Kaldi 学习基础篇（二）--Shell 学习基础

原创：静默 Kaldi 学习 昨天

本节主要是介绍 Shell 相关的内容，如果读者接触过或者熟悉 Shell 这部分内容，请主动忽略该节。本节并不是全面的介绍 Shell 章节，而是简单介绍 Kaldi 使用到的大部分 Shell 内容。如果读者希望详尽的学习 Shell知识，请主动寻找相关专业介绍 Shell 的书籍、视频学习。

变量

绝大多数语言都存在变量一说，当然 Shell 也不例外。在 Shell 脚本中，变量的定义很简单，如果读者接触过类似 Python 这类语言，那么理解起来会非常简单。

Shell 变量量的定义如下:

```
变量量名=命名
your_name='kaldi'
```

注意: 虽然变量的命名与其他语言很类似，但也有它独特的地方。

- 变量名与等号之间不能有空格。
- 变量的命名只能使用英文字母、数字和下划线，并且首字母不能以数字开头
- 不能使用 bash 中的关键字作为变量名称

如果读者接触过 Python 这类高级语言，那么在变量调用的时候，只需要使用变量名即可，当然 Shell 在这一点上与这类高级语言有差异。在 Shell 中调用变量需要在该变量前加上符号: $ ，例如, 我们输出上文变量 your_name 的内容:

```
echo $your_name
echo ${your_name}
```

注意: 大家可能看到有两种方方式，那么在真实情况下，那种方式更好呢?其实这两种方式都是正确的，但是为了让编译器能够更好的识别变量的边界，这里推荐使用第二二种方式。

Shell 中的变量与 Python、C++、Java 等这类高级语言一样，都存在全局变量、局部变量。因此读者在编写相关 Shell 脚本时需要格外注意。

控制语句

与 Python 等高高级语言一一样，Shell 也存在一定的控制语句。只不过他们的使用格式有一定的区别，但是功能是一致的，因此，如果读者之前接触过高级语言，那么，这里只需要粗略略看一下控制语句的编写格式即可。

if 语句

if 语句流程控制格式如下:

```
if [ 条件 ] ; then
  执行内容
fi
```

if else 语句的流程控制格式如下:

```
if [ 条件 ] ; then
  执行内容
else
  执行内容
fi
```

if elif else 语句句流程控制格式如下:

```
if [ 条件 ] ; then
  执行内容
elif[ 条件 ] ; then
  执行内容
else
  执行内容
fi
```

注意: 对于if条件原理跟其他高级语言的原理是一致的，但是这里有几点需要特别注意:

- 只要 if 出现，那么必须带有 fi ，刚开始写 Shell 的小伙伴经常会忽略略掉最后的 fi
- 在if 的条件判断中，需要特别注意在中括号 [] 和条件之间需要有空格间隔。

for 循环控制

与其他语言类似，Shell 同样支持 for 循环控制。循环控制格式如下:

```
for var in item1 item2 ... itemN ;
do
  执行内容
done
```

注意: 对于 for 循环来说，可能存在多种方式，这里只是列举出最常用的一种，同时对于刚接触 Shell 的小伙伴需要特别注意，for 循环体中的do 和 done 。

while 循环控制

同样，对于 while 循环语句与其他语言中的 while 循环也是类似的。格式如下:

```
while 条件
do
  执行内容
done
```

注意点跟 for 循环类似，这里不在重复。

多分支支条件判断

与 python、c++等语言类似，Shell 中也存在多分支条件判断语句 case。case 语句的格式如下:

```
case 值 in
模式1)
    执行内容
    ;;
模式2)
    执行内容
    ;;
esac
```

注意: Shell 中的 case 语句与 python、C++ 中的意义是一致的，但是格式上有些差异

- 尾部需要 esac结束
- 每个条件结束处需要双冒号 ;;

跳出语句

break 在于跳出所有循环，continue 在于跳出当前循环。这两个语句与其他语言的意义一致。所以这里不多做解释。

判断条件

上文已经粗略的介绍了变量、控制语句等内容， 相信读者已经可以编写简单的脚本。本节主要针对判断介绍 一些对应的判断符号比大于、小于等。

正如读者看到的，上一节介绍了流程控制，在流程控制中都会存在条件这个名词，而在 Shell 中如何进行变量之间相似性对比等条件呢？这里会做介绍。

在 Shell 中既支支持类似于 Python 中的=，==，<, > 等符号，也支持类似于 -eq, -ne, -lt 等判断符号。 符号意义如下:

```
-eq 等于
-ne 不等于
-gt 大于
-lt 小于
-ge 大于等于
-le 小于等于
```

注意: 可能读者会产生这样的困惑，既然两种方式都支持，是不是说我们可以随便便使用任何一种都可以呢？答案是可以的.