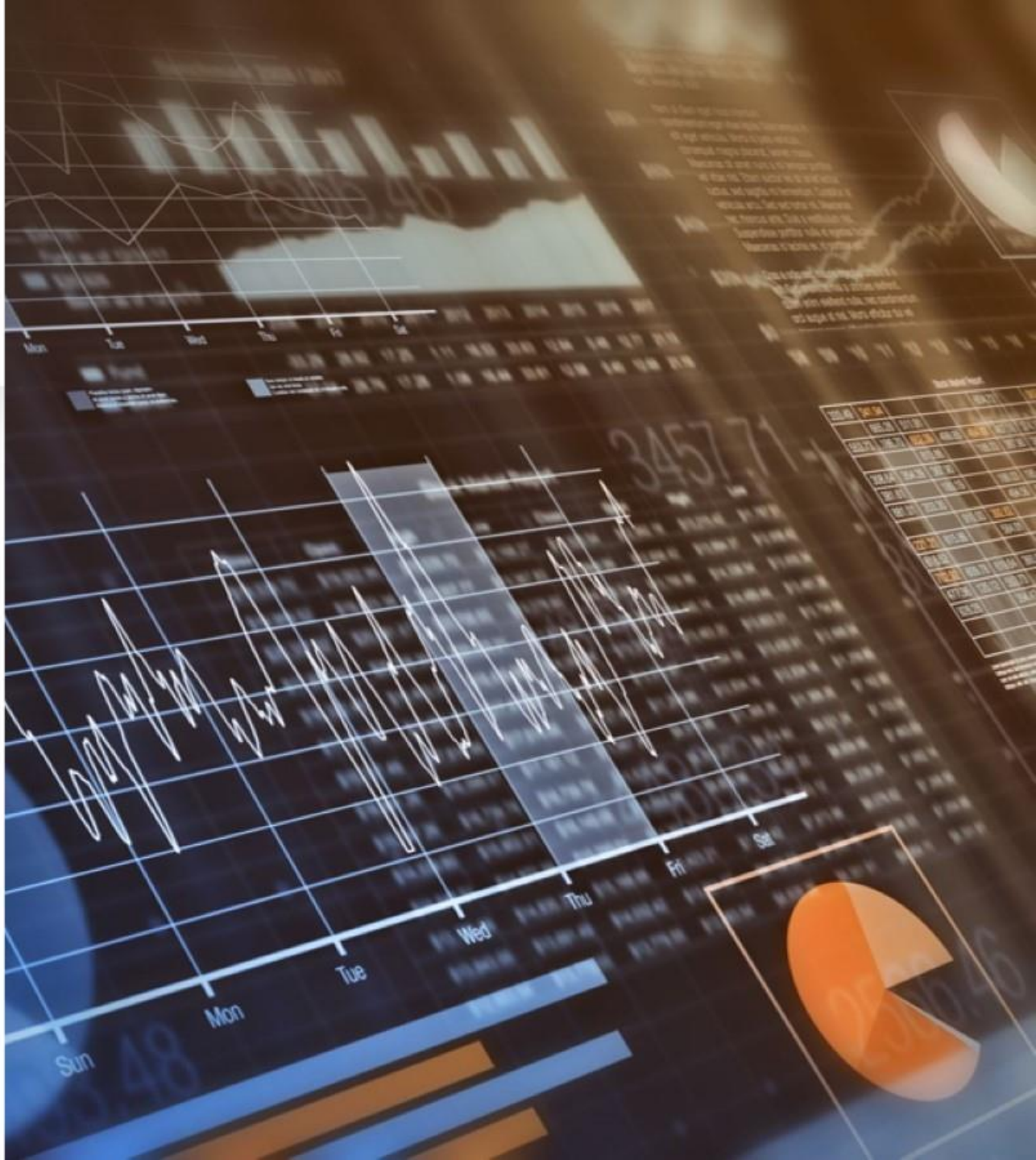
Aprende C

Diego Iván Oliveros Acosta

Introducción

- C es un lenguaje de programación de propósito general que ha sido ampliamente utilizado durante más de 50 años.
- C es muy potente; se ha utilizado para desarrollar sistemas operativos, bases de datos, aplicaciones, etc.

Diego Iván Oliveros Acosta @scalapp.co



¿Qué es C?

- C es un lenguaje de programación de propósito general creado por Dennis Ritchie en los Laboratorios Bell en 1972.
- Es un lenguaje muy popular, a pesar de ser antiguo. La principal razón de su popularidad es porque es un lenguaje fundamental en el campo de la informática.
- C está fuertemente asociado con UNIX, ya que fue desarrollado para escribir el sistema operativo UNIX.

Diego Iván Oliveros Acosta @scalapp.co

¿Por qué aprender C?

- Es uno de los lenguajes de programación más populares del mundo.
- Si conoces C, no tendrás problemas para aprender otros lenguajes de programación populares como Java, Python, C++, C#, etc., ya que la sintaxis es similar.
- C es muy rápido, en comparación con otros lenguajes de programación, como Java y Python.
- C es muy versátil; se puede utilizar tanto en aplicaciones como en tecnologías.

Diferencia entre C y C++

- C++ se desarrolló como una extensión de C, y ambos lenguajes tienen casi la misma sintaxis.
- La principal diferencia entre C y C++ es que C++ admite clases y objetos, mientras que C no.

¿Qué se requiere para empezar?

- Para empezar a utilizar C, necesitas dos cosas:
- Un editor de texto, como el Bloc de notas, para escribir código C
- Un compilador, como GCC, para traducir el código C a un lenguaje que la computadora pueda entender.
- Hay muchos editores de texto y compiladores entre los que elegir. En este tutorial, utilizaremos un ***IDE*** (ver a continuación).

C Instalar IDE

- Se utiliza un IDE (entorno de desarrollo integrado) para editar Y compilar el código.
- Entre los IDE más populares se encuentran Code::Blocks, Eclipse y **Visual Studio**. Todos son gratuitos y se pueden utilizar para editar y depurar código C.
- Nota: Los IDE basados en web también pueden funcionar, **pero la funcionalidad es limitada.**
- Usaremos Code::Blocks en nuestro tutorial, que creemos que es un buen lugar para comenzar.
- Puede encontrar la última versión de Codeblocks en http://www.codeblocks.org/ . Descargue el archivo mingw-setup.exe, que instalará el editor de texto con un compilador.

Mi primera chamba...

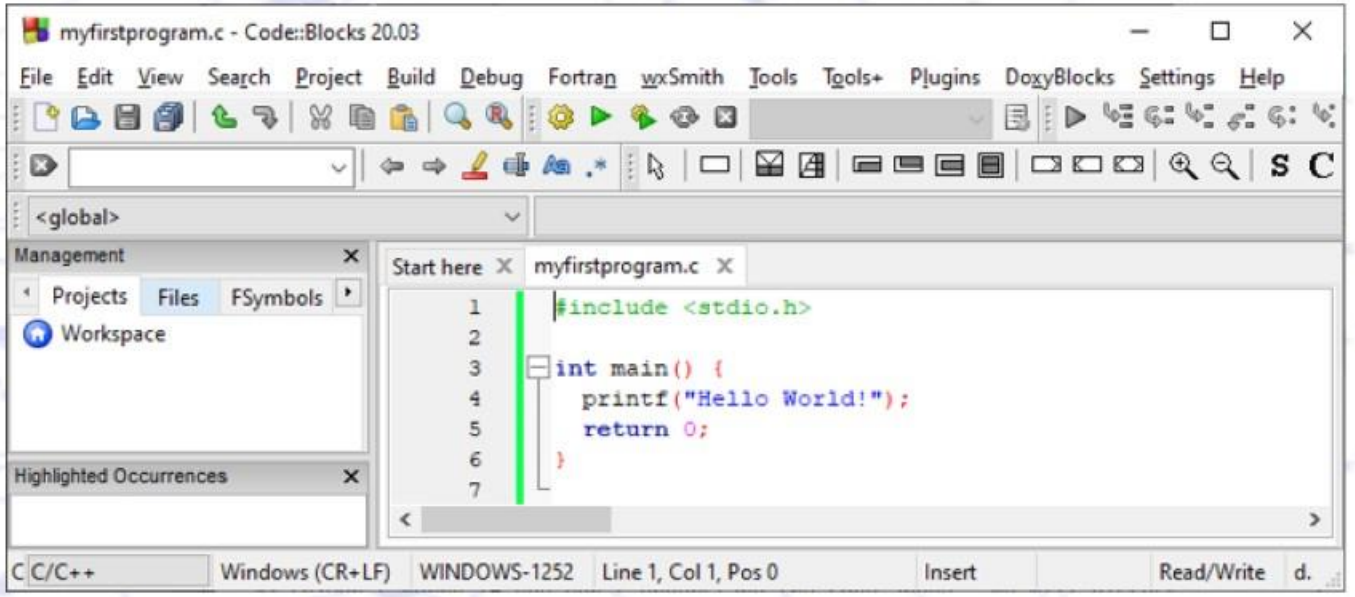
- Abra Codeblocks y vaya a **Archivo > Nuevo > Archivo vacío** .
- Escriba el siguiente código C y guarde el archivo como myfirstprogram.c (Archivo > Guardar archivo como):

```c
#include <stdio.h>

int main() {
  printf("Hello World!");
  return 0;
}
```

En Codeblocks debería verse así:

- No te preocupes si no entiendes el código anterior: lo analizaremos en detalle en capítulos posteriores. Por ahora, concéntrate en cómo ejecutar el código.



Luego, vaya a **Generar > Generar y ejecutar** para ejecutar el programa.
El resultado será similar a esto:

```
Hello World!
Process returned 0 (0x0) execution time : 0.011 s
Press any key to continue.
```

¡Felicitaciones ! Ya ha escrito y ejecutado su primer programa en C.

Sintaxis vs Semántica

- **Línea 1**: #include <stdio.h> es una biblioteca de archivos de encabezado que nos permite trabajar con funciones de entrada y salida, como printf()(usada en la línea 4). Los archivos de encabezado agregan funcionalidad a los programas C.
- **Línea 2:** una línea en blanco. C ignora los espacios en blanco, pero los usamos para que el código sea más legible.
- **Línea 3:** Otra cosa que siempre aparece en un programa en C es main(). Esto se llama función . Cualquier código dentro de sus llaves {}se ejecutará.
- **Línea 4:** printf() es una función que se utiliza para imprimir texto en la pantalla. En nuestro ejemplo, mostrará "¡Hola mundo!".
- **Línea 5**: return 0 finaliza la main()función.
- **Línea 6:** No olvide agregar la llave de cierre }para finalizar la función principal.
- **No te preocupes** si no entiendes cómo #include <stdio.h>funciona. Piensa en ello como algo que (casi) siempre aparece en tu programa.
- **Ten en cuenta que:** cada declaración C termina con un punto y coma;
- **Nota:** El cuerpo int main()también podría escribirse así:
  int main(){printf("Hello World!");return 0;}
- **Recuerde:** el compilador ignora los espacios en blanco. Sin embargo, si se usan varias líneas, el código resulta más legible.

```c
#include <stdio.h>

int main() {
  printf("Hello World!");
  return 0;
}
```

Declaraciones C

- Un **programa de computadora** es una lista de "instrucciones" que deben ser "ejecutadas" por una computadora.
- En un lenguaje de programación, estas instrucciones de programación se denominan **declaraciones** .
- La siguiente declaración "instruye" al compilador a imprimir el texto "Hola mundo" en la pantalla:

```
printf("Hello World!");
```

- Es importante que termines la declaración con punto y coma.;

- Si olvida el punto y coma (;), se producirá un error y el programa no se ejecutará:

```
printf("Hello World!")
```

```
error: expected ';' before
'return'
```

Salida (Texto impreso)

- Para generar valores o imprimir texto en C, puede utilizar la printf() función:
- Doble comillas
- Cuando se trabaja con texto, éste debe estar entre comillas dobles "".
- `printf("This sentence will work!");`
- Si olvida las comillas dobles, se produce un error:
- printf(This sentence will produce an error.);

```
"c:\Users\ITOS\projects\helloworld\"ho
la
Hello World!I am learning C.And it is
awesome!
[Done] exited with code=0 in 0.384
seconds
```

Muchas funciones printf:

- Puede utilizar tantas funciones printf() como desee. Sin embargo , tenga en cuenta que no inserta una nueva línea al final de la salida:

```
#include <stdio.h>

int main() {
  printf("Hello World!");
  printf("I am learning C.");
  printf("And it is awesome!");
  return 0;
}
```

Nuevas lineas

- Para insertar una nueva línea, puede utilizar los carácteres \n :

```c
#include <stdio.h>

int main() {
   printf("Hello World!\n");
   printf("I am learning C.");
   return 0;
}
```

varias líneas

- También puedes generar varias líneas con una sola función **printf().**
- Sin embargo, esto podría dificultar la lectura del código:

```c
#include <stdio.h>

int main() {
  printf("Hello World!\nI am learning C.\nAnd it is awesome!");
  return 0;
}
```

¿Qué es \nexactamente?

- Consejo: Dos \ncaracteres uno después del otro crearán una línea en blanco:

- El carácter de nueva línea (\n) se denomina secuencia de escape y obliga al cursor a cambiar su posición al principio de la siguiente línea en la pantalla.
- Esto da como resultado una nueva línea.

```c
#include <stdio.h>

int main() {
  printf("Hello World!\n\n");
  printf("I am learning C.");
  return 0;
}
```

Ejemplos de otras secuencias de escape válidas son:

Secuencia de escape	Descripción	Intentalo

\t	Crea una pestaña horizontal	`#include <stdio.h>`
		`int main() {`
		`  printf("Hello World!\t");`
		`  printf("I am learning C.");`
		`  return 0;`
		`}`

\\	Inserta un carácter de barra invertida (\)	`#include <stdio.h>`
		`int main() {`
		`  printf("Hello World!\\");`
		`  printf("I am learning C.");`
		`  return 0;`
		`}`

" "	Inserta un carácter de comillas dobles	`#include <stdio.h>`
		
		`int main() {`
		`  printf("They call him \"Johnny\".");`
		`  return 0;`
		`}`

Comentarios en C

- Los comentarios se pueden utilizar para explicar el código y hacerlo más legible. También se pueden utilizar para evitar la ejecución al probar código alternativo.
- Los comentarios pueden ser **de una sola línea** o **de varias líneas** .

Comentarios de una sola línea

- Los comentarios de una sola línea comienzan con dos barras diagonales (//).
- Cualquier texto entre //y el final de la línea es ignorado por el compilador (no se ejecutará).
- Este ejemplo utiliza un comentario de una sola línea antes de una línea de código:

```
// This is a comment
printf("Hello World!");
//Este ejemplo utiliza un comentario de una sola línea al final de una línea de código:
printf("Hello World!"); // This is a comment
```

C Comentarios de varias líneas

- **¿Comentarios de una o varias líneas?**
- Depende de usted cuál desea utilizar. Normalmente, lo utilizamos //para comentarios breves y /* */para comentarios más largos.
- Es bueno saber: antes de la versión C99 (lanzada en 1999), solo se podían usar comentarios de varias líneas en C.
- Los comentarios de varias líneas comienzan con /*y terminan con */.
- Cualquier texto entre /*y */será ignorado por el compilador:

```
/* The code below will print the words Hello World!
to the screen, and it is amazing */
printf("Hello World!");
```

Referencias

- http://www.codeblocks.org/
- https://github.com/DiegOliveros/Programaci-n-en-C