

项目 3 健康数据分析



学习目标

知识目标：

1. 了解程序设计的基本结构。
2. 掌握 if、if...else 和 if...elif...else 语句的基本结构和用法。
3. 掌握 for 循环、while 循环语句的基本结构与用法。
4. 掌握循环中常用的 break、continue 和 pass 语句。

技能目标：

1. 能够熟练进行分支结构的程序设计。
2. 能够熟练进行循环结构的程序设计。

素质目标：

1. 培养学生分析问题的能力。
2. 培养学生编写程序解决问题的能力。
3. 严格遵守守法，增强社会责任感。

3.1 项目情景

某运动俱乐部要为每个成员制订一套合理的运动健身计划，在此之前，教练建议每人进行一次 BMI(body mass index, 身体质量指数测试)。了解成员 BMI 可以帮助教练设计出最适合成员当前体重和身体状况的训练方案。例如，如果一个人非常瘦弱，教练可能会推荐力量训练来增加肌肉量；而如果一个人超重，教练可能会推荐有氧运动来促进脂肪燃烧。现需要设计一个程序计算每个成员的 BMI，并给出分析结果。

BMI 指数即身体质量指数，是衡量人体肥胖程度的一个常用指标。项目经理列出需要完成任务清单，如表 3-1 所示。

表 3-1 项目 3 任务清单

任务序号	任务名称	知识储备
T3-1	健康数据分析	• 程序基本结构 • 分支语句 • 循环语句 • 其他常用语句

3.2 相关知识

3.2.1 分支语句

Python 程序中有顺序结构、分支结构(选择结构)和循环结构三种。顺序结构是指按语句出现的先后顺序执行的最简单的程序结构。分支结构是按照给定条件有选择地执行程序中的语句,分为单分支、双分支和多分支三种形式。在 Python 语言中一般使用 if 语句(单分支)、if...else 语句(双分支)和 if...elif...else 语句(多分支)实现分支结构。

1. if 语句

if 语句用于执行特定的代码块,当给定条件为真时执行。其语句格式如下:

```
if 判断条件:  
    代码块  
后续语句
```

单分支语句执行中先判断条件是否成立,若条件为 True,执行代码块,否则跳过代码块,继续执行后续语句。其流程图如图 3-1 所示。

注意: 判断条件后需加冒号,代码块应以缩进方式表达。Python 中的冒号和缩进可以帮助程序员区分代码之间的层次,理解语句间的逻辑关系,增强了程序的可读性。

例 3-1 判断成绩是否合格。

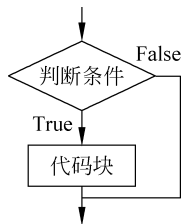


图 3-1 单分支语句流程图

```
# 获取用户成绩  
score = int(input("编写程序判断成绩是否合格,请输入你的得分:"))  
# 根据得分判断成绩是否合格  
if score >= 60:  
    print("成绩合格!")
```

执行结果如下:

```
编写程序判断成绩是否合格,请输入你的得分:88  
成绩合格
```

微课 3-1: 分支语句



2. if...else 语句

if...else 语句的语法格式如下：

```
if 判断条件:
    代码块 1
else:
    代码块 2
```

双分支语句执行中也是先判断条件是否成立，若判断条件为 True 时，执行代码块 1；反之，判断条件为 False 时，执行代码块 2。其流程图如图 3-2 所示。

例 3-2 判断是否成年。

```
# 获取用户年龄
age = int(input("请输入你的年龄:"))
# 根据年龄判断是否成年
if age >= 18:
    print("成年人")
else:
    print("未成年人")
```

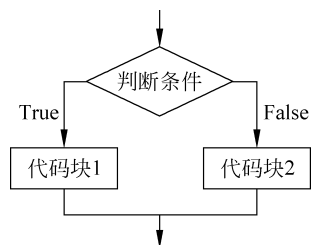


图 3-2 双分支语句流程图

执行结果 1：

```
请输入你的年龄:16
未成年人
```

执行结果 2：

```
请输入你的年龄:26
成年人
```

3. if...elif...else 语句

在 Python 中，if 和 elif 是用于控制程序流程的条件语句，它们允许根据条件的成立与否执行不同的代码块。

elif 是 else if 的缩写，用于在多个条件之间进行判断。若 if 语句中条件为 True，则执行代码块 1；若 if 语句中的条件为 False，便会执行 elif 语句；当所有条件都不满足时，执行 else 语句。

if...elif...else 语句的语法格式如下：

```
if 条件 1:
    代码块 1
elif 条件 2:
    代码块 2
...
elif 条件 n-1:
    代码块 n-1
else:
    代码块 n
```

注意：每个条件后有“:”，代码块 1、代码块 2 等都要对齐并向右边缩进。

多分支语句中，根据条件执行相应的代码块。即若条件 1 成立，执行代码块 1；若条件 1 不成立，条件 2 成立，执行代码块 2；以此类推，当所有条件都不成立，则执行 else 语句后的代码块 n 。其流程图如图 3-3 所示。

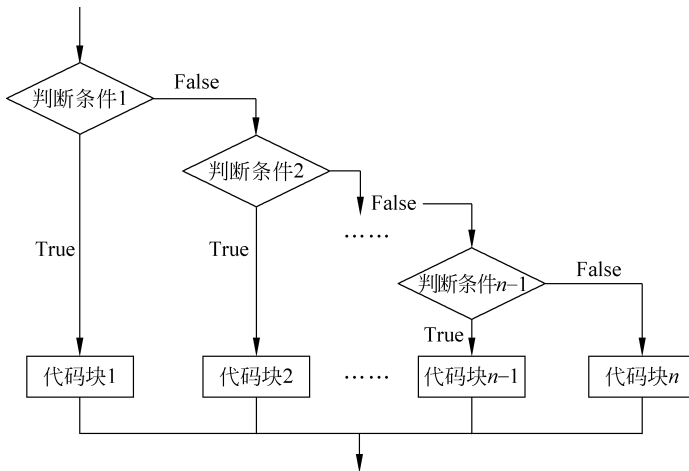


图 3-3 多分支语句流程图

例 3-3 遵守交通规则。

```

print("遵守交通规则,争做文明公民")
# 此处 1 表示红灯, 2 表示绿灯, 3 表示黄灯
a = int(input("请输入交通信号灯:"))
if a == 1:
    print("红灯停")
elif a == 2:
    print("绿灯行")
elif a == 3:
    print("遇到黄灯停一停")
else:
    print("输入错误,请重新输入!")
  
```

执行结果 1:

```

遵守交通规则,争做文明公民
请输入交通信号灯:1
红灯停
  
```

执行结果 2:

```

遵守交通规则,争做文明公民
请输入交通信号灯:2
绿灯行
  
```

执行结果 3:

```
遵守交通规则,争做文明公民
请输入交通信号灯:3
遇到黄灯停一停
```

职业素养提升

交通信号灯的紅、黃、綠灯光的颜色变化是根据一定的时间周期和交通流量通过程序进行控制的。交通信号灯转换的具体时间间隔会根据不同的交通状况和时间段进行调整。遵守交通规则能有效避免交通事故的发生,保护自身和他人的生命安全,同时有助于维护社会公共秩序。遵守交通规则不仅是基本的社会公德,更是法律规定,是每一位公民的职责和义务。我们应提高交通安全意识,增强法治观念,身体力行,自觉遵守交通规则,为构建和谐社会作出贡献。

3.2.2 循环语句

1. while 循环

在 Python 编程中,while 语句用于循环执行程序,即在某条件下,循环执行某段程序,以处理需要重复处理的相同任务。while 循环执行流程如图 3-4 所示。

while 语句的语法格式如下:

```
while 表达式:
    循环体
```

在执行 while 循环中,首先判断条件是否满足。如果条件为 True 时,那么执行循环体代码,然后判断条件是否满足。如此循环下去,直至条件不满足时停止循环。

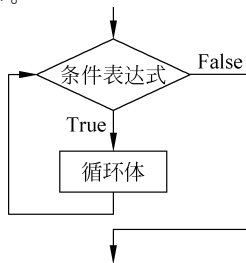


图 3-4 while 循环执行流程

例 3-4 坚持锻炼,增强体质。

```
print("第一天跳绳 200 个,每天增加 50 个")
i = 1
s = 200
print("坚持锻炼一个月")
while i < 31:
    s = s + 50
    i = i + 1
print("第", i-1, "天时,跳绳", s, "个")
```

执行结果如下:

```
第一天跳绳 200 个,每天增加 50 个
坚持锻炼一个月
第 30 天时,跳绳 1700 个
```



职业素养提升

“健康中国”是中国共产党提出的一个全国性的健康促进计划，旨在改善国民的健康状况和提升医疗卫生服务水平。这一计划与中国政府的“中国梦”理念相契合，强调预防为主，倡导健康生活方式，减少疾病发生，并通过优化医疗资源配置及推进医疗服务均等化等措施，提高全民健康水平。

作为一名程序设计人员，更要注意强身健体。选择适合的运动方式，制订合理的锻炼计划，持之以恒，坚持锻炼可以提高身体素质，增强免疫力，预防疾病，同时有助于保持良好的心理状态，减少压力，提高工作效率。良好的体质不仅对个人有益，也是为了更好地服务于社会和国家。通过保持健康的生活方式，我们可以更有效地发挥自己的潜力，为实现个人的梦想和国家的繁荣富强作出积极的贡献。

while 语句中可以使用 if 语句根据需要执行不同的代码块，两者组合使用，可以实现更复杂的程序逻辑。

例 3-5 计算 10 以内奇数的和。

```
print("计算 10 以内奇数的和")
s = 0
i = 1
while i < 11:
    if i % 2 == 1:
        s = s+i
    i = i+1
print("s=", s)
```

执行结果如下：

```
计算 10 以内奇数的和
s= 25
```

微课 3-2：循环语句



2. for 循环

在 Python 中，for 循环可以遍历任何序列，比如字符串或者后期要学的列表、元组等。其语句格式如下：

```
for 变量 in 序列 :
    循环语句
```

例 3-6 竖式输出“我爱你，中国”。

```
s = "我爱你, 中国"
for i in s:
    print(i)
```

执行结果如下：

```
我  
爱  
你  
,  
中  
国
```

for 循环也常常和 range() 函数搭配使用, range() 可以创建一个整数序列。其语法格式如下：

```
range(start, stop, step)
```

参数说明如下：

(1) start 是序列的起始值, 默认从 0 开始。如 range(3) 等同于 range(0, 3)。

(2) stop 是序列的结束值, 但不包括 stop。如 range(3) 指 [0, 1, 2)。

(3) step 是步长, 默认为 1。如 range(3) 等同于 range(0, 3, 1)。

例 3-7 打印星号三角。

```
print("打印星号三角")  
for i in range(1, 6):  
    print("*" * i)
```

执行结果如下：

```
打印星号三角  
*  
**  
***  
****  
*****
```

3.2.3 其他常用语句

在 while 循环中, 当条件表达式为 True 时, 程序会执行循环, 所以使用 while 循环时, 一定要确保循环条件最终会变为 False, 否则条件永远成立, 便会形成无限循环, 即死循环。为避免这种情况的发生, 可以用 break 语句或 continue 语句跳出循环。

1. break 语句

使用 break 语句会立即终止循环, 不再执行循环体中剩余代码。

例 3-8 验证用户输入的数字是否在指定范围内：在用户输入验证中, 当用户输入不符合要求时, 可以使用 break 语句跳出循环, 避免无限循环。

```
while True:  
    user_input = input("请输入一个 1 ~ 100 的数字:")  
    if user_input.isdigit():
```

```

number = int(user_input)
if 1 <= number <= 100:
    print("输入正确!")
    break
else:
    print("输入错误,请输入 1 ~ 100 的数字。")
else:
    print("输入错误,请输入数字。")

```

以上程序的执行效果如图 3-5 所示。

2. continue 语句

使用 continue 语句会跳出本次循环,继续执行下一次循环。如遇循环嵌套时,两者都是跳出最内层循环。

例 3-9 输出 10 以内的奇数。

```

print("输出 10 以内的奇数")
i = 1
for i in range(1, 11):
    if i % 2 == 0:
        continue
    else:
        print(i)

```

执行结果如下:

```

1
3
5
7
9

```

例 3-10 跳出当前循环,继续下一次循环。

```

print("欢迎光临**景区 预约门票")
while True:
    age = int(input("请输入你的年龄:"))
    # 根据年龄判断景区票价
    if age >= 60 or age < 6:
        print("60 岁以上老年人及 6 岁以下未成年人免门票,可以凭身份证直接入园!")
        continue
    elif 18 > age >= 6:
        print("6 至 18 岁未成年人,景区半价:57 元/人")
        break
    else:
        print("景区票价:115 元/人")
        break
print("请确认付款金额,扫码购票!")

```

```

请输入一个1 ~ 100 的数字: p
输入错误, 请输入数字。
请输入一个1 ~ 100 的数字: 111
输入错误, 请输入 1 ~ 100 的数字。
请输入一个1 ~ 100 的数字: 90
输入正确!

```

Process finished with exit code 0

图 3-5 应用 break 语句的效果

例 3-10 的执行结果如图 3-6 所示。

```
欢迎光临**景区 预约门票
请输入你的年龄: 65
60岁以上老年人及6岁以下未成年人免门票,可以凭身份证直接进入园!
请输入你的年龄: 19
景区票价: 115元/人
请确认付款金额,扫码购票!
```

图 3-6 应用 continue 语句的效果

3. pass 语句

Python 中的 pass 语句是一个占位语句,不做任何事,其作用在于保存程序结构的完整性。比如 while 语句中,pass 语句可以作为空的执行体,无限次循环,无任何实际操作,具体写法如下:

```
while True:
    pass
```

以上语句进入死循环,所有只限举例,现实中不建议使用这样。

微课 3-3: 其他常用语句



3.3 项目实现

该项目包含一个任务,任务序号是 T3-1,任务名称是健康数据分析。

3.3.1 分析与设计

1. 需求分析

BMI 指数的计算公式为: 体重(千克)/身高(米)的平方。例如,一个体重为 70 千克,身高为 1.75 米的人,其 BMI 值为 $70/(1.75 \times 1.75) = 22.86$ 。

根据世界卫生组织的标准,BMI 指数的分类如下。

偏瘦: $BMI < 18.5$ 。

正常: $18.5 \leq BMI < 24$ 。

超重: $24 \leq BMI < 28$ 。

肥胖: $BMI \geq 28$ 。

需要注意的是,BMI 指数并不能完全准确地反映一个人的身体脂肪含量和分布情况,因为肌肉发达的人可能会被误判为超重或肥胖。此外,对于不同种族、年龄和性别的人群,BMI 指数的标准也可能有所不同。因此,在评估个人健康状况时,还需要结合其他因素进行综合判断。

(1) 用户输入。输入体重并保存到 w 中,输入身高保存到 h 中,再将输入的数据转换成

数字型。

(2) 系统处理。根据公式计算 BMI 值,使用多分支语句并根据 BMI 值所在的范围进行判断,并输出分析结果。

(3) 系统循环。使用 while 循环并根据用户的输入需求,确定是否继续进行 BMI 分析。系统测试数据如下:

```
55,1.63
60,1.50
70,1.90
55,1.83
```

2. 流程设计

该项目流程图如图 3-7 所示。

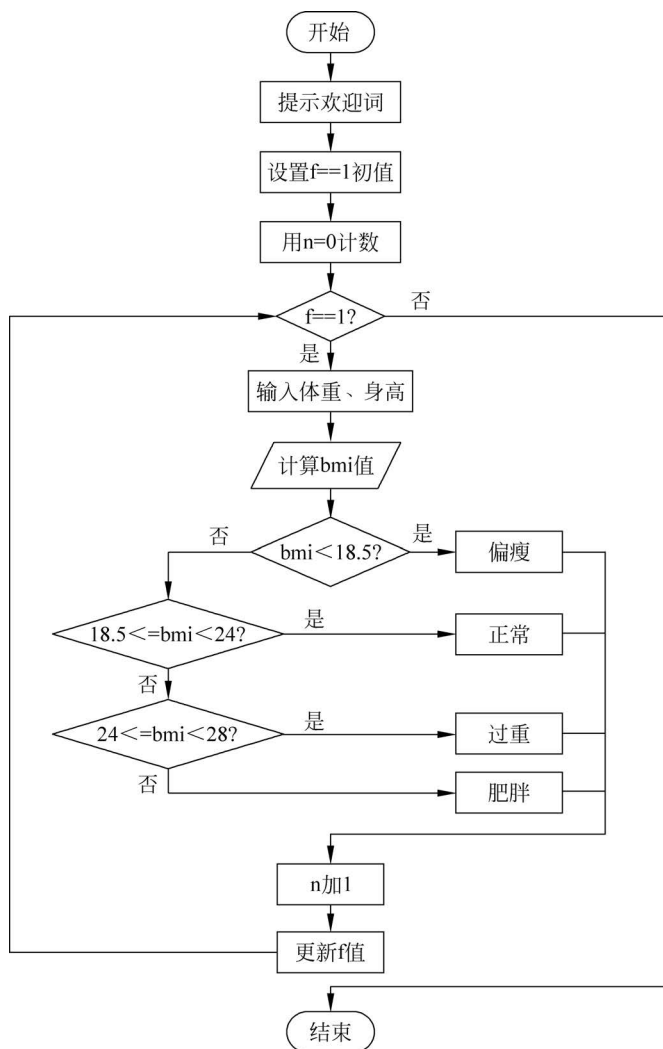


图 3-7 项目 3 流程图