**第六章 分支循环 条件与枚举**

6-1 什么是表达式

表达式(expression) 是运算符(operator) 和操作数(operand)所构成的序列。

序列 是多个元素构成的且有顺序

1+2\*3 要按照顺序计算 按照优先级计算

a=1+2\*3 也是表达式

a=1 b=2 c=3

a+b\*c ===> ? 先乘除后加减 1+6=7

a or b and c ===> ?

b and c ===> 2 and 3 = True 2 and 3 = 3

1 or True ===> True 1 or 3 = 1

6-2 表达式的优先级 6-3 优先级练习 答案

优先级高

| \*\* 指数(次幂)运算

| ~ + - 补码，一元加减(最后两个的方法名称是 +@ 和 -@ )

| \* / % // 乘法，除法，模数和地板除 整除

| + -

| >> << 向右和向左移动

| & 按位与

| ^ | 按位异或和常规的 "OR"

| <= < > >= 比较运算符

| <> == != 等于运算符

| = %= /= //= -= += \*= \*\*= 赋值运算符

| is is not 身份运算符

| in not in 成员运算符

| not or and 逻辑运算符 但是在同一级别的逻辑运算符中 not > and > or

优先级低

同一级别的优先级 IDLE默认从左到右计算 逻辑运算符除外

() 小括号中的优先计算 优先级最高

(3+1)\*2 ==> 8

如果出现2个括号则按照从左到右的规则

a=1 b=2 c=3 (a or b) and (c + 1) ====>4

从左到右的规则 专业名称叫做 左结合 赋值除外 赋值表达式为右结合 c = a or b 先执行 a or b 再赋值

a = 1 b = 2 c = 2

not a or b + 2 == c

上面的表达式因为没有赋值 = 所以是左结合

1. b+2=2+2=4

2. 4==c ====> False

3. not a ====> False

4. False or False ====> False

6-4 在文本文件中编写Python代码

IDLE中 New File(ctrl+n) ====> 写代码 ====> 保存 (Helloworld.py)

Mac terminal

windows shell，powershell

执行 helloworld.py ====> python helloworld.py (python helloworld.py)

Python 代码的文件要保存以.py结尾

开发工具: Pycharm VSCode Sublime

IDE Integrated Development Environment

6-5 熟悉 VScode 开发环境与Python插件安装

插件网页商店: https://marketplace.visualstudio.com/

插件 Python vim VScode-iCon view in Brower terminal

记得保存成.py 结尾的文件 terminal快捷键 ctrl+~

6-6 6-7 流程控制语句之条件控制 一，二

Python是根据缩进来区分 代码和代码块的

注释：

单行注释 # vscode 快捷键 ctrl+/ mac 里是 command+/

多行注释 ''' '''注释内容''' ''' 只是一个三引号 但是是成对出现的 vscode 快捷键 alt+shift+a

# 条件控制 能解决 现实生活中选择性问题

if 条件控制

mood = True

if mood:

print("go left")

print("go right")

结果 是两个print都会被打印出来

mood = True

if mood:

print("go left")

else:

print("go right")

这样 就只会打印 go left 同样的代码 如果设定 mood = False 打印结果就会是 go right

if后面不一定是一个变量 还可以是一个表达式

a=1

b=2

if a>b:

print("a>b")

else:

print('a<b')

================================================================

接下来制作一个login function 如果用户输入正确 返回 successful 如果用户输入错误 返回 Fail

account = 'ZhanyuanChen'

password = '123456' #设定 用户名和密码

print('Enter your username:') #要求用户输入用户名

user\_account = input() #input()获取用户输入的用户名 并且 把用户输入的用户名存到一个用户名变量中

print('Enter your Password:') #以同样的方式获得用户输入的密码

user\_password = input()

# if user\_account == account and user\_password == password: #当用户输入的用户名和密码与我们先前设定的用户名和密码一样的时候

# print("Successful") #打印成功

# else:

# print("Failed") #否则打印失败

#这种方法万一用户输入错误了 不能确定用户输入的那个数据错误了

if user\_account == account: #当使用多个if 条件语句嵌套的时候就可以确定用户输入的那个数据出错了

if user\_password == password:

print("successful")

print("you have entered correct username and password")

else:

print("Password Error")

else:

print("Username Error")

print("you entered user name is: " + user\_account) #最后打印出用户输入的用户名和密码 告诉用户 你输入的是这个 错了 或者 对了

print("you entered password is: " + user\_password)

6-8 常量与Pylint的规范

Python 中没有常量 所以常量就把所有的字母都写成大写

modual 模块 一个文件就是一个模块

一个正常的Python项目是由多个模块组成的

Python 建议每一个模块都要有一段文字来描述这个模块的意义和作用 中英文都可以 copyright 也写进去。

Pylint 是一个检查Python语法的

Pylint 认为 如果你的变量不在函数或者类中则都认为是常量

if 循环语句的冒号: 前面不能加空格

程序的结尾 要有一行空行

比较运算符前面要加空格 但是不只有比较运算符前需要加空格吧

VS code 中 一个 tab 等于四个空格 这个用户可以自己进行更改

6-9 流程控制语句之条件控制 三 snippet 嵌套分支 代码块的概念

IDE 中的snippet snippet意思是 片段

tab 到下一个可编写的代码行。

回到上一个代码行 shift+tab

if 和 else 不用一定要合在一起使用 if 可以单独使用 但是 else 必须与if配对使用

if condition:

pass

else:

pass

pass 是空语句 也叫占位语句 它的作用是保持代码结构的完整性

在 API接口 里面 Pass 会经常出现的

if else 可以嵌套使用 但是不可过多的嵌套 阅读感会很差 如果需要有多个if else 或者过多的代码的时候可以把这些代码写成一个函数

==========一个结构体==========

if condition:

if condition:

pass

else:

pass # pass 中的代码被称为一个代码块

else: # 同一个代码块 一行被执行 则全部被执行

pass

=============================

认清结构体非常重要 可以判断变量的作用域

=============================

if condition: # 这就是一个完成的结构体

pass

=============================

Code 1 code 1,2,5 为一个代码块

Code 2

Code 3 code 3,4 为另外一个代码块

Code 4

Code 5

这个怎么判断是不是一个代码块呢？

根据缩进来判断是否为一个代码块 缩进一样则为一个代码块

封装 就是把这些代码装进 Function 或者 modual 中。

6-10 流程控制语句之条件控制 四 elif的优点

问题：

a=x 当a=1 ----> 吃苹果

当a=2 ----> 吃orange

当a=3 ----> 吃banana

当a=其他 ----> go shopping

如何表达这个条件？

并且 这种方法是错的 因为无论你输入什么都会返回 go shopping

原因是在 terminal中输入的是String 而循环中对比的是int 用string和int 对比一定会一直返回False

我们可以进行验证一下 在 a=input() 下面一行 加一行代码看看输入的类型是什么

print(type(a)) =========> <class 'str'>

返回结果是 str 这是因为 input() 函数 默认把所有用户输入的都认为是 str

笨方法：

a=input()

if a==1:

print('Eat Apple.')

else:

if a==2:

print('Eat Orange.')

else:

if a==3:

print('Eat banana')

else:

print('go shopping')

所以我们先要把 str 转换成 int 然后在进行比较 所以代码完整版如下:

a=input()

a=int(a)

if a==1:

print('Eat Apple.')

else:

if a==2:

print('Eat Orange.')

else:

if a==3:

print('Eat banana')

else:

print('go shopping')

这个代码使用了三个if else 语句进行层层嵌套 非常的难以阅读 所以不推荐！！

一般一层嵌套就可以了 再多的就不要用了

a=input()

a=int(a)

if a==1:

print("Eat Apple")

elif a==2:

print("Eat Orange")

elif a==3:

print("Eat Banana")

else:

print("Go Shopping")

elif 就是 else if 不可以单独使用 要与 if 一起使用

elif 简化代码级别 减少代码行数

Python中没有Switch 语句但是其他的语言都有 会比 elif 更好 但是Python里面没有也没办法啊

所以官方推荐使用 if....elif....elif....else 来代替 switch语句

或者

使用dict去代替switch 这种办法更好。

6-11 思考题解答与改变定势思维

a=1 b=0 #两个变量一个为True 一个为False 但是要返回为True的值

if else 可行 但是 我们也可以用 or

===============================================================

a=1

b=0

if a==1:

print("a is True")

else:

print("a is False")

===============================================================

print(a or b)

这会返回 1 ， 一真一假 使用 or 运算符来计算 则会返回为真的元素

然后判断哪个value 等于1 就返回哪个variable 就可以了