

C++ 内置类型

number数据类型选择

1. 若不可能为负数，用 `unsigned` 类型
2. 整数运算用 `int`。 `short` 太小， `long` 和 `int` 一般一样长。若数据很大，可以用 `long long`
3. 不要用 `char` 和 `bool` 做算术运算。它们只能用来存放 `characters` 和 `truth values`。 `char` 在有些机器上是 `signed`，在另一些机器上是 `unsigned`。若实在要用来表示 `tiny integer`，要明确地用 `signed char` 或 `unsigned char`。
4. 浮点计算用 `double`。 `float` 经常精度不够。 `double` 浮点运算有时在某些机器上并不 `float` 慢。 `long double` 提供的精度高，但运算速度慢，性价比不高。

类型转换

```
1  bool b = 42;           // b is true
2  b = 0;                 // b is false
3  b = -1;                // b is true
4  int i = 3.14;          // i has value 3
5  double pi = i;         // pi has value 3.0
6  unsigned char c = -1;  // assuming 8-bit chars, c has
                          // value 255
7  signed char c2 = -1;   // assuming 8-bit chars, the
                          // value of c2 is undefined.
8  c2 = -2;               // assuming 8-bit chars, c2 has
                          // value 254
9  c2 = -3;               // assuming 8-bit chars, c2 has
                          // value 253
```

1. 把非布尔数值类型对象赋值给 `bool` 时，若其值为0，则布尔值为 `false`，否则为 `true`。
2. 把 `bool` 赋值给其它数值对象时， `true`->1， `false`->0。
3. 把浮点数赋值给整数，浮点数会被截断，只保留整数部分。
4. 把整数赋值给浮点数，小数部分为0。若整数比浮点数有更多的 `bits`，可能会造成精度丢失。
5. 把一个 `out-of-range` 数值赋值给一个 `unsigned` 类型，直接按补码解释。
6. 把一个 `out-of-range` 数值赋值给一个 `signed` 类型，结果是 `undefined`，可能正常运行，可能崩溃，或产生垃圾数值。