

Shell语法基础--各种括号、引号的用法

原创 锦夏挽秋 于 2021-07-18 00:04:07 发布 阅读量999 收藏 2 点赞数

版权

分类专栏: 笔记 文章标签: shell

笔记 专栏收录该内容

0 订阅 3 篇文章

订阅专栏

记录一下Shell 脚本中经常用到的括号、引号表达式。

括号

1. 单层圆括号 ()

1.1 (代码段) 结构

代表命令组，可用括号外定义的变量，但括号中的变量定义和赋值只能在括号中生效。

即：加了一层命名空间。

```
1  aaa="love you"
2  (
3      bbb="fuck you"
4      echo $aaa  #输出 Love you
5      echo $bbb  #输出 fuck you
6      aaa="like you"
7      echo $aaa  #输出 Like you
8  )
9  echo $bbb      #输出空白，因为此处
10 echo $aaa      #输出 Love you
```



锦夏挽秋

关注

0



2



1



1.2 \$(cmd) 结构

代表命令替换，先将括号中命令执行一次，将其标准输出代替其在原语句中的值。

与反引号 `` 作用相同，但推荐用 `$()` 。

```
1 aa=$(date)
2 echo $aa #此处会先执行 date 命令，得到输出 "Sun Jul 18 00:47:37 CST 2021"，然后此结果会替换掉 $aa。
```

1.3 数组赋值，arrayname=(a b c d)

```
1 abc=(This is a example)
2 echo $abc # This is a example
```

1.4 算术运算中表示高优先级

```
1 略
```

2. 双层圆括号 (()), 用于扩展的数学运算

双圆括号命令允许在比较过程中使用高级数学表达式：

((expression))

其中，expression 可以是任意的数学赋值或表达式。相比 test 命令只能在比较中使用简单的算术操作，双圆括号命令提供了更多的数学符号，可以在双小括号中进行各种逻辑运算、数学运算，也支持更多的运算符（如 ++、-- 等）。

常使用的双小括号来在 for 循环中实现 C 语言风格的迭代：

2.1 ((a=\$b+1)) 结构

用于给变量重新赋值，主要是将变量的内容解释为数值类型，然后进行数学运算，结果复制给一个新变量。

如果没有双圆括号，则无法进行数学运算。

```
1 略
```

2.2



锦夏挽秋

关注

👍 0



🌟 2



💬 1



引号

双引号常用于包含一组字符串，在双引号中，除了“\$”、“”、“`”(反引号)有特殊含义外，其余字符（如IFS、换行符、回车符等）没有特殊含义。

```
$ a=3
```

```
$ echo "$a"
```

输出结果为 3，在双引号中 \$ 符仍有特殊含义。

2、单引号 ''

单引号的功能与双引号类似，不过单引号中的所有字符都没有特殊含义：

```
$ a=3
```

```
$ echo '$a'
```

输出结果为 \$a，可见在单引号中 \$ 符是不起作用的。

3、反引号

反引号的功能是命令替换，在反引号 中的内容通常是命令行，程序会优先执行反引号中的内容，并使用运行结果替换掉反引号处的内容。举个例子：

```
$ echo `date`
```

这行命令会先执行反引号中的命令 date，然后用 echo 命令打印出 date 命令的结果（与直接使用 date 命令效果一样）。再举个例子：

复制代码

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

```
a=3
```

```
b=5
```

```
c= `expr $a \* $b`
```

```
echo $c
```

```
exit 0
```

复制代码

4、\$ + 小括号 \$()

\$(...) 的作用与反引号一样，也是命令替换：

复制代码

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

```
for file in $(ls /)
```

```
do
```



锦夏挽秋

关注



0



2



1



```
echo $file
```

```
done
```

```
exit 0
```

复制代码

这个脚本使用 for 循环打印根目录下所有文件的文件名，使用 \$(ls /) 先获得根目录下的所有文件，并将其作为参数列表传给 for 结构。

5、\$ + 双小括号 \$(())

\$(()) 的功能是进行算术运算，括号中的内容为数学表达式，使用 \$(()) 可以求数学表达式的值：

复制代码

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

```
a=3
```

```
b=5
```

```
c=$((a * $b))
```

```
echo $c
```

```
exit 0
```

复制代码

上述脚本的输出结果为 15。

使用 \$(()) 进行数学运算时，不需要担心乘号(*)等运算符被 shell 误解为其他含义，因为它们都在括号内。

6、\$ + 中括号 \$[]

\$[] 的功能与 \$(()) 一样，都是用于算术运算。

7、\$ + 大括号 \${ }

\${ } 的功能是变量替换，类似于 \$ 符，但是 \${

复制代码

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

```
a=3
```

```
b=5
```



锦夏挽秋

关注

👍 0



🌟 2



💬 1



```
echo $ab
echo ${a}b
```

```
exit 0
```

复制代码

这段脚本，第一次输出 \$ab 的时候，会把 ab 视作一个变量，然后打印 ab 的值，显然为空；第二次使用 \${a}b，则会先输出 a 的值然后再向 STDOUT 打印一个字符 b。

8、小括号 ()

小括号可以用来定义一个数组变量，如下：

```
array1=(1 2 3 4 5)           // 在 shell 中定义一个数组变量
array2=(one two three four five)
```

取数组元素的值也是使用 \$ 符号，如下：

复制代码

```
$ echo $array1
$ 1
$
$ echo ${array1[2]} # 取数组中的一个元素，索引用方括号括起来，和大部分语言一样，数组的索引是从 0 开始的
$ 3
$
$ echo ${array2[0]}
$ one
$
$ echo ${array2[*]} # 输出整个数组
$ one two three four five
```

复制代码

这里使用了 \${} 表达式，进行变量替换。



锦夏挽秋

关注

👍 0



🌟 2



💬 1



注意：在 shell 中使用数组变量有时会引起一些问题，而且数组变量的可移植性并不好，因此在 shell 编程中，数组变量使用得并不多。

9、双小括号 (())

复制代码

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

```
for ((i = 0; i < 10; i++))  
do  
echo -n "$i "  
done  
echo ""
```

exit 0

复制代码

10、中括号 []

单个的中括号的功能与 test 命令一样，都是用作条件测试。

复制代码

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

```
read -p "please enter a number: " num
```

```
if [ $num -gt 10 ]; then  
echo "num > 10"  
else  
echo "num <= 10"  
fi
```

复制代码

11、双中括号 [[]]

双中括号提供了针对字符串比较的高级特性，

复制代码

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

```
if [[ hello == hell* ]]; then  
echo "equal"  
else
```



锦夏挽秋

关注



0



2



1



```
echo "unequal"
```

```
fi
```

```
exit 0
```

复制代码

12、大括号 {}

大括号用于括起一个语句块。如果需要在某些只能使用单个语句的地方（如AND、OR列表中）使用多条语句，则可以用大括号将这多条语句括起来构造一个语句块。

总结：

功能 符号

引用字符串，字符串中部分特殊符号有意义 双引号 ""

引用字符串，字符串中特殊符号全都没有意义 单引号 ''

命令替换 反引号 `、\$()

算术运算 (()、[]、(())

变量替换 \${}

数组初始化 ()

条件测试 []

字符串比较 [[]]

括起一个语句块 {}

mysql-shell-8.0.18-windows-x86-64bit.zip

01-12

mysql-shell-8.0.18-windows-x86-64bit.zip MySQL Shell is an advanced command-line client and code editor for MySQL. In addition to SQL, MySQL Shell also offers scripting capabilities for Jav...

shell中 #*, ##*, %*, %%*的含义及用法 最新发布

szw906689771的博客 535

此时可以看出，最后一个/以及左边的全被删除了。可以看出 左边第一个/和它左边的东西删除了 此时 第一条/和它左边的都被删除了 是小/和它左边的就是找的第一个/

1 条评论



大家一起学编程 (python)



锦夏挽秋

关注

0

2

1

1

【精选】shell脚本详解(一)——初学shell脚本必看哦_宝贝富贵猪的博客...

10-19

要求文件必须有执行(x)权限 方法二:sh 、bash脚本文件路径(这种方式可以不对脚本文件添加执行权限) bash first.sh 或 sh first.sh 指定shell来解释脚本,不要求文件必须有写(w)的权限 方法三:sourc...