**学习笔记----C++ 中iomanip头文件的使用**

转载来自百度百科：<http://baike.baidu.com/link?url=QBZmcyp47EibAoYDWnR5iHYsVSmT8kVTn8R_lgrK1PYgstAXT9G5_Gg3Sr70panAwW3pOcM0seZfhLj4tQzAy_>

iomanip

在c++程序里面经常见到下面的头文件

#include <iomanip>

io代表输入输出，manip是manipulator（操纵器）的缩写(在c++上只能通过输入缩写才有效。）

[iomanip](http://baike.baidu.com/view/1655215.htm)的作用:

主要是对cin,cout之类的一些操纵运算子，比如setfill,setw,setbase,setprecision等等。它是I/O流控制头文件,就像C里面的格式化输出一样.以下是一些常见的控制函数的:

dec 置基数为10 相当于"%d"

hex 置基数为16 相当于"%X"

oct 置基数为8 相当于"%o"

setfill( 'c' ) 设填充字符为c

setprecision( n ) 设显示有效数字为n位

setw( n ) 设 [域宽](http://baike.baidu.com/view/1023138.htm)为n个字符

这个控制符的意思是保证输出宽度为n。如：

cout << setw( 3 ) << 1 << setw( 3 ) << 10 << setw( 3 ) << 100 << endl; 输出结果为

1 10100 （默认是 [右对齐](http://baike.baidu.com/view/1268354.htm)）当输出长度大于3时(<<1000)，setw(3)不起作用。

▲setw(n)用法： 通俗地讲就是预设宽度

如 cout<<setw(5)<<255<<endl;

结果是:

(空格)(空格)255

▲setfill(char c) 用法 : 就是在预设宽度中如果已存在没用完的宽度大小，则用设置的字符c填充

如 cout<<setfill(‘@‘)<<setw(5)<<255<<endl;

结果是:

@@255

应当指出，setfill和setw只作用于紧随其后的部分，例如

cout<<setfill('\*')<<setw(6)<<123<<456;的运行结果为\*\*\*123456，这里set('\*')<<setw(6)只对 123 起作用输出了\*\*\*123,456作为另一部分随后输出。

▲setbase(int n) : 将数字转换为 n 进制.

如 cout<<setbase(8)<<setw(5)<<255<<endl;

cout<<setbase(10)<<setw(5)<<255<<endl;

cout<<setbase(16)<<setw(5)<<255<<endl;

结果是:

(空格)(空格)377

(空格)(空格) 255

(空格)(空格)(空格) f f

▲ setprecision用法

使用setprecision(n)可控制输出流显示浮点数的数字个数。C++默认的流输出数值有效位是6。

如果setprecision(n)与setiosflags(ios::fixed)合用，可以控制小数点右边的数字个数。setiosflags(ios::fixed)是用定点方式表示实数。

如果与setiosflags(ios::scientific)合用， 可以控制指数表示法的小数位数。setiosflags(ios::scientific)是用指数方式表示实数。

setiosflags(ios::fixed) 固定的浮点显示

setiosflags(ios::scientific) 指数表示

setiosflags(ios::left)  [左对齐](http://baike.baidu.com/view/1268353.htm)

setiosflags(ios::right)  [右对齐](http://baike.baidu.com/view/1268354.htm)

setiosflags(ios::skipws) 忽略前导空白

setiosflags(ios::uppercase) 16进制数大写输出

setiosflags(ios::lowercase) 16进制小写输出

setiosflags(ios::showpoint) 强制显示小数点

setiosflags(ios::showpos) 强制显示符号

举例：

#include < [iostream](http://baike.baidu.com/view/1618473.htm)>

#include < [iomanip](http://baike.baidu.com/view/1655215.htm)>

using namespace std;

int main()

{

cout<<12345.0<<endl;//输出"12345"

cout<<setiosflags(ios::fixed)<<setprecision(3)<<1.2345<<endl;输出"1.234"（遵循 [四舍六入五成双](http://baike.baidu.com/view/1245064.htm)的原则，而不是四舍五入的原则）

cout<<setiosflags(ios::scientific)<<12345.0<<endl;//输出"1.234500e+004 "

cout<<setprecision(3)<<12345.0<<endl;//输出"1.23e+004 "

return 0;

}

## c++中控制输出的精度：

**#include <iostream>**

**#include <complex>**

**#include <iomanip>**

**using namespace std;**

**int main()**

**{**

**double a = 1.234567890;**

**cout << fixed << setprecision(1) << a << endl;**

**return 0;**

**}**

