

Shell程序设计 - echo命令

目录

01

命令行参数和语法

02

常用示例

03

进阶用法

目录

01

命令行参数和语法

02

常用示例

03

进阶用法

命令行参数和语法

- echo命令的基本语法如下:

```
echo <options> <string>
```

options表示命令选项。在默认情况下，每执行一次echo命令，都会在输出信息的行尾加上一个换行符。参数string表示要输出的文本，用户可以同时指定多个文本，这些文本之间用空格隔开。

| 选项 | 含义 |
|----|-------------|
| -n | 禁止以换行符结尾 |
| -e | 解析反斜线转义字符 |
| -E | 禁止解析反斜线转义字符 |

命令行参数和语法

■ echo换行符处理

```
echo "What is your first name? "  
read first  
echo "What is your last name? "  
read last  
echo "What is your middle name? "  
read middle  
echo -n "What is your birthday? "  
read birthday  
~
```

```
ot@rocky shell-test]# bash ex5-13.sh  
t is your first name?  
ky^H^H^H^H  
What is your last name?  
roc  
What is your middle name?  
david  
What is your birthday? 2020  
[root@rocky shell-test]#
```

目录

01

命令行参数和语法

02

常用示例

03

进阶用法

常用示例

■ 显示转义字符

- 通过转义字符，echo命令可以控制输出的格式，或者输出某些特殊的字符，例如退格符、换页符，以及制表符等。

| 字符 | 说明 |
|----|---------------------|
| \a | 报警符，相当于ASCII码的BEL字符 |
| \b | 退格符 |
| \f | 换页符 |
| \n | 换行符 |
| \c | 禁止继续输出文本 |
| \r | 回车符 |
| \t | 水平制表符 |
| \v | 垂直制表符 |
| \\ | 反斜线 |

常用示例

■ 显示转义字符

```
#!/bin/bash
# 退格符
echo -e "this is a\b string."
# 禁止继续输出后面的文本
echo -e "hello \c world."
# 换行符
echo -e "hello \n world."
# 使用制表符输出表格
echo -e "Alice\t99"
echo -e "John\t82"
echo -e "Tom\t91"
```

```
[root@rocky shell-test]# bash ex5-14.sh
this is string.
hello hello
world.
Alice 99
John 82
Tom 91
[root@rocky shell-test]#
```

常用示例

■ 显示变量

- 用户可以使用echo语句将程序中的变量值打印出来

```
#!/bin/bash

echo -n "Please input a name:"

read name
#输出变量的值
echo "Hello, $name"

v1="sing"
v2="danc"
#错误的输出变量值的[] 法
echo "We are $v1ing, we are $v2ing."
```

```
[root@rocky shell-test]# bash ex5-15.sh
Please input a name:rocky
Hello, rocky
We are , we are .
[root@rocky shell-test]#
```

常用示例

■ 显示命令输出结果

```
echo <`命令`>
```

```
#!/bin/bash  
  
#显示date命令的执行结果  
echo `date`  
# 显示ls命令的执行结果  
echo `ls`  
~  
~
```

```
[root@rocky shell-test]# bash ex5-16.sh  
2023年 02月 13日 星期一 14:39:00 CST  
ex3-14.sh ex3-15.sh ex3-16.sh ex3-17.sh ex5-10.sh ex5-11.sh e  
x5-12.sh ex5-13.sh ex5-14.sh ex5-15.sh ex5-16.sh ex5-1.sh ex5  
-2.sh ex5-3.sh ex5-4.sh ex5-5.sh ex5-6.sh ex5-7.sh ex5-8.sh e  
x5-9.sh test.sh  
[root@rocky shell-test]#
```

常用示例

■ echo命令执行结果的重定向

```
echo <输出的信息> >/>> <文件>
```

```
#!/bin/bash
```

```
# 将要输出的信息入文件
```

```
echo "Hello, world." > hello.txt
```

```
# 将输出的信息追加到文件的结尾
```

```
echo "Hello, Chunxiao." >> hello.txt
```

```
~
```

```
[root@rocky shell-test]# cat hello.txt  
Hello, world.  
Hello, Chunxiao.  
[root@rocky shell-test]#
```

目录

01

命令行参数和语法

02

常用示例

03

进阶用法

进阶用法

- echo使用高级用法时，必须加上 “\033[“或” \e[“
 - 高级用法基本语法如下

```
echo <高级属性> <字符串>
```

| 属性 | 说明 |
|---------|--------|
| \033[0m | 关闭所有属性 |
| \033[1m | 设置高亮度 |
| \033[4m | 下划线 |
| \033[5m | 闪烁 |
| \033[7m | 反显 |
| \033[8m | 消隐 |
| \033[nA | 光标上移n行 |

进阶用法

| 属性 | 说明 |
|-----------|--------------------------|
| \033[nB | 光标下移n行 |
| \033[nC | 光标右移n行 |
| \033[nD | 光标左移n行 |
| \033[y;xH | 设置光标位置,y表示行数, x表示从第几个字符开 |
| \033[2J | 清屏 |
| \033[K | 清除从光标到行尾的内容 |
| \033[s | 保存光标位置 |
| \033[u | 恢复光标位置 |

进阶用法

- 为字符串设置下划线

```
echo -e "\033[4m字符串\033[0m"
```

- 字符串后所有输出都有下划线

```
echo -e "\033[4m字符串"
```

```
[root@rocky ~]# echo -e "\033[4mtest\033[0m"  
test  
[root@rocky ~]# echo -e "\033[4mtest"  
test  
[root@rocky ~]#
```

进阶用法

- echo可以设置字体和背景颜色，字体和背景颜色均提供有8种颜色。字体颜色开头数字为3，背景颜色开头数字为4。

- 字体颜色常见设置如下表所示：

| 设置 | 颜色 |
|----------|-----|
| \033[30m | 黑色 |
| \033[31m | 红色 |
| \033[32m | 绿色 |
| \033[33m | 黄色 |
| \033[34m | 蓝色 |
| \033[35m | 紫红色 |
| \033[36m | 青色 |
| \033[37m | 白色 |

进阶用法

- 背景颜色常见设置如下表所示：

| 设置 | 背景颜色 |
|----------|------|
| \033[40m | 黑色 |
| \033[41m | 红色 |
| \033[42m | 绿色 |
| \033[43m | 黄色 |
| \033[44m | 蓝色 |
| \033[45m | 紫红色 |
| \033[46m | 青色 |
| \033[47m | 白色 |

进阶用法

- 设置test字体颜色为红色，背景色为青色

```
echo -e "\033[31mtest\033[46m"
```

```
[root@rocky ~]# echo -e "\033[31mtest\033[46m"  
test  
[root@rocky ~]#  
[root@rocky ~]#  
[root@rocky ~]#
```

- 取消当前终端设置的所有的特殊属性，回到默认状态。

```
echo -e "\033[0m"
```

```
[root@rocky ~]#  
[root@rocky ~]# echo -e "\033[0m"  
  
[root@rocky ~]# ls  
1.py re-test.py  
anaconda-ks.cfg rpmbuild  
cgroup-task-check-20210111.sh shell-test
```

谢谢