Shell 编程：探索 Shell 的基本概念与用法

Shell 简介

Shell 是一个用 C 语言编写的程序，它是用户使用 Linux 的桥梁。Shell 既是一种命令语言，又是一种程序设计语言。

Shell 是指一种应用程序，这个应用程序提供了一个界面，用户通过这个界面访问操作系统内核的服务。

Shell 脚本（shell script），是一种为 shell 编写的脚本程序，shell 和 shell script 是两个不同的概念。

Shell 脚本

打开文本编辑器(可以使用 vi/vim 命令来创建文件)，新建一个文件 test.sh，扩展名为 sh（sh代表shell），扩展名并不影响脚本执行，见名知意就好，如果你用 php 写 shell 脚本，扩展名就用 php 好了。

Bash

```
#!/bin/bash
echo "Hello World !"
```

#! 是一个约定的标记，它告诉系统这个脚本需要什么解释器来执行，即使用哪一种 Shell。

echo 命令用于向窗口输出文本。

Shell 脚本运行

1、作为可执行程序

Bash

```
chmod +x ./test.sh  #使脚本具有执行权限

./test.sh  #执行脚本
```

2、作为解释器参数

Bash

```
/bin/sh test.sh

/bin/php test.php
```

Shell 变量

变量是一种存储和操作数据的基本方式。在Shell脚本中，你可以创建、赋值、修改和引用变量。

只能使用英文字母、不能以数字开头、中间不能有空格、可以使用下划线、不能使用标点符号、避免使用Shell关键字。

1、创建变量和赋值

```Bash
name="John"
age=25
```

2、引用变量

要引用变量的值，可以使用$符号。

```Bash
echo $name
echo "My name is $name"
```

3、修改变量的值

可以使用赋值语句来修改变量的值。

```Bash
age=30
```

4、只读变量

你可以使用 **readonly** 命令将变量设置为只读，这意味着你不能修改它的值。

```Bash
readonly age
```

5、删除变量

使用 **unset** 命令可以删除一个变量。

```bash
unset age
```

6、环境变量

当谈到环境变量时，你可以将其想象成是一种全局变量，对于整个操作系统或进程而言都是可见的。环境变量存储了一些配置信息、路径和其他重要的数据，它们可以被不同的程序和脚本访问和使用。

环境变量是在整个Shell会话中都可用的特殊变量。你可以使用 **export** 命令将一个变量升级为环境变量。

```bash
export MY_VARIABLE="Hello"
```

Shell 字符串操作

1、拼接字符串

使用连接操作符 . 可以将两个字符串拼接在一起。

```bash
greeting="Hello"
name="Alice"
message=$greeting" "$name
echo $message
```

或者使用变量引用的方式。

```bash
message="${greeting} ${name}"
```

2、字符串长度

使用 **${#string}** 可以获取字符串的长度。

```Bash
text="Hello, World!"
length=${#text}
echo "字符串长度为：$length"
```

3、字符串截取

使用 **${text:7:5}** 将会从第7个字符开始（从0开始计数），截取5个字符，得到的子字符串是 **"World"**。

```Bash
text="Hello, World!"
substring=${text:7:5}
echo "截取的子字符串：$substring"
```

Shell 数组

1、创建数组

可以使用括号来创建数组，并在括号内用空格分隔数组元素。

```Bash
fruits=("apple" "banana" "cherry")
```

2、访问数组元素

通过索引来访问数组元素，索引从0开始计数。

```Bash
echo ${fruits[0]}  # 输出：apple
echo ${fruits[1]}  # 输出：banana
echo ${fruits[2]}  # 输出：cherry
```

shell 传递参数

1、位置参数

```bash
# 脚本名：myscript.sh
echo "第一个参数是：$1"
echo "第二个参数是：$2"
```

执行

```bash
./myscript.sh arg1 arg2
```

输出

```bash
第一个参数是：arg1
第二个参数是：arg2
```

2、特殊变量

除了位置参数，还有一些特殊变量用于获取有关脚本自身和其环境的信息，**$0**：脚本名称、**$#**：传递给脚本的参数个数、**$@**：所有参数的列表、**$***：所有参数的列表，作为单个字符串、**$?**：上一个命令的退出状态。

```bash
# 脚本名：special.sh
echo "脚本名：$0"
echo "参数个数：$#"
echo "参数列表：$@"
echo "参数列表（作为单个字符串）：$*"
echo "上一个命令的退出状态：$?"
```

执行

```bash
./special.sh arg1 arg2 arg3
```

输出

```
脚本名：./special.sh
参数个数：3
参数列表：arg1 arg2 arg3
参数列表（作为单个字符串）：arg1 arg2 arg3
上一个命令的退出状态：0
```

Shell echo命令

1、打印文本消息

```bash
echo "Hello, World!"
```

2、显示变量值

```bash
name="Alice"
echo "My name is $name"
```

3、打印多行文本

```bash
echo "Line 1"
echo "Line 2"
```

4、输出特殊字符

\t：代表制表符（Tab键）、\n：代表换行符；

```bash
echo "New\t line\n"
```

5、输出到文件

这将把 "Hello, File!" 输出到名为 **output.txt** 的文件中，如果文件存在则覆盖内容。

```bash
echo "Hello, File!" > output.txt
```

6、追加到文件

这将把 "More content" 追加到 **output.txt** 文件末尾。

```bash
echo "More content" >> output.txt
```