

Level Up Coding · [Follow publication](#)

This member-only story is on us. [Upgrade](#) to access all of Medium.

★ Member-only story

Running C/C++ Code in Google Colab

3 min read · Mar 12, 2025



Parth Mahakal

[Follow](#)

▶ Listen

Share

More



Image from canva

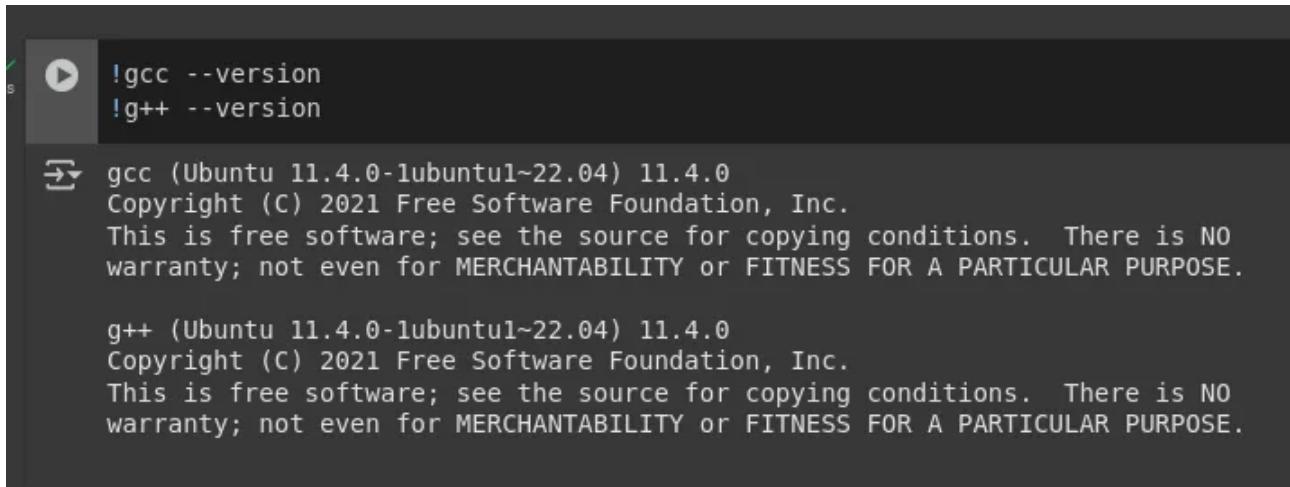
Absolutely, it is possible to set up and run C/C++ code in Google Colab, and it can be a great option for many users! I recently worked on a C++ project and initially tried setting up my environment in VS Code. However, I encountered several errors and configuration issues. While many people have no trouble with local setups for C and C++, there are times when you only need a quick, temporary environment to test or run your code. In such cases, Google

Colab proves to be an excellent choice. It provides a hassle-free, cloud-based solution that eliminates the need for complex local configurations, making it ideal for short-term coding tasks or experimentation.

Step 1: Create a New Colab Notebook

1. Go to [Google Colab](#).
2. Click on File > New Notebook to create a new notebook.

Step 2: Check if GCC/G++ is Installed



```
!gcc --version
!g++ --version

→ gcc (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0
Copyright (C) 2021 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

g++ (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0
Copyright (C) 2021 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Image by Author

Google Colab already comes with gcc and g++ pre-installed, but let's verify it.

1. Create a new code cell.
2. Run the following commands to check the versions of gcc and g++:

```
!gcc --version
!g++ --version
```

- The ! symbol allows you to run shell commands in Colab.
- If these commands output the version numbers (e.g., gcc (Ubuntu 9.4.0-1ubuntu1~20.04.2) 9.4.0), then gcc and g++ are installed.

Step 3: Write Your C or C++ Code

You cannot directly execute C/C++ code in a Python cell, so you need to write your code into a file (e.g., .c for C or .cpp for C++) and then compile and run it using shell commands.

For C Code

1. Create a new code cell and write your C code into a file using the %%writefile magic command. For example:

```
%%writefile hello_c.c
#include <stdio.h>
int main() {
    printf("Hello, C in Google Colab!\n");
    return 0;
}
```

This will save the code into a file named hello_c.c.

For C++ Code

1. Similarly, for C++ code, use the .cpp extension:

```
%%writefile hello_cpp.cpp
#include <iostream>
using namespace std;
int main() {
    cout << "Hello, C++ in Google Colab!" << endl;
    return 0;
}
```

This will save the code into a file named hello_cpp.cpp.

Step 4: Compile the Code

Now you need to compile the code using gcc for C or g++ for C++.

For C Code

1. Create a new code cell and compile the C code using gcc:

```
!gcc hello_c.c -o hello_c
```

- hello.c is the source file.
- -o hello specifies the output executable name (hello).

For C++ Code

1. Create a new code cell and compile the C++ code using g++:

```
!g++ hello_cpp.cpp -o hello_cpp
```

- `hello_cpp.cpp` is the source file.
- `-o hello` specifies the output executable name (`hello_cpp`).

If there are no errors, the compilation will succeed, and an executable file (`hello`) will be created.

Step 5: Run the Executable

After compilation, you can run the executable file.

1. Create a new code cell and run the compiled program:

```
!./hello_c
```

- `!` allows you to execute shell commands.
- `./hello_c` runs the compiled executable. and same for the run c++ file like `./hello_cpp`.

Notes

- Google Colab's environment is temporary. Files are deleted after the session ends, so save your code if needed.
- Use `%%writefile` to overwrite files and avoid errors if the file already exists.
- For complex projects, consider using a local IDE or cloud-based C/C++ environment like Replit or an online compiler.

That's it! You now have a fully functional C and C++ setup in Google Colab. Let me know if you encounter any issues!

[Google Colab](#)[C](#)[Cpp](#)[Follow](#)

Published in Level Up Coding

254K followers · Last published 13 hours ago

Coding tutorials and news. The developer homepage gitconnected.com && skilled.dev && levelup.dev



Follow

Written by Parth Mahakal

38 followers · 7 following

[Open in app](#) ↗

Medium



Search



R

What are your thoughts?

More from Parth Mahakal and Level Up Coding



In Data Engineer Things by Parth Mahakal

Open-Source Model for Your Text to Video Generation Project

Here I find out the best open-source model for your text-to-video generation project in this article I present the best six open-source...

Jan 8



...



In Level Up Coding by Michele Ronan

Why OpenAI Suddenly Erased Jony Ive from their Website

A billion-dollar collaboration... quietly wiped out overnight. Here's what happened.

Jun 25 2.1K 70



...



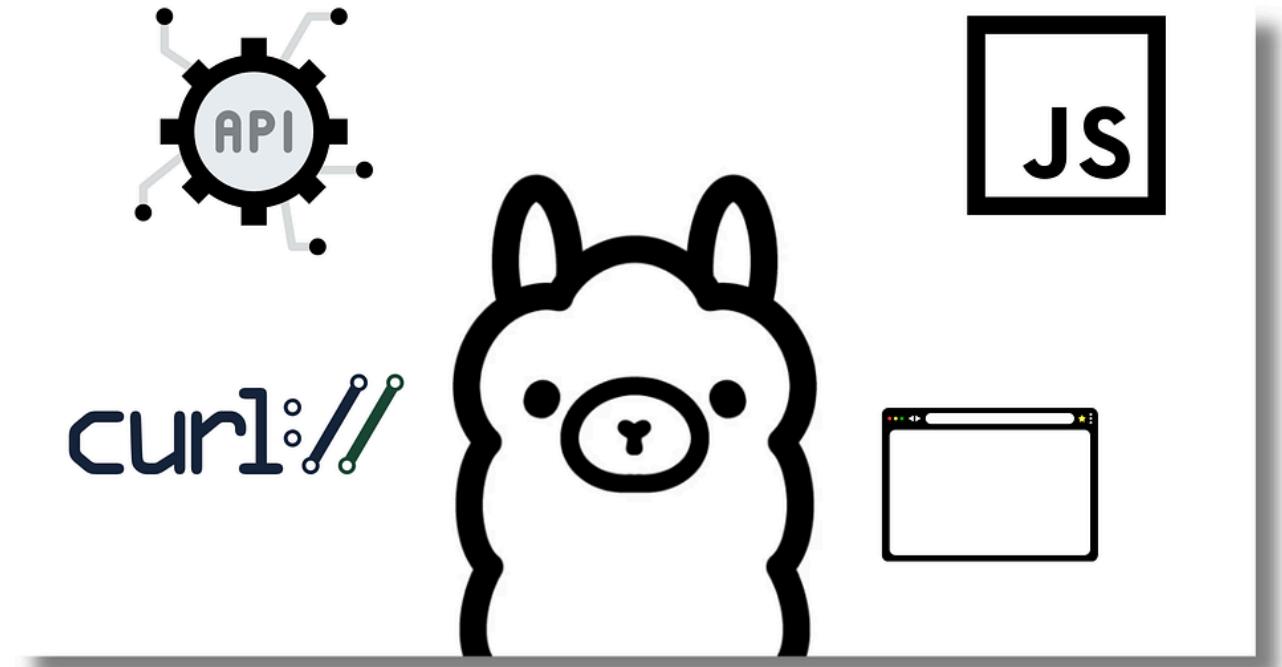
In Level Up Coding by Michele Ronan

Did Apple Just Fire Their Liquid Design Team Lead

And lost \$75 billion in Market Value as the Stock fell 2.5% during WWDC Live Keynote

Jun 11 1.2K 57

W+ ...



In Level Up Coding by Parth Mahakal

Calling the Ollama API in a Local Browser Using JavaScript

Using the Ollama API, create webpage with better user interfaces. To demonstrate this, we'll create a simple web...

Nov 3, 2024 3 1

W+ ...

See all from Parth Mahakal

See all from Level Up Coding

Recommended from Medium



 In DevOps.dev by Syed Hamed Raza

Mastering Pointers in C++

A Comprehensive Guide

Mar 15

5



...



Simple Directmedia Layer

 David Delassus

Trying out SDL3 by writing a C++ Game Engine

SDL3 is around the corner! SDL3_image has been released, and we're waiting for SDL3_ttf, SDL3_mixer and SDL3_net to be released soon as...

 Feb 27  20  1





 Jose Garcia

Install Neovim on Debian-based

A guide on the best ways to install Neovim on Debian-based distros

Apr 28  1



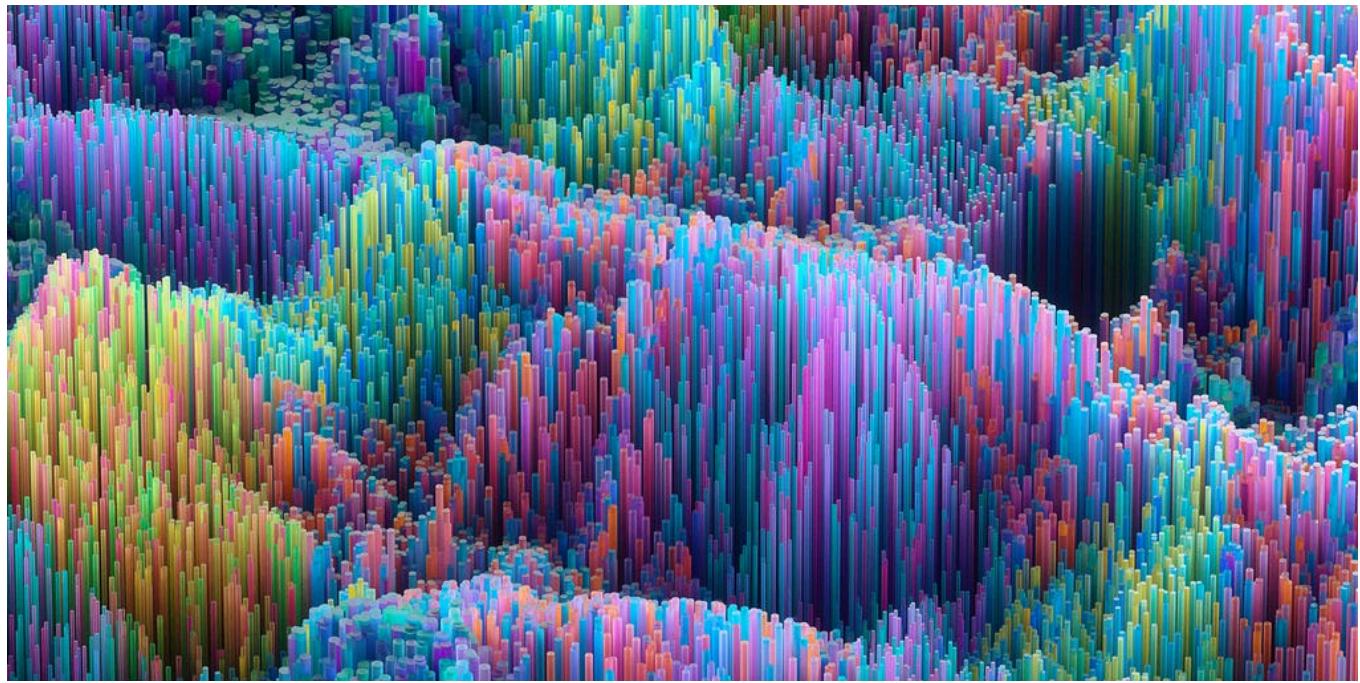
 ThreadSafe Diaries

Go Was My First Love. Zig Made Me Feel Unsafe Again—In a Good Way

I fell hard for Go in 2016.

 Jun 16  5



In ImageCraft by Francisco Zavala

Image histogram analysis

Histogram Calculation Explained: Counting Pixel Frequencies Across channel Levels.

Mar 13 2



...



 Manyi

Running LLMs Locally on Mac with Apple's MLX

The ability to run Large Language Models (LLMs) locally on personal hardware has become an interest for developers, researchers, and...

 May 25  10

[See more recommendations](#)