# Shell命令附表

|  |  |  |  |
| --- | --- | --- | --- |
| readonly | 只读 |  |  |
| unset | 删除变量 |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 基础

## 注释

单行注释格式

**# 注释内容...**

多行注释格式

**:<<!**

**注释内容...**

**!**

注意#!写在文档开头，规定脚本运行环境，分为Bourne Shell（/usr/bin/sh或**/bin/sh**），Bourne Again Shell（/bin/bash），C Shell（/usr/bin/csh），K Shell（/usr/bin/ksh），Shell for Root（/sbin/sh）；一般用红色那个**#! /bin/sh**。

## 变量

### **默认变量**

$# 传递进脚本的参数个数

$n n从1开始，对应传入的第n个参数

$$ 脚本运行当前的进程号

$? 命令退出状态，0正确，其他值表示有错误

### **变量定义**

* “=”左右不能有空格（num = 10**×**，num=10**√**）
* 变量名不能含标点符号（下划线除外）。

Shell变量有string，number，一维数组。String用'或"均可。

# 数字

num=81;

# 字符串

str='stringtest00';

# 数组

arr=('jack' 'tom' 'alice' 'jarry');

### **锁定变量**

readonly 变量名；锁定后变量的值无法再次被修改

### **删除变量**

unset b变量名；删除变量名后变量无法再使用，但删不了锁定变量。

### **使用变量**

格式：${变量名}

### 数组使用

取整个数组：${数组名[@]}

取某个元素：${数组名[下标]}

取数组长度：${**#**数组名[@]}

取元素长度：${**#**数组名[下标]}

## 向shell执行时传参

参数传递格式：./xxx.sh param1 param2 param3...

参数用：$1,$2,$3...

$# 传递到脚本的参数个数

$\* 以一个单字符串显示所有向脚本传递的参数。如"$\*"用「"」括起来的情况、以"$1 $2 … $n"的形式输出所有参数。

$$ 脚本运行的当前进程ID号

$! 后台运行的最后一个进程的ID号

$@ 与$\*相同，但是使用时加引号，并在引号中返回每个参数。如"$@"用「"」括起来的情况、以"$1" "$2" … "$n" 的形式输出所有参数。

$- 显示Shell使用的当前选项，与set命令功能相同。

$? 显示最后命令的退出状态。0表示没有错误，其他任何值表明有错误。

## 执行运算expr

|  |  |  |
| --- | --- | --- |
| 运算符 |  | 示例 |
| + | 加法（数字） | `expr $a + $b` 结果为 30。 |
| - | 减法（数字） | `expr $a - $b` 结果为 -10。 |
| \* | 乘法，\*前必须加‘\’（数字） | `expr $a **\\*** $b` 结果为 200。 |
| / | 除法（数字） | `expr $b / $a` 结果为 2。 |
| % | 取余（数字） | `expr $b % $a` 结果为 0。 |
| = | 赋值（数字） | a=$b 将把变量 b 的值赋给 a。 |
| == | 相等（数字） | [ $a == $b ] 返回 false。 |
| != | 不相等（数字） | [ $a != $b ] 返回 true。 |
| -eq | 相等 | [ $a -eq $b ] 返回 false |
| -ne | 不相等 | [ $a -ne $b ] 返回 true |
| -gt | 大于 | [ $a -gt $b ] 返回 false |
| -lt | 小于 | [ $a -lt $b ] 返回 true |
| -ge | 大于等于 | [ $a -ge $b ] 返回 false |
| -le | 小于等于 | [ $a -le $b ] 返回 true |
| ! | 非 |  |
| -o | 或 |  |
| -a | 与 |  |

**计算整型**

计算整型格式1

let k=`expr $i + $j`

let k=`expr $i - $j`

let k=`expr $i \\* $j`

let k=`expr $i / $j`

let k=`expr $i % $j`

计算整型格式2

let k=i+j

let k=i-j

let k=i\*j

let k=i/j

let k=i%j

## 判断if...else...

注意：判断的前后以及条件与被比较值之间都要加上空格，不然报错！

**if [** condition1 ]

**then**

command1

**elif [** condition2 **]**

**then**

command2

**else**

commandN

**fi**

**if [** condition1 ]

**then**

command1

**else**

command2

**fi**

## 循环

for i in …

do

...

done

while []

do

…

done

## 函数

函数定义

funName(){

......

}

Shell函数没有域的概念，函数中定义的变量在函数外仍可使用（shell也提供函数return语法）。Shell函数仅支持返回整数型数据，返回字符串型会有警告。**函数( )中不能写任何参数**，传参数到函数直接用，比如$1,$2,$3之类的。

**定义**

fun(){

echo “${1} ${2} ${3} $4 $5”

return `expr ${1} \\* ${2} \\* {3} \\* {4} \\* {5}`

}

**调用【函数必须定义在使用前**，而且调用函数这样调用】

无参函数：fun

有参函数：fun 1 2 3 4 5

有返回值函数：res=`fun 1 2 3 4 5`

## 文件引入

引入方式：srouce . 文件路径（绝对或相对）（如：. ./test01.sh，source ./test01.sh）

引入必须写在头部，可以随意使用被引入的shell中的方法和变量。

# 定时服务

## 定时任务

### **添加定时任务**

命令：crontab -e 进入定时任务编辑器添加定时任务，格式为：f1 f2 f3 f4 f5 . shell路径+program

f1 分钟（0-59）

f2 小时（0-23）

f3 月份（1-12）

f4 一个月中第几天（1-31）

f5 一周中的星期几（0-7（0代表周日）或sun,mon,tue,wed,thu,fri,sat）

【\*取值范围内所有数字。\*/n 每过n数字。x-z 从x到z。, 散列时间】，:q退出后下一分钟生效。

### **查看定时任务**

命令：crontab -l

### **判断上一个shell任务是否成功**

$? 返回0代表上一个shell执行成功。

启动进程

监视线程

关闭线程