

Shell打印输出命令

2022年10月21日 17:31

1. echo命令

- Shell 的 echo 指令与 PHP 的 echo 指令类似，都是用于字符串的输出。命令格式：echo string
- 您可以使用echo实现更复杂的输出格式控制。

1.1 显示普通字符串

```
echo"It is a test"
```

这里的双引号完全可以省略，以下命令与上面实例效果一致：

```
echoIt is a test
```

1.2 显示转义字符

```
echo"\It is a test\"""#结果将是:#"It is a test"
```

1.3 显示变量

- read 命令从标准输入中读取一行，并把输入行的每个字段的值指定给 shell 变量

```
#!/bin/shreadname
```

```
echo"$nameIt is a test"
```

- 以上代码保存为 test.sh，name 接收标准输入的变量，结果将是：

```
[root@www ~]#sh test.shOK #标准输入OK It is a test#输出
```

1.4 显示换行

```
echo-e "OK! \n"#-e 开启转义echo"It is a test"#输出结果: #OK!##It is a test
```

1.5 显示不换行

```
#!/bin/shecho-e "OK! \c"#-e 开启转义 \c 不换行echo"It is a test"#输出结果: ##OK! It is a test
```

1.6 显示结果定向至文件

```
echo"It is a test">myfile
```

1.7 原样输出字符串，不进行转义或取变量(用单引号)

```
echo'$name\""'#输出结果: #$name\""
```

1.8 显示命令执行结果

```
echo`date`#注意: 这里使用的是反引号 `，而不是单引号 '。#结果将显示当前日期#Thu Jul 24 10:08:46 CST 2014
```

2. printf 命令

- printf 命令模仿 C 程序库 (library) 里的 printf() 程序。
- printf 由 POSIX 标准所定义，因此使用 printf 的脚本比使用 echo 移植性好。
- printf 使用引用文本或空格分隔的参数，外面可以在 * printf 中使用格式化字符串，还可以制定字符串的宽度、左右对齐方式等。默认 printf 不会像 echo 自动添加换行符，我们可以手动添加 \n。

printf 命令的语法：printf format-string [arguments...]

- 参数说明：

- format-string: 为格式控制字符串
- arguments: 为参数列表。

- 实例如下：

```
$ echo"Hello, Shell~Hello, Shell
```

```
$ printf"Hello, Shell\n~Hello, Shell
```

2.1 printf实例

```
#!/bin/bashprintf"%-10s %-8s %-4s\n"姓名 性别 体重kg
```

```
printf"%-10s %-8s %-4.2f\n"郭靖 男 66.1234
```

```
printf"%-10s %-8s %-4.2f\n"杨过 男 48.6543
```

```
printf"%-10s %-8s %-4.2f\n"郭芙 女 47.9876
```

- 执行脚本，输出结果如下所示：

```
姓名      性别      体重kg
```

```
郭靖      男         66.12
```

```
杨过      男         48.65
```

```
郭芙      女         47.99
```

- %s %c %d %f都是格式替代符

- %-10s 指一个宽度为10个字符
- -表示左对齐，没有则表示右对齐，任何字符都会被显示在10个字符宽的字符内，如果不足则自动以空格填充，超过也会将内容全部显示出来。
- %-4.2f 指格式化为小数，其中.2指保留2位小数。

更多实例：

```
#!/bin/bash#format-string为双引号printf"%d %s\n"1 "abc"#单引号与双引号效果一样printf'%d %s\n'1 "abc"#没有引号也可以输出printf%s abcdef
```

#格式只指定了一个参数，但多出的参数仍然会按照该格式输出，format-string 被重用printf%s abc def
printf%s\n"abc def
printf%s %s %s\n"a b c d e f g h i j
#如果没有 arguments, 那么 %s 用NULL代替, %d 用 0 代替printf%s and %d \n"
执行脚本，输出结果如下所示：

```
1 abc
1 abc
abcdefabcdefabc
def
a b c
d e f
g h i
j
and 0
```

2.2 printf的转义序列

序列	说明
\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留在格式字符串中的字符，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	
\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"a string, no processing:<%s>\n""A\nB"a string, no processing:<A\nB>\$ printf"a string, no processing:<%b>\n""A\nB"a string, no processing:<AB>\$ printf"www.runoob.com \a"www.runoob.com \$ #不换行	

3. read读取控制台的输入

- read [选项] (参数)
 - -p: 指定读取值时的提示符
 - -t: 指定读取值时等待的时间（秒），如果没有在指定的时间内输入，就不再等待了。
- 参数: 变量: 指定读取值的变量名