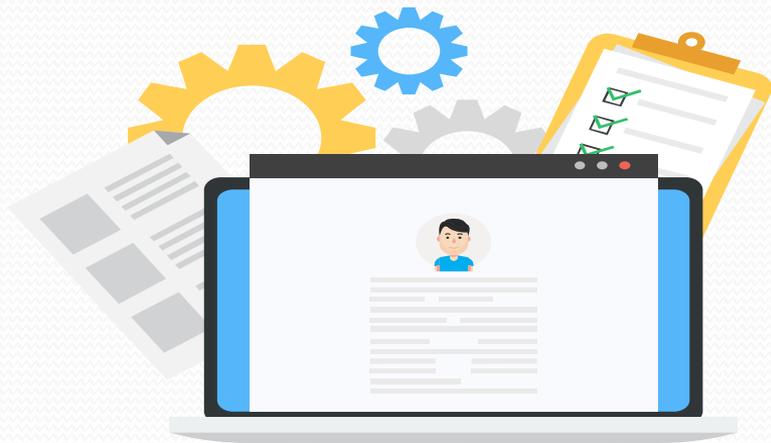


第三章 程序控制

Contents

- ▶ 顺序结构
- ▶ 分支结构
- ▶ 循环结构
- ▶ 排序实例
- ▶ 本章小结
- ▶ 习题





01

顺序结构

高级语言程序只有顺序、分支和循环三种程序结构，这些结构可以实现所有的计算机算法。作为一种高级语言，Python的程序控制具有极致精简的特点，每行代码为一条执行语句，处于同一缩进位置的所有语句按位置构成顺序结构；由if语句或match语句及其下一级缩进的语句构成分支结构；由while语句或for语言及其下一级缩进的语句构成循环结构。本节将详细介绍Python语言程序的控制结构。

3.1 顺序结构

顺序结构的程序的执行按照其语句的位置先后顺序执行。顺序结构是计算机程序的总体执行结构，只有在程序的部分功能模块中才可以采取分支结构或循环结构。

3.1 顺序结构

```
1  import math
2  x1=input('Please input x of Point A:')
3  y1=input('Please input y of Point A:')
4  x2=input('Please input x of Point B:')
5  y2=input('Please input y of Point B:')
6  x1=float(x1)
7  y1=float(y1)
```

3.1 顺序结构

```
8 x2=float(x2)
```

```
9 y2=float(y2)
```

```
10 d=math.sqrt((x2-x1)**2+(y2-y1)**2)
```

```
11 print(f'Distance of A({x1},{y1}) and B({x2},{y2}) is: {d}.')
```

第1行装载math模块，第10行使用了math模块中的sqrt函数。

第2行调用input函数读取键盘的输入字符串，作为点A的x坐标，第6行“x1=float(x1)”将x1转化为浮点数，仍使用标签x1。

3.1 顺序结构

第3行读取键盘的输入字符串，作为点A的y坐标。

第7行将y1转化为浮点数，仍使用标签y1。

第4行读取键盘的输入字符串，作为点B的x坐标

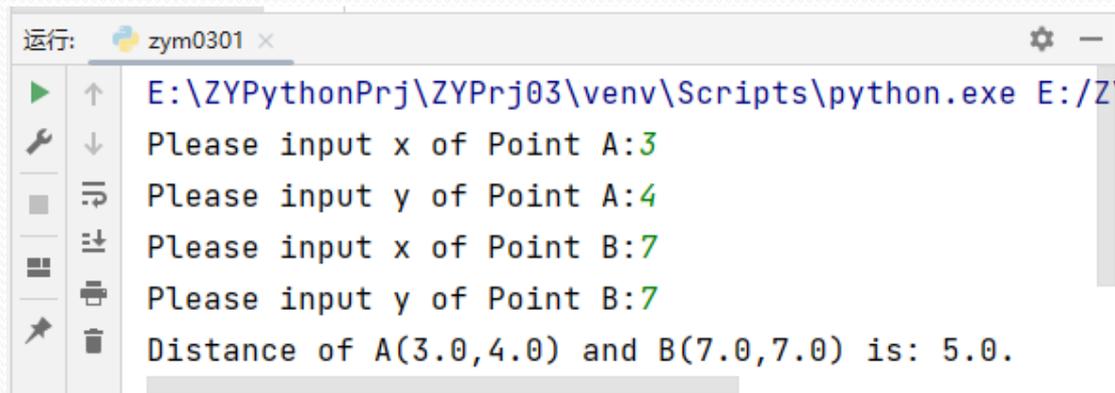
第8行将x2转化为浮点数，仍使用标签x2。

第5行读取键盘的输入字符串，作为点B的y坐标。

第9行将y2转化为浮点数，仍使用标签y2。

3.1 顺序结构

第10行调用math模块的sqrt函数计算两点间的距离。
第11行输出两个点A和B以及它们的欧氏距离d。



```
运行: zym0301 x
E:\ZYPythonPrj\ZYPrj03\venv\Scripts\python.exe E:/Z
Please input x of Point A:3
Please input y of Point A:4
Please input x of Point B:7
Please input y of Point B:7
Distance of A(3.0,4.0) and B(7.0,7.0) is: 5.0.
```

模块zym0301执行结果



02

分支结构

3.2 分支结构

程序的分支结构用于实现有条件的选择特定语句执行。
Python语言中使用缩进表示同一分支结构中的语句。例如：

if 条件表达式:

 语句1

 语句2

else:

 语句3

 语句4

3.2 分支结构

Python语言中，缩进也用于表示循环结构中的归属关系。

对缩进格式而言，一条语句（一般为控制语句）下的具有相同缩进量的所有语句均属于该语句的控制部分。只是为了程序代码美观，每级缩进均使用相同的缩进量。

分支结构也称选择结构，Python语言中，使用if或match关键字实现分支控制。

3.2.1 if语句

➤ if语句具有四种常用结构，如下所示：

(1) 单个if结构

if 条件表达式:

语句1

.....

语句n

上述结构为单个if结构，即当“条件表达式”为真时，执行“语句1”至“语句n”。

3.2.1 if语句

(2) 标准if-else结构

if 条件表达式:

语句1

.....

语句m

else:

语句m+1

.....

语句n

3.2.1 if语句

第二种结构为标准的if-else结构，即当“条件表达式”为真时，执行“语句₁”至“语句_m”；否则，执行“语句_{m+1}”至“语句_n”。如果if部分没有语句，用“pass”语句表示。

3.2.1 if语句

(3) if-elif-else结构

if 条件表达式1:

语句1

.....

语句k

elif 条件表达式2:

语句k+1

.....

语句m

3.2.1 if语句

elif 条件表达式3:

语句m+1

.....

语句p

..... #其他的elif语句

else:

语句t

.....

语句n

3.2.1 if语句

第三种结构是典型的多分支结构，即如果“条件表达式1”为真时，执行“语句1”至“语句k”；否则，如果“条件表达式2”为真时，执行“语句k+1”至“语句m”；否则，如果“条件表达式3”为真时，执行“语句m+1”至“语句p”；否则，当上述所有“条件表达式”均为假时，执行“else”部分的“语句t”至“语句n”。

3.2.1 if语句

(4) if语句的嵌套结构

if结构中的if部分、else部分和elif部分中均可再嵌入if结构，此时，注意各个部分的缩进关系，以保证if部分与其相应的elif或else部分相一致，不至于出现“张冠李戴”的问题。

3.2.1 if语句

➤ 例：输入一数值x，计算分段函数f(x)的值：

$$f(x) = \begin{cases} x + 1, & x > 1 \\ 3x - 1, & 0 < x \leq 1 \\ -x^2 - 1, & -1 < x \leq 0 \\ 2x^3, & x \leq -1 \end{cases}$$

3.2.1 if语句

```
1    x=input('Please input a number:')
2    x=float(x)
3    y=0
4    if x>1:
5        y=x+1
6    elif 0<x<=1:
7        y=3*x-1
```

3.2.1 if语句

```
8     elif -1<x<=0:  
9         y=-x**2-1  
10        else:  
11            y=2*x**3  
12        print(f'{x} produces {y:.2f}')
```

3.2.1 if语句

第1~12行为一个功能模块，求解上述的“问题”。第1行“`x=input('Please input a number:')`”从键盘输入一个表达数值的字符串给x。第2行“`x=float(x)`”将x转化为浮点数，仍用x作为标签。第3行“`y=0`”将0赋给y。

第4~11行为一个多分支结构，对应于式(1)的四种情况，注意在Python语言中，支持这种“ $0 < x \leq 1$ ”级联的关系不等式。第4行“`if x > 1:`”如果x大于1，则执行第43行“`y=x+1`”将x与1的和赋给y；

3.2.1 if语句

如果第6行“`elif 0<x<=1:`”中 x 大于0且小于等于1，则执行第7行“`y=3*x-1`”，将 $3x-1$ 的值赋给 y ；否则，若第8行“`elif -1<x<=0:`”中 x 大于-1且小于等于0，则执行第9行“`y=-x**2-1`”将 $-x^2-1$ 赋给 y ；在上述情况都不成立时，执行第11行“`y=2*x**3`”将 $2x^3$ 赋给 y 。

第12行“`print(f {x} produces {y:.2f})`”在屏幕上打印 x 和 y 的值。

3.2.1 if语句

if结构可以写成一条语句，形如：

y=表达式1 if 条件 else 表达式2

表示当“条件”为真时，将“表达式1”的值赋给y，否则将“表达式2”的值赋给y。“表达式2”还可以嵌入新的if结构。

3.2.2 match语句

match多分支控制语句是Python 3.10新添加的控制语句，其基本语法形式为：

match 表达式：

 case 表达式1：

 语句1

 语句k

3.2.2 match语句

case 表达式2:

语句k+1

.....

语句p

..... #其他的case情况

case _:

语句m

.....

语句n

3.2.2 match语句

上述match分支的功能非常强大，表现在这里的“表达式”可以为字符串、数值、逻辑值、列表、元组和字典等，每个case部分可以添加条件限制，用“if 表达式”表示。当match后面的“表达式”与某个case后面的表达式匹配后，则执行该case部分的语句。

3.2.2 match语句

➤ 例：当输入x为0或1或2时，输出“低分”；当输入3或4，输出“中分”；当输入5时，输出“高分”；当输入小于0或大于5的数时，提示输入有误

```
1  if __name__ == '__main__':  
2      x=int(input('Input a number:'))  
3      match x:  
4          case 0 | 1 | 2:  
5              print('Low mark.')6          case 3 | 4:  
7              print('Medium mark.')
```

3.2.2 match语句

```
8     case 5:  
9         print('High mark.')
```

10 case _ if x>5:
11 print('Input is too big.')

12 case _ if x<0:
13 print('Input is too small.')

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/538116066006006101>