**确定字符串是否表示数值（C# 编程指南）**

若要确定字符串是否是指定数值类型的有效表示形式，请使用静态的 **TryParse** 方法，该方法由所有基元数值类型以及诸如 [DateTime](https://msdn.microsoft.com/zh-CN/library/system.datetime.aspx) 和 [IPAddress](https://msdn.microsoft.com/zh-CN/library/system.net.ipaddress.aspx) 这样的类型实现。下面的示例演示如何确定“108”是否为有效的 [int](https://msdn.microsoft.com/zh-CN/library/5kzh1b5w.aspx) 类型。

int i = 0;

string s = "108";

bool result = int.TryParse(s, out i); //i now = 108

如果字符串包含非数值字符或者所包含的数值对于指定的特定类型而言太大或太小，**TryParse** 都将返回 false 并将 out 参数设置为零。否则，它将返回 true，并且将 out 参数设置为字符串的数值。

| **https://wizardforcel.gitbooks.io/msdn-csharp/content/img/note.jpg 注意** |
| --- |
| 有时，虽然字符串可能只包含数值字符，对于您使用的 **TryParse** 方法所属的类型却仍然会无效。例如，“256”对于 **byte** 类型不是有效值，但对于 **int** 类型有效。"98.6”对于 **int** 类型不是有效值，但对于 **decimal** 类型有效。 |

下面的示例演示如何对 **long**、**byte** 和 **decimal** 值的字符串表示形式使用 **TryParse**。

string numString = "1287543"; //"1287543.0" will return false for a long

long number1 = 0;

bool canConvert = long.TryParse(numString, out number1);

if (canConvert == true)

Console.WriteLine("number1 now = {0}", number1);

else

Console.WriteLine("numString is not a valid long");

byte number2 = 0;

numString = "255"; // A value of 256 will return false

canConvert = byte.TryParse(numString, out number2);

if (canConvert == true)

Console.WriteLine("number2 now = {0}", number2);

else

Console.WriteLine("numString is not a valid byte");

decimal number3 = 0;

numString = "27.3"; //"27" is also a valid decimal

canConvert = decimal.TryParse(numString, out number3);

if (canConvert == true)

Console.WriteLine("number3 now = {0}", number3);

else

Console.WriteLine("number3 is not a valid decimal");

## 可靠编程

基元数值类型还实现了 **Parse** 静态方法，此方法会在字符串不是有效数字时引发异常。 **TryParse** 通常更加有效，因为它在数字无效时只是返回 false。