<http://msdn.microsoft.com/es-es/library/0c899ak8.aspx>

# Cadenas con formato numérico personalizado

**.NET Framework 4**

[Otras versiones](javascript:;)

http://i3.msdn.microsoft.com/Areas/Epx/Content/Images/ImageSprite.png

* [.NET Framework 4.5](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.110).aspx)
* [Silverlight](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.95).aspx)
* [.NET Framework 3.5](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.90).aspx)
* [.NET Framework 2.0](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.80).aspx)

Personas que lo han encontrado útil: 2 de 3 - [Valorar este tema](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx#feedback)

Puede crear una cadena de formato numérico personalizado, formada por uno o varios especificadores numéricos personalizados, para definir cómo debe darse formato a los datos numéricos.Una cadena de formato numérico personalizado es cualquier cadena que no sea una [cadena de formato numérico estándar](http://msdn.microsoft.com/es-es/library/dwhawy9k(v=vs.100).aspx).

Algunas sobrecargas del método ToString de todos los tipos numéricos admiten las cadenas de formato numérico personalizado.Por ejemplo, se puede proporcionar una cadena de formato numérico a los métodos [ToString(String, IFormatProvider)](http://msdn.microsoft.com/es-es/library/d830955a(v=vs.100).aspx) y [ToString(String)](http://msdn.microsoft.com/es-es/library/8wch342y(v=vs.100).aspx) del tipo [Int32](http://msdn.microsoft.com/es-es/library/system.int32(v=vs.100).aspx).La [característica de formato compuesto](http://msdn.microsoft.com/es-es/library/txafckwd(v=vs.100).aspx) de .NET Framework, que utilizan algunos métodos Write y WriteLine de las clases [Console](http://msdn.microsoft.com/es-es/library/system.console(v=vs.100).aspx) y [StreamWriter](http://msdn.microsoft.com/es-es/library/system.io.streamwriter(v=vs.100).aspx), el método [String.Format](http://msdn.microsoft.com/es-es/library/system.string.format(v=vs.100).aspx) y el método [StringBuilder.AppendFormat](http://msdn.microsoft.com/es-es/library/system.text.stringbuilder.appendformat(v=vs.100).aspx), admite también cadenas de formato numérico personalizado.

|  |
| --- |
| **SugerenciaSugerencia** |
| Puede descargar la [utilidad de formato](http://go.microsoft.com/fwlink/?LinkId=209564), una aplicación que permite aplicar cadenas de formato a valores numéricos o de fecha y hora y muestra la cadena de resultado. |

En la tabla siguiente se describen los especificadores de formato numérico personalizado y se muestran las salidas de ejemplo generadas por cada especificador de formato.Vea la sección [Notas](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx#NotesCustomFormatting) para obtener información adicional sobre cómo usar las cadenas de formato numérico personalizado y la sección [Ejemplo](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx#example) para ver una ilustración completa de su uso.

|  |  |  |  |
| --- | --- | --- | --- |
| **Especificador de formato** | **Nombre** | **Descripción** | **Ejemplos** |
| "0" | Marcador de posición cero | Reemplaza el cero con el dígito correspondiente si hay alguno presente; de lo contrario, el cero aparece en la cadena de resultado.  Más información: [El especificador personalizado "0"](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx#Specifier0). | 1234.5678 ("00000") -> 01235  0.45678 ("0.00", en-US) -> 0.46  0.45678 ("0.00", fr-FR) -> 0,46 |
| "#" | Marcador de posición de dígito. | Reemplaza la almohadilla con el dígito correspondiente si hay alguno presente; de lo contrario, no aparece ningún dígito en la cadena de resultado.  Más información: [El especificador personalizado "#"](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx#SpecifierD). | 1234.5678 ("#####") -> 1235  0.45678 ("#.##", en-US) -> .46  0.45678 ("#.##", fr-FR) -> ,46 |
| "." | Separador decimal | Determina la ubicación del separador decimal en la cadena de resultado.  Más información: [El especificador personalizado "."](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx#SpecifierPt). | 0.45678 ("0.00", en-US) -> 0.46  0.45678 ("0.00", fr-FR) -> 0,46 |
| "," | Separador de grupos y escala numérica | Actúa como separador de grupos y como especificador de escala numérica.Como separador de grupos, inserta un carácter separador de grupos adaptado entre cada grupo.Como especificador de escala numérica, divide un número por 1000 por cada coma especificada.  Más información: [El especificador personalizado ","](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx#SpecifierTh). | Especificador de separador de grupos:  2147483647 ("##,#", en-US) -> 2,147,483,647  2147483647 ("##,#", es-ES) -> 2.147.483.647  Especificador de escala:  2147483647 ("#,#,,", en-US) -> 2,147  2147483647 ("#,#,,", es-ES) -> 2.147 |
| "%" | Marcador de posición de porcentaje. | Multiplica un número por 100 e inserta un símbolo de porcentaje adaptado en la cadena de resultado.  Más información: [El especificador personalizado "%"](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx#SpecifierPct). | 0.3697 ("%#0.00", en-US) -> %36.97  0.3697 ("%#0.00", el-GR) -> %36,97  0.3697 ("##.0 %", en-US) -> 37.0 %  0.3697 ("##.0 %", el-GR) -> 37,0 % |
| "‰" | Marcador de posición de "por mil" | Multiplica un número por 1000 e inserta un símbolo de "por mil" adaptado en la cadena de resultado.  Más información: [El especificador personalizado "‰"](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx#SpecifierPerMille). | 0.03697 ("#0.00‰", en-US) -> 36.97‰  0.03697 ("#0.00‰", ru-RU) -> 36,97‰ |
| "E0"  "E+0"  "E-0"  "e0"  "e+0"  "e-0" | Notación exponencial | Si va seguido al menos de un 0 (cero), da formato al resultado usando notación exponencial.El modelo de mayúsculas de "E" o "e" indica el modelo de mayúsculas del símbolo de exponente en la cadena de resultado.El número de ceros que siguen al carácter "E" o "e" determina el número mínimo de dígitos en el exponente.Un signo más (+) indica que un carácter de signo precede siempre al exponente.Un signo menos (-) indica que un carácter de signo solo precede a los exponentes negativos.  Más información: [Los especificadores personalizados "E" y "e"](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx#SpecifierExponent). | 987654 ("#0.0e0") -> 98.8e4  1503.92311 ("0.0##e+00") -> 1.504e+03  1.8901385E-16 ("0.0e+00") -> 1.9e-16 |
| \ | Carácter de escape | Hace que el carácter siguiente se interprete como un literal en lugar de como un especificador de formato personalizado.  Más información: [El carácter de escape "\"](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx#SpecifierEscape). | 987654 ("\###00\#") -> #987654# |
| 'cadena'  "cadena" | Delimitador de cadena literal | Indica que los caracteres enmarcados se deben copiar a la cadena de resultado sin modificar. | 68 ("# ' grados'") -> 68 grados  68 ("#' grados'") -> 68 grados |
| ; | Separador de secciones | Define secciones con cadenas de formato diferentes para los números positivos, negativos y cero.  Más información: [El separador de sección ";"](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx#SectionSeparator). | 12.345 ("#0.0#;(#0.0#);-\0-") -> 12.35  0 ("#0.0#;(#0.0#);-\0-") -> -0-  -12.345 ("#0.0#;(#0.0#);-\0-") -> (12.35)  12.345 ("#0.0#;(#0.0#)") -> 12.35  0 ("#0.0#;(#0.0#)") -> 0.0  -12.345 ("#0.0#;(#0.0#)") -> (12.35) |
| Otros | Todos los demás caracteres | El carácter se copia a la cadena de resultado sin modificar. | 68 ("# °") -> 68 ° |

En las secciones siguientes se proporciona información detallada sobre cada uno de los especificadores de formato numérico personalizado.

[El especificador personalizado "0"](javascript:void(0))

El especificador de formato personalizado "0" actúa como un símbolo de marcador de posición cero.Si el valor al que se está dando formato tiene un dígito en la posición donde aparece el cero en la cadena de formato, se copia ese dígito a la cadena de resultado; de lo contrario, aparecerá un cero en la cadena de resultado.La posición del cero que aparece más a la izquierda antes del separador decimal y la del cero que está más a la derecha después del separador decimal determinan el intervalo de dígitos que están siempre presentes en la cadena de resultado.

El especificador "00" hace que el valor se redondee al dígito más próximo que precede al decimal, donde siempre se utiliza el redondeo para evitar el cero.Por ejemplo, al aplicar el formato a 34.5 con "00" el resultado del valor es 35.

En el ejemplo siguiente se muestran varios valores a los que se les ha aplicado cadenas de formato personalizado que incluyen marcadores de posición cero.

C#

[VB](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-1)

[Copiar](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_bf9b5649-0d39-4784-8a86-099a9dcb4894');" \o "Copiar al portapapeles.)

double value;

value = 123;

Console.WriteLine(value.ToString("00000"));

// Displays 00123

value = 1.2;

Console.WriteLine(value.ToString("0.00", CultureInfo.InvariantCulture));

// Displays 1.20

Console.WriteLine(value.ToString("00.00", CultureInfo.InvariantCulture));

// Displays 01.20

Console.WriteLine(value.ToString("00.00",

CultureInfo.CreateSpecificCulture("da-DK")));

// Displays 01,20

value = .56;

Console.WriteLine(value.ToString("0.0", CultureInfo.InvariantCulture));

// Displays 0.6

value = 1234567890;

Console.WriteLine(value.ToString("0,0", CultureInfo.InvariantCulture));

// Displays 1,234,567,890

Console.WriteLine(value.ToString("0,0",

CultureInfo.CreateSpecificCulture("el-GR")));

// Displays 1.234.567.890

value = 1234567890.123456;

Console.WriteLine(value.ToString("0,0.0", CultureInfo.InvariantCulture));

// Displays 1,234,567,890.1

value = 1234.567890;

Console.WriteLine(value.ToString("0,0.00", CultureInfo.InvariantCulture));

// Displays 1,234.57

[Volver a la tabla](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx#table)

[El especificador personalizado "#"](javascript:void(0))

El especificador de formato personalizado "#" actúa como un símbolo de marcador de posición de dígitos.Si el valor al que se está dando formato tiene un dígito en la posición donde aparece el signo de almohadilla en la cadena de formato, ese dígito se copia a la cadena de resultado.En caso contrario, no se almacena nada en esa posición de la cadena de resultado.

Tenga en cuenta que este especificador nunca muestra un cero que no sea un dígito significativo, incluso aunque el cero sea el único dígito de la cadena.Solo mostrará cero si es un dígito significativo del número que se está mostrando.

La cadena de formato "##" hace que el valor se redondee al dígito más próximo que precede al decimal, donde siempre se utiliza el redondeo para evitar el cero.Por ejemplo, al aplicar el formato a 34.5 con "##" el resultado del valor es 35.

En el ejemplo siguiente se muestran varios valores a los que se les ha aplicado cadenas de formato personalizado que incluyen marcadores de posición de dígitos.

C#

[VB](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-2)

[Copiar](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_099a20da-bb97-4565-a15c-9589ac1f3532');)

double value;

value = 1.2;

Console.WriteLine(value.ToString("#.##", CultureInfo.InvariantCulture));

// Displays 1.2

value = 123;

Console.WriteLine(value.ToString("#####"));

// Displays 123

value = 123456;

Console.WriteLine(value.ToString("[##-##-##]"));

// Displays [12-34-56]

value = 1234567890;

Console.WriteLine(value.ToString("#"));

// Displays 1234567890

Console.WriteLine(value.ToString("(###) ###-####"));

// Displays (123) 456-7890

[Volver a la tabla](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx#table)

[El especificador personalizado "."](javascript:void(0))

El especificador de formato personalizado "." inserta un separador decimal localizado en la cadena del resultado.El primer punto de la cadena de formato determina la ubicación del separador decimal en el valor con formato y se omite cualquier punto adicional.

El carácter que se usa como separador decimal en la cadena de resultado no es siempre un punto; viene determinado por la propiedad [NumberDecimalSeparator](http://msdn.microsoft.com/es-es/library/system.globalization.numberformatinfo.numberdecimalseparator(v=vs.100).aspx) del objeto [NumberFormatInfo](http://msdn.microsoft.com/es-es/library/system.globalization.numberformatinfo(v=vs.100).aspx) que controla la aplicación de formato.

En el ejemplo siguiente se utiliza el especificador de formato "." para definir la ubicación del separador decimal en varias cadenas de resultado.

C#

[VB](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-3)

[Copiar](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_06462b7a-c612-4759-9c2c-17331d2bdd81');)

double value;

value = 1.2;

Console.WriteLine(value.ToString("0.00", CultureInfo.InvariantCulture));

// Displays 1.20

Console.WriteLine(value.ToString("00.00", CultureInfo.InvariantCulture));

// Displays 01.20

Console.WriteLine(value.ToString("00.00",

CultureInfo.CreateSpecificCulture("da-DK")));

// Displays 01,20

value = .086;

Console.WriteLine(value.ToString("#0.##%", CultureInfo.InvariantCulture));

// Displays 8.6%

value = 86000;

Console.WriteLine(value.ToString("0.###E+0", CultureInfo.InvariantCulture));

// Displays 8.6E+4

[Volver a la tabla](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx#table)

[El especificador personalizado ","](javascript:void(0))

El carácter "," actúa como separador de grupos y como especificador de escala numérica.

* Separador de grupos: si se especifican una o varias comas dos marcadores de posición de dígitos (0 ó #) que dan formato a los dígitos enteros de un número, se insertará un carácter separador de grupos entre cada grupo de números en la parte entera de la salida.

Las propiedades [NumberGroupSeparator](http://msdn.microsoft.com/es-es/library/system.globalization.numberformatinfo.numbergroupseparator(v=vs.100).aspx) y [NumberGroupSizes](http://msdn.microsoft.com/es-es/library/system.globalization.numberformatinfo.numbergroupsizes(v=vs.100).aspx) del objeto [NumberFormatInfo](http://msdn.microsoft.com/es-es/library/system.globalization.numberformatinfo(v=vs.100).aspx) actual determinan el carácter utilizado como separador de grupos de números y el tamaño de cada grupo de números.Por ejemplo, si se utiliza la cadena "#,#" y la referencia cultural de todos los idiomas para dar formato al número 1000, el resultado será "1,000".

* Especificador de escala numérica: si se especifican una o varias comas inmediatamente a la izquierda del punto decimal explícito o implícito, el número al que se va a dar formato se divide por 1000 por cada coma.Por ejemplo, si se utiliza la cadena "0,," para dar formato al número 100 millones, el resultado será "100".

Puede usar especificadores de separador de grupos y de escala numérica en la misma cadena de formato.Por ejemplo, si se utiliza la cadena "#,0,," y la referencia cultural de todos los idiomas para dar formato al número mil millones, el resultado será "1,000".

En el ejemplo siguiente se muestra el uso de la coma como separador de grupos.

C#

[VB](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-4)

[Copiar](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_8c1666e5-d768-43ce-a70f-09cea867decf');)

double value = 1234567890;

Console.WriteLine(value.ToString("#,#", CultureInfo.InvariantCulture));

// Displays 1,234,567,890

Console.WriteLine(value.ToString("#,##0,,", CultureInfo.InvariantCulture));

// Displays 1,235

En el ejemplo siguiente se muestra el uso de la coma como especificador de escala numérica.

C#

[VB](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-5)

[Copiar](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_fa0fc4d7-c50f-49b3-ad39-bcc429ebdfde');)

double value = 1234567890;

Console.WriteLine(value.ToString("#,,", CultureInfo.InvariantCulture));

// Displays 1235

Console.WriteLine(value.ToString("#,,,", CultureInfo.InvariantCulture));

// Displays 1

Console.WriteLine(value.ToString("#,##0,,", CultureInfo.InvariantCulture));

// Displays 1,235

[Volver a la tabla](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx#table)

[El especificador personalizado "%"](javascript:void(0))

Un signo de porcentaje (%) en una cadena de formato hace que se multiplique un número por 100 antes de darle formato.El símbolo de porcentaje adaptado se inserta en el número en la ubicación donde aparece % en la cadena de formato.La propiedad [PercentSymbol](http://msdn.microsoft.com/es-es/library/system.globalization.numberformatinfo.percentsymbol(v=vs.100).aspx) del objeto [NumberFormatInfo](http://msdn.microsoft.com/es-es/library/system.globalization.numberformatinfo(v=vs.100).aspx) actual define el carácter de porcentaje empleado.

En el ejemplo siguiente se definen varias cadenas de formato personalizado que incluyen el especificador personalizado "%".

C#

[VB](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-6)

[Copiar](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_31aaeaed-d3c9-4df4-b7e9-c01ffa19974b');)

double value = .086;

Console.WriteLine(value.ToString("#0.##%", CultureInfo.InvariantCulture));

// Displays 8.6%

[Volver a la tabla](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx#table)

[El especificador personalizado "‰"](javascript:void(0))

Un carácter de "por mil" (‰ o \u2030) en una cadena de formato hace que un número se multiplique por 1000 antes de darle formato.El símbolo de "por mil" adecuado se inserta en la cadena devuelta, en la ubicación de la cadena de formato en la que aparece el símbolo ‰.La propiedad [NumberFormatInfo.PerMilleSymbol](http://msdn.microsoft.com/es-es/library/system.globalization.numberformatinfo.permillesymbol(v=vs.100).aspx) del objeto que proporciona la información de formato específica de la referencia cultural es la que determina el carácter de "por mil" que se utiliza.

En el ejemplo siguiente se define una cadena de formato personalizado que incluye el especificador personalizado "‰".

C#

[VB](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-7)

[Copiar](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_8207077e-b172-4b4d-baf1-9c35d802b60c');)

double value = .00354;

string perMilleFmt = "#0.## " + '\u2030';

Console.WriteLine(value.ToString(perMilleFmt, CultureInfo.InvariantCulture));

// Displays 3.54‰

[Volver a la tabla](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx#table)

[Los especificadores personalizados "E" y "e"](javascript:void(0))

Si alguna de las cadenas "E", "E+", "E-", "e", "e+", o "e-" está presente en la cadena de formato y va seguida inmediatamente de al menos un cero, se da formato al número mediante notación científica con una 'E' o una 'e' insertadas entre el número y el exponente.El número de ceros que hay a continuación del indicador de notación científica determina el número mínimo de dígitos para el exponente.Los formatos 'E+' y 'e+' indican que un signo más o un signo menos debe preceder siempre al exponente.Los formatos 'E', 'E-', 'e' o 'e-' indican que un carácter de signo debe preceder solo a exponentes negativos.

En el ejemplo siguiente se da formato a varios valores numéricos utilizando los especificadores de notación científica.

C#

[VB](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-8)

[Copiar](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_e36d8c83-74b9-4005-902b-87b0ff4ab505');)

double value = 86000;

Console.WriteLine(value.ToString("0.###E+0", CultureInfo.InvariantCulture));

// Displays 8.6E+4

Console.WriteLine(value.ToString("0.###E+000", CultureInfo.InvariantCulture));

// Displays 8.6E+004

Console.WriteLine(value.ToString("0.###E-000", CultureInfo.InvariantCulture));

// Displays 8.6E004

[Volver a la tabla](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx#table)

[El carácter de escape "\"](javascript:void(0))

Los símbolos "#", "0", ".", ",", "%" y "‰" en una cadena de formato se interpretan como especificadores de formato en lugar de como caracteres literales.Dependiendo de su posición en una cadena de formato personalizado, la "E" en mayúsculas y minúsculas así como los símbolos + y - también se pueden interpretar como especificadores de formato.

Para evitar que un carácter se interprete como un especificador de formato, puede precederlo con una barra diagonal inversa, que es el carácter de escape.El carácter de escape significa que el siguiente carácter es un carácter literal que se debe incluir en la cadena de resultado sin modificar.

Para incluir una barra diagonal inversa en una cadena de resultado, debe indicar su secuencia de escape con otra barra diagonal inversa (\\).

|  |
| --- |
| **NotaNota** |
| Algunos compiladores, como los compiladores de C# y C++, también pueden interpretar un único carácter de barra diagonal inversa como un carácter de escape.Para asegurarse de que una cadena se interpreta correctamente al darle formato, puede usar el carácter literal de cadena textual (el carácter @) antes de la cadena en C# o puede agregar otro carácter de barra diagonal inversa delante de cada barra diagonal inversa en C# y C++.En el siguiente ejemplo de C# se muestran ambos enfoques. |

En el ejemplo siguiente se usa el carácter de escape para evitar que la operación de formato interprete los caracteres "#", "0" y "\" como caracteres de escape o especificadores de formato.En el ejemplo de C# se usa una barra diagonal inversa adicional para asegurarse de que una barra diagonal inversa se interprete como un carácter literal.

C#

[VB](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-9)

[Copiar](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_d3e0ce2b-b3fe-4bc4-b6c9-ae7f28ba2bba');)

int value = 123;

Console.WriteLine(value.ToString("\\#\\#\\# ##0 dollars and \\0\\0 cents \\#\\#\\#"));

// Displays ### 123 dollars and 00 cents ###

Console.WriteLine(value.ToString(@"\#\#\# ##0 dollars and \0\0 cents \#\#\#"));

// Displays ### 123 dollars and 00 cents ###

Console.WriteLine(value.ToString("\\\\\\\\\\\\ ##0 dollars and \\0\\0 cents \\\\\\\\\\\\"));

// Displays \\\ 123 dollars and 00 cents \\\

Console.WriteLine(value.ToString(@"\\\\\\ ##0 dollars and \0\0 cents \\\\\\"));

// Displays \\\ 123 dollars and 00 cents \\\

[Volver a la tabla](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx#table)

[El separador de secciones ";"](javascript:void(0))

El punto y coma (;) es un especificador de formato condicional que aplica distinto formato a un número dependiendo de si su valor es positivo, negativo o cero.Para generar este comportamiento, una cadena de formato personalizado puede contener hasta tres secciones separadas por signos de punto y coma.Estas secciones se describen en la siguiente tabla.

|  |  |
| --- | --- |
| **Número de secciones** | **Descripción** |
| Una sección | La cadena de formato se aplica a todos los valores. |
| Dos secciones | La primera sección se aplica a valores positivos y ceros, y la segunda, sólo a valores negativos.  Si el número al que se va a dar formato es negativo, pero se convierte en cero después de redondearlo según el formato de la segunda sección, se da formato al cero resultante según la primera sección. |
| Tres secciones. | La primera sección se aplica a valores positivos y ceros, la segunda, sólo a valores negativos, y la tercera, a ceros.  La segunda sección se puede dejar vacía (no dejando nada entre los signos de punto y coma) y, en ese caso, la primera sección se aplica a los valores distintos de cero.  Si el número al que se va a dar formato es distinto de cero, pero se convierte en cero después de redondearlo según el formato de la primera o la segunda sección, se da formato al cero resultante según la tercera sección. |

Los separadores de sección omiten cualquier formato preexistente asociado a un número al dar formato al valor final.Por ejemplo, los valores negativos se muestran siempre con signo menos cuando se utilizan separadores de sección.Si se desea que el valor con formato final tenga un signo menos, debe incluir explícitamente el signo menos como parte del especificador de formato personalizado.

En el ejemplo siguiente se usa el especificador de formato ";" para aplicar un formato diferente a los números positivos, negativos y cero.

C#

[VB](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-10)

[Copiar](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_bcdc438b-5d4d-4c91-86f5-db244ab00c0e');)

double posValue = 1234;

double negValue = -1234;

double zeroValue = 0;

string fmt2 = "##;(##)";

string fmt3 = "##;(##);\*\*Zero\*\*";

Console.WriteLine(posValue.ToString(fmt2)); // Displays 1234

Console.WriteLine(negValue.ToString(fmt2)); // Displays (1234)

Console.WriteLine(zeroValue.ToString(fmt3)); // Displays \*\*Zero\*\*

[Volver a la tabla](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx#table)

[Notas](javascript:void(0))

### Infinitos de punto flotante y NaN

Independientemente de la cadena de formato, si el valor de un tipo de punto flotante [Single](http://msdn.microsoft.com/es-es/library/system.single(v=vs.100).aspx) o [Double](http://msdn.microsoft.com/es-es/library/system.double(v=vs.100).aspx) es infinito positivo, infinito negativo o NaN (Not a Number, no es un número), la cadena con formato será el valor de la propiedad [PositiveInfinitySymbol](http://msdn.microsoft.com/es-es/library/system.globalization.numberformatinfo.positiveinfinitysymbol(v=vs.100).aspx), [NegativeInfinitySymbol](http://msdn.microsoft.com/es-es/library/system.globalization.numberformatinfo.negativeinfinitysymbol(v=vs.100).aspx) o [NaNSymbol](http://msdn.microsoft.com/es-es/library/system.globalization.numberformatinfo.nansymbol(v=vs.100).aspx) respectiva especificada por el objeto [NumberFormatInfo](http://msdn.microsoft.com/es-es/library/system.globalization.numberformatinfo(v=vs.100).aspx) aplicable actualmente.

### Configuración del Panel de control

Los valores de configuración del elemento Configuración regional y de idioma del Panel de control influyen en la cadena de resultado generada por una operación de formato.Estos valores de configuración se utilizan para inicializar el objeto [NumberFormatInfo](http://msdn.microsoft.com/es-es/library/system.globalization.numberformatinfo(v=vs.100).aspx) asociado a la referencia cultural del subproceso actual, y la referencia cultural del subproceso actual proporciona valores que se utilizan para controlar el formato.Los equipos que usan configuraciones diferentes generarán cadenas de resultado distintas.

Asimismo, si se usa el constructor CultureInfo.CultureInfo(String) para crear instancias de un nuevo objeto [CultureInfo](http://msdn.microsoft.com/es-es/library/system.globalization.cultureinfo(v=vs.100).aspx) que representa la misma referencia cultural que la referencia cultural del sistema actual, cualquier personalización establecida por el elemento Configuración regional y de idioma del Panel de control se aplicará al nuevo objeto [CultureInfo](http://msdn.microsoft.com/es-es/library/system.globalization.cultureinfo(v=vs.100).aspx).Puede usar el constructor CultureInfo.CultureInfo(String, Boolean) para crear un objeto [CultureInfo](http://msdn.microsoft.com/es-es/library/system.globalization.cultureinfo(v=vs.100).aspx) que no refleje las personalizaciones de un sistema.

### Cadenas de formato de punto fijo y redondeo

Para las cadenas de formato de punto fijo (es decir, las cadenas de formato que no contienen caracteres de formato de notación científica), los números se redondean hasta tantos decimales como marcadores de posición de dígitos haya a la derecha del separador decimal.Si la cadena de formato no contiene ningún separador decimal, el número se redondea al entero más próximo.Si el número tiene más dígitos que marcadores de posición de dígitos a la izquierda del separador decimal, los dígitos adicionales se copian en la cadena de resultado justo antes del primer marcador de posición de dígitos.

[Volver a la tabla](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx#table)

[Ejemplo](javascript:void(0))

En el siguiente ejemplo se muestran dos cadenas de formato numérico personalizado.En ambos casos, el marcador de posición de dígitos (#) muestra los datos numéricos y todos los demás caracteres se copian a la cadena de resultado.

C#

[VB](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-11)

[Copiar](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_8631fddb-76ca-4d26-a928-4643a0258e5a');)

double number1 = 1234567890;

string value1 = number1.ToString("(###) ###-####");

Console.WriteLine(value1);

int number2 = 42;

string value2 = number2.ToString("My Number = #");

Console.WriteLine(value2);

// The example displays the following output:

// (123) 456-7890

// My Number = 42

[Volver a la tabla](http://msdn.microsoft.com/es-es/library/0c899ak8(v=vs.100).aspx#table)

[Vea también](javascript:void(0))

#### Tareas

[Cómo: Rellenar un número con ceros iniciales](http://msdn.microsoft.com/es-es/library/dd260048(v=vs.100).aspx)

#### Referencia

[NumberFormatInfo](http://msdn.microsoft.com/es-es/library/system.globalization.numberformatinfo(v=vs.100).aspx)

#### Conceptos

[Aplicar formato a tipos](http://msdn.microsoft.com/es-es/library/26etazsy(v=vs.100).aspx)

[Cadenas con formato numérico estándar](http://msdn.microsoft.com/es-es/library/dwhawy9k(v=vs.100).aspx)