变量与参数

笨办法学

概述

- 变量替换 Variable Substitution
- 变量赋值 Variable Assignment
- Bash变量是不区分类型的 Bash Variables Are Untyped
- 特殊的变量类型 Special Variable Types

$ 变量替换(引用变量的内容). Variable substitution

- 变量的名字就是变量保存值的地方
- 引用变量的值就叫做变量替换
- 在以下情况下，变量名没有前缀名$
  - 变量被声明或被赋值
  - 变量被**unset**
  - 变量被**export**
  - 变量代表一种信号
- 在引用时，$的变化：
  - 双引号(" ")，弱引用，发生变量替换
  - 单引号(' ')，强引用，保持字面意思

```
bash$ variable1=23

bash$ echo variable1
variable1

bash$ echo $variable1
23

bash$ echo ${variable1}
23
```

变量的名字 vs.变量的值

变量赋值 Variable Assignment

变量的赋值有多种方法：

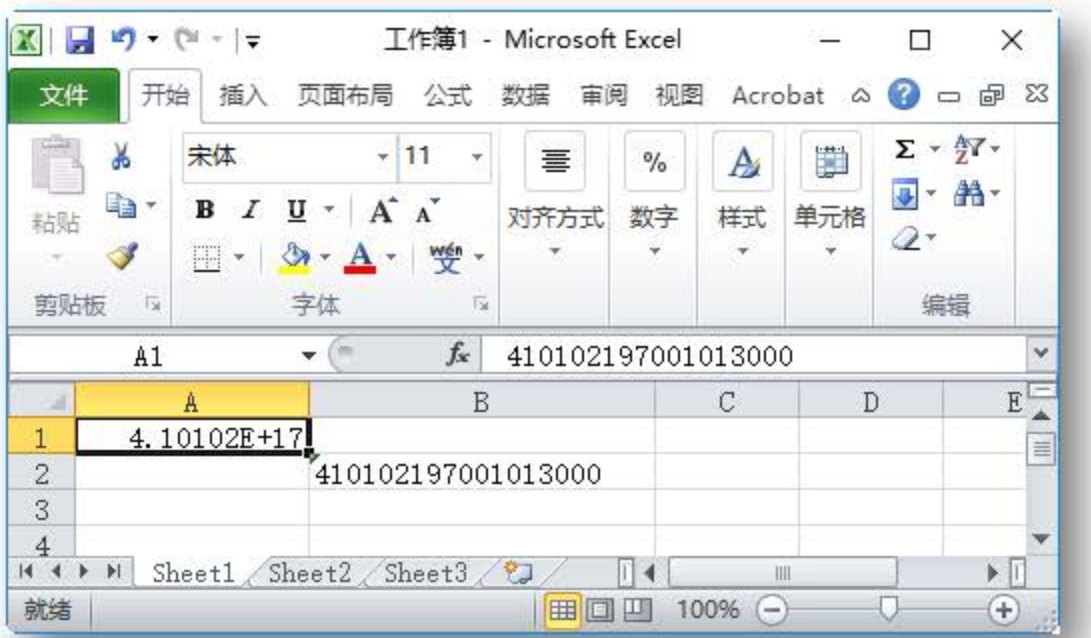
- 使用=
- 使用let
- 使用for循环
- 使用read

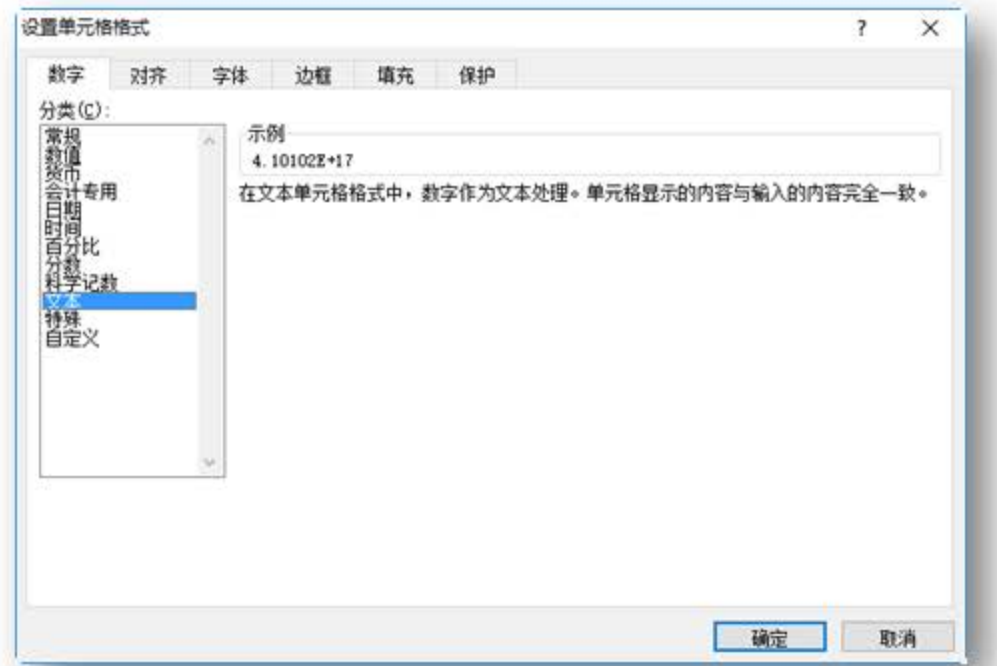
- 使用命令替换
  - `command`
  - $(...)

```
1 var1=111
2 let var2=1+2
3 for var3 in 1  2 3
4 read var4
5 var5=`uname -a`
6 var6=$(cat /etc/redhat-release)
```

Bash变量是不区分类型的 Untyped

- 本质来说，Bash变量都是字符串
- 根据不同的上下文环境，Bash也允许比较操作和整数操作
  - 如果全部是“数字”，可以被当作整型
  - 否则，就是字符串





特殊的变量类型

- 局部变量 Local variables
  - 只有在代码块或者函数中才可见
- 环境变量 Environmental variables
  - 环境变量将影响用户接口和Shell的行为
  - 需要将变量`export`，才能成为环境变量
  - 添加新的或者更现有环境变量，会立即生效
  - 子进程是**不能**通过`export`变量来影响产生自己的父进程的环境的
- 位置参数 Positional parameters

() 圆括号 parenthesis

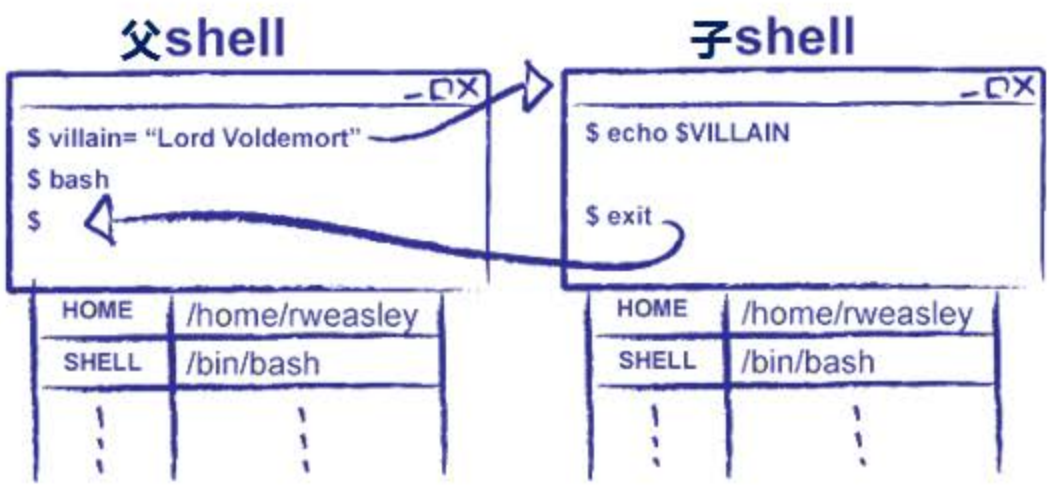
- 命令组

```
1 bash$ a=123

2 bash$ (echo "a = $a"; a=321; echo "a = $a" )
  a = 123
  a = 321

3 bash$ echo "a = $a"
  a = 123
```

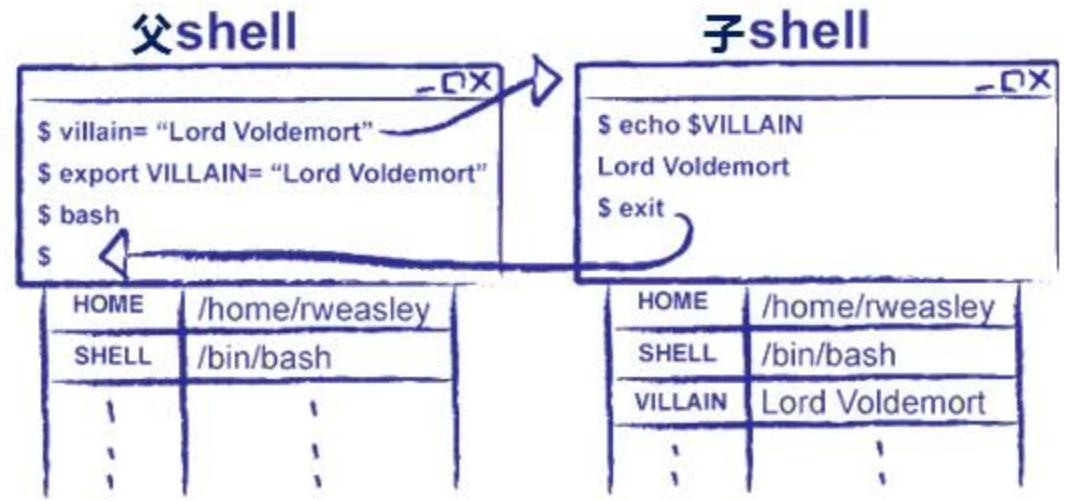
- 数组初始化

```
bash$ Array=(element1 element2 element3)
```

export示例




```
1  $ VILLAIN="Lord Voldemort"
2  $ bash
3  $ echo $VILLAIN

4  $ exit
```




```
1  $ export VILLAIN="Lord Voldemort"
2  $ bash
3  $ echo $VILLAIN
   Lord Voldemort
4  $ exit
```

特殊的变量类型

- ~~局部变量 Local variables~~
  - ~~只有在代码块或者函数中才可见~~
- ~~环境变量 Environmental variables~~
  - ~~环境变量将影响用户接口和Shell的行为~~
  - ~~需要将变量`export`，才能成为环境变量~~
  - ~~添加新的或者更现有环境变量，会立即生效~~
  - ~~子进程是**不能**通过`export`变量来影响产生自己的父进程的环境的~~
- 位置参数 Positional parameters

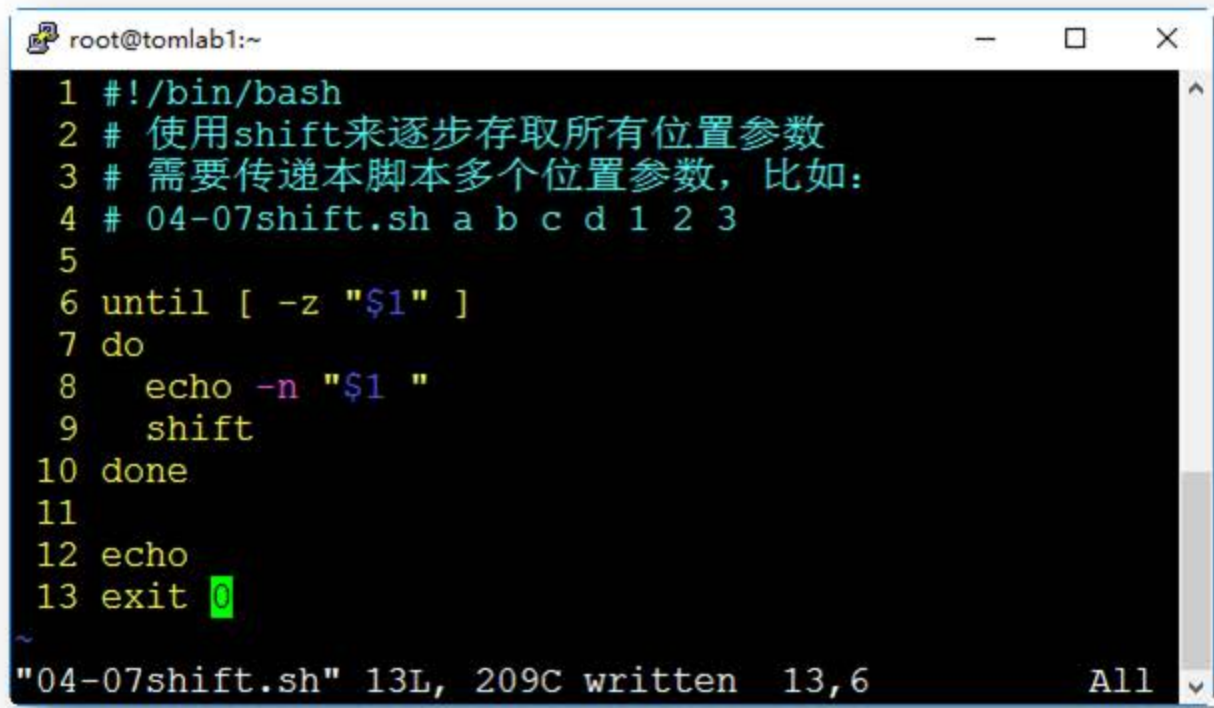
$#,$*,$@,...位置参数

- $0, $1, $2, 等等
  - 位置参数：从命令行传递到脚本，或者传递给函数
  - $0是脚本文件自身的名字，$1是第一个参数，$2是第二个参数.....
  - $9之后的必须用大括号括起来了。比如，${10}, ${11}, ${12}
- $#
  - 命令行参数或者位置参数的个数
- $*
  - 所有的位置参数都被看作为一个单词
- $@
  - 与$*相同，但是每个参数都是一个独立的引用字符串

./ScriptName

shift命令会重新分配位置参数

- 把所有的位置参数都向左移动一个位置
- **$1 <--- $2, $2 <--- $3, $3 <--- $4……**
- 原来的$1就消失了，但是$0（脚本名）不会改变



使用${}参数替换来实现默认参数

- 如果变量未被声明或赋值，那么就替换为默认值
- 语法：
  - ${parameter-default}
  - ${parameter:-default}

```
echo ${username-`whoami`}
```

```
DEFAULT_FILENAME=generic.data
filename=${1:-$DEFAULT_FILENAME}
```

总结

- 变量替换 Variable Substitution
- 变量赋值 Variable Assignment
- Bash变量是不区分类型的 Bash Variables Are Untyped
- 特殊的变量类型 Special Variable Types

其它课程


KVM
虚拟化
进阶与提高


Linux
开源虚拟化 KVM
极速入门


群集原理与实践精讲
Linux
24/7 service
LINUX Cluster
Theory AND Practice


做实验 ①
学 ②
存储


从零开始
构、建
你自己的 Linux
Linux


OpenStack
极速入门