printf

格式化并输出结果。

目录

内建命令

概要

```
printf [-v var] format [arguments]
```

主要用途

- 格式化参数并输出。

选项

```
-v var　将结果输出到变量var中而不是输出到标准输出。
```

参数

format　输出格式。

arguments　一到多个参数。

```
转义序列：除了支持printf(1)和printf(3)的转义序列，内建printf还支持以下转义序列：

%b        展开参数中的反斜杠转义字符。
%q        将参数扩起以用作shell输入。
%(fmt)T   根据strftime(3)中的转义字符来输出日期时间字符串。
```

返回值

返回状态为成功除非给出了非法选项、写错误、赋值错误。

例子

```
# %-5s 格式为左对齐且宽度为5的字符串代替（'-'表示左对齐），不使用则默认右对齐。
# %-4.2f 格式为左对齐宽度为4，保留两位小数。

printf "%-5s %-10s %-4s\n" NO Name Mark
printf "%-5s %-10s %-4.2f\n" 01 Tom 90.3456
printf "%-5s %-10s %-4.2f\n" 02 Jack 89.2345
printf "%-5s %-10s %-4.2f\n" 03 Jeff 98.4323

# 输出
NO    Name       Mark
01    Tom        90.35
```

```
02    Jack       89.23
03    Jeff       98.43


# %b %q %(fmt)T 的例子。
# see it again with a newline.
printf "%s\n" 'hello world'
# 展开换行符，和上面的结果一样。
printf "%b" 'hello world\n'

printf '%q\n' 'a b c'
# 输出
a\ b\ c

# %z为时区，%n为换行符。
printf "%(%F %T %z%n)T"
# 输出
2019-09-10 01:48:07 +0000
```

注意

1. 该命令是bash内建命令，相关的帮助信息请查看help命令。

外部命令

概要

```
printf FORMAT [ARGUMENT]...
printf OPTION
```

主要用途

- 格式化参数并输出。

选项

```
--help 显示帮助信息并退出。
--version 显示版本信息并退出。
```

参数

format：输出格式。

arguments：一到多个参数。

```
在这里忽略了：%b %q：如果你安装的coreutils版本支持它们，那么请参考上面的例子。
支持的转义序列：

\"    双引号
\\    反斜杠
\a          响铃
\b          退格
\c          截断输出
```

```
\e          退出
\f          翻页
\n          换行
\r          回车
\t          水平制表符
\v          竖直制表符
\NNN        八进制数(1到3位数字)
\xHH        十六进制数(1到2位数字)
\uHHHH      Unicode字符附加4位十六进制数字
\UHHHHHHHH  Unicode字符附加8位十六进制数字
%%    百分号

以及'diouxXfeEgGcs'中的一个结尾的C格式规范，将被转换为正确的类型并处理可变宽度。
```

例子

```
# 使用 /usr/bin/printf 确保调用的不是内建命令。
# 当然，在你关闭内建printf以及确认当前环境没有printf函数的情况下，可直接使用printf详见末尾"注意"的链接。

# 按行打印数组和关联数组的下标及值。

# 声明数组可以不加'declare -a'或'local -a'在函数内声明的局部变量）。
arr=('line1' 'line2')
/usr/bin/printf "%s\n" ${!arr[@]}
# 输出下标
0
1
/usr/bin/printf "%s\n" ${arr[@]}
# 输出值
line1
line2

#声明关联数组（也就是字典）必须加'declare -A'或'local -A'在函数内声明的局部变量）。
declare -A assoc_arr=(['key1']='value1' ['key2']='value2')
/usr/bin/printf "%s\n" ${!assoc_arr[@]}
# 输出键。
key2
key1
/usr/bin/printf "%s\n" ${assoc_arr[@]}
# 输出值。
value2
value1
```

返回值

返回状态为成功除非给出了非法选项等。

注意

1. 该命令是GNU coreutils包中的命令，相关的帮助信息请查看man -s 1 printf或info coreutils 'pwd invocation'

2. 启动或关闭内建命令请查看`enable`命令，关于同名优先级的问题请查看`builtin`命令的例子部分的相关讨论。

3. 我通过和`bug-bash@gnu.org`的交流，得到了关于这几个格式说明符`%b %q %(fmt)T`的解释：

> printf(1)中的%b格式说明符是printf(3)支持的格式之外增加的一个POSIX特性。
>
> %q和%T说明符是非标准的，并且不受所有独立实现的printf的支持。

更多细节请参考链接：

- POSIX printf `APPLICATION USAGE`段落的第五节。
- POSIX printf格式说明符 的`Description`段落。