



APRENDERAPROGRAMAR.COM

CONFIGURAR EDITOR
CODE::BLOCKS. REVISOR
ORTOGRÁFICO. NUMERAR
LÍNEAS. PLEGADO (FOLD).
AUTOCOMPLETAR
(CU00516F)

Sección: Cursos

Categoría: Curso básico de programación en lenguaje C desde cero

Fecha revisión: 2031

Resumen: Entrega nº16 del curso básico "Programación C desde cero".

Autor: Mario Rodríguez Rancel

CONFIGURAR EL EDITOR DE CODE::BLOCKS

Ya hemos visto dónde debemos escribir el código asociado a un programa. Hay que destacar que Code::Blocks facilita el que podamos escribir código con rapidez a través de:



- Propuestas para elección de nombres de variables.
- Corrección o indicación automática de errores que podamos cometer detectados por el compilador.
- Editor que incorpora resaltado de código en distintos colores, números de línea, etc.

Code::Blocks dispone de multitud de opciones que podemos activar, desactivar o modificar según nos resulten o no convenientes. Ya hemos hablado de algunas de ellas y vamos ahora a comentar algunas cuestiones adicionales al respecto. En el menú Settings - Editor..., disponemos de diferentes opciones relacionadas con el editor donde escribimos código:

- Opciones Generales o *General Settings*.
- Plegado del código o *Folding*.
- Márgenes del editor y opciones del cursor o *Margins and Caret*.
- Juego de colores para el código o *Syntax highlighting*.
- Código por defecto al abrir el editor o *Default Code*.
- Instrucciones con relleno automático o *Abbreviations*.
- Autoformato del código o *Source Formatter*.
- Opciones para últimos ficheros utilizados o *Browse Tracker*.
- Ayuda en la escritura o *Code Completion*.
- Delimitadores de comentarios para análisis estadístico o *Code Statistic Settings*.
- Formatos para comentarios o *DoxyBlocks*.
- Opciones de scrolling y ratón o *Mouse Drag Scrolling*.
- Opciones de remarcado del texto para búsquedas o *Incremental Search Settings*.
- Atajos de teclado o *Keyboard Shortcuts*.
- Revisor ortográfico o *SpellChecker*.
- Remarcado de texto igual al seleccionado u *Occurrences Highlighting*.
- Otros.

Existen decenas de opciones configurables, por lo que no vamos a analizar todas ellas. Simplemente vamos a indicar algunas que pensamos pueden resultar útiles u orientativas de las grandes posibilidades que ofrece Code::Blocks.

Las opciones de configuración están basadas en algunas ocasiones en plugins o extensiones de Code::Blocks, por tanto la instalación o desinstalación de estos plugins puede hacer que nos aparezca un mayor o menor número de posibilidades de configuración.

Crea un proyecto nuevo con un código sencillo y accede al menú Settings - Editor... – General Settings. En el apartado End-of-line options activa la opción "Show end-of-line chars". Comprobarás que al final de cada línea aparecen unos caracteres especiales con la marca CR y LF, que corresponden a retorno de carro (CR o carriage return) y salto de línea (LF o line feed).

```

1  #include <stdio.h>CR LF
2  #include <stdlib.h>CR LF
3  CR LF
4  int main() {CR LF
5      printf("Curso aprenderaprogramar.com!\n");CR LF
6      return 0;CR LF
7  }CR LF
8
    
```

Normalmente trabajaremos con estos caracteres ocultos, pero en algún caso nos puede interesar activar su visualización.

Aquí también podemos modificar el número de caracteres que avanza el tabulador, modificando el valor establecido en "Tab size in spaces", o mostrar u ocultar los números de línea usando la opción "Show line numbers". Muchas opciones de configuración pueden modificarse desde distintos puntos de Code::Blocks. Por ejemplo también puedes hacer que se muestren u oculten los números de línea accediendo a través del menú Edit – Editor Tweaks... – Show Line Numbers: obtendrás el mismo resultado que si realizas la configuración a través del menú Settings - Editor....

Accede al menú Settings - Editor... – Folding. Cuando la opción "Enable folding" está activada tenemos opción a pulsar en un pequeño icono que permite plegar un contenido agrupado (por ejemplo una función).

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main() {
5      printf("Curso aprenderaprogramar.com!\n");
6      return 0;
7  }
8
    
```

Al pulsar sobre el signo de plegado se oculta el contenido interno. Esto puede ser útil cuando tenemos programas largos donde queremos analizar sólo ciertas partes del código manteniendo ocultas las otras partes.

Si activamos la opción "Fold all on file open" todos los elementos plegables aparecerán plegados cuando abramos un fichero.

Accede ahora al menú Settings - Editor... – Margins and Caret. Modifica la opción "Caret Width" y ponle un valor 3. Comprobarás que el guión del cursor sobre la pantalla se visualiza con mayor grosor del habitual. Esto puede ser adecuado para personas que tengan problemas de vista o simplemente para aquellos que prefieran verlo así. Si no tienes necesidades especiales, recomendamos mantener el valor de defecto que es 1.

Accede al menú Settings - Editor... – Syntax Highlighting. Si te fijas en el código verás que aparecen distintos colores como verde para las directivas del preprocesador, rojo para los puntos y coma y símbolos igual, azul claro para el texto entrecomillado, etc. Estos colores pueden ser modificados desde aquí si se desea.

El menú Settings - Editor... – Default code permite establecer un código por defecto para que aparezca al crear nuevos ficheros de código. Puede usarse por ejemplo para poner los datos de autor, versión, etc. en un proyecto que conste de numerosos ficheros.

El menú Settings - Editor... – Abbreviations permite disponer de una colección de instrucciones o fragmentos de código de uso frecuente. Algunas instrucciones vienen por defecto en Code::Blocks, aunque nosotros también podemos añadir nuestros propios fragmentos de código de uso habitual. Accede a esta opción: verás que te aparece una columna de Keywords como class, for, forb, ..., switch, ..., while, whileb, etc.

Vete ahora al editor de código y escribe switch, y a continuación pulsa las teclas CTRL + J (teclas control y j simultáneamente). Verás que el editor genera un fragmento de código correspondiente a esta instrucción. Esto puede ser útil cuando tengamos que escribir programas largos o donde se repiten con frecuencia ciertos fragmentos de código. Estos son algunos ejemplos de abreviaturas:

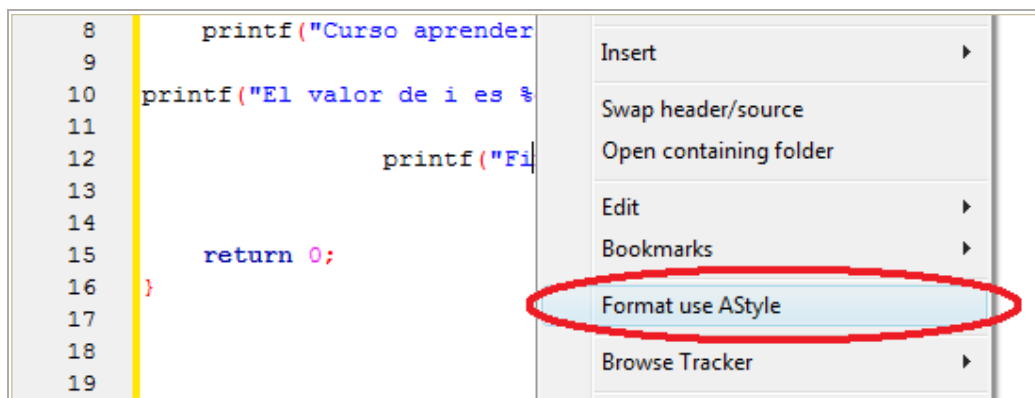
Palabra clave	Efecto al escribirla en el editor y pulsar CTRL + J
for	<pre>for (; ;) ;</pre>
forb	<pre>for (; ;) { }</pre>
switch	<pre>switch () { case : break; default: break; }</pre>

El código de las abbreviations es editable. Es decir, puedes modificarlo para que aparezca a tu gusto. También puedes crear las nuevas abbreviations que te resulten necesarias.

Accede al menú Settings - Editor... – Source Formatter. Aquí aparecen distintas opciones para el formato de código (posición de las llaves de apertura y cierre, tabulaciones, etc.) que se desee emplear. Entre las opciones disponibles tenemos formato Allman (Ansi), formato Java, formato K&R, formato Stroustrup, formato Linux, etc. Cada formato da lugar a una organización diferente del código en cuanto a su visualización (posición de las llaves de apertura y cierre, saltos de línea, tabulaciones) aunque el resultado de ejecución será el mismo independientemente de cuál sea el formato de presentación. A continuación vemos algunos ejemplos:

Ejemplo formato Allman	Ejemplo formato Java	Ejemplo formato Linux
<pre>int Foo(bool isBar) { if (isBar) { bar(); return 1; } else return 0; }</pre>	<pre>int Foo(bool isBar) { if (isBar) { bar(); return 1; } else return 0; }</pre>	<pre>int Foo(bool isBar) { if (isBar) { bar(); return 1; } else return 0; }</pre>

Es interesante elegir un formato y respetarlo. Muchas veces al crear código se produce una cierta desorganización en la apertura y cierre de llaves, tabulaciones, etc. Para conseguir una presentación siempre uniforme Code::Blocks, al igual que otros editores y entornos de desarrollo, tiene una opción que nos permite aplicar un formateado automático al código. Para ello, una vez elegida la opción de formato, nos situamos encima del código y pulsamos el botón derecho del ratón, donde debemos elegir la opción "Format use AStyle".

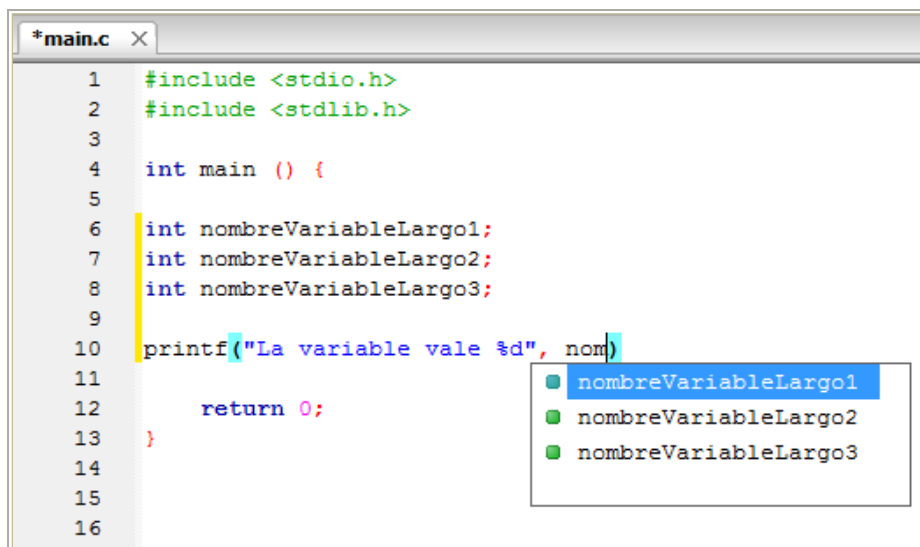


Escribe un código donde no exista una buena organización de apertura de llaves, tabulaciones, etc. como el que mostramos a continuación. Seguidamente pon el ratón encima del código, pulsa botón derecho y elige "Format use AStyle". El código se autoorganizará automáticamente como en este ejemplo:

Antes del formateado automático	Después del formateado automático (tipo Java)
<pre>#include <stdio.h> #include <stdlib.h> // Ejemplo aprenderaprogramar.com int main () { int i; i=5; printf("Curso aprenderaprogramar.com\n"); printf("El valor de i es %d\n", i); printf("Fin del programa\n"); return 0; }</pre>	<pre>#include <stdio.h> #include <stdlib.h> // Ejemplo aprenderaprogramar.com int main () { int i; i=5; printf("Curso aprenderaprogramar.com\n"); printf("El valor de i es %d\n", i); printf("Fin del programa\n"); return 0; }</pre>

El código correctamente ordenado es más fácil de leer y entender, por eso te recomendamos que apliques el formateado automático en tus programas.

Accede al menú Settings - Editor... – Code Completion. Cuando la opción “Disable code completion” está desactivada nos aparecen sugerencias de nombres de variables o instrucciones a medida que escribimos, y esto nos permite escoger el nombre de variable o instrucción adecuada. No obstante, en ocasiones puede resultar incómodo el continuo salto de sugerencias y basta marcar la opción indicada para desactivarlo.



Existen distintas opciones adicionales de configuración en las que no vamos a entrar como las relacionadas con los formatos de los comentarios, comportamiento del ratón, forma en que aparecen los resultados de búsqueda, atajos de teclado, etc.

ORDENACIÓN DE LÍNEAS. NUMERACIÓN DE LÍNEAS.

La ordenación del código con C será tal y como hemos trabajado con el pseudocódigo en el curso "Bases de la programación nivel I" de aprenderaprogramar.com, es decir, las líneas dentro un módulo se ejecutan de izquierda a derecha y de arriba hacia abajo. En cuanto a numeración de líneas, ya hemos indicado que se puede activar o desactivar la presencia de números de línea en el editor de Code::Blocks a través del menú Settings - Editor... - General Settings opción "Other options - Line numbers".

Los números de línea no forman parte del código, son únicamente información auxiliar. Recomendamos mantener los números de línea visibles porque cuando tengamos errores en nuestro programa los mensajes de advertencia generalmente harán referencia al número de línea en que se ha producido el error. Tenerlos visibles nos será útil por tanto para localizar con mayor facilidad los errores.

EJERCICIO

Responde razonadamente a estas preguntas:

- a) ¿Crees que sería posible escribir código C utilizando un editor simple como el bloc de notas de Windows?
- b) Si no es posible, ¿por qué? Si es posible, ¿crees que podríamos compilar y ejecutar ese código sin usar Code::Blocks?
- c) ¿En qué casos piensas que puede resultar útil tener el Spell Checker o corrector ortográfico de Code::Blocks activado?
- d) ¿Crees que si escribimos un código C en Code::Blocks bajo Windows, ese mismo código podría ser compilado y ejecutado en Linux ó en Macintosh?

Para comprobar si tus respuestas son correctas puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU00517F

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:
http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=82&Itemid=210