C#中的整数和浮点数

1. int类型，整数

int a = 18;

int b = 6;

int c = a + b;

Console.WriteLine(c);

1. 运算顺序：

C# 语言使用与数学运算规则一致的规则，定义不同数学运算的优先级。 乘法和除法的优先级高于加法和减法，可以在要优先执行的一个或多个运算前后添加括号，从而强制改变运算顺序，整数除法始终生成整数结果。

1. int类型有最小限制和最大限制：

Console.WriteLine( $"{int.MaxValue}-{int.MinValue}" );

1. 使用双精度类型（double）：

double 数字类型表示双精度浮点数。 这些词可能是第一次听说。 浮点数可用于表示数量级可能非常大或非常小的非整数。 双精度意味着存储这些数字时使用的精度高于单精度。 在新式计算机上，使用双精度数字比使用单精度数字更为常见。 接下来，将探索双精度类型

双精度值的范围远大于整数值，

double max = double.MaxValue;

double min = double.MinValue;

Console.WriteLine($"The range of double is {min} to {max}");

Output：

The range of double is -1.79769313486232E+308 to 1.79769313486232E+308

打印出来的这些值用科学记数法表示。 E 左侧为有效数字。 右侧为是 10 的 n 次幂。

1. 固定点类型

decimal 类型。 decimal 类型的范围较小，但精度高于 double。 “固定点”一词意味着，十进制小数点（或二进制小数点）不会移动