Coding Workflow

ROB 102: Introduction to AI & Programming

Lab Session 1

2021/09/03

Admin

Join Slack! (https://um-fa21-rob102.slack.com/)

Bookmark the course website (robotics102.org)

Office hours start next week

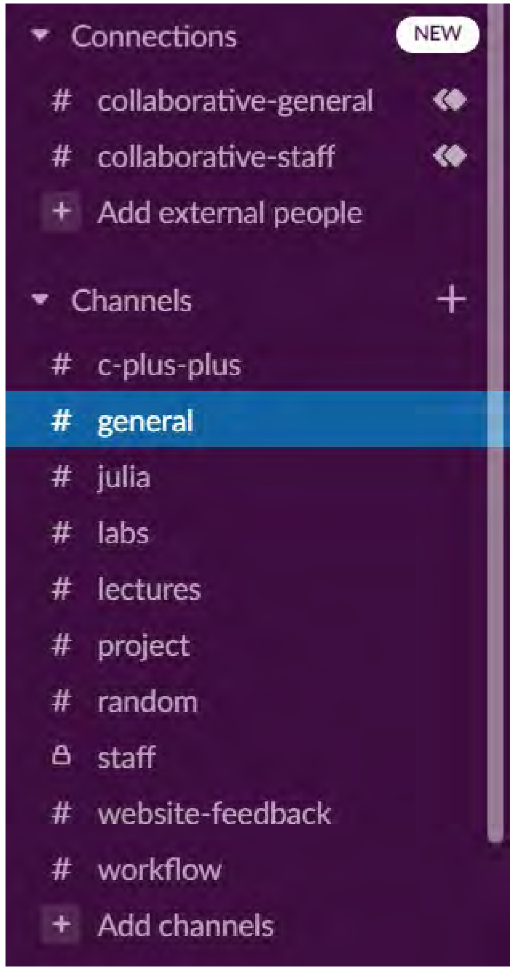
Robotics Building

Prof Jenkins: MW 1-3PM @ FRB 2236

Jana: Tu 10AM-12PM, W 3-5 PM @ FRB 2000

IAs: TBD

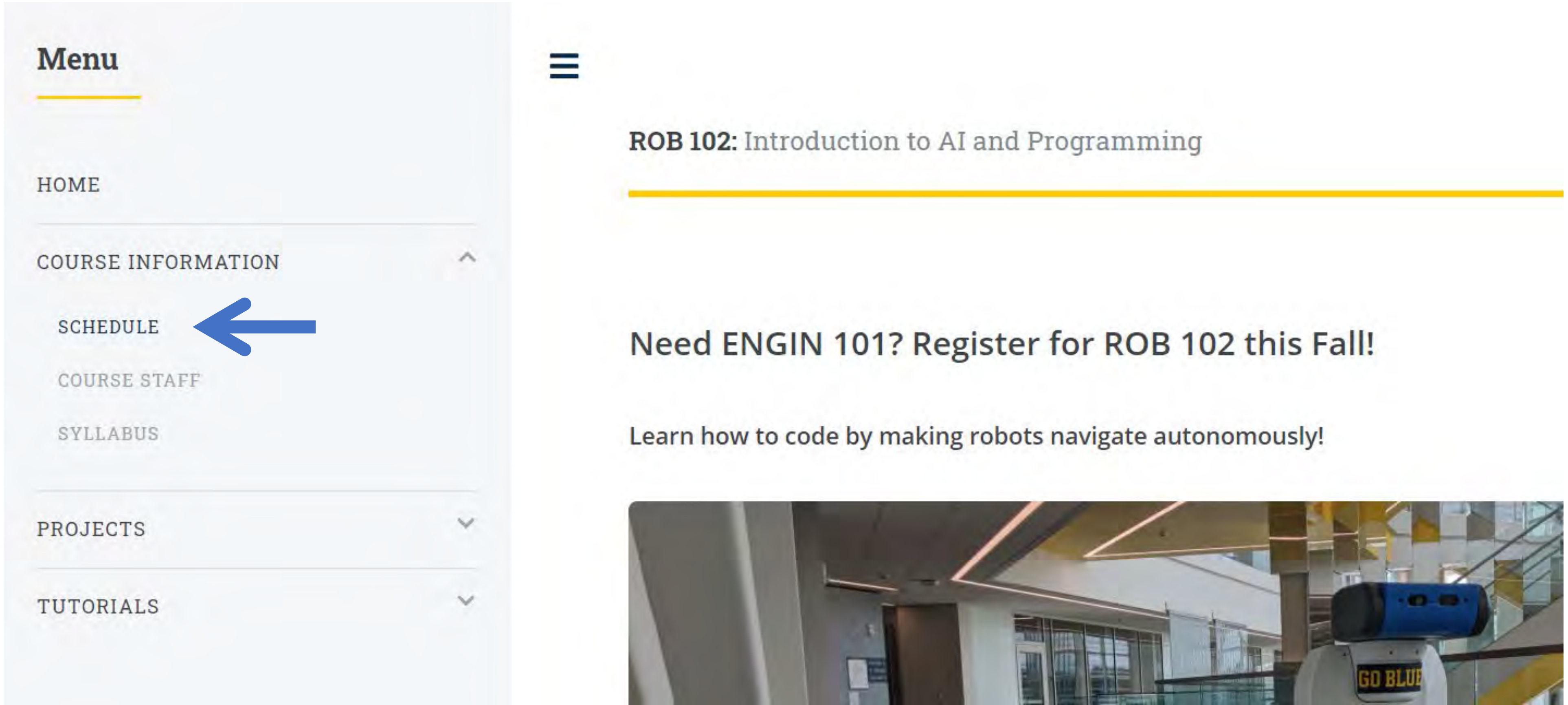
The lab space with the maze where we did the Pair Navigation demo.



Admin

Project 0 is out. We will go through it in class.

Due Sept. 20 at 11:59 PM.

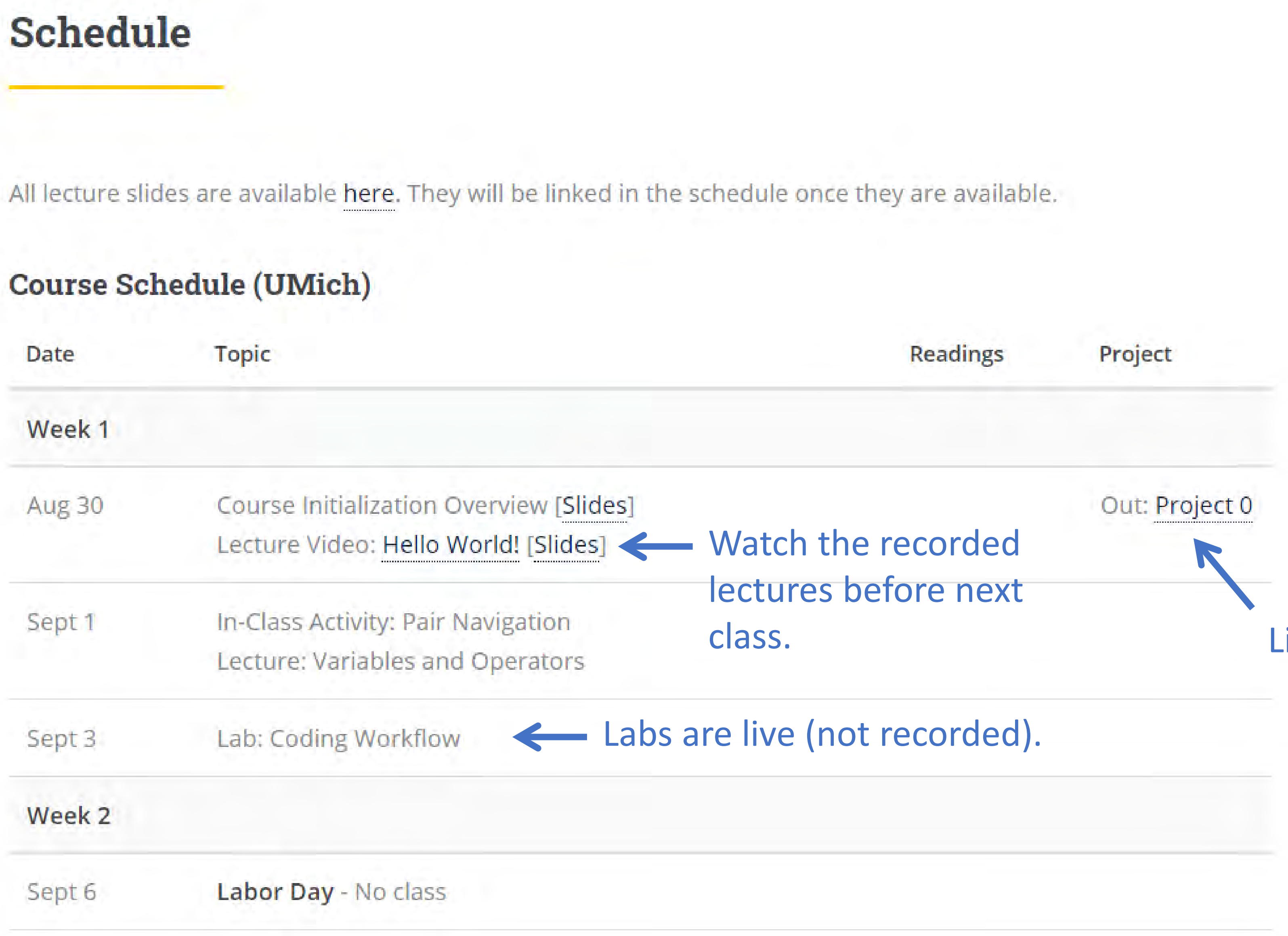


ROB 102: Introduction to AI and Programming

Need ENGIN 101? Register for ROB 102 this Fall!

Learn how to code by making robots navigate autonomously!

Admin



Schedule

All lecture slides are available here. They will be linked in the schedule once they are available.

Course Schedule (UMich)

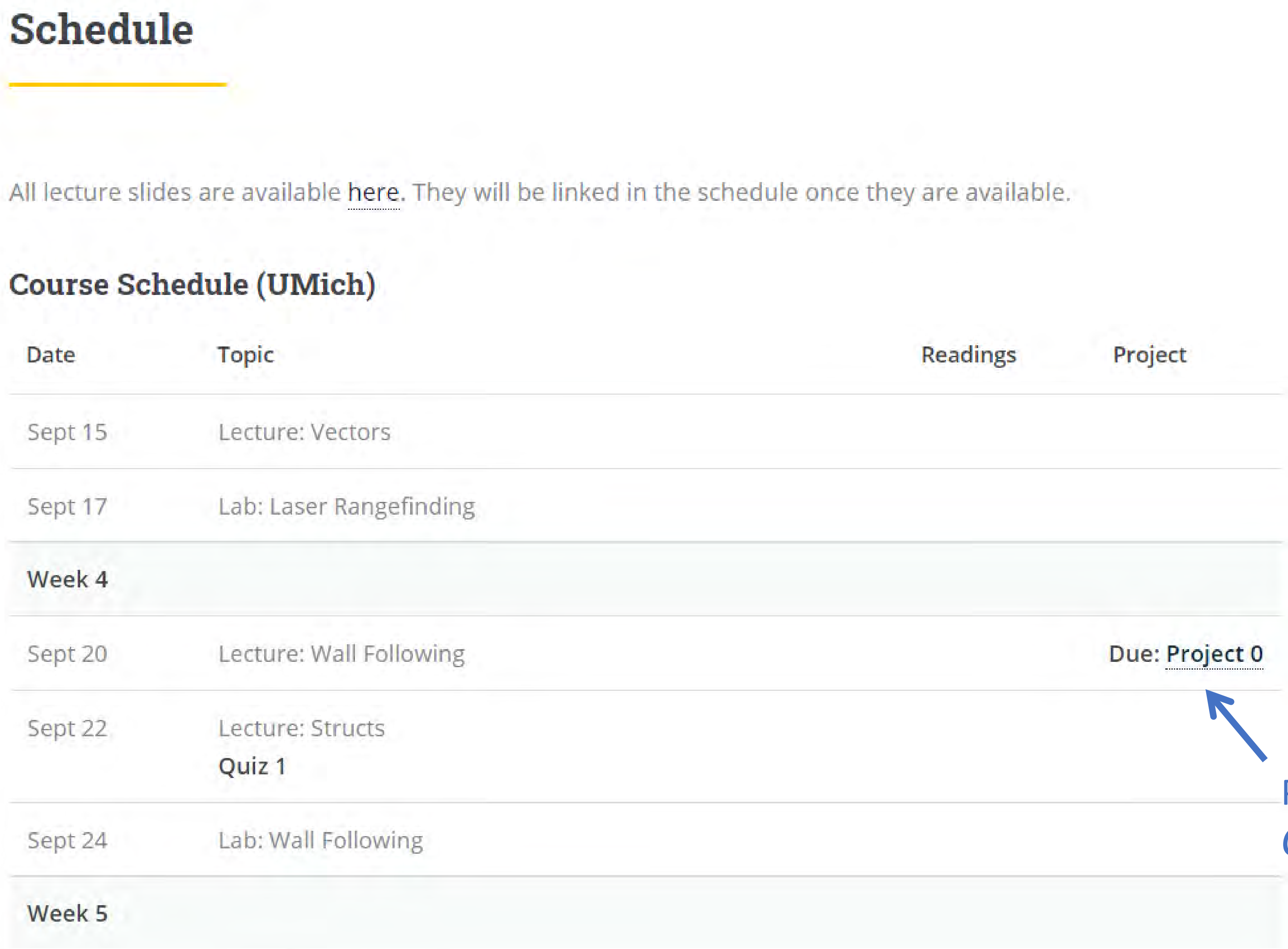
Date	Topic	Readings	Project
Week 1			
Aug 30	Course Initialization Overview [Slides] Lecture Video: Hello World! [Slides]		Out: Project 0
Sept 1	In-Class Activity: Pair Navigation Lecture: Variables and Operators		
Sept 3	Lab: Coding Workflow		
Week 2			
Sept 6	**Labor Day** - No class		

Watch the recorded lectures before next class.

Link to project 0

Labs are live (not recorded).

Admin



Schedule

All lecture slides are available here. They will be linked in the schedule once they are available.

Course Schedule (UMich)

Date	Topic	Readings	Project
Sept 15	Lecture: Vectors		
Sept 17	Lab: Laser Rangefinding		
Week 4			
Sept 20	Lecture: Wall Following		Due: Project 0
Sept 22	Lecture: Structs **Quiz 1**		
Sept 24	Lab: Wall Following		
Week 5			

Project 0 due **Sept 20 @ 11:59 PM**

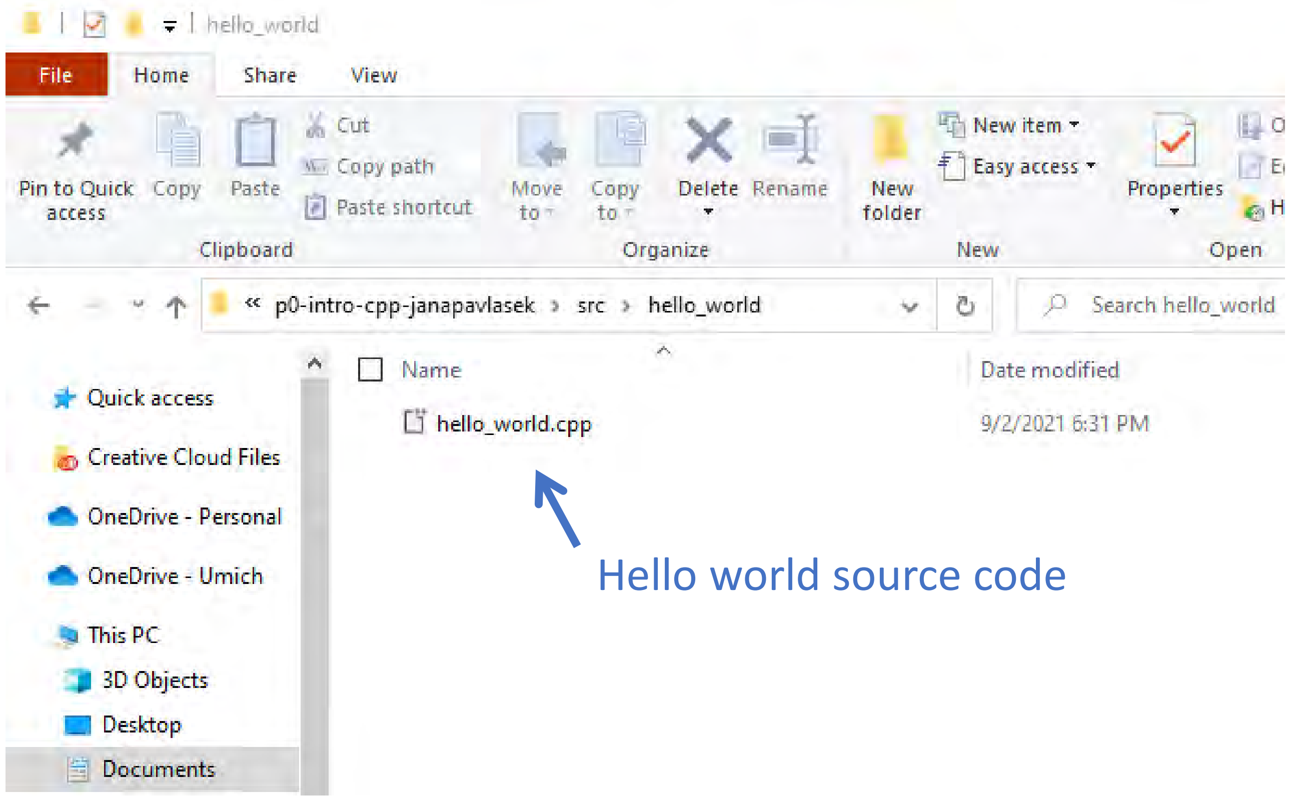
Today…

- Coding Workflow
- Git & version control
- VSCode & programming environment
- Docker & running your code
- *Computer setup help*

Tasks to complete:

- ✓ Install all software
- ✓ Push license to GitHub repo
- ✓ Run Hello World on your computer
- ✓ Push changes to your repository

Coding workflow

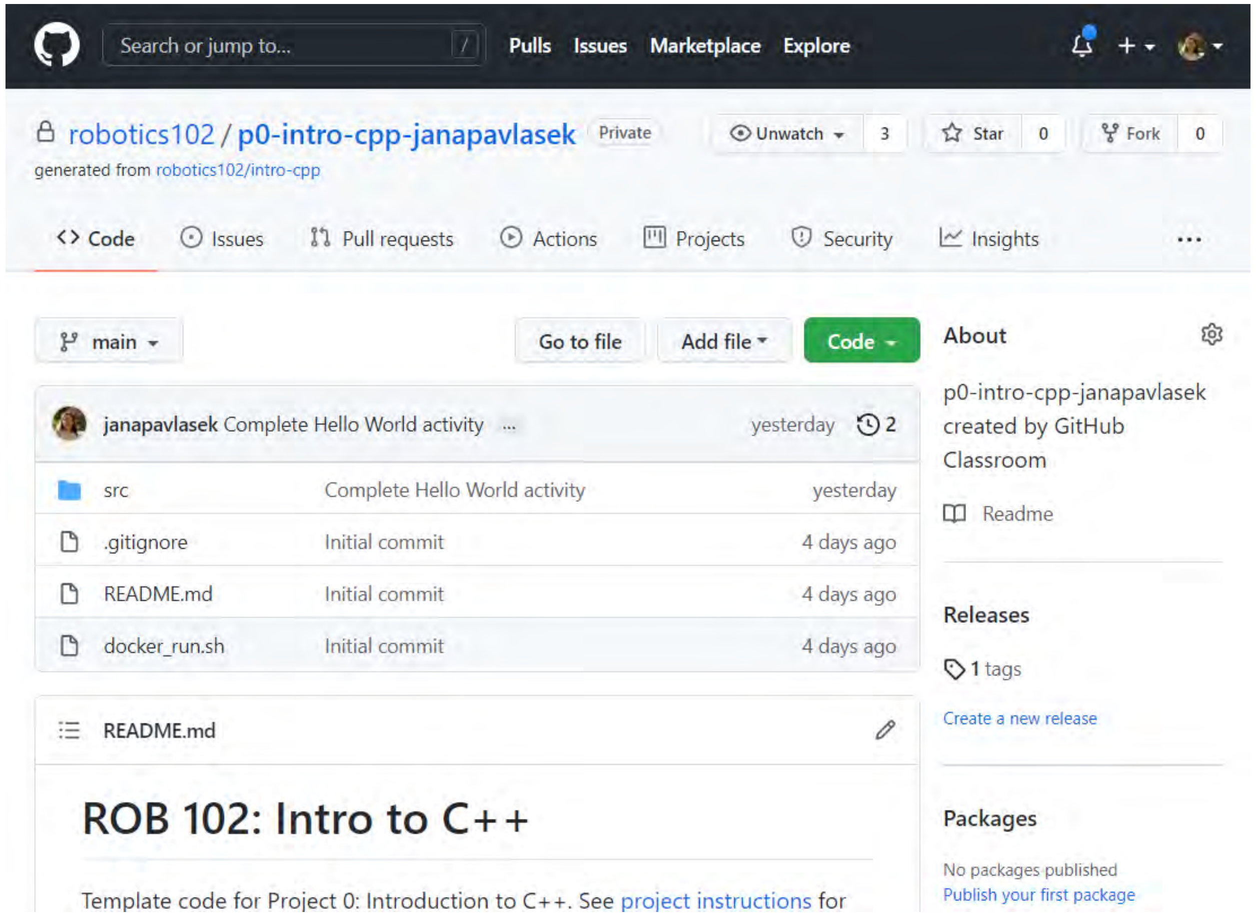


Hello world source code

We need to get the code on our computers first!

Open code

Coding workflow



robotics102 / p0-intro-cpp-janapavlasek Private

generated from robotics102/intro-cpp

janapavlasek Complete Hello World activity ... yesterday 2

src	Complete Hello World activity	yesterday
.gitignore	Initial commit	4 days ago
README.md	Initial commit	4 days ago
docker_run.sh	Initial commit	4 days ago

README.md

ROB 102: Intro to C++

Template code for Project 0: Introduction to C++. See project instructions for

About

p0-intro-cpp-janapavlasek created by GitHub Classroom

Readme

Releases

1 tags

Create a new release

Packages

No packages published

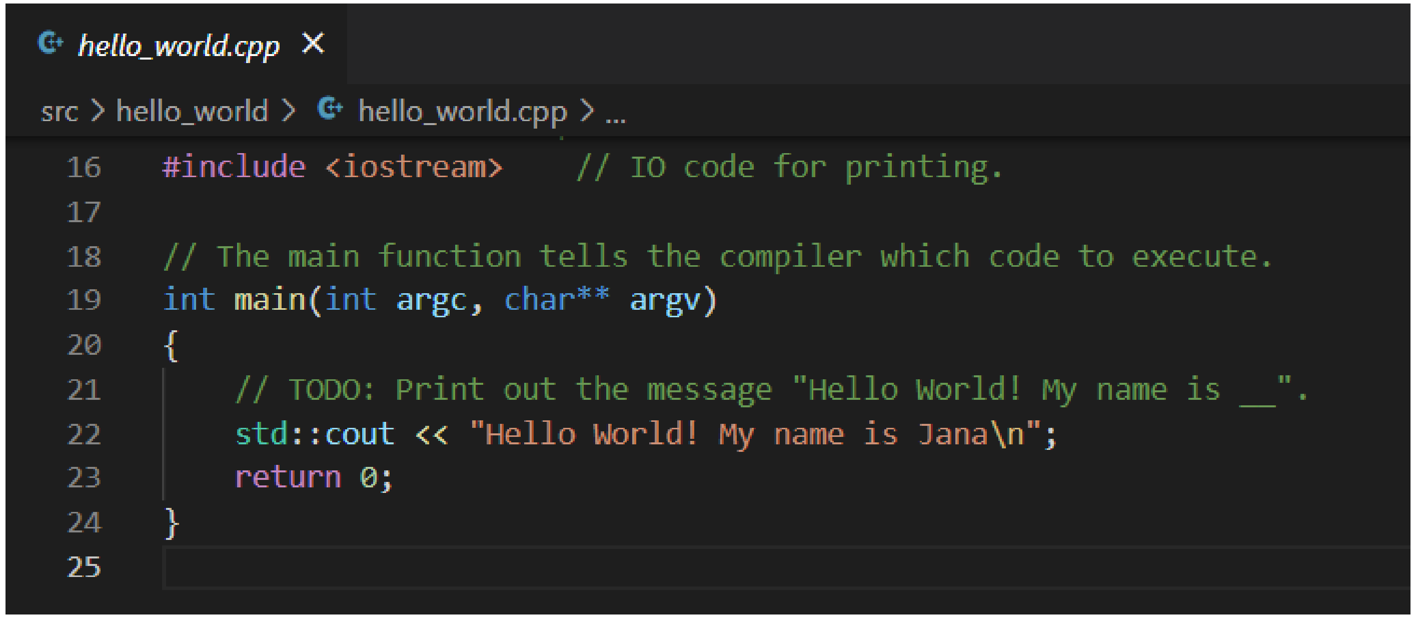
Publish your first package

Pull code from GitHub

$\downarrow$

Open code

Coding workflow



```cpp
#include <iostream>    // IO code for printing.

// The main function tells the compiler which code to execute.
int main(int argc, char** argv)
{
    // TODO: Print out the message "Hello World! My name is __".
    std::cout << "Hello World! My name is Jana\n";
    return 0;
}

```

Pull code from GitHub

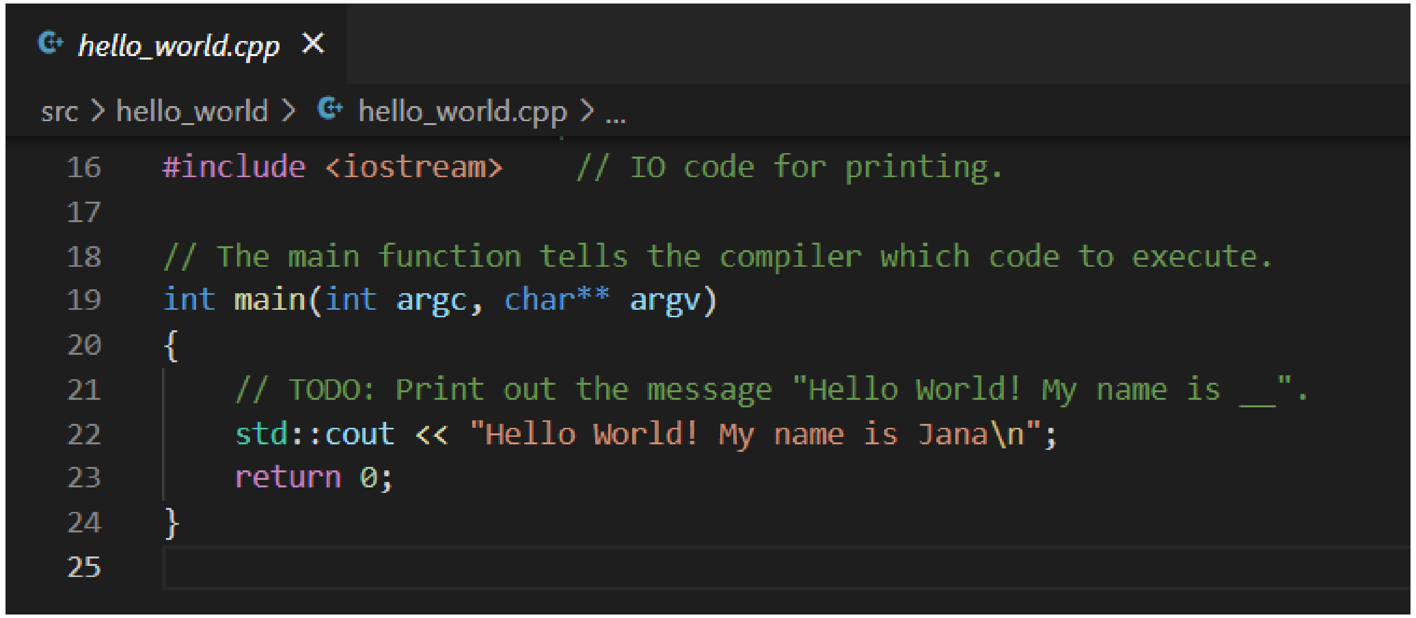
↓

Open code

↓

Update code

Coding workflow

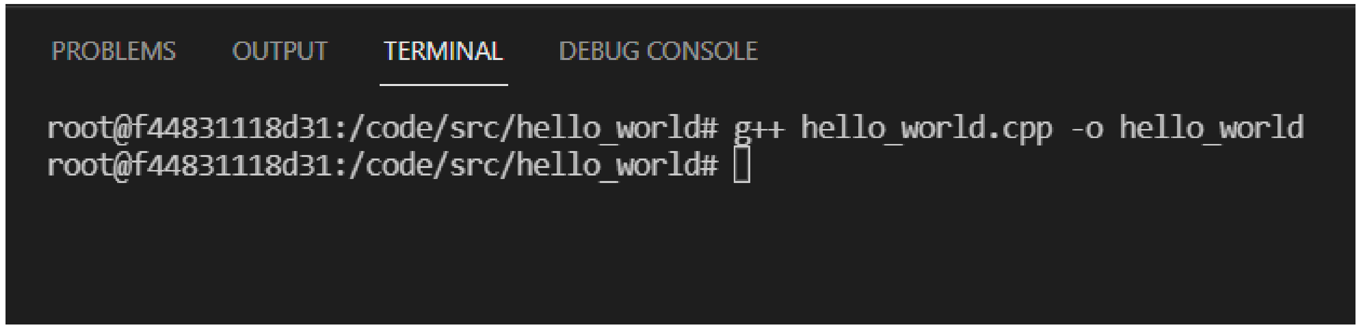

hello_world.cpp ×

src > hello_world > hello_world.cpp > ...

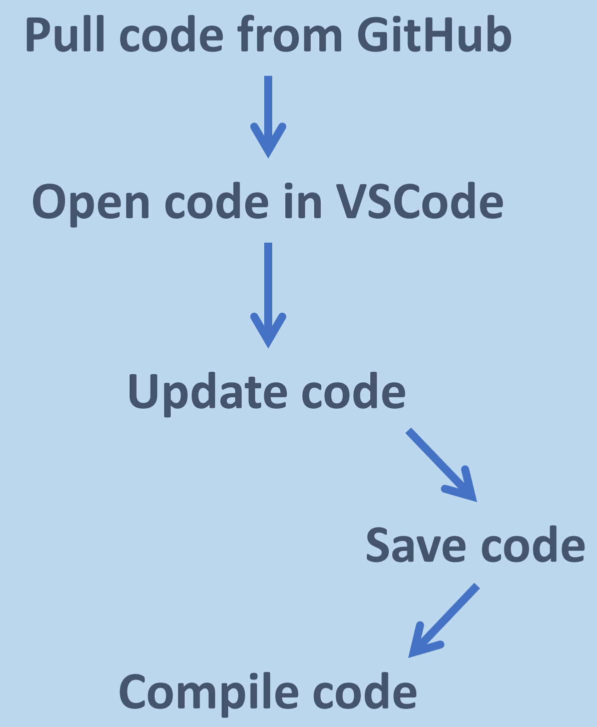
```cpp
#include <iostream>    // IO code for printing.

// The main function tells the compiler which code to execute.
int main(int argc, char** argv)
{
    // TODO: Print out the message "Hello World! My name is __".
    std::cout << "Hello World! My name is Jana\n";
    return 0;
}

```

$$\begin{tikzcd}
\text{Pull code from GitHub} \arrow[d] & \\
\text{Open code in VSCode} \arrow[d] & \\
\text{Update code} \arrow[dr] & \\
 & \text{Save code}
\end{tikzcd}$$

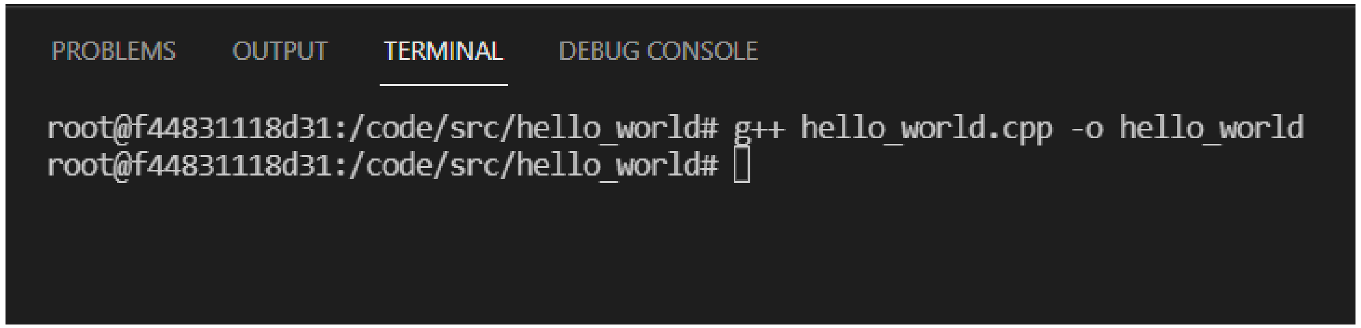
Coding workflow



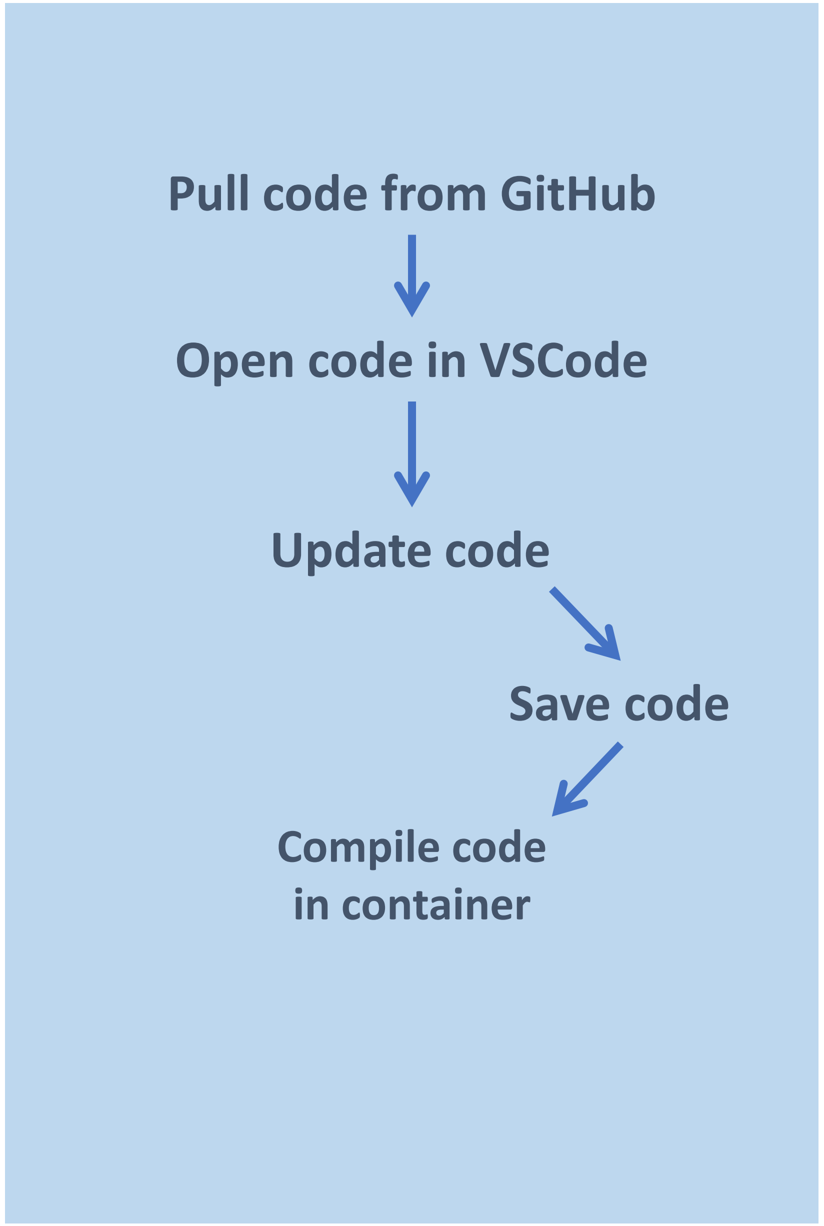
```
root@f44831118d31:/code/src/hello_world# g++ hello_world.cpp -o hello_world
root@f44831118d31:/code/src/hello_world# 
```



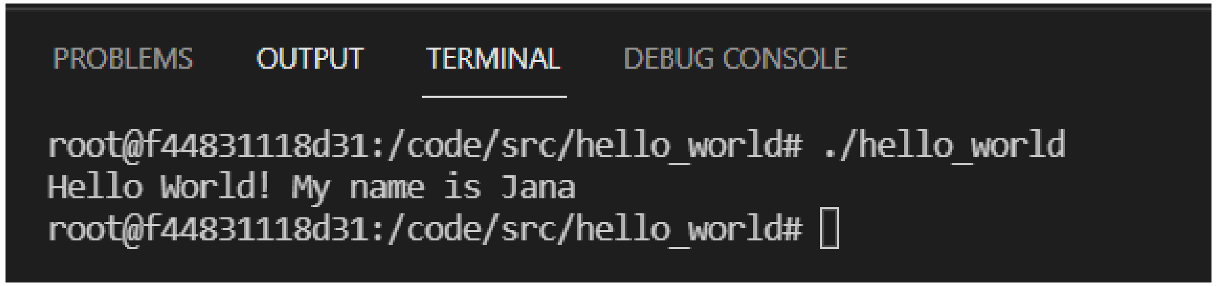
Coding workflow



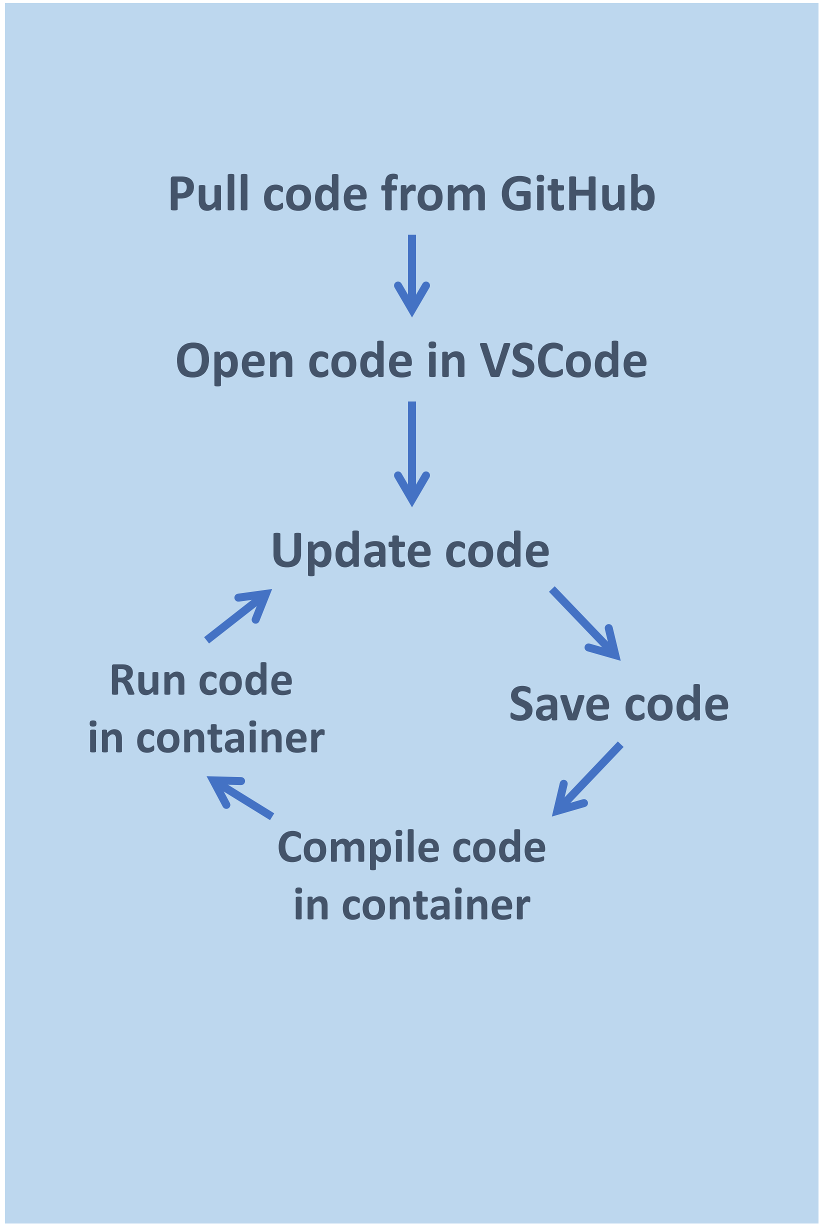
```
root@f44831118d31:/code/src/hello_world# g++ hello_world.cpp -o hello_world
root@f44831118d31:/code/src/hello_world# 
```



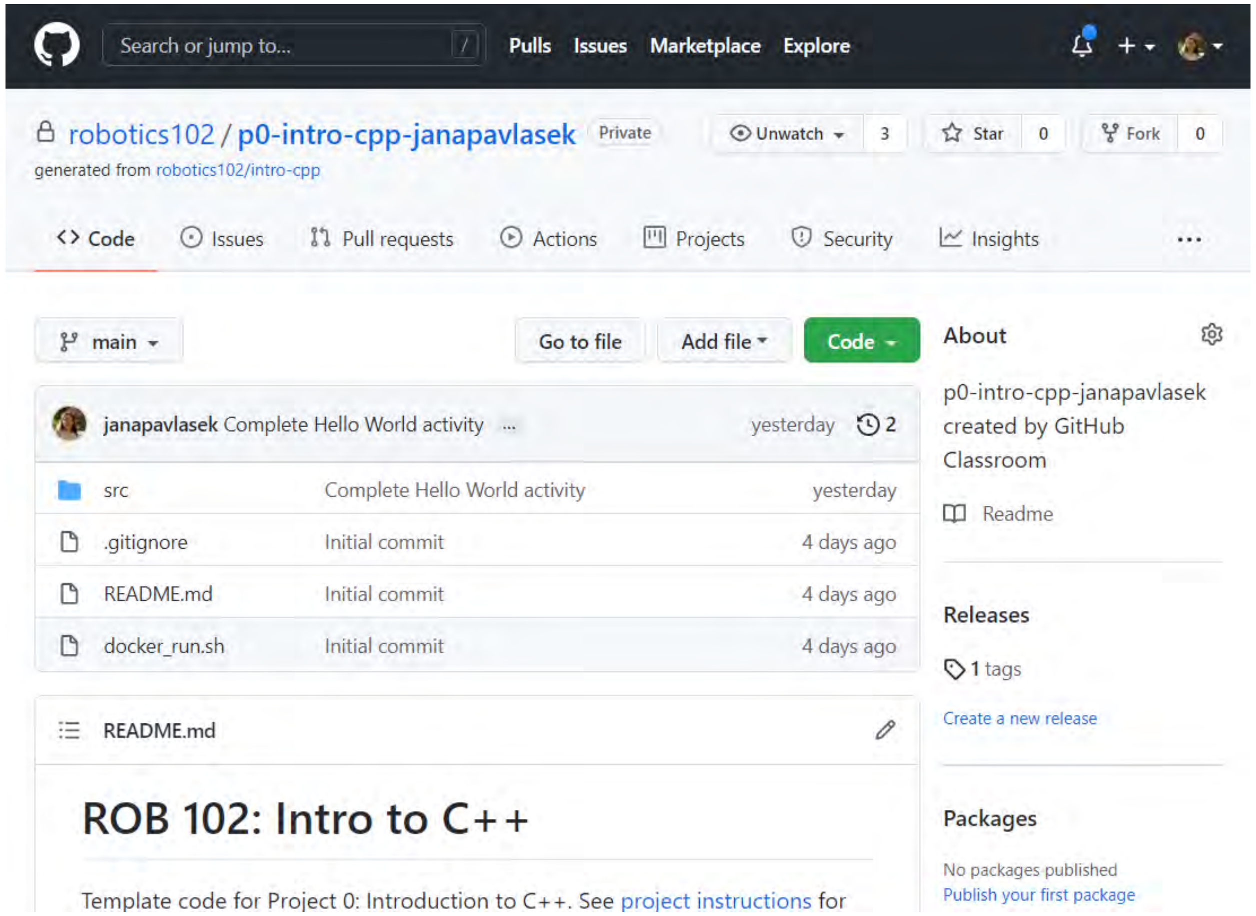
Coding workflow



```
root@f44831118d31:/code/src/hello_world# ./hello_world
Hello World! My name is Jana
root@f44831118d31:/code/src/hello_world# 
```



Coding workflow



Search or jump to... / Pulls Issues Marketplace Explore

robotics102 / p0-intro-cpp-janapavlasek Private

generated from robotics102/intro-cpp

Unwatch 3 Star 0 Fork 0

Code Issues Pull requests Actions Projects Security Insights ...

main Go to file Add file Code

janapavlasek Complete Hello World activity ... yesterday 2

src	Complete Hello World activity	yesterday
.gitignore	Initial commit	4 days ago
README.md	Initial commit	4 days ago
docker_run.sh	Initial commit	4 days ago

README.md

ROB 102: Intro to C++

Template code for Project 0: Introduction to C++. See project instructions for

About

p0-intro-cpp-janapavlasek created by GitHub Classroom

Readme

Releases

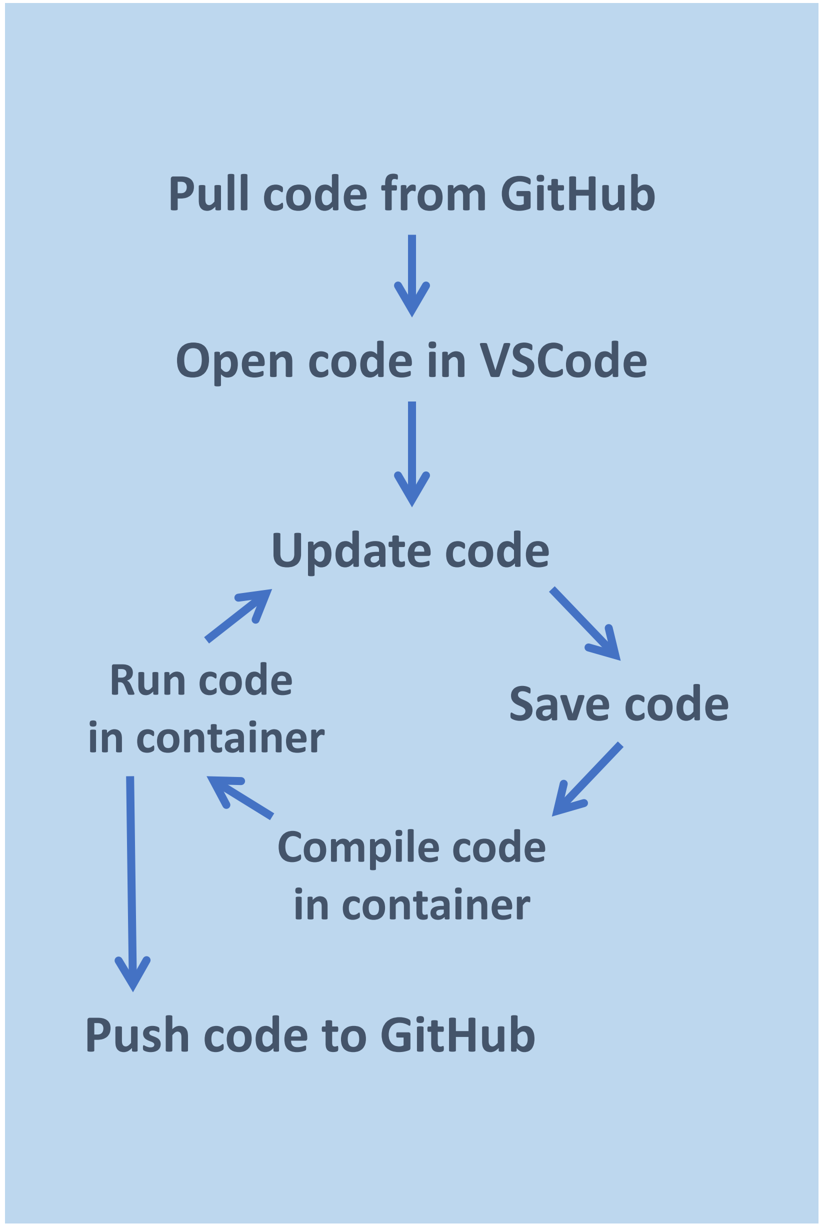
1 tags

Create a new release

Packages

No packages published

Publish your first package



What is Git?

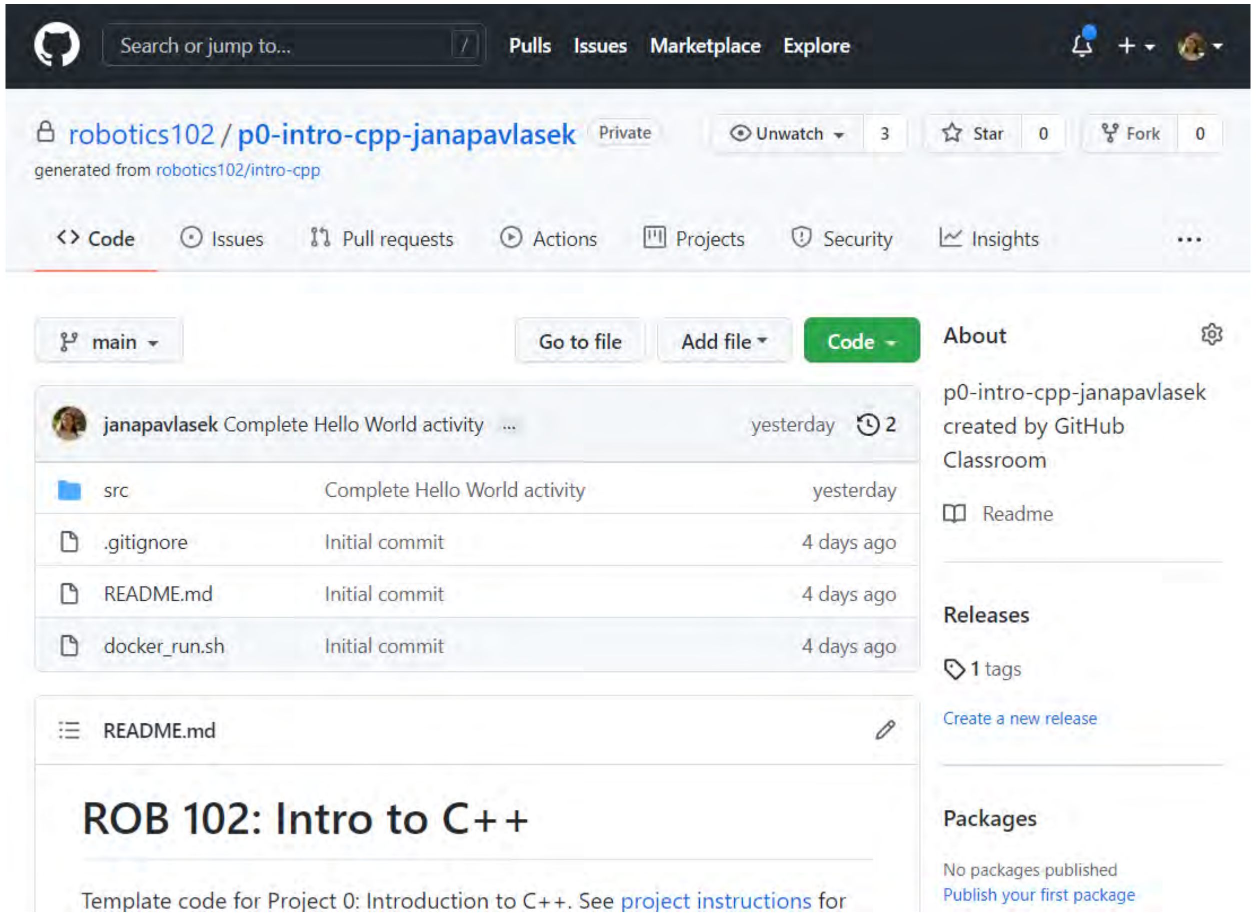
Git is **version control** software, meaning that it tracks changes to your files as you work on them over time

Similar to “track changes” feature in document writing programs, except you must choose which versions to include in the tracking—it is not automatic

Widely used in academia and industry

The only way for you to submit your homework for this course!

Git Workflow



Search or jump to...

Pulls Issues Marketplace Explore

robotics102 / p0-intro-cpp-janapavlasek Private

generated from robotics102/intro-cpp

Unwatch 3 Star 0 Fork 0

Code Issues Pull requests Actions Projects Security Insights

main Go to file Add file Code

janapavlasek Complete Hello World activity ...		yesterday 2
src	Complete Hello World activity	yesterday
.gitignore	Initial commit	4 days ago
README.md	Initial commit	4 days ago
docker_run.sh	Initial commit	4 days ago

README.md

ROB 102: Intro to C++

Template code for Project 0: Introduction to C++. See project instructions for

About

p0-intro-cpp-janapavlasek created by GitHub Classroom

Readme

Releases

1 tags

Create a new release

Packages

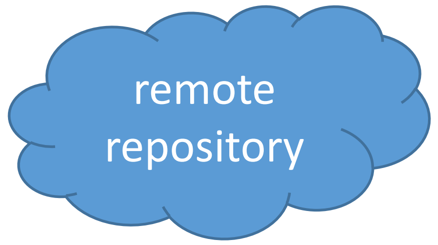
No packages published

Publish your first package

The cloud

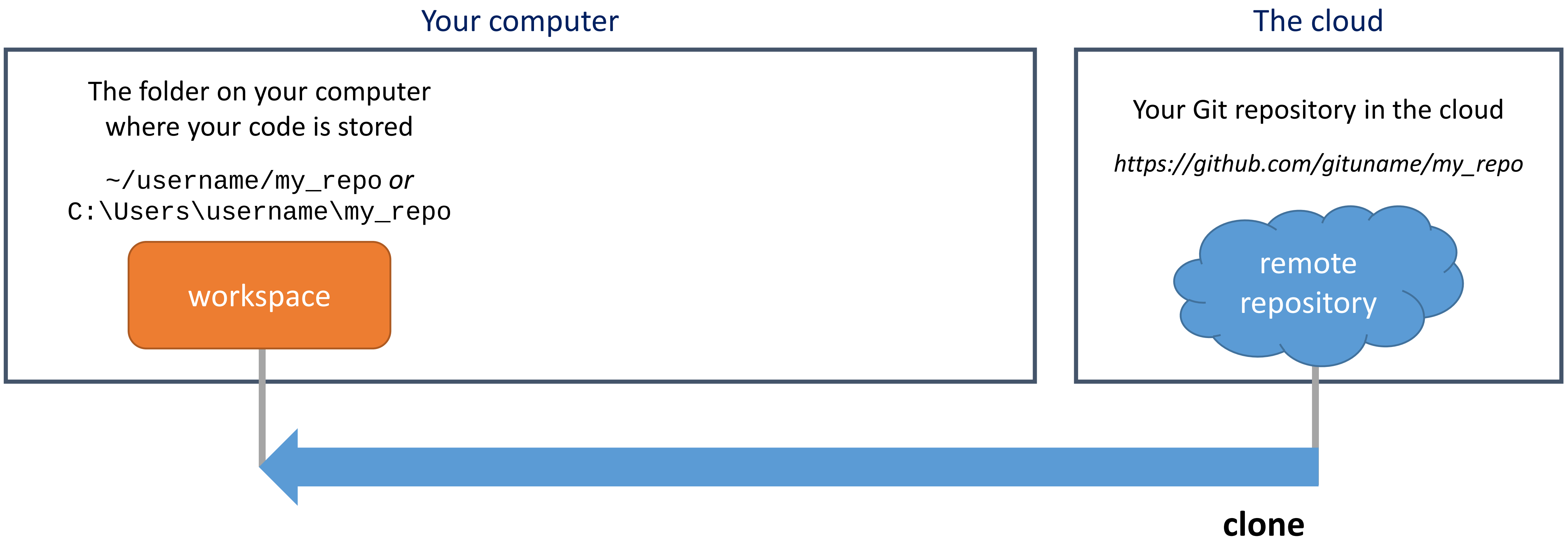
Your Git repository in the cloud

https://github.com/gituname/my_repo



https://github.com/robotics102/p0-intro-cpp-janapavlasek

Getting the code


Your computer
The cloud
The folder on your computer where your code is stored
~/username/my_repo *or* C:\Users\username\my_repo
workspace
Your Git repository in the cloud
https://github.com/gituname/my_repo
remote repository
clone

Where's my code?

Your computer

The folder on your computer where your code is stored

`~/username/my_repo` *or* `C:\Users\username\my_repo`

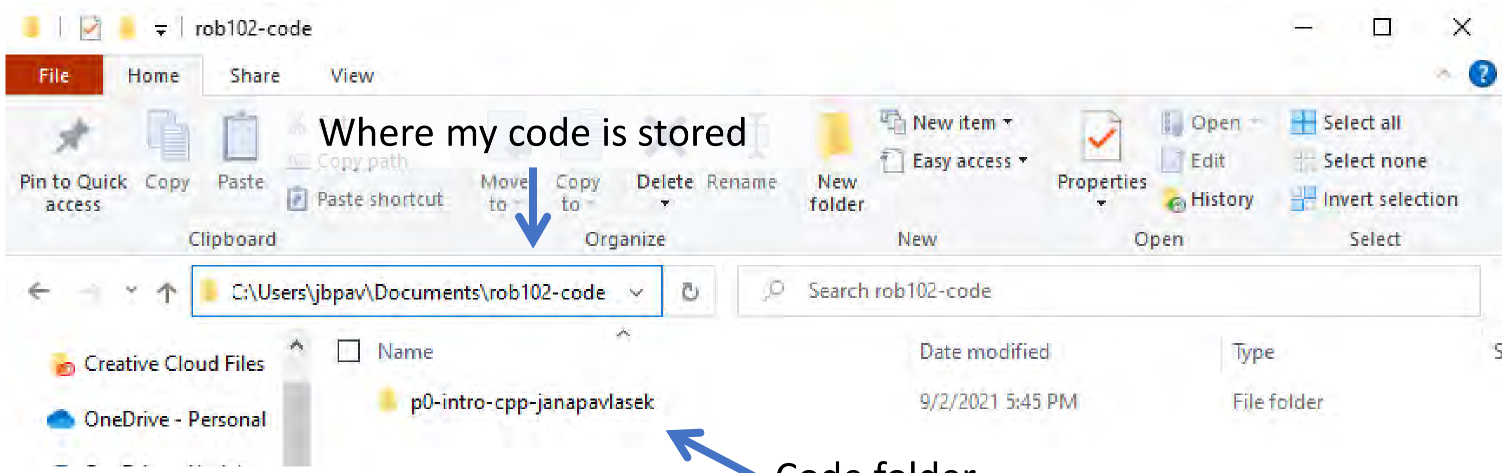
workspace

The cloud

Your Git repository in the cloud

https://github.com/robotics102/my_repo

remote repository



rob102-code

File | Home | Share | View

Pin to Quick access | Copy | Paste | Copy path | Paste shortcut | Move to | Copy to | Delete | Rename | New folder | New item | Easy access | Properties | Open | Edit | History | Select all | Select none | Invert selection

Clipboard | Organize | New | Open | Select

C:\Users\jbpav\Documents\rob102-code

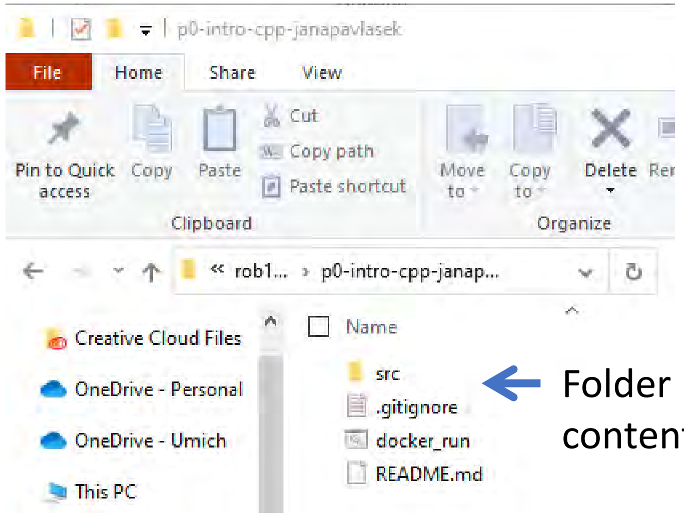
Search rob102-code

Creative Cloud Files
OneDrive - Personal

Name	Date modified	Type
p0-intro-cpp-janapavlasek	9/2/2021 5:45 PM	File folder

Where my code is stored

Code folder



p0-intro-cpp-janapavlasek

File | Home | Share | View

Pin to Quick access | Copy | Paste | Cut | Copy path | Paste shortcut | Move to | Copy to | Delete | Rer

Clipboard | Organize

« rob1... › p0-intro-cpp-janap...

Creative Cloud Files
OneDrive - Personal
OneDrive - Umich
This PC

Name
src
.gitignore
docker_run
README.md

Folder contents

Where's my code?

Your computer

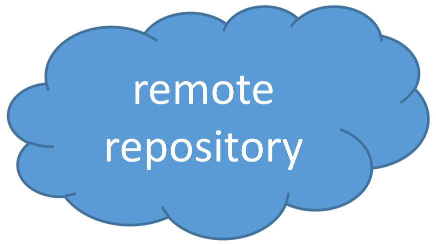
The folder on your computer where your code is stored

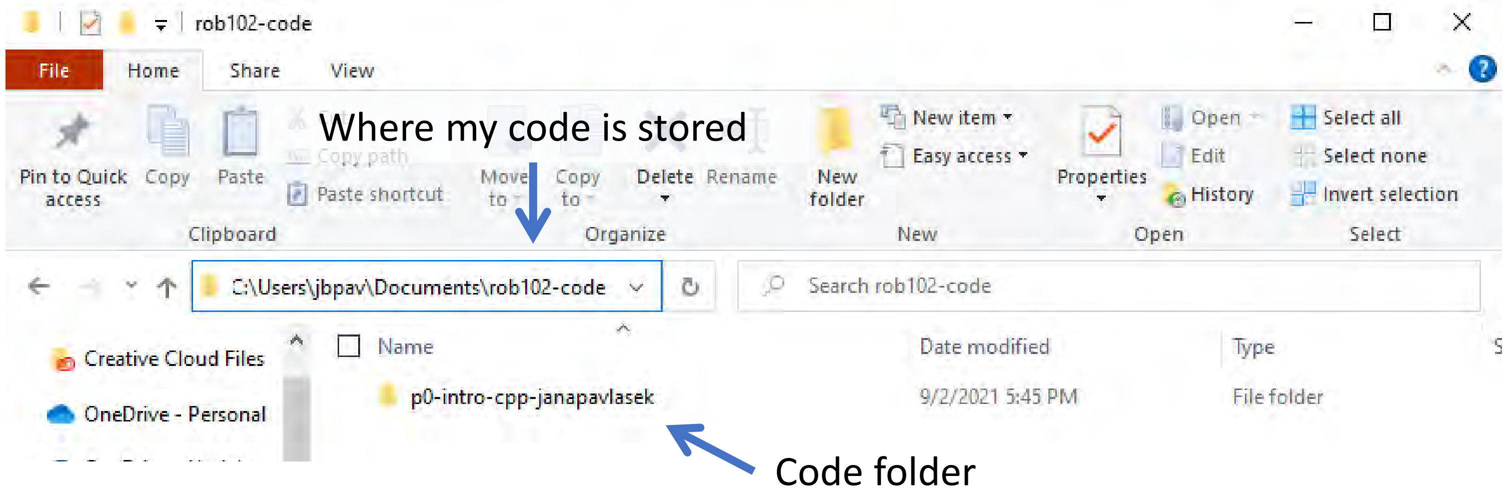
`~/username/my_repo` *or* `C:\Users\username\my_repo`

The cloud

Your Git repository in the cloud

https://github.com/robotics102/my_repo





Where my code is stored

C:\Users\jbpav\Documents\rob102-code

Search rob102-code

Name	Date modified	Type
p0-intro-cpp-janapavlasek	9/2/2021 5:45 PM	File folder

Code folder

Your computer organizes all the applications and files it contains in its **filesystem**, under different *folders* (like Documents, Downloads, etc…)

Make sure you know where in your computer's filesystem you are storing your code!!

Where's my code?

Your computer

The folder on your computer where your code is stored

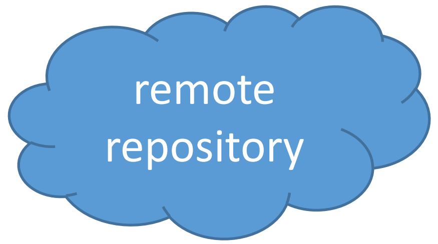
`~/username/my_repo` *or* `C:\Users\username\my_repo`

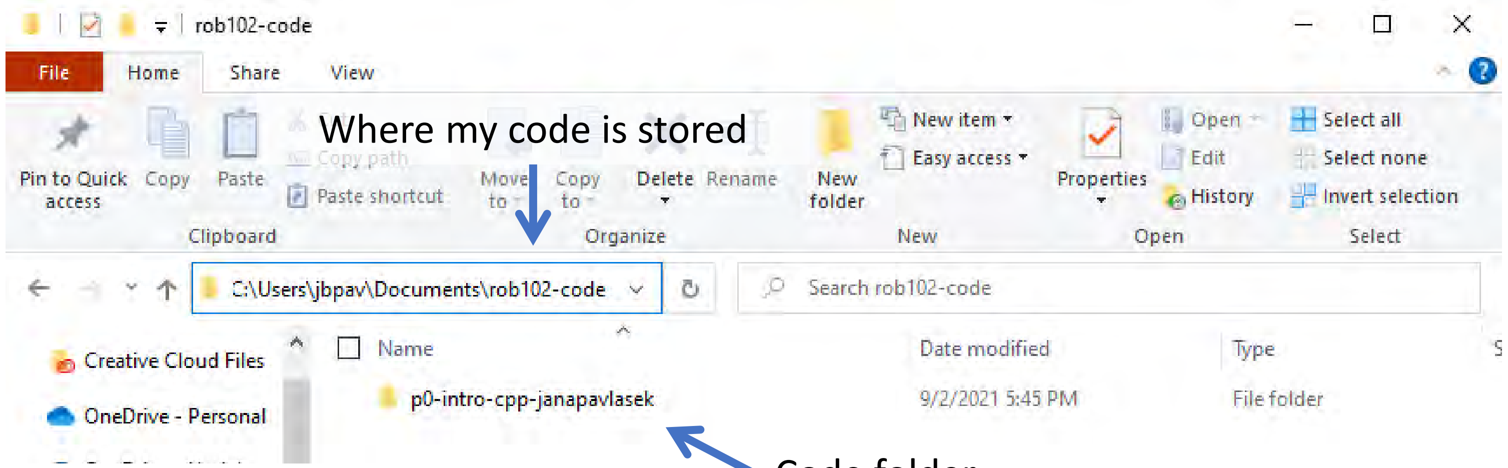
workspace

The cloud

Your Git repository in the cloud

https://github.com/robotics102/my_repo





Where my code is stored

C:\Users\jbpav\Documents\rob102-code

Name	Date modified	Type
p0-intro-cpp-janapavlasek	9/2/2021 5:45 PM	File folder

Code folder

Tip: Create a folder for all your ROB 102 code files (ex. in Documents folder).

Tip: Give the folder a name without spaces! We'll be using the command line, and spaces will be hard to deal with (ex. `rob102-code`, not `ROB 102 Code`).

Clone a repository: Your turn!

If you haven't already, make a GitHub account

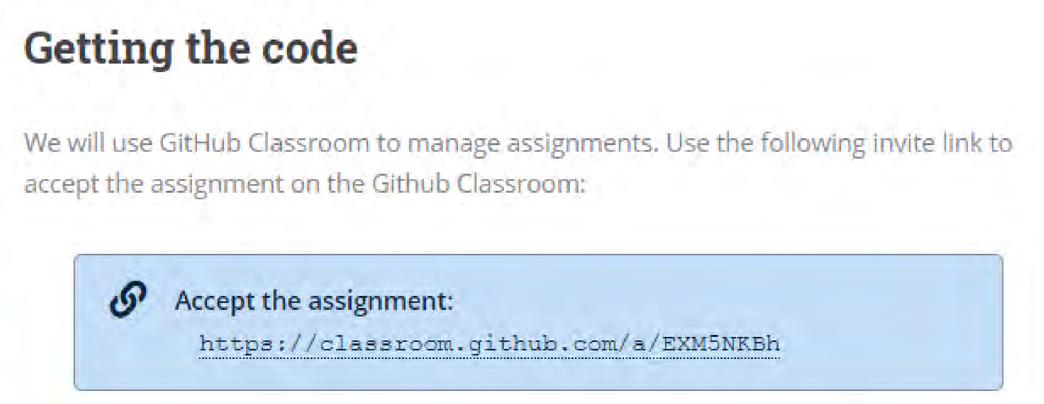
- https://github.com

If you haven't already, install Git

- https://robotics102.github.io/tutorials/setup

If you haven't already, accept the assignment for P0

- https://robotics102.github.io/projects/a0



Getting the code

We will use GitHub Classroom to manage assignments. Use the following invite link to accept the assignment on the Github Classroom:

Accept the assignment:
https://classroom.github.com/a/EXM5NKBh

1. Make a GitHub account (github.com)
2. Install Git (robotics102.github.io/tutorials/setup)
3. Accept P0 assignment (robotics102.github.io/projects/a0)

Clone a repository: Your turn!

1. Make a folder where you will store your code for ROB 102.
2. Open a terminal.
3. Use the `cd` command go to the folder you created.
4. Use the `git clone` command to clone your code.

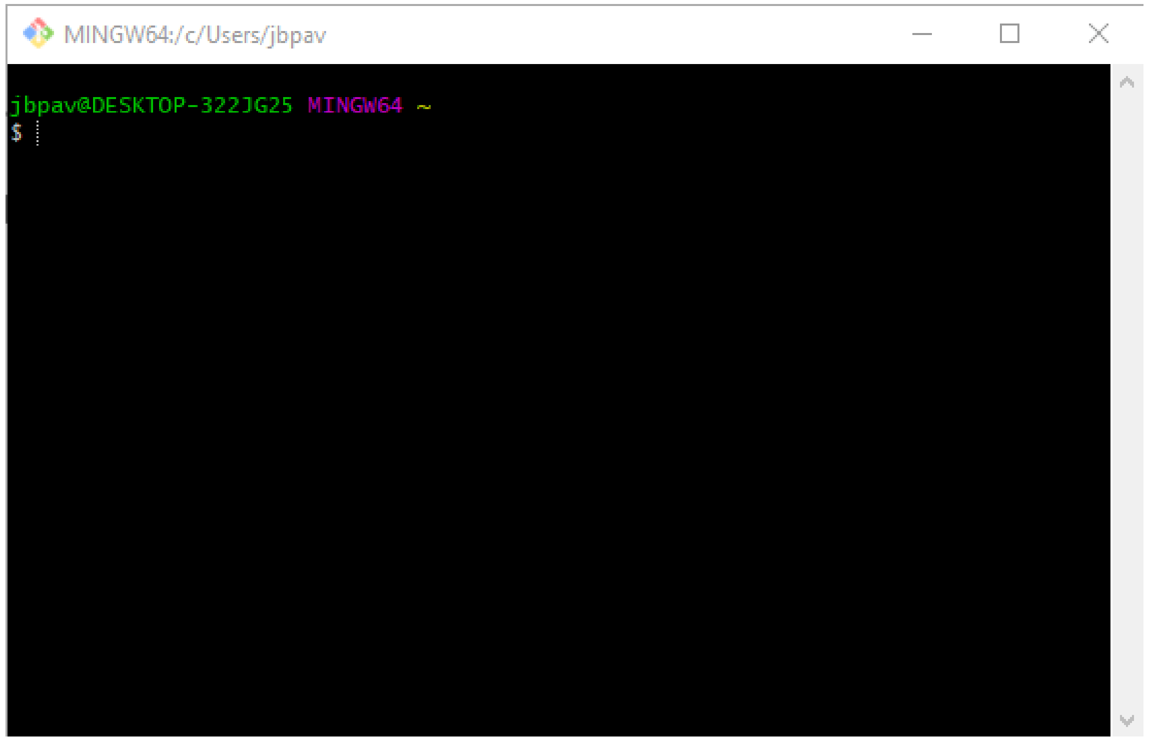
Let’s go through the steps together.

Clone a repository: Your turn!

1. Make a folder where you will store your code for ROB 102
   - Remember: No spaces!
   - Good names: `rob102-code`, `ROB102Code`, etc.
   - Bad names: `ROB 102 Code`, etc.

Clone a repository: Your turn!

1. Make a folder where you will store your code for ROB 102.
2. Open a terminal
   - Windows: Search for “Git Bash”
   - Mac: Search for “Terminal”



```
jbpav@DESKTOP-322JG25 MINGW64 ~
$
```

Clone a repository: Your turn!

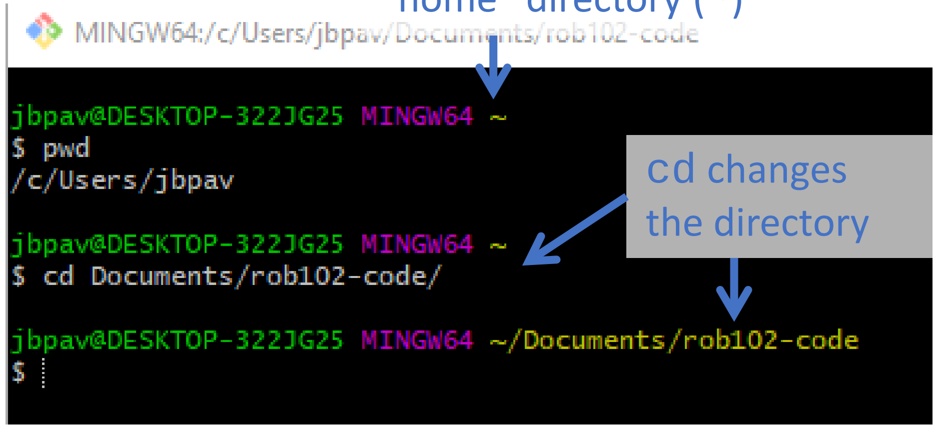
1. Make a folder where you will store your code for ROB 102
2. Open a terminal
3. Use the `cd` command go to the folder you created.

Our first terminal command!

`cd` stands for **C**hange **D**irectory.

Our terminal is always open in some directory (the "working" directory) on the filesystem.

The working directory is listed before the cursor. Or, use `pwd` to **P**rint **W**orking **D**irectory.



MINGW64:/c/Users/jbpav/Documents/rob102-code
Terminal opens in the "home" directory (~)
cd changes the directory

```
jbpav@DESKTOP-322JG25 MINGW64 ~
$ pwd
/c/Users/jbpav

jbpav@DESKTOP-322JG25 MINGW64 ~
$ cd Documents/rob102-code/

jbpav@DESKTOP-322JG25 MINGW64 ~/Documents/rob102-code
$
```

Clone a repository: Your turn!

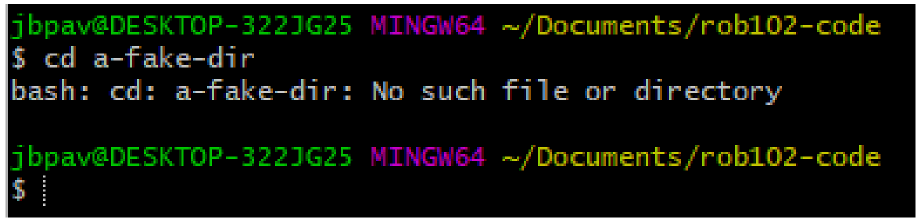
1. Make a folder where you will store your code for ROB 102
2. Open a terminal
3. Use the `cd` command go to the folder you created.

Our first terminal command!

`cd` stands for **C**hange **D**irectory.

Our terminal is always open in some directory (the "working" directory) on the filesystem.

The working directory is listed before the cursor. Or, use `pwd` to **P**rint **W**orking **D**irectory.



```
jbpav@DESKTOP-322JG25 MINGW64 ~/Documents/rob102-code
$ cd a-fake-dir
bash: cd: a-fake-dir: No such file or directory

jbpav@DESKTOP-322JG25 MINGW64 ~/Documents/rob102-code
$ 
```

The folder must exist, or you'll get an error!

Clone a repository: Your turn!

1. Make a folder where you will store your code for ROB 102
2. Open a terminal
3. Use the `cd` command go to the folder you created.
4. **Use the `git clone` command to clone your code.**

Another terminal command!

All Git terminal commands look like:

Starts with `git`

```
git cmd [arguments]
```

Some commands have arguments.

A Git command (`clone`, `commit`, `pull`…)

Remember how we didn't put spaces in our folder names? In the terminal, spaces separate different commands and arguments!

Clone a repository: Your turn!

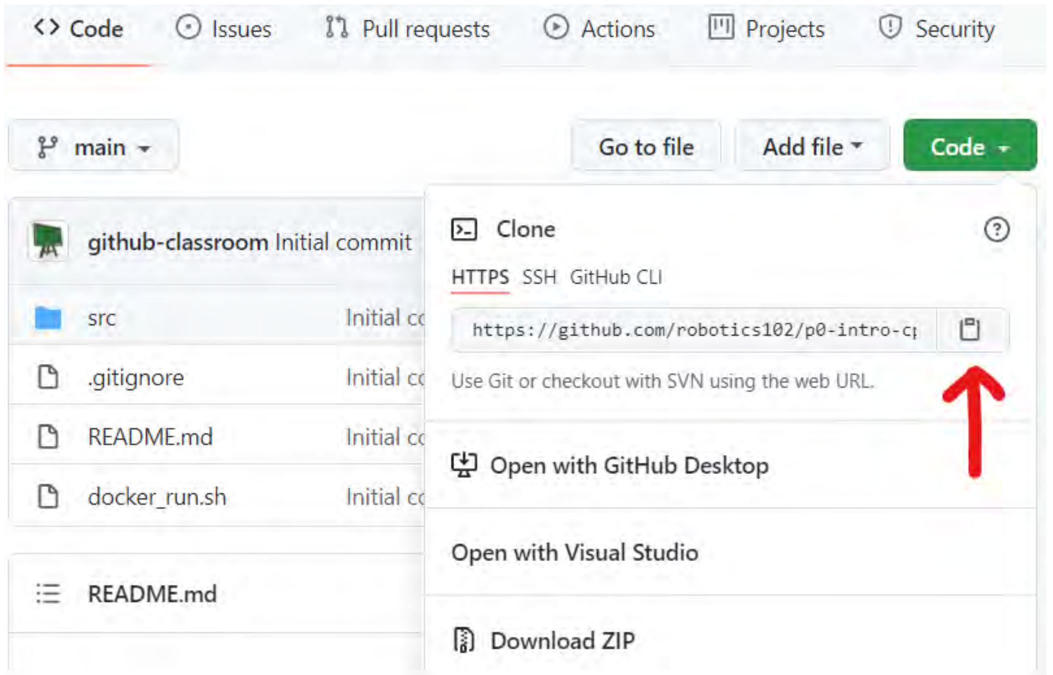
1. Make a folder where you will store your code for ROB 102
2. Open a terminal
3. Use the `cd` command go to the folder you created.
4. **Use the `git clone` command to clone your code.**

All Git terminal commands look like:

```
git *cmd* [arguments]
```

To clone your Git repository, type:

```
git clone ADDRESS-TO-YOUR-REPO
```



```
https://github.com/robotics102/p0-intro-c
```

Use Git or checkout with SVN using the web URL.

Clone a repository: Your turn!

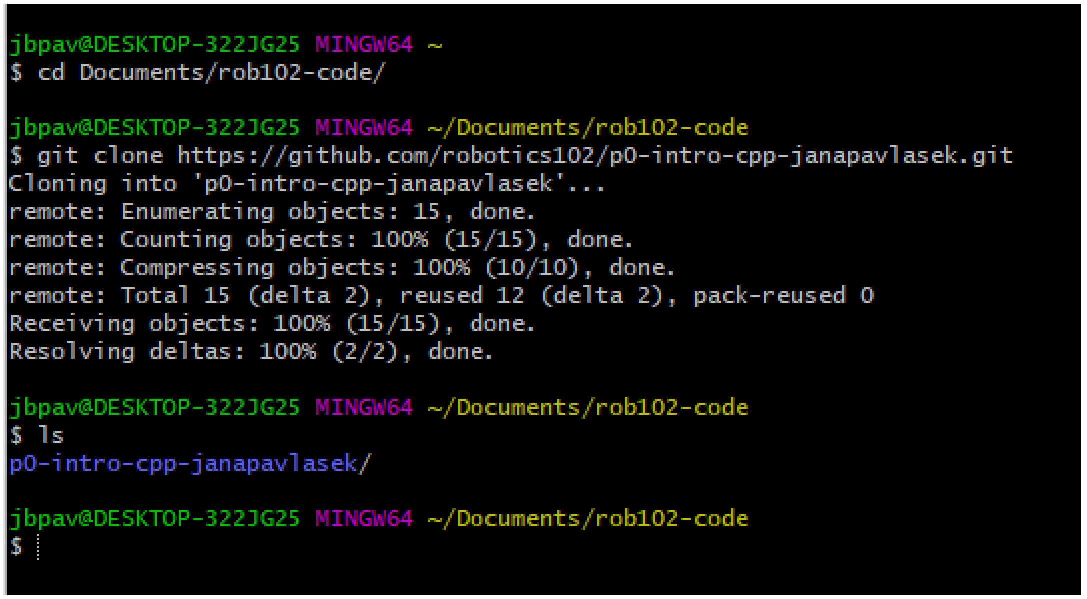
1. Make a folder where you will store your code for ROB 102
2. Open a terminal
3. Use the `cd` command go to the folder you created.
4. Use the `git clone` command to clone your code.

All Git terminal commands look like:

```
git cmd [arguments]
```

To clone your Git repository, type:

```
git clone ADDRESS-TO-YOUR-REPO
```



```
jbpav@DESKTOP-322JG25 MINGW64 ~
$ cd Documents/rob102-code/

jbpav@DESKTOP-322JG25 MINGW64 ~/Documents/rob102-code
$ git clone https://github.com/robotics102/p0-intro-cpp-janapavlasek.git
Cloning into 'p0-intro-cpp-janapavlasek'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 15 (delta 2), reused 12 (delta 2), pack-reused 0
Receiving objects: 100% (15/15), done.
Resolving deltas: 100% (2/2), done.

jbpav@DESKTOP-322JG25 MINGW64 ~/Documents/rob102-code
$ ls
p0-intro-cpp-janapavlasek/

jbpav@DESKTOP-322JG25 MINGW64 ~/Documents/rob102-code
$
```

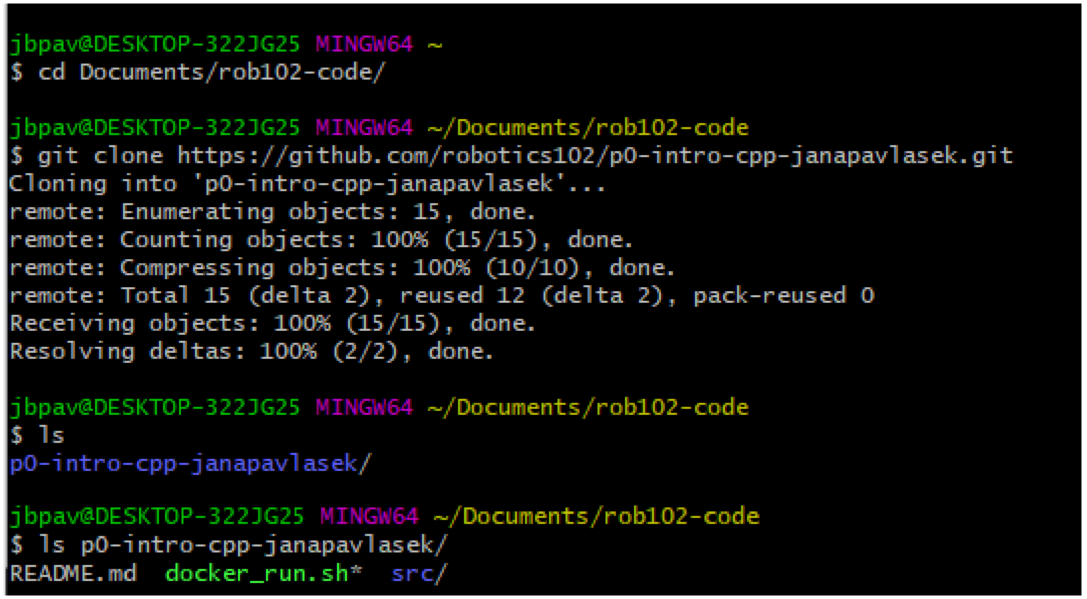
Clone a repository: Your turn!

1. Make a folder where you will store your code for ROB 102
2. Open a terminal
3. Use the `cd` command go to the folder you created.
4. Use the `git clone` command to clone your code.

`ls` is another useful terminal command to list all the files in a directory

My directory now contains a single folder with the same name as the repository

Adding the name of a folder lists the contents of that folder



```
jbpav@DESKTOP-322JG25 MINGW64 ~
$ cd Documents/rob102-code/

jbpav@DESKTOP-322JG25 MINGW64 ~/Documents/rob102-code
$ git clone https://github.com/robotics102/p0-intro-cpp-janapavlasek.git
Cloning into 'p0-intro-cpp-janapavlasek'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 15 (delta 2), reused 12 (delta 2), pack-reused 0
Receiving objects: 100% (15/15), done.
Resolving deltas: 100% (2/2), done.

jbpav@DESKTOP-322JG25 MINGW64 ~/Documents/rob102-code
$ ls
p0-intro-cpp-janapavlasek/

jbpav@DESKTOP-322JG25 MINGW64 ~/Documents/rob102-code
$ ls p0-intro-cpp-janapavlasek/
README.md  docker_run.sh*  src/
```

Clone a repository: Your turn!

1. Make a GitHub account (github.com)
2. Install Git (robotics102.github.io/tutorials/setup)
3. Accept P0 assignment (robotics102.github.io/projects/a0)

1. Make a folder where you will store your code for ROB 102
2. Open a terminal
3. Use the `cd` command go to the folder you created.
4. Use the `git clone` command to clone your code.

We learned:

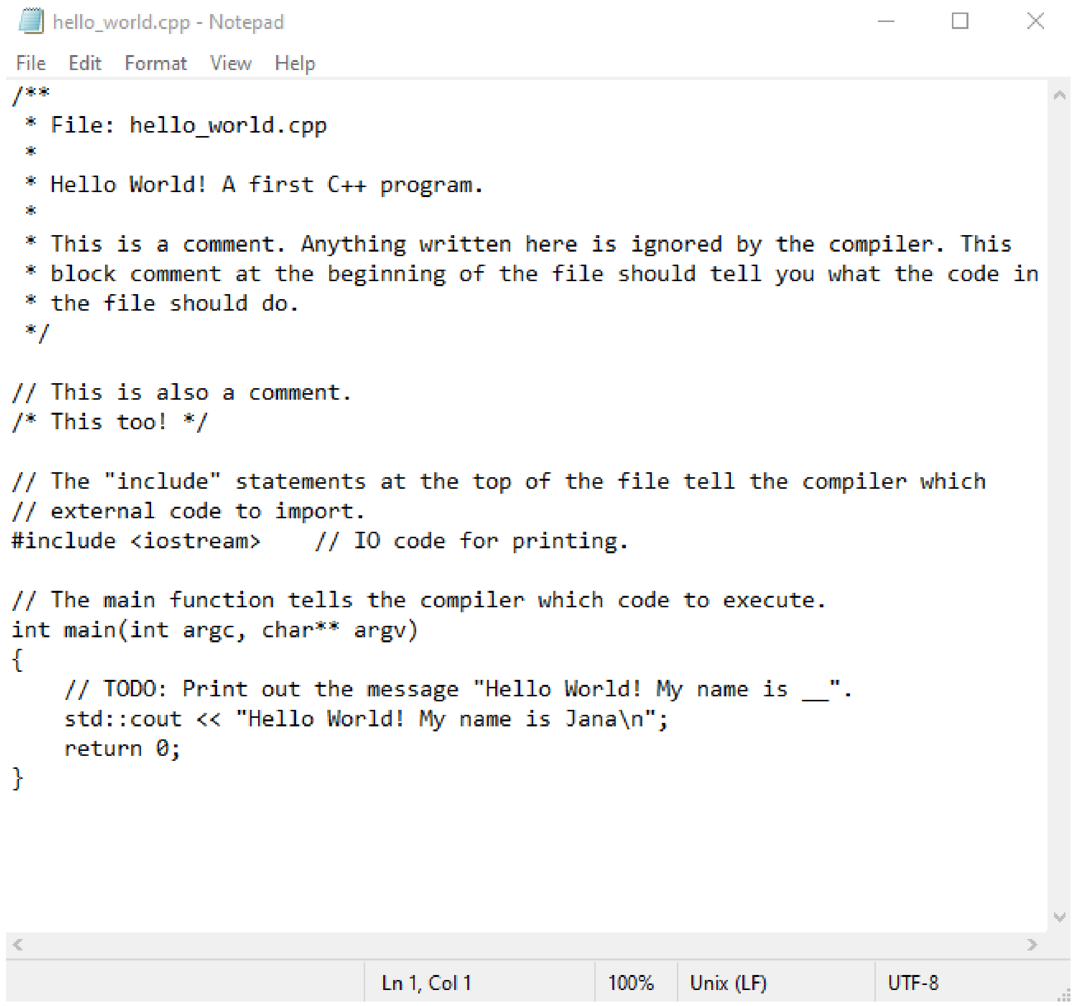
- ✓ How to change directories in a terminal with `cd`
- ✓ How to list files in a directory with `ls`
- ✓ How to clone a Git repository with `git clone`

Editing your code

We will need some way to view and edit the code files in the repository.

Idea #1: Notepad

This would work, but there's a much better way!



```
/**
 * File: hello_world.cpp
 *
 * Hello World! A first C++ program.
 *
 * This is a comment. Anything written here is ignored by the compiler. This
 * block comment at the beginning of the file should tell you what the code in
 * the file should do.
 */

// This is also a comment.
/* This too! */

// The "include" statements at the top of the file tell the compiler which
// external code to import.
#include <iostream>    // IO code for printing.

// The main function tells the compiler which code to execute.
int main(int argc, char** argv)
{
    // TODO: Print out the message "Hello World! My name is __".
    std::cout << "Hello World! My name is Jana\n";
    return 0;
}
```

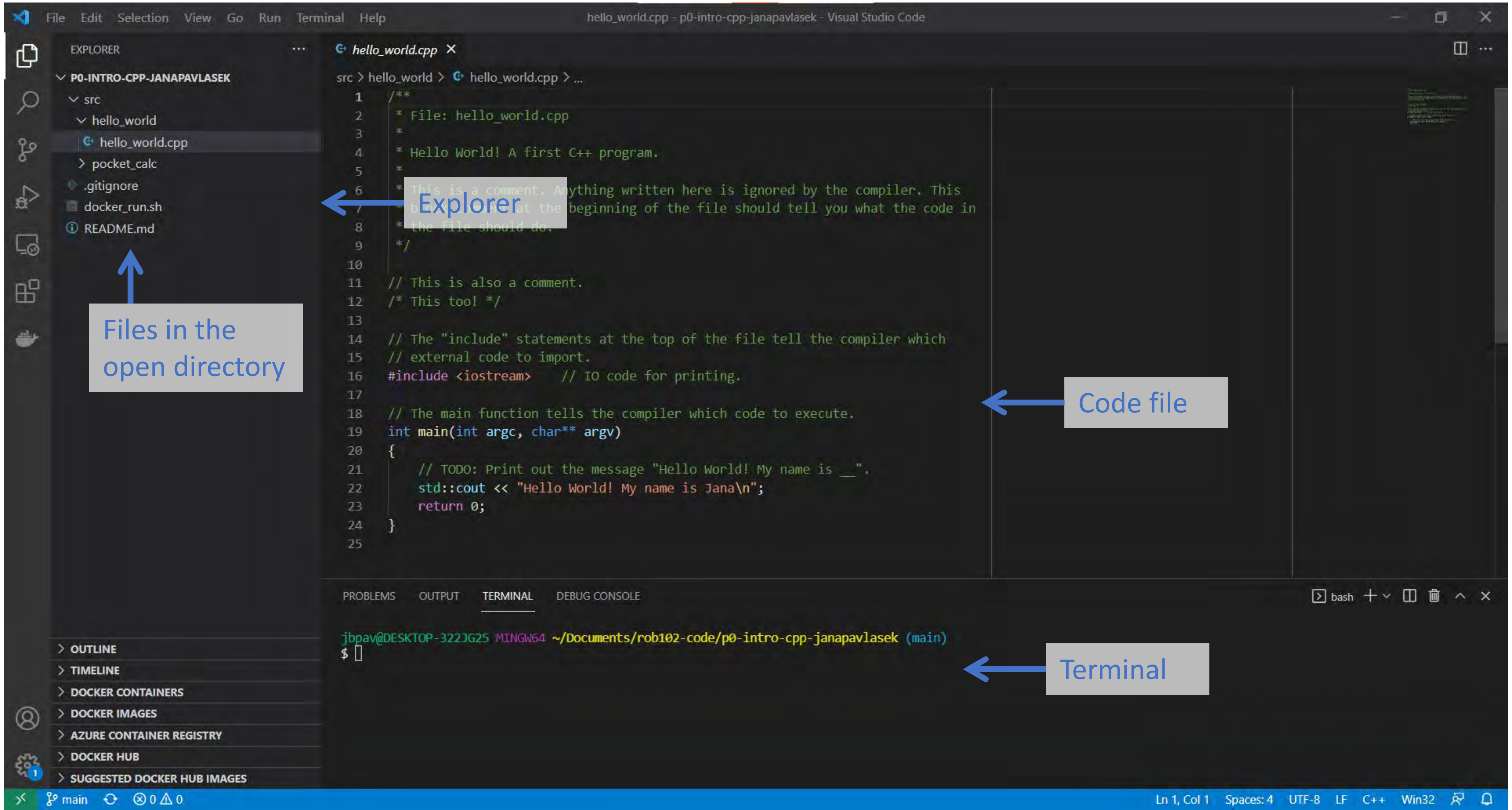
Visual Studio Code

VSCode (short for Visual Studio Code) is an Integrated Development Environment (IDE).

It has many features, like a built-in terminal, syntax highlighting, Git diff highlighting (shows you what code changed since you last committed), and much more!

Plus, you can install extensions to get even more features.

VSCode



File Edit Selection View Go Run Terminal Help
hello_world.cpp - p0-intro-cpp-janapavlasek - Visual Studio Code
EXPLORER
P0-INTRO-CPP-JANAPAVLASEK
src
hello_world
hello_world.cpp
pocket_calc
.gitignore
docker_run.sh
README.md
OUTLINE
TIMELINE
DOCKER CONTAINERS
DOCKER IMAGES
AZURE CONTAINER REGISTRY
DOCKER HUB
SUGGESTED DOCKER HUB IMAGES
src > hello_world > hello_world.cpp > ...
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
bash
main
Ln 1, Col 1 Spaces: 4 UTF-8 LF C++ Win32
Explorer
Files in the open directory
Code file
Terminal

```cpp
/**
 * File: hello_world.cpp
 *
 * Hello World! A first C++ program.
 *
 * This is a comment. Anything written here is ignored by the compiler. This
 * block comment at the beginning of the file should tell you what the code in
 * the file should do.
 */

// This is also a comment.
/* This too! */

// The "include" statements at the top of the file tell the compiler which
// external code to import.
#include <iostream>    // IO code for printing.

// The main function tells the compiler which code to execute.
int main(int argc, char** argv)
{
    // TODO: Print out the message "Hello World! My name is __".
    std::cout << "Hello World! My name is Jana\n";
    return 0;
}
```

```
jbpav@DESKTOP-322JG25 MINGW64 ~/Documents/rob102-code/p0-intro-cpp-janapavlasek (main)
$ 
```

VSCode: Your turn!

1. If you haven’t already, install VSCode
   - https://robotics102.github.io/tutorials/setup
2. Install the C++ extension
3. Open your repository in VSCode

Careful: There is another IDE called “Visual Studio.” We are not using that one. If you’re searching for information on VSCode, make sure you aren’t looking at Visual Studio.

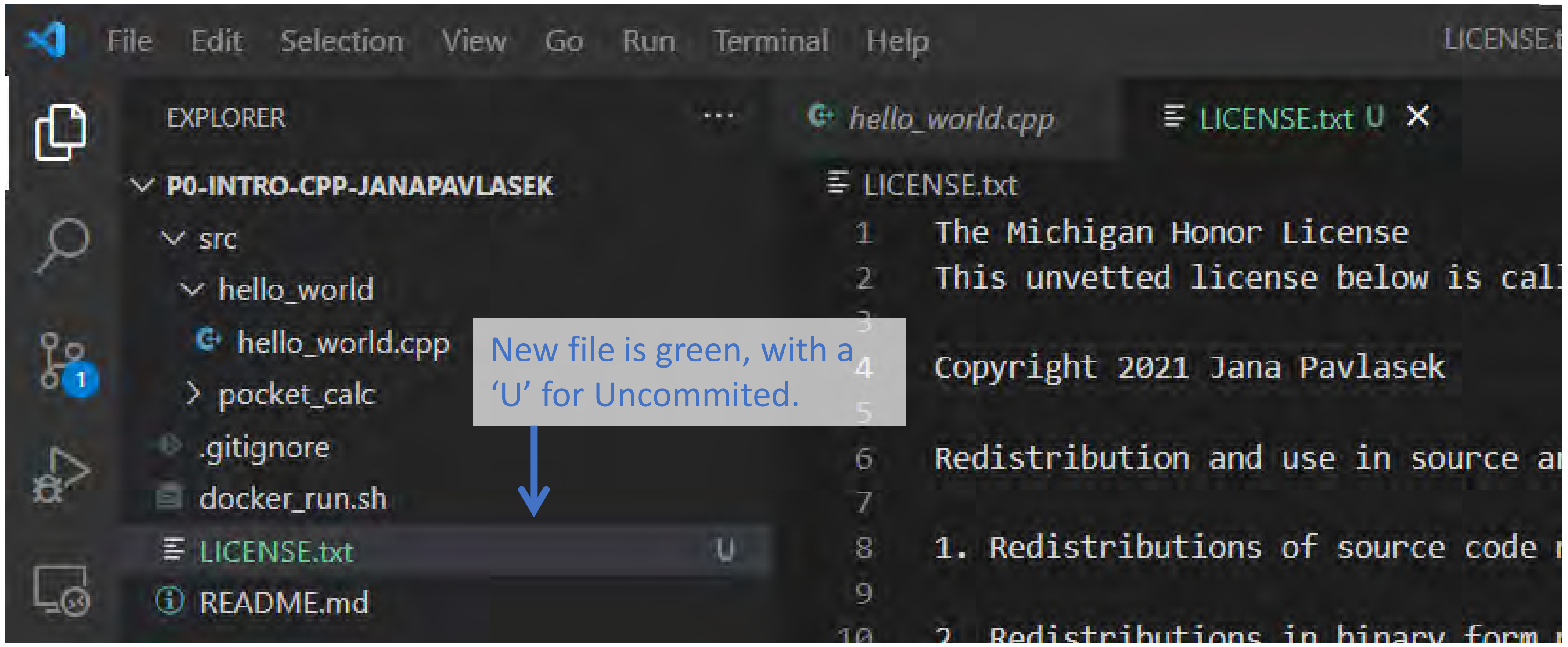
Adding the License

Now that we can edit code, let’s make a change to the repository.

1. In VSCode, make a new file (File > New File)
2. Go to autorob.org/MichiganHonorLicense.txt
3. Copy the contents into the new file
4. Replace <YEAR> with 2021 and <COPYRIGHT HOLDER> with your name
5. Save the file with the name “LICENSE.txt”

Adding the License

Now that we can edit code, let’s make a change to the repository.



```
The Michigan Honor License
This unvetted license below is cal

Copyright 2021 Jana Pavlasek

Redistribution and use in source a

1. Redistributions of source code 

2. Redistributions in binary form 
```

Git Workflow: Pushing Changes

Your computer

The folder on your computer where your code is stored

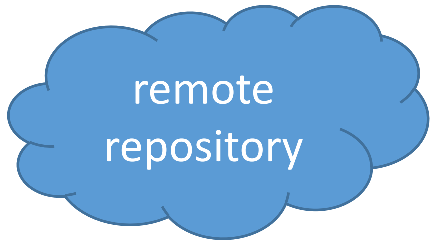
`~/username/my_repo` *or* `C:\Users\username\my_repo`

workspace

The cloud

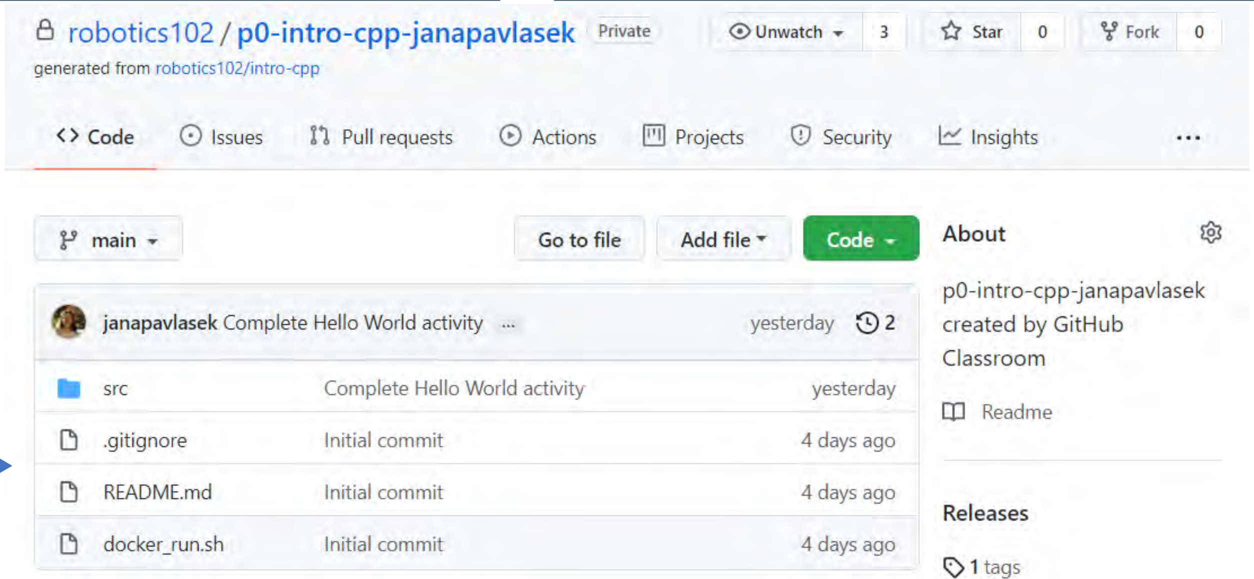
Your Git repository in the cloud

https://github.com/gituname/my_repo



Our workspace has changes which are not synced to the remote repository!

No LICENSE.txt →



robotics102 / p0-intro-cpp-janapavlasek Private

generated from robotics102/intro-cpp

janapavlasek Complete Hello World activity ... yesterday 2

src	Complete Hello World activity	yesterday
.gitignore	Initial commit	4 days ago
README.md	Initial commit	4 days ago
docker_run.sh	Initial commit	4 days ago

About

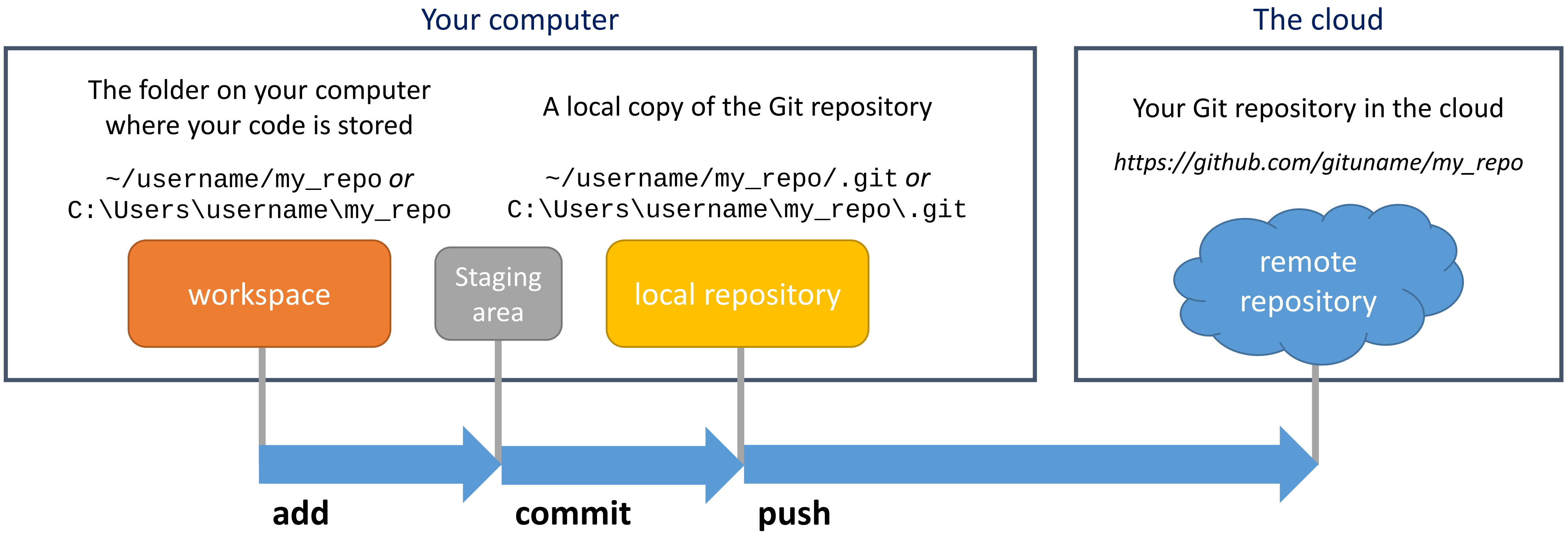
p0-intro-cpp-janapavlasek created by GitHub Classroom

Readme

Releases

1 tags

Git Workflow: Pushing Changes


Your computer
The cloud
The folder on your computer where your code is stored
~/username/my_repo *or* C:\Users\username\my_repo
A local copy of the Git repository
~/username/my_repo/.git *or* C:\Users\username\my_repo\.git
Your Git repository in the cloud
https://github.com/gituname/my_repo
workspace
Staging area
local repository
remote repository
add
commit
push

We need to **commit** our changes to our local repository and then **push** them to the remote repository.

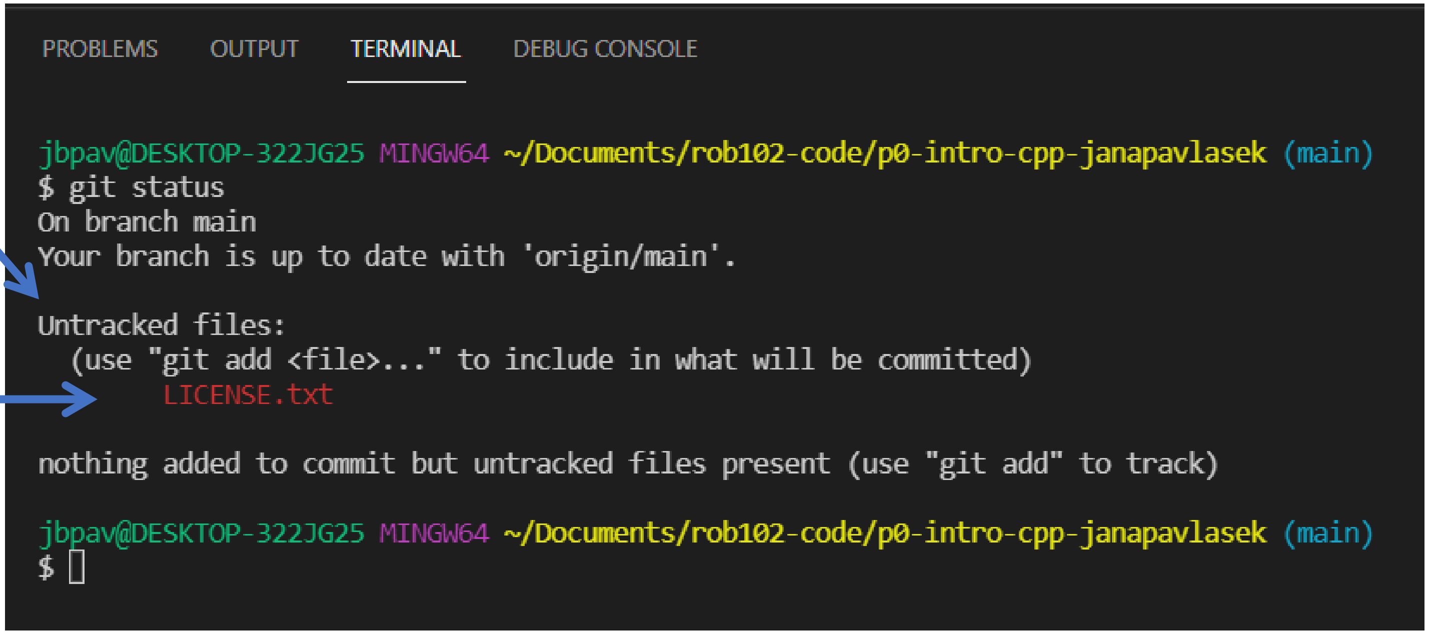
Git Workflow: Pushing Changes

Open a terminal in VSCode and type:

```
git status
```

Untracked means this is a **new** file

Shows which files have been added or changed



```
jbpav@DESKTOP-322JG25 MINGW64 ~/Documents/rob102-code/p0-intro-cpp-janapavlasek (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        LICENSE.txt

nothing added to commit but untracked files present (use "git add" to track)

jbpav@DESKTOP-322JG25 MINGW64 ~/Documents/rob102-code/p0-intro-cpp-janapavlasek (main)
$ 
```

Notice: VSCode opens a terminal in your open code folder by default.

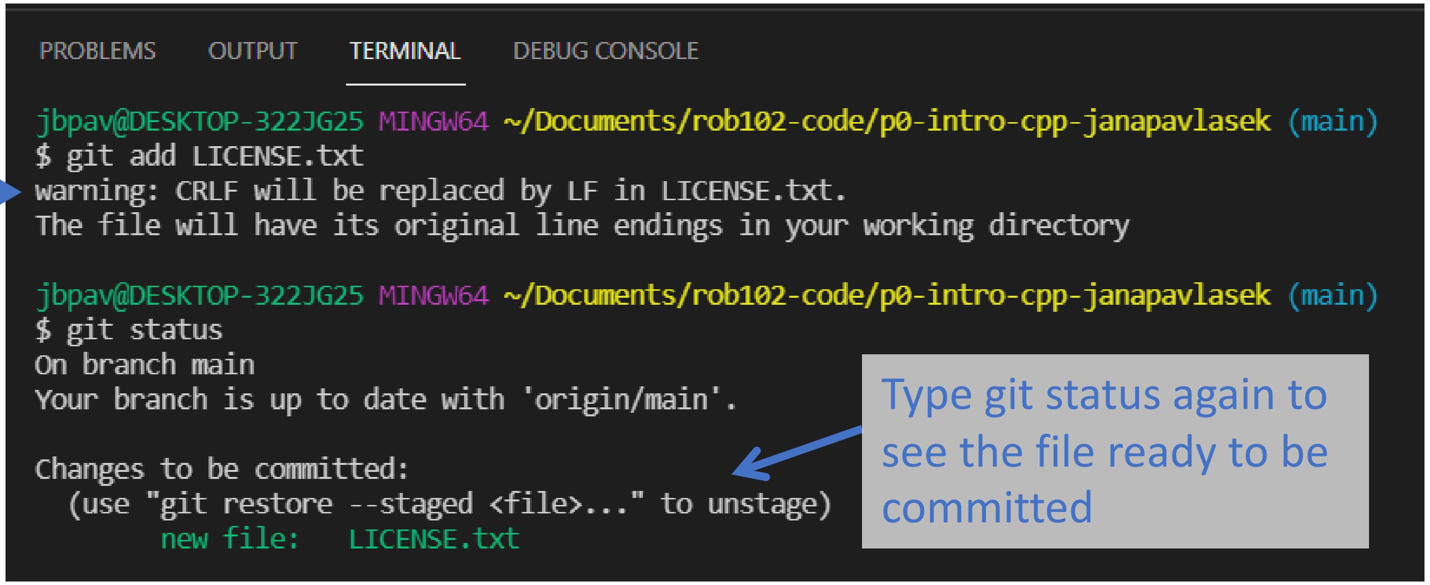
Git Workflow: Staging Changes

Argument: name of file to add.

In your VSCode terminal, type:

```
git add LICENSE.txt
```

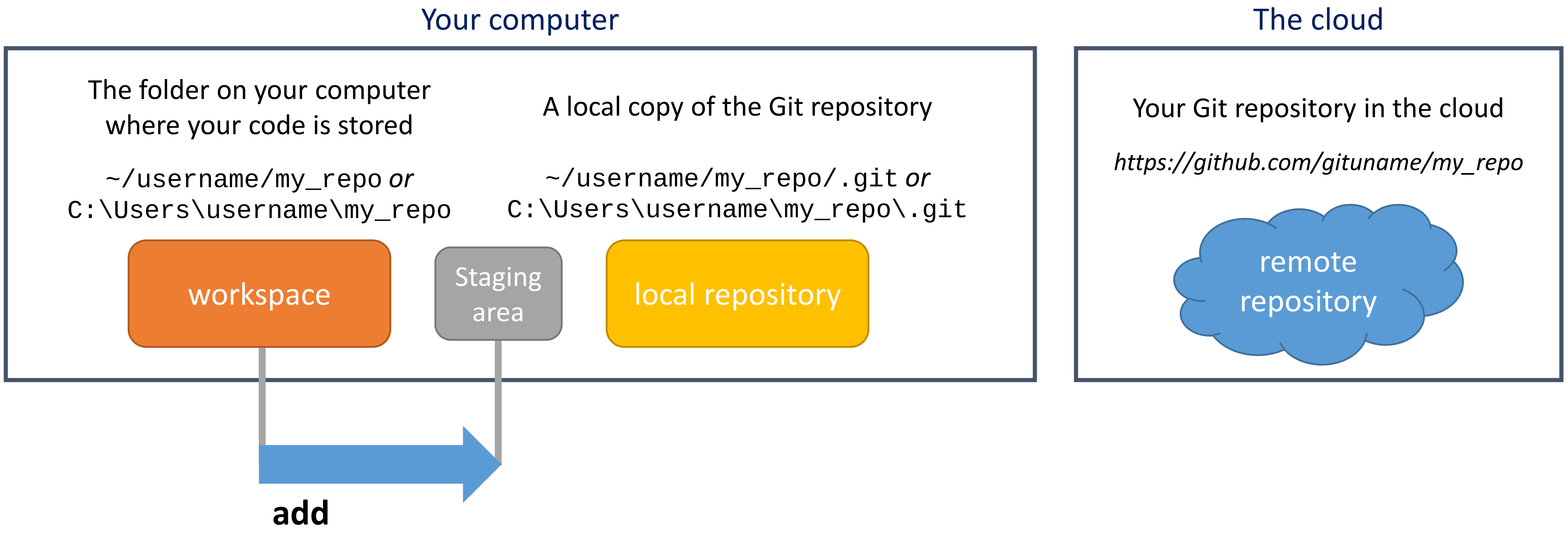
If you see this warning, ignore it!



```
jbpav@DESKTOP-322JG25 MINGW64 ~/Documents/rob102-code/p0-intro-cpp-janapavlasek (main)
$ git add LICENSE.txt
warning: CRLF will be replaced by LF in LICENSE.txt.
The file will have its original line endings in your working directory

jbpav@DESKTOP-322JG25 MINGW64 ~/Documents/rob102-code/p0-intro-cpp-janapavlasek (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   LICENSE.txt
```

Type git status again to see the file ready to be committed

Git Workflow: Staging Changes


Your computer
The cloud
The folder on your computer where your code is stored
~/username/my_repo or C:\Users\username\my_repo
A local copy of the Git repository
~/username/my_repo/.git or C:\Users\username\my_repo\.git
Your Git repository in the cloud
https://github.com/gituname/my_repo
workspace
Staging area
local repository
remote repository
add

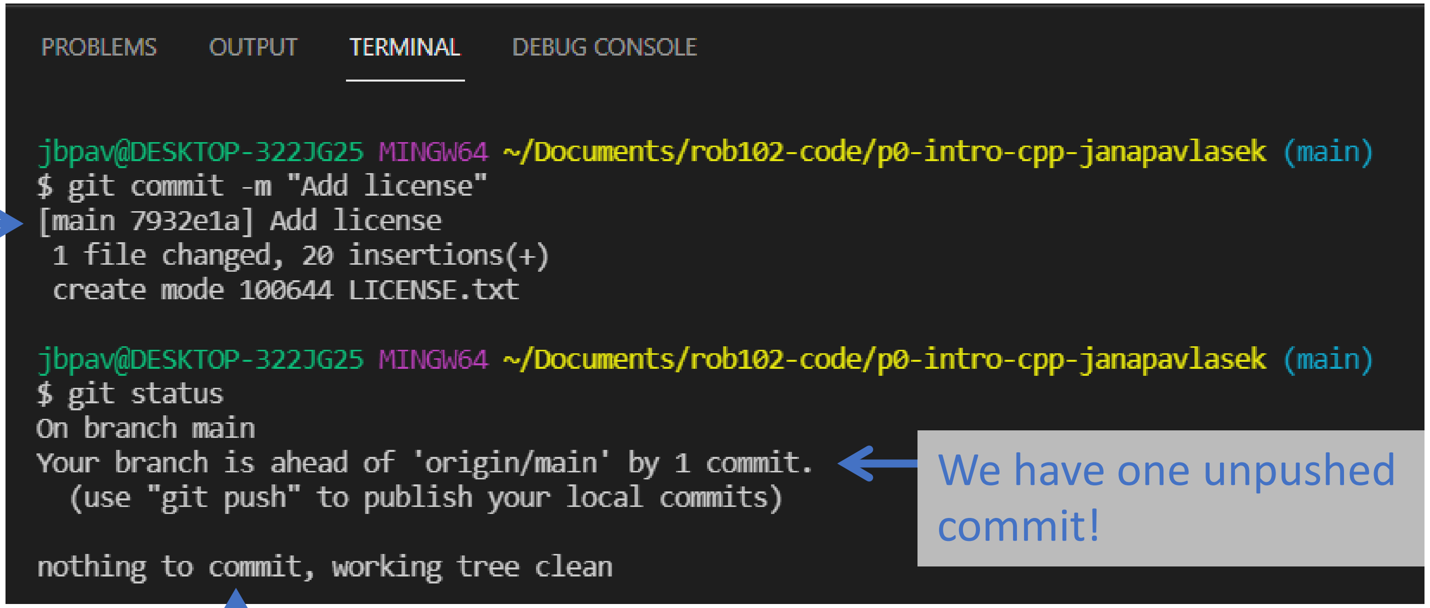
Now our change is **staged** and ready to be committed!

Git Workflow: Committing Changes

Argument: commit message.
Make the message descriptive!

In your VSCode terminal, type:

```
git commit -m “Add license”
```



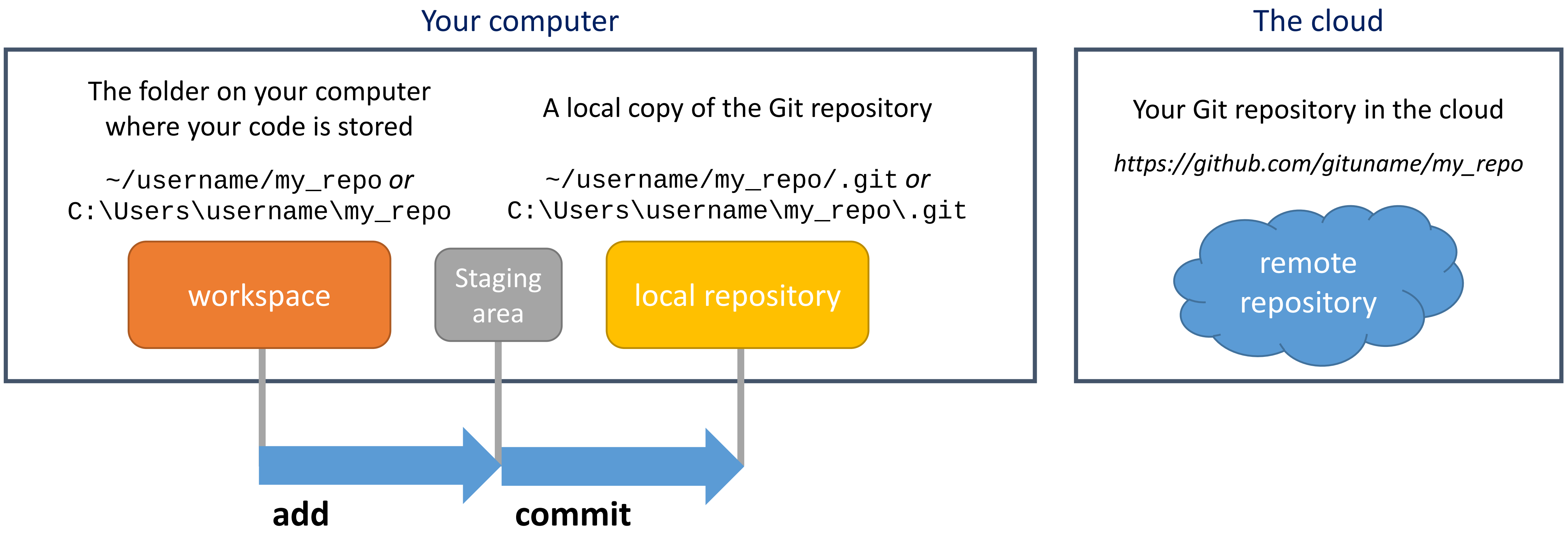
```
jbpav@DESKTOP-322JG25 MINGW64 ~/Documents/rob102-code/p0-intro-cpp-janapavlasek (main)
$ git commit -m "Add license"
[main 7932e1a] Add license
 1 file changed, 20 insertions(+)
 create mode 100644 LICENSE.txt

jbpav@DESKTOP-322JG25 MINGW64 ~/Documents/rob102-code/p0-intro-cpp-janapavlasek (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

Commit added to the “main” branch

We have one unpushed commit!

No more uncommitted changes in our workspace

Git Workflow: Committing Changes

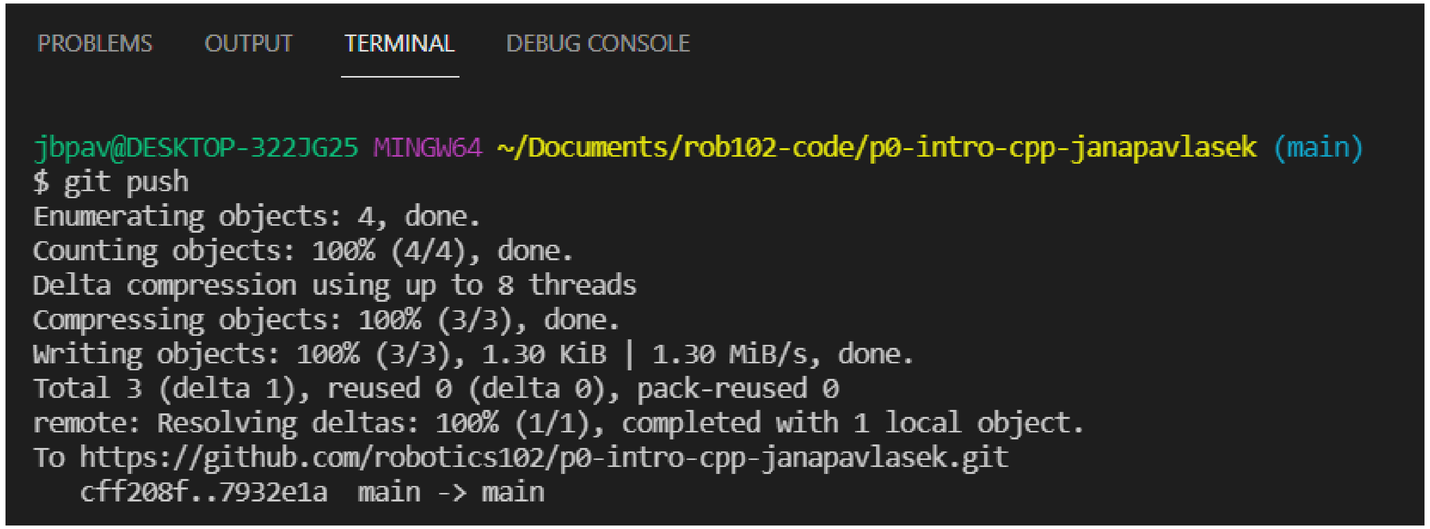

Your computer
The cloud
The folder on your computer where your code is stored
~/username/my_repo or C:\Users\username\my_repo
A local copy of the Git repository
~/username/my_repo/.git or C:\Users\username\my_repo\.git
Your Git repository in the cloud
https://github.com/gituname/my_repo
workspace
Staging area
local repository
remote repository
add
commit

Now our change is **committed** and ready to be **pushed**!

Git Workflow: Pushing Changes

In your VSCode terminal, type:

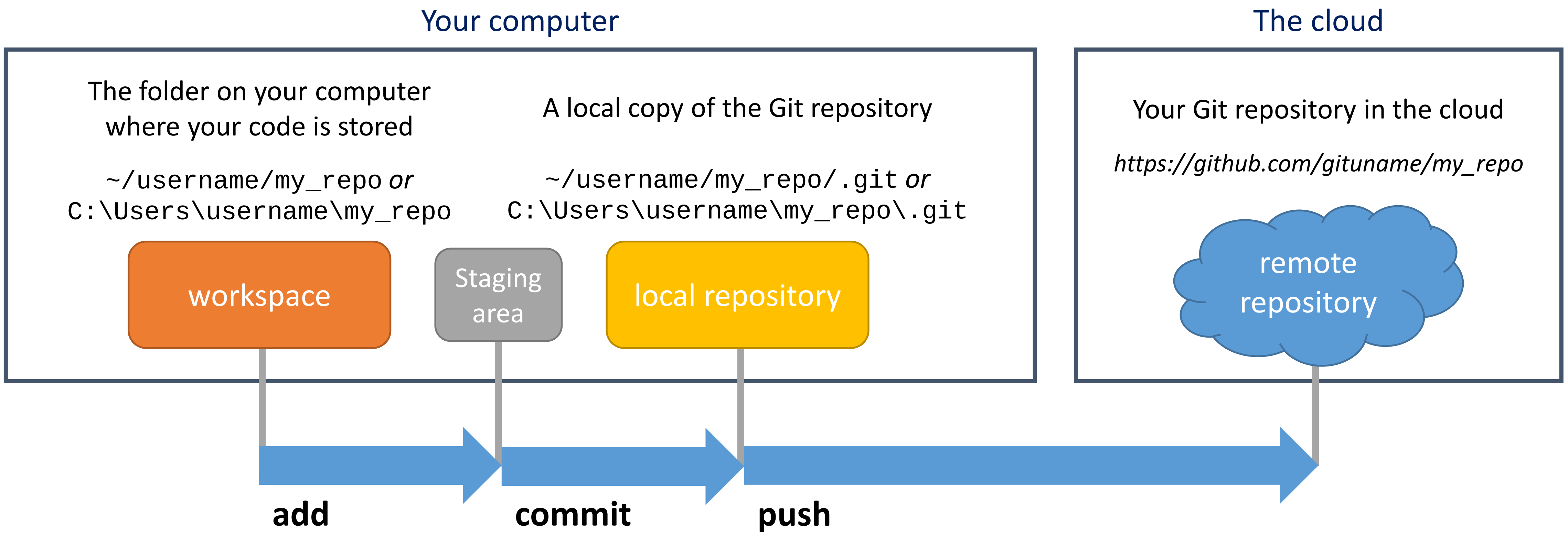
```
git push
```



```
jbpav@DESKTOP-322JG25 MINGW64 ~/Documents/rob102-code/p0-intro-cpp-janapavlasek (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.30 KiB | 1.30 MiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/robotics102/p0-intro-cpp-janapavlasek.git
   cff208f..7932e1a  main -> main
```

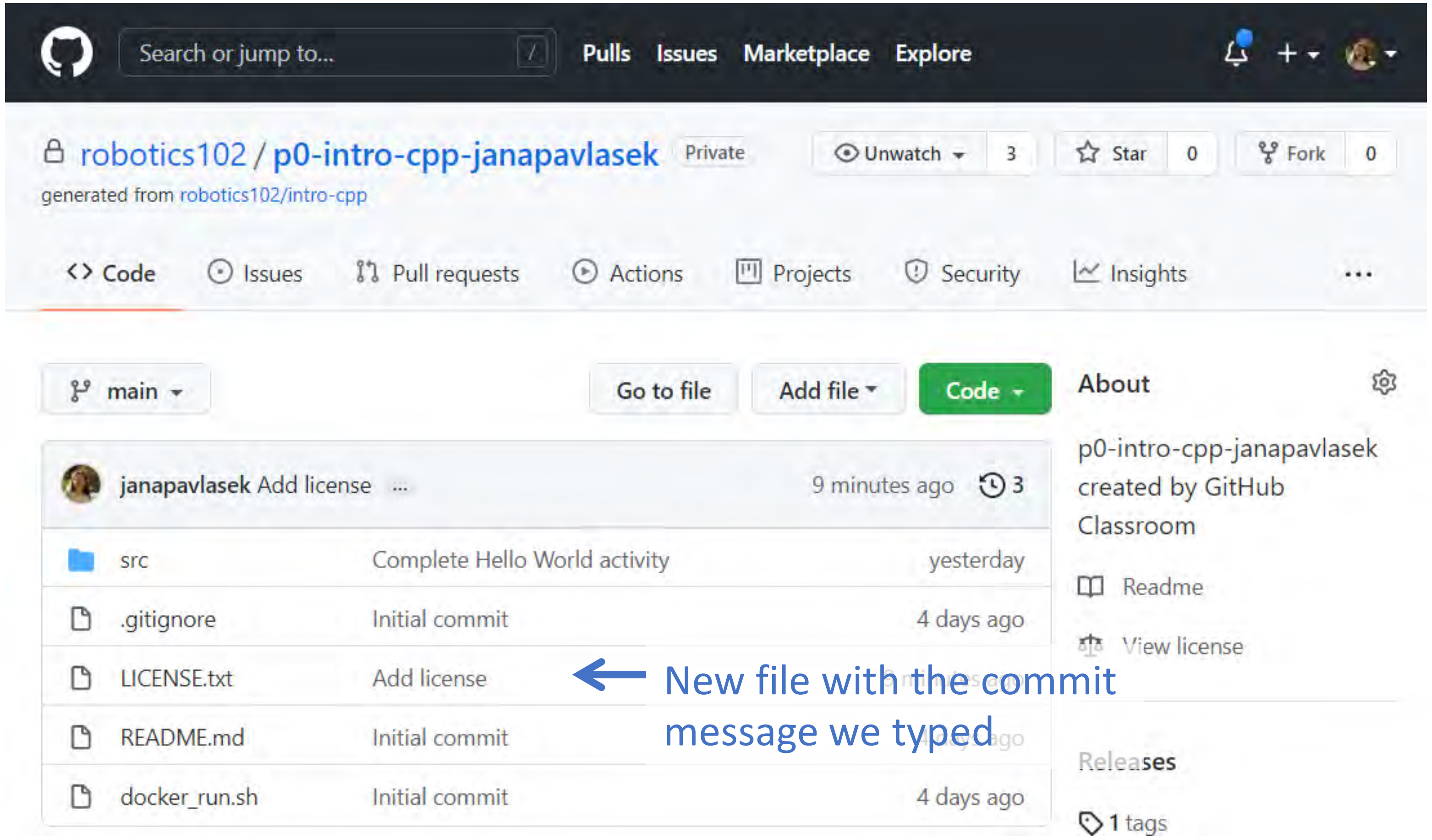
Commit pushed to “main” branch!

Git Workflow: Pushing Changes


Your computer
The cloud
The folder on your computer where your code is stored
~/username/my_repo *or* C:\Users\username\my_repo
A local copy of the Git repository
~/username/my_repo/.git *or* C:\Users\username\my_repo\.git
Your Git repository in the cloud
https://github.com/gituname/my_repo
workspace
Staging area
local repository
remote repository
add
commit
push

Now our license file is synced to the remote repository in the cloud!

Git Workflow: Pushing Changes



Search or jump to... Pulls Issues Marketplace Explore

robotics102 / p0-intro-cpp-janapavlasek Private

generated from robotics102/intro-cpp

Unwatch 3 Star 0 Fork 0

Code Issues Pull requests Actions Projects Security Insights

main Go to file Add file Code

janapavlasek Add license ... 9 minutes ago 3

src	Complete Hello World activity	yesterday
.gitignore	Initial commit	4 days ago
LICENSE.txt	Add license	
README.md	Initial commit	
docker_run.sh	Initial commit	4 days ago

About

p0-intro-cpp-janapavlasek created by GitHub Classroom

Readme

View license

Releases

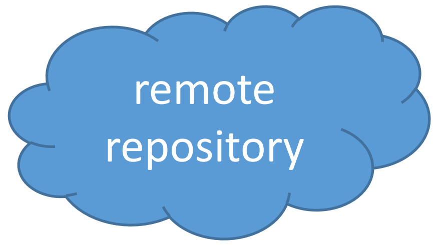
1 tags

← New file with the commit message we typed

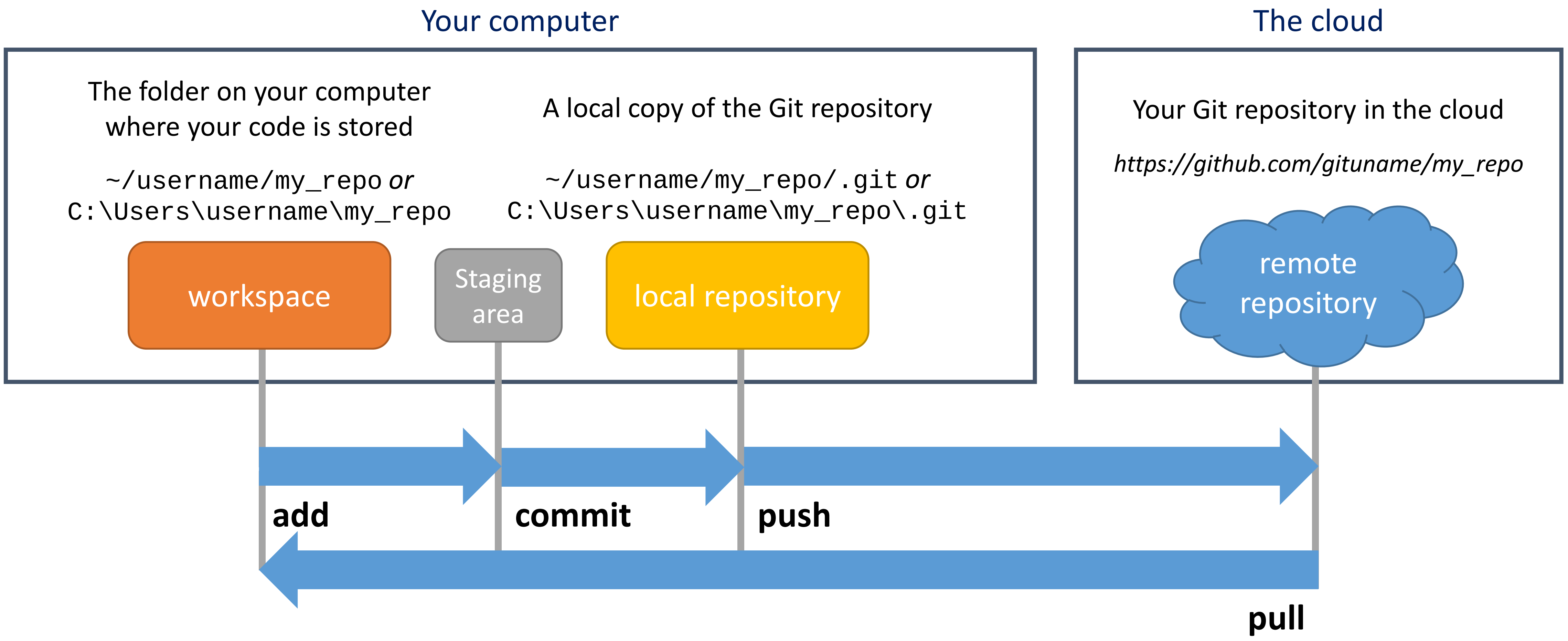
The cloud

Your Git repository in the cloud

https://github.com/gituname/my_repo



Git Workflow: Pulling Changes


Your computer
The cloud
The folder on your computer where your code is stored
~/username/my_repo *or* C:\Users\username\my_repo
A local copy of the Git repository
~/username/my_repo/.git *or* C:\Users\username\my_repo\.git
Your Git repository in the cloud
https://github.com/gituname/my_repo
workspace
Staging area
local repository
remote repository
add
commit
push
pull

If someone else (the instructor, a teammate, you on a different computer…) has pushed changes to your repository, you can **pull** them using the command `git pull`

Running your code: Docker

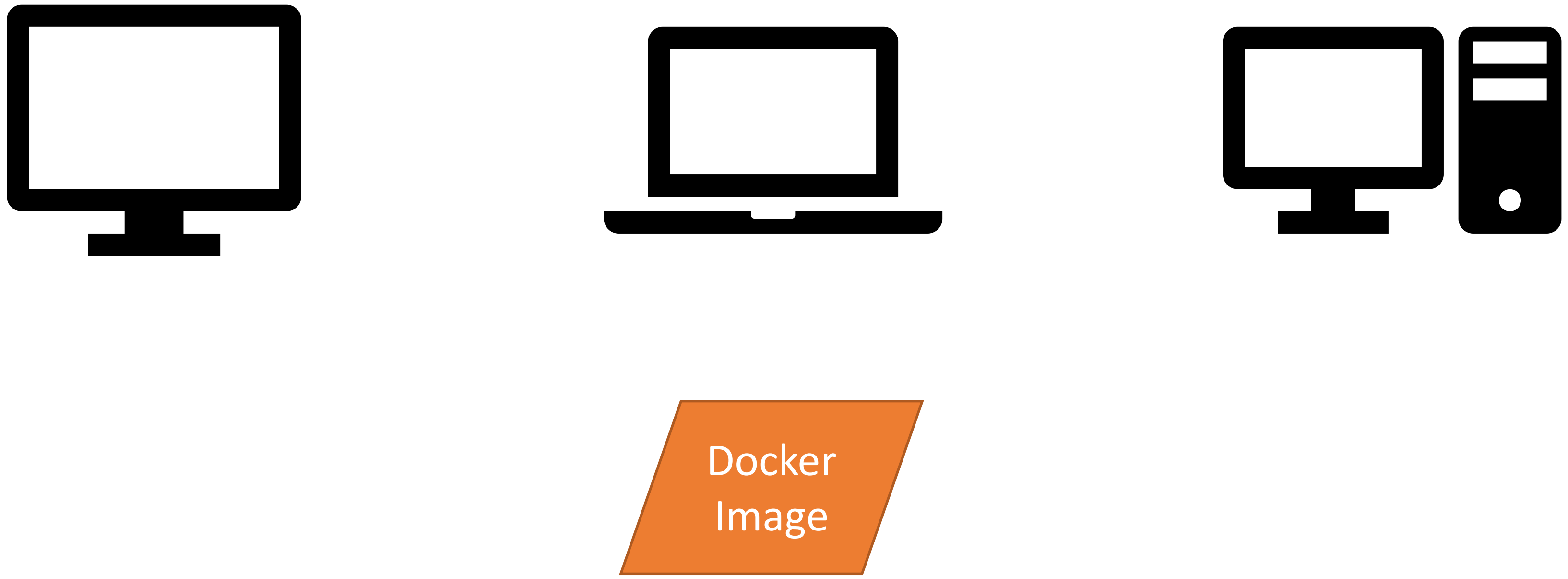
In ROB 102, we will use a program called **Docker** to run our code.

Docker is a program that lets us run **containers**. Containers are environments that imitate a computer.

Docker sets up a container based off an **image**, which is a snapshot of the state of a computer at a given point. Ours will use the **Linux** operating system.

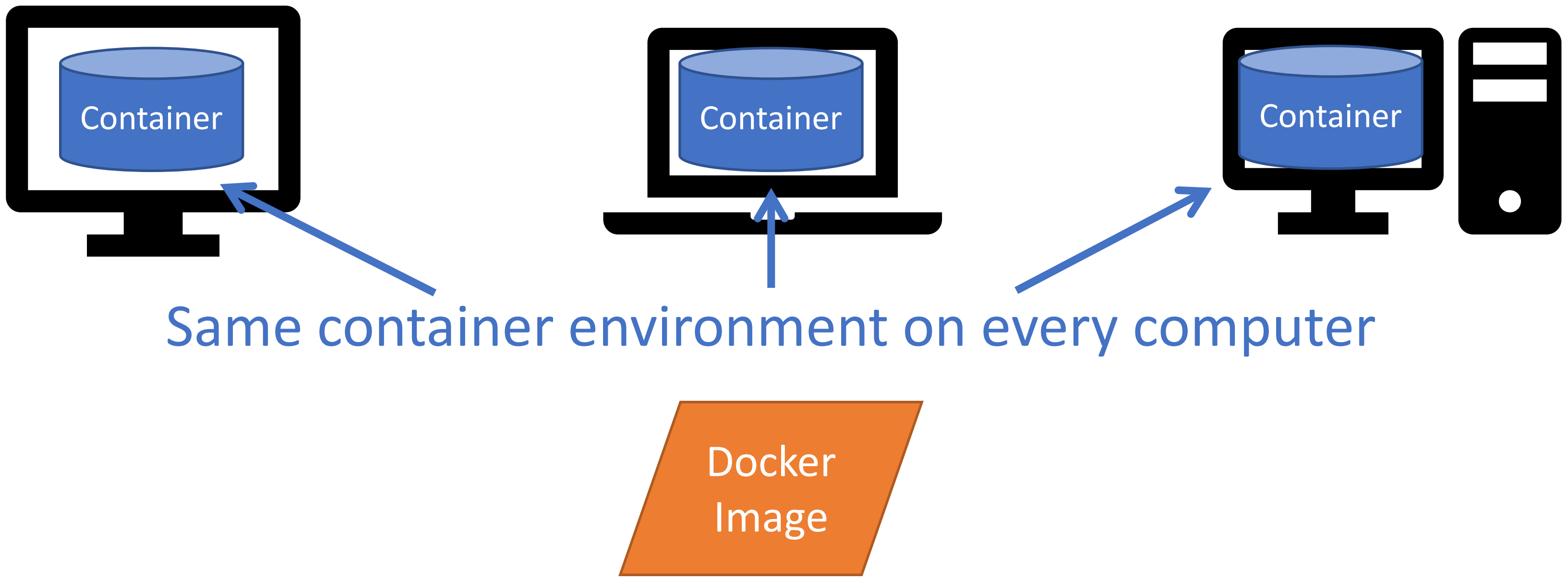
Running your code: Docker

We can use a Docker image to run containers that look exactly the same on every computer! It contains *all the dependencies* we need to get started running our code.



Running your code: Docker

We can use a Docker image to run containers that look exactly the same on every computer! It contains *all the dependencies* we need to get started running our code.



Docker: Your turn!

If you haven’t already, install Docker

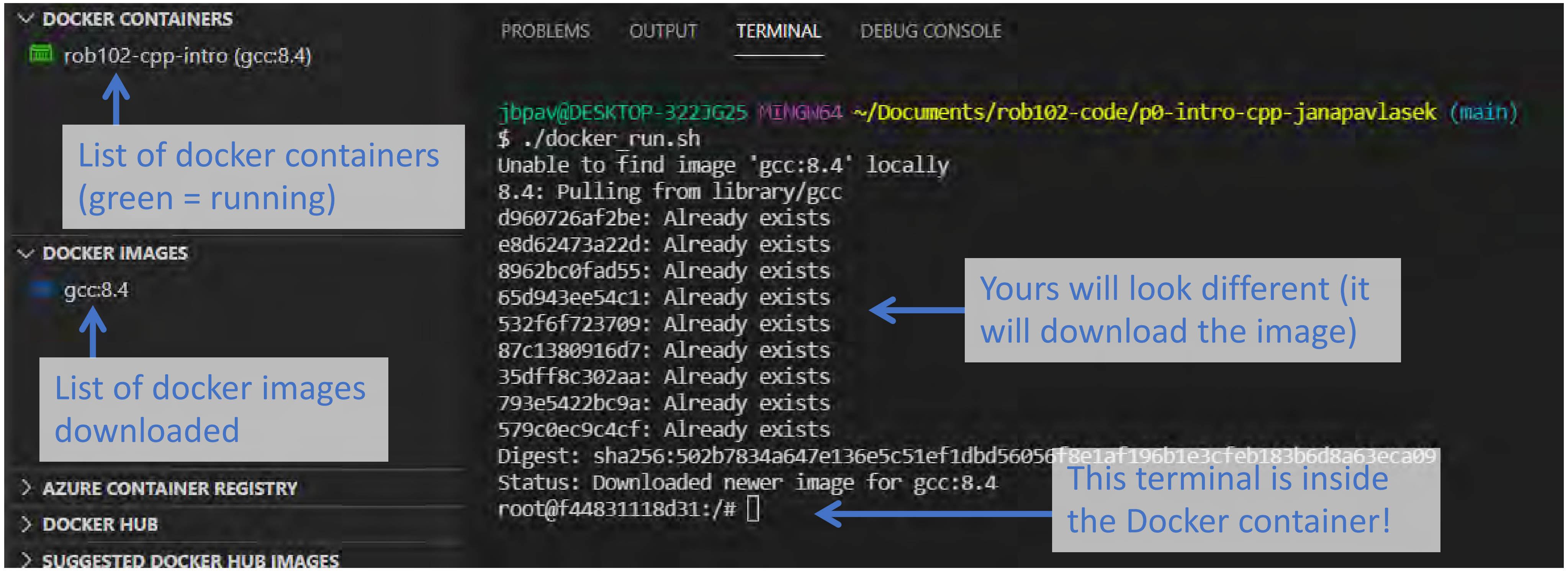
- https://robotics102.github.io/tutorials/setup

Also install the Docker and Docker Explorer VSCode extensions.

Docker: Your turn!

In your VSCode terminal, type:

```
./docker_run.sh
```



List of docker containers (green = running)

List of docker images downloaded

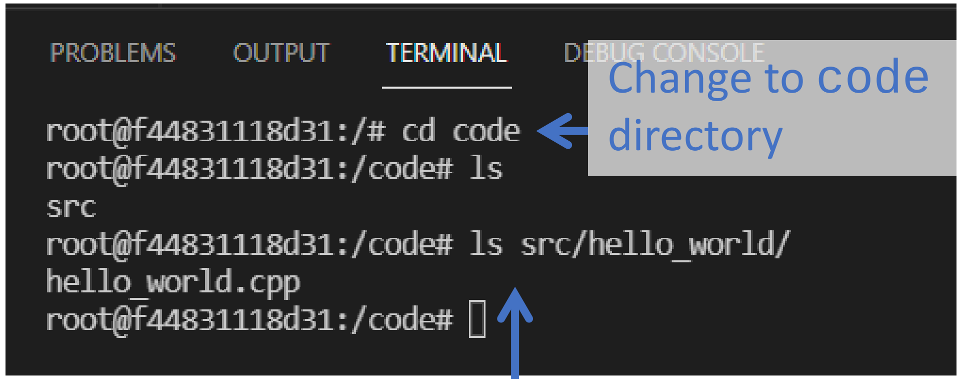
```
jbpav@DESKTOP-322JG25 MINGW64 ~/Documents/rob102-code/p0-intro-cpp-janapavlasek (main)
$ ./docker_run.sh
Unable to find image 'gcc:8.4' locally
8.4: Pulling from library/gcc
d960726af2be: Already exists
e8d62473a22d: Already exists
8962bc0fad55: Already exists
65d943ee54c1: Already exists
532f6f723709: Already exists
87c1380916d7: Already exists
35dff8c302aa: Already exists
793e5422bc9a: Already exists
579c0ec9c4cf: Already exists
Digest: sha256:502b7834a647e136e5c51ef1dbd56056f8e1af196b1e3cfeb183b6d8a63eca09
Status: Downloaded newer image for gcc:8.4
root@f44831118d31:/#
```

Yours will look different (it will download the image)

This terminal is inside the Docker container!

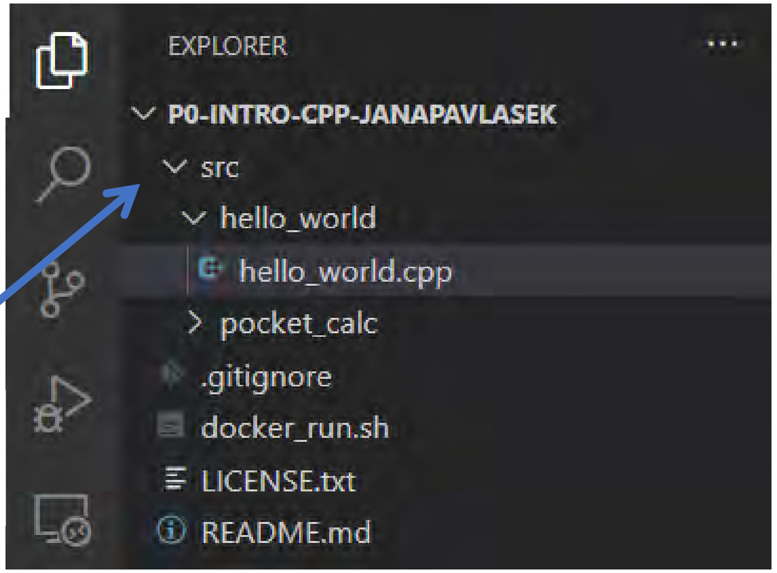
Docker: Exploring the container

The `src` folder in the repository is **mounted** inside the container.



```
root@f44831118d31:/# cd code
root@f44831118d31:/code# ls
src
root@f44831118d31:/code# ls src/hello_world/
hello_world.cpp
root@f44831118d31:/code# 
```

`/code/src/` mirrors `C:\Users\jana\p0-intro-cpp-janapavlasek/src`



You will write code in VSCode, and compile and run it in the container.

Docker: Running your code

Edit the file `src/hello_world/hello_world.cpp` to print the message “Hello World!”

`cd /code/src/hello_world`

$\downarrow$

In the Docker container, change directories to “`src/hello_world`”. Type the command:

```
g++ hello_world.cpp -o hello_world
```

$\uparrow$ The C++ compiler we will be using

$\uparrow$ The C++ source file to compile

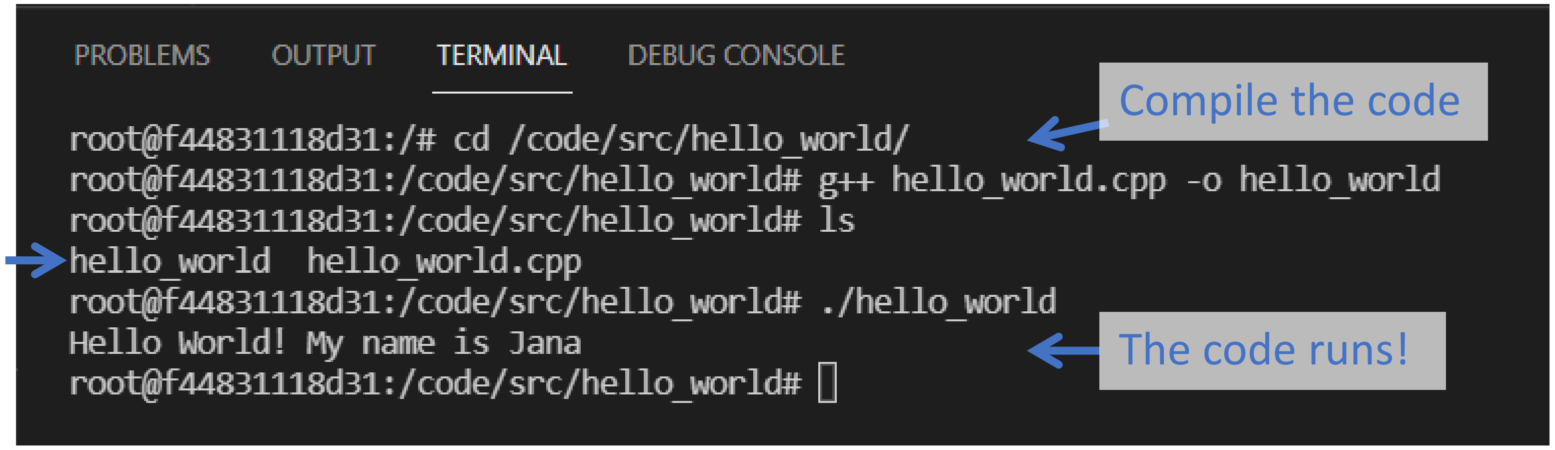
$\uparrow$ The name we want to give our executable

Docker: Running your code

In the Docker container (still in the `hello_world` directory), type the command:

```
./hello_world
```

Our executable is created by the compiler



```
root@f44831118d31:/# cd /code/src/hello_world/
root@f44831118d31:/code/src/hello_world# g++ hello_world.cpp -o hello_world
root@f44831118d31:/code/src/hello_world# ls
hello_world  hello_world.cpp
root@f44831118d31:/code/src/hello_world# ./hello_world
Hello World! My name is Jana
root@f44831118d31:/code/src/hello_world# 
```

Command line cheat sheet

`cd my_directory/`	Changes the working directory to `my_directory`. • If we write `/my_directory/` with a slash at the beginning, it is interpreted as an absolute path. Without the slash, the path is relative to the current directory. • Typing just `cd` with no argument brings you to the home directory.
`ls`	Lists all the files and folders in the current directory. Typing `ls my_dir/` lists the files in the directory `my_dir/`.
`pwd`	Prints the path of the current working directory.
`./my_exec`	Runs executable called “`my_exec`”.
`g++ my_code.cpp –o my_exec`	Compiles the code in the file `my_code.cpp` into an executable called `my_exec`.
`exit`	Closes the current terminal.

Git cheat sheet

`git clone ADDRESS`	Clone the repository located at ADDRESS to the current folder.
`git status`	Show the status of the Git repository (untracked files, modified files, current branch).
`git add FILE`	Add `FILE` for staging (next step: commit).
`git commit –m “My commit msg”`	Commit the files currently added for staging with the given commit message (next step: push).
`git push`	Push the commits created to the remote repository.
`git pull`	Pull any new commits in the remote repository into the local repository.
`git tag p0-v1`	Tag all commits currently on this branch with tag `p0-v1`.
`git push origin p0-v1`	Push tag `p0-v1` to the remote repository.

Tasks for Today

- ✓ Install all software
- ✓ Push license to GitHub repo
- ✓ Run Hello World on your computer
- ✓ Push the changes to your repository