**格式化输出**

1. 引进头文件 #include <iostream>
2. 引用std命名空间，但是为了避免命名空间的的污染，还是不建议用
3. 例如3.1415926 输出都保留小数点二位小数
   1. 代码
      1. #include <iostream>
      2. #include <stdio.h>
      3. #include <iomanip>
      4. using namespace std;
      5. int main()
      6. {
      7. float a = 3.1415926;
      8. printf("printf is : %0.3f\n", a);
      9. cout << "cout is : " <<setiosflags(ios::fixed)<<setprecision(3) << a;
      10. return 0;
      11. }
   2. 分析
      1. 首先我们要引进 iostream 和 iomanip 都文件
      2. setiosflags()
         1. setiosflags(ios::fixed) : 表示固定
         2. setiosflags(ios::left) : 表示靠左
         3. setiosflags(ios::right) : 表示靠右
      3. setprecision(3)： 表示精度 --> 小数点后面几位小数
      4. setw(n) : 表示无论你要输出的数字有多长，我们只输出n位就行
      5. setfill(str) : 表示用 str 来填充空白的位置
   3. 既然我们现在已经大知道怎么回事了，那我们来做一个小例子
      1. 用00:01:30的格式来输出时间
      2. hour = 0; min = 1; sec = 30;
      3. 代码：
         1. #include <iostream>
         2. #include <stdio.h>
         3. #include <iomanip>
         4. using namespace std;
         5. int main()
         6. {
         7. int hour = 0;
         8. int min = 1;
         9. int sec = 30;
         10. cout << "format before" << hour << ":" << min << ":" << sec << endl;
         11. cout << "format after" <<setfill('0')<<setiosflags(ios::right)<<setw(2)<<hour<<":"
         12. <<setiosflags(ios::right)<<setw(2)<<min<<":"<<setiosflags(ios::right)<<setw(2)<<sec;
         13. return 0;
         14. }

以上代码大家可以进行试验一下，我们可以用sleep(1) 来写一个秒表的功能来进行是是实现，哈啊哈