

如果你是一个使用过C语言的程序员，那么你一定会很容易理解printf命令，printf命令模仿了C语言中printf( )函数。

即使你没有用过C语言和printf( ) 函数，看完这篇文章以后，你也会很快的掌握printf命令的用法。

printf命令的主要作用就是输出文本，不不不，这样说并不准确，应该说，printf命令的作用是按照我们指定的格式输出文本。

而提到输出文本，你一定会想到另一个**常用命令**，echo命令，echo命令我们已经总结过，此处不再赘述，我们来对比一下，echo与printf有什么不同

```
[www.zsythink.net]# echo testString
testString
[www.zsythink.net]# printf testString
testString[www.zsythink.net]#
```

仅将文本发送到当前选项卡

zsythink.net 未双印博客

从上述示例中可以看出，在输出文本时，echo命令会对输出的文本进行换行，而printf命令则不会对输出的文本进行换行，那么，如果我们想要利用printf将输出的该怎么做呢？没错，聪明如你一定想到了，使用转义符\n，示例如下

```
[www.zsythink.net]# printf "testString\n"
testString
[www.zsythink.net]#
```

看到这里你可能会说，这样多麻烦，还不如直接使用echo命令比较方便，没错，如果只是就输出功能而言，它们并没有什么不同，但是，刚才已经强调过，printf格式化输出文本，那么，我们就来通过一个小例子，来见识一下printf的格式化输出文本的能力。

假设，我们有一串文本需要输出，如下

```
[www.zsythink.net]# echo "abc def ghi jkl mno"
abc def ghi jkl mno
[www.zsythink.net]#
```

现在我们有一个小需求，就是将上述文本按照空格分段，每段单独输出在一行里面。

当然，有很多方法可以实现上述要求，假设我们并不会使用其他命令，也不是特别会写脚本，那么我们可能会使用一些笨办法进行输出，比如，使用下图中的命令1:

```
[www.zsythink.net]# echo -e "abc \ndef \nghi \n jkl \nmno" ①
abc
def
ghi
jkl
mno
[www.zsythink.net]# printf "abc \ndef \nghi \n jkl \nmno \n" ②
abc
def
ghi
jkl
mno
[www.zsythink.net]# printf "%s\n" abc def ghi jkl mno ③
abc
def
ghi
jkl
mno
[www.zsythink.net]#
```

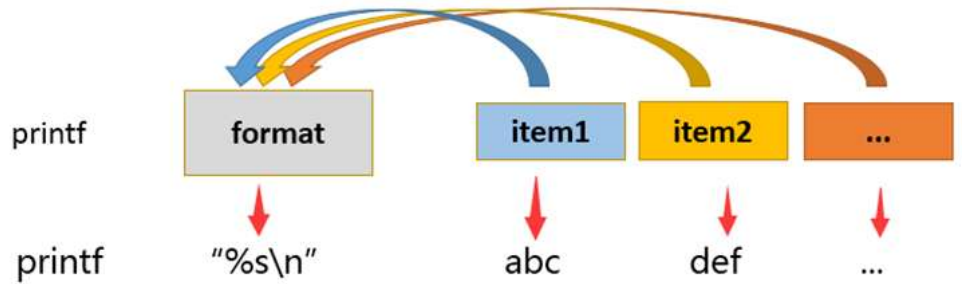
zsythink.net 未双印博客

上述示例中，命令1与命令2在输出文本时，都使用了转义字符\n将文本换行了，但是，在写命令1与命令2的时候，我的内心是崩溃的，因为，上述示例还算简单，文本分成5行输出即可，但是，如果是一个100段的文本呢？难道我们要在每一段中都添加一个转义字符"\n"，那我的手不就打残了，而命令3则不同，我只是通过printf固定的"格式"，后面的每一段文本都按照指定的格式进行了换行，即使有10000段文本需要换行，我也不担心手残的问题了，从这个小例子中，我们就能体会到printf能力，上图中命令3中使用到的"格式"会在后面进行详细的描述，不要着急。

说到这里，我发现我们还没有写出printf命令的使用语法，printf命令的语法如下

printf "指定的格式" "文本1" "文本2" "文本3" .....

还记得刚才的命令3吗？现在，我们对比着命令3和语法一起看，会更加明了。



zsythink.net 朱双印博客

聪明如你一定想到了，上述语法中的每一个“文本”都会被当做参数项传入printf命令，而每个被传入的参数都会按照指定的“格式”被“格式化”。

没错，命令3中的“%s\n”即为我们指定的“格式”，而后面的每一段字符串，都被当做参数传入到了printf命令中，并按照我们指定的格式进行了格式化。那么，我们解释一下上图中的“%s\n”是什么意思。

我们先说说“%s”是什么意思，“%s”是一个“替身演员”，我们使用“%s”代替传入的参数，也就是说，“%s”代替了命令3中的abc，代替了def，代替了ghi，代替了每数，在我们指定的“格式”中，它代表了每一个传入的参数，所以，如果我们指定的格式为“%s\n”，当abc被当做参数传入printf命令时，printf就会把“%s\n”中的%s于是，abc就变成了我们指定的格式“abc\n”，最终printf输出的就是格式化后的“abc\n”，以此类推，每一段文本都被当做一个参数传入printf命令，然后按照指定

而“替身演员”只是我给“%s”起的一个外号，它的真名叫“格式替换符”，而printf中，“格式替换符”不只有“%s”一种，“%s”代替了每一个传入的参数，并将他们转化E型”，我们再认识一个新的替身演员，“%f”，“%f”也代替了每一个传入的参数，与“%s”不同的是，“%f”会将每一个传入的参数转换成“浮点类型”，我们来看一个V

```
[www.zsythink.net]# printf "%s\n" 1 19 18 666
1
19
18
666
[www.zsythink.net]# printf "%f\n" 1 19 18 666
1.000000
19.000000
18.000000
666.000000
[www.zsythink.net]#
```

zsythink.net 朱双印博客

上例中，我们分别使用了“%s”替换符和“%f”替换符格式化了相同的内容，但是格式化后的结果却不同。

“%f”自动将传入的数字添加了点，将传入的数字参数替换成了浮点数。

聪明如你一定想到了，我们可以根据传入参数的不同，使用不同的“格式替换符”去替换。

那么，还有哪些格式替换符呢？常用的格式替换符总结如下。

%s 字符串

%f 浮点格式（也就是我们概念中的float或者double）

%b 相对应的参数中包含转义字符时，可以使用此替换符进行替换，对应的转义字符会被转义。

%c ASCII字符。显示相对应参数的第一个字符

%d, %i 十进制整数

%o 不带正负号的八进制值

%u 不带正负号的十进制值

%x 不带正负号的十六进制值，使用a至f表示10至15

%X 不带正负号的十六进制值，使用A至F表示10至15

%% 表示“%”本身

说完了“格式替换符”，再来说说“转义字符”，刚才的示例中，我们只用到了“\n”这个转义符，还有很多其他的转义符，printf中的转义字符与其他程序中的转义字符同，此处我们只是总结出来，方便大家使用。printf常用的转义符如下。

\a 警告字符，通常为ASCII的BEL字符

\b 后退

\c 抑制（不显示）输出结果中任何结尾的换行字符（只在%b格式指示符控制下的参数字符串中有效），而且，任何留在参数里的字符、任何接下来的参数以及任何串中的字符，都被忽略

\f 换页（formfeed）

\n 换行

\r 回车（Carriage return）

\t 水平制表符

\v 垂直制表符

\\ 一个字面上的反斜杠字符，即"\"本身。

\ddd 表示1到3位数八进制值的字符，仅在格式字符串中有效

\0ddd 表示1到3位的八进制值字符

printf的格式化功能我们还没有说完，先别急着关闭这个网页，继续往下看。

现在，我们已经会基本的使用printf命令了，但是还有很多东西没有细聊，在"深入"之前，我们先"浅出"一下，把刚才的知识应用一遍，动手做一些小例子，为后面坚实的基础。

假设，我想为每个传入的参数添加一对"括号"，并且括号内侧需要有空格，那么我们可以使用如下命令。

```
[www.zsythink.net]# printf "( %s )" 1 19 18 666 ; echo""
( 1 ) ( 19 ) ( 18 ) ( 666 )
[www.zsythink.net]#
```

假设，我们想要将每个传入的参数使用"制表符"隔开，那么，可以使用如下命令。

```
[www.zsythink.net]# printf "%s\t" 1 19 18 666 ; echo""
1      19      18      666
[www.zsythink.net]#
```

好了，我知道你的动手能力一定比我强，所以，你自己玩吧，之前的知识点随便试，我就不再瞎BB了。

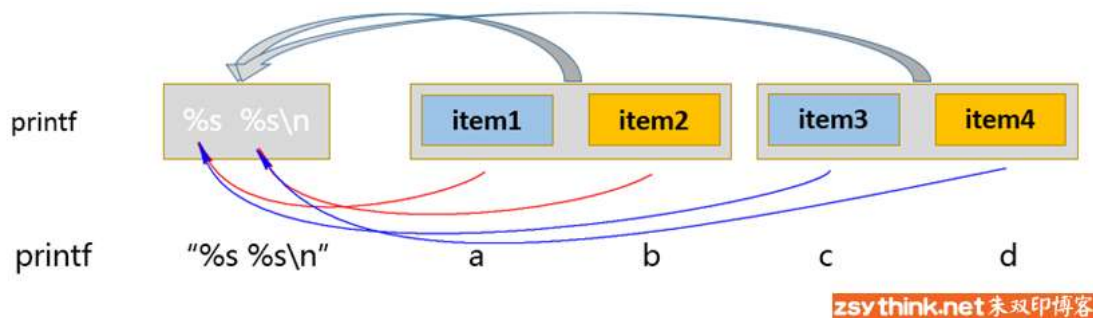
现在，我们继续聊printf。

刚才的举例中，我们使用到的"格式"其实很简单，每个"格式"中都只用到了一个"格式替换符"，现在，我们扩展一下，在格式中设置多个"格式替换符"试试，示例如

```
[www.zsythink.net]# printf "%s\n" a b c d e f
a
b
c
d
e
f
[www.zsythink.net]# printf "%s %s\n" a b c d e f
a b
c d
e f
[www.zsythink.net]#
```

zsythink.net 朱双印博客

看完上图，聪明如你一定想到了，我们所指定的"格式"中所包含的"格式替换符"的数量，就代表每次格式化的参数的数量，每个"格式替换符"与参数都是一一对应的。定的"格式"中包含两个"格式替换符"，那么printf每次进行"格式化"操作时，就会传入两个参数，然后前一个参数对应第一个替换符，后一个参数对应第二个替换符。化操作完成以后，再传入下一波参数，示意图如下



zsythink.net 朱双印博客

那么，我们再验证一遍上述理论，把之前的格式改为如下图中的模样

```
[www.zsythink.net]# printf "%s %s %s\n" a b c d e f
a b c
d e f
[www.zsythink.net]#
```

zsythink.net 朱双印博客

按照之前的理论，因为"格式"中包含3个"格式替换符"，所以每轮格式化都可以一次性格式化3个参数，于是，第一次格式化操作将"%s %s %s\n"替换成了"a b c\n"，格式化后的输出如上图所示。

好了，我们再看一个例子，跟上例其实没有区别，只是传入的数据更贴合实际，所以格式化后的数据可读性更高了。

```
[www.zsythink.net]# printf "%s %s %s\n" 姓名 性别 年龄 尼玛 男 20 尼美 女 18
姓名 性别 年龄
尼玛 男 20
尼美 女 18
[www.zsythink.net]#
```

zsythink.net 朱双印博客

如果你跟我一样，有所谓的强迫症，那么你对上图中的格式化效果肯定不满意，上图中，被格式化以后的数据虽然可读性更高了，但是并没有完全对齐，"年龄"字段快跑到"性别"字段里面了，这怎么能忍？其实我就是说说，真忍也就忍了。不过咱们也可以不这么委屈自己，因为printf有能力"满足"这样的"需求"。（嗯~~~说这想歪了）

我们看看，printf是怎样拯救强迫症患者的，示例如下。

```
[www.zsythink.net]# printf "%s %s %s\n" 姓名 性别 年龄 尼玛 男 20 尼美 女 18
姓名 性别 年龄
尼玛 男 20
尼美 女 18
[www.zsythink.net]# printf "%7s %5s %4s\n" 姓名 性别 年龄 尼玛 男 20 尼美 女 18
姓名 性别 年龄
尼玛 男 20
尼美 女 18
[www.zsythink.net]#
```

zsythink.net 未双印博客

我们只是在原来的"格式替换符"中间加入了特定的数字，貌似显示效果就比刚才好了一点，起码年龄字段对应数字与"年龄"两个字对齐了。那么这些数字是什么意思一个"%7s"中间的7表示当前替换符对应的输出宽度为7个字符宽，如果对应的输出不足7个字符，则用空格补全，如果输出的长度超过7个字符，超出的部分也会显示；"5"表示当前替换符对应的输出宽度为5个字符的宽度。而这些数字，我们可以将其称之为"修饰符"，修饰符会对相应的"替换符"进行修饰。

神马？看右对齐不顺眼，想要左对齐？好吧，满足你！看下图。

```
[www.zsythink.net]# printf "%7s %5s %4s\n" 姓名 性别 年龄 尼玛 男 20 尼美 女 18
姓名 性别 年龄
尼玛 男 20
尼美 女 18
[www.zsythink.net]# printf "%-7s %-5s %-4s\n" 姓名 性别 年龄 尼玛 男 20 尼美 女 18
姓名 性别 年龄
尼玛 男 20
尼美 女 18
[www.zsythink.net]#
```

zsythink.net 未双印博客

与之前的"格式"相比，只是在原来的修饰符前面加入了"-"，"-"表示左对齐，默认不加"-"时表示右对齐，其实"-"也是修饰符。

除了数字和"-"，还有另一种修饰符，就是"+"

"+"可不代表右对齐，因为我们前面已经说过，不加"-"修饰符的时候，默认就是右对齐。那么"+"修饰符表示什么意思呢？看下图你就会明白。

```
[www.zsythink.net]# printf "灵宠名称 体温\n"; printf "%10s %5d \n" 烈火兽 180 冰晶兽 -70
灵宠名称 体温
烈火兽 180
冰晶兽 -70
[www.zsythink.net]# printf "灵宠名称 体温\n"; printf "%10s %+5d \n" 烈火兽 180 冰晶兽 -70
灵宠名称 体温
烈火兽 +180
冰晶兽 -70
[www.zsythink.net]#
```

zsythink.net 未双印博客

如上图所示，在默认没有添加"+"作为修饰符时，烈火兽的体温输出为180，而使用了"+"修饰符以后，烈火兽的体温输出为+180，看到这里你应该明白了，当替换为数字时，可以使用"+"作为修饰符，输出"正数"前面的"正号"。

现在我们已经学会了3种修饰符

"数字"

"+"

"-"

其实数字修饰符还有另外的一些特殊用法，如下图所示。

```
[www.zsythink.net]# printf "灵宠名称 体温\n"; printf "%-10s %-12f \n" 烈火兽 180.5289 冰晶兽 -70.666
灵宠名称 体温
烈火兽 180.528900
冰晶兽 -70.666000
[www.zsythink.net]# printf "灵宠名称 体温\n"; printf "%-10s %-12.3f \n" 烈火兽 180.5289 冰晶兽 -70.666
灵宠名称 体温
烈火兽 180.529
冰晶兽 -70.666
[www.zsythink.net]#
```

zsythink.net 未双印博客

上图中，第一条命令的数字修饰符为12，表示对应的替换符"%f"的输出宽度为12个字符，第二条命令的数字修饰符为12.3，表示对应的替换符"%f"的输出宽度为12个字符，小数点的精度为3。


没错，当替换符为"%f"时，如果数字修饰符带有小数点，则数字修饰符小数点后的数字表示对应小数精度。

而当格式替换符为"%d"时，如果数字修饰符带有小数点，则数字修饰符小数点后的数字表示整数的长度，长度不够时，高位用0补全，如下图所示。

好了，printf命令的用法就总结到这里，个人感觉这篇文章还是比较啰嗦的，但是对于入门的同学来说，这篇文章还是比较详细的，希望这篇文章能够对你有所帮助

你的评论、点赞、收藏和推荐是对我的最大鼓励，客官您要常来光顾哦~~么么哒~~~





**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~