

readonly

标记shell变量或函数为只读

语法

```
readonly [-aAf] [name[=value] ...]  
readonly -p
```

主要用途

- 定义一到多个变量并设置只读属性。
- 为已定义的一到多个变量设置只读属性。
- 显示全部包含只读属性的变量。
- 为已定义的一到多个函数设置只读属性。
- 显示全部包含只读属性的函数。

选项

```
-a 指向数组。  
-A 指向关联数组。  
-f 指向函数。  
-p 显示全部只读变量。  
--：在它之后的选项无效。
```

参数

```
name[可选]：变量名或函数名  
value[可选]：变量的值
```

返回值

readonly返回true除非你提供了非法选项或非法名称。

例子

```
# 定义变量并增加只读属性  
readonly var1=13 var2  
readonly -a arr1=(1 2 3 4 5) arr2=('z' 'x' 'c')  
# 必须有 '-A' 选项  
readonly -A dict1(['key1']='value1')
```

```
# 先定义变量、函数，然后再为它们添加只读属性  
max=3  
readonly max
```

```
# 数组定义时可以不加 `declare -a`  
seasons=('spring' 'summer' 'autumn' 'winter')  
# 为数组添加只读属性时可以不加 `-a` 选项
```

readonly seasons

```
declare -A man=([ 'age' ]=23 [ 'height' ]='190cm')
```

为关联数组添加只读属性时可以不加 ``-A`` 选项

```
readonly man
```

```
function foo(){ echo 'bar'; }
```

为函数添加只读属性时必须加 ``-f`` 选项

```
readonly -f foo
```

显示全部只读变量，以下两个命令的显示结果一样

```
readonly
```

```
readonly -p
```

显示全部拥有只读属性的数组

```
readonly -a
```

显示全部拥有只读属性的关联数组

```
readonly -A
```

显示全部拥有只读属性的函数

```
readonly -f
```

常见错误

对于只读变量而言，若用户对其值进行修改，则会立即报错。例如，使用该指令定义一个只读变量"test"并且将其值初始化为"ok"输入如下命令：

```
[root@localhost ~]# readonly test='ok' #定义只读变量并初始化
```

那么当用户直接修改该只读变量时就会报错，如下所示：

```
[root@localhost ~]# test='my' #试图修改只读变量的值
-bash: test: readonly variable
```

当用户试图修改只读变量的值时，会被提示该变量为只读变量。

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. declare +r不能去除只读属性，unset不能删除只读变量。

From:
<https://rd.irust.top/> - 学习笔记

Permanent link:
<https://rd.irust.top/doku.php?id=command:readonly>

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

