

# C++ <iomanip> 常用操纵符

## 1 简介

在 C++ 中, <iomanip> 头文件提供了一组常用的**输出格式控制**操纵符, 用于设置输出宽度、对齐方式、填充字符、浮点数显示格式、进制输出、布尔显示格式等。它们通常与 `cout` 配合使用, 使输出更整齐、更易读。

## 2 基本使用方法

使用操纵符前, 需要包含头文件:

```
1 #include <iostream>
2 #include <iomanip>
3 using namespace std;
```

示例程序 (先看一个“表格输出”效果):

```
1 int main() {
2     cout << left << setw(10) << "Name"
3         << right << setw(6) << "Score" << endl;
4
5     cout << left << setw(10) << "Alice"
6         << right << setw(6) << 95 << endl;
7
8     cout << left << setw(10) << "Bob"
9         << right << setw(6) << 88 << endl;
10
11     return 0;
12 }
```

### 3 常用操纵符速查表

操纵符	说明
<code>setw(n)</code>	设置下一次输出的最小宽度为 $n$ （一次性）
<code>setprecision(n)</code>	设置输出精度（与 <code>fixed</code> 搭配时为小数位数）
<code>fixed</code>	使用固定小数格式
<code>scientific</code>	使用科学计数法
<code>setfill(c)</code>	设置填充字符为 <code>c</code>
<code>left</code>	左对齐输出
<code>right</code>	右对齐输出（默认）
<code>hex/dec/oct</code>	十六/十/八进制输出整数
<code>setbase(n)</code>	设置进制基数（ $n = 8, 10, 16$ ）
<code>boolalpha</code>	布尔值显示为 <code>true/false</code>
<code>noboolalpha</code>	布尔值显示为 <code>1/0</code>
<code>showpoint</code>	总是显示小数点（及后续小数位）
<code>noshowpoint</code>	不强制显示小数点

### 4 每个操纵符的简单示例

#### 4.1 `setw(n)` —— 设置输出宽度（一次性）

```
1 cout << setw(6) << 123 << endl;  
2 // 输出: 123
```

#### 4.2 `setprecision(n)` —— 设置精度

```
1 double x = 3.14159;  
2 cout << setprecision(3) << x << endl;  
3 // 输出: 3.14
```

#### 4.3 `fixed` —— 固定小数格式

```
1 double x = 3.14159;  
2 cout << fixed << setprecision(2) << x << endl;  
3 // 输出: 3.14
```

#### 4.4 `scientific` —— 科学计数法

```
1 double x = 1234.56;  
2 cout << scientific << x << endl;  
3 // 输出示例: 1.234560e+03
```

#### 4.5 setfill(c) ——设置填充字符

```
1 cout << setfill('*') << setw(6) << 123 << endl;  
2 // 输出: ***123
```

#### 4.6 left ——左对齐

```
1 cout << left << setw(6) << 123 << "|" << endl;  
2 // 输出: 123 |
```

#### 4.7 right ——右对齐

```
1 cout << right << setw(6) << 123 << "|" << endl;  
2 // 输出: 123|
```

#### 4.8 hex/dec/oct ——进制输出

```
1 int x = 26;  
2  
3 cout << hex << x << endl; // 1a  
4 cout << dec << x << endl; // 26  
5 cout << oct << x << endl; // 32
```

#### 4.9 setbase(n) ——设置进制基数

```
1 int x = 26;  
2  
3 cout << setbase(16) << x << endl; // 1a  
4 cout << setbase(10) << x << endl; // 26  
5 cout << setbase(8) << x << endl; // 32
```

#### 4.10 boolalpha ——布尔值显示为 true/false

```
1 bool flag = true;  
2 cout << boolalpha << flag << endl; // true
```

#### 4.11 noboolalpha ——布尔值显示为 1/0

```
1 bool flag = true;  
2 cout << noboolalpha << flag << endl; // 1
```

### 4.12 showpoint ——强制显示小数点

```
1 double x = 10;
2 cout << showpoint << x << endl;
3 // 输出示例: 10.0000
```

### 4.13 noshowpoint ——不强制显示小数点

```
1 double x = 10;
2 cout << noshowpoint << x << endl;
3 // 输出: 10
```

## 5 注意事项

- `setw` 是**一次性**操纵符：只对**下一次**输出生效。
- `left/right/fixed/scientific/hex/boolalpha` 等多为**状态型**操纵符：会持续生效，直到被改回去。
- 常见组合：`setfill + setw + left/right` 用于表格对齐输出。
- `hex/dec/oct/setbase` 主要影响**整数**输出，对浮点数无效。