

# 5.2 设计简单程序

---

高教版《信息技术》基础模块(下册)



# 复习回顾



## 网购电影票显示付款金额

$$\text{金额} = \text{单价} * \text{数量}$$

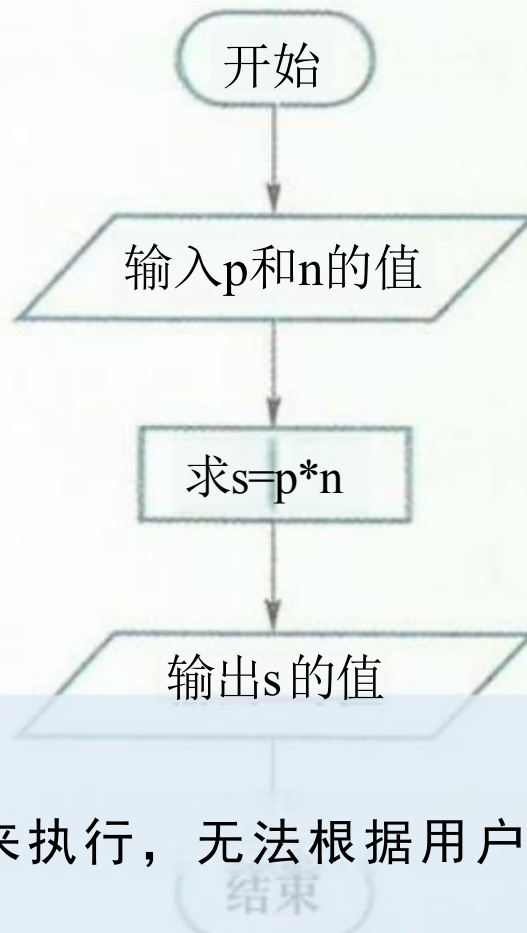
↓            ↓            ↓  
s            p            n

#输入单价和数量

```
p = float (input (请输入单价: ))  
n = int (input ('请输入数量: '))
```

```
s = n * p #计算金额
```

如果没有**流程控制语句**，整个程序将按照**线性顺序**来执行，无法根据用户的需求决定程序“怎么做”。



顺序结构

# CONTENT

---

1

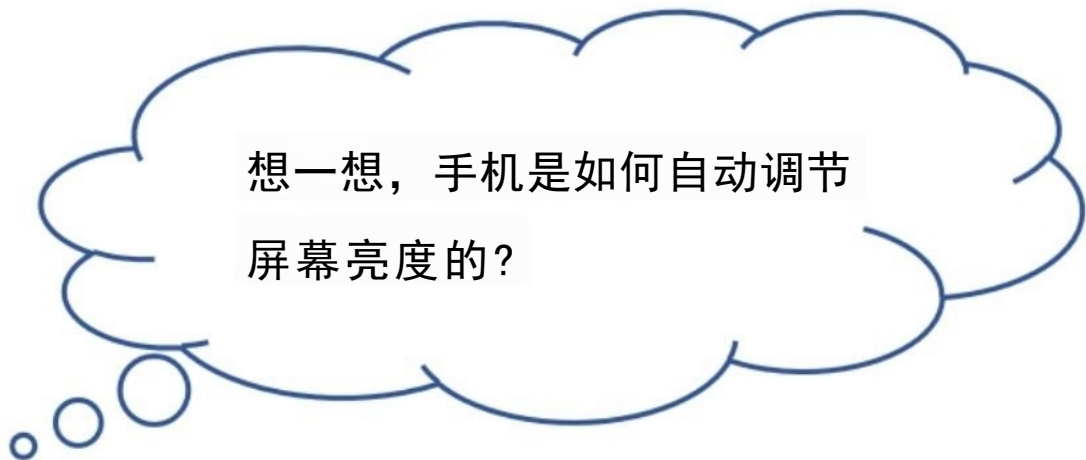
构

使用选择结

使用循环结构

使用函数

# 使用选择结构



程序设计时，需要使用关系运算符、逻辑运算符及选择结构来实现判断和选择。

# 使用选择结构

## 1 数据类型

Python 有6种标准数据类型：**数字、字符串、元组、列表、集合和字典**。其中，元组、列表、集合和字典可以保存多项数据，通常也称“数据结构”类型。

表 5-4 Python 部分数据类型

数据类型		数据类型符	说明	举例
数字	整数类型	int	存储整数数值	50
	浮点类型	float	存储小数数值	3.14、2.15e2 (e2表示 $10^2$ )
	复数类型	complex	存储复数	1+2j
	布尔类型	bool	存储逻辑值	True、False
字符串		str	存储字符串	“abc”、'12345'
列表		list	数据结构类型	[20, 10, 50, 30]

# 使用选择结构

如果想在字符串中包括一些特殊的字符，如换行符、制表符等，则需要使用转义符。

表5-5 Python转义符

字符表示	说明
\t	水平制表符
\n	换行
\r	回车
\"	双引号
\'	单引号
\\	反斜线

example.py-C:\Users\54781\Desktop\example.py (3.7.0)

Python 3.7.0 Shell

□ ×

File Edit Format Run Options

print 在Python

Window Help

print('c: Users Desktop')

File Edit Shell Debug Options Window Help

Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.19146  
4-bit(AMD64)] on win32

Type "copyright", "credits" or "license()" for more information.

>>>

=====  
=====RESTART:C:\Users\54781\Desktop\example.py

=====  
=====

在Python中，文件路径中的反斜线

需要用转义符来表示，如：

**C:\Users\Desktop**

>>>

# 使用选择结构

## 2 关系运算符

- 判断数据大小关系的运算符称为关系运算符，关系运算符有“>”（大于）、“>=”（大于等于）、“<”（小于）、“<=”（小于等于）、“==”（等于）、“!=”（不等于）。
- 由关系运算符连接而成的表达式称为关系表达式。关系表达式的值为True或False，对应于整数1或0直接参与其他运算。例如，“5>3”的值为True，“9<8”的值为False。

## 3 逻辑运算符

表5-6 Python逻辑运算符

运算符	优先级	含义	举例
not	1	逻辑非，将当前逻辑值取反	not逻辑表达式
and	2	逻辑与，前后表达式均为真时，结果为真	逻辑表达式1and逻辑表达式2
or	3	逻辑或，前后表达式只要有一个为真，结果为真	逻辑表达式1or逻辑表达式2

# 使用选择结构

## 4 选择语句

选择语句也称分支语句，Python的选择语句有3种。



if-else 语句

elif 语句

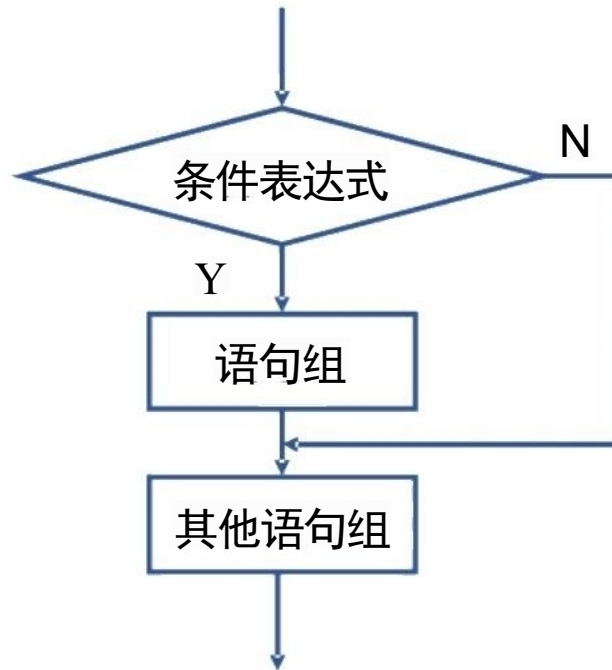
条件表达式

```
if <条件表达式>:  
    <语句组>
```

```
score = int(input('请输入成绩: '))  
if score >= 60:  
    print('成绩合格')
```

```
请输入成绩: 92  
成绩合格  
>>>
```

```
请输入成绩: 55  
>>>
```





# 使用选择结构

## 4 选择语句

选择语句也称分支语句，Python的选择语句有3种。

if 语句

if-else 语句

elif 语句

条件表达式

```
if <条件表达式>:
```

```
<语句组1>
```

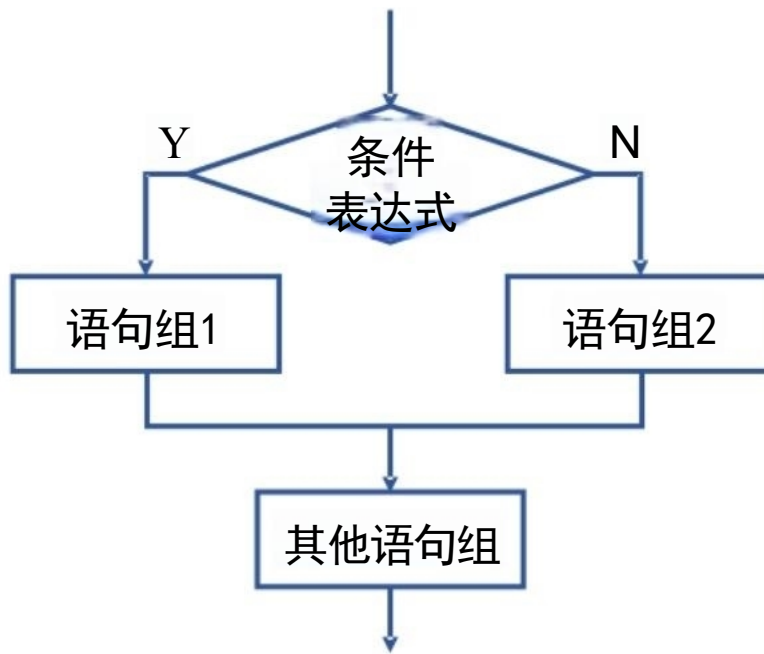
```
else: 否则
```

```
<语句组2>
```

```
score = int(input('请输入成绩: '))  
if score >= 60:  
    print('成绩合格')  
else:  
    print('成绩不合格')
```

```
请输入成绩: 92  
成绩合格  
>>>
```

```
请输入成绩: 55  
成绩不合格  
>>>
```



# 使用选择结构

## 4 选择语句

选择语句也称分支语句，Python的选择语句有3种。

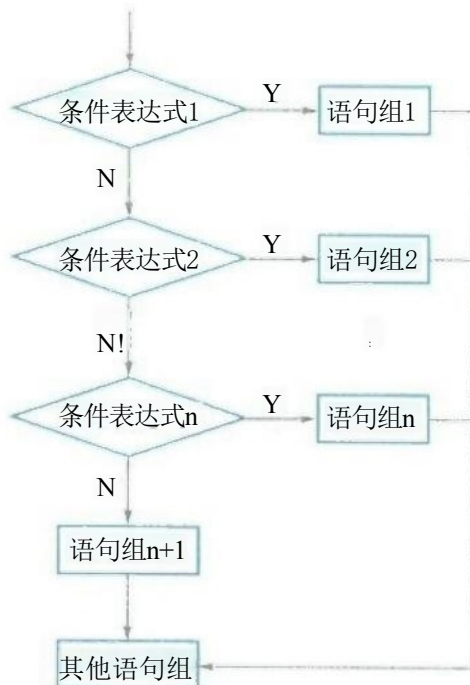
if 语句

if-else 语句

elif 语句

条件表达式

```
if <条件表达式>:  
    <语句组1>  
elif <条件表达式2>:  
    <语句组2>  
...  
elif <条件表达式n>:  
    <语句组n>  
else:  
    <语句组n+1>
```



```
score=int(input('请输入成绩: '))  
if score>=90:  
    print('成绩优')  
elif score>=80:  
    print('成绩良')  
elif score<=60:  
    print('成绩合格')  
else:  
    print('成绩不合格')
```

```
请输入成绩: 92  
成绩优  
>>>
```

```
请输入成绩: 85  
成绩良
```

```
请输入成绩: 71  
成绩合格  
>>>
```

```
请输入成绩: 45  
成绩不合格  
>>>
```

# 使用选择结构

## 4 选择语句

选择语句也称分支语句，Python的选择语句有3种。

if 语句

if-else 语句

elif 语句

**<表达式1> if <条件表达式> else <表达式2>**

```
score = int(input('请输入成绩: '))  
print("成绩合格" if score >= 60 else "成绩不合格")
```

```
score = int(input('请输入成绩: '))  
if score >= 60:  
    print('成绩合格')  
else:  
    print('成绩不合格')
```

if-else 语句不是表达式，没有返回值。而条件表达式不仅可以进行条件判断，还有返回值。

### 编写模拟空调温度设定和控制的程序



夏天打开空调制冷时，当设定温度后，为了节能，空调传感器检测室内温度，当温度高于设定温度时，开始制冷，而当温度降至设定温度时，则转为通风状态，如何根据温度控制制冷还是通风呢？

分析问题

设计算法

编写程序

调试运行程序

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/048020105063006105>