

第 3

章 选择结构

顺序结构是最简单的程序结构,语句按照从上到下的顺序依次执行。但在实际使用中,还经常需要根据给定的条件进行分析、比较和判断,并按判断后的不同情况进行不同的处理,这属于程序设计中的选择结构。

Python 提供了多种形式的选择结构,有单分支结构、双分支结构和多分支结构,可以根据指定的条件选择执行不同的程序分支。

3.1 任务一 马拉松成绩

3.1.1 任务目标

全民健身,人人参与。专业选手小马和业余选手小白一起参加了北京马拉松,两位都顺利完赛。编写程序,从键盘输入两个人的成绩(小时和分钟),计算时间间隔并输出。

3.1.2 操作步骤

(1) 在 IDLE 中创建新文件,输入代码,如程序段 3-1 所示。

程序段 3-1

```
hour1,minute1=input()          #输入专业选手成绩
hour2,minute2=input()          #输入业余选手成绩
hour=hour2-hour1
minute=minute2-minute1
if minute<0:
    minute=minute+60
    hour=hour-1
print "时间间隔是{:d}小时{:d}分钟".format(hour,minute)
```

(2) 运行程序,结果如图 3-1 所示。

在 Shell 窗口中分别输入两个成绩,输出计算结果。

```
=====RESTART: D:\Python\p3-1.py=====
2,48
3,56
时间间隔是1小时8分钟
>>>
=====RESTART: D:\Python\p3-1.py=====
2,50
4,17
时间间隔是1小时27分钟
>>>
```

图 3-1 任务一运行结果

3.1.3 必备知识

3.1.3.1 单分支 if 语句格式和执行过程

单分支是最简单的选择结构,语法格式如下:

```
if 条件:
    语句块
```

语句执行时先对条件进行判断,当条件成立,也就是条件为 True 时,执行语句块。条件不成立时,直接执行 if 语句后面的语句。

语句块执行结束后,将接着执行 if 语句后面的语句。

3.1.3.2 单分支 if 语句使用说明

单分支 if 语句的条件可以是关系表达式或逻辑表达式,也可以是其他类型的数据。对于非布尔类型的数据,Python 设定 0 和空值(如空字符串、空列表)为 False,任何非 0 和非空值为 True。例如下面的 if 语句条件为 True:

```
n=31.6
if n%7:           #条件是非 0 值(n%7 结果为 3.6)
```

而下面两个 if 语句条件则为 False:

```
if 0:             #条件是 0 值
```

和

```
s=""
if s:             #条件是空字符串
```

语句块是一条或多条要执行的 Python 语句。语句块必须要有缩进,如果是多条语句,则所有语句的缩进量必须相同。

条件后面的冒号不能省略,它表示接下来是满足条件后要执行的语句块。

3.1.3.3 Python 缩进规则

Python 语法的一个重要特色是使用缩进来表示语句块,而不是像其他语言一样使用花括号。Python 对于代码缩进有严格的规定,不同的缩进量表示代码的不同层次。

(1) 同一语句块的所有语句都要缩进,而且缩进量必须相同。

如果同一语句块的不同语句有不同的缩进量,那么在运行代码时会引发错误提示。例如程序段 3-1 的 if 语句部分写成如下格式:

```
if minute<0:
    minute=minute+60
    hour=hour-1
```

运行该程序时,会弹出如图 3-2 所示的窗口,提示两条语句的缩进量不相同。

(2) 不能随意使用缩进。

对于没有包含在选择、循环、函数、类等结构中的语句,不能使用缩进,否则会在运行时出现 unexpected indent 的语法错误提示。

(3) 缩进量没有统一规定。

Python 要求语句块必须缩进,但是对缩进量没有具体的规定。编写代码时一般使用空格或者 Tab 键实现缩进,可以是 n 个空格,也可以是 n 个 Tab 键的位置。

建议不同层次之间使用 4 个空格或 1 个 Tab 键位置作为缩进,但是不要混用。

IDLE 开发环境是以 4 个空格作为代码的默认缩进量。选择菜单栏中的 Options→Configure IDLE 命令,会弹出如图 3-3 所示的对话框,可以在其中对默认缩进量进行设置。

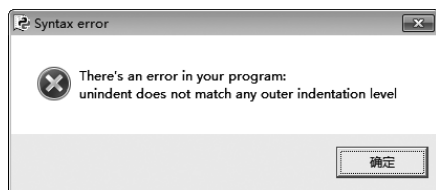


图 3-2 语法错误提示窗口

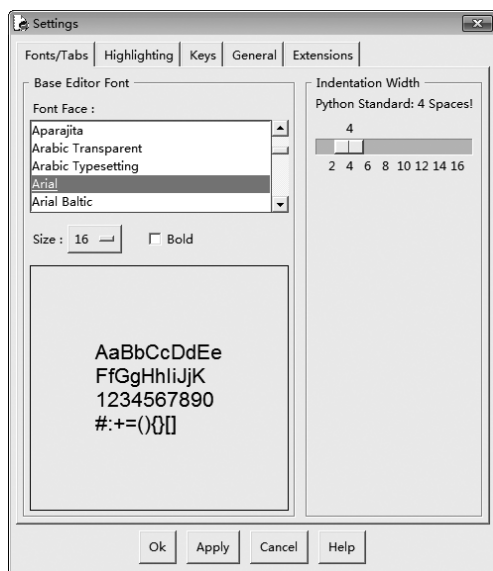


图 3-3 配置对话框

3.2 任务二 闰年

3.2.1 任务目标

编写程序,从键盘输入年份,判断该年是否为闰年。

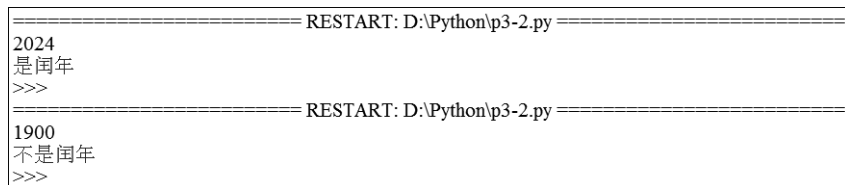
3.2.2 操作步骤

(1) 在 IDLE 中创建新文件,输入代码,如程序段 3-2 所示。

程序段 3-2

```
import calendar
y=input()
if calendar.isleap(y)==True:
    print "是闰年"
else:
    print "不是闰年"
```

(2) 运行程序,结果如图 3-4 所示。



```
=====RESTART: D:\Python\p3-2.py=====
2024
是闰年
>>>
=====RESTART: D:\Python\p3-2.py=====
1900
不是闰年
>>>
```

图 3-4 任务二运行结果

3.2.3 必备知识

3.2.3.1 双分支 if 语句格式和执行过程

双分支 if 语句的语法格式如下:

```
if 条件:
    语句块 1
else:
    语句块 2
```

语句执行时先对条件进行判断,当条件为 True 时,执行语句块 1,否则执行 else 后面的语句块 2。

语句块 1 或语句块 2 执行结束后,将接着执行 if 语句后面的语句。

3.2.3.2 双分支 if 语句使用说明

双分支 if 语句的条件和语句块的含义与单分支 if 语句相同。每次运行程序时,语句块 1 和语句块 2 之间只能执行一个。

条件和 else 后面的冒号都不能省略。else 子句不能单独出现,必须在 if 后面使用。

calendar 是 Python 中的日历模块,模块中包含与日历相关的一些函数。isleap(year)是其中之一,它的功能是判断闰年。如果 year 是闰年则函数值为 True,否则为 False。

使用 calendar 模块中的函数前,必须先导入模块,语句如下:

```
import calendar
```

有关 Python 函数使用的更多知识可参见本书后续章节的内容。

3.3 任务三 空气质量指数

3.3.1 任务目标

保护生态环境,从生活点滴开始,从加深了解入手。编写程序,根据从键盘输入的空气质量指数 AQI,输出相应的等级。

3.3.2 操作步骤

(1) 在 IDLE 中创建新文件,输入代码,如程序段 3-3 所示。

程序段 3-3

```
aqi=input()
if 0<=aqi<=50:
    print "优"
elif 51<=aqi<=100:
    print "良"
elif 101<=aqi<=150:
    print "轻度污染"
elif 151<=aqi<=200:
    print "中度污染"
elif 201<=aqi<=300:
```

```
print "重度污染"
elif aqi>300:
    print "严重污染"
```

(2) 运行程序,结果如图 3-5 所示。

```
=====RESTART: D:/Python/p3-3.py=====
96
良
>>>
=====RESTART: D:/Python/p3-3.py=====
151
中度污染
>>>
=====RESTART: D:/Python/p3-3.py=====
482
严重污染
>>>
=====RESTART: D:/Python/p3-3.py=====
50
优
>>>
```

图 3-5 任务三运行结果

3.3.3 必备知识

3.3.3.1 多分支 if 语句格式和执行过程

多分支 if 语句的语法格式如下:

```
if 条件 1:
    语句块 1
elif 条件 2:
    语句块 2
:
elif 条件 n:
    语句块 n
[else:
    语句块 n+1]
```

语句执行时,先对条件 1 进行判断,如果值为 True,则执行语句块 1。如果值为 False,则对条件 2 进行判断。如果条件 2 为 True,则执行语句块 2。以此类推,当某个条件为 True 时,就执行下面对应的语句块。如果所有的条件都为 False,而且语句有 else 项,则执行语句块 n+1。若没有 else 项,则直接执行 if 语句后面的语句。

任一语句块执行结束后,将接着执行 if 语句后面的语句。

3.3.3.2 多分支 if 语句使用说明

多分支 if 语句的条件和语句块的含义与单分支 if 语句相同。elif 是 else if 的缩写。

elif 和 else 子句都不能单独使用,必须和 if 一起出现,并且要正确配对。

一个多分支 if 语句中可以包含多个 elif 子句,但结尾只能有一个 else 子句。

多分支 if 语句执行时,不管有多少个分支,都只能执行一个分支,或者一个也不执行,不能同时执行多个分支。因此,即使语句中有多个条件为 True,也只执行第一个条件为 True 的分支,其他分支将不再执行。对于多分支语句,要注意条件书写的顺序。

3.4 任务四 出租车费用

3.4.1 任务目标

城市中不同类型的网约车起步价和计费分别为:1 类车起步价 14 元/3 千米,3 千米以外 1.5 元/千米;2 类车起步价 16 元/3 千米,3 千米以外 1.8 元/千米;3 类车起步价 20 元/3 千米,3 千米以外 2.5 元/千米。从键盘输入网约车的车型及距离,计算应付的车费并输出。

3.4.2 操作步骤

(1) 在 IDLE 中创建新文件,输入代码,如程序段 3-4 所示。

程序段 3-4

```
model=input("输入车型: ")
dist=input("输入距离: ")
if model==1:
    if dist<=3:
        cost=14
    else:
        cost=14+(dist-3)*1.5
elif model==2:
    if dist<=3:
        cost=16
    else:
        cost=16+(dist-3)*1.8
elif model==3:
    if dist<=3:
        cost=20
    else:
        cost=20+(dist-3)*2.5
print "车费是",cost
```

(2) 运行程序,结果如图 3-6 所示。

```
=====RESTART: D:\Python\p3-4.py=====
输入车型: 1
输入距离: 2
车费是 14
>>>
=====RESTART: D:\Python\p3-4.py=====
输入车型: 2
输入距离: 15.8
车费是 39.04
>>>
=====RESTART: D:\Python\p3-4.py=====
输入车型: 3
输入距离: 46
车费是 127.5
>>>
```

图 3-6 任务四运行结果

3.4.3 必备知识

3.4.3.1 if 语句的嵌套

if 语句的嵌套是在 if 语句的语句块中包含另一个 if 语句。例如,在单分支 if 语句中嵌套双分支 if 语句,形式如下:

```
if 条件 1:
    if 条件 2:
        语句块 1
    else:
        语句块 2
```

或者在双分支 if 语句中嵌套单分支 if 语句,形式如下:

```
if 条件 1:
    if 条件 2:
        语句块 1
else:
    语句块 2
```

在 Python 中,单分支 if 语句、双分支 if-else 语句和多分支 if-elif-else 语句之间可以相互嵌套。在相互嵌套时,一定要严格遵守不同层次语句块的缩进规则。

嵌套的 if 语句一般都可以用多分支 if 语句实现,因此,任务四的代码也可以如程序段 3-5 所示。

程序段 3-5

```
model=input("输入车型: ")
dist=input("输入距离: ")
if model==1 and dist<=3:
```

```

        cost=14
elif model==1 and dist>3:
    cost=14+(dist-3)*1.5
elif model==2 and dist<=3:
    cost=16
elif model==2 and dist>3:
    cost=16+(dist-3)*1.8
elif model==3 and dist<=3:
    cost=20
elif model==3 and dist>3:
    cost=20+(dist-3)*2.5
print "车费是",cost

```

3.4.3.2 pass 语句

编写代码是一个不断完善的过程,不是一蹴而就的事情。当有些代码的思路还不完整时,可能会出现空代码,但在运行程序时,Python 会对空代码部分给出错误提示。例如下面的程序:

```

x=input()
if x%9==0:
    #语句待补充
else:
    print x

```

运行时会给出如图 3-7 所示的错误提示。



图 3-7 空代码错误提示

而将程序改为如下形式则可以正确运行。

```

x=input()
if x%9==0:
    pass
else:
    print x

```

pass 是 Python 的空语句,解释器执行 pass 语句时,除了检查语法是否合法,其他什么也不做。对于选择、循环、函数定义、类定义等不允许出现空代码的结构,pass 语句可以避免出现错误。

3.5 小 结

本章主要知识点有：

- 单分支 if 语句；
- 双分支 if 语句；
- 多分支 if 语句；
- if 语句的嵌套；
- pass 语句。

3.6 动手写代码

1. 编写程序,判断输入的数字是否为 21 的倍数。
2. 编写程序,将输入的任意 3 个数按从大到小的顺序输出。
3. 编写程序,计算一元二次方程 $ax^2+bx+c=0$ 的实根。有以下几种可能:
 - (1) $a=0$,不是一元二次方程；
 - (2) $b^2-4ac=0$,有两个相等的实根；
 - (3) $b^2-4ac>0$,有两个不等的实根。
 - (4) $b^2-4ac<0$,无实根。
4. 编写程序,根据输入的 3 条边长值计算三角形面积。如果输入值不符合“任意两边之和大于第三边”,则给出错误提示。
5. 编写程序,计算个人所得税。起征点为 5000 元,超过 5000 元的部分按表 3-1 规则计算。

表 3-1 计算规则

应纳税所得额	税率	速算扣除数
不超过 3000 元	3%	0
超过 3000 元至 12000 元	10%	210
超过 12000 元至 25000 元	20%	1410
超过 25000 元至 35000 元	25%	2660
超过 35000 元至 55000 元	30%	4410
超过 55000 元至 80000 元	35%	7160
超过 80000 元	45%	15160

提示：如收入为 10000 元,则应纳税所得额为 $10000-5000=5000$ 元,按表 3-1 中第 2 行税率进行计算,所得税为 $5000\times 10\%-210=290$ 元。

6. 编写程序,将一年中的 12 个月分成 4 个季节输出。