

第 3 章

分支 (*C/C++*)



“分支”是 C/C++ 的一种语法结构，其典型流程及语法格式如下表所示。

	流程图	执行内容	语法
一般用法			<pre>switch(表达式) { case 数值 1: 过程 1 break; case 数值 2: 过程 2 break; case 数值 3: 过程 3 break; case 数值 4: 过程 4 break; default: 过程 5 }</pre>
特殊用法		将表达式（或者变量）与 case 后的常量（可以是整型、浮点型、字符型）逐一比对，如果相等，则执行该 case 后面的相应过程，直到遇到第一个 break 跳出分支结构。如果表达式与所有 case 后的常量均不匹配，则执行 default 后的过程	<pre>switch(表达式) { case 数值 1: case 数值 2: 过程 1 case 数值 3: 过程 2 break; case 数值 4: 过程 3 case 数值 5: 过程 4 break; default: 过程 5 }</pre>

Python 中是没有分支结构。有传言未来的 Python 版本可能会增加这一结构，但未经证实。下面看 C/C++ 中的实例。

例 3.1 今天是星期几 (2) (C/C++ 版)

任务描述

在窗口中输出“今天是星期几？”并输入一个数。

如果是 1、3、5，输出“今天吃米饭”；

如果是 2、4、6，输出“今天吃馒头”；

如果是 7，输出“今天吃面条”；如果是其他数值，输出“请输入 1 ~ 7”。

程序代码如下。

C++

源程序

```
#include <iostream>
using namespace std;

int main()
{
    int a;
    cout<<"今天是星期几? ";
    cin>>a;
    switch(a)                                // 判断 a 的值
    {
        case 1:                             // 为 1
        case 3:                             // 为 3
        case 5:                             // 为 5
            cout<<"今天吃米饭 ";
            break;
        case 2:                             // 为 2
        case 4:                             // 为 4
        case 6:                             // 为 6
            cout<<"今天吃馒头 ";
            break;
        case 7:                             // 为 7
            cout<<"今天吃面条 ";
            break;
        default:                             // 其他情况
            cout<<"请输入 1 ~ 7";
    }
}
```

运行结果 (粗斜体字为输入)

今天是星期几? **3**
今天吃米饭

今天是星期几? **4**
今天吃馒头

今天是星期几? **7**

今天吃面条

今天是星期几? 12

请输入 1 ~ 7

Python

Python 中没有分支结构。本例可以使用选择结构 if/elif/else 结合元组实现，见例 8.1。

例 3.2 对算式计算 (1) (C/C++ 版)

任务描述

输入一个两个数的算式。

如果中间的数学符号为 +-* / 中的一个，则对两个数进行相应的计算并输出计算结果。

如果为其他，则输出计算结果为 0。

程序代码如下。

C++

源程序

```
#include <iostream>
using namespace std;

int main()
{
    float a,b,c;
    char d;
    cin>>a>>d>>b;
    switch(d)
    {
        case '+':
            c=a+b;
            break;
        case '-':
            c=a-b;
            break;
        case '*':
            c=a*b;
            break;
        case '/':
            c=a/b;
            break;
        default:
            c=0;
    }
    cout<<a<<d<<b<<"="<<c<<endl;
}
```

程序注解

- `char d`: `d` 定义为字符型变量。

在 C/C++ 中, 字符型常量以字符外加单引号表示, 如 `'a'`、`'+'`、`'0'` 等。注意, 字符外的单引号 (`'`) 不可省略。

字符可以转化为整型数, 也可以与整型数直接运算。其值为字符的 ASCII 值。如 `'a'` 的值为 97, 空格的值为 32, `'0'` 的值为 48, 注意 `'0'` 不等于 0。

- 字符和字符串:

在 C/C++ 中, 字符串外加双引号 (`" "`), 如 `"hello""123""a"` 等, 在 C/C++ 中, 字符串不作为一种单独的数据类型, 而是作为字符数组来管理 (详见第 7 章)。

注意区分 `'a'` 与 `"a"`。`'a'` 是一个字符变量, 而 `"a"` 是一个包含一个字符的字符串, 两者不能混用。

- `cout<<endl`: 输出一个换行符。

`endl` 也可以写成 `\n`。

C 语言中很多键盘无法直接输入 (表示) 的字符都使用这种方法表示, 如 `\r` 表示回车、`\t` 表示 Tab、`\0` 表示 ASCII 值为 0 的字符。

Python 也这样表示。

如果想输出一个反斜杠 (`\`) 呢? 对了, 与前面的百分号 (`%`) 一样, 写成双反斜杠 (`\\`) 就可以输出单反斜杠 (`\`) 了。

如 `cout<<"c:\\windows"`, 输出窗口中显示的是 `c:\windows`。



运行结果 (粗斜体字为输入)

```
25+34
25+34=59
```

```
12*27
12*27=324
```

Python

Python 只能一次性读入整行字符串, 并不能像 C/C++ 那样, 在一行中读入多个数据。对于此例需要使用选择结构结合字符串操作, 见例 8.3。

本章要点

- 本章学习了 C/C++ 中分支结构的基本用法, 其语法关键字如下表所示。

语言	语法关键字
C/C++	switch、case、break、default

- 注意 C/C++ 中字符和字符串的关系。

练习 3 今天是星期几 (3)

将例 2.2 今天是星期几 (1)，改写为分支语句的格式。