

Tipos de datos integrados (Visual C# Express)

Visual Studio 2008 Personas que lo han encontrado útil: 5 de 6

Actualización: noviembre 2007

C# es un lenguaje con establecimiento inflexible de tipos. Antes de que un valor se pueda almacenar en una variable, se debe especificar el tipo de la variable, como en los ejemplos siguientes:

```
int a = 1;
string s = "Hello";
XmlDocument tempDocument = new XmlDocument();
```

Observe que el tipo se debe especificar igualmente para los tipos simples integrados como [int](#) y para los tipos complejos o personalizados como [XmlDocument](#).

C# es compatible con los siguientes tipos de datos integrados:

Tipo de datos	Intervalo
byte	0 .. 255
sbyte	-128 .. 127
short	-32,768 .. 32,767
ushort	0 .. 65,535
int	-2,147,483,648 .. 2,147,483,647
uint	0 .. 4,294,967,295
long	-9,223,372,036,854,775,808 .. 9,223,372,036,854,775,807
ulong	0 .. 18,446,744,073,709,551,615
float	-3.402823e38 .. 3.402823e38
double	-1.79769313486232e308 .. 1.79769313486232e308
decimal	-79228162514264337593543950335 .. 79228162514264337593543950335
char	Un carácter Unicode.
string	Una cadena de caracteres Unicode.
bool	True o False.
object	Un objeto.

Estos nombres de tipo de datos son los alias para los tipos predefinidos en el espacio de nombres [System](#). Estos se muestran en la sección [Tabla de tipos integrados \(Referencia de C#\)](#). Todos estos tipos, salvo el objeto y la cadena, son tipos de valores. Para obtener más información, vea [Tipos de referencia y de valor \(Visual C# Express\)](#).

Utilizar los tipos de datos integrados

Los tipos de datos integrados se utilizan dentro de un programa C# de diferentes maneras.

Como variables:

C#

```
int answer = 42;
string greeting = "Hello, World!";
```

Como constantes:

C#

```
const int speedLimit = 55;
const double pi = 3.14159265358979323846264338327950;
```

Como valores devueltos y parámetros:

C#

```
long CalculateSum(int a, int b)
{
    long result = a + b;
    return result;
}
```

Para definir sus propios tipos de datos, utilice [Clases \(Visual C# Express\)](#), [Enumeraciones \(Visual C# Express\)](#) o [Estructuras \(Visual C# Express\)](#).

Convertir tipos de datos

Los tipos de datos se pueden convertir implícitamente, en cuyo caso el compilador hace automáticamente la conversión, o explícitamente utilizando un operador de conversión explícita, donde el programador fuerza la conversión y asume el riesgo de perder información.

Por ejemplo:

C#

```
int i = 0;
double d = 0;

i = 10;
d = i;           // An implicit conversion

d = 3.5;
i = (int) d;    // An explicit conversion, or "cast"
```

Vea también

Conceptos

[Manual del lenguaje C#](#)

[Constantes y variables \(Visual C# Express\)](#)

Referencia

[Tabla de tipos integrados \(Referencia de C#\)](#)

Otros recursos

[Aplicar formato a la tabla de resultados numéricos \(Referencia de C#\)](#)

Adiciones de comunidad
