```
// Michigan Robotics 102
// Introduction to AI and Programming

// Making our first program in C++:
//    code, compile, run, repeat

#include <iostream>
int main()
{

   std::cout << "Hello World!";

}
```

A simple program in C++

```
#include <iostream>
int main()
{
   std::cout << "Hello World!";
}
```

A simple program in C++

What is C++?

```
#include <iostream>
int main()
{
    std::cout << "Hello World!";
}
```

What is C++?

C++

From Wikipedia, the free encyclopedia

"CXX" redirects here. For other uses, see CXX (dis

C++ (/ˌsiːˈplʌsˈplʌs/) is a general-purpose programming language created by Bjarne Stroustrup as an extension of the C programming language, or "C with Classes". The language has expanded significantly over time, and modern C++ now has object-oriented, generic, and functional features in addition to facilities for low-level memory manipulation. It is almost always implemented as a compiled language, and many vendors provide C++ compilers, including the Free Software Foundation, LLVM, Microsoft, Intel, Oracle, and IBM, so it is available on many platforms.[9]

C++ was designed with an orientation toward system programming and embedded, resource-constrained software and large systems, with performance, efficiency, and flexibility of use as its design highlights.[10] C++ has also been found useful in many other contexts, with key strengths being software infrastructure and resource-constrained applications,[10] including desktop applications, video games, servers (e.g. e-commerce, web search, or databases), and performance-critical applications (e.g. telephone switches or space probes).[11]

C++ is standardized by the International Organization for Standardization (ISO), with the latest standard version ratified and published by ISO in December 2020 as *ISO/IEC 14882:2020* (informally known as C++20).[12] The C++ programming language was initially standardized in 1998 as *ISO/IEC 14882:1998*, which was then amended by the C++03, C++11, C++14, and C++17 standards. The current C++20 standard supersedes these with new features and an enlarged standard library. Before the initial standardization in 1998, C++ was developed by Danish computer scientist Bjarne Stroustrup at Bell Labs since 1979 as an extension of the C language; he wanted an efficient and flexible language similar to C that also provided high-level features for program organization.[13] Since 2012, C++ has been on a three-year release schedule[14] with C++23 as the next planned standard.[15]

Contents [hide]

C++

The C++ logo endorsed by Standard C++

Paradigms	Multi-paradigm: procedural, functional, object-oriented, generic, modular
Family	C
Designed by	Bjarne Stroustrup
Developer	ISO/IEC JTC1 (Joint Technical Committee 1) / SC22 (Subcommittee 22) / WG21 (Working Group 21)
First appeared	1985; 36 years ago
Stable release	C++20 (ISO/IEC 14882:2020) / 15 December 2020; 8 months ago
Preview release	C++23 / 18 June 2021; 2 months ago

What is C++?

High-level programming language

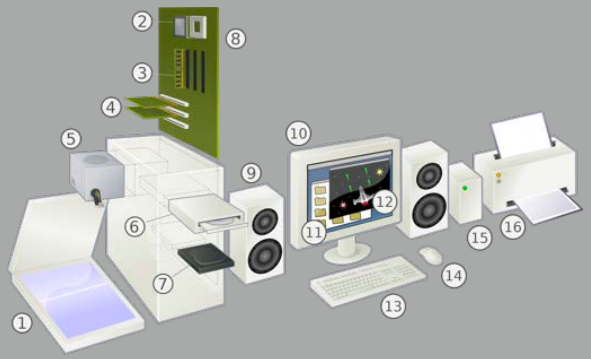
What is C++?

High-level programming language

Computing Hardware

What is C++?

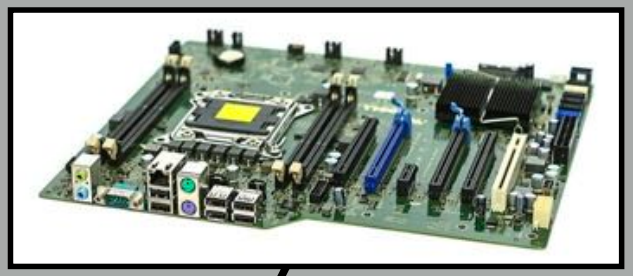
High-level programming language



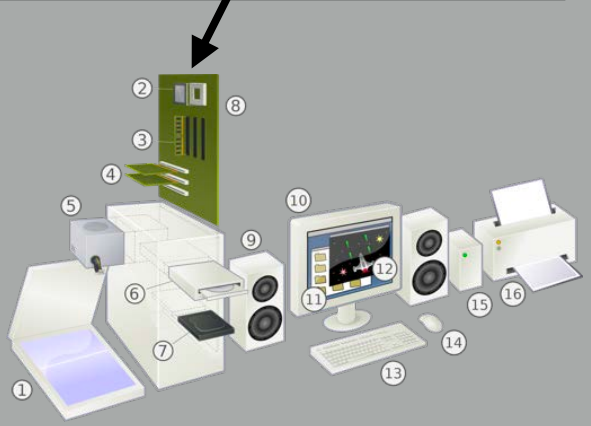
Computing Hardware

What is C++?

High-level programming language



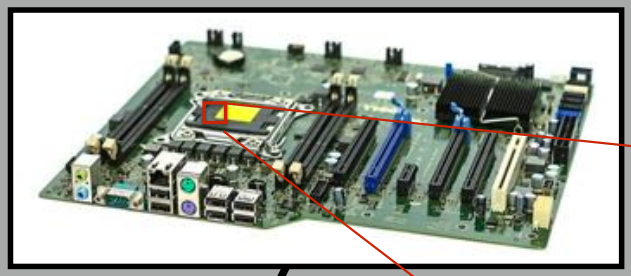
Main circuit board
or "***Motherboard***"



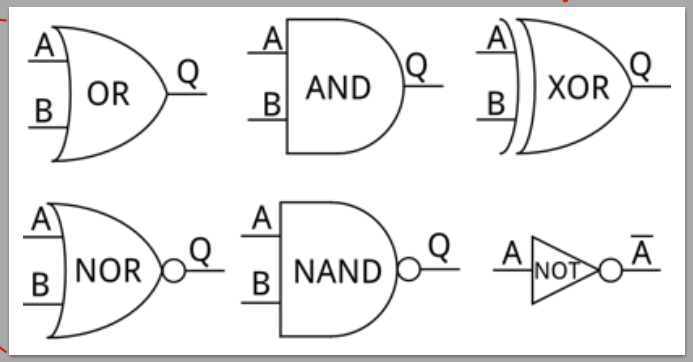
Computing Hardware

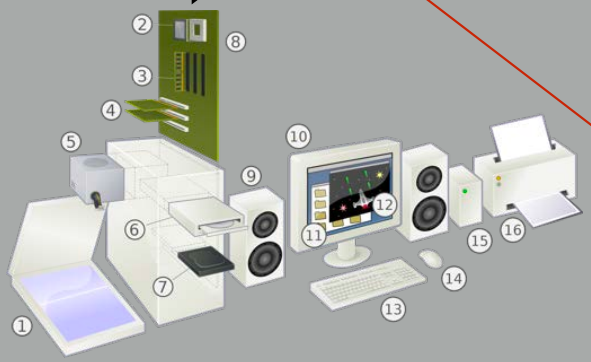
What is C++?

High-level programming language



Electronics that can carry out digital logic

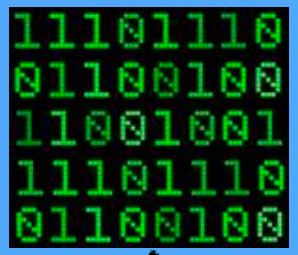




Computing Hardware

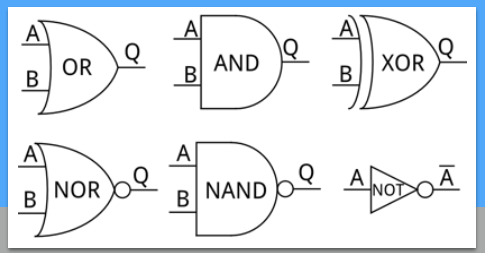
What is C++?

High-level programming language

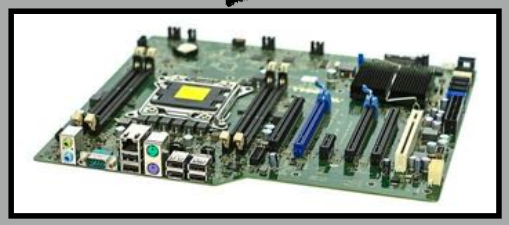


A program in binary code

Executable Software



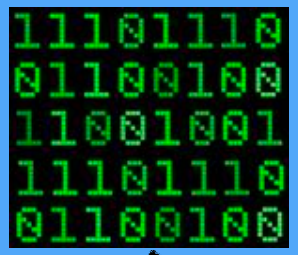
Electronics that can carry out digital logic



Computing Hardware

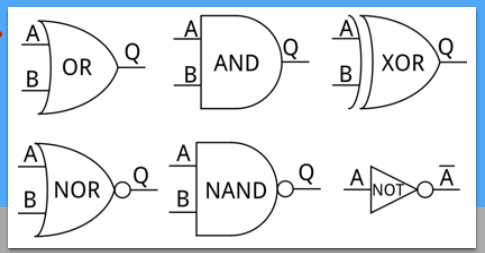
What is C++?

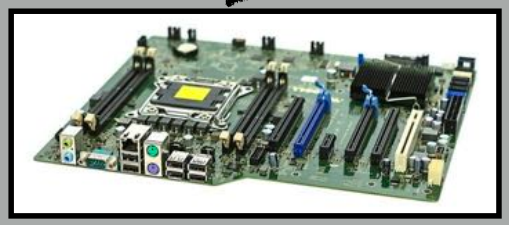
High-level programming language



A program in binary code specifying a sequence of digital logic instructions that run on a computer

Executable Software





Computing Hardware

Electronics that can carry out digital logic

What is C++?

High-level programming language

```
#include <iostream>
int main()
{
   std::cout << "Hello World!";
}
```

Source Code

A program in readable text that can be compiled into a software executable

Executable Software

A program in binary code specifying a sequence of digital logic instructions that run on a computer

Computing Hardware

Electronics that can carry out digital logic

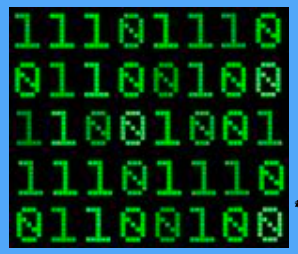
What is C++?

Hig *"Coding" is writing source code*

```
#include <iostream>
int main()
{
   std::cout << "Hello World!";
}
```

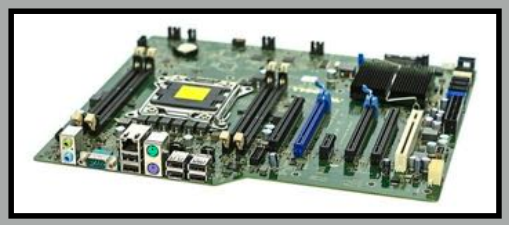
Source Code

A program in readable text that can be compiled into a software executable



Executable Software

A program in binary code specifying a sequence of digital logic instructions that run on a computer



Computing Hardware

Electronics that can carry out digital logic

What is C++?

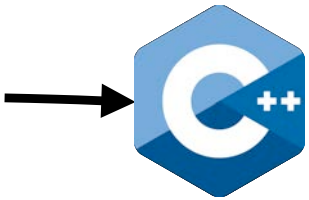
High-level programming language

Compiles source files into executable apps

Source Code in C++ Language

```
#include <iostream>
int main()
{
   std::cout << "Hello World!";
}
```

C++ Compiler



Executable

What is C++?

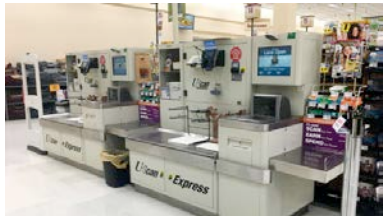
High-level programming language

Portable across nearly all computers

Source Code in C++ Language

```
#include <iostream>
int main()
{
    std::cout << "Hello World!";
}
```

C++ Compiler



What is C++?

High-level programming language

Portable across nearly all computers

Compiler

Source Code in C++ Language

```
#include <iostream>
int main()
{
    std::cout << "Hello World!";
}
```

What is C++?

High-level programming language

Portable across nearly all computers

Source Code in C++ Language

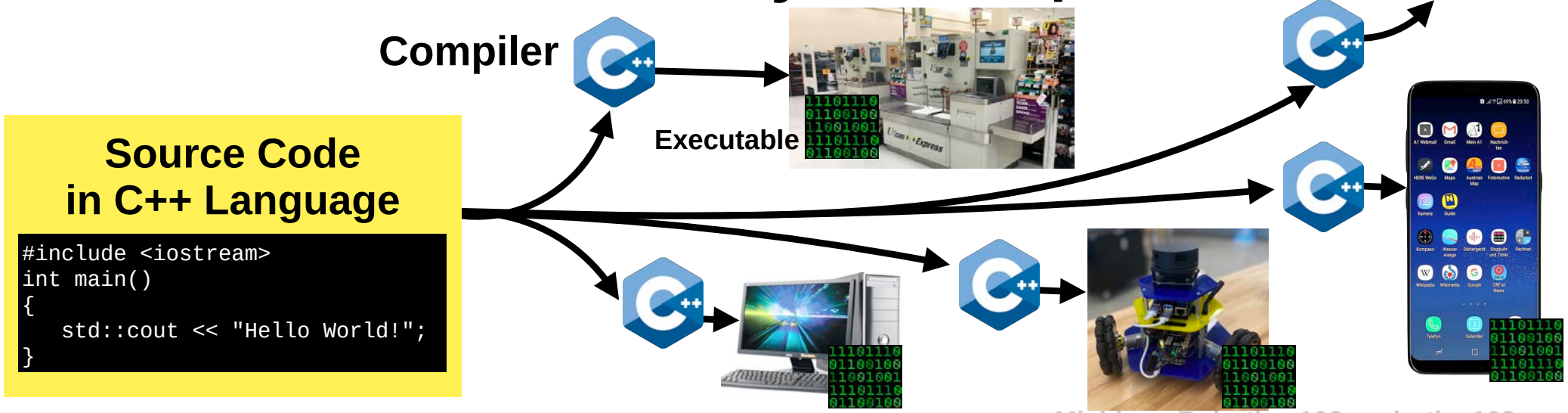
```
#include <iostream>
int main()
{
    std::cout << "Hello World!";
}
```

Compiler

What is C++?

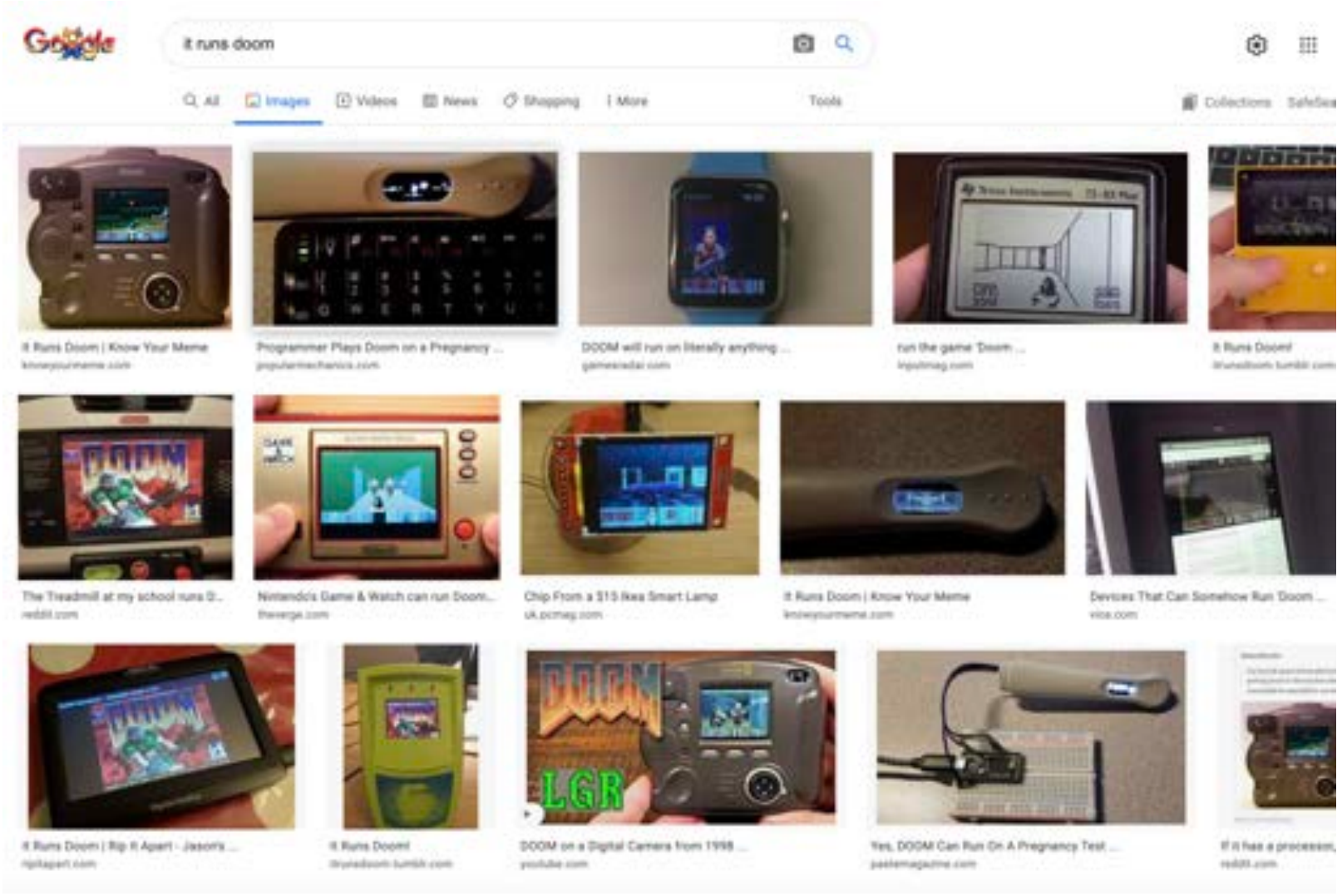
High-level programming language

Portable across nearly all computers



Software Portability

It runs Doom!


it runs doom
All
Images
Videos
News
Shopping
More
Tools
Collections
It Runs Doom | Know Your Meme
Programmer Plays Doom on a Pregnancy ...
DOOM will run on literally anything ...
run the game 'Doom' ...
It Runs Doom!
The Treadmill at my school runs D...
Nintendo's Game & Watch can run Doom...
Chip From a $15 Ikea Smart Lamp
It Runs Doom | Know Your Meme
Devices That Can Somehow Run 'Doom' ...
It Runs Doom | Rip It Apart! - Jason's ...
It Runs Doom!
DOOM on a Digital Camera from 1998 ...
Yes, DOOM Can Run On A Pregnancy Test ...
If it has a processor,

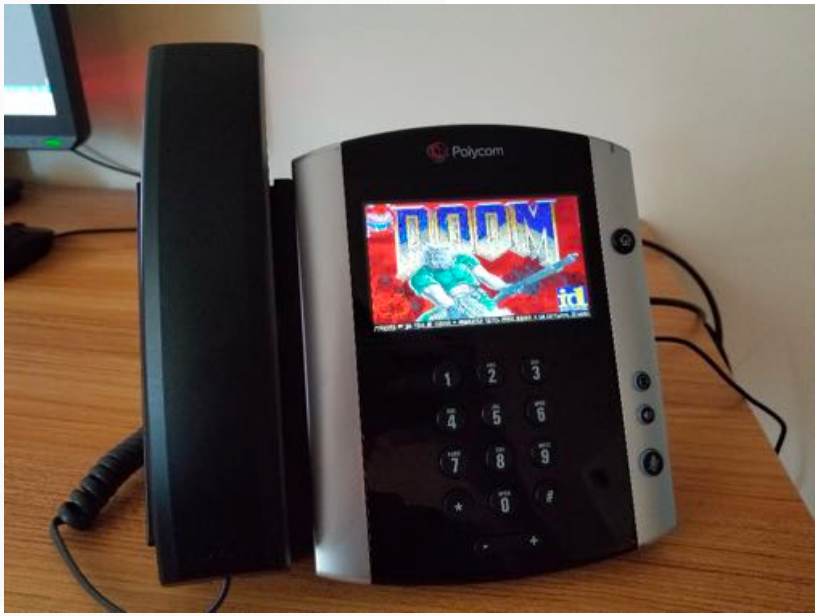

it runs doom
DOOM
id
FHKE
19:19
Start
OPEN
Kodak
GAME BOY ADVANCE SP
TEXAS INSTRUMENTS TI-83 Plus
DOOM
Continue
New Game
> Options
Quit
v0.12

neilbostian.github.io/doomphone

Neil's Place / DoomPhone

Technical Review: How I hacked an office telephone to play DOOM

Published 2021-09-01



Contents

Background

In late 2017 I was a new hire on a small (maybe 10 person) IT team. We generally had an option to take hardware for ourselves if

What is C++?

High-level programming language

Portable across nearly all computers

Compiles source files into executable apps

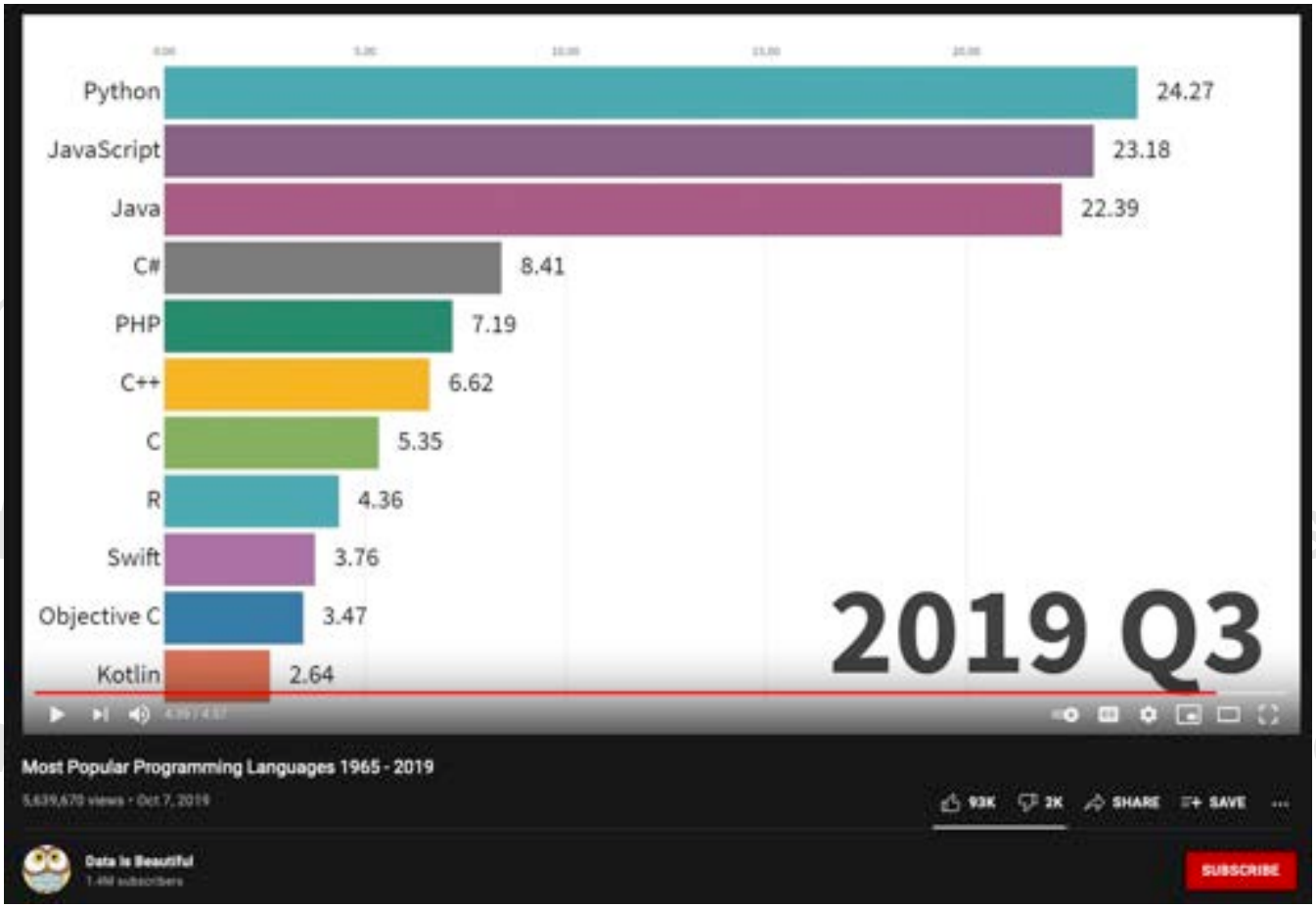
What is C++?

High-level programming language

Portable across nearly all computers

Compiles source files into executable apps

One of the most popular languages



One of the most popular languages

You code in C++.
You can learn any of these.

One of the most popular languages

A simple program in C++

How to code in C++ ?

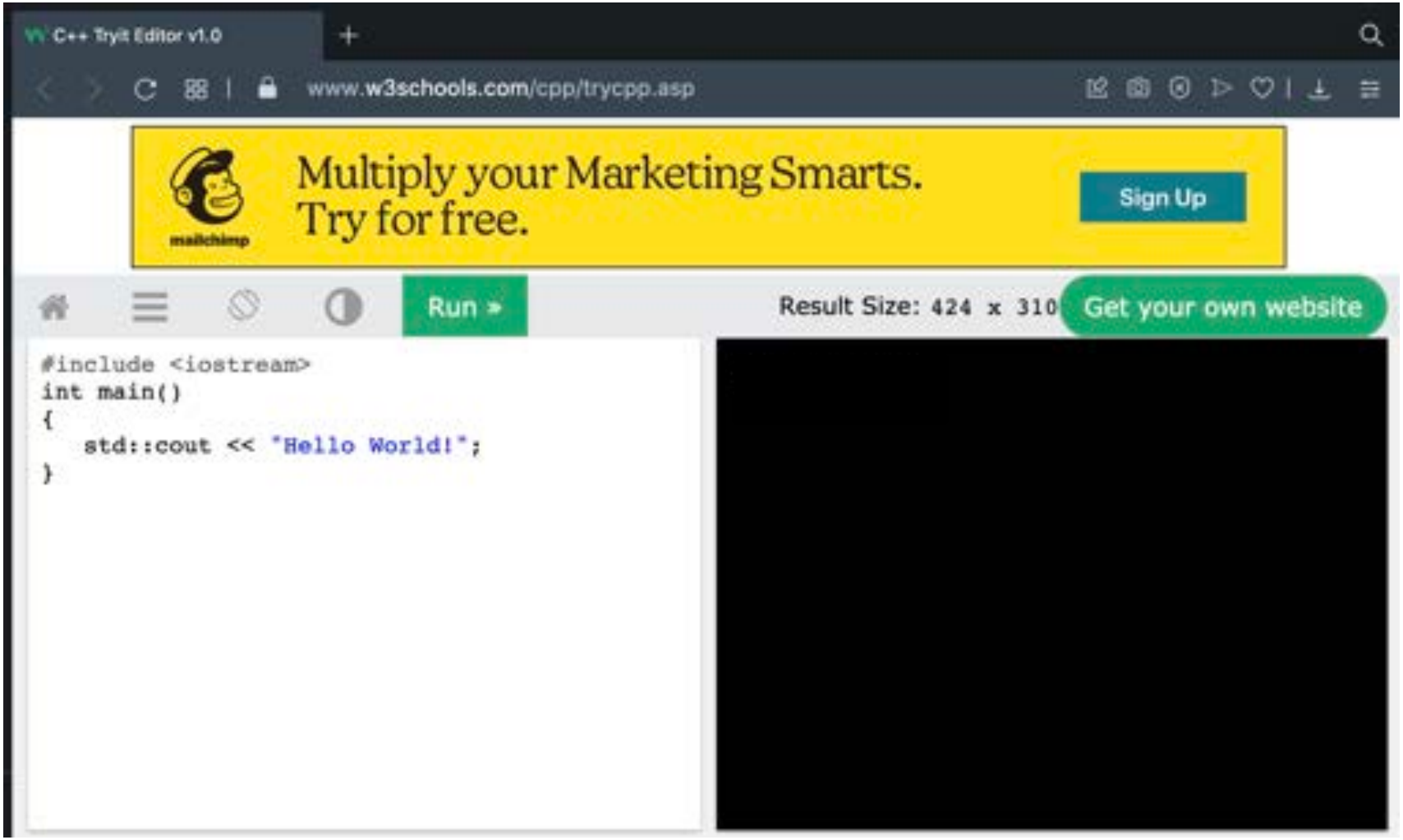
```
#include <iostream>
int main()
{
    std::cout << "Hello World!";
}
```

How to code in C++ ?

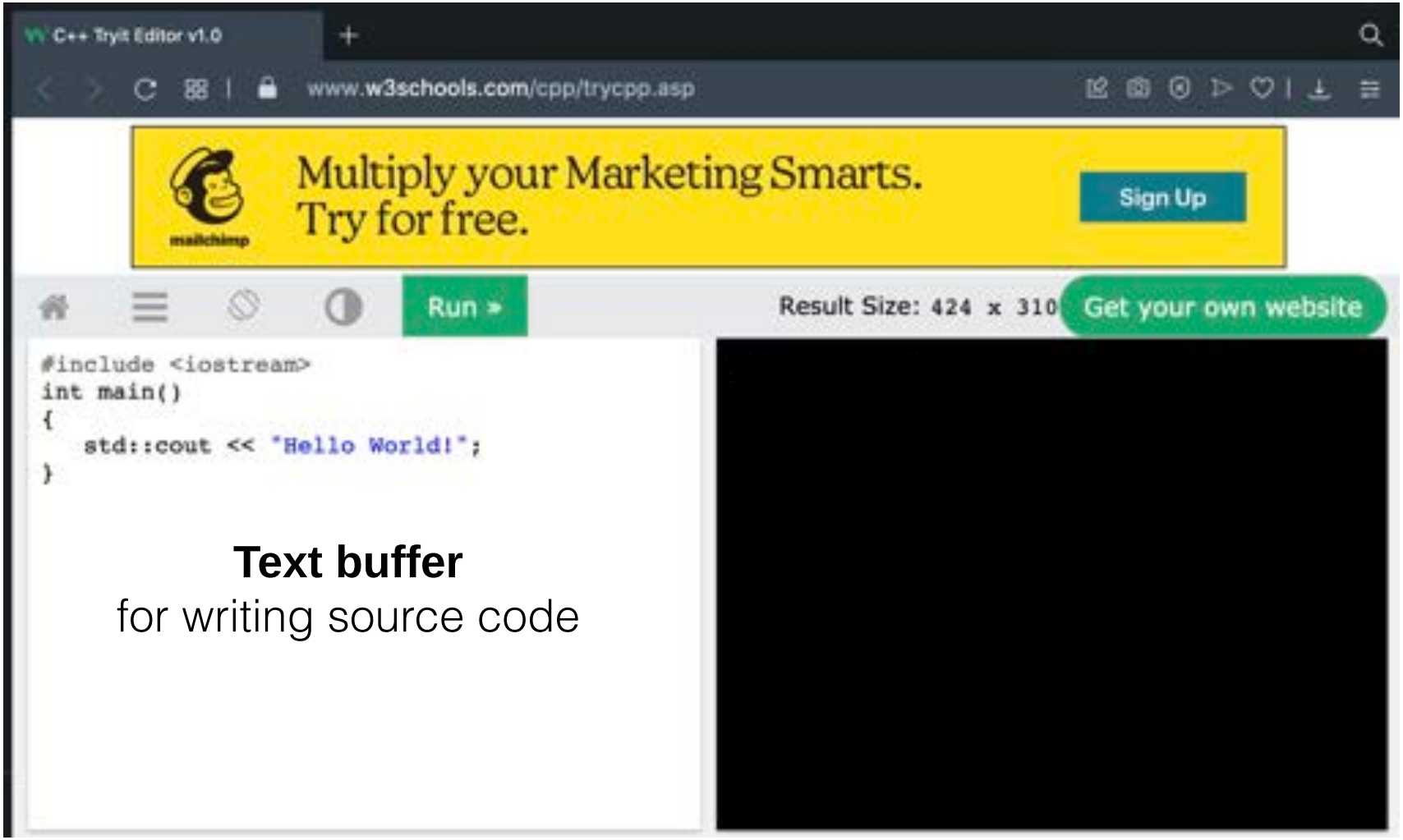
Step 1: Type code into a file

```
#include <iostream>
int main()
{
    std::cout << "Hello World!";
}
```

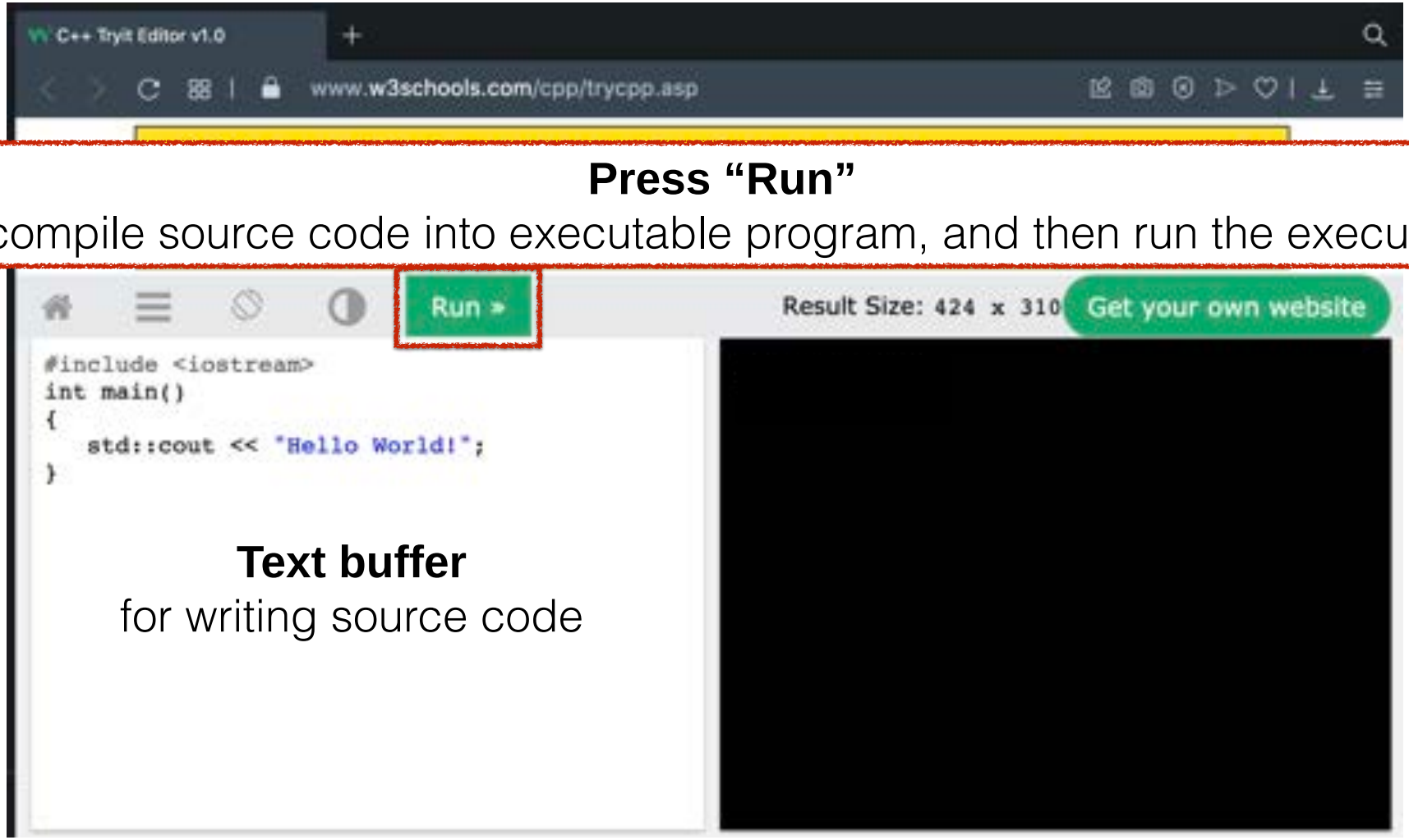
One option: online C++ editors



One option: online C++ editors



One option: online C++ editors



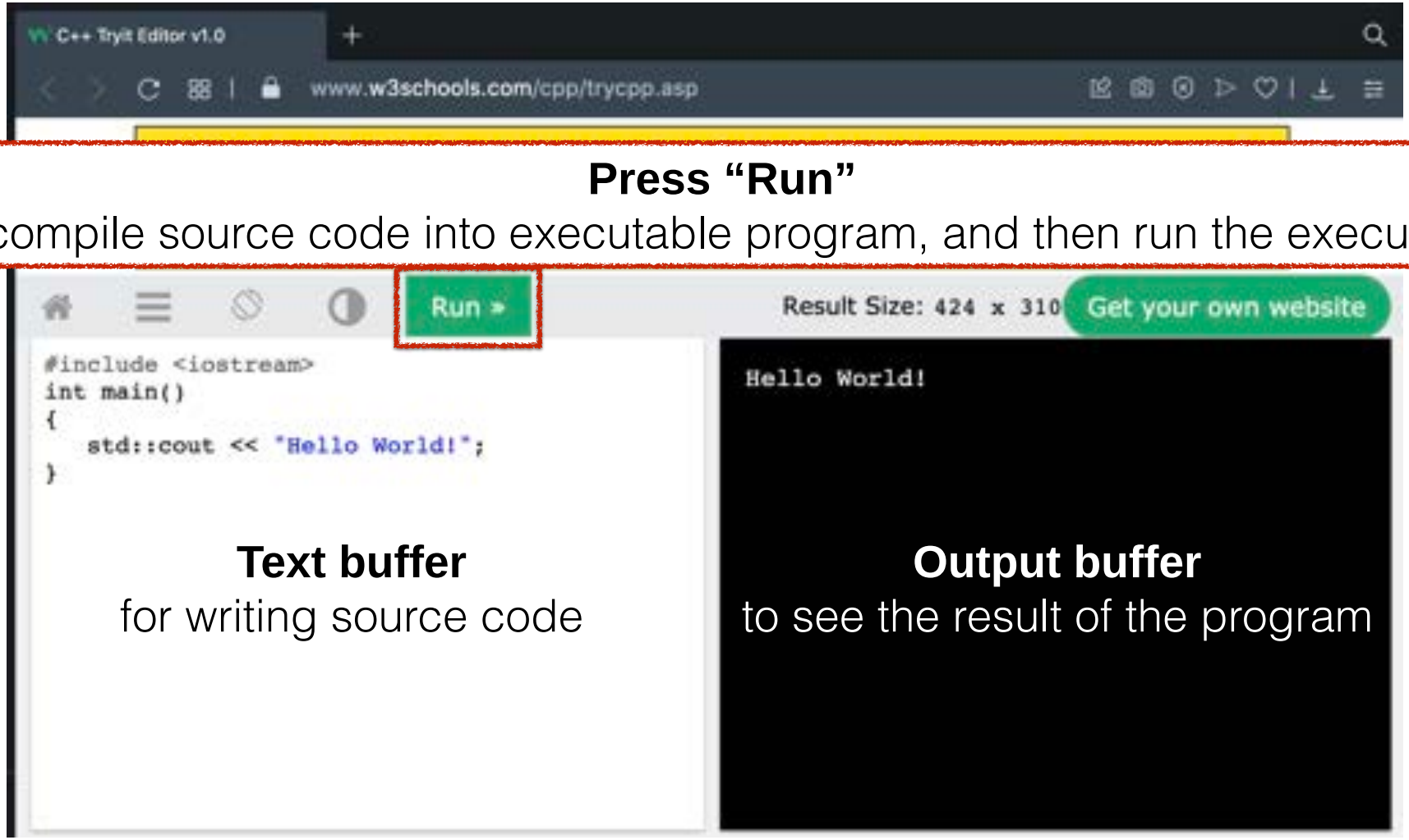
Press “Run”

to compile source code into executable program, and then run the executable

Text buffer

for writing source code

One option: online C++ editors

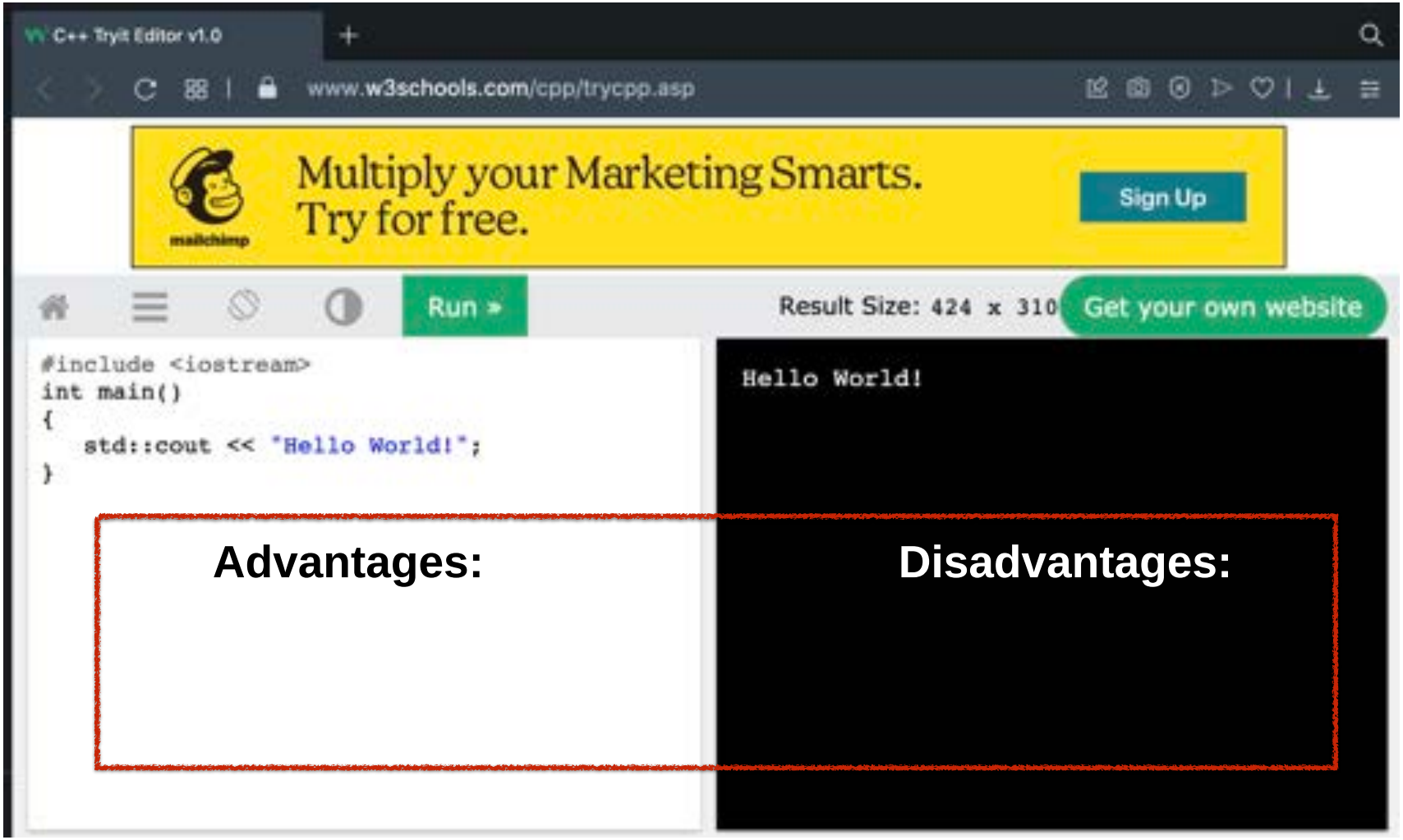


Press “Run”
to compile source code into executable program, and then run the executable

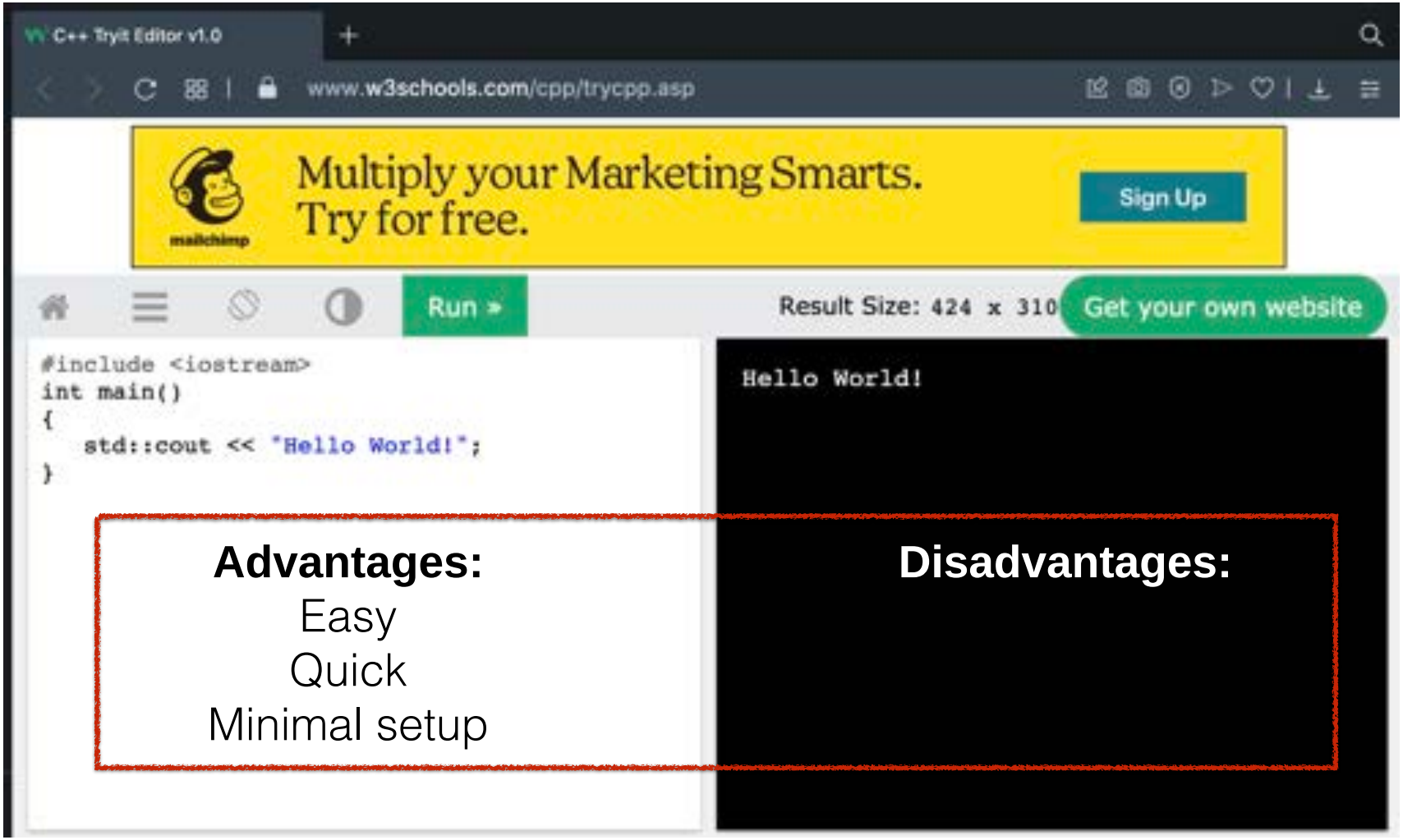
Text buffer
for writing source code

Output buffer
to see the result of the program

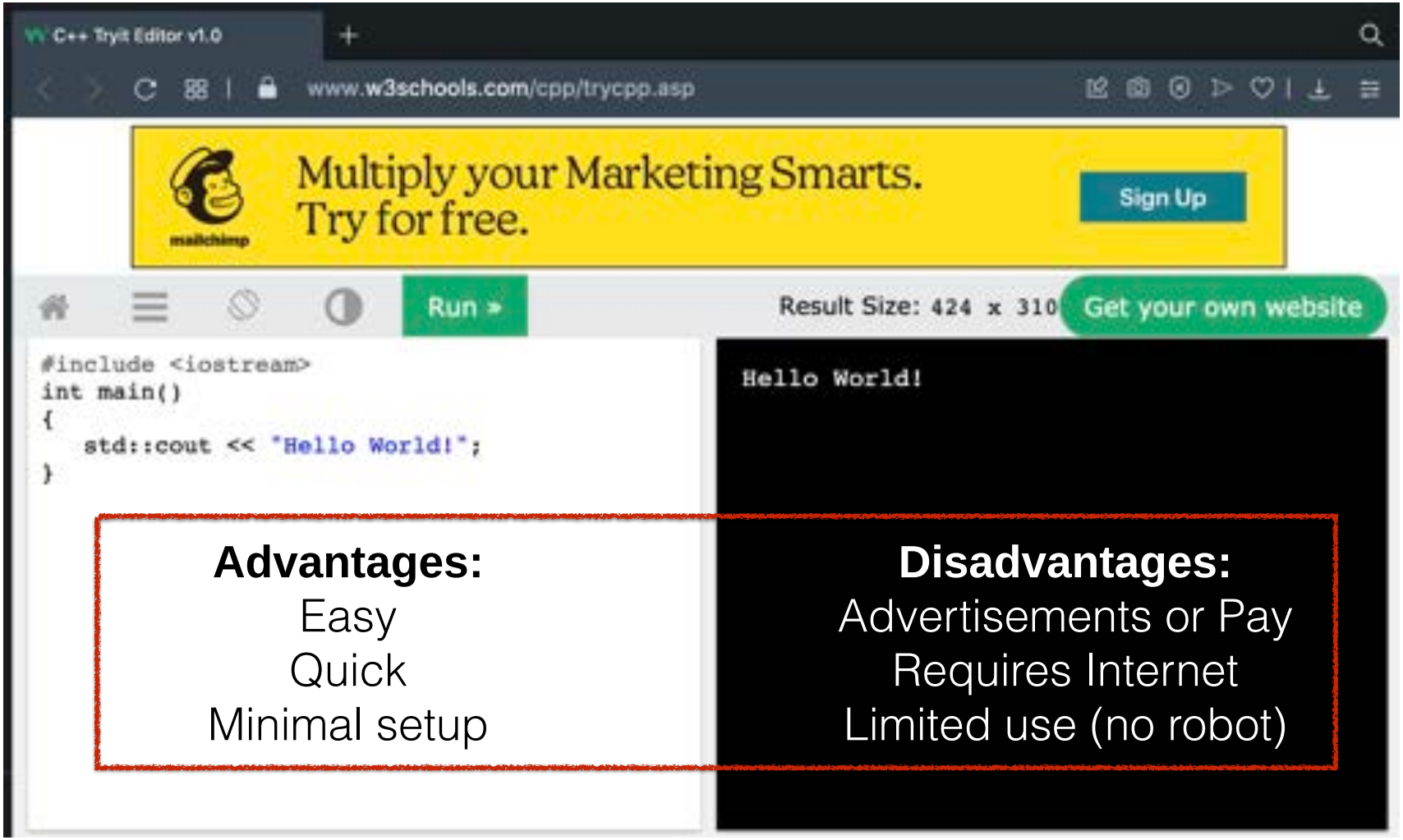
One option: online C++ editors



One option: online C++ editors



One option: online C++ editors



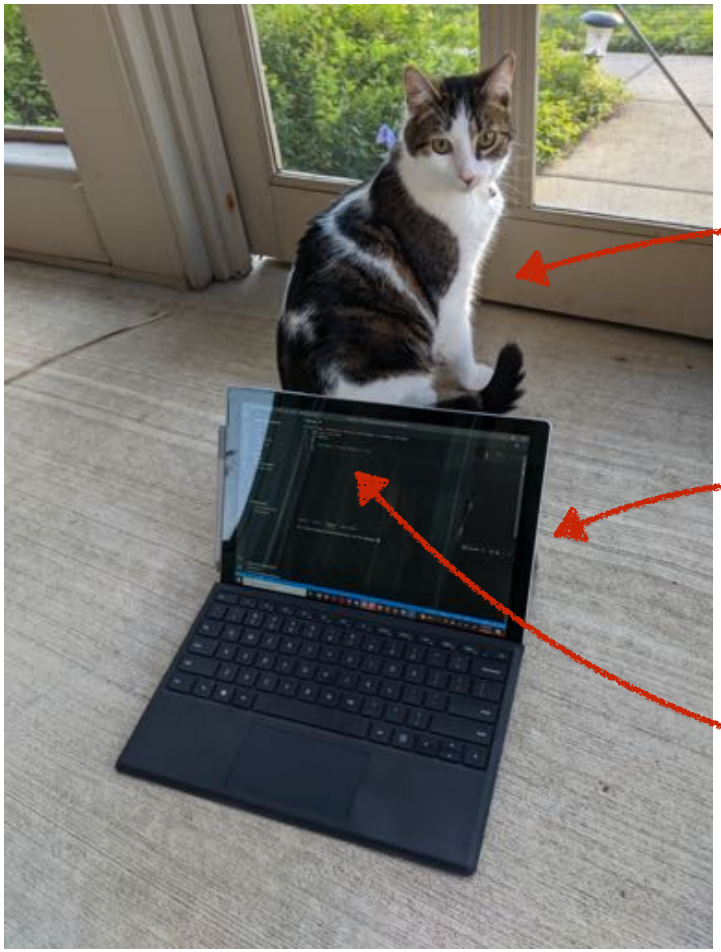
How to code in C++ ?

Step 1: Type code into a file

```
#include <iostream>
int main()
{
    std::cout << "Hello World!";
}
```

hello.cpp

Another option: coding in local files



Catari 2600

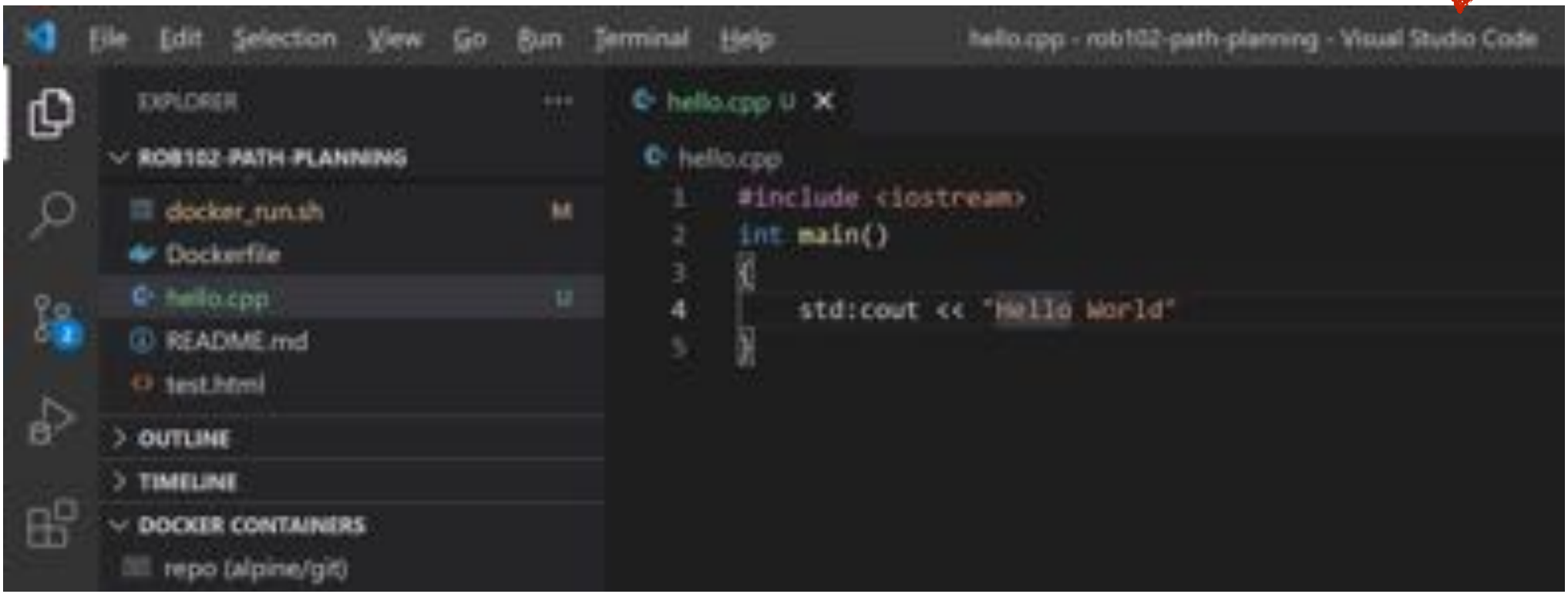
Microsoft Surface running
Visual Studio Code (VS Code)

```
#include <iostream>
int main()
{
    std::cout << "Hello World!";
}
```

hello.cpp

Step 1: Type code into a file

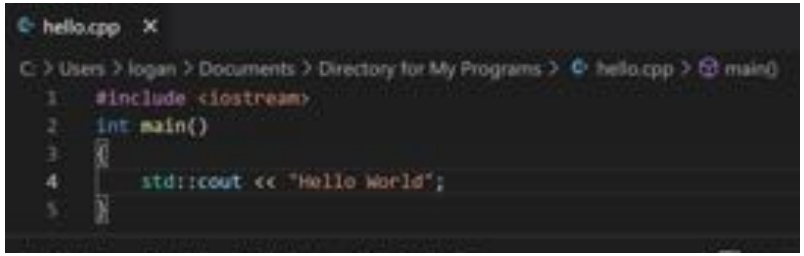
hello.cpp in VS Code



Step 1: Type code into a file

Text editor

Make changes to C++ code



hello.cpp

Step 1: Type code into a file

Text editor

Make changes to C++ code

```
#include <iostream>
int main()
{
   std::cout << "Hello World!";
}
```

hello.cpp

Step 1: Type code into a file

Text editor

Make changes to C++ code

```
#include <iostream>
int main()
{
   std::cout << "Hello World!";
}
```

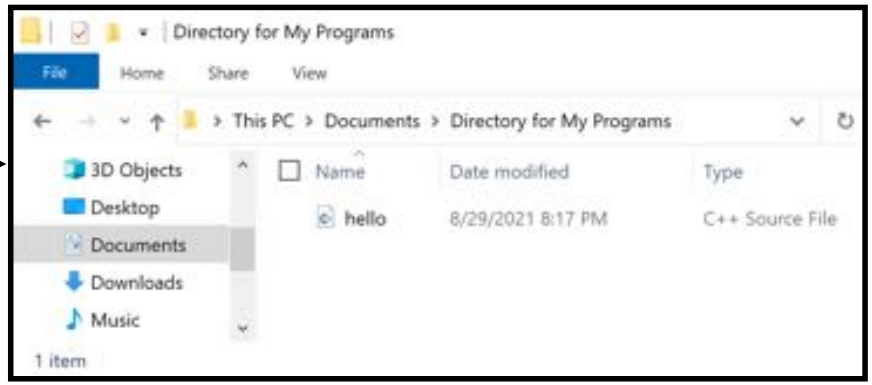
hello.cpp

OPEN FILE

SAVE FILE

Source code

C++ files on computer file system containing your code



Step 1: Type code into a file

What is a filesystem?

Text editor

Make changes to C++ code

```
#include <iostream>
int main()
{
    std::cout << "Hello World!";
}
```

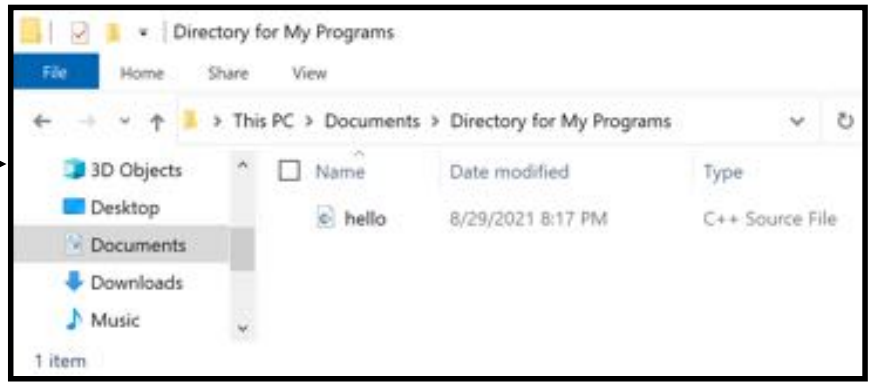
hello.cpp

OPEN FILE

SAVE FILE

Source code

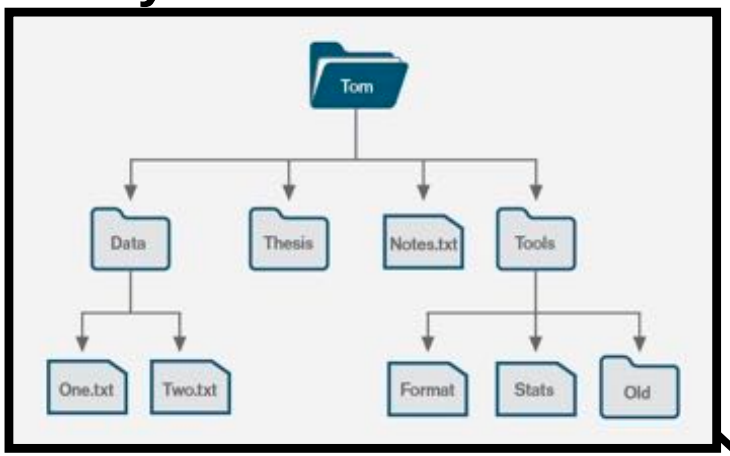
C++ files on computer file system containing your code

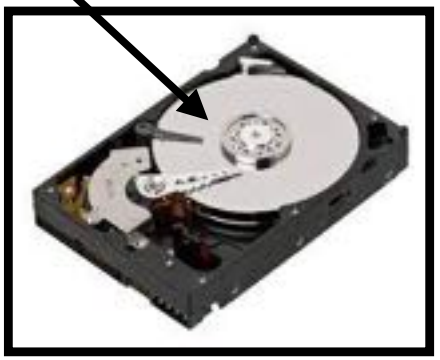


What is a filesystem?

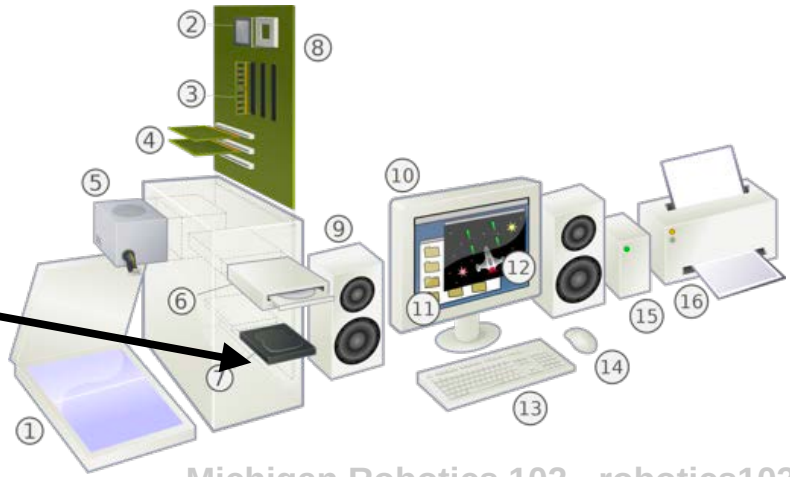
A filesystem organizes information on a storage device into a hierarchy of *directories* that contain data in *files*

Filesystem





Hard Drive Storage



Step 2: Compile source code

Text editor

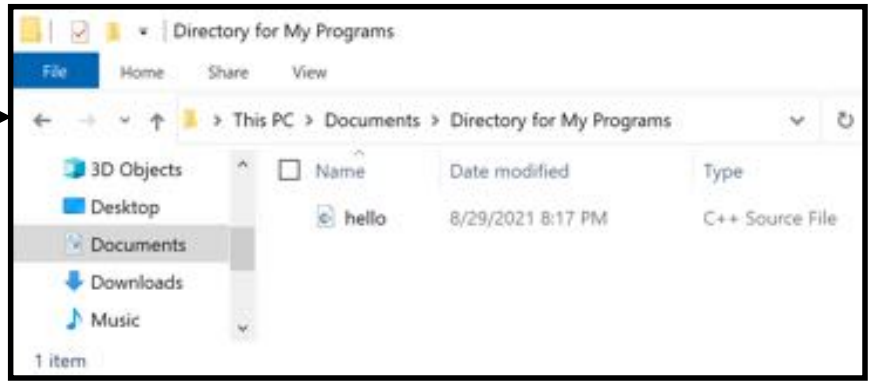
Make changes to C++ code

```
#include <iostream>
int main()
{
   std::cout << "Hello World!";
}
```

hello.cpp

OPEN FILE

SAVE FILE



Source code

C++ files on computer file system containing your code

Step 2: Compile source code

Compiler
Build executable program from C++ source files

Text editor
Make changes to C++ code

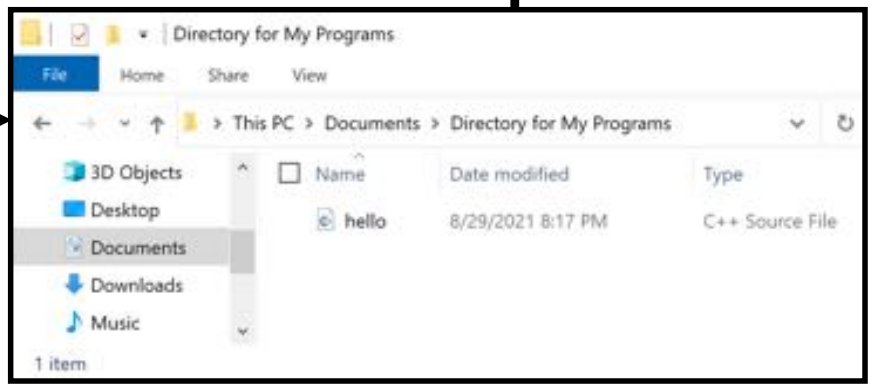
```
#include <iostream>
int main()
{
   std::cout << "Hello World!";
}
```

hello.cpp

OPEN FILE

SAVE FILE

```
bash                    ⌥⌘3
$ g++ hello.cpp -o hello
```

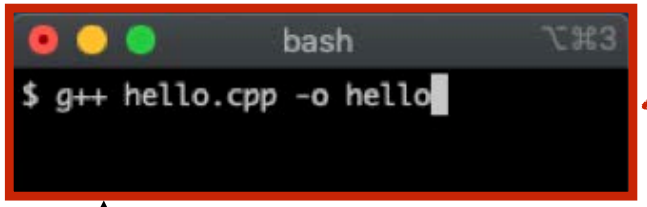


Source code
C++ files on computer file system containing your code

Step 2: Compile source code

Compiler
Build executable program from C++ source files

COMMAND LINE TERMINAL



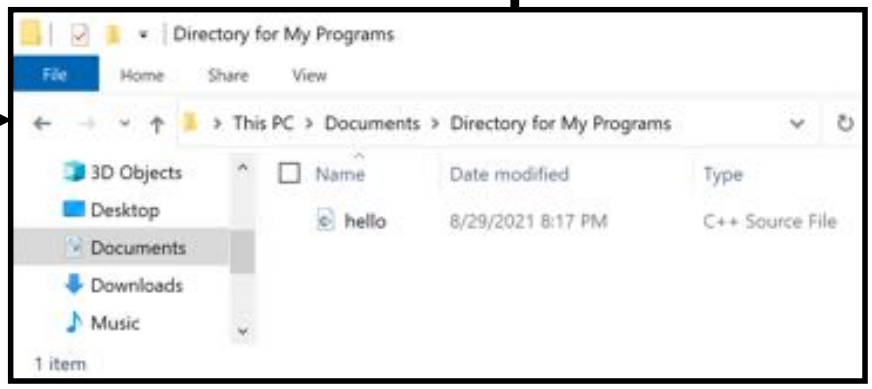
Text editor
Make changes to C++ code

```
#include <iostream>
int main()
{
   std::cout << "Hello World!";
}
```

hello.cpp

OPEN FILE

SAVE FILE



Source code
C++ files on computer file system containing your code

Step 2: Compile source code

Compiler
Build executable program from C++ source files

COMMAND LINE TERMINAL

Text editor
Make changes to C++ code

```
#include <iostream>
int main()
{
    std::cout << "Hello World!";
}
```

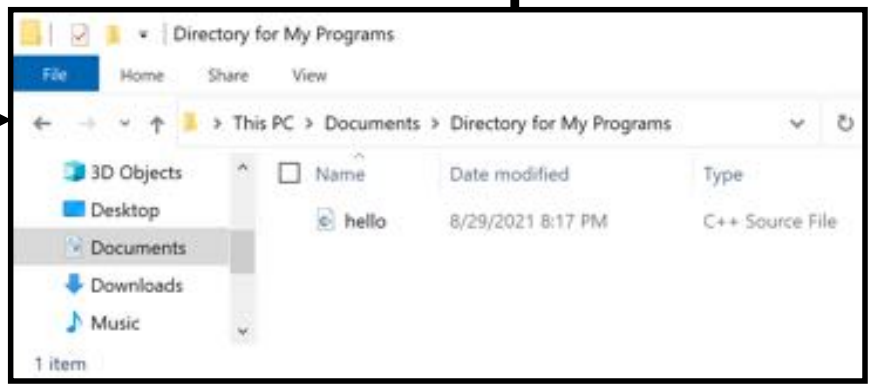
hello.cpp

OPEN FILE

SAVE FILE

```
bash                    ⌥⌘3
$ g++ hello.cpp -o hello
```

COMPILER PROGRAM



Source code
C++ files on computer file system containing your code

Step 2: Compile source code

Compiler
Build executable program from C++ source files

COMMAND LINE TERMINAL

Text editor
Make changes to C++ code

```
#include <iostream>
int main()
{
   std::cout << "Hello World!";
}
```

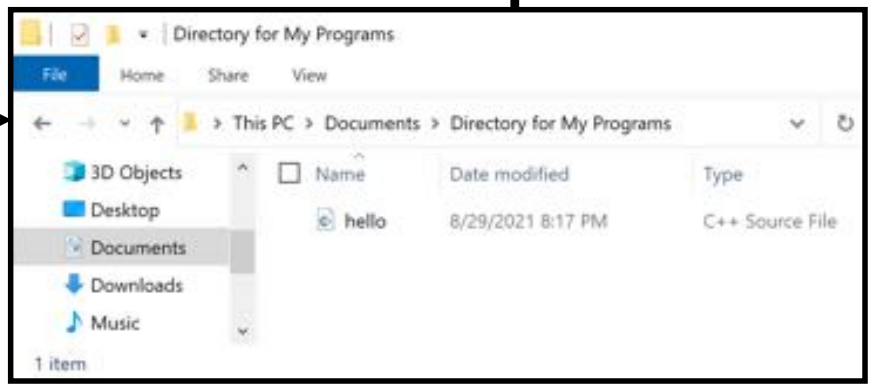
hello.cpp

OPEN FILE

SAVE FILE

```
bash ⌥⌘3
$ g++ hello.cpp -o hello
```

COMPILER PROGRAM



Source code
C++ files on computer file system containing your code

Step 2: Compile source code

Compiler
Build executable program from C++ source files

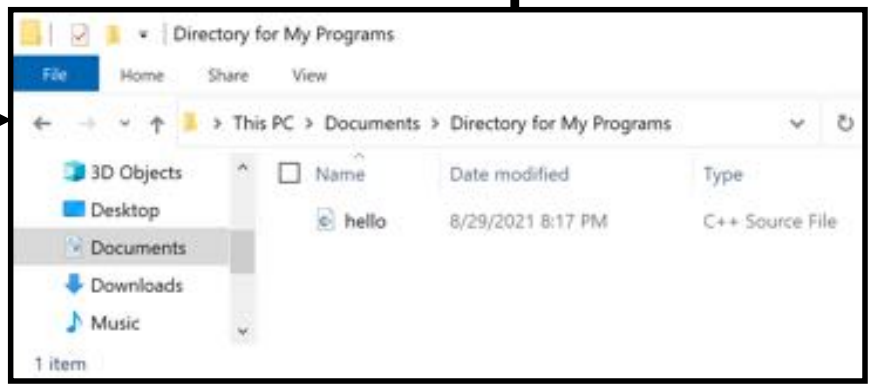
Text editor
Make changes to C++ code

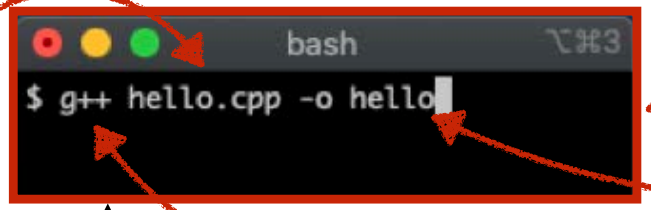
```
#include <iostream>
int main()
{
    std::cout << "Hello World!";
}
```

hello.cpp

OPEN FILE

SAVE FILE





COMMAND LINE TERMINAL

HELLO WORLD EXECUTABLE

COMPILER PROGRAM

Source code
C++ files on computer file system containing your code

Step 2: Compile source code

Compiler
Build executable program from C++ source files

Text editor
Make changes to C++ code

```
#include <iostream>
int main()
{
    std::cout << "Hello World!";
}
```

hello.cpp

OPEN FILE

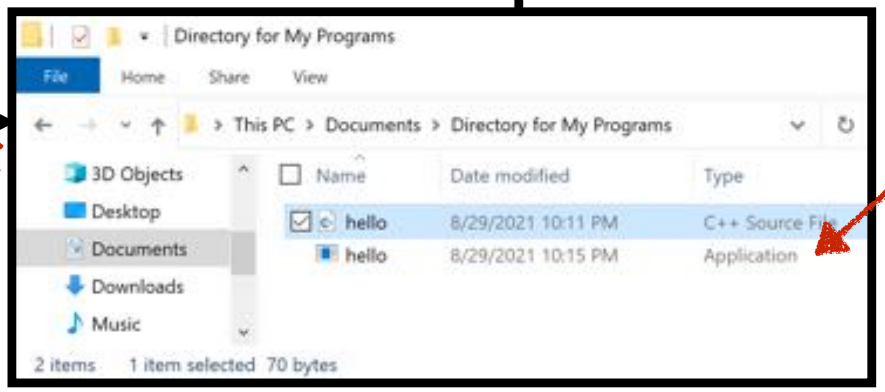
SAVE FILE

COMMAND LINE TERMINAL

```
bash ⌥⌘3
$ g++ hello.cpp -o hello
```

COMPILER PROGRAM

HELLO WORLD EXECUTABLE



Source code
C++ files on computer file system containing your code

Step 2: Compile source code

Compiler
Build executable program from C++ source files

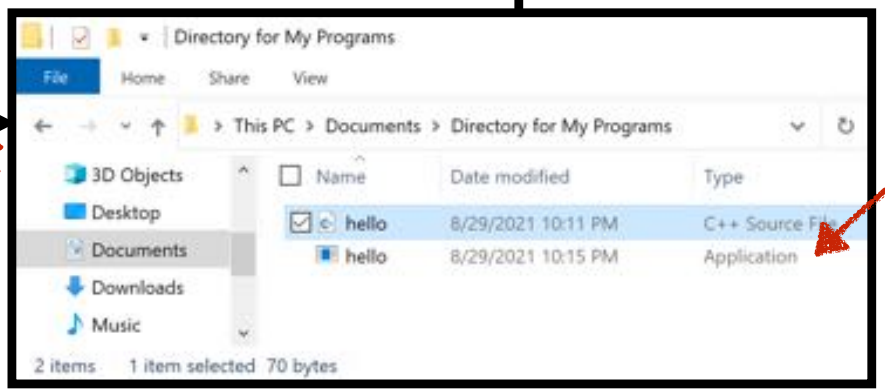
Text editor
Make changes to C++ code

```
#include <iostream>
int main()
{
    std::cout << "Hello World!";
}
```

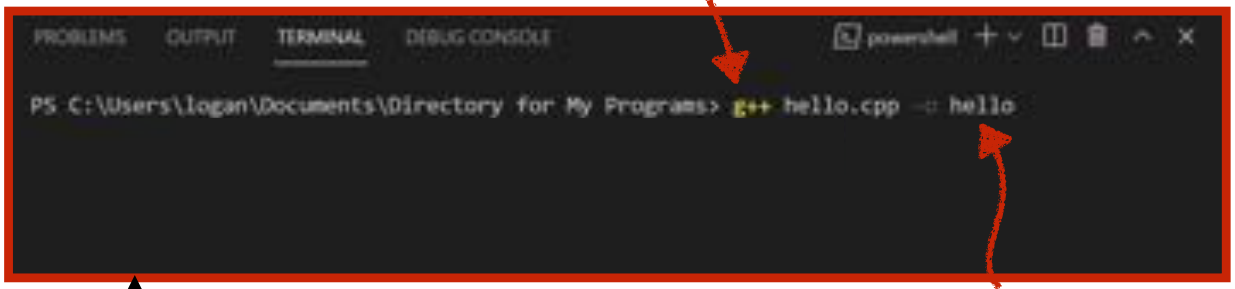
hello.cpp

OPEN FILE

SAVE FILE



VS Code Terminal



HELLO WORLD EXECUTABLE

Source code
C++ files on computer file system containing your code

Step 3: Run Executable Program

Compiler

Build executable program from C++ source files

Text editor

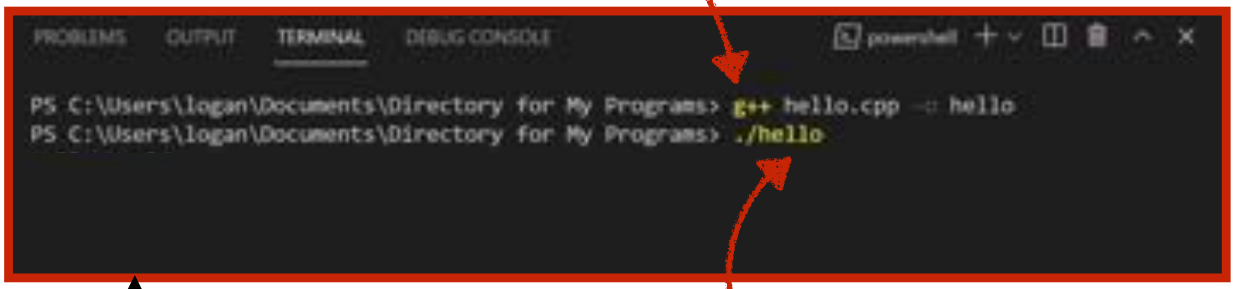
Make changes to C++ code

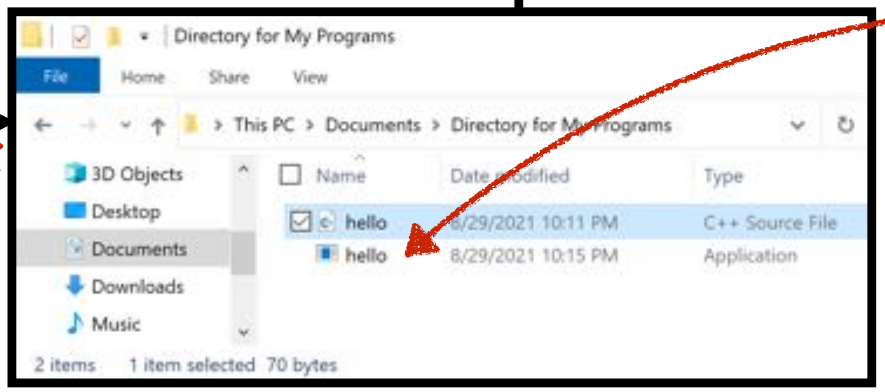
```
#include <iostream>
int main()
{
    std::cout << "Hello World!";
}
```

hello.cpp

OPEN FILE

SAVE FILE





Executable

Binary file on computer file system that can be run

Source code

C++ files on computer file system containing your code

Step 3: Run Executable Program

Compiler

Build executable program from C++ source files

Text editor

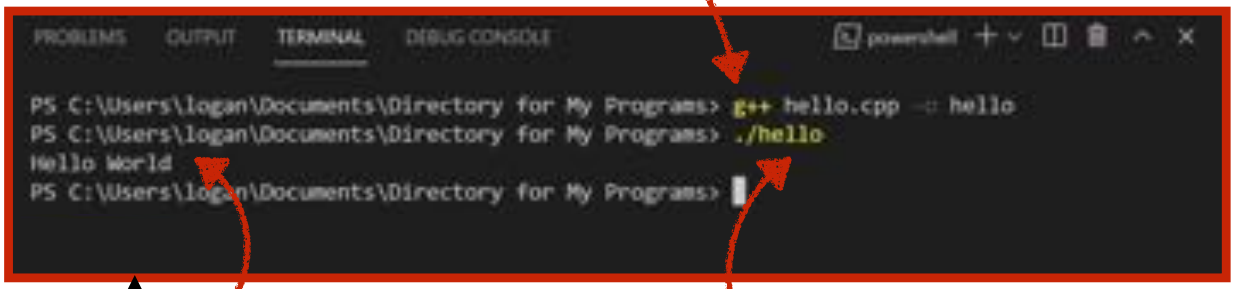
Make changes to C++ code

```
#include <iostream>
int main()
{
    std::cout << "Hello World!";
}
```

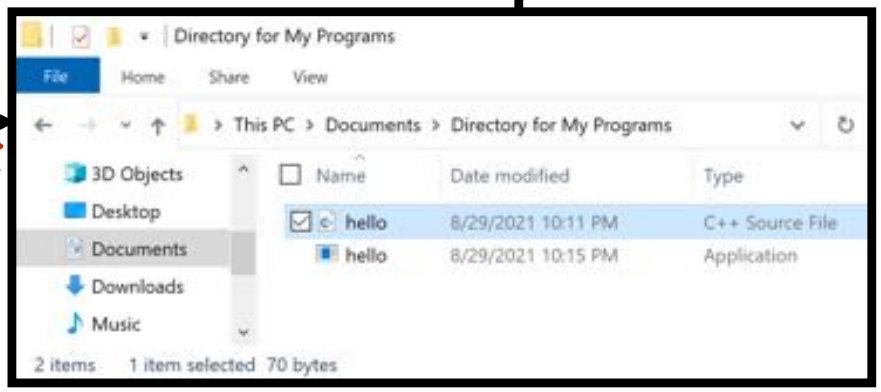
hello.cpp

OPEN FILE

SAVE FILE



PROGRAM OUTPUT



Executable

Binary file on computer file system that can be run

Source code

C++ files on computer file system containing your code

Compiler

Executable

Text editor

```
#include <iostream>
int main()
{
    std::cout << "Hello World!";
}
```

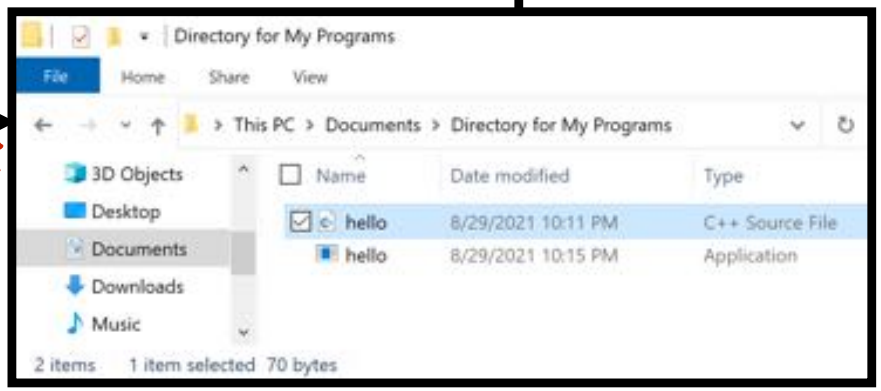
hello.cpp

OPEN FILE

SAVE FILE

COMPILE

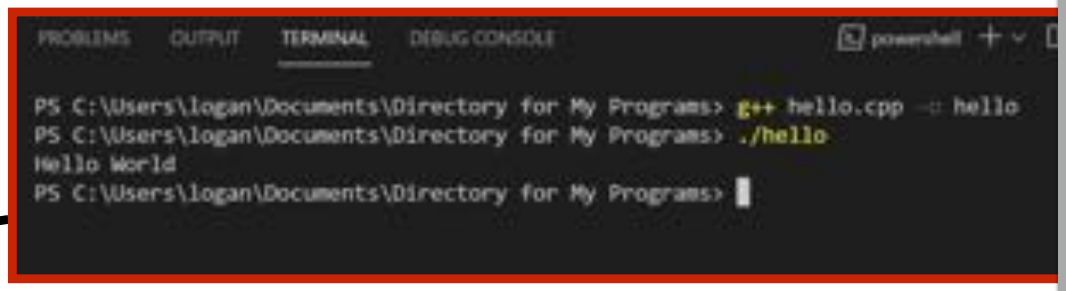
AND EXECUTE



Source code

Compiler

Executable



Ideas

Text editor

```
#include <iostream>
int main()
{
   std::cout << "Hello World!";
}
```

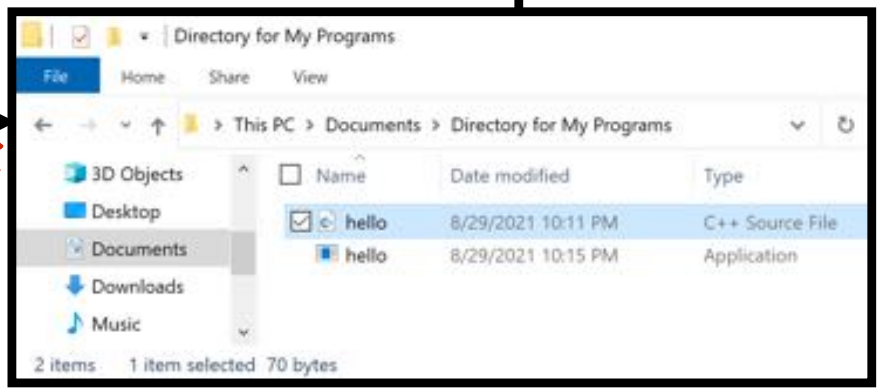
hello.cpp

Open file

Save file

Compile

And Execute



Source code

Coding process

Get ideas

↓

Open C++ file

↓

Update code

↓

Save C++ file

↓

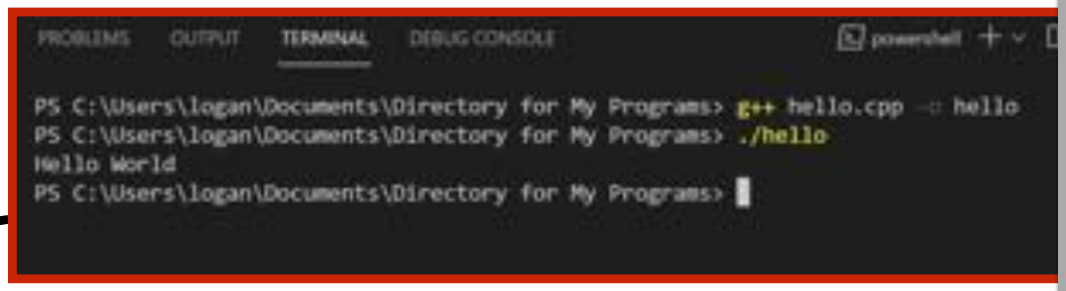
Compile C++ file

↓

Run Executable

Compiler

Executable



IDEAS

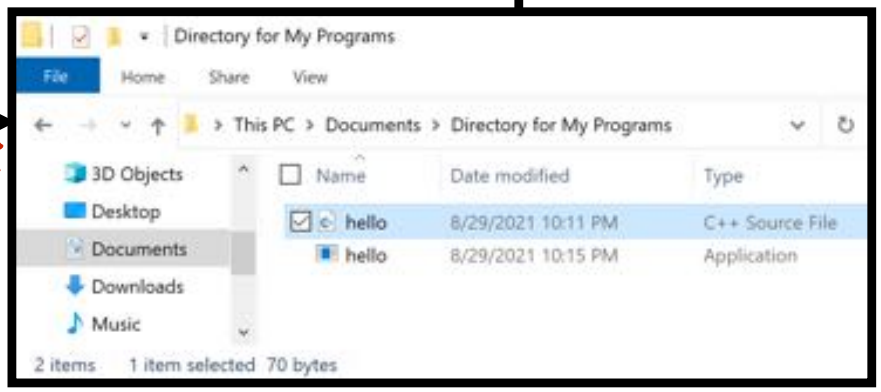
Text editor

```
#include <iostream>
int main()
{
   std::cout << "Hello World!";
}
```

hello.cpp

OPEN FILE

SAVE FILE



Source code

COMPILE AND EXECUTE

Coding process

GET IDEAS

OPEN C++ FILE

UPDATE CODE

SAVE C++ FILE

COMPILE C++ FILE

RUN EXECUTABLE

New idea:
Print that I am in 102

Compiler

Executable

Ideas

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE                powershell
PS C:\Users\logan\Documents\Directory for My Programs> g++ hello.cpp -o hello
PS C:\Users\logan\Documents\Directory for My Programs> ./hello
Hello World
PS C:\Users\logan\Documents\Directory for My Programs>
```

Text editor

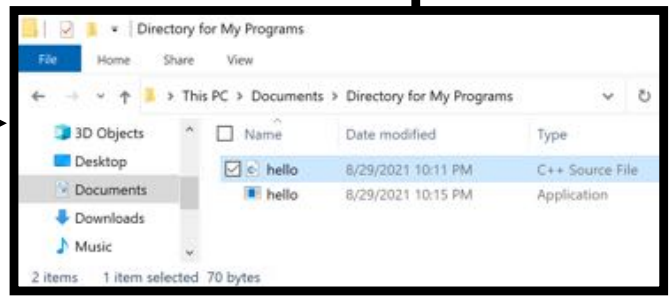
```
#include <iostream>
int main()
{
   std::cout << "Hello World";
   std::cout << "Chad is in Robotics 102";
}
```

hello.cpp

Open file

Save file

Source code



Compile

And Execute

Coding process

Get ideas

↓

Open C++ file

↓

Update code

↓

Save C++ file

↓

Compile C++ file

↓

Run Executable

Compiler

Executable

Ideas

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE                powershell

PS C:\Users\logan\Documents\Directory for My Programs> g++ hello.cpp -o hello
PS C:\Users\logan\Documents\Directory for My Programs> ./hello
Hello World
PS C:\Users\logan\Documents\Directory for My Programs> g++ hello.cpp -o hello
PS C:\Users\logan\Documents\Directory for My Programs> ./hello
Hello WorldChad is in Robotics 102
PS C:\Users\logan\Documents\Directory for My Programs>
```

Text editor

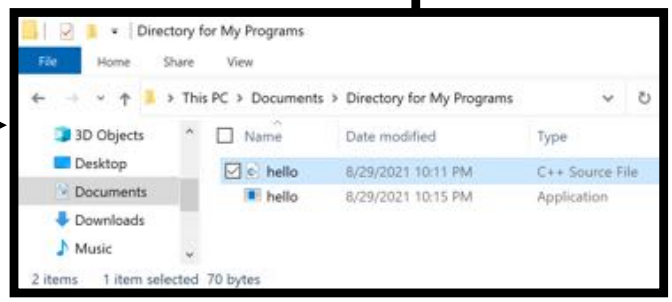
```
#include <iostream>
int main()
{
   std::cout << "Hello World”;
   std::cout << "Chad is in Robotics 102";
}
```

hello.cpp

Open file

Save file

Source code



Compile

And Execute

Coding process

Get ideas

↓

Open C++ file

↓

Update code

↓

Save C++ file

↓

Compile C++ file

↓

Run Executable

Compiler

Executable

Ideas

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE                powershell

PS C:\Users\logan\Documents\Directory for My Programs> g++ hello.cpp -o hello
PS C:\Users\logan\Documents\Directory for My Programs> ./hello
Hello World
PS C:\Users\logan\Documents\Directory for My Programs> g++ hello.cpp -o hello
PS C:\Users\logan\Documents\Directory for My Programs> ./hello
Hello WorldChad is in Robotics 102
PS C:\Users\logan\Documents\Directory for My Programs>
```

Text editor

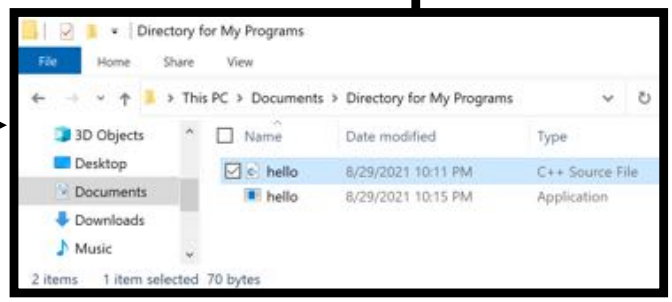
```
#include <iostream>
int main()
{
   std::cout << "Hello World";
   std::cout << "Chad is in Robotics 102";
}
```

hello.cpp

Open file

Save file

Source code



Compile

And Execute

Coding process

Get ideas

↓

Open C++ file

↓

Update code

↓

Save C++ file

↓

Compile C++ file

↓

Run Executable

New idea:
Break each line

Compiler

Executable

Ideas

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE                powershell

PS C:\Users\logan\Documents\Directory for My Programs> g++ hello.cpp -o hello
PS C:\Users\logan\Documents\Directory for My Programs> ./hello
Hello World
PS C:\Users\logan\Documents\Directory for My Programs> g++ hello.cpp -o hello
PS C:\Users\logan\Documents\Directory for My Programs> ./hello
Hello WorldChad is in Robotics 102
PS C:\Users\logan\Documents\Directory for My Programs> g++ hello.cpp -o hello
PS C:\Users\logan\Documents\Directory for My Programs> ./hello
Hello World
Chad is in Robotics 102
PS C:\Users\logan\Documents\Directory for My Programs>
```

Text editor

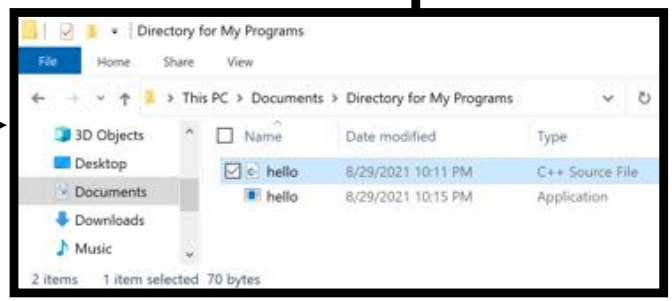
```
#include <iostream>
int main()
{
   std::cout << "Hello World" << "\n";
   std::cout << "Chad is in Robotics 102";
}
```

hello.cpp

Compile

And Execute

Open file



Save file

Source code

Coding process

Get ideas

↓

Open C++ file

↓

Update code

↓

Save C++ file

↓

Compile C++ file

↓

Run Executable


Compiler
Executable
IDEAS
Text editor
#include <iostream>
int main()
{
std::cout << "Hello World" << "\n";
std::cout << "Chad is in Robotics 102";
}
hello.cpp
OPEN FILE
SAVE FILE
Source code
AND EXECUTE
Coding process
GET IDEAS
OPEN C++ FILE
Wait
You did what?

Anatomy of a C++ Program

```
#include <iostream>
int main()
{
   std::cout << "Hello World” << “\n”;
   std::cout << "Chad is in Robotics 102";
}
```

hello.cpp

Anatomy of a C++ Program

All programs start in the main function

```
#include <iostream>
int main()
{
   std::cout << "Hello World” << “\n”;
   std::cout << "Chad is in Robotics 102";
}
```

hello.cpp

Anatomy of a C++ Program

All programs start in the main function

```
#include <iostream>
int main()
{
    std::cout << "Hello World" << "\n";
    std::cout << "Chad is in Robotics 102";
}
```

hello.cpp

Scope of main function delimited by matching braces

Anatomy of a C++ Program

All programs start in the main function

```
#include <iostream>
int main()
{
    std::cout << "Hello World" << "\n";
    std::cout << "Chad is in Robotics 102";
}
```

hello.cpp

Scope of main function delimited by matching braces

Statements are program instructions and end with a semicolon

First statement prints to "standard output" (the screen) the string (in quotes) followed by a new line character

All programs start in the main function

```
#include <iostream>
int main()
{
    std::cout << "Hello World" << "\n";
    std::cout << "Chad is in Robotics 102";
}
```

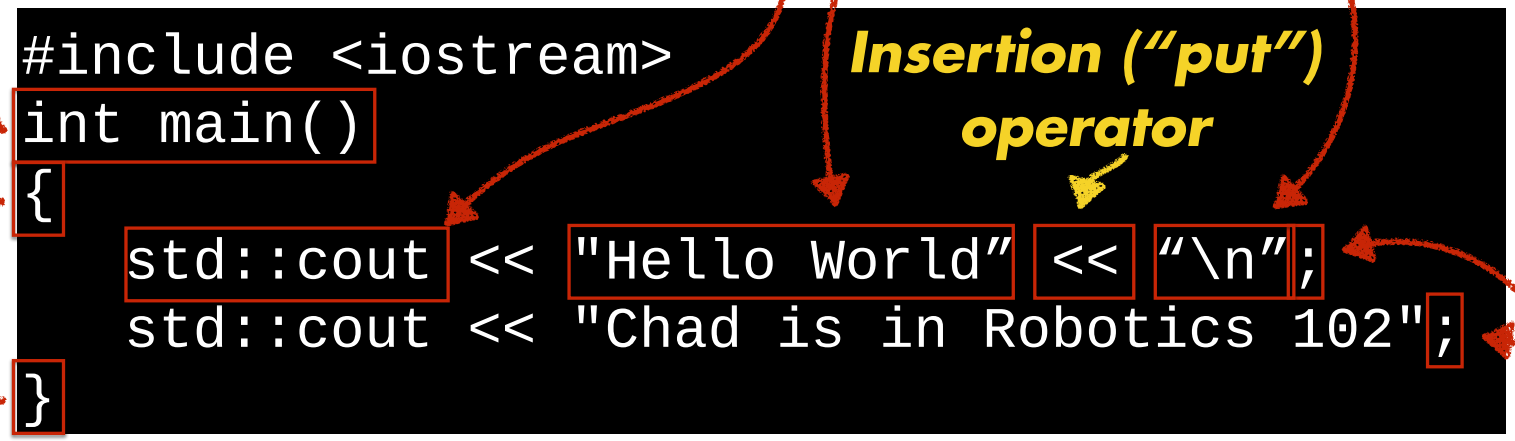
hello.cpp

Scope of main function delimited by matching braces

Statements are program instructions and end with a semicolon

First statement prints to "standard output" (the screen) the string (in quotes) followed by a new line character

All programs start in the main function



hello.cpp

Scope of main function delimited by matching braces

Statements are program instructions and end with a semicolon

Include the C++ Input-Output Stream Library, which provides std::cout

All programs start in the main function

First statement prints to "standard output" (the screen) the string (in quotes) followed by a new line character

Insertion ("put") operator

```
#include <iostream>
int main()
{
    std::cout << "Hello World" << "\n";
    std::cout << "Chad is in Robotics 102";
}
```

hello.cpp

Scope of main function delimited by matching braces

Statements are program instructions and end with a semicolon

Statements are executed in sequential order based on where they appear in the program

```
#include <iostream>
int main()
{
   std::cout << "Hello World” << “\n”;
   std::cout << "Chad is in Robotics 102";
}
```

hello.cpp

This statement is executed after this statement

Compiler

Executable

Ideas

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE                    powershell

PS C:\Users\logan\Documents\Directory for My Programs> g++ hello.cpp -o hello
PS C:\Users\logan\Documents\Directory for My Programs> ./hello
Hello World
PS C:\Users\logan\Documents\Directory for My Programs> g++ hello.cpp -o hello
PS C:\Users\logan\Documents\Directory for My Programs> ./hello
Hello WorldChad is in Robotics 102
PS C:\Users\logan\Documents\Directory for My Programs> g++ hello.cpp -o hello
PS C:\Users\logan\Documents\Directory for My Programs> ./hello
Hello World
Chad is in Robotics 102
PS C:\Users\logan\Documents\Directory for My Programs>
```

Text editor

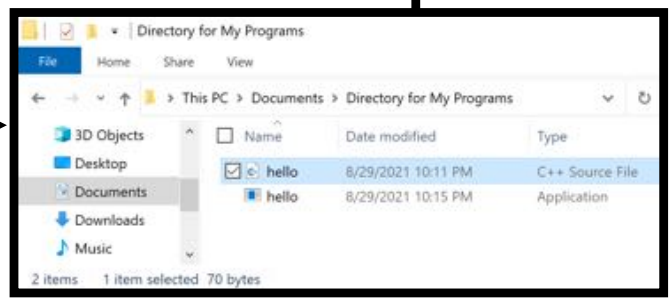
```
2  int main()
3  {
4      std::cout << "Hello World" << "\n";
5      std::cout << "Chad is in Robotics 102";
6  }
```

hello.cpp

Compile

And Execute

Open file



Save file

Source code

Coding process

Get ideas

↓

Open C++ file

↓

Update code

↓

Save C++ file

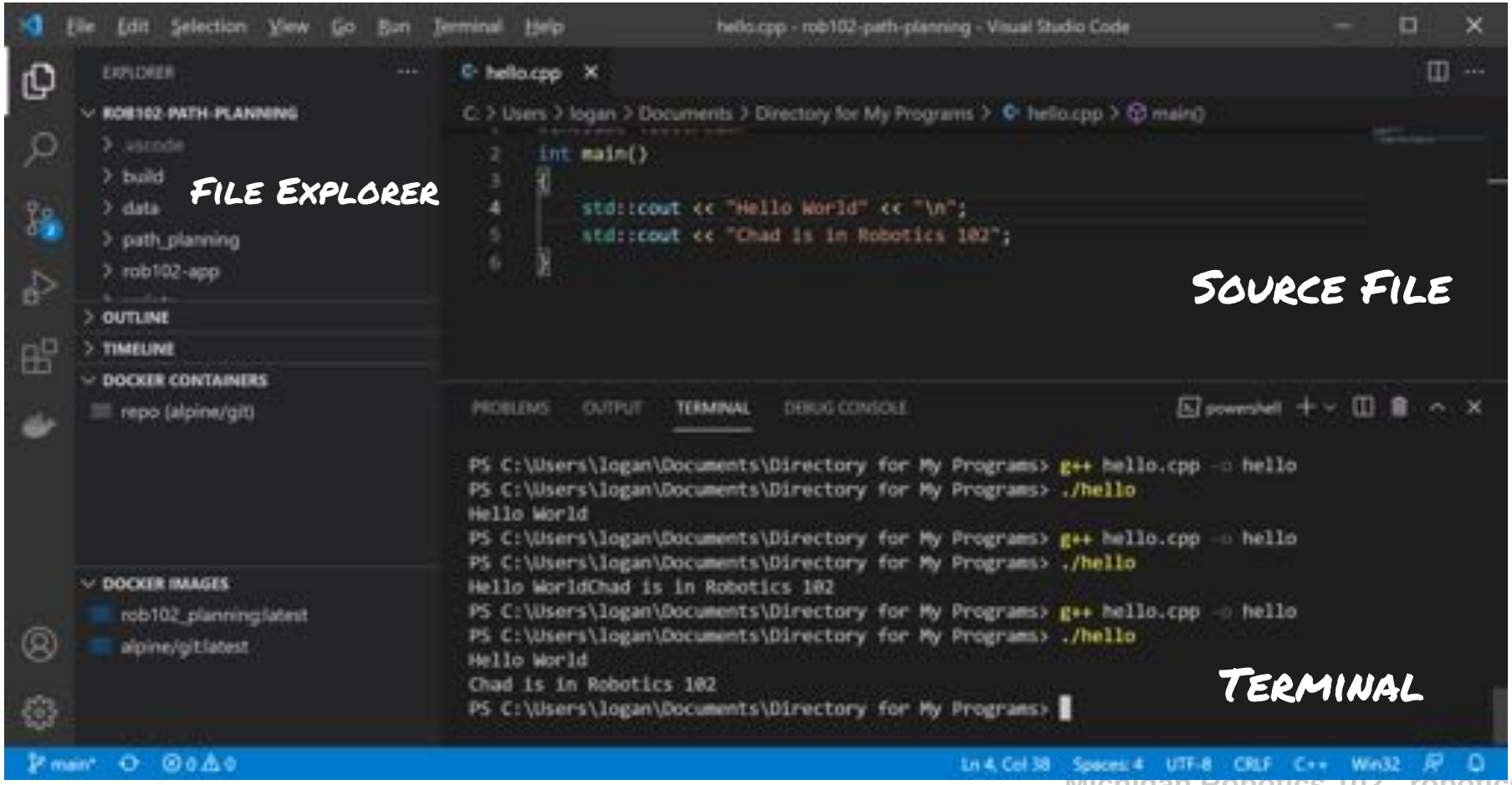
↓

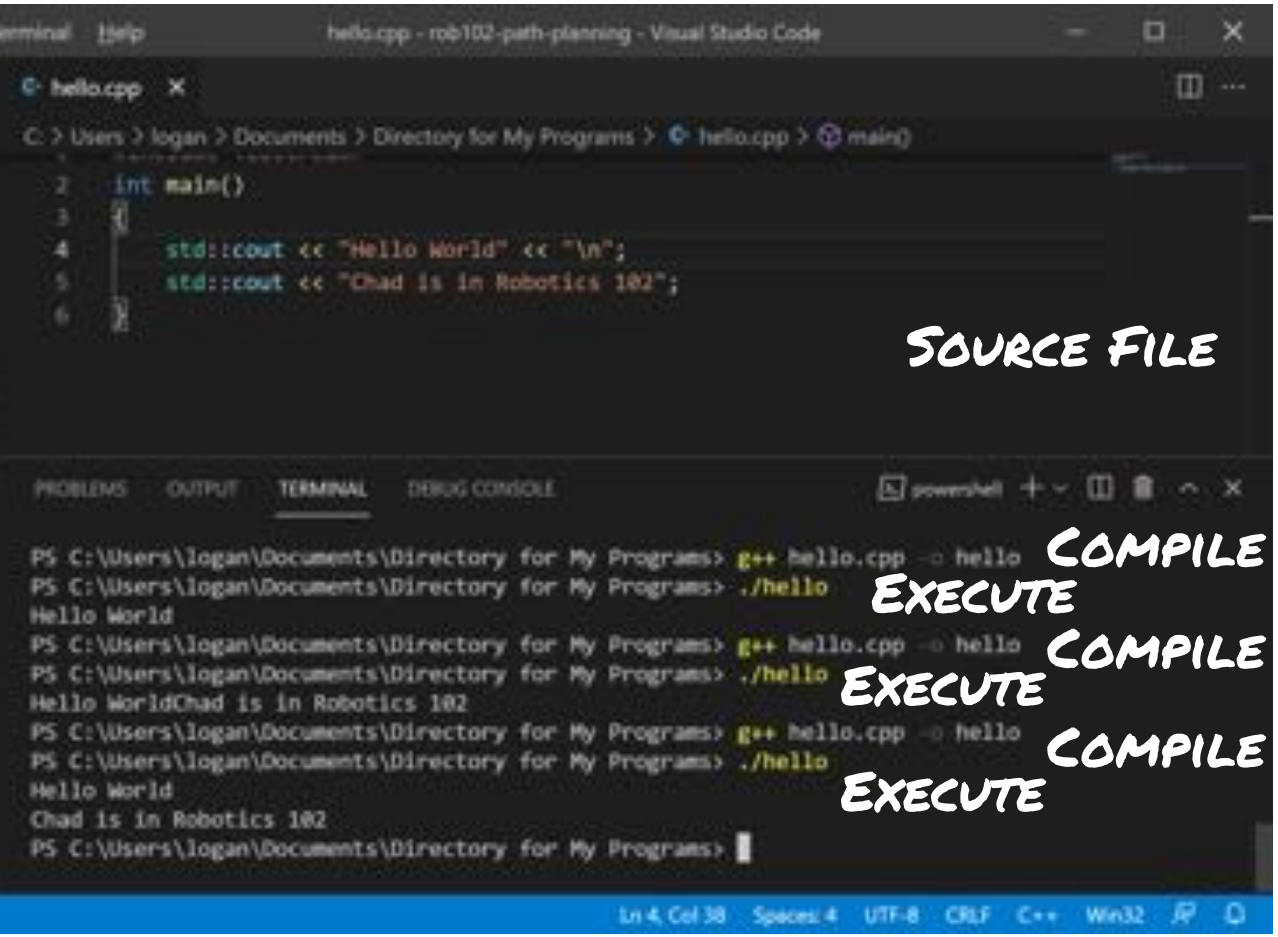
Compile C++ file

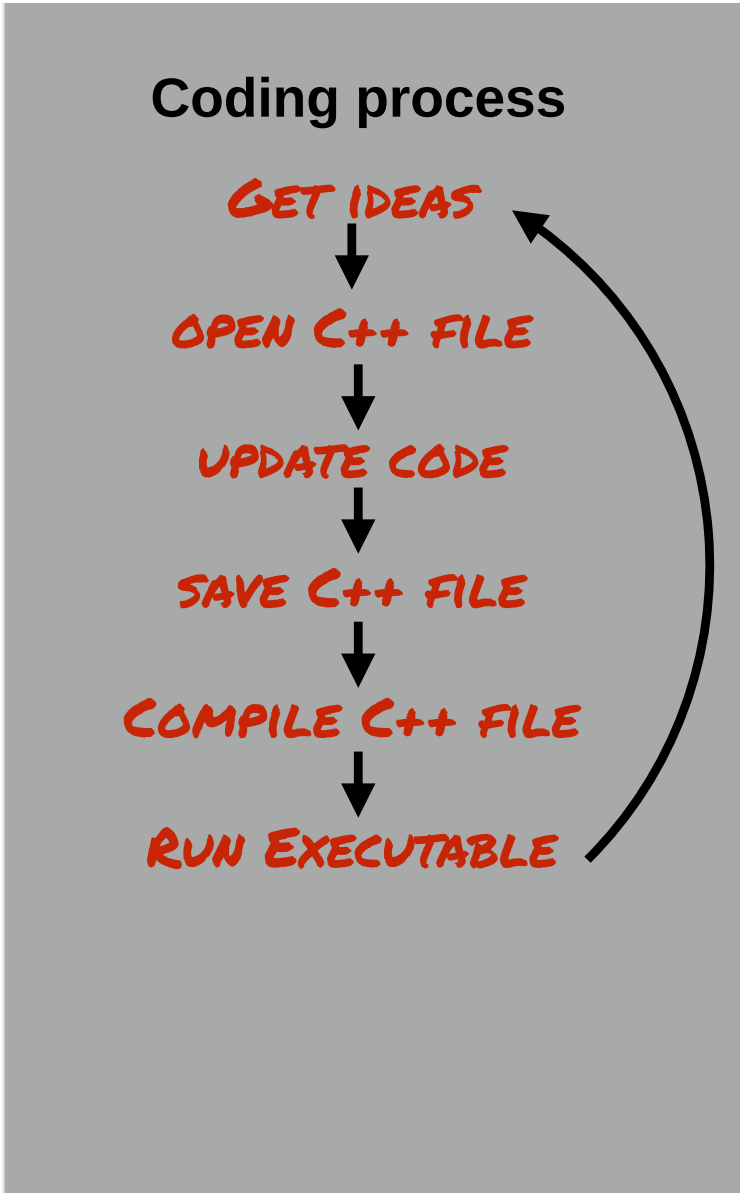
↓

Run Executable

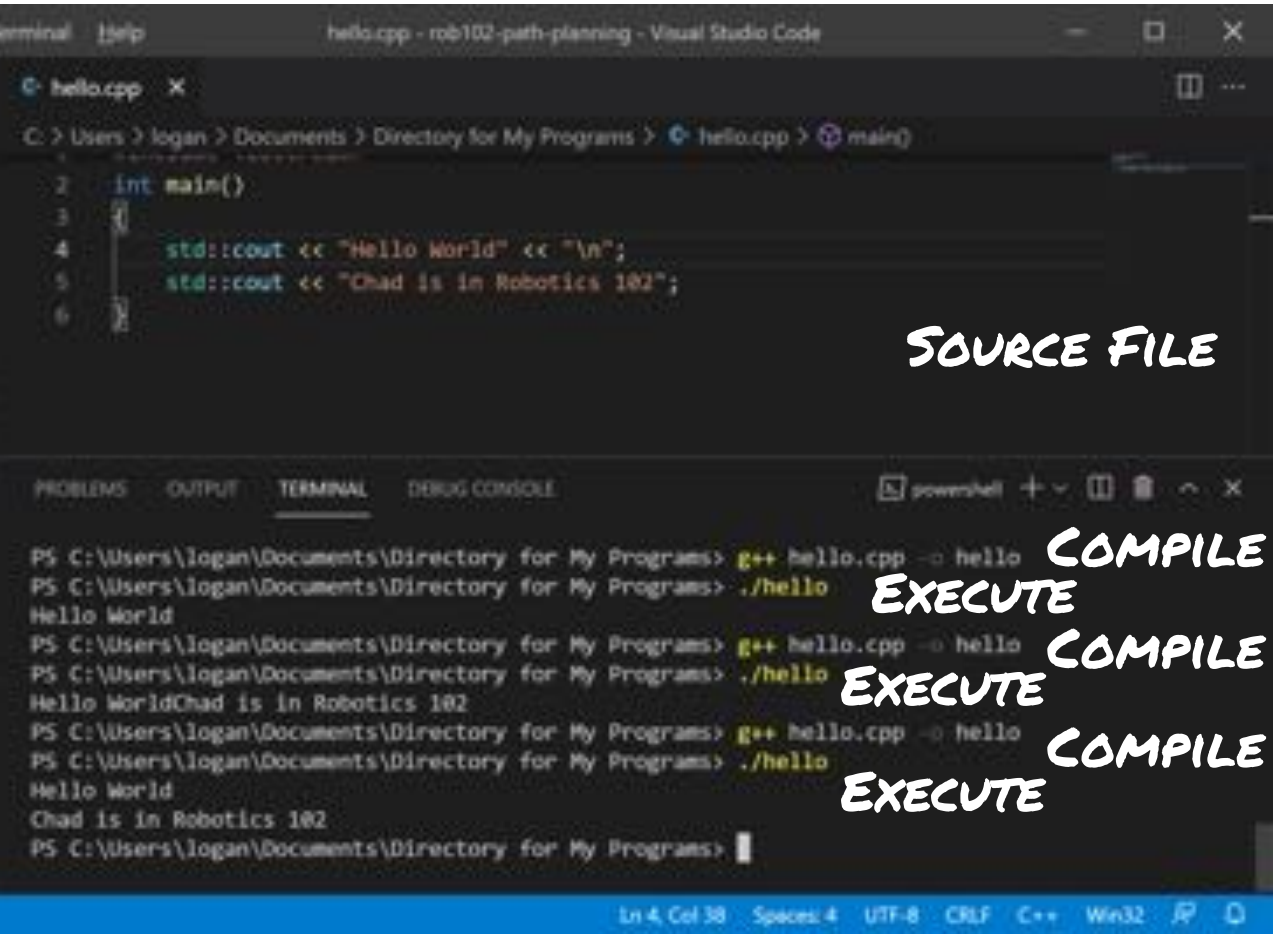
VS Code is an Integrated Development Environment (or IDE)

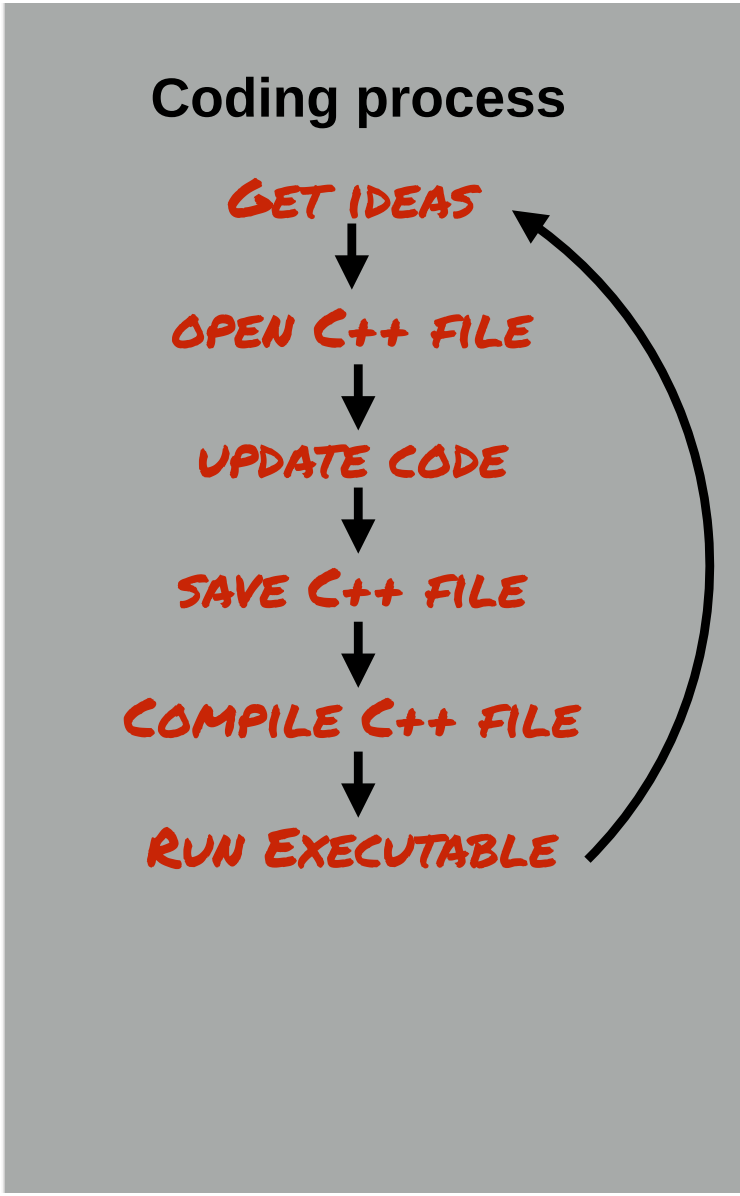



hello.cpp - rob102-path-planning - Visual Studio Code
hello.cpp
C: > Users > logan > Documents > Directory for My Programs > hello.cpp > main()
int main()
{
std::cout << "Hello World" << "\n";
std::cout << "Chad is in Robotics 102";
}
Source File
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
powershell
PS C:\Users\logan\Documents\Directory for My Programs> g++ hello.cpp -o hello
Compile
PS C:\Users\logan\Documents\Directory for My Programs> ./hello
Execute
Hello World
PS C:\Users\logan\Documents\Directory for My Programs> g++ hello.cpp -o hello
Compile
PS C:\Users\logan\Documents\Directory for My Programs> ./hello
Execute
Hello WorldChad is in Robotics 102
PS C:\Users\logan\Documents\Directory for My Programs> g++ hello.cpp -o hello
Compile
PS C:\Users\logan\Documents\Directory for My Programs> ./hello
Execute
Hello World
Chad is in Robotics 102
PS C:\Users\logan\Documents\Directory for My Programs>
Ln 4, Col 38
Spaces: 4
UTF-8
CRLF
C++
Win32

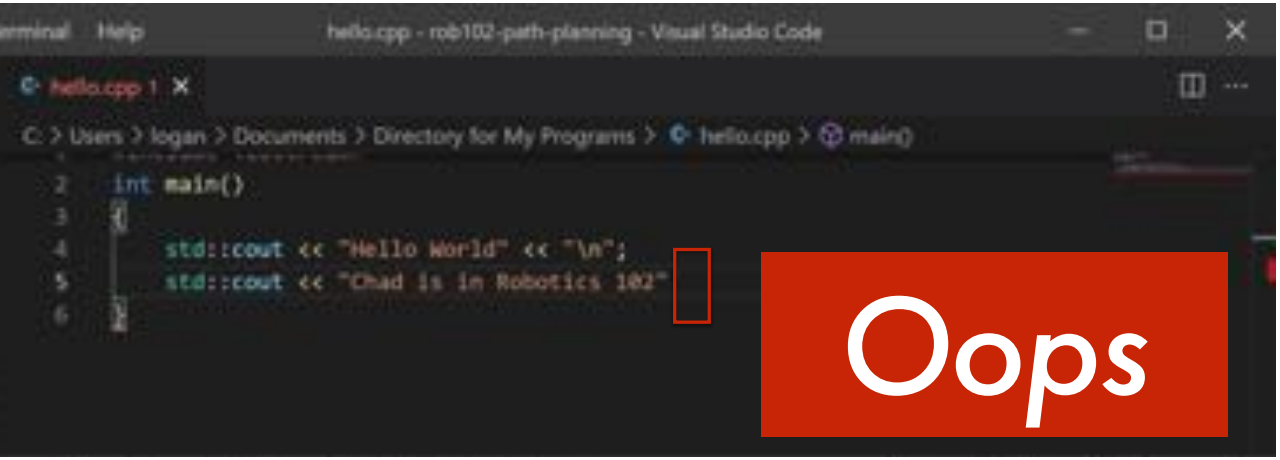

Coding process
Get ideas
Open C++ file
Update code
Save C++ file
Compile C++ file
Run Executable

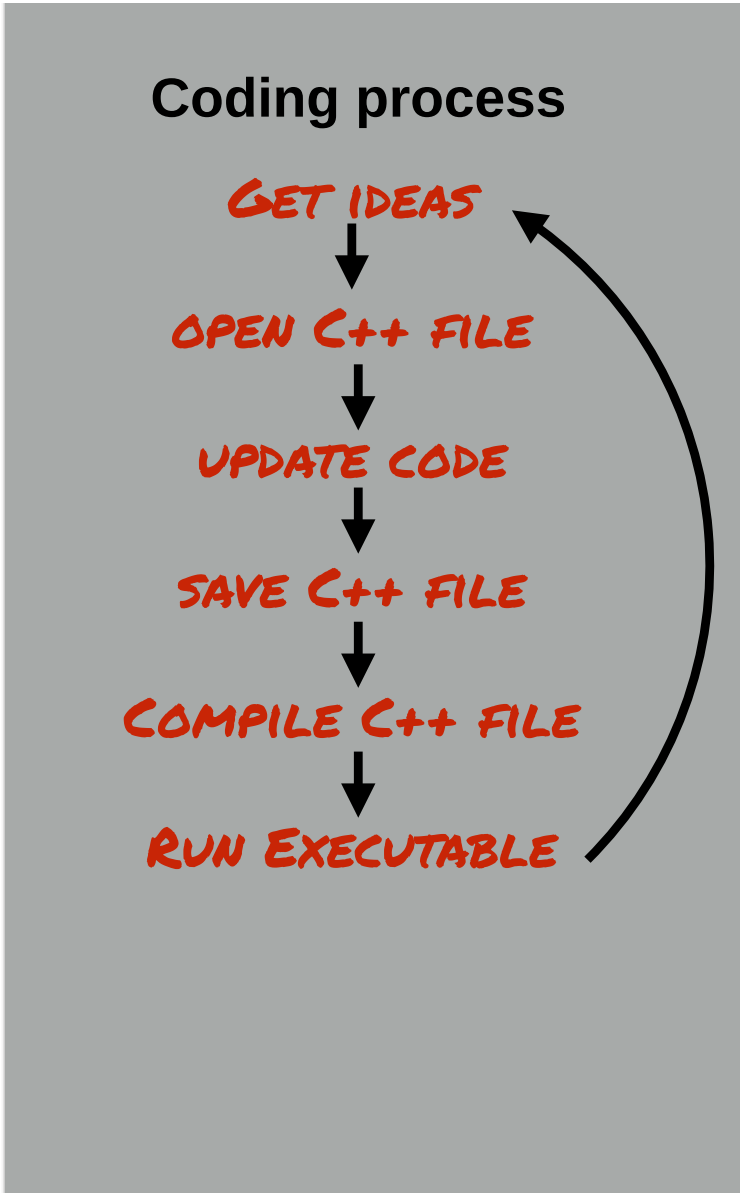
What happens if I make a mistake?



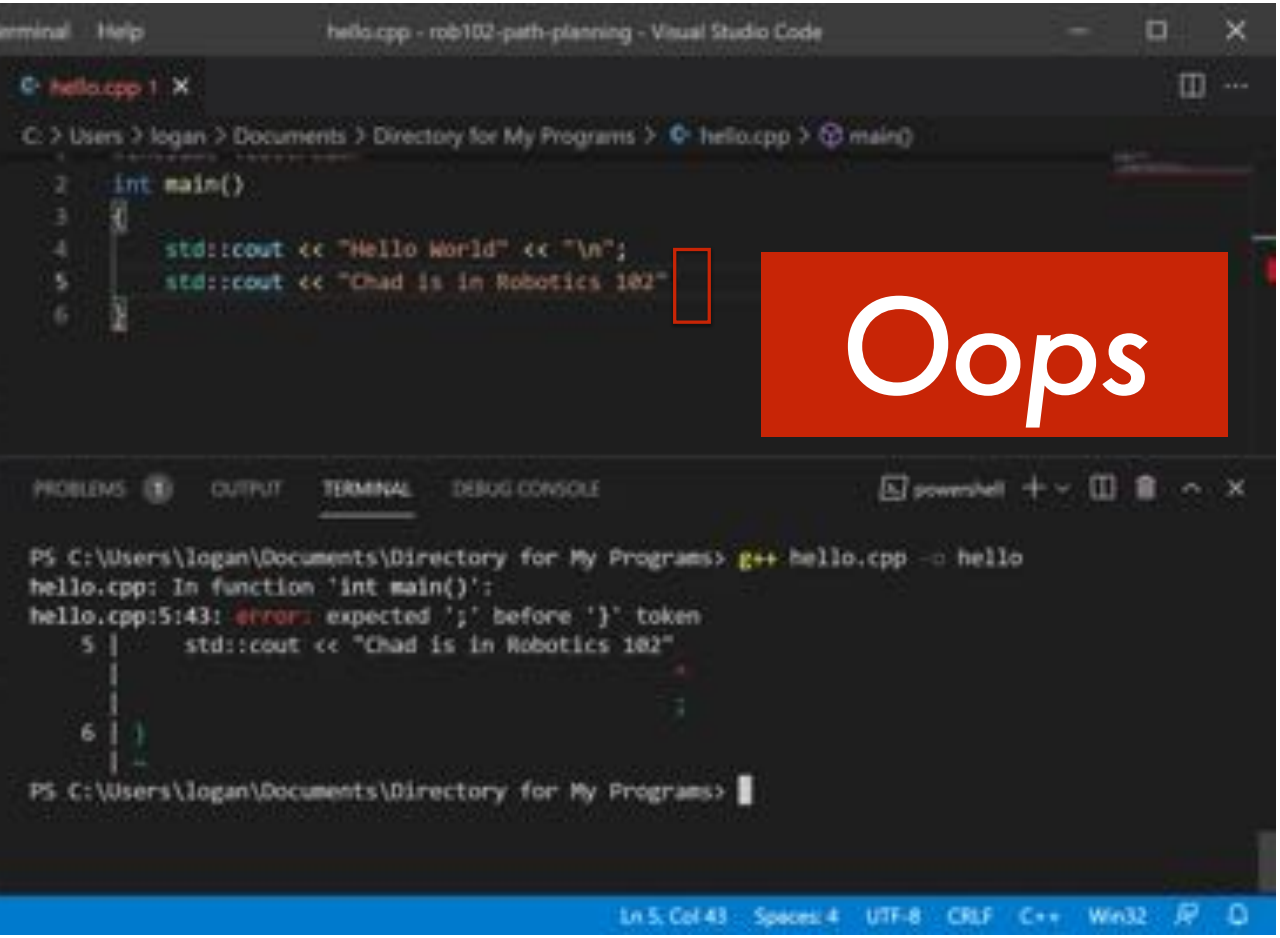


Suppose a semicolon is forgotten

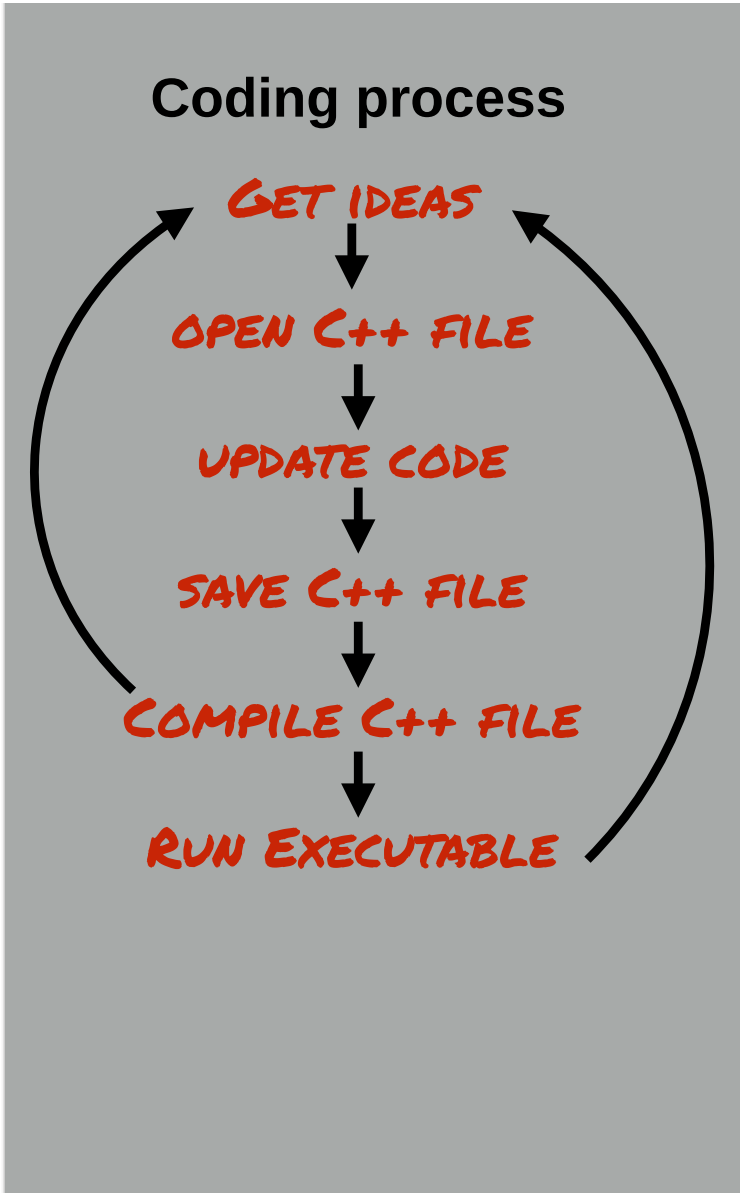




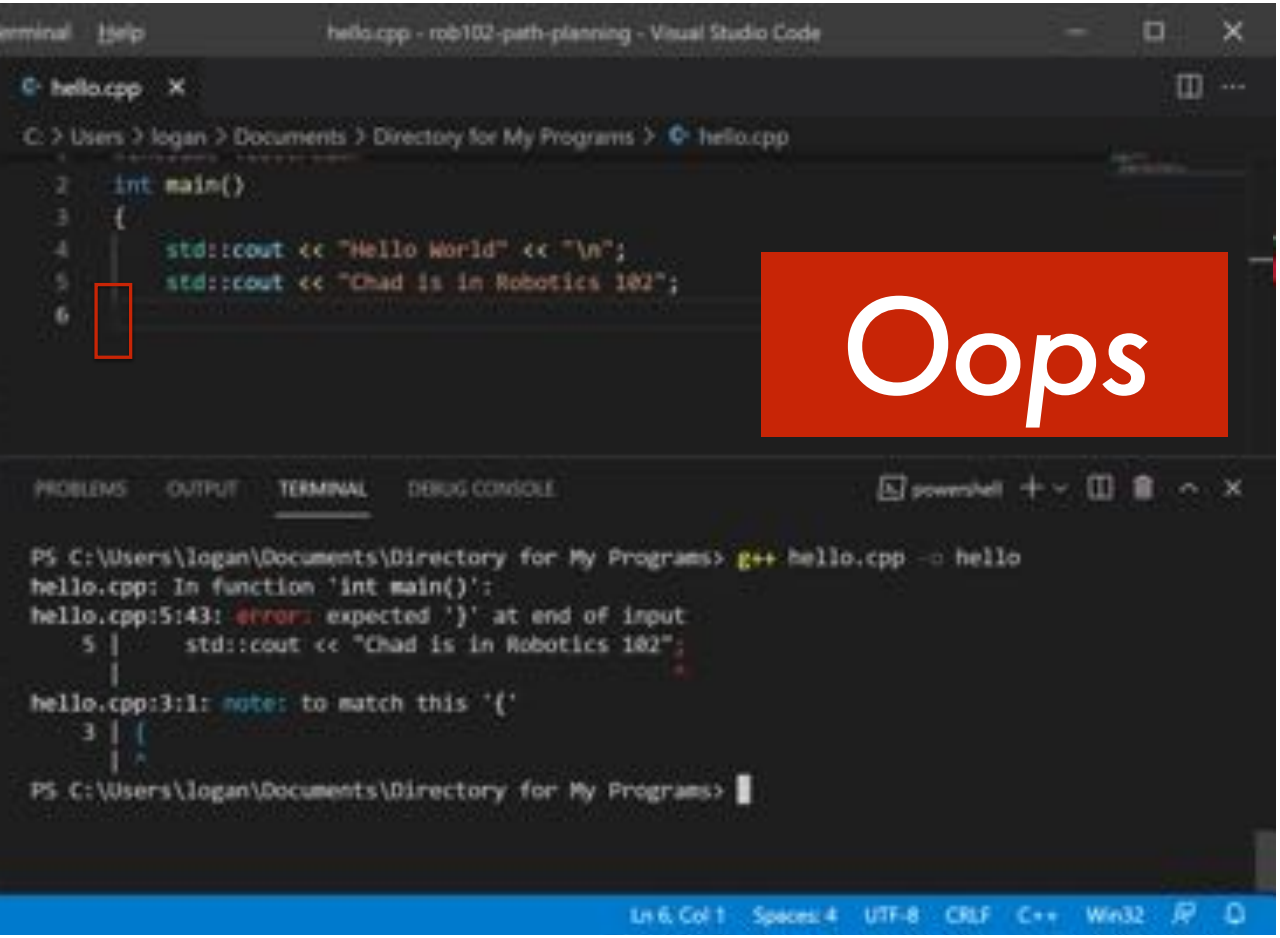
Suppose a semicolon is forgotten



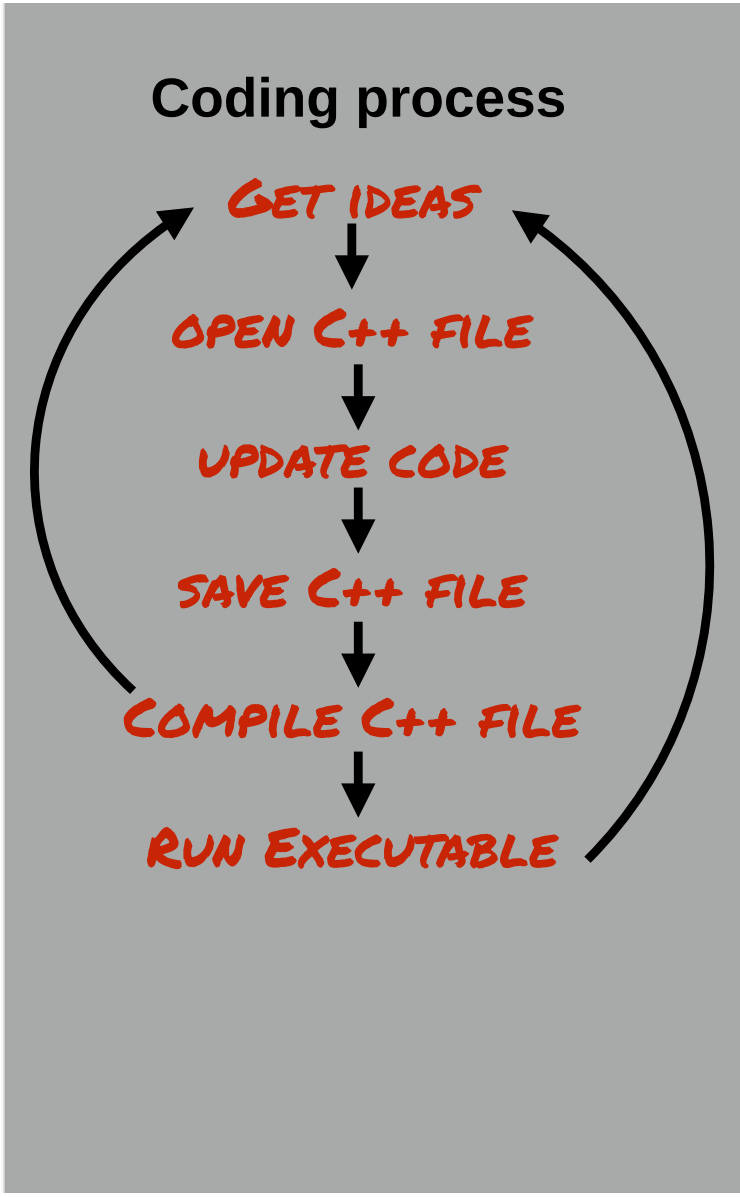
Compilation will fail with an error



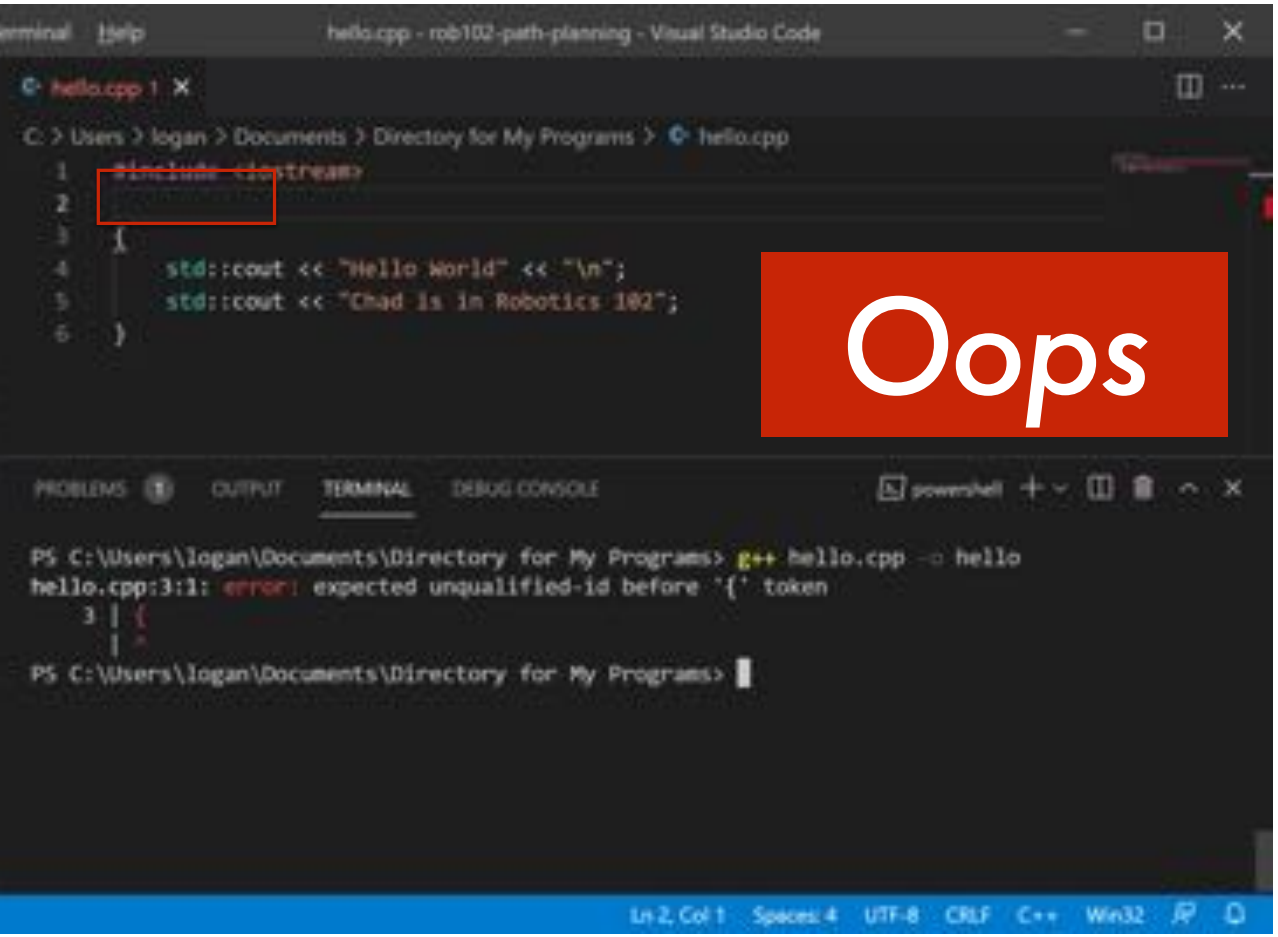
Suppose a scoping brace is forgotten



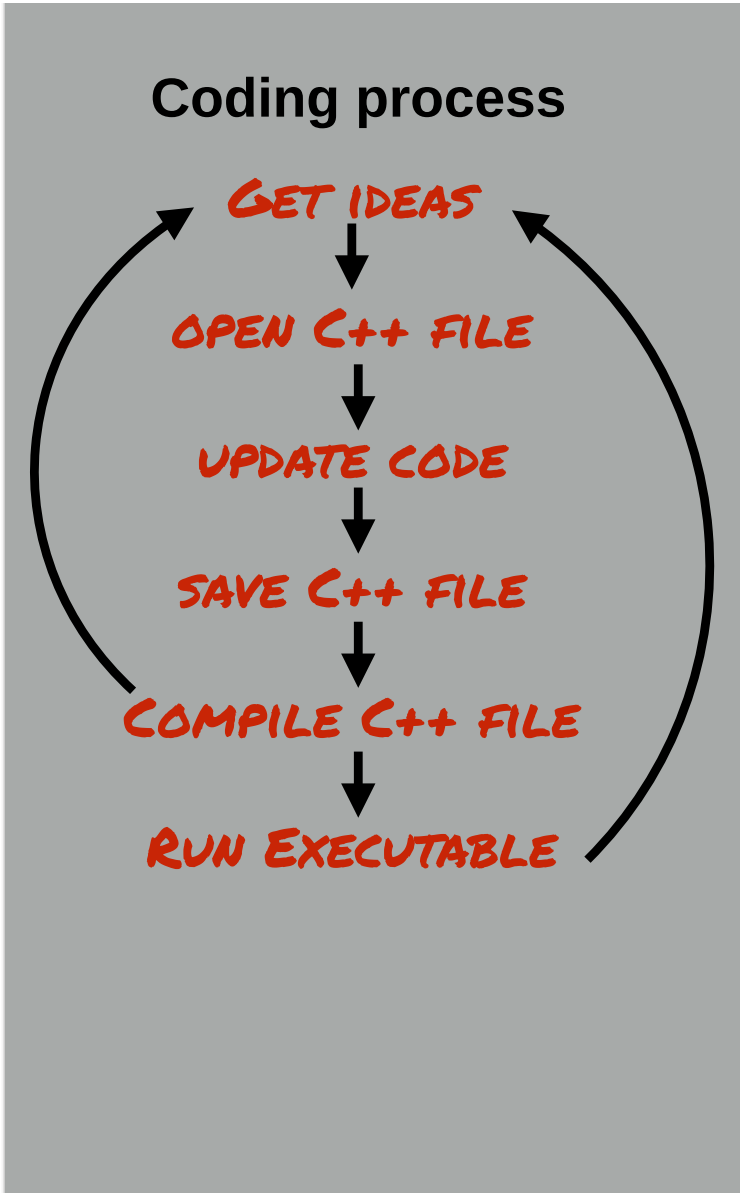
Compilation will fail with an error



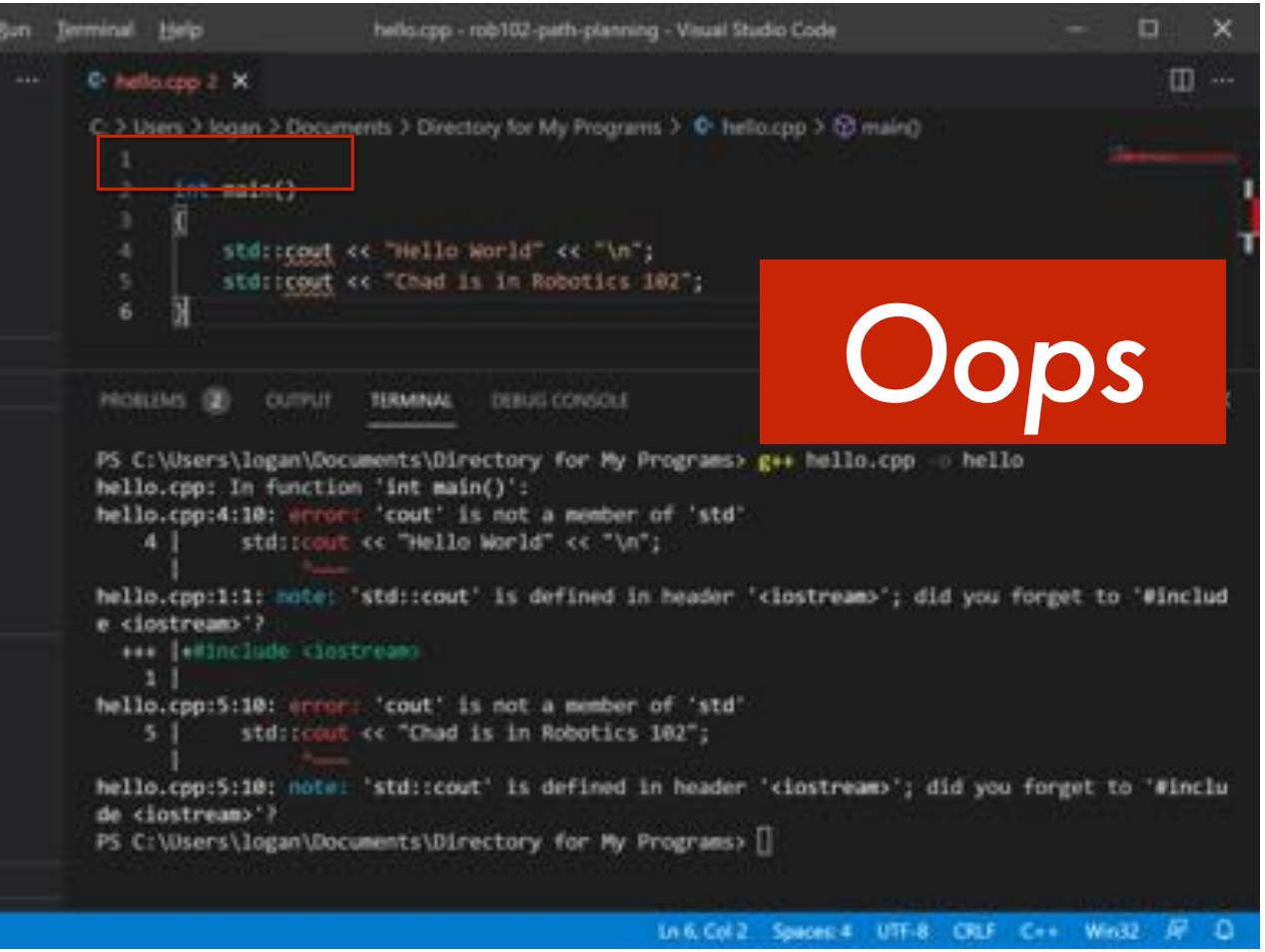
Suppose main function is forgotten



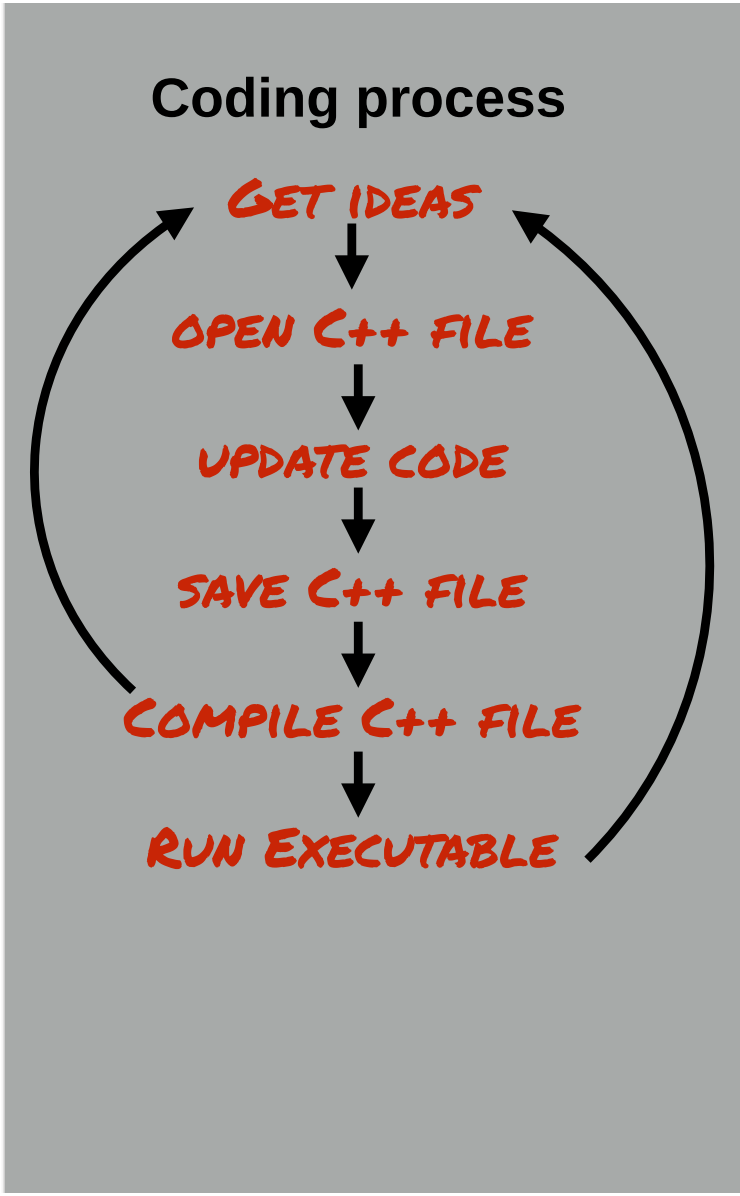
Compilation will fail with an error



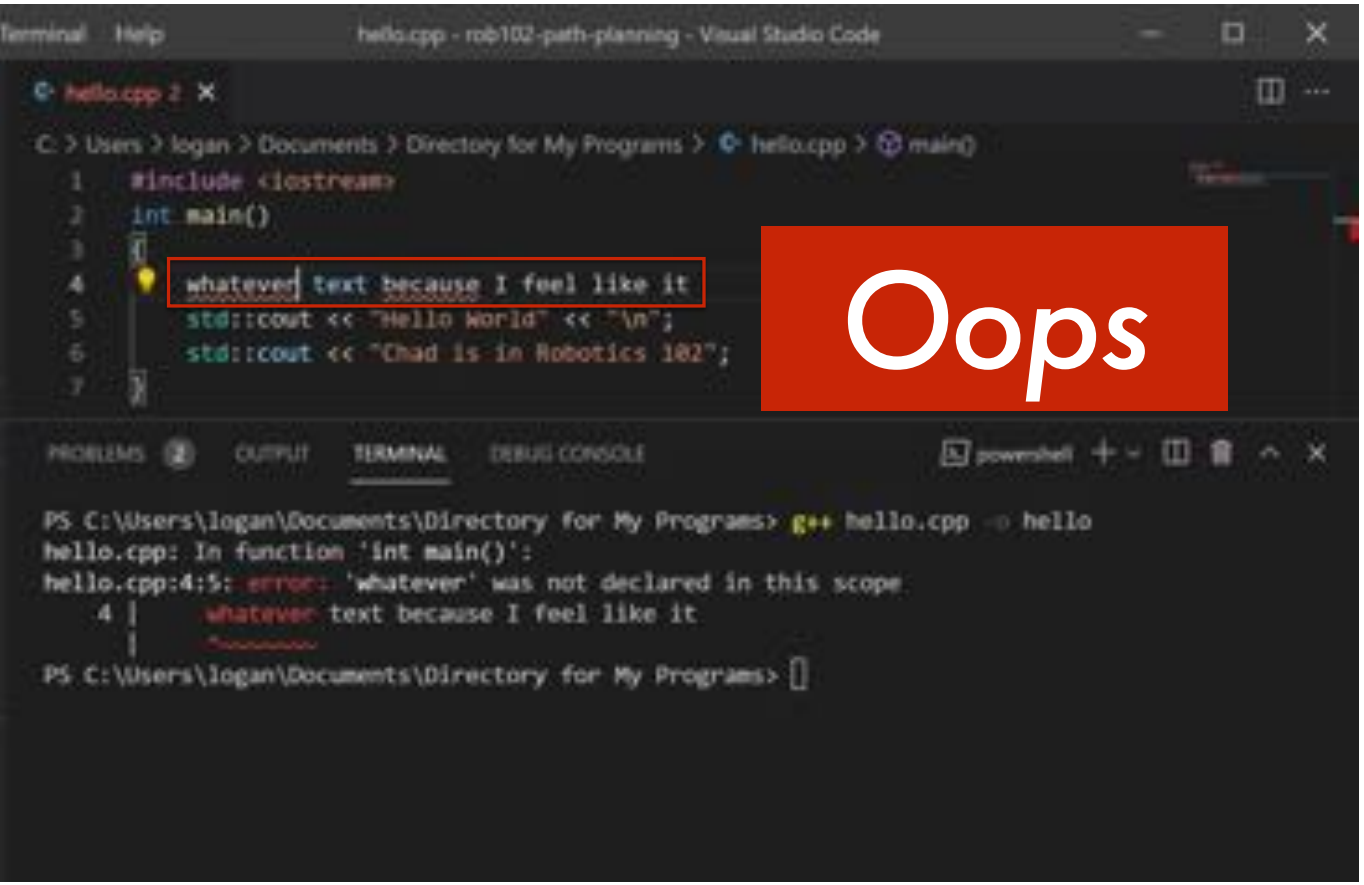
Suppose needed library is forgotten



Compilation will fail with errors

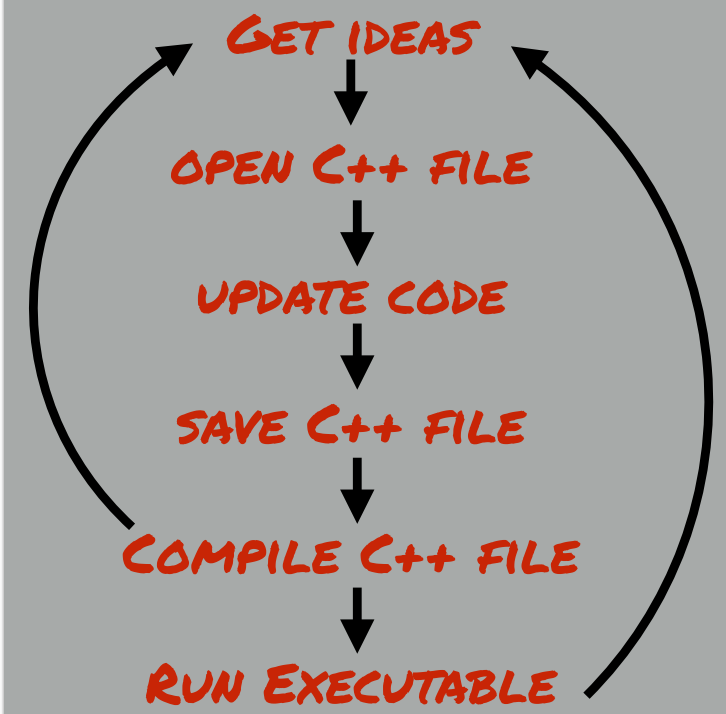


Suppose I am just careless

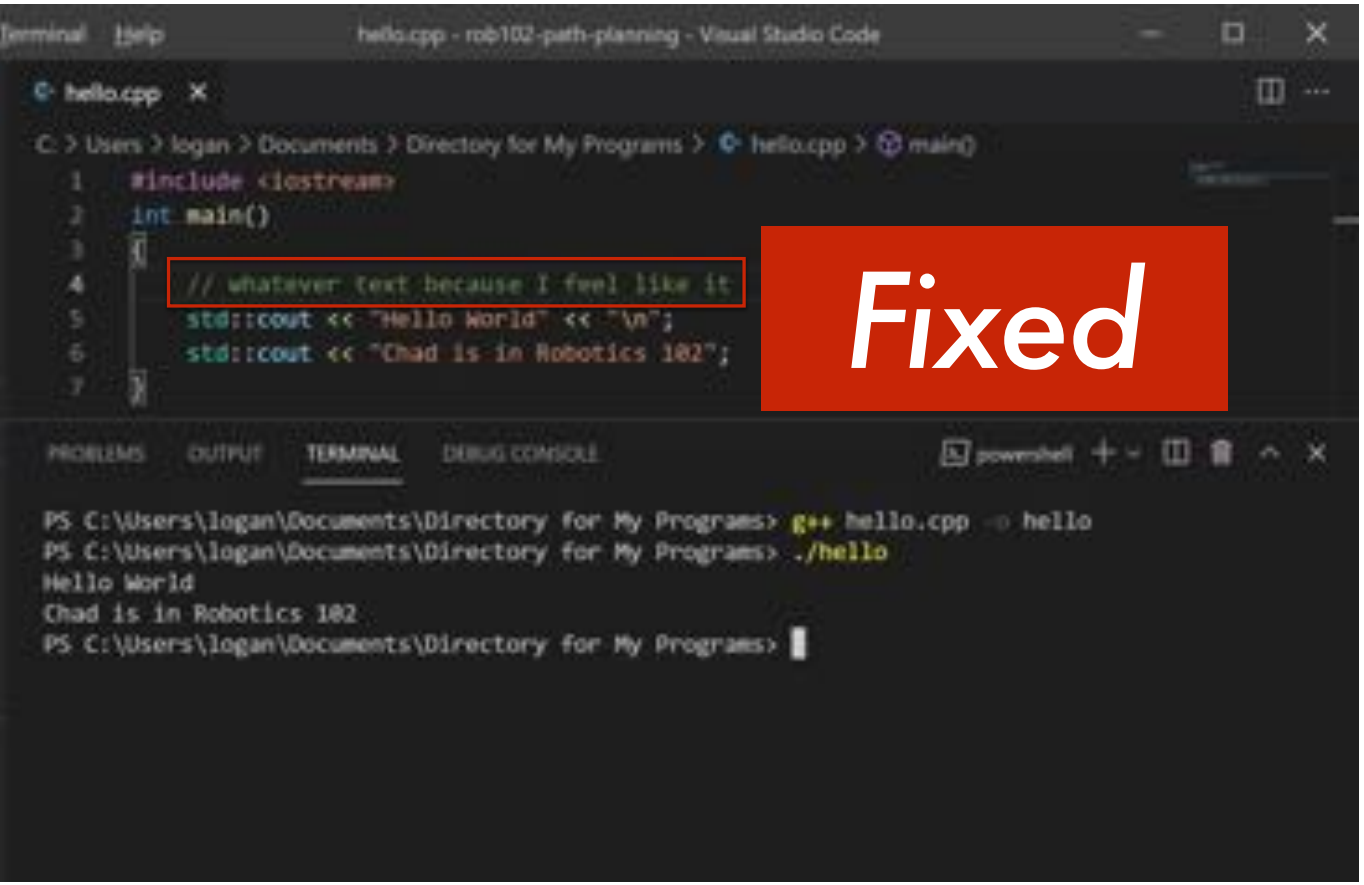


Compilation will fail with an error

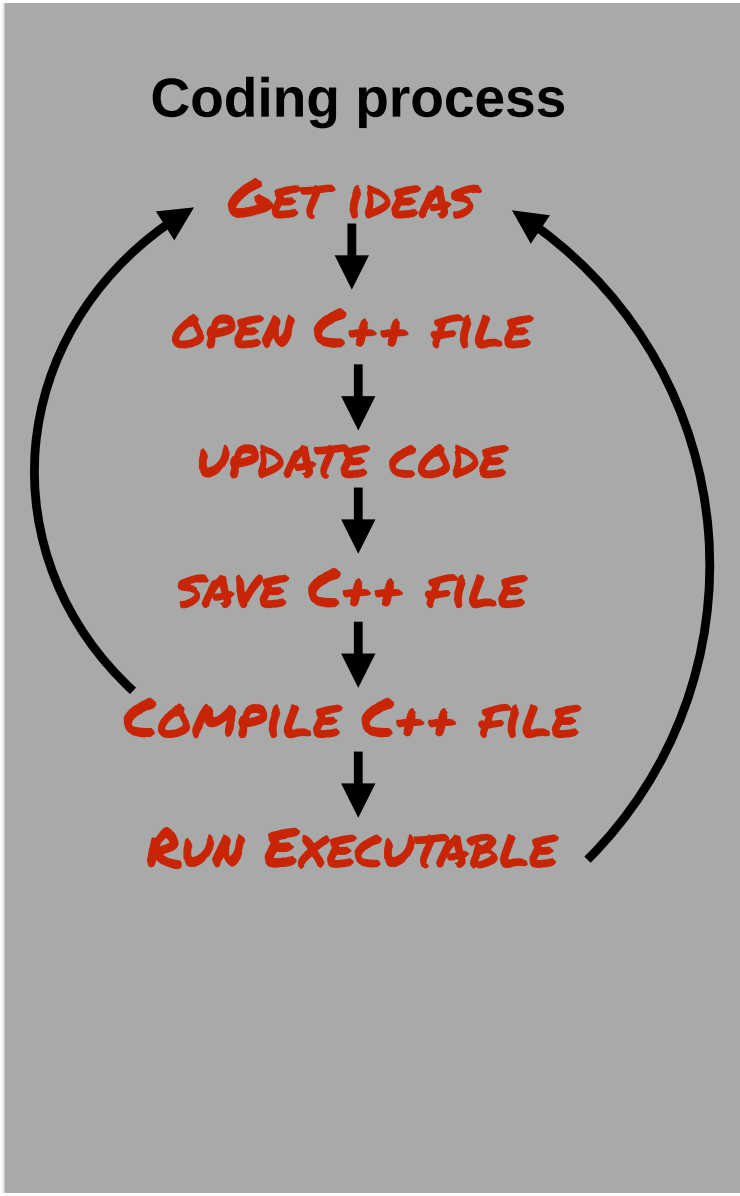
Coding process



Suppose I am just careless



That can be “commented out”



C++ Comments

Comments are ignored by the compiler and not included in the program

```
#include <iostream>
/*
   This is a multi-line comment. It is ignored by my program.
   This is our first program.
   The program code below prints messages to the screen.
*/

int main()
{
   std::cout << "Hello World" << "\n";  // A single-line comment
   std::cout << "Chad is in Robotics 102";  // "\n" creates a new line
}
```

hello.cpp

Anything after double slashes on a line is ignored as a comment

C++ Comments

Comments are ignored by the compiler and not included in the program

Anything in between delimeters /* and */ is ignored as a comment

```
#include <iostream>
/*
   This is a multi-line comment. It is ignored by my program.
   This is our first program.
   The program code below prints messages to the screen.
*/

int main()
{
   std::cout << "Hello World" << "\n";  // A single-line comment
   std::cout << "Chad is in Robotics 102";  // "\n" creates a new line
}
```

hello.cpp

Anything after double slashes on a line is ignored as a comment

Your code is a battleground

```
#include <iostream>
/*
   This is a multi-line comment. It is ignored by my program.
   This is our first program.
   The program code below prints messages to the screen.
*/

int main()
{
   std::cout << "Hello World” << “\n”;  // A single-line comment
   std::cout << "Chad is in Robotics 102”;  // "\n" creates a new line
}
```

hello.cpp

Your code is a battleground

```
#include <iostream>
/* Hello World - A first C++ Program
   Copyright 2021 Odest Chadwicke Jenkins at the University of Michigan
   Licensed under Michigan Honor License in the LICENSE file and
   available to view at https://autorob.org/MichiganHonorLicense.txt
*/

int main()
{
   std::cout << "Hello World" << "\n";  // A single-line comment
   std::cout << "Chad is in Robotics 102";  // "\n" creates a new line
}
```

hello.cpp

```
#include <iostream>
/* Hello World - A first C++ Program
   Copyright 2021 Odest Chadwicke Jenkins at the University of Michigan
   Licensed under Michigan Honor License in the LICENSE file and
   available to view at https://autorob.org/MichiganHonorLicense.txt
*/

int main()
{
   std::cout << "Hello World” << “\n”;  // A single-line comment
   std::cout << "Chad is in Robotics 102”;  // "\n" creates a new line
}
```

hello.cpp

LICENSE

```
The Michigan Honor License
This unvetted license below is called the "Michigan Honor License" as the 3-Clause BSD
License plus two clauses for academic integrity.

Copyright <YEAR> <COPYRIGHT HOLDER>

Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer.
```

```
#include <iostream>
/* Hello World - A first C++ Program
   Copyright 2021 Odest Chadwicke Jenkins at the University of Michigan
   Licensed under Michigan Honor License in the LICENSE file and
   available to view at https://autorob.org/MichiganHonorLicense.txt
*/

int main()
{
   std::cout << "Hello World" << "\n";  // A single-line comment
   std::cout << "Chad is in Robotics 102";  // "\n" creates a new line
}
```

hello.cpp

LICENSE

```
The Michigan Honor License
This unvetted license below is called the "Michigan Honor License" as the 3-Clause BSD
License plus two clauses for academic integrity.

Copyright <YEAR> <COPYRIGHT HOLDER>

Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer.
```

LICENSE

The Michigan Honor License
This unvetted license below is called the "Michigan Honor License" as the 3-Clause BSD License plus two clauses for academic integrity.

BSD 3-Clause License

https://opensource.org/licenses/BSD-3-Clause

t modification, are

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

4. Manuscripts and publications using this source code or its binary forms must properly cite the this project and its author(s).

5. Redistributions of source code, if used for credit in an academic program, must retain the academic integrity honor pledge below.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF

LICENSE

```
The Michigan Honor License
This unvetted license below is called the "Michigan Honor License" as the 3-Clause BSD
License plus two clauses for academic integrity.

Copyright <YEAR> <COPYRIGHT HOLDER>

Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of
conditions and the following disclaimer in the documentation and/or other materials provided
with the distribution.

mes of its contributors may be used to
without specific prior written
```

Michigan Engineering Honor Code

https://bulletin.engin.umich.edu/rules/

```
4. Redistributions of source code, if used for credit in an academic program, must retain
the following honor pledge, and only append the names all individuals who contributed
modifications.

the academic integrity honor pledge below.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
```

LICENSE

The Michigan Honor License
This unvetted license below is called the "Michigan Honor License" as the 3-Clause BSD License plus two clauses for academic integrity.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

Attribution similar to CC BY 4.0 License
https://creativecommons.org/licenses/by/4.0/

inary forms must properly

5. Manuscripts and publications using this source code or its binary forms must properly cite this project and its author(s).

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF

Share your code with the world

and your future self

Compiler

Executable

IDEAS

```
PS C:\Users\logan\Documents\Directory for My Programs> g++ hello.cpp -o hello
PS C:\Users\logan\Documents\Directory for My Programs> ./hello
Hello World
PS C:\Users\logan\Documents\Directory for My Programs> g++ hello.cpp -o hello
PS C:\Users\logan\Documents\Directory for My Programs> ./hello
Hello WorldChad is in Robotics 102
PS C:\Users\logan\Documents\Directory for My Programs> g++ hello.cpp -o hello
PS C:\Users\logan\Documents\Directory for My Programs> ./hello
Hello World
Chad is in Robotics 102
PS C:\Users\logan\Documents\Directory for My Programs>
```

Text editor

```
int main()
{
    std::cout << "Hello World" << "\n";
    std::cout << "Chad is in Robotics 102";
}
```

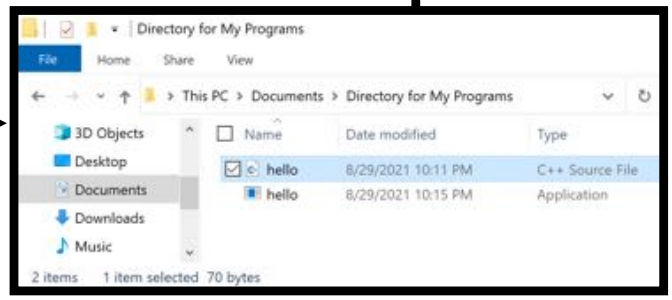
hello.cpp

OPEN FILE

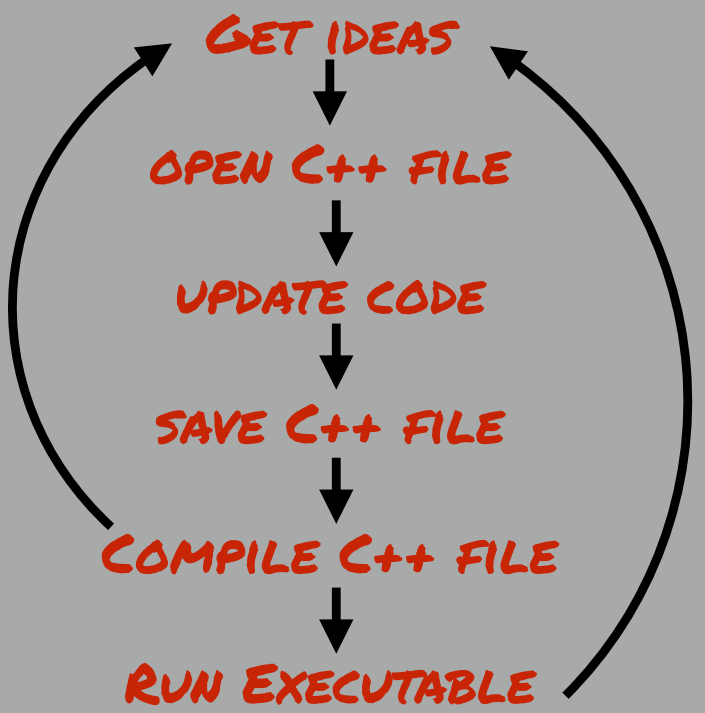
SAVE FILE

Source code

COMPILE AND EXECUTE



Coding process



Compiler

Executable

Ideas

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE                    powershell

PS C:\Users\logan\Documents\Directory for My Programs> g++ hello.cpp -o hello
PS C:\Users\logan\Documents\Directory for My Programs> ./hello
Hello World
PS C:\Users\logan\Documents\Directory for My Programs> g++ hello.cpp -o hello
PS C:\Users\logan\Documents\Directory for My Programs> ./hello
Hello WorldChad is in Robotics 102
PS C:\Users\logan\Documents\Directory for My Programs> g++ hello.cpp -o hello
PS C:\Users\logan\Documents\Directory for My Programs> ./hello
Hello World
Chad is in Robotics 102
PS C:\Users\logan\Documents\Directory for My Programs>
```

Text editor

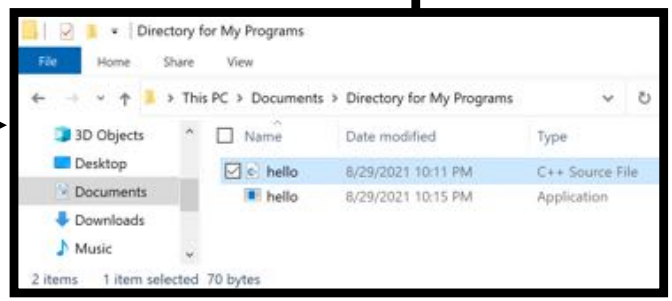
```
2  int main()
3  {
4      std::cout << "Hello World" << "\n";
5      std::cout << "Chad is in Robotics 102";
6  }
```

hello.cpp

Compile

And Execute

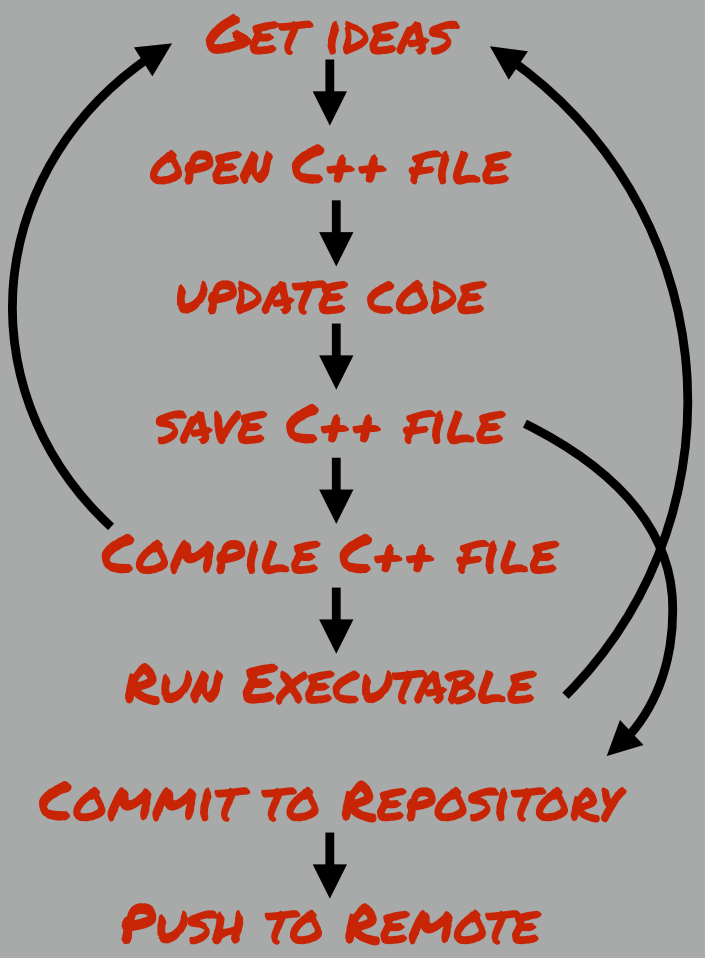
Open file



Save file

Source code

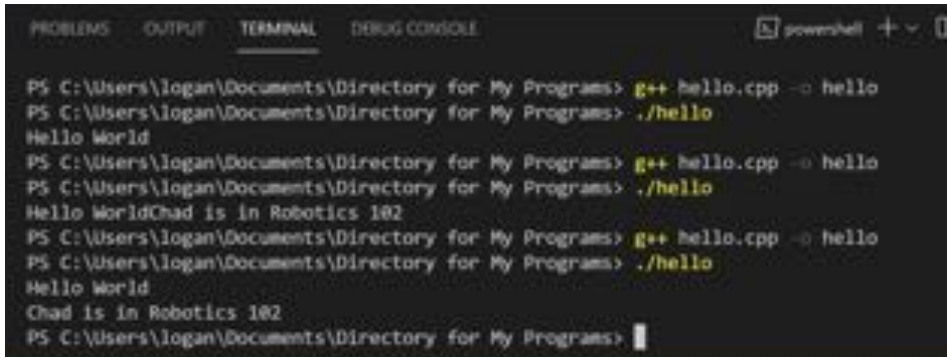
Coding process



Compiler

Executable

IDEAS



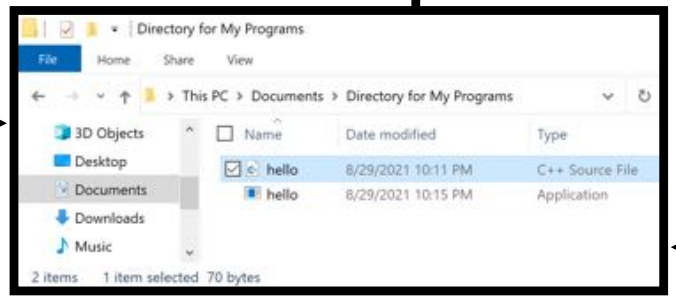
Text editor

hello.cpp

OPEN FILE

SAVE FILE

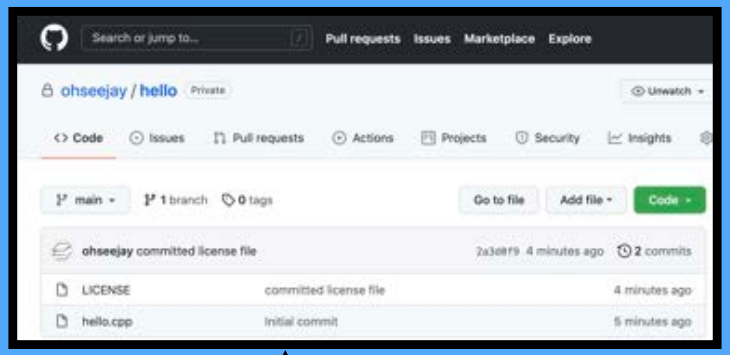
Source code



COMPILE AND EXECUTE

git repository

store history of code changes and collaborate with others



PUSH REPO

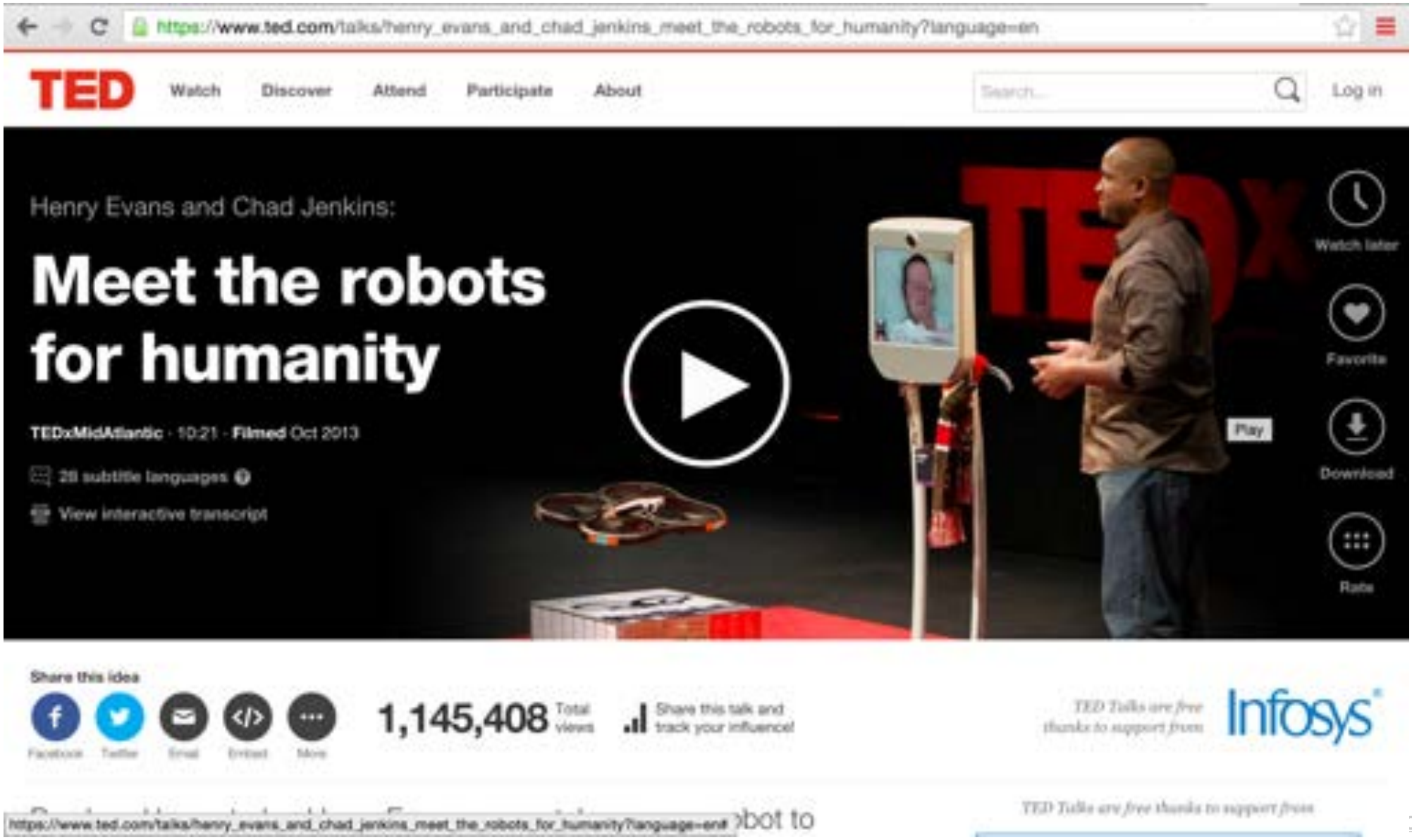
PULL REPO

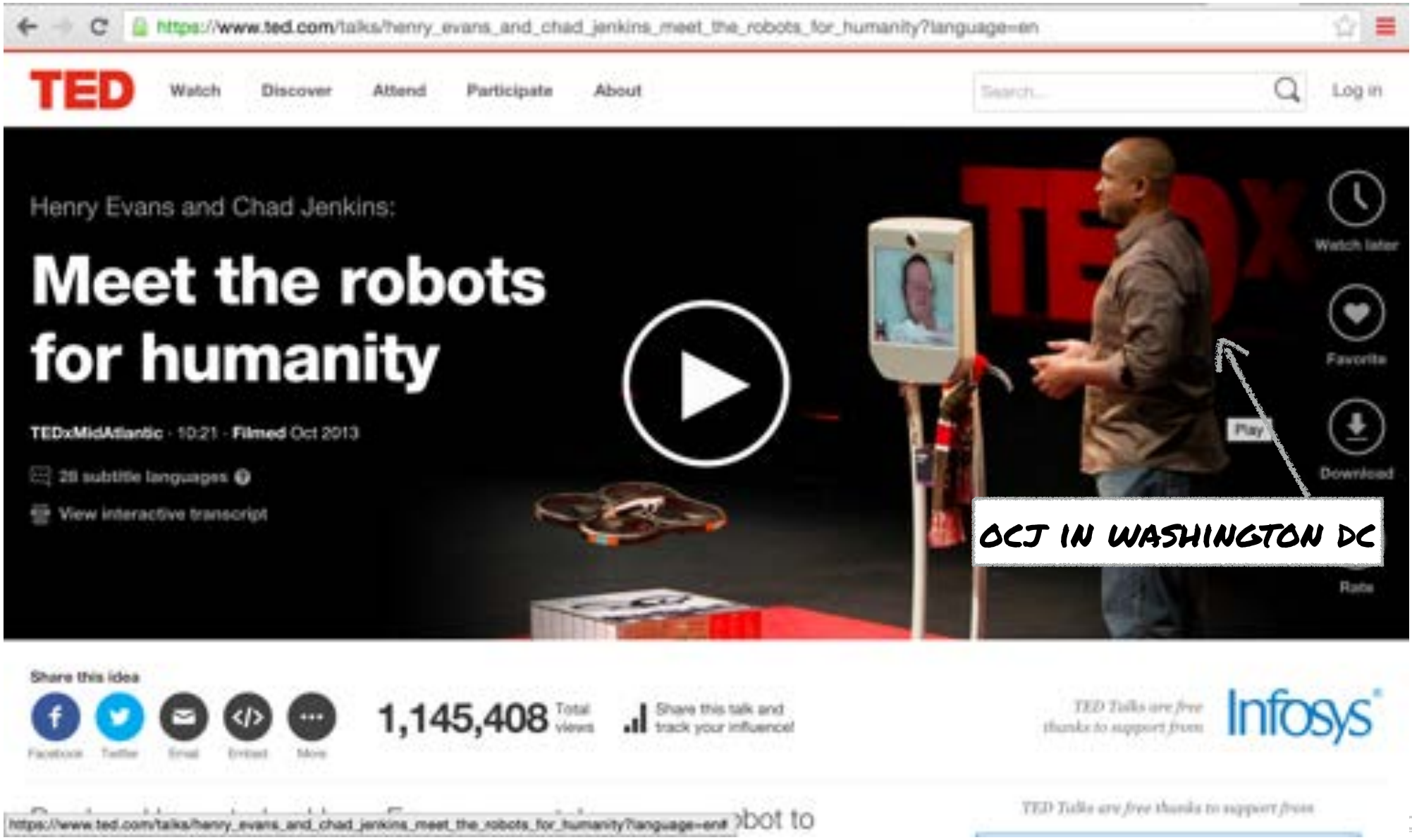
Version Control Using git

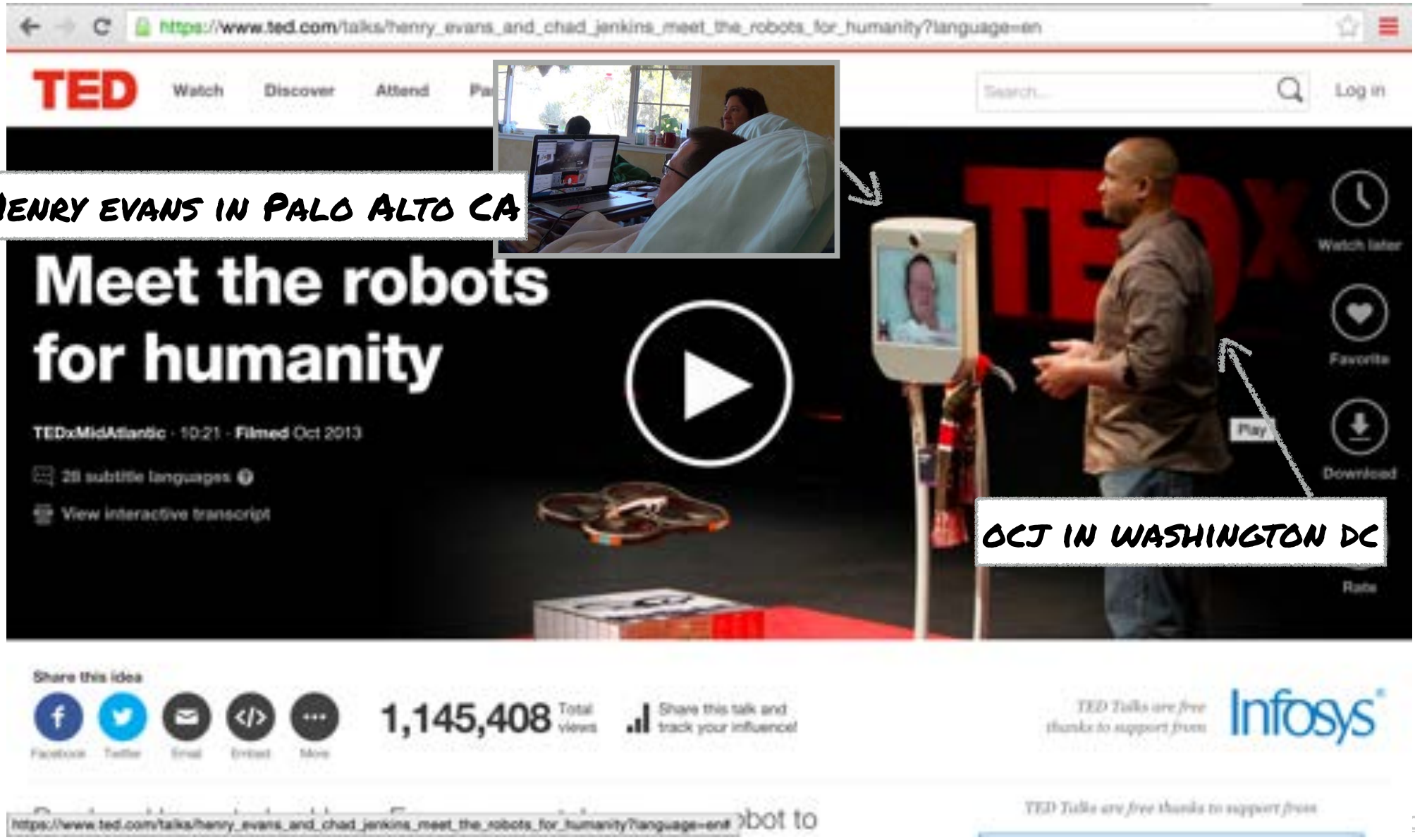
What is Version Control?

- Maintains a past history of changes for your code (or any project)
- History of changes (or “commits”) maintained in a repository
- Basic workflow
  - Code is “checked out” (or “pulled”) from a repository, then modified
  - These updates are then “checked in” (or “committed”) to the repository
  - Repository maintains history as “diffs”, the changes between before and after checking in a commit

For example… ocj’s TED talk

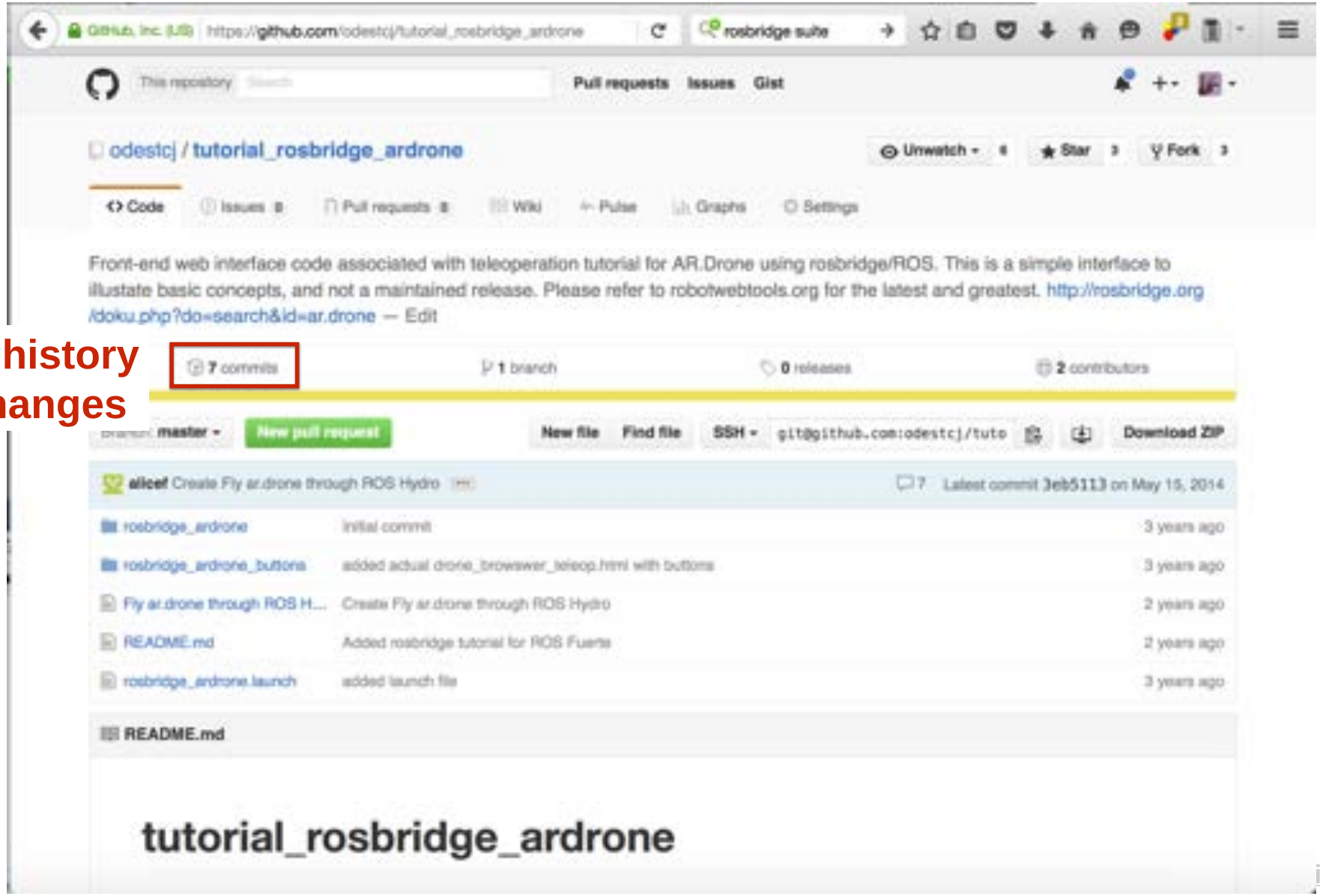

https://www.ted.com/talks/henry_evans_and_chad_jenkins_meet_the_robots_for_humanity?language=en
TED
Watch
Discover
Attend
Participate
About
Log in
Henry Evans and Chad Jenkins:
Meet the robots for humanity
TEDxMidAtlantic · 10:21 · Filmed Oct 2013
28 subtitle languages
View interactive transcript
Play
Watch later
Favorite
Download
Rate
Share this idea
Facebook
Twitter
Email
Embed
More
1,145,408 Total views
Share this talk and track your influence
TED Talks are free thanks to support from
Infosys
https://www.ted.com/talks/henry_evans_and_chad_jenkins_meet_the_robots_for_humanity?language=en#
TED Talks are free thanks to support from

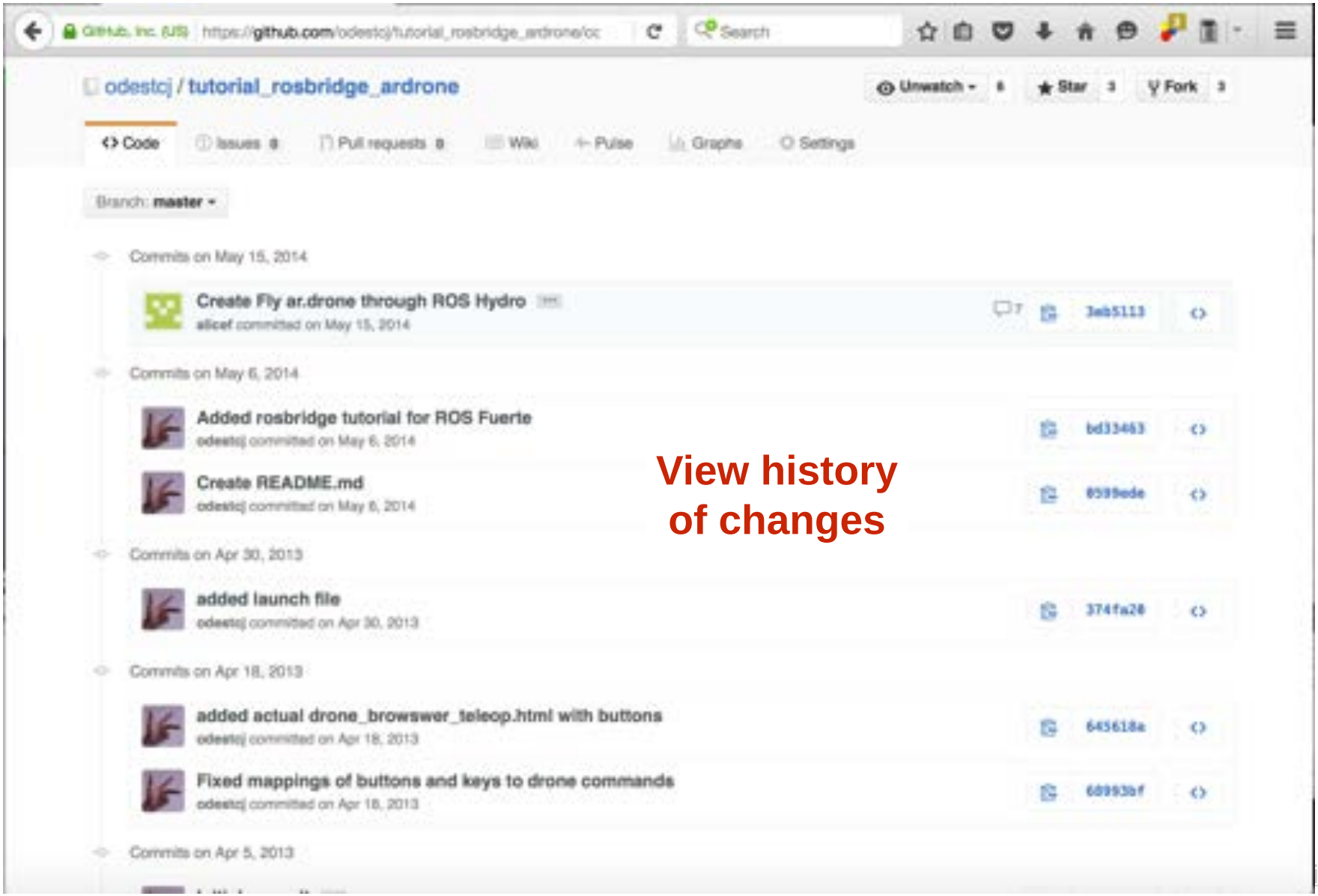

https://www.ted.com/talks/henry_evans_and_chad_jenkins_meet_the_robots_for_humanity?language=en
TED
Watch
Discover
Attend
Participate
About
Log in
Henry Evans and Chad Jenkins:
Meet the robots for humanity
TEDxMidAtlantic · 10:21 · Filmed Oct 2013
28 subtitle languages
View interactive transcript
Watch later
Favorite
Download
Play
Rate
OCJ IN WASHINGTON DC
Share this idea
1,145,408 Total views
Share this talk and track your influence
TED Talks are free thanks to support from
Infosys
https://www.ted.com/talks/henry_evans_and_chad_jenkins_meet_the_robots_for_humanity?language=en#
bot to
TED Talks are free thanks to support from


https://www.ted.com/talks/henry_evans_and_chad_jenkins_meet_the_robots_for_humanity?language=en
TED
Watch
Discover
Attend
Log in
HENRY EVANS IN PALO ALTO CA
Meet the robots for humanity
TEDxMidAtlantic · 10:21 · Filmed Oct 2013
28 subtitle languages
View interactive transcript
Watch later
Favorite
Download
Play
Rate
OCJ IN WASHINGTON DC
Share this idea
Facebook
Twitter
Email
Embed
More
1,145,408 Total views
Share this talk and track your influence
TED Talks are free thanks to support from
Infosys
https://www.ted.com/talks/henry_evans_and_chad_jenkins_meet_the_robots_for_humanity?language=en#
TED Talks are free thanks to support from


IN-BROWSER DRONE CONTROL

View history of changes




odestcj / tutorial_rosbridge_ardrone
Unwatch
Star
Fork
Code
Issues
Pull requests
Wiki
Pulse
Graphs
Settings
Branch: master
Commits on May 15, 2014
Create Fly ar.drone through ROS Hydro
alicef committed on May 15, 2014
Commits on May 6, 2014
Added rosbridge tutorial for ROS Fuerte
odestcj committed on May 6, 2014
Create README.md
odestcj committed on May 6, 2014
Commits on Apr 30, 2013
added launch file
odestcj committed on Apr 30, 2013
Commits on Apr 18, 2013
added actual drone_browswer_teleop.html with buttons
odestcj committed on Apr 18, 2013
Fixed mappings of buttons and keys to drone commands
odestcj committed on Apr 18, 2013
Commits on Apr 5, 2013
View history of changes

Large open source projects…

ROBOT WEB ARCHITECTURE

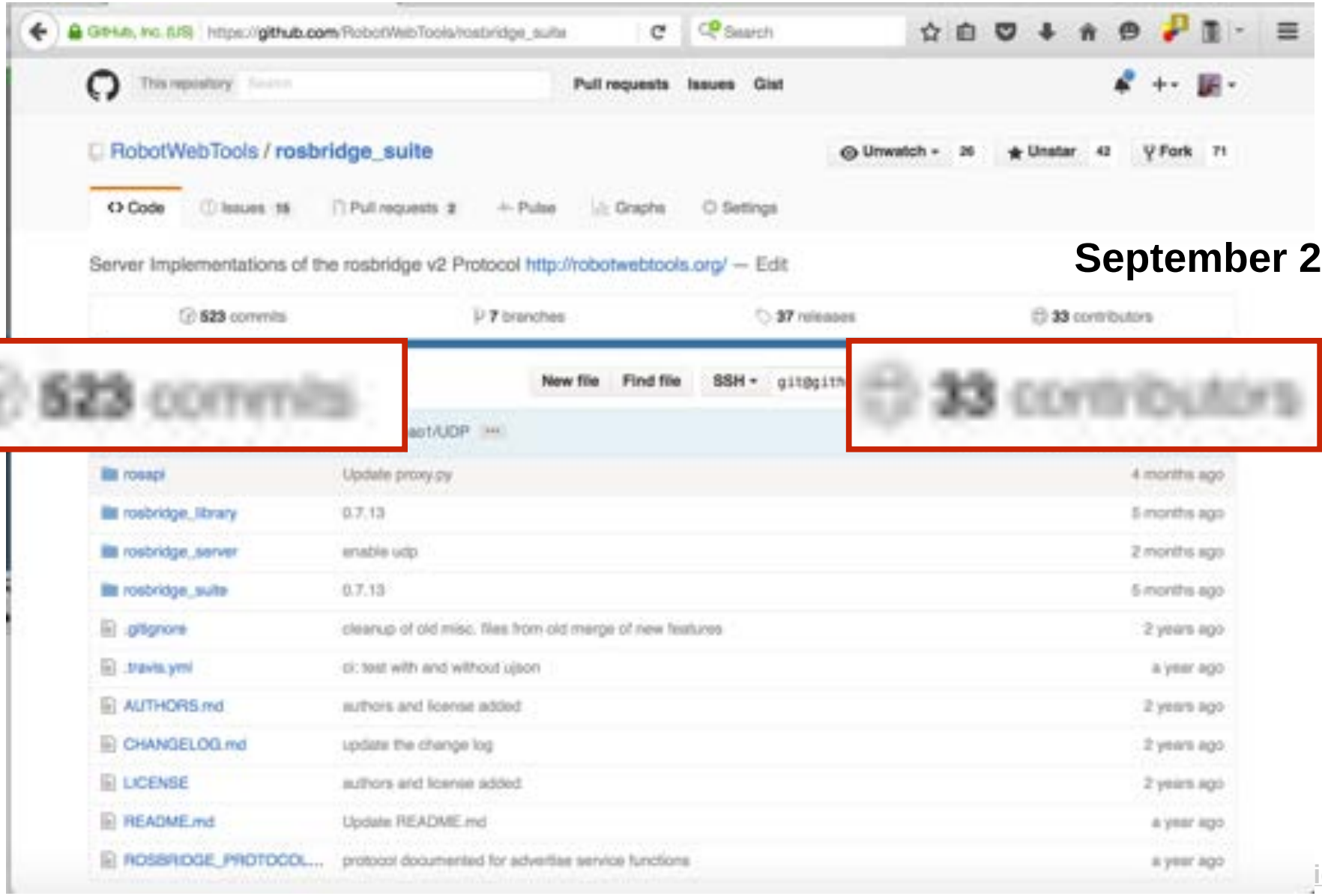
BRIDGING ROBOTS AND THE WEB

A variety of routes are available for architecting a robot web

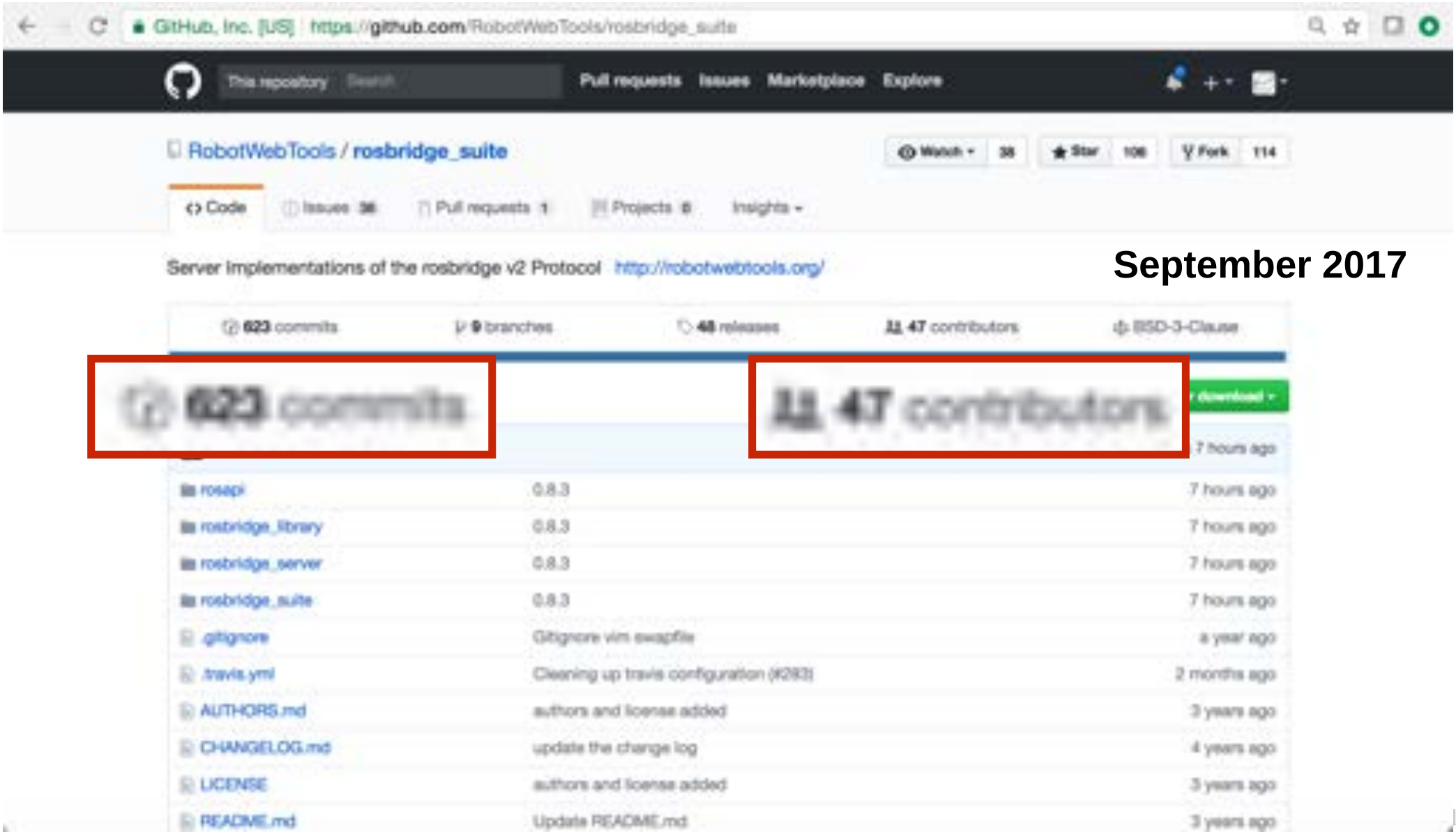
ROSBRIDGE AS A TRANSPORT

USING JSON TO SPEAK TO YOUR ROBOT

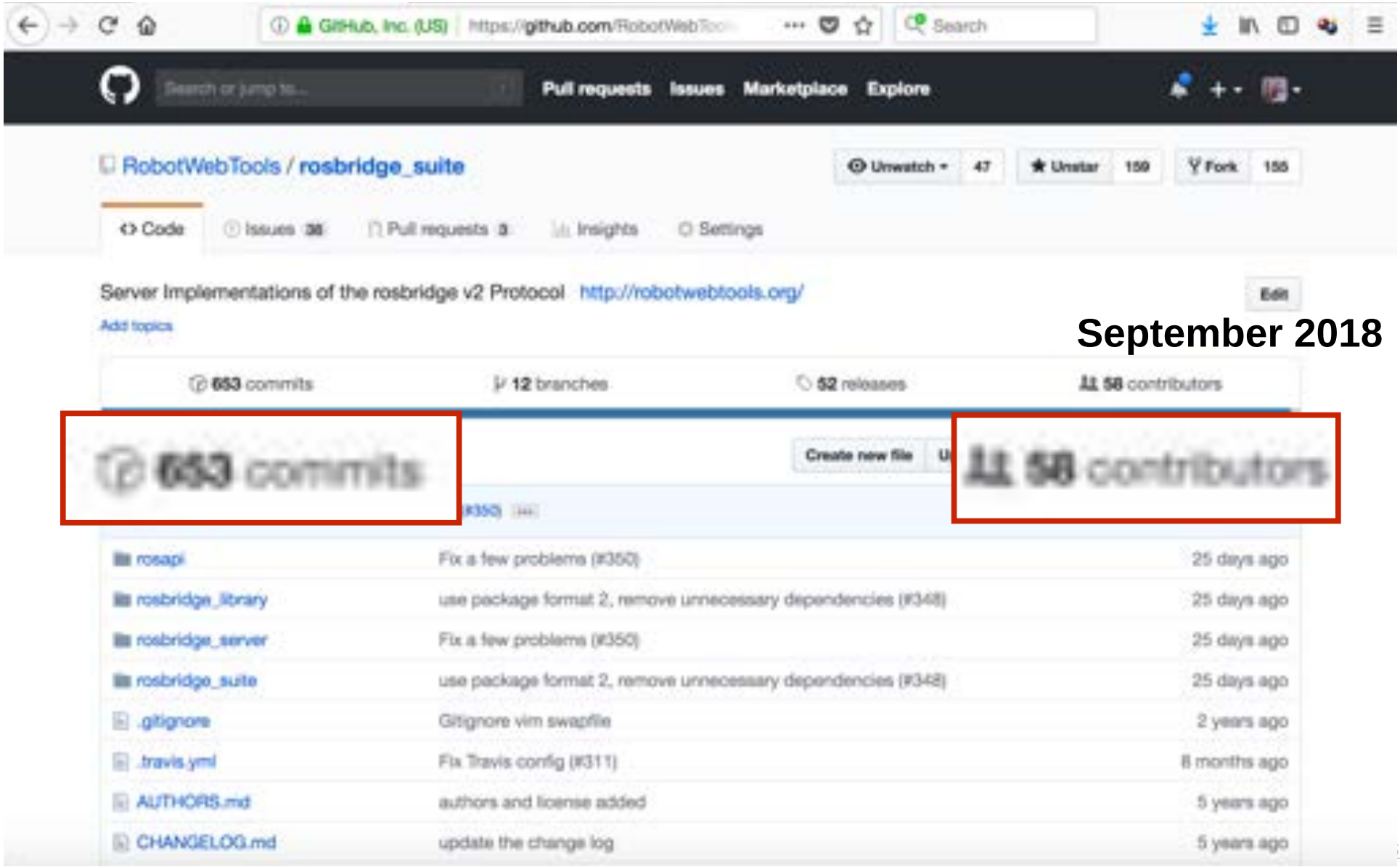
While ROS works great for applications on the robot, another layer is



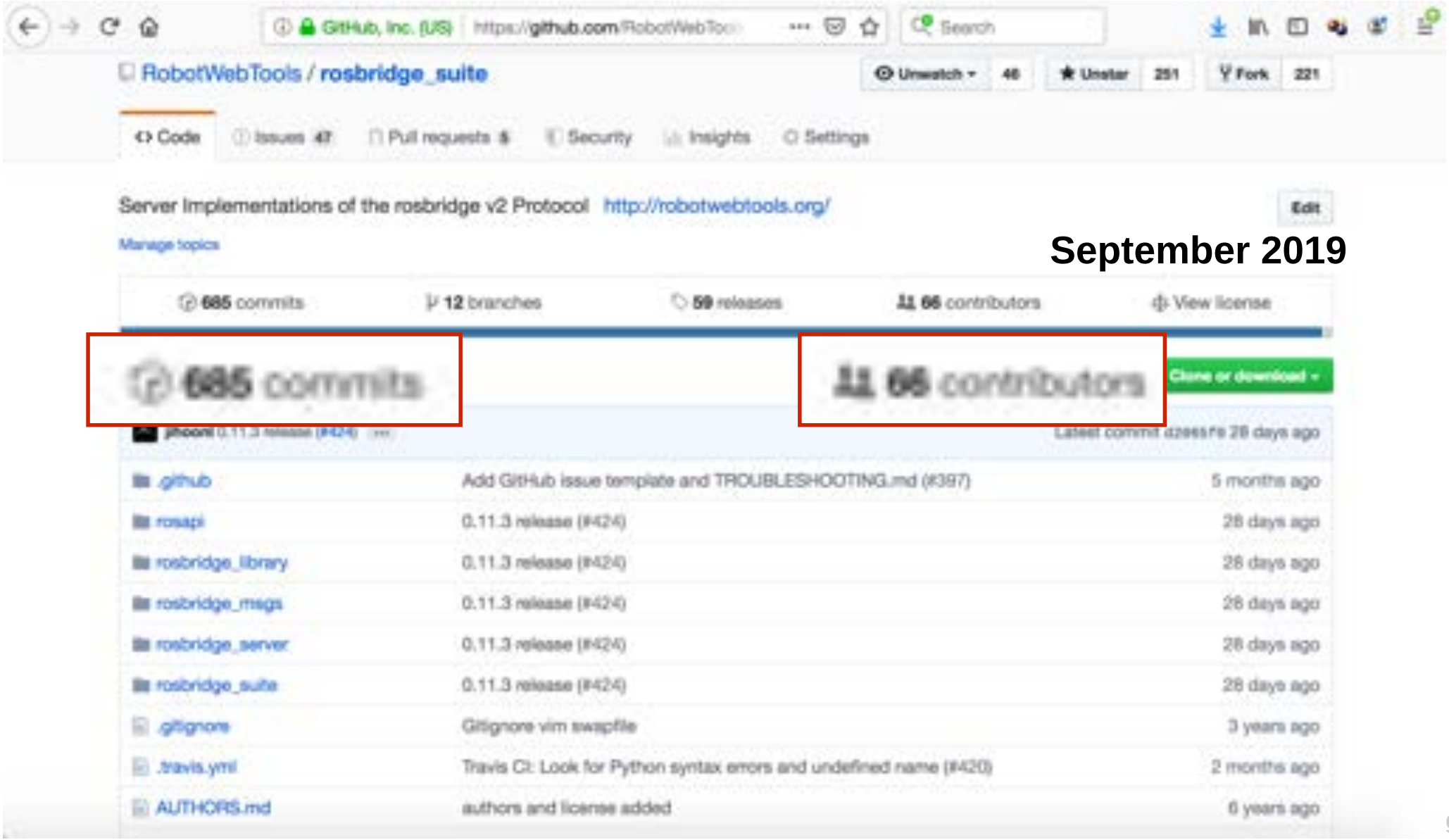
September 2016

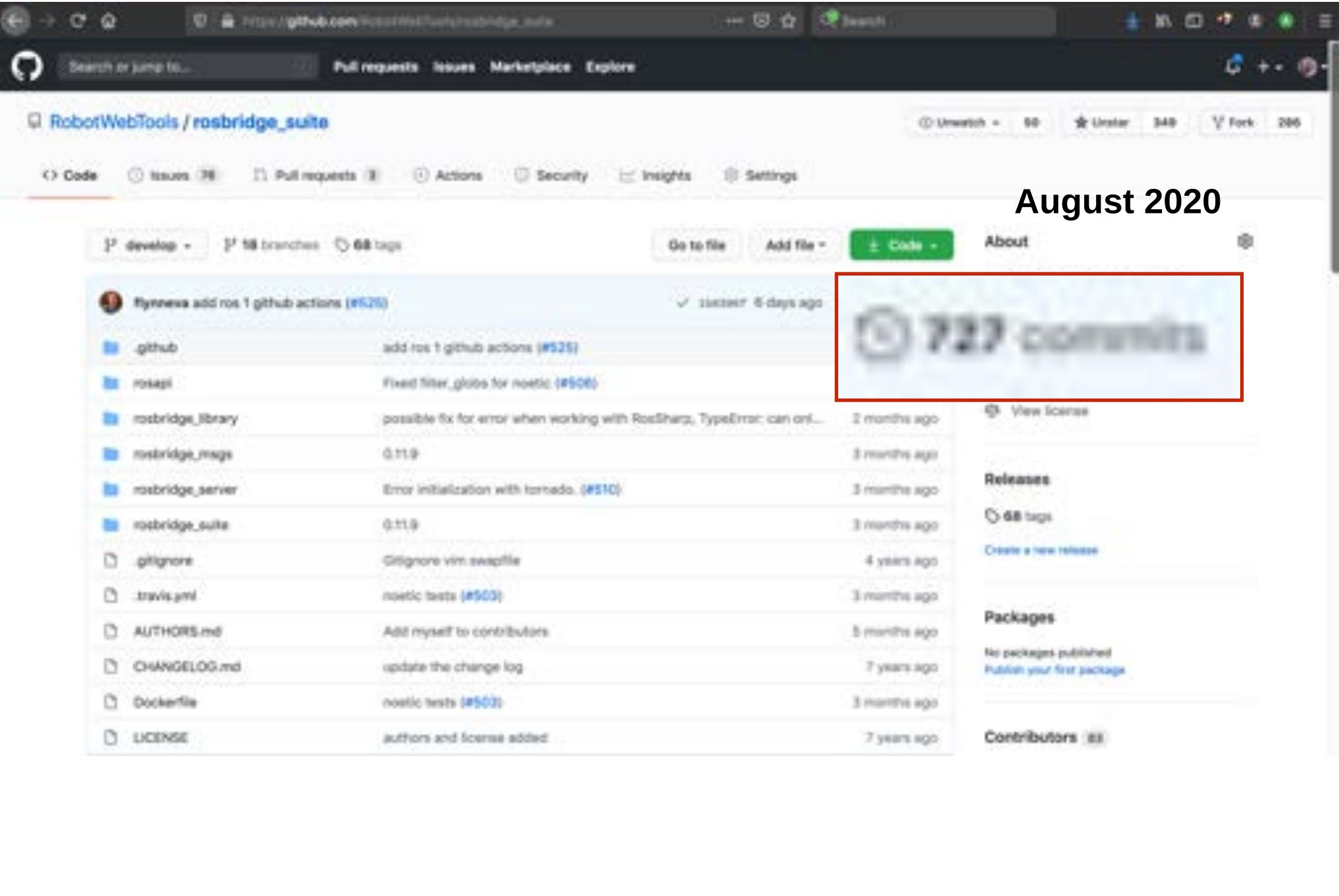


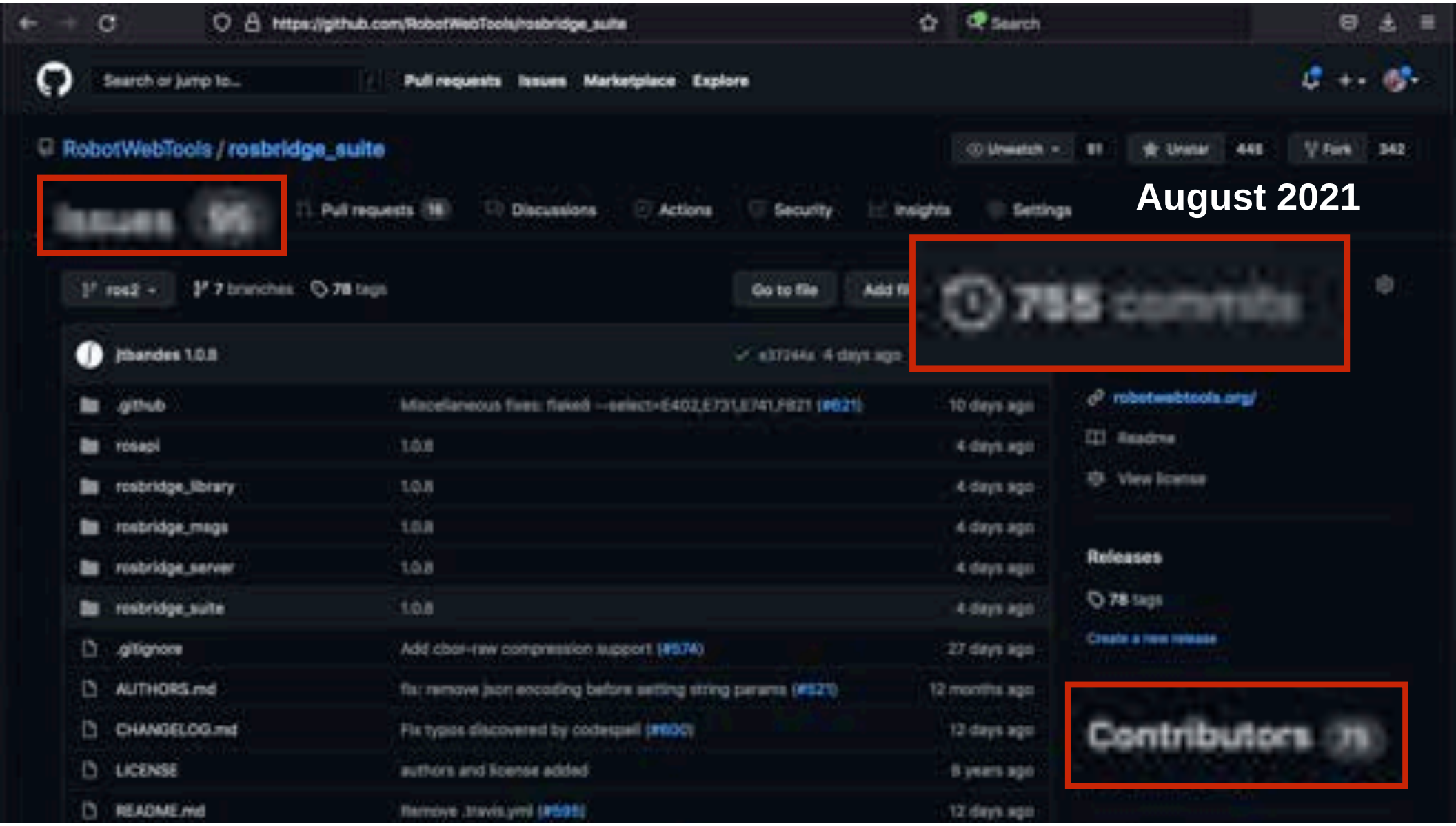
September 2017

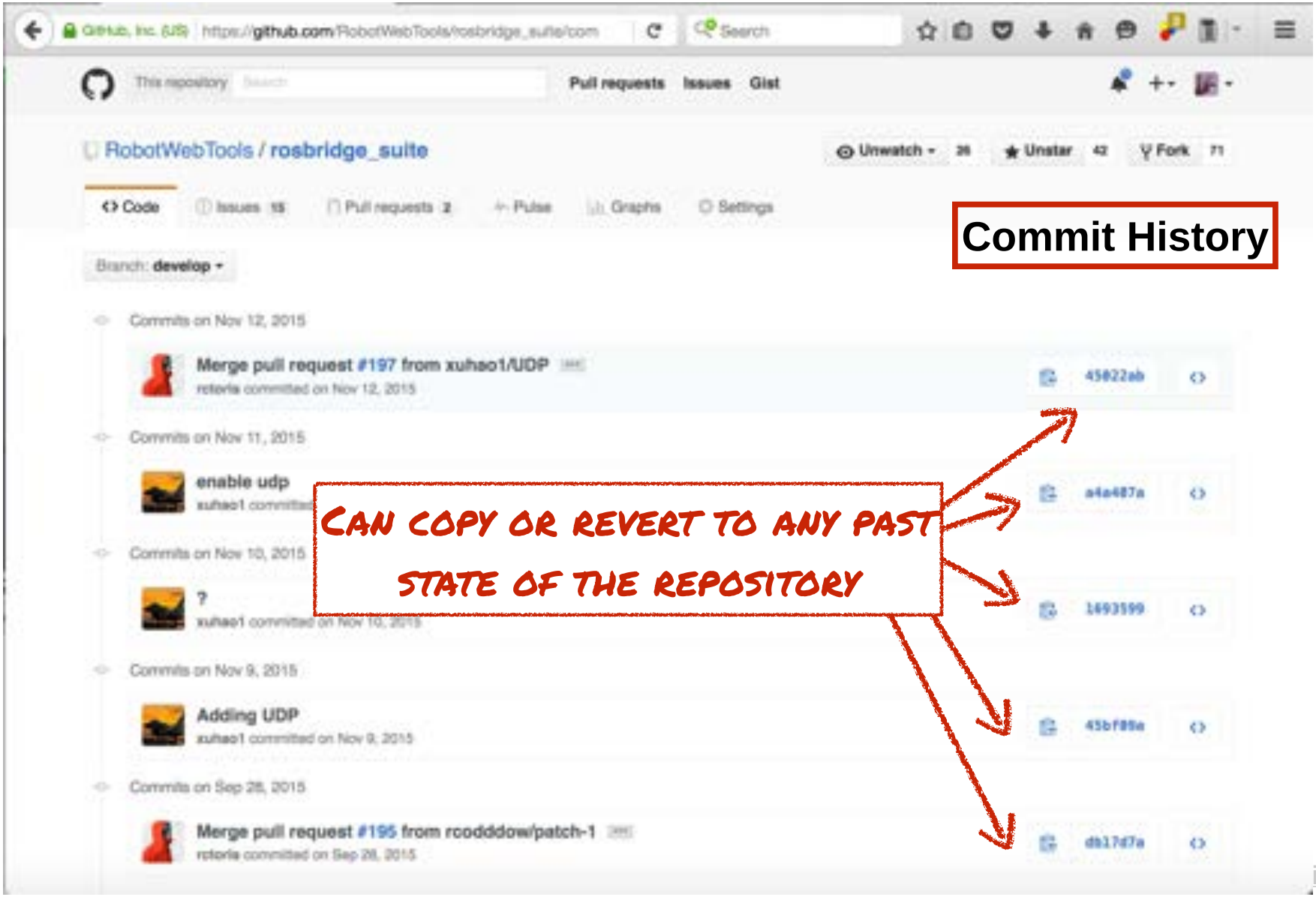


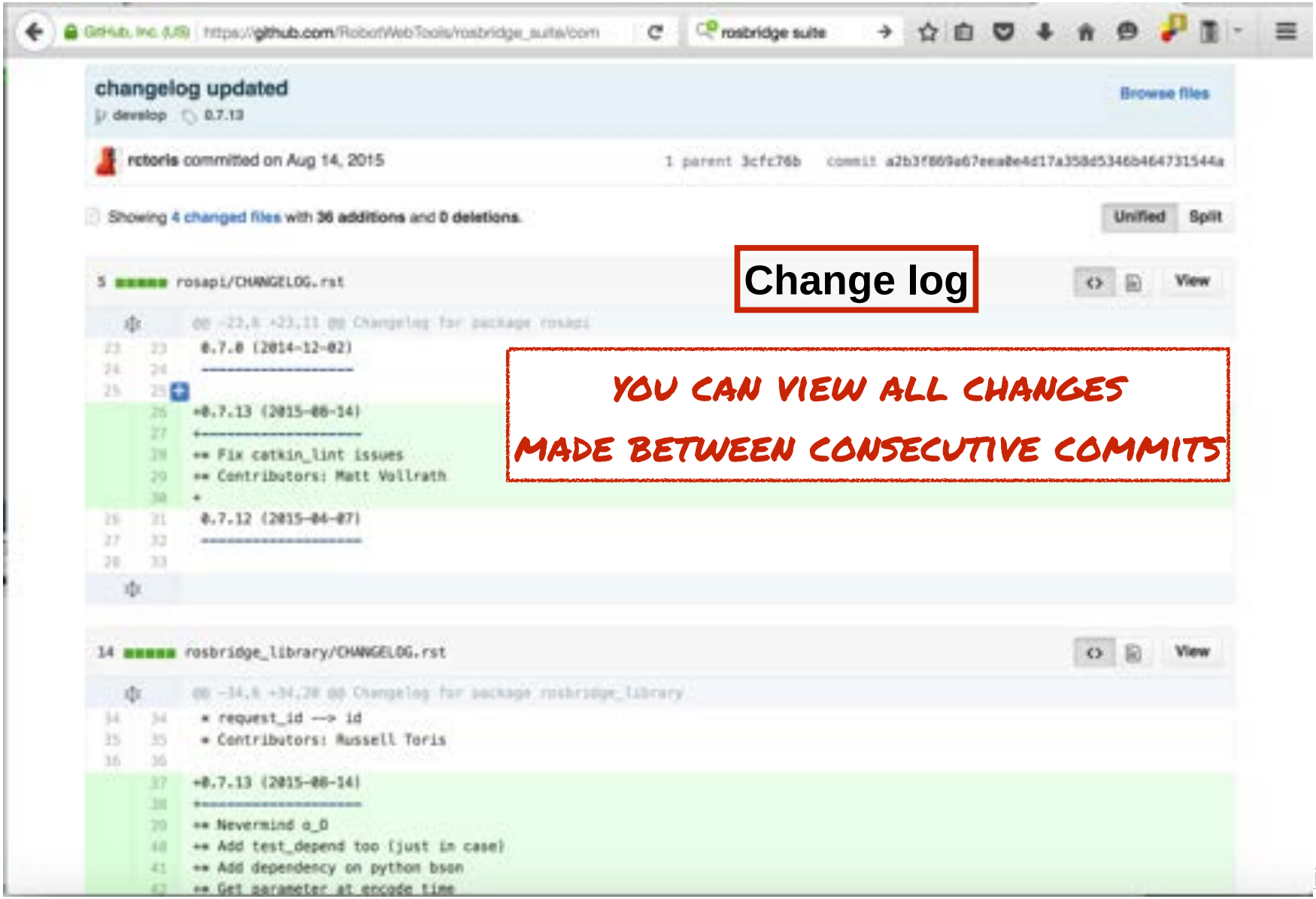
September 2018


GitHub, Inc. [US]
Search
RobotWebTools / rosbridge_suite
Unwatch 46
Unstar 251
Fork 221
Code
Issues 47
Pull requests 5
Security
Insights
Settings
Server Implementations of the rosbridge v2 Protocol http://robotwebtools.org/
Edit
Manage topics
September 2019
685 commits
12 branches
59 releases
66 contributors
View license
Clone or download
Latest commit
.github
Add GitHub issue template and TROUBLESHOOTING.md (#397)
5 months ago
rosapi
0.11.3 release (#424)
28 days ago
rosbridge_library
0.11.3 release (#424)
28 days ago
rosbridge_msgs
0.11.3 release (#424)
28 days ago
rosbridge_server
0.11.3 release (#424)
28 days ago
rosbridge_suite
0.11.3 release (#424)
28 days ago
.gitignore
Gitignore vim swapfile
3 years ago
.travis.yml
Travis CI: Look for Python syntax errors and undefined name (#420)
2 months ago
AUTHORS.md
authors and license added
6 years ago

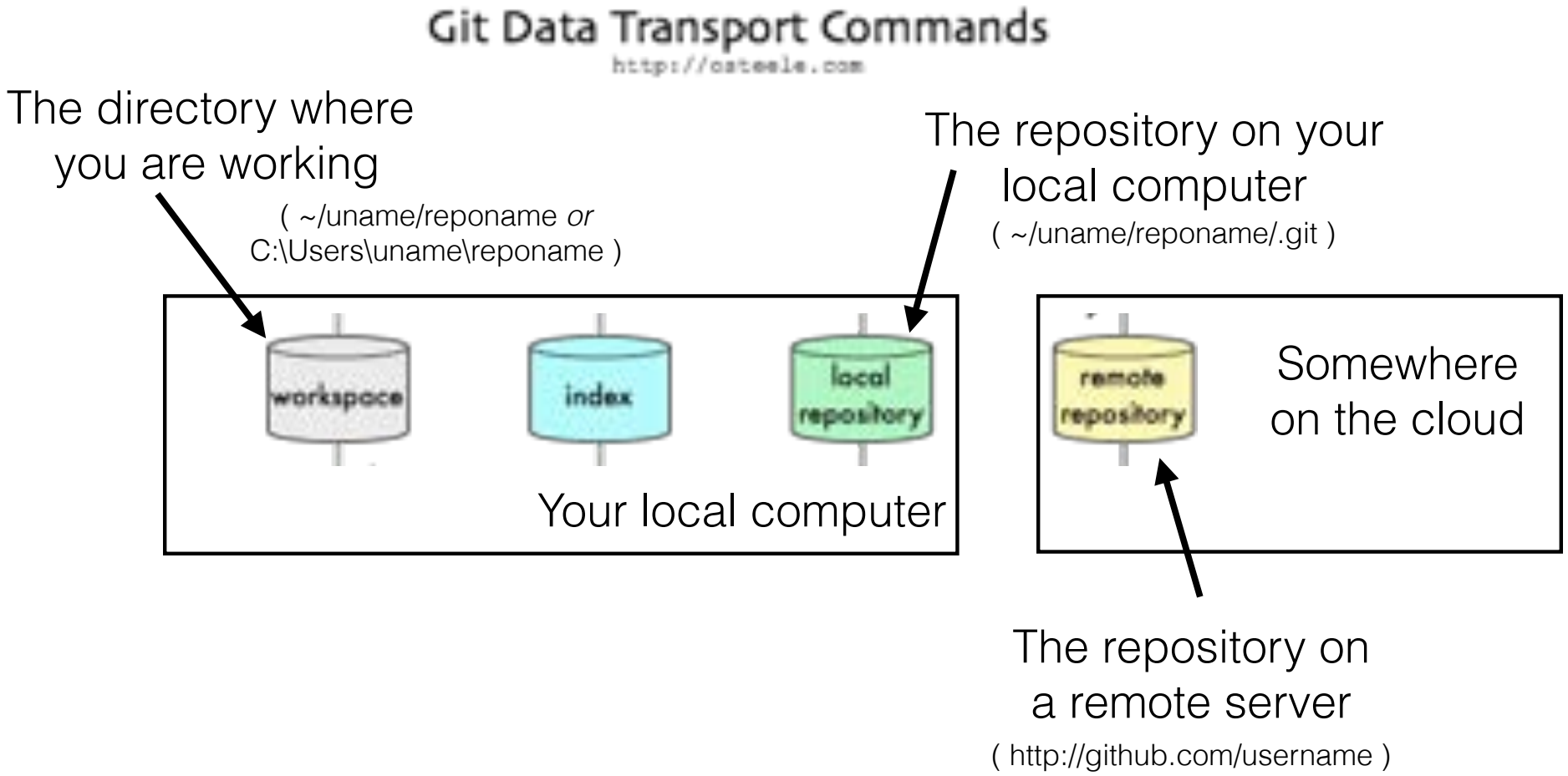

Pull requests
Issues
Marketplace
Explore
RobotWebTools / rosbridge_suite
Code
Issues 76
Pull requests 3
Actions
Security
Insights
Settings
August 2020
develop
18 branches
68 tags
Go to file
Add file
Code
About
727 commits
flynneva add ros 1 github actions (#525)
6 days ago
.github
add ros 1 github actions (#525)
rosapi
Fixed filter_globs for noetic (#506)
rosbridge_library
possible fix for error when working with RosSharp, TypeError: can onl...
2 months ago
rosbridge_msgs
0.11.9
3 months ago
rosbridge_server
Error initialization with tornado. (#510)
3 months ago
rosbridge_suite
0.11.9
3 months ago
.gitignore
Gitignore vim swapfile
4 years ago
.travis.yml
noetic tests (#503)
3 months ago
AUTHORS.md
Add myself to contributors
5 months ago
CHANGELOG.md
update the change log
7 years ago
Dockerfile
noetic tests (#503)
3 months ago
LICENSE
authors and license added
7 years ago
View license
Releases
68 tags
Create a new release
Packages
No packages published
Publish your first package
Contributors 83


https://github.com/RobotWebTools/rosbridge_suite
Search or jump to...
Pull requests
Issues
Marketplace
Explore
RobotWebTools / rosbridge_suite
Unwatch 81
Unstar 445
Fork 342
Pull requests 16
Discussions
Actions
Security
Insights
Settings
August 2021
ros2
7 branches
78 tags
Go to file
jtbandes 1.0.8
e37244a 4 days ago
.github
Miscellaneous fixes: flake8 --select=E402,E731,E741,F821 (#821)
10 days ago
rosapi
1.0.8
4 days ago
rosbridge_library
1.0.8
4 days ago
rosbridge_msgs
1.0.8
4 days ago
rosbridge_server
1.0.8
4 days ago
rosbridge_suite
1.0.8
4 days ago
.gitignore
Add cbor-raw compression support (#574)
27 days ago
AUTHORS.md
fix: remove json encoding before setting string params (#521)
12 months ago
CHANGELOG.md
Fix typos discovered by codespell (#806)
12 days ago
LICENSE
authors and license added
8 years ago
README.md
Remove .travis.yml (#591)
12 days ago
robotwebtools.org/
Readme
View license
Releases
78 tags
Create a new release
Contributors


https://github.com/RobotWebTools/rosbridge_suite/com
Search
This repository
Pull requests
Issues
Gist
RobotWebTools / rosbridge_suite
Unwatch
Unstar
Fork
Code
Issues
Pull requests
Pulse
Graphs
Settings
Commit History
Branch: develop
Commits on Nov 12, 2015
Merge pull request #197 from xuhao1/UDP
45022ab
Commits on Nov 11, 2015
enable udp
a4a487a
Commits on Nov 10, 2015
?
1693599
Commits on Nov 9, 2015
Adding UDP
45bf89e
Commits on Sep 28, 2015
Merge pull request #195 from rcodddow/patch-1
db17d7a
CAN COPY OR REVERT TO ANY PAST STATE OF THE REPOSITORY


changelog updated
develop
0.7.13
Browse files
rctoris committed on Aug 14, 2015
1 parent 3cfc76b
Showing 4 changed files with 36 additions and 0 deletions.
Unified
Split
Change log
5 rosapi/CHANGELOG.rst
View
0.7.0 (2014-12-02)
+0.7.13 (2015-08-14)
* Fix catkin_lint issues
* Contributors: Matt Vollrath
0.7.12 (2015-04-07)
14 rosbridge_library/CHANGELOG.rst
View
* request_id --> id
* Contributors: Russell Toris
+0.7.13 (2015-08-14)
* Nevermind o_O
* Add test_depend too (just in case)
* Add dependency on python bson
* Get parameter at encode time
YOU CAN VIEW ALL CHANGES
MADE BETWEEN CONSECUTIVE COMMITS

How does git work?


Git Data Transport Commands
http://osteele.com
The directory where you are working
(~/uname/reponame *or* C:\Users\uname\reponame)
The repository on your local computer
(~/uname/reponame/.git)
workspace
index
local repository
Your local computer
remote repository
Somewhere on the cloud
The repository on a remote server
(http://github.com/username)

Git Data Transport Commands

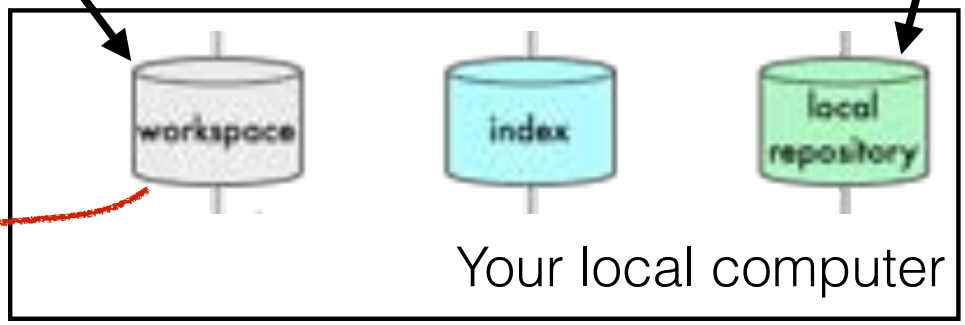
http://osteele.com

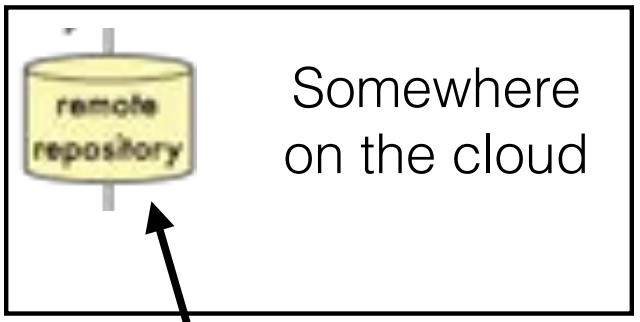
The directory where you are working

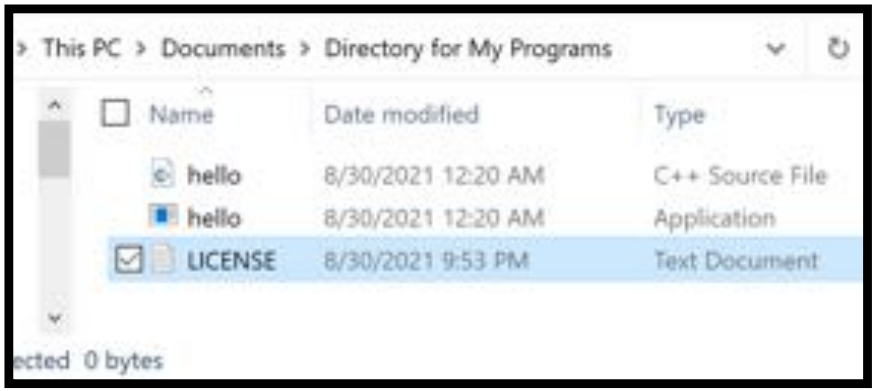
(~/uname/reponame *or* C:\Users\uname\reponame)

The repository on your local computer

(~/uname/reponame/.git)







The repository on a remote server

(http://github.com/username)

Git Data Transport Commands

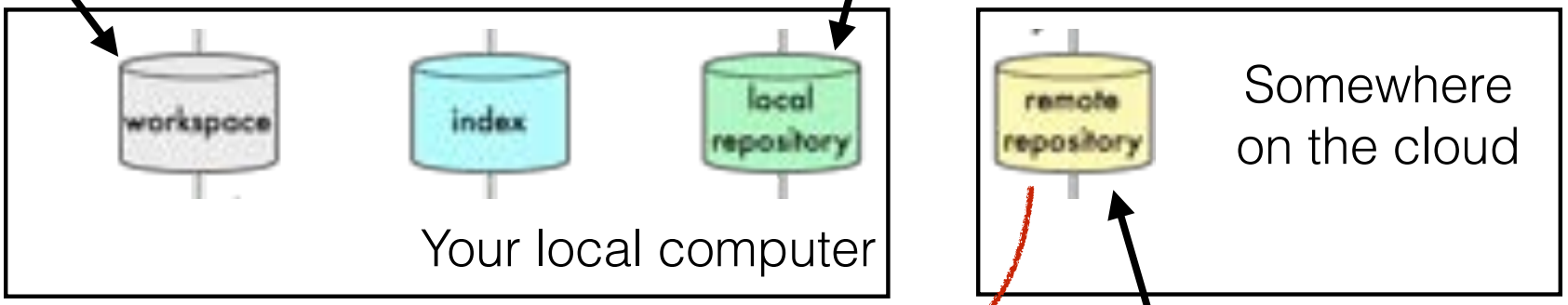
http://osteele.com

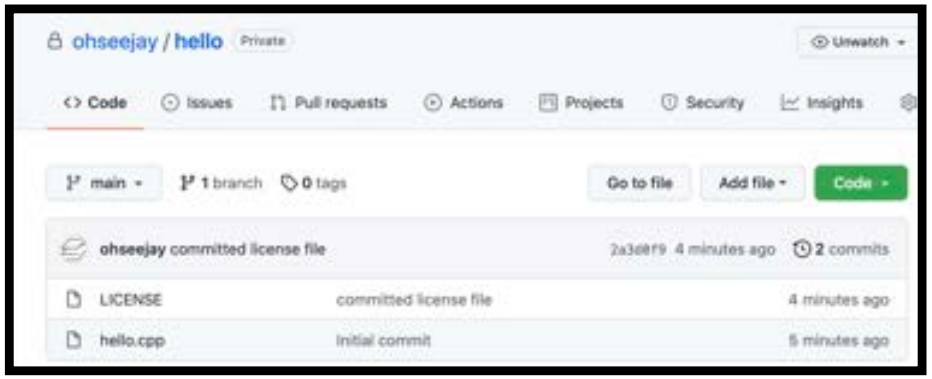
The directory where you are working

(~/uname/reponame *or* C:\Users\uname\reponame)

The repository on your local computer

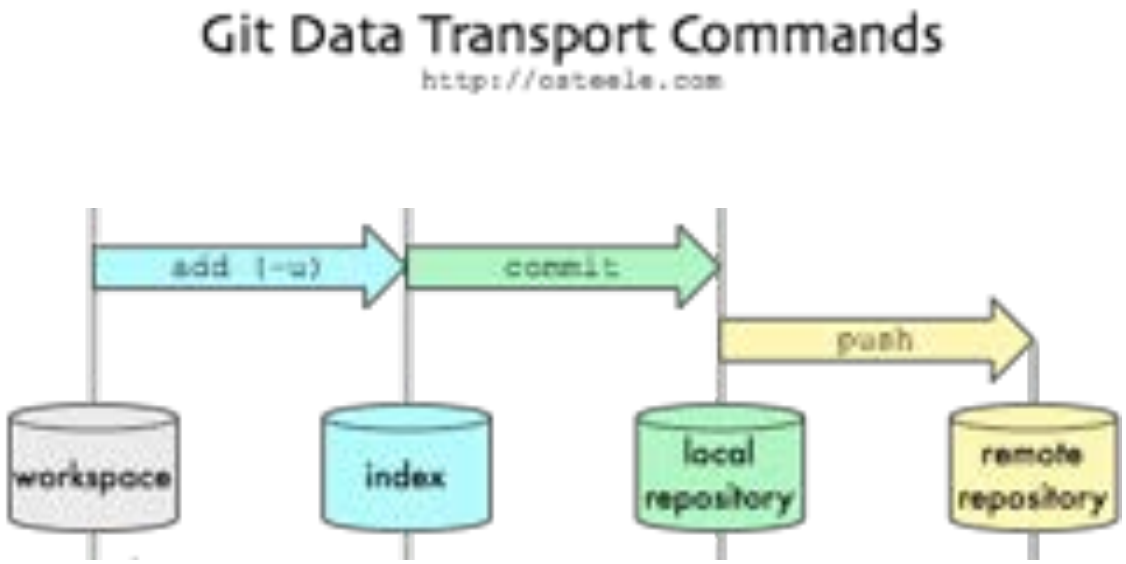
(~/uname/reponame/.git)





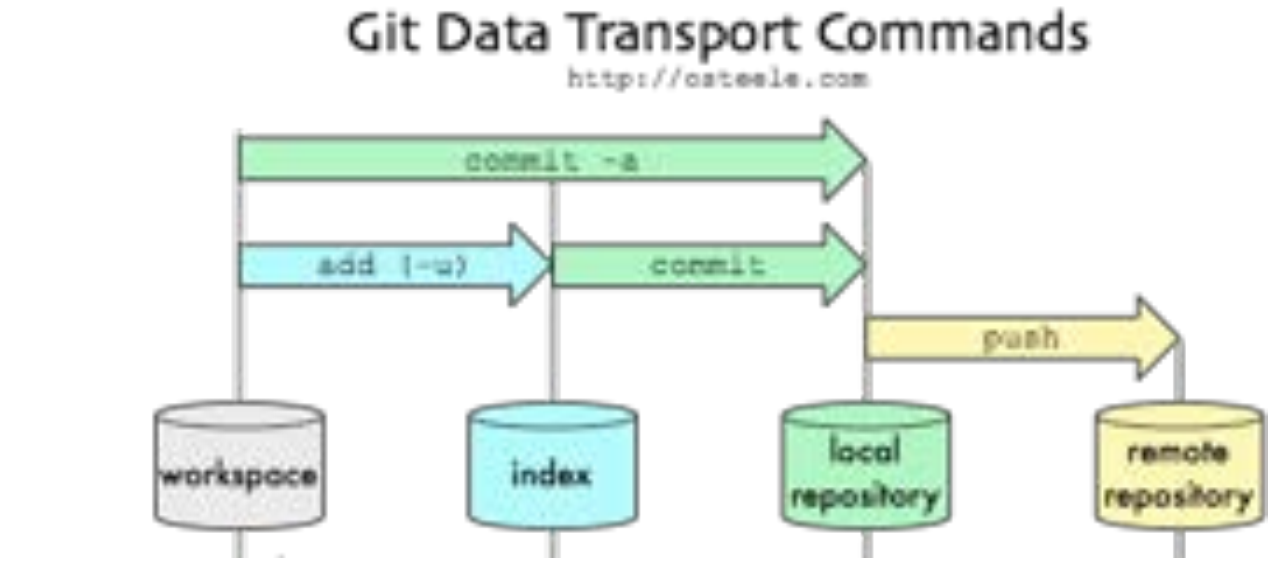
The repository on a remote server

(http://github.com/username)



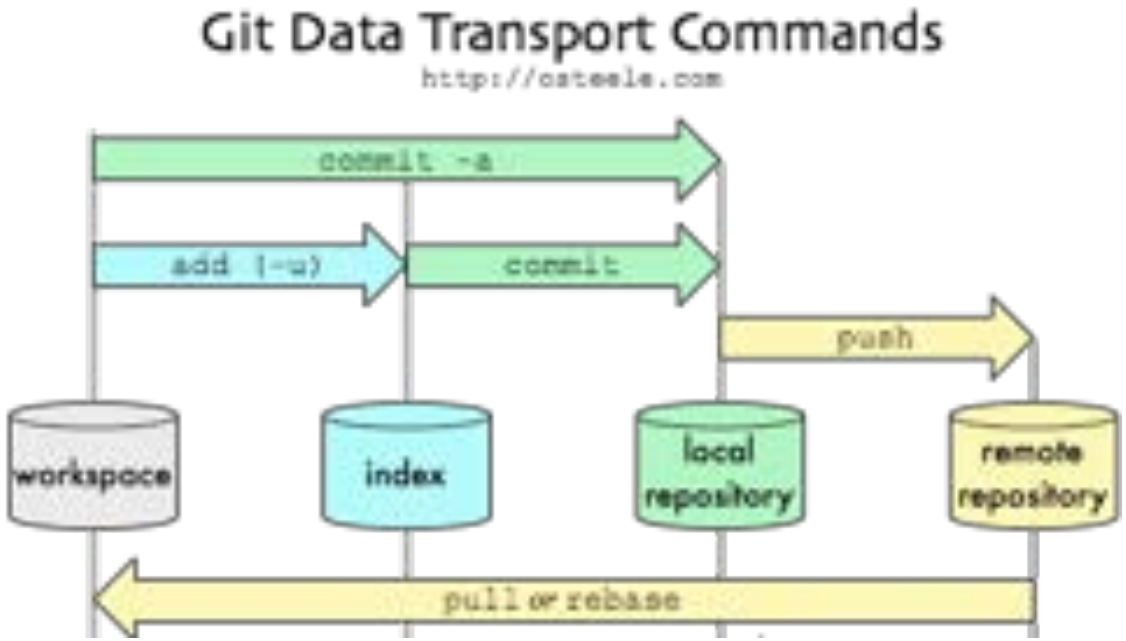
~/uname/reponame ~/uname/reponame/.git http://github.com/username

After making local changes, you can add, commit, and push to your remote repository



~/uname/reponame ~/uname/reponame/.git http://github.com/username

If there are no files to add, just commit and push



~/uname/reponame ~/uname/reponame/.git http://github.com/username

A pull command updates the local workspace with changes from the remote repository

git basics: commands

- Push completed project to repository (or just to update)
  - add files to a repository: `git add <file listing>`
  - commit changes to local repo: `git commit -a -m “<msg>”`
  - push local changes to a remote repository: `git push`
- Pull to updates your local repository (and workspace) from remote
  - pull remote changes to a local repository: `git pull`

Our first challenge: Project 0

Our first challenge: Project 0

Pocket Calculator

Our first challenge: Project 0

Pocket Calculator

```
$ ./calculator
Please type a number and press enter: 100
Please type a math operator (one of: + - * /): +
Please type a number and press enter: 2
100+2= 102
```

Actually,
it will look more like this

Next Lecture: Operators and Variables

```
#include <iostream>

/*
   Let's write a calculator program
*/

int main()
{
   std::cout << "What is 100 plus 2?" << "\n";
}
```

calculator.cpp (Version 00)