

let

简单的计算器，执行算术表达式。

概要

```
let arg [arg ...]
```

主要用途

- 执行一个或多个算术表达式。

参数

arg 算术表达式

返回值

当let最后一个执行的表达式的计算结果为0时返回1，否则返回0。当let执行的表达式的除数为0时，返回1并报错。

运算符优先级递减表

<code>`id++, id--`</code>	<code>`变量后增量、变量后减量`</code>
<code>`++id, --id`</code>	<code>`变量预增量、变量预减量`</code>
<code>`-, +`</code>	<code>`正号、负号`</code>
<code>`!, ~`</code>	<code>`逻辑否、按位取反`</code>
<code>`**`</code>	<code>`幂运算`</code>
<code>`*, /, %`</code>	<code>`乘法、除法、取余`</code>
<code>`+, -`</code>	<code>`加法、减法`</code>
<code>`<<, >>`</code>	<code>`按位左移、右移`</code>
<code>`<=, >=, <, >`</code>	<code>`比较`</code>
<code>`==, !=`</code>	<code>`等于、不等于`</code>
<code>`&`</code>	<code>`按位与`</code>
<code>`^`</code>	<code>`按位异或`</code>
<code>` `</code>	<code>`按位或`</code>
<code>`&&`</code>	<code>`逻辑与`</code>
<code>`  `</code>	<code>`逻辑或`</code>
<code>`expr ? expr : expr`</code>	<code>`条件运算符（三元运算符）`</code>
<code>`=, *=, /=, %=, +=, -=, <br><=, >=, &=, ^=,  =`</code>	<code>`赋值`</code>

例子

```
# 尝试直接在终端中执行算术表达式（就像在python的IDLE）
3+4
bash 3+4 command not found...
# 换一种方式。
3 + 4
```

```
bash$ command not found...  
# 看来不行。
```

```
# let命令赋值。  
let a=3**4  
echo ${a}  
# 显示81。  
# ((...))和let命令等效。  
((a=3**4))
```

```
# let常用于变量赋值，而外部命令expr可直接返回表达式的值。  
let 3+4  
# 没有显示7。  
# 执行后显示7，注意空格。  
expr 3 + 4
```

```
# 条件表达式。  
if ((8>4)); then  
    echo '8 is greater than 4.'  
else  
    echo 'error'  
fi  
# 注意空格。  
if [[ 12 -le 10 ]]; then  
    echo 'error'  
else  
    echo '12 is greater than 10.'  
fi
```

```
# 可以通过declare命令设置整型属性的方法来进行算术运算。  
# local命令与此类似。
```

```
# 没有指定整型属性，输出为字符串'a+b'  
declare a=3 b=4 c  
c=a+b  
echo ${c}  
# 不过可以使用以下方式赋值。  
c=$((a+b))  
echo ${c}  
# 显示7
```

```
# 设置了整型属性就可以直接加了。  
declare -i a=3 b=4 c  
c=a+b  
echo ${c}  
# 同上。  
declare -i a  
a=2*3  
echo ${a}  
# 显示6。
```

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。
2. 执行算术计算的命令除了let，还有外部命令expr[]bc等。

From:

<https://rd.irust.top/> - 学习笔记

Permanent link:

<https://rd.irust.top/doku.php?id=command:let>

Last update: **2021/10/15 14:58**

