

---

是什么让一个字符特别？如果它具有超出其字面含义的含义，即元含义，那么我们将其称为特殊字符。除了命令和关键字，特殊字符也是 Bash 脚本的组成部分。

---

## 在脚本和其他地方发现的特殊字符

---

### 注释 '#'

---

注释。以 # 开头的行（#! 除外）是注释，不会被执行。

```
# This line is a comment.
```

---

注释也可能出现在命令结束之后。

```
echo "A comment will follow." # Comment here.  
#                               ^ Note whitespace before #
```

注释也可以跟在行首的空格后面。

```
# A tab precedes this comment.
```

评论甚至可以嵌入到管道中。

```
initial=( `cat "$startfile" | sed -e '/#/d' | tr -d '\n' | \  
# Delete lines containing '#' comment character.  
          sed -e 's/\./\./g' -e 's/_/_/g` )  
# Excerpted from life.sh script
```

命令不能跟在同一行的注释后面。没有终止注释的方法，以便“实时代码”在同一行开始。为下一个命令使用新行。

---

当然，在 echo 语句中引用或转义的 # 不会开始注释。同样，# 出现在某些参数替换结构和数值常量表达式中。

---

```

echo "The # here does not begin a comment."
echo 'The # here does not begin a comment.'
echo The \# here does not begin a comment.
echo The # here begins a comment.

echo ${PATH#*:}      # Parameter substitution, not a comment.
echo $(( 2#101011 )) # Base conversion, not a comment.

# Thanks, S.C.

```

标准的引用和转义字符 (" ' \) 转义 #。

某些模式匹配操作也使用#。

## 命令分隔符 ';'

允许将两个或多个命令放在同一行。

```

echo hello; echo there

if [ -x "$filename" ]; then    # Note the space after the semicolon.
#+                          ^^
    echo "File $filename exists."; cp $filename $filename.bak
else #                          ^^
    echo "File $filename not found."; touch $filename
fi; echo "File test complete."

```

注意 ";" 有时需要转义。

' ; ; '

case 选项中的终止符 [双分号]。

```

case "$variable" in
    abc) echo "\$variable = abc" ;;
    xyz) echo "\$variable = xyz" ;;
esac

```

' . '

"dot" 命令 [句号]。等价于 source (参见示例 15-22)。这是一个内置的 bash。

