

# shell编程 - Day01笔记

## ■ Shell格式

- ```
1 1、扩展名：xxx.sh
2 2、正文第一行必须指定解释器：#!/bin/bash
```

## ■ shell执行方式

- ```
1 # 方式一：加权限， ./xxx.sh 执行
2 1、 chmod +x xxx.sh
3 2、 ./xxx.sh
4
5 # 方式二：手动指定解释器
6 bash xxx.sh
```

## ■ 自定义变量

- ```
1 # 1. 定义变量
2 变量名=值      ----> 注意：=两侧绝对不能有空格
3 eg1: name="take me to your heart"
4
5 # 2. 调用变量的格式
6 echo $变量名
7
8 # 01_variable_1.sh
9 name="风云雄霸天下"
10 echo $name
11
12 # 3. 小细节：单引号和双引号的区别
13 单引号：无法获取变量的值
14 双引号：可以获取变量的值
```

## ■ 环境变量+位置变量+预设变量

```

1 # 环境变量
2 echo $USER -- 当前用户
3 echo $UID -- 当前用户的UID号
4 echo $PWD -- 当前路径
5 echo $PATH -- 命令搜索路径
6
7 # 位置变量
8 $1 $2 $3 ... .. shell的位置变量
9
10 # 预定义变量
11 $# $* $?
12
13 # $? : 返回上一条命令执行的状态(0代表正确,非0代表失败)

```

## 练习

```

1 输出$1+$2,例如输出结果: 3+5
2 #!/bin/bash
3 echo $1+$2

```

### ■ 变量赋值 - 接收用户从终端输入的值

```

1 # 语法格式
2 read -p 提示信息 变量名
3
4 # 05_read.sh
5 #!/bin/bash
6 read -p 请输入姓名: name
7 echo "您输入的姓名是:$name"
8
9 # 指定超时时间
10 read -p 提示信息 变量名
11 read -t n -p 提示信息 变量名
12
13 # 06_read_t.sh 限制3秒中之内必须输入值
14 #!/bin/bash
15 read -t 3 -p 请输入用户名: username

```

## 练习

```

1 1、输入学生姓名: 赵敏
2 2、输入学生成绩: 88
3 3、输出: 赵敏的成绩为88分

```

### Ubuntu设置sudo免密码

```

1 # 通过更改 /etc/sudoers 实现
2 1、备份: sudo cp /etc/sudoers .
3 2、修改: sudo vi /etc/sudoers
4 添加: tarena ALL=(ALL) NOPASSWD : ALL

```

## 练习

1 | 请使用位置变量的方式实现用户名的创建

## ■ shell - 算术运算符

```
1 # 运算符
2 1、+ - * / %
3 2、++ : 自加1运算,类似于python中 i++ 等同于 i+=1
4 3、-- : 同++
5
6 # 运算命令
7 1、let 运算表达式
8   i=1
9   let i++
10  echo $i
11 2、expr 运算表达式
12   i=1
13   sum=`expr $i + 5` # +两侧要有空格
14   echo $sum
15 3、$[]
16   echo ${1+1}
17   echo ${1-1}
18   echo ${a+a} # 调用变量不用多次添加$符号
19   echo ${1*1} # 乘法无需转义
```

## 练习

```
1 | 使用 位置变量+以上方法一、二中任何一种,实现2个数字的相加
2 | #!/bin/bash
3 | echo ${1+2}
4 | echo `expr $1 + $2`
```

## ■ shell - 比较运算符

```
1 # 语法格式
2 [ 判断语句 ] # 注意括号必须有空格
3 # 组合
4 && 并且
5 || 或者
6 A命令 && B命令 //仅当A成功时才执行B
7 A命令 || B命令 //仅当A失败时才执行B
8 思考: [ a == a ] && echo Y || echo N 代表什么意思?
9
10 # 1、字符比较
11 [ A == A ] #相等(等号两边需要有空格)
12 [ A != B ] #不相等
13 [ -z $变量 ] #判断是否为空
14 思考(Y 还是 N): [ $USER == tarena ] && echo Y || echo N
15
16 练习: 用户输入用户名,不存在时则创建,否则不做任何操作
17 #!/bin/bash
18
19 # 2、数字比较
20 -eq 等于(equal)
21 -ne 不等于(not equal)
```

```

22  -gt 大于(greater than)
23  -ge 大于等于(great or equal)
24  -lt 小于(less than)
25  -le 小于等于(less or equal)
26  思考输出什么:
27  [ 10 -eq 10 ] && echo Y || echo N
28  [ 11 -le 10 ] && echo Y || echo N
29  [ 12 -ge 3 ] && echo Y || echo N
30
31  # 3、文件|目录比较
32  [ -e 文件或目录 ]      #是否存在exist
33  [ -f 文件 ]            #存在且为文件file
34  [ -d 目录 ]            #存在且为目录directory
35  [ -r 文件或目录 ]      #判断是否可读read
36  [ -w 文件或目录 ]      #判断是否可写write
37  [ -x 文件或目录 ]      #判断是否可执行
38  思考输出:
39  [ -e /etc/passwd ] && echo Y || echo N
40  [ -f /etc/passwd ] && echo Y || echo N
41  [ -d /etc/passwd ] && echo Y || echo N

```

## if分支结构

```

1  # 1、单分支语法格式
2      if 判断 ;then
3          命令
4      fi
5
6  # 2、双分支语法格式
7      if 判断 ;then
8          命令1
9      else
10         命令2
11     fi
12 # 3、多分支语法格式
13     if 判断;then
14         命令1
15     elif 判断 ;then
16         命令2
17     else
18         命令3
19     fi
20 # 示例
21 #!/bin/bash
22 if [ $USER == tarena ];then
23     echo "Yes,You are Tarena."
24 else
25     echo "You are other man."

```

## 练习

1 使用shell编写猜数字游戏,无须循环

## for循环

```
1 # 语法格式
2 for 变量 in 值序列
3 do
4     命令
5 done
6 # 示例
7 for i in 1 2 3 4 5
8 do
9     echo "hello world"
10 done
```

## 练习

```
1 把猜数字游戏改为for循环, 猜测100次
2
3 # 练习:判断指定网段的IP地址哪些可以用, 哪些不能用?
```

## ■ while循环

```
1 # 语法格式
2 while 条件判断
3 do
4     命令
5 done
6
7 # 示例
8 #!/bin/bash
9 i=1
10 while [ $i -lt 5 ]
11 do
12     echo baby
13     let i++
14 done
```