PCS110　逻辑分支

概述

这里讲的逻辑分支主要是指 if ... else ...语句，if ... else ...也是比较重要的流程控制语句之一，使用比较简单。基本语法和其他语言的 if 语句很类似，因为 Python 中没有提供 switch 语句和三元运算符，但可以通过 if 语句来实现。

应用

最基本的用法

以下一段代码使用 if 分支来实现比较两个数大小的功能：

```
 # -*- coding: utf-8 -*-

a = int(raw_input("a = "))          #从键盘输入一个 int 类型的变量
b = int(raw_input("b = "))
if a == b:
   print 'a 等于 b'
elif a < b:
   print 'a 小于 b'
 elif a > b:
   print 'a 大于 b'
else:
   print '在本宇宙中这是不可能滴！'
```

elif 就是 else if 的缩写。

使用 if 代替 switch

Python 没有 switch 语句，不过 if elif else 组合已经足以应付大部分情况。而对于另外的一种 switch 应用场景，Python 还有更加优雅的方式来处理：

```
# 伪码 根据输入的不同参数选择程序的不同行为
switch( sys.argv[1] ):
   case '-e':
      walk_cd(); break;
   case '-d':
      search_cd(); break;
   ...
   default:
      raise CommandException("Unknown Commend: " + sys.argv[1])

# 使用 if 替代
if sys.argv[1]=='-e':
   walk_cd()
elif sys.argv[1]=='-d':
   search_cd()
...
else:
   raise CommandException("Unknown Command: " + sys.argv[1])

# 更好的做法
commands = {
   '-e'   : walk_cd,
   '-d'   : search_cd,
}
try:
   commands[ sys.argv[1] ]()
except KeyError:
   raise CommandException("Unknown Command: " + sys.argv[1])
```

最后一种方式不管从可读性（这是显然的）、性能（哈希表 vs 普通查找）上都高出很多。另外最后一种做法将参数与行为的映射完全独立出来了，一来修改起来极其方便，到时候也很容易将它们分离到配置文件中去。

三元运算符

三元运算符类似于 C 语言中的条件表达式，在 Python 中可以用下述代码实现：

```
>>> def get_length(s):
...     # 等价于:
...     # if(s!=None)
...     #     return len(s)
...     # else:
...     #     return len('None')
...     return len(s) if s!=None else len('None')
...
>>> get_length(None)
4
```

使用 and 和 or 技巧可以将上述代码写成：

```
>>> def get_length(s):
...     return s!=None and len(s) or len('None')
...
>>> get_length(None)
4
```

不过，第一种写法看起来更易读些。

小结

if ... else ...语句是经常用的语法。上面只介绍了其最基本的使用方法，在下面的链接中可以找到更多的资料。

- 过程控制：*http://wiki.woodpecker.org.cn/moin/ObpLovelyPython/LpyQLearn-3-process*
  精巧地址：*http://bit.ly/3RsfJm*
- Python 绝对简明手册：*http://wiki.woodpecker.org.cn/moin/PyAbsolutelyZipManual*
  精巧地址：*http://bit.ly/2EEz6I*