

# C# 数据类型

在 C# 中，变量分为以下几种类型：

- 值类型 ( Value types )
- 引用类型 ( Reference types )
- 指针类型 ( Pointer types )

## 值类型 ( Value types )

值类型变量可以直接分配给一个值。它们是从类 **System.ValueType** 中派生的。

值类型直接包含数据。比如 **int**、**char**、**float**，它们分别存储数字、字符、浮点数。当您声明一个 **int** 类型时，系统分配内存来存储值。

下表列出了 C# 2010 中可用的值类型：

类型	描述	范围	默认值
bool	布尔值	True 或 False	False
byte	8 位无符号整数	0 到 255	0
char	16 位 Unicode 字符	U +0000 到 U +ffff	'\0'
decimal	128 位精确的十进制值，28-29 有效位数	$(-7.9 \times 10^{28}$ 到 $7.9 \times 10^{28}) / 10^0$ 到 28	0.0M
double	64 位双精度浮点型	$(+/-)5.0 \times 10^{-324}$ 到 $(+/-)1.7 \times 10^{308}$	0.0D
float	32 位单精度浮点型	$-3.4 \times 10^{38}$ 到 $+ 3.4 \times 10^{38}$	0.0F
int	32 位有符号整数类型	-2,147,483,648 到 2,147,483,647	0
long	64 位有符号整数类型	-923,372,036,854,775,808 到 9,223,372,036,854,775,807	0L
sbyte	8 位有符号整数类型	-128 到 127	0
short	16 位有符号整数类型	-32,768 到 32,767	0
uint	32 位无符号整数类型	0 到 4,294,967,295	0
ulong	64 位无符号整数类型	0 到 18,446,744,073,709,551,615	0
ushort	16 位无符号整数类型	0 到 65,535	0

如需得到一个类型或一个变量在特定平台上的准确尺寸，可以使用 **sizeof** 方法。表达式 `sizeof(type)` 产生以字节为单位存储对象或类型的存储尺寸。下面举例获取任何机器上 `int` 类型的存储尺寸：

```
using System;

namespace DataTypeApplication
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Size of int: {0}", sizeof(int));
            Console.ReadLine();
        }
    }
}
```

当上面的代码被编译和执行时，它会产生下列结果：

```
Size of int: 4
```

## 引用类型 ( Reference types )

引用类型不包含存储在变量中的实际数据，但它们包含对变量的引用。

换句话说，它们指的是一个内存位置。使用多个变量时，引用类型可以指向一个内存位置。如果内存位置的数据是由一个变量改变的，其他变量会自动反映这种值的变化。**内置的** 引用类型有：**object**、**dynamic** 和 **string**。

### 对象 ( Object ) 类型

**对象 ( Object ) 类型** 是 C# 通用类型系统 ( Common Type System - CTS ) 中所有数据类型的终极基类。Object 是 System.Object 类的别名。所以对象 ( Object ) 类型可以被分配任何其他类型 ( 值类型、引用类型、预定义类型或用户自定义类型 ) 的值。但是，在分配值之前，需要先进行类型转换。

当一个值类型转换为对象类型时，则被称为 **装箱**；另一方面，当一个对象类型转换为值类型时，则被称为 **拆箱**。

```
object obj;
obj = 100; // 这是装箱
```

### 动态 ( Dynamic ) 类型

您可以存储任何类型的值在动态数据类型变量中。这些变量的类型检查是在运行时发生的。

声明动态类型的语法：

```
dynamic <variable_name> = value;
```

例如：

```
dynamic d = 20;
```

动态类型与对象类型相似，但是对象类型变量的类型检查是在编译时发生的，而动态类型变量的类型检查是在运行时发生的。

## 字符串（String）类型

**字符串（String）类型** 允许您给变量分配任何字符串值。字符串（String）类型是 System.String 类的别名。它是从对象（Object）类型派生的。字符串（String）类型的值可以通过两种形式进行分配：引号和 @引号。

例如：

```
String str = "runoob.com";
```

一个 @引号字符串：

```
@"runoob.com";
```

C# string 字符串的前面可以加 @（称作"逐字字符串"）将转义字符（\）当作普通字符对待，比如：

```
string str = @"C:\Windows";
```

等价于：

```
string str = "C:\\Windows";
```

@ 字符串中可以任意换行，换行符及缩进空格都计算在字符串长度之内。

```
string str = @"<script type=""text/javascript"">
    <!--
    -->
</script>";
```

用户自定义引用类型有：class、interface 或 delegate。我们将在以后的章节中讨论这些类型。

## 指针类型（Pointer types）

指针类型变量存储另一种类型的内存地址。C# 中的指针与 C 或 C++ 中的指针有相同的功能。

声明指针类型的语法：

```
type* identifier;
```

例如：

```
char* cptr;  
int* iptr;
```

我们将在章节"不安全的代码"中讨论指针类型。

[← C# 基本语法](#)[C# 类型转换 →](#)**7 篇笔记**** 写笔记**