

1、整型数值类型：代表一种没有小数点的整数数值。以下 C# 中的整数类型从小范围到大范围，使用时注意根据具体值大小选择合适的类型，避免运算溢出的错误。类型/关键字，是对应的 .NET 类型别名，他们可以互换。比如：
int a=123; 和 System.Int32 b=123; 是一样的。但是我们用的是关键字。整型默认值：0

类型/关键字	说明	字节数	范围	.NET 类型
sbyte	有符号的 8 位整数	1	-128 ~ 127	System.SByte
Short	有符号的 16 位整数	2	-32768 ~ 32767	System.Int16
int	有符号的 32 位整数	4	-2147483648 ~ 2147483648	System.Int32
long	有符号的 64 位整数	8	-9223372036854775808 ~ 9223372036854775808	System.Int64
byte	无符号的 8 位整数	1	0 ~ 255	System.Byte
ushort	无符号的 16 位整数	2	0 ~ 65535	System.UInt16
uint	无符号的 32 位整数	4	0 ~ 4294967295	System.UInt32
ulong	无符号的 64 位整数	8	0 ~ 18446744073709551615	System.UInt64

2、浮点型数值类型：浮点数值类型表示实数。浮点型默认值：0

类型/关键字	说明	精度	字节	大致范围	.NET 类型
float	有符号 32 位，单精度	6-9 位数字	4	$\pm 1.5 \times 10^{-45}$ 至 $\pm 3.4 \times 10^{38}$	System.Single
double	有符号 64 位，双精度	15-17 位数字	8	$\pm 5.0 \times 10^{-324}$ 到 $\pm 1.7 \times 10^{308}$	System.Double
decimal	有符号 128 位，高精度	28-29 位数字	16	$\pm 1.0 \times 10^{-28}$ 至 $\pm 7.9228 \times 10^{28}$	System.Decimal

使用注意：

decimal: 通常用于财务应用程序、货币金额(比如\$1.00、利率 2.625%等)。特别注意精确到只有一个小数的偶数用 decimal 类型处理会更准确。例如 0.1 用 decimal 实例可以精确表示。而没有精确表示 0.1 的 double 或 float 实例。由于数值类型存在这种差异，因此当你对十进制数据使用 double 或 float 时，算术计算可能会出现意外的舍入错误。decimal 不是 C# 的基础类型，使用时对计算的性能有一定影响。

double 是 64 位的，比 float 精度高，但是这些浮点数对于十进制数字来说都是近似数。所以，很有可能在数值运算过程中出现很长的小数，如果通过舍入不能满足要求的时候，可以改成 decimal 类型即可。

在使用中不做任何设置，包含小数的都会默认是 double 类型。比如 9.27 默认就是 double，也就是说如果你直接定义一个 float 类型的变量，赋值为 9.27 是不对的。需要通过 f 或 F 强制指定为 float 类型。比如下面：

```
float a = 9.27f;
```

```
float b = 10.25F;
```

```
float c = (float)10.25; 也可以通过强制转换。
```

其实 double 也可以通过 d 或 D 强制指定，比如下面的三种情况都是正确的：

```
double a = 9.27d;
```

```
double b = 10.25D;
```

```
double c = 10.09;
```

同理，decimal 用 m 或 M 结尾指定。比如： decimal a = 10.9m;

内置的类型转换参考：

<https://docs.microsoft.com/zh-cn/dotnet/csharp/language-reference/builtin-types/numeric-conversions>

3、布尔类型：主要用来表示 true/false 值，一个布尔类型的变量，只能有这两个值，不能将其他的值指定给布尔类型的变量，布尔类型变量也不能跟其他明确的类型之间做转换。默认值：false

类型/关键字	说明	值	.NET 类型
bool	布尔类型	true/false	System.Boolean

4、char 类型：表示 Unicode UTF-16 字符。默认值：`\0` 也就是 U+0000

类型/关键字	范围	大小	.NET 类型
char	U+0000 到 U+FFFF	16 位	System.Char

使用注意：

char 类型可隐式转换为以下整型类型：ushort、int、uint、long 和 ulong。它也可以隐式转换为内置浮点数值类型：float、double 和 decimal。它可以显式转换为 sbyte、byte 和 short 整型类型。

无法将其他类型隐式转换为 char 类型。但是，任何整型或浮点数值类型都可显式转换为 char。

备注：

对于 Unicode 转义序列，必须指定全部四位十六进制值。也就是说，`\u006A` 是一个有效的转义序列，而 `\u06A` 和 `\u6A` 是无效的。

对于十六进制转义序列，可以省略前导零。也就是说，`\x006A`、`\x06A` 和 `\x6A` 转义序列是有效的，并且对应于同一个字符。

参考代码：

```
var chars = new[]
{
    'j',
    '\u006A',
    '\x006A',
    (char)106,
};
Console.WriteLine(string.Join(" ", chars)); // output: j j j j
```

官网参考：<https://docs.microsoft.com/zh-cn/dotnet/csharp/language-reference/builtin-types/char>

其他：

枚举类型

结构类型

对象类型