



seq命令可以输出连续的数字，或者输出固定间隔的数字，或者输出指定格式的数字，这样说比较模糊，看示例就很容易理解。

seq命令最简单的使用方法如下

```
1 | seq 1 5
```

上述命令表示连续输出1到5，效果如下

```
[root@zsythink ~]# seq 1 5
1
2
3
4
5
[root@zsythink ~]#
```

zsythink.net 朱双印博客

seq命令还可以实现步进输出，比如从1 开始，每次步进 2，最大到10，效果如下。

```
[root@zsythink ~]# seq 1 2 10
1
3
5
7
9
[root@zsythink ~]#
```

zsythink.net 朱双印博客

seq命令还有一些常用选项，使用这些选项能输出更多的格式，此处先列出seq命令的常用选项，具体使用会在示例中展示。

-s 指定输出的分隔符，默认为\n，即默认为回车换行

-w 指定为定宽输出，不能和-f一起用

-f 按照指定的格式输出，不能和-w一起使用

常用选项示例：

-s选项：指定分隔符，下图示例为指定加号为分隔符，输出的数字将会使用"+"连接，默认情况下回车换行(\n)为分隔符。

```
[root@zsythink ~]# seq -s + 1 10
1+2+3+4+5+6+7+8+9+10
[root@zsythink ~]#
[root@zsythink ~]#
```

zsythink.net 朱双印博客

使用如下方法可以使用制表符(\t)作为分隔符，相当于我们键盘上的tab键。

```
[root@cos72ini ~]# seq -s "`echo -e '\t'" 9 11
9      10      11
[root@cos72ini ~]#
```

zsythink.net 朱双印博客

注意：上图示例中使用了命令替换，也就是说，先使用echo命令输出制表符，然后用输出的制表符作为seq命令输出数字的连接符。

-w 选项：指定为定宽输出，例如下图，最大值为11，是两位数，那么不到两位的数前面自动用0补全，当然，如果指定的位数最大为三位数字，那么一位数和两位数都会用0补全，也就是说，以最大值的位数为标准宽度，不足标准宽度的数字将会用0补位，示例如下图。

```
[root@zsythink ~]# seq -w 8 11
08
09
10
11
[root@zsythink ~]#
[root@zsythink ~]# seq -w 99 103
099
100
101
102
103
[root@zsythink ~]#
```

[zsythink.net](http://www.zsythink.net) 朱双印博客

-f选项：按照指定的格式输出生成的数字，在没有使用-f选项指定格式时，默认格式为%g，可以理解为使用-f指定模式为"%g"，跟不指定格式没有任何区别。

```
[root@zsythink ~]# seq 8 11
8
9
10
11
[root@zsythink ~]# seq -f '%g' 8 11
8
9
10
11
[root@zsythink ~]#
```

[zsythink.net](http://www.zsythink.net) 朱双印博客

上述示例中，输出的数字没有任何区别，因为在不指定格式的情况下，默认格式为'%g'。

那么除了上述的默认格式，我们经常还会使用如下常用格式，"%3g"这种格式表示指定"位宽"为三位，那么数字位数不足部分用空格补位，如下示例中，编号1对应格式为默认格式，与不指定任何格式输出的结果相同，编号2对应的命令使用的格式表示输出的数字位宽固定为3位，不足三位的数字用空格在数字左侧补齐。

"%02g"表示指定位宽为两位，数字位数不足用0补位，这种写法跟上述提到的-w选项类似，只不过-w是以指定的最大值的位数为最大位数，而-f选项可以直接指定格式，我们也对比了-w选项与-f选项的区别。

上述例子中的格式中，都包含一个'%',其实%前面还可以指定字符串。

例如在屏幕上打印5个名为dir1, dir2 .. dir5 的字符串，这时候就用到这种写法。

所以，结合上述示例中的seq命令的特性，再结合其他命令，就能为我们带来许多方便。

例如一次性创建5个名为dir001, dir002 .. dir005 的目录，这时候就用到这种写法。

```
1 | mkdir $(seq -f 'dir%03g' 1 10)
```

或者如下命令，与上述命令的效果相同。

```
1 | seq -f 'dir%03g' 1 5 | xargs mkdir
```



我的微信公众号

关注"实用运维笔记"微信公众号，当博客中有新文章时，可第一时间得知哦~