匹配 US 的行, 并打印其后的 2 行。
```

16, 执行多个命令:

```
sed '{/he/{/18/p}}' people.txt -n 意思是==>匹配所有/he/的行之后, 再匹配/18/的行, 然后打印出来。
```