

# C++语言基础

迂者 - 贺利坚

<http://blog.csdn.net/sxhelijian/>

<http://edu.csdn.net>



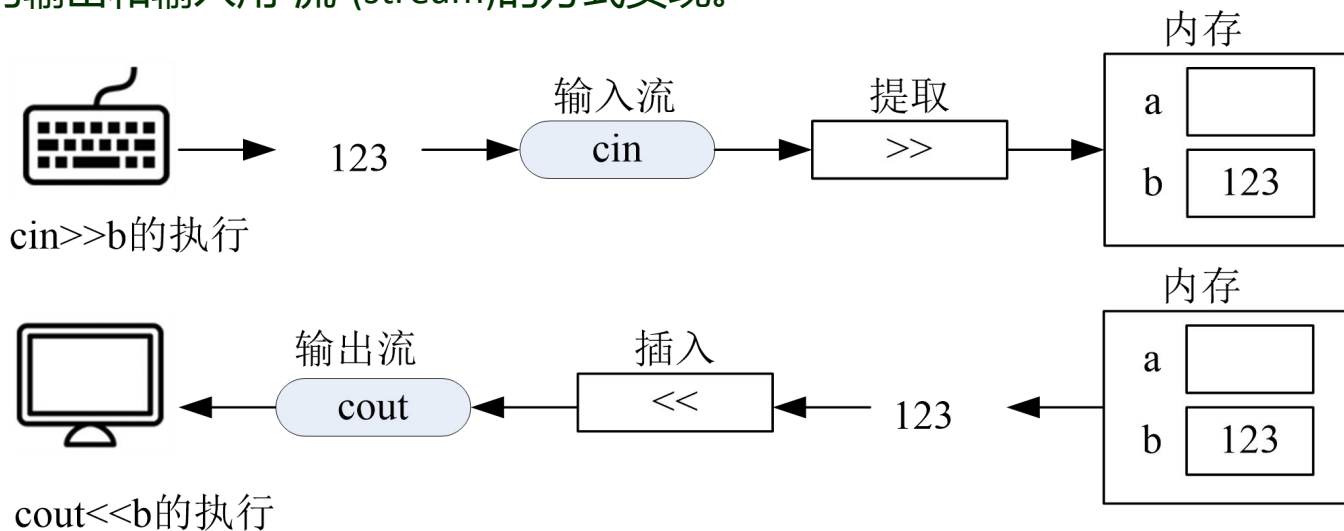


本节主题：

C++中的输入和输出

# C++的输入与输出

☞ C++的输出和输入用“流”(stream)的方式实现。



流对象cin、cout和流运算符的定义存放在C++的输入输出流库iostream中;  
cout、cin并不是C++本身提供的语句,运算符<<和>>也不是C++本身提供的运算符  
要在程序中使用cin、cout和流运算符,必须先: **#include <iostream>**

# 输入流与输出流的基本操作

## 格式

```
cout<<表达式1<<表达式2<<.....<<表达式n;
```

```
cin>>变量1>>变量2>>.....>>变量n;
```

## 例

```
cin>>a>>b>>c;
```

```
cout<<"a="<<a<<"\tb="<<b<<"\tc="<<c<<endl;
```

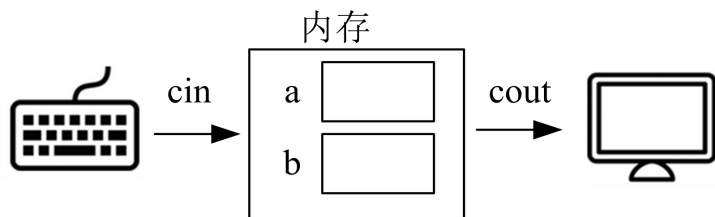
## <<和提取运算符>>的优先级与结合性？

```
int a=1,b=4,c=8;
```

```
cout<<a,b,c;           //输出1
```

```
cout<<a,b,c<<endl;     //出错，如何理解？
```

```
cout<<a+b+c<<endl;     //输出13
```



## cin和>>

- 输入中，有多种方式对数据进行分隔

```
int a, b, c, d;
```

```
cin>>a>>b>>c>>d;
```

- 下面是各种可选的输入形式

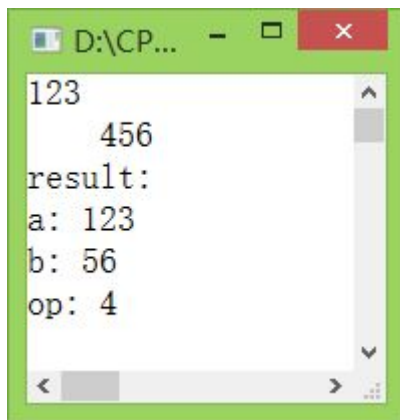
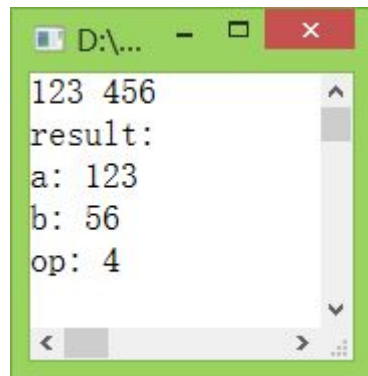
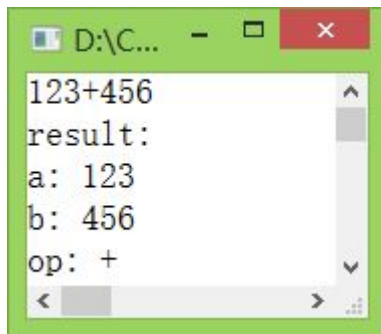
输入形式
1 2 3 4

输入形式
1
2
3
4

输入形式
1 2
3 4

## 程序将自行识别符号

```
#include <iostream>
using namespace std;
int main()
{
    int a,b;
    char op;
    cin>>a>>op>>b
    cout<<"result: "<<"\n";
    cout<<"a: "<<a<<"\n";
    cout<<"b: "<<b<<"\n";
    cout<<"op: "<<op<<"\n";
    return 0;
}
```



cin不会读入空格、回车，  
将其当分隔符。  
将“分隔符”读入，找  
getchar()函数

## cout和<<

- ❏ 用cout输出时，系统会自动判别输出数据的类型，使输出的数据按相应的类型输出。
- ❏ 在必要时，一条语句中多个数据输出合理分隔

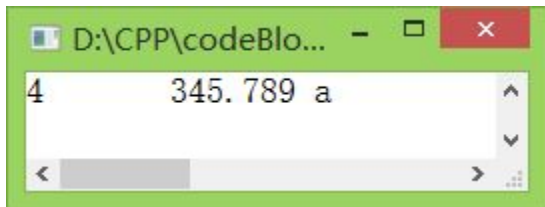
```
int a=4;
```

```
float b=345.7888435;
```

```
char c='a';
```

```
cout<<a<<'\t'<<b<<' '<<c<<endl;
```

- ❏ 输出形式



# 输出格式可以由程序员控制吗？

## 目标：输入输出时满足一些特殊的要求

- 规定字段宽度
- 保留的小数位数
- 数据向左或向右对齐等。

## 方法：使用输入输出流中的控制符

- 设置数值的基数(8,10,16)
- 设置填充
- 设置精度
- 设置对齐
- 设置其他方式

```
#include <iostream>
```

```
#include <iomanip> //不要忘记包含此头文件
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    double d=225.0/7.0;
```

```
    cout<<"d="<<d<<endl;
```

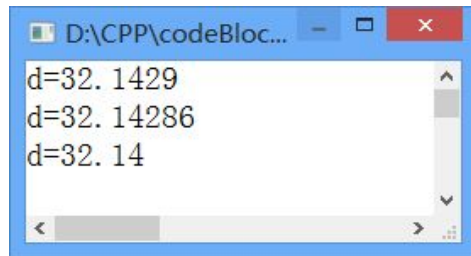
```
    cout<<setiosflags(ios::fixed);
```

```
    cout<<"d="<<setprecision(5)<<d<<endl;
```

```
    cout<<"d="<<setprecision(2)<<d<<endl;
```

```
    return 0;
```

```
}
```





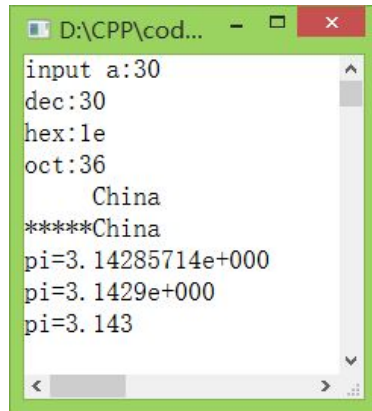
# 输入输出的格式控制符

控制符	作用
dec	设置整数的基数为 10
hex	设置整数的基数为 16
oct	设置整数的基数为 8
setbase(n)	设置整数的基数为 n(n 只能是 8,10,16 三者之一)
setfill(c)	设置填充字符 c,c 可以是字符常量或字符变量
setprecision(n)	设置实数的精度为 n 位。在以一般十进制小数形式输出时 n 代表有效数字。在以 fixed(固定小数位数)形式和 scientific(指数)形式输出时 n 为小数位数
setw(n)	设置字段宽度为 n 位
setiosflags(ios::fixed)	设置浮点数以固定的小数位数显示
setiosflags(ios::scientific)	设置浮点数以科学记数法(即指数形式)显示
setiosflags(ios::left)	输出数据左对齐
setiosflags(ios::right)	输出数据右对齐
setiosflags(ios::skipws)	忽略前导的空格
setiosflags(ios::uppercase)	在以科学记数法输出 E 和以十六进制输出字母 X 时以大写表示
setiosflags(ios::showpos)	输出正数时给出“+”号
resetioflags( )	终止已设置的输出格式状态,在括号中应指定内容

不要强记  
不必强记

## 例：对照代码、结果和上页表格

```
#include <iostream>
#include <iomanip> //不要忘记包含此头文件
using namespace std;
int main()
{
    int a;
    cout<<"input a:";
    cin>>a;
    cout<<"dec:"<<dec<<a<<endl; //以十进制输出整数
    cout<<"hex:"<<hex<<a<<endl; //以十六进制输出整数a
    cout<<"oct:"<<setbase(8)<<a<<endl; //以八进制输出整数a
    char pt[10]="China";
    cout<<setw(10)<<pt<<endl; //指定域宽输出字符串
    cout<<setfill('*')<<setw(10)<<pt<<endl; //指定域宽输出字符串,空白处填充'*'
    double pi=22.0/7.0; //计算pi值
    cout<<setiosflags(ios::scientific)<<setprecision(8); //按指数形式输出,8位小数
    cout<<"pi="<<pi<<endl; //输出pi值
    cout<<"pi="<<setprecision(4)<<pi<<endl; //改为位小数
    cout<<"pi="<<setiosflags(ios::fixed)<<pi<<endl; //改为小数形式输出
    return 0;
}
```



```
input a:30
dec:30
hex:1e
oct:36
China
*****China
pi=3.14285714e+000
pi=3.1429e+000
pi=3.143
```

```
dec
hex
oct
setbase(n)
setfill(c)
setprecision(n)

setw(n)
setiosflags(ios::fixed)
setiosflags(ios::scientific)
setiosflags(ios::left)
setiosflags(ios::right)
setiosflags(ios::skipws)
setiosflags(ios::uppercase)
setiosflags(ios::showpos)
resetioflags()
```

# THANKS

本课程由 迂者-贺利坚 提供

CSDN网站：[www.csdn.net](http://www.csdn.net)

企业服务：<http://ems.csdn.net/>

人才服务：<http://job.csdn.net/>

CTO俱乐部：<http://cto.csdn.net/>

高校俱乐部：<http://student.csdn.net/>

程序员杂志：<http://programmer.csdn.net/>

CODE平台：<https://code.csdn.net/>

项目外包：<http://www.csto.com/>

CSDN博客：<http://blog.csdn.net/>

CSDN论坛：<http://bbs.csdn.net/>

CSDN下载：<http://download.csdn.net/>