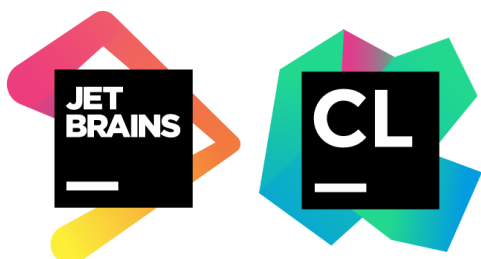


خودآموز آشنایی با محیط برنامه نویسی JetBrains' CLion

فهرست مطالب

1	خودآموز آشنایی با محیط برنامه نویسی JetBrains' CLion
1	مراحل نصب
7	اجرای (Run) اولین برنامه
9	رفع خطا (Debug)
12	ذخیره (Save)، بازیابی (Load) و بستن
12	تحقیق کنید! (اختیاری)



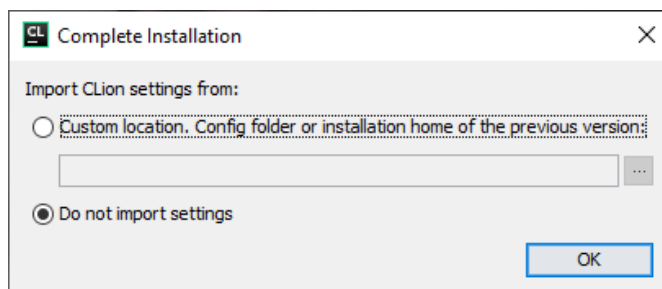
هدف این خودآموز آشنایی با محیط برنامه نویسی CLion از شرکت JetBrains است که در ادامه این درس به عنوان محیط استاندارد برنامه نویسی شما برای زبان C محسوب خواهد شد.

مراحل نصب

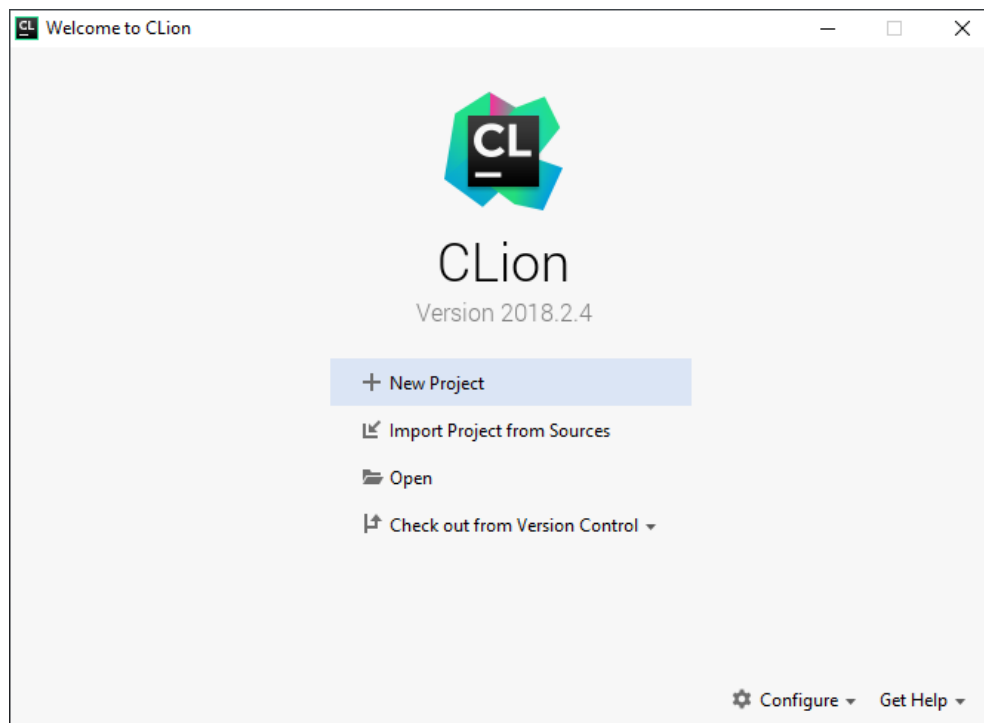
ابتدا محیط برنامه نویسی را نصب نمایید. می توانید این نرم افزار را از طریق لینک زیر دانلود کنید.

<https://www.jetbrains.com/clion/download/>

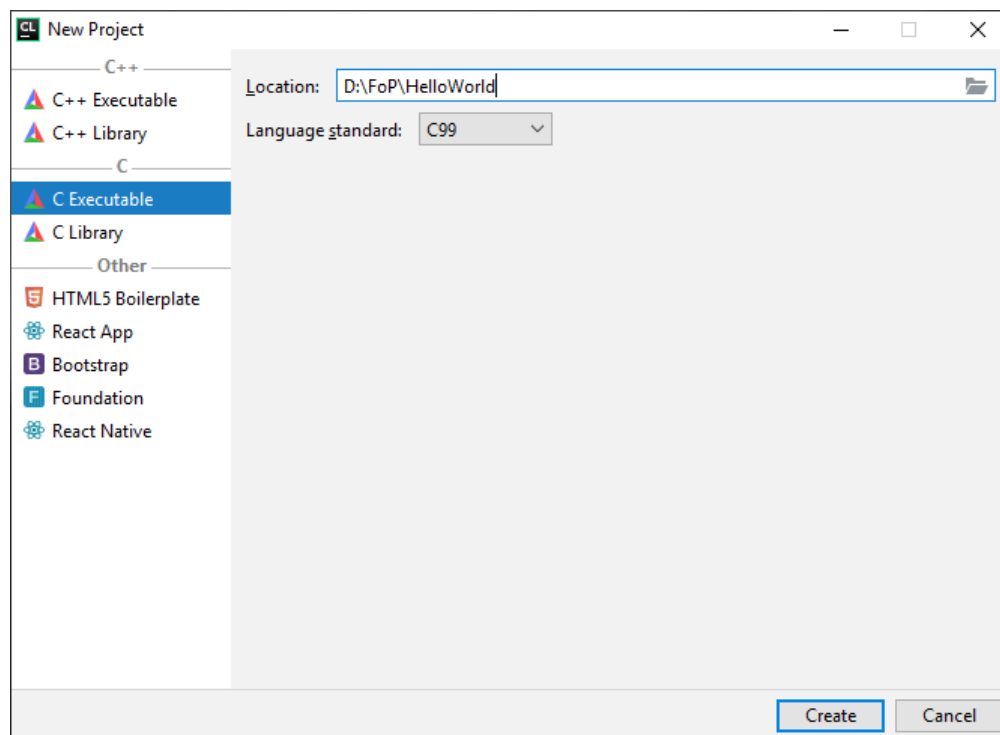
پس از نصب برنامه صفحه زیر ظاهر می شود. اگر این برنامه را برای بار اول نصب می کنید و نیازی به اضافه کردن تنظیمات نسخه قبلی را نداشتید از گزینه Do not import settings استفاده کنید. اما در صورتی که این نرم افزار را قبلا نصب کرده بودید و خواستید از تنظیمات همان نسخه استفاده کنید، از گزینه Custom location... استفاده کنید.



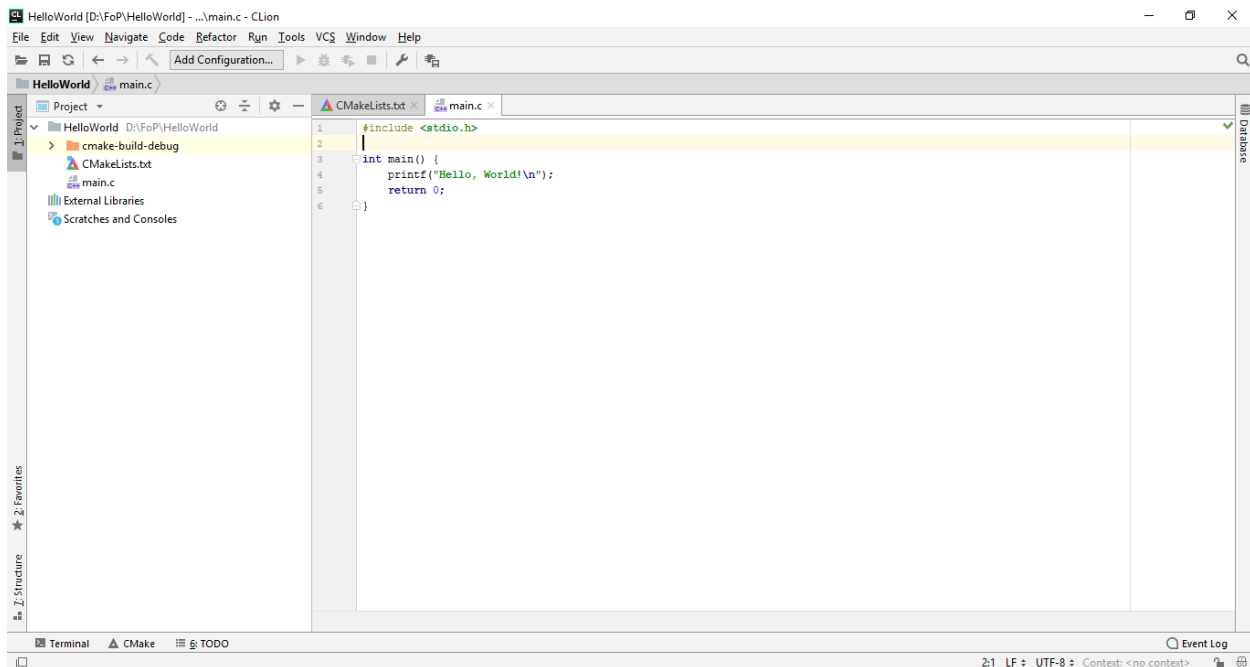
پس از این مرحله صفحه دیگری ظاهر می شود و می توانید تنظیمات دلخواه خود را انتخاب کنید. در ادامه می توانید در صفحه زیر پروژه جدید بسازید. برای این کار روی گزینه New Project کلیک کنید تا مراحل ایجاد پروژه جدید نمایش داده شود.



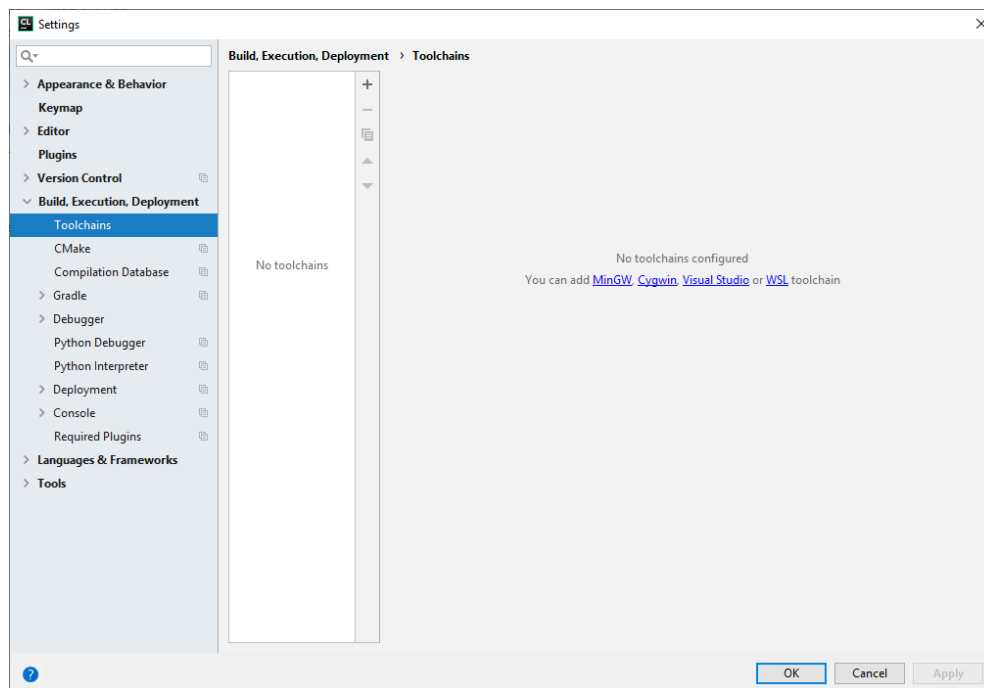
با کلیک روی گزینه New Project صفحه زیر ظاهر می شود. از قسمت سمت چپ C Executable را انتخاب کنید سپس در قسمت Location آدرس پروژه را تعیین کنید و با انتخاب گزینه Create پروژه را بسازید.



پنجره جدیدی به شکل زیر ایجاد می شود که می توان فرایندهای برنامه نویسی را در بخش های مختلف آن انجام داد. از این پس می توانید فرایند ایجاد پروژه را از مسیر **File | New Project...** انجام دهید. می توانید مسیر **View | Toolbar** را فعال کنید تا امکانات بیشتری مشاهده کنید.



در ابتدا وارد مسیر **File | Settings | Build, Execution, Deployment | Toolchains** شوید.



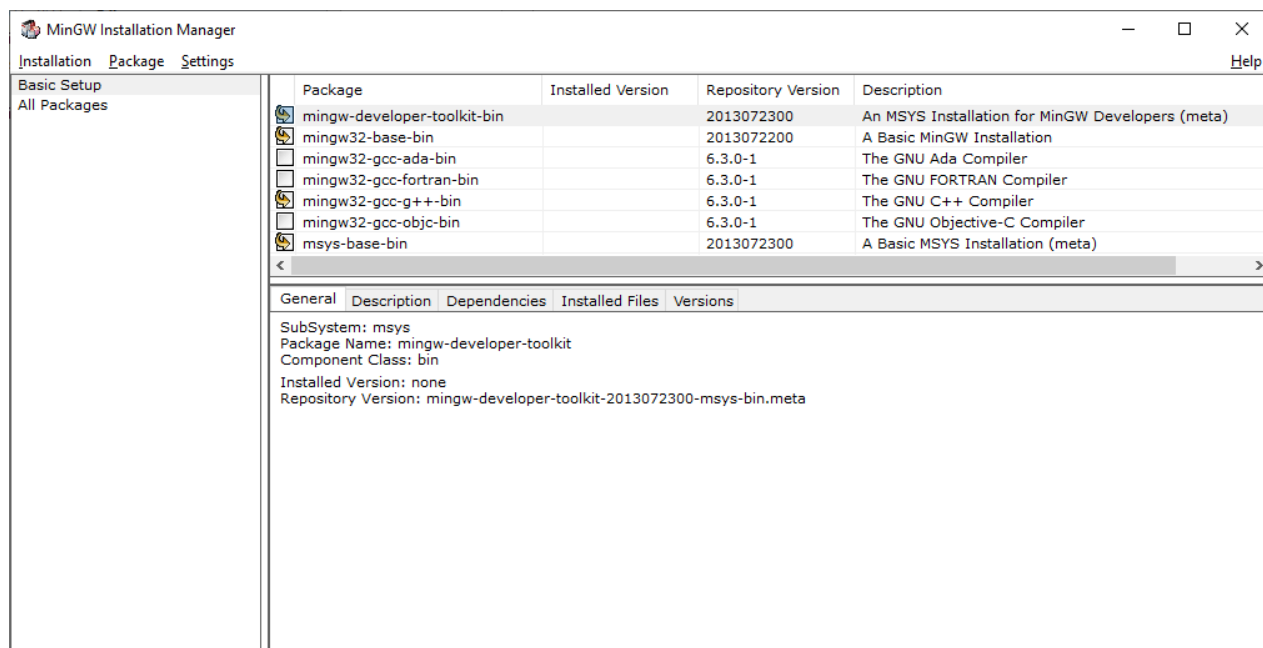
در صورتی که لیست خالی بود باید یک محیط اضافه کنید. برای این کار یک محیط MinGW اضافه خواهیم کرد. ابتدا به سایت <http://mingw.org> مراجعه کنید و از قسمت دانلود نرم افزار نصب را دانلود کنید. برای دانلود به لینک زیر خواهید رفت.

<https://osdn.net/projects/mingw/releases/>

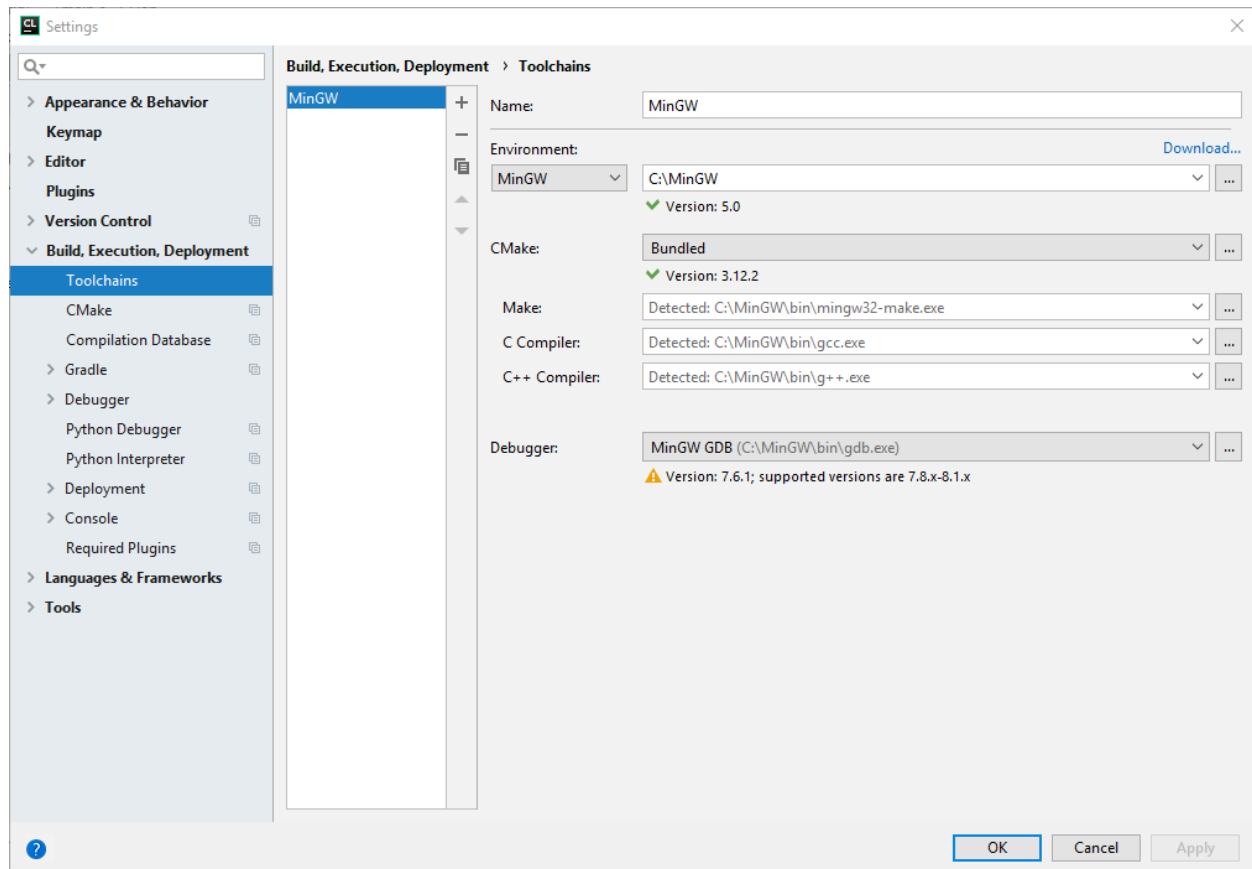
حال برنامه را نصب و اجرا کنید. با اجرای آن پنجره زیر نمایش داده می شود. موارد زیر را فعال کنید.

- mingw-developer-toolkit-bin
- msys-base-bin
- mingw32-base-bin
- mingw32-gcc-g++-bin

حال مسیر Installation | Apply Changes را اجرا کنید. با اجرای این مرحله محیط داندلود و نصب می شود.



حال دوباره به مسیر File | Settings | Build, Execution, Deployment | Toolchains بروید. با کلیک روی علامت مثبت یک محیط اضافه کنید. مورد MinGW را به عنوان نوع محیط انتخاب کنید. مسیر نصب را انتخاب کنید و منتظر بمانید تا همه موارد تایید شوند.




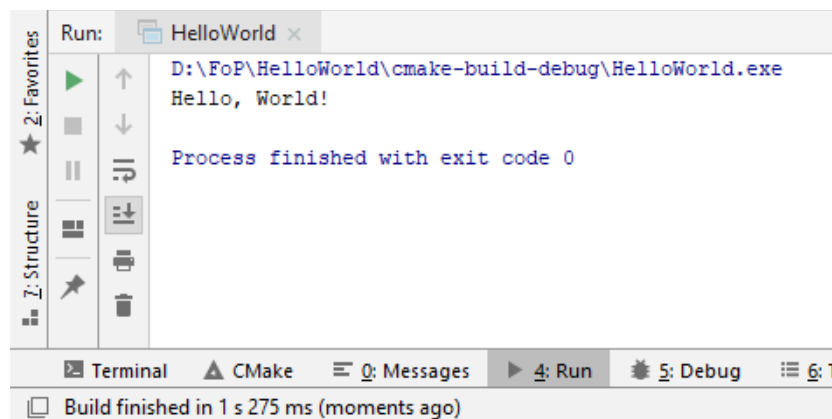
با مشاهده پیام های بالا محیط شما آماده می شود. گزینه OK را انتخاب و به صفحه اصلی بازگردید.

اجرای (Run) اولین برنامه

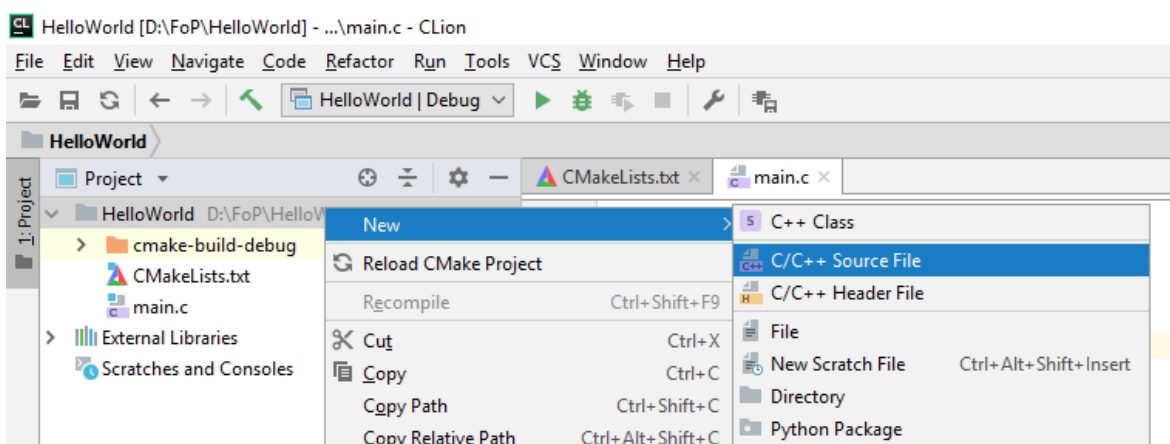
اولین برنامه ای که به اجرای آن خواهیم پرداخت Hello World نام دارد. با ایجاد پروژه این برنامه به صورت پیش فرض وجود دارد.

```
1      #include <stdio.h>
2
3      int main() {
4          printf("Hello, World!\n");
5          return 0;
6      }
```

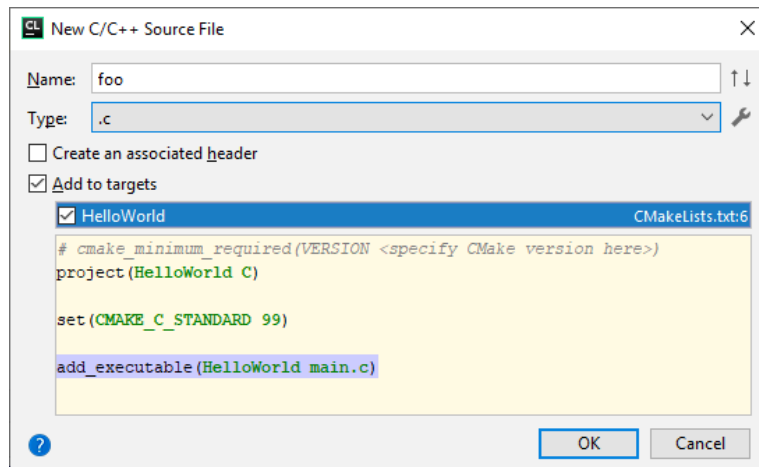
ابا کلیک روی دکمه  برنامه را Compile و اجرا کنید و در نتیجه باید بتوانید پنجره زیر را مشاهده کنید. می توانید از کلید میانبر Shift + F10 برای اجرای برنامه استفاده کنید.



برای ساختن فایل های بیشتر می توانید در پنجره Project روی پوشه مورد نظر کلیک راست کرده و مانند شکل زیر مسیر New | C/C++ Source File را انتخاب کنید.



در پنجره جدیدی که باز می شود، نام مناسبی برای فایل اختصاص دهید و دقت کنید که برای گزینه Type حتما مورد c را انتخاب کنید.

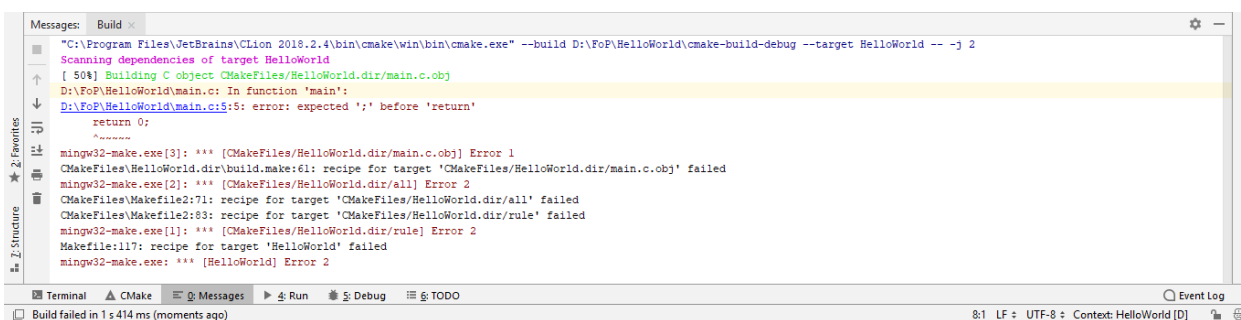


رفع خطا (Debug)

خطاهای برنامه نویسی موارد معمولی هستند که در فرایند برنامه نویسی رخ می دهند. روش های مختلفی برای آن ها وجود دارد. تصویر زیر نمونه ای از یک کد با خطای نحوی (Syntax Error) است.

```
1      #include <stdio.h>
2
3      int main() {
4          printf("Hello, World!\n");
5          return 0;
6      }
```

همانطور که مشخص است خطای موجود در این خط با خط چین قرمز مشخص شده است. با کلیک روی برنامه را می سازیم (Build می کنیم). پیام زیر نمایش داده می شود و به این معنی است که برنامه build نشده است. برنامه همواره باید قبل از اجرا build شود و نرم افزار CLion این کار را برایمان انجام می دهد. دلیل استفاده از این دکمه در اینجا نمایش تفاوت خطای بالا و خطای بعدی است.



در این پیام خطی که خطا در آن رخ داده است و نوع خطایی که رخ داده است نمایش داده شده است. با رفع خطا می توان دوباره به ساخت برنامه پرداخت. اما کد زیر را در نظر بگیرید.

```
1      #include <stdio.h>
2
3      int main() {
4          int a = 1;
5          int b = 0;
6          int c = 0;
7          c = a / b;
8          printf("%d\n", c);
9          return 0;
10     }
```


این برنامه به ظاهر مشکلی ندارد و وقتی پروژه را می سازیم و پروژه با موفقیت ساخته می شود. اما وقتی برنامه را اجرا می کنیم، برنامه با موفقیت اجرا نمی شود. اگر برنامه با موفقیت اجرا شود باید مقدار c در خروجی چاپ شود ولی این مقدار در خروجی نمایش داده نمی شود. (نشانه دیگری نیز برای عدم اجرای موفق برنامه پیدا کنید!)



این دست از خطاها ممکن است همواره رخ ندهند! برای مثال اگر مقدار b از کاربر دریافت شود ممکن است بعضی وقت ها کاربر مقدار صفر و بعضی وقت ها مقداری غیر از صفر وارد کند. به همین دلیل پیدا کردن آن ها قدری سخت تر است. یک راه برای شروع رفع خطای این دست پرینت کردن مقادیر است تا خطاهایی را که ممکن است رخ دهد پیدا کنیم. اما راه آسان تر برای این کار استفاده از امکان debugging نرم افزار CLion است. سمت راست شماره خط در خط $\text{int } a = 1;$ کلیک کنید تا یک Breakpoint ساخته شود و به شکل زیر تغییر یابد.

```

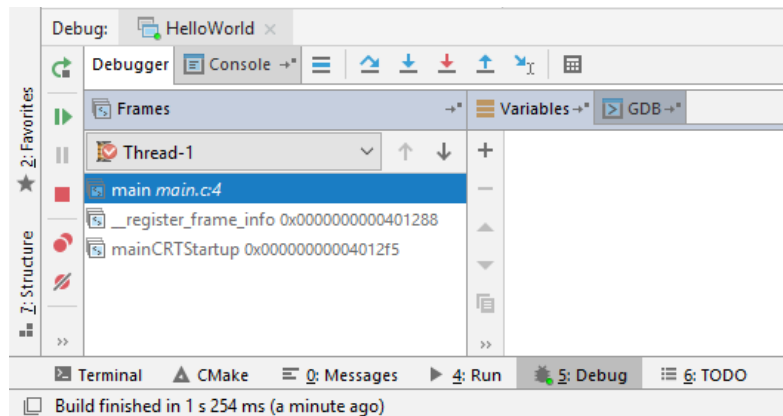
1      #include <stdio.h>
2
3      int main() {
4          int a = 1;
5          int b = 0;
6          int c = 0;
7          c = a / b;
8          printf("%d\n", c);
9          return 0;
10     }
```


حال با کلیک روی دکمه  برنامه را در حالت debug اجرا می کنیم.

```

1      #include <stdio.h>
2
3      int main() {
4          int a = 1;
5          int b = 0;
6          int c = 0;
7          c = a / b;
8          printf("%d\n", c);
9          return 0;
10     }
```

خط آبی نشان دهنده خطی که برنامه در حال حاضر تا آن اجرا شده است و خط بعدی که اجرا می شود است. پنجره زیر پنجره Debug است که دارای امکاناتی است که برای debug کردن مفید هستند.

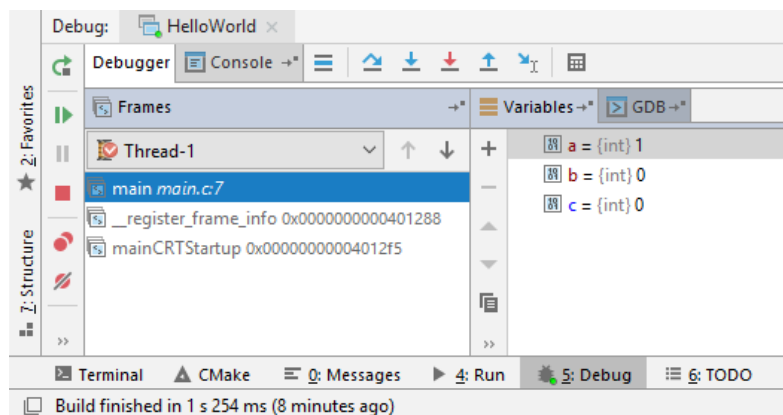


با کلیک روی دکمه  که Step Over نام دارد می توان خط مشخص شده را اجرا کرد. با دکمه F8 نیز می توان این کار را انجام داد. دو بار دیگر نیز F8 را می زنیم تا برنامه به حالت زیر برود (دکمه های دیگر پس از یادگیری توابع (function) استفاده خواهند شد).

```

1      #include <stdio.h>
2
3      int main() {
4          int a = 1; a: 1
5          int b = 0; b: 0
6          int c = 0; c: 0
7          c = a / b;
8          printf("%d\n", c);
9          return 0;
10     }
```

با رسیدن به این خط پنجره Debug به حالت زیر می رسد.



در قسمت Variables متغیرهای تعریف شده و مقادیر فعلیشان نمایش داده می شود. می توان با استفاده از کلید جمع متغیرهای دیگر یا ترکیبی از آن ها را اضافه کرد. حال مشخص است که در خط بعدی باید a / b محاسبه شود و از طرفی با توجه به قسمت Variables مقدار b برابر صفر است؛ پس تقسیم بر صفر رخ می دهد که این تقسیم تعریف نشده است و باعث می شود اجرای برنامه با مشکل روبه رو شود. حال با تغییر مقدار a به ۴ و b به ۲ برنامه را دوباره اجرا می کنیم و نتیجه را مشاهده می کنیم.

ذخیره (Save)، بازیابی (Load) و بستن

با مسیر File | Save All یا کلید ترکیبی $Ctrl + S$ می توانید تغییرات خود را ذخیره کنید. با مسیر File | Open... می توانید پروژه دیگری را باز کنید. با مسیر File | Open Recent می توانید پروژه های قبلی خود را باز کنید. در نهایت می توانید با مسیر File | Close Project به صفحه اول برگردید یا با File | Exit نرم افزار CLion را ببندید.

تحقیق کنید! (اختیاری)

- در رابطه با کلمات زیر تحقیق کنید.
 - IDE
 - Compiler
 - Editor
 - Debugger
- در رابطه با کامپایل، ویرایش و دیباگ در محیط لینوکس تحقیق کنید. از کلیدواژه های زیر نیز استفاده کنید.
 - gcc
 - vim
 - gdb
- شاید برایتان جالب باشد که بدانید کلمه bug از کجا آمده است و چرا موقع رفع ایراد های برنامه از debugger (سوسک کش!) استفاده می کنیم.

موفق باشید