

# 循环和关系表达式

---

## for循环

---

和c语言类似，可以将第一个参数改为`int i = 0;`

`++` and `--`

$\int_a^b f(x)dx = F(a) - F(b)$

```
y = 2 * x++ * (3- ++x);
```

不同编译器的结果不同，尽量不要写类似语句

## 复合语句(代码块)

---

如果在语句块中定义一个新的变量，则晋档程序执行该语句块中的语句时，该变量才存在。

如果在语句块中声明外部已有的变量，新变量将隐藏旧变量。

## 逗号运算符

---

for循环中第三个参数：

```
for(i = 0; i < N; i++, j--){  
    body;  
}
```

逗号具有顺序性，先计算前面的。逗号运算符的优先级是最低的。

## 字符串比较

---

c语言`strcmp(str1, str2)`

string类，直接使用`==`比较即可

## while循环

---

和c一样

for循环可以省略条件，这样条件看做true，但是while不能这么做。

## 库ctime

---

包含有clock()函数，可以返回时间。clock\_t作为其返回类型。

clock()返回数值除以CLOCKS\_PER\_SEC结果的单位为秒

## typedef

---

```
typedef typeName aliasName;  
  
typedef char * byte_pointer;
```

## 基于范围的for循环

---

示例：

```
double prices[5] = {4.99,10.99,6.87,7.99,8.49};  
for (double x : prices)  
    cout << x << std::endl;  
//类似于python  
for(double &x : prices) //&x是引用  
    x = x * 0.80;  
//不同与Python，可以修改引用变量的值，而python不可以  
for(int x : {3, 5, 2, 8, 6})  
    cout << x << ' ';  
cout << '\n';
```

## cin输入

---

```
cin.get(); //读取一个字符，返回这个字符  
  
cin.get(char); //读取一个字符，并赋予char中  
  
cin.get(name,ArSize); //读取ArSize个字符，赋予name中
```

C++函数的重载

## 文件的结尾

在Unix或者windows命令提示符模式下，有重定向运算符，可以把文本文件输入到程序中

```
gofish < fishtale
```

许多PC环境都把Ctrl+Z视为模拟的EOF。

在windows下，可以使用Ctrl+Z和回车来模拟输入EOF。（在linux中，Ctrl+Z是指将程序挂起，通过fg命令恢复）

如果检测到EOF时，cin将eofbit和failbit都设置为1，可以通过cin.eof()来检查eofbit是否被设置

检测到EOF时，cin.eof()将返回true.

如果程序检测到EOF，所有的cin输入的命令都将会被跳过，需要使用cin.clear()来恢复

在某些系统中，Ctrl+Z将结束输入输出，即便使用cin.clear()也没有作用

## 嵌套循环和二维数组

---

### 初始化二维数组

```
int maxtemps[4][5] = //2-D array
{
    {...}
    {...}
    {...}
    {...}
}
```