[Reglas para la sintaxis de línea de comandos del compilador de C#](javascript:void(0)" \o ")

El compilador de C# utiliza las siguientes reglas para interpretar los argumentos empleados en la línea de comandos del sistema operativo:

* Los argumentos van delimitados por espacio en blanco, que puede ser un carácter de espacio o una tabulación.
* El carácter de intercalación (^) no se reconoce como carácter de escape ni como delimitador. El analizador de la línea de comandos del sistema operativo procesa este carácter antes de pasarlo a la matriz argv del programa.
* Una cadena entre comillas ("cadena") se interpreta como un solo argumento, sin importar el espacio en blanco que contenga. Se puede incrustar una cadena entre comillas dentro de un argumento.
* Unas comillas precedidas por una barra diagonal inversa (\") se interpretan como un literal de cadena de comillas (").
* Las barras diagonales inversas se interpretan literalmente, a menos que precedan inmediatamente a unas comillas.
* Si a un número par de barras diagonales inversas le siguen unas comillas, se coloca una barra diagonal inversa en la matriz argv por cada par de barras diagonales y se interpretan las comillas como delimitador de cadenas.
* Si a un número impar de barras diagonales inversas le siguen unas comillas, se coloca una barra diagonal inversa en la matriz argv por cada par de barras diagonales y se interpretan las comillas como un carácter de escape gracias a la barra diagonal restante. Esto hace que se agregue un literal de comillas dobles (") a argv.

[Líneas de comandos de ejemplo para el compilador de C#](javascript:void(0))

* Compila File.cs y genera File.exe:
* csc File.cs
* Compila File.cs y genera File.dll:
* csc /target:library File.cs
* Compila File.cs y crea My.exe:
* csc /out:My.exe File.cs
* Compila todos los archivos de C# del directorio actual, con optimizaciones activadas, y define el símbolo DEBUG.Como resultado se genera el archivo File2.exe:
* csc /define:DEBUG /optimize /out:File2.exe \*.cs
* Compila todos los archivos de C# del directorio actual, generando una versión de depuración de File2.dll.No se muestra logotipo ni ningún tipo de advertencias:
* csc /target:library /out:File2.dll /warn:0 /nologo /debug \*.cs
* Compila todos los archivos de C# del directorio actual, generando Something.xyz (una DLL):

csc /target:library /out:Something.xyz \*.cs

## Compilar el código

Para compilar el archivo MathLibrary.DLL, compile los dos archivos Add y Mult mediante la siguiente línea de comandos.

C#

csc /target:library /out:MathLibrary.DLL Add.cs Mult.cs

La opción [/target:library](https://msdn.microsoft.com/es-es/library/e13syb43.aspx) indica al compilador que genere un archivo DLL en vez de un archivo EXE.La opción [/out](https://msdn.microsoft.com/es-es/library/bw3t50f3.aspx) del compilador seguida de un nombre de archivo se utiliza para especificar el nombre de archivo de la DLL.Si no se utiliza, el compilador usa el primer archivo (Add.cs) como nombre de la DLL.

Para compilar el archivo ejecutable, TestCode.exe, utilice la siguiente línea de comandos:

C#

csc /out:TestCode.exe /reference:MathLibrary.DLL TestCode.cs

La opción **/out** indica al compilador que genere un archivo EXE y especifica el nombre del archivo de salida (TestCode.exe).Esta opción del compilador no es de uso obligatorio.La opción [/reference](https://msdn.microsoft.com/es-es/library/yabyz3h4.aspx) del compilador especifica el archivo o archivos DLL que utiliza el programa.Para obtener más información, vea [/reference](https://msdn.microsoft.com/es-es/library/yabyz3h4.aspx) para C# y [/reference (Visual Basic)](https://msdn.microsoft.com/es-es/library/czhbsf4x.aspx) para Visual Basic.