

Day03 - 分支结构

分支结构的应用场景

迄今为止，我们写的Python代码都是一条一条语句顺序执行，这种结构的代码我们称之为顺序结构。然而仅有顺序结构并不能解决所有的问题，比如我们设计一个游戏，游戏第一关的通关条件是玩家获得1000分，那么在完成本局游戏后我们要根据玩家得到分数来决定究竟是进入第二关还是告诉玩家“Game Over”，这里就会产生两个分支，而且这两个分支只有一个会被执行，这就是程序中分支结构。类似的场景还有很多，给大家一分钟的时间，你应该可以想到至少5个以上这样的例子，赶紧试一试。

if语句的使用

在Python中，要构造分支结构可以使用if、elif和else关键字。所谓关键字就是有特殊含义的单词，像if和else就是专门用于构造分支结构的关键字，很显然你不能够使用它作为变量名（事实上，用作其他的标识符也是不可以）。下面的例子中演示了如何构造一个分支结构。

```
.....
```

用户身份验证

Version: 0.1

Author: 骆昊

Date: 2018-02-28

```
.....
```

```
username = input('请输入用户名: ')
password = input('请输入口令: ')
# 如果希望输入口令时 终端中没有回显 可以使用getpass模块的getpass函数
# import getpass
# password = getpass.getpass('请输入口令: ')
if username == 'admin' and password == '123456':
    print('身份验证成功!')
else:
    print('身份验证失败!')
```

唯一需要说明的是和C/C++、Java等语言不同，Python中没有用花括号来构造代码块而是使用了缩进的方式来设置代码的层次结构，如果if条件成立的情况下需要执行多条语句，只要保持多条语句具有相同的缩进就可以了，换句话说连续的代码如果又保持了相同的缩进那么它们属于同一个代码块，相当于是一个执行的整体。

当然如果要构造出更多的分支，可以使用if...elif...else...结构，例如下面的分段函数求值。

$$f(x) = \begin{cases} 3x - 5 & (x > 1) \\ x + 2 & (-1 \leq x \leq 1) \\ 5x + 3 & (x < -1) \end{cases}$$

```
.....
```

分段函数求值

$$f(x) = \begin{cases} 3x - 5 & (x > 1) \\ x + 2 & (-1 \leq x \leq 1) \\ 5x + 3 & (x < -1) \end{cases}$$

Version: 0.1

Author: 骆昊

Date: 2018-02-28

```
.....
```

```
x = float(input('x = '))
if x > 1:
    y = 3 * x - 5
elif x >= -1:
    y = x + 2
else:
    y = 5 * x + 3
print('f(%.2f) = %.2f' % (x, y))
```

当然根据实际开发的需要，分支结构是可以嵌套的，例如判断是否通关以后还要根据你获得的宝物或者道具的数量对你的表现给出等级（比如点亮两颗或三颗星星），那么我们就需要在if的内部构造出一个新的分支结构，同理elif和else中也可以再构造新的分支，我们称之为嵌套的分支结构，也就是说上面的代码也可以写成下面的样子。

```
.....
```

分段函数求值

$$f(x) = \begin{cases} 3x - 5 & (x > 1) \\ x + 2 & (-1 \leq x \leq 1) \\ 5x + 3 & (x < -1) \end{cases}$$

Version: 0.1

Author: 骆昊

Date: 2018-02-28

```

"""

x = float(input('x = '))
if x > 1:
    y = 3 * x - 5
else:
    if x >= -1:
        y = x + 2
    else:
        y = 5 * x + 3
print('f(%.2f) = %.2f' % (x, y))

```

说明：大家可以自己感受一下这两种写法到底是哪一种更好。在之前我们提到的Python之禅中有这么一句话“Flat is better than nested.”，之所以提出这个观点是因为嵌套结构的嵌套层次多了之后会严重的影响代码的可读性，如果可以使用扁平化的结构就不要去用嵌套，因此之前的写法是更好的做法。

练习

练习1：英制单位与公制单位互换

```

"""

英制单位英寸和公制单位厘米互换

Version: 0.1
Author: 骆昊
Date: 2018-02-28

"""

value = float(input('请输入长度: '))
unit = input('请输入单位: ')
if unit == 'in' or unit == '英寸':
    print('%f英寸 = %f厘米' % (value, value * 2.54))
elif unit == 'cm' or unit == '厘米':
    print('%f厘米 = %f英寸' % (value, value / 2.54))
else:
    print('请输入有效的单位')

```

练习2：掷骰子决定做什么

```

"""

```

掷骰子决定做什么事情

Version: 0.1

Author: 骆昊

Date: 2018-02-28

.....

```
from random import randint
```

```
face = randint(1, 6)
if face == 1:
    result = '唱首歌'
elif face == 2:
    result = '跳个舞'
elif face == 3:
    result = '学狗叫'
elif face == 4:
    result = '做俯卧撑'
elif face == 5:
    result = '念绕口令'
else:
    result = '讲冷笑话'
print(result)
```

说明：上面的代码中使用了random模块的randint函数生成指定范围的随机数来模拟掷骰子。

练习3：百分制成绩转等级制

.....

百分制成绩转等级制成绩

90分以上 --> A
80分~89分 --> B
70分~79分 --> C
60分~69分 --> D
60分以下 --> E

Version: 0.1

Author: 骆昊

Date: 2018-02-28

.....

```
score = float(input('请输入成绩: '))
```

```
if score >= 90:
    grade = 'A'
elif score >= 80:
    grade = 'B'
elif score >= 70:
    grade = 'C'
elif score >= 60:
    grade = 'D'
else:
    grade = 'E'
print('对应的等级是:', grade)
```

练习4：输入三条边长如果能构成三角形就计算周长和面积

```
.....

判断输入的边长能否构成三角形
如果能则计算出三角形的周长和面积

Version: 0.1
Author: 骆昊
Date: 2018-02-28

.....

import math

a = float(input('a = '))
b = float(input('b = '))
c = float(input('c = '))
if a + b > c and a + c > b and b + c > a:
    print('周长: %f' % (a + b + c))
    p = (a + b + c) / 2
    area = math.sqrt(p * (p - a) * (p - b) * (p - c))
    print('面积: %f' % (area))
else:
    print('不能构成三角形')
```

说明：上面的代码中使用了`math`模块的`sqrt`函数来计算平方根。用边长计算三角形面积的公式叫做海伦公式。

练习5：实现一个个人所得税计算器。

```
.....
```

输入月收入 and 五险一金计算个人所得税

Version: 0.1

Author: 骆昊

Date: 2018-02-28

.....

```
salary = float(input('本月收入: '))
insurance = float(input('五险一金: '))
diff = salary - insurance - 3500
if diff <= 0:
    rate = 0
    deduction = 0
elif diff < 1500:
    rate = 0.03
    deduction = 0
elif diff < 4500:
    rate = 0.1
    deduction = 105
elif diff < 9000:
    rate = 0.2
    deduction = 555
elif diff < 35000:
    rate = 0.25
    deduction = 1005
elif diff < 55000:
    rate = 0.3
    deduction = 2755
elif diff < 80000:
    rate = 0.35
    deduction = 5505
else:
    rate = 0.45
    deduction = 13505
tax = abs(diff * rate - deduction)
print('个人所得税: ￥%.2f元' % tax)
print('实际到手收入: ￥%.2f元' % (diff + 3500 - tax))
```

说明: 上面的代码中使用了Python内置的`abs()`函数取绝对值来处理-0的问题。