## 运行 Shell 脚本有两种方法

### 作为可执行程序

代码保存为 test.sh，并 cd 到相应目录：

chmod +x ./test.sh #使脚本具有执行权限

./test.sh #执行脚本

### 作为解释器参数

直接运行解释器，其参数就是 shell 脚本的文件名：

/bin/sh test.sh

/bin/php test.php

### Shell 注释

以"#"开头的行就是注释，会被解释器忽略。

sh里没有多行注释，只能每一行加一个#号。

## Shell 变量

### 变量命名赋值

与java一致

变量名、等号、值之间不能有空格。

a=1;

### 使用变量

在变量名前面加$，使用定义过的变量。

name=yun

echo $name #输出yun

echo ${name} #推荐加{}，帮助解释器识别变量边界

再次给变量赋值，不需使用$符号。

name=lin;

### 只读变量

使用 readonly 命令可以将变量定义为只读变量，只读变量的值不能被改变。

a=1;

readonly a;

a=2; #报错

### 删除变量

使用 unset 命令可以删除变量。语法：

unset a

变量被删除后不能再次使用。unset 命令不能删除只读变量。

### 变量类型

运行shell时，会同时存在三种变量：

1) 局部变量 局部变量在脚本或命令中定义，仅在当前shell实例中有效，其他shell启动的程序不能访问局部变量。

2) 环境变量 所有的程序，包括shell启动的程序，都能访问环境变量，有些程序需要环境变量来保证其正常运行。必要的时候shell脚本也可以定义环境变量。

3) shell变量 shell变量是由shell程序设置的特殊变量。shell变量中有一部分是环境变量，有一部分是局部变量，这些变量保证了shell的正常运行

## Shell 字符串

字符串可以用单引号，也可以用双引号，也可以不用引号。

### 单引号

str='this is a string'

单引号字符串的限制：

单引号里的任何字符都会原样输出，单引号字符串中的变量是无效的；

单引号字串中不能出现单引号（对单引号使用转义符后也不行）。

### 双引号

your\_name='qinjx'

str="Hello, I know your are \"$your\_name\"! \n"

双引号的优点：

双引号里可以有变量

双引号里可以出现转义字符

### 拼接字符串

直接拼接，无操作符。

your\_name="qinjx"

greeting="hello, "$your\_name" !"

greeting\_1="hello, ${your\_name} !"

echo say$greeting $greeting\_1

### 获取字符串长度

操作符#，${#varName}

string="abcd"

echo ${#string} #输出 4

### 提取子字符串

操作符:，格式：${var:start[:end]}；索引从1开始，不含头只含尾。

string="runoob is a great site"

echo ${string:1:4} # 输出 unoo

### 查找子字符串

格式：`expr index “${var}” substr`

string="runoob is a great company"

echo `expr index "$string" is` # 输出 8

注意： 以上脚本中 "`" 是反引号，而不是单引号 "'"。

## Shell 数组

bash支持一维数组（不支持多维数组），并且没有限定数组的大小。

类似与C语言，数组元素的下标由0开始编号。获取数组中的元素要利用下标，下标可以是整数或算术表达式，其值应大于或等于0。

### 定义数组

在Shell中，用括号来表示数组，数组元素用"空格"符号分割开。格式：

arr=(val1 val2...)

arr=(1 2 3 4)

arr1[0]=1; arr1[2]=2;

可以不使用连续的下标，而且下标的范围没有限制。

### 读取数组

格式：

${数组名[下标]}

valuen=${arr[0]} #1

下标为@、\*符号时，获取数组中的所有元素，例如：

echo ${arr[@]} #1 2 3 4

### 获取数组的长度

格式：

${#数组名[@/\*]} #获取数组长度

${#数组名[index]} #获取数组单个元素长度

echo ${#arr[@]} #4

echo ${#arr[1]} #1

## Shell 基本运算符

算数运算符：+，-，\\*(需转义)，/，%，=，==，！=

格式：`expr 2 + 2`；expr命令，反引号，表达式和运算符之间必须有空格。

关系运算符：-eq，-ne，-gt，-lt，-ge，-le

格式：[10 -eq 20]，中括号括起，空格。只支持数值。

布尔运算符：!，-o(或)，-a(与)

格式：[ 10 -eq 20 -o 1 eq 1]，一个中括号，空格

逻辑运算符：||，&&

格式：[[ 10 -eq 20 || 1 eq 1]]，两个中括号，空格

字符串运算符：=，!=，-z(长度为0)，-n(长度不为0)，str([$strVar]，不为空返回true)

格式：[‘aa’ = ‘bb’]，[-z $str]，，中括号括起，空格

文件测试运算符：

-b