



UTM
UNIVERSITI TEKNOLOGI MALAYSIA



MODULE 4 : WORKING WITH GIT & GITHUB (INDIVIDUAL REPOSITORY)

PREPARED BY:

Dr. Nurulaqilla Khamis

Center for Artificial Intelligence and Robotics
Universiti Teknologi Malaysia



QUICK RECAP!

Notes:

CLI much faster!



Programmer
(Web Apps, Desktop Apps,
Mobile Apps)



Command Line Interface (CLI)



To track changes, modify and
version control:

version1

version2

.

versionN

GitHub

Storage Space Your Project
Repo/Online Hosting

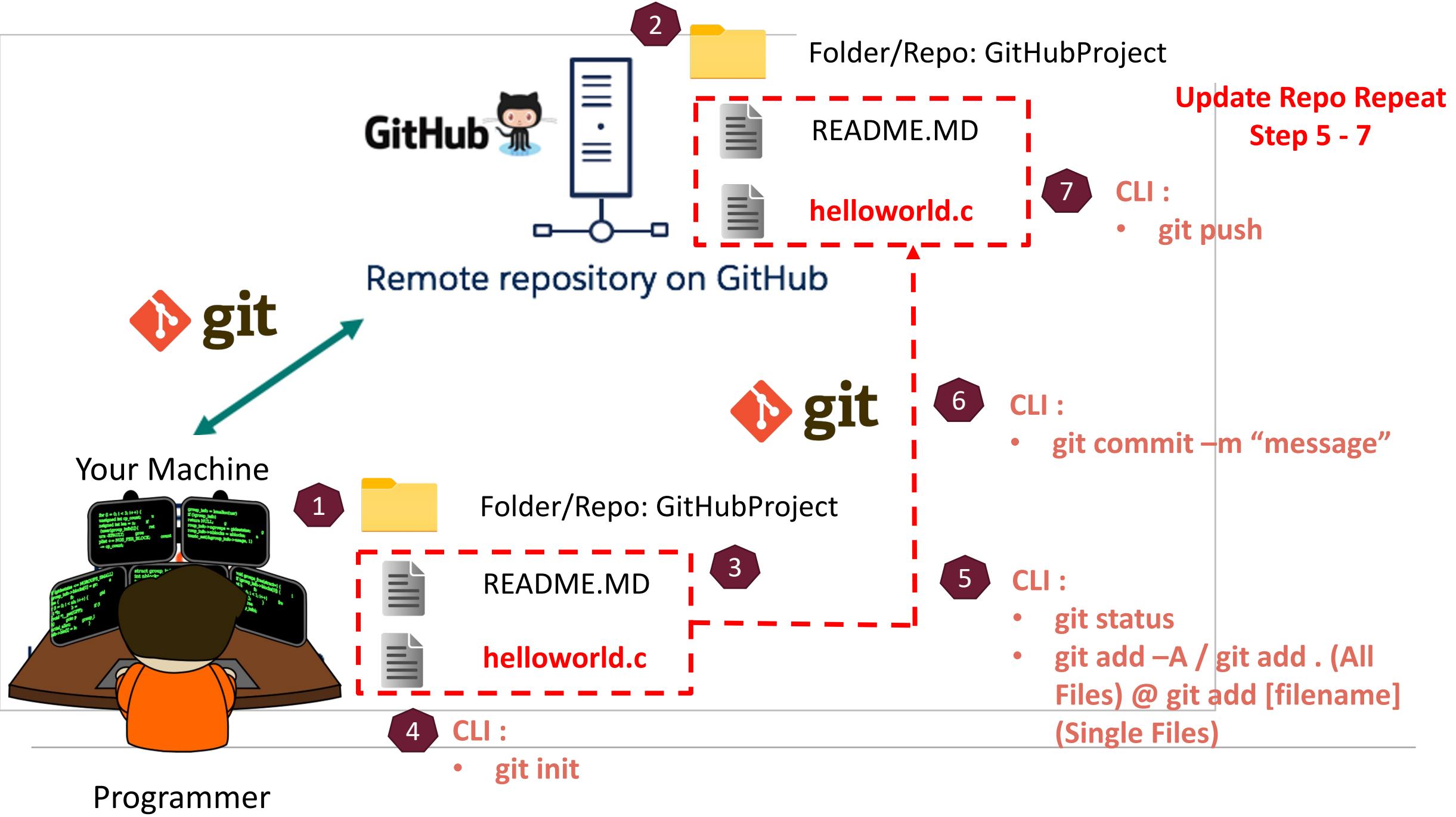
WORKING WITH GIT AND GITHUB (INDIVIDUAL REPOSITORY)



Note: We will **create new repo for this project** and use

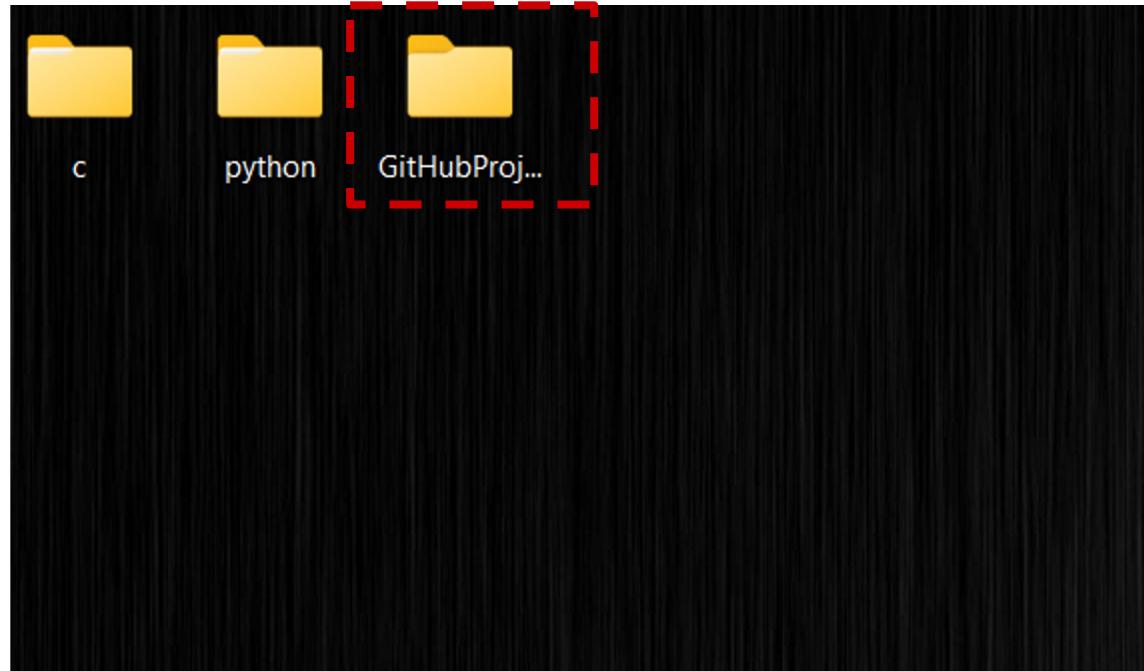
Atom (source code editor)

1. Create Folder/Repo Project in your Local Machine
2. Create New Repository (GitHub)
3. Create New File (README, project file)
4. Initialize Project in Git
5. Add File from Local Machine to GitHub Repository
6. Commit Files in Git
7. Push Files in GitHub
8. Update Any Changes (Files)



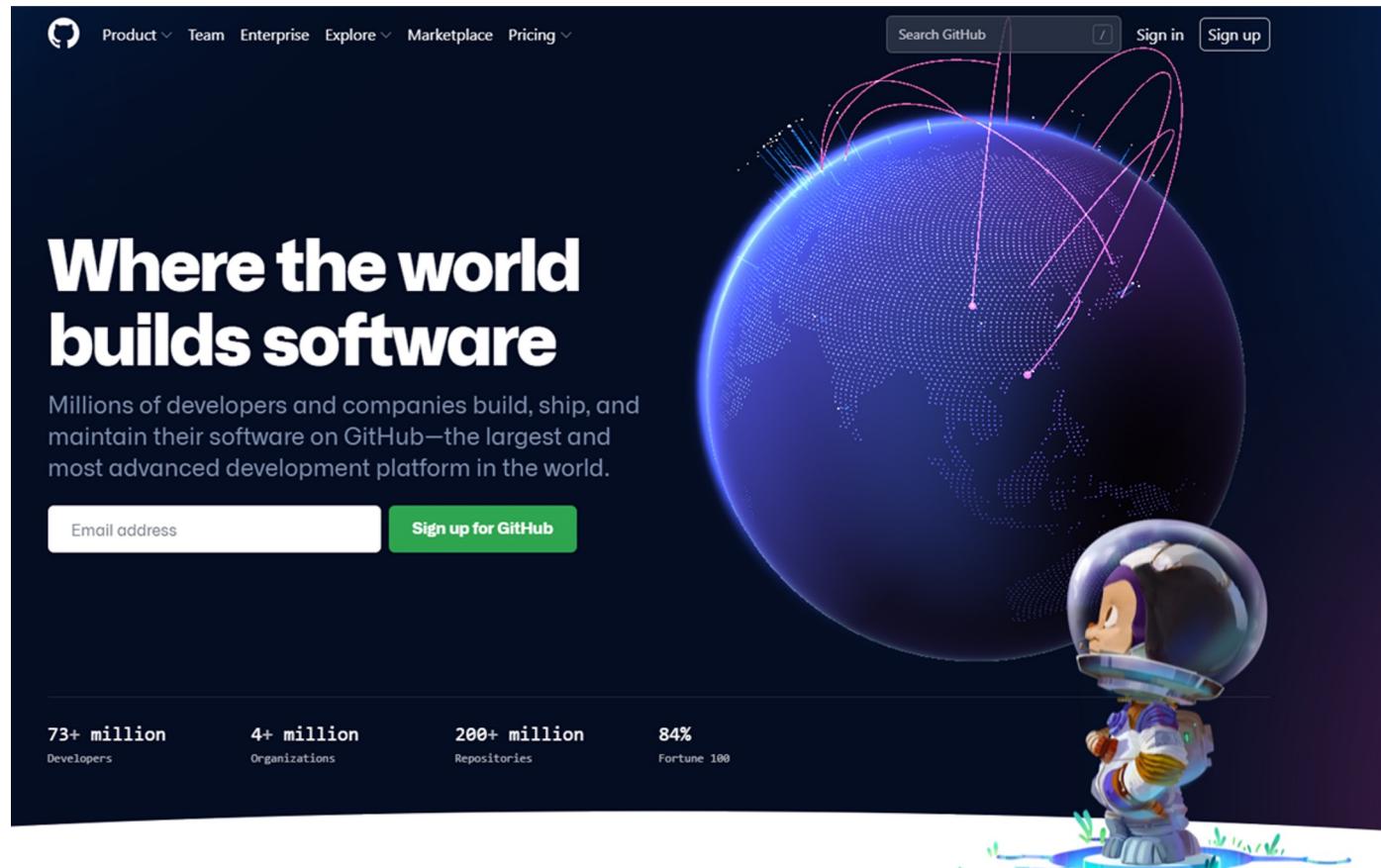
CREATE REPOSITORY PROJECT (LOCAL MACHINE)

1. Go to your “Desktop” > “Right Click” > “New Folder” > “GitHubProject”



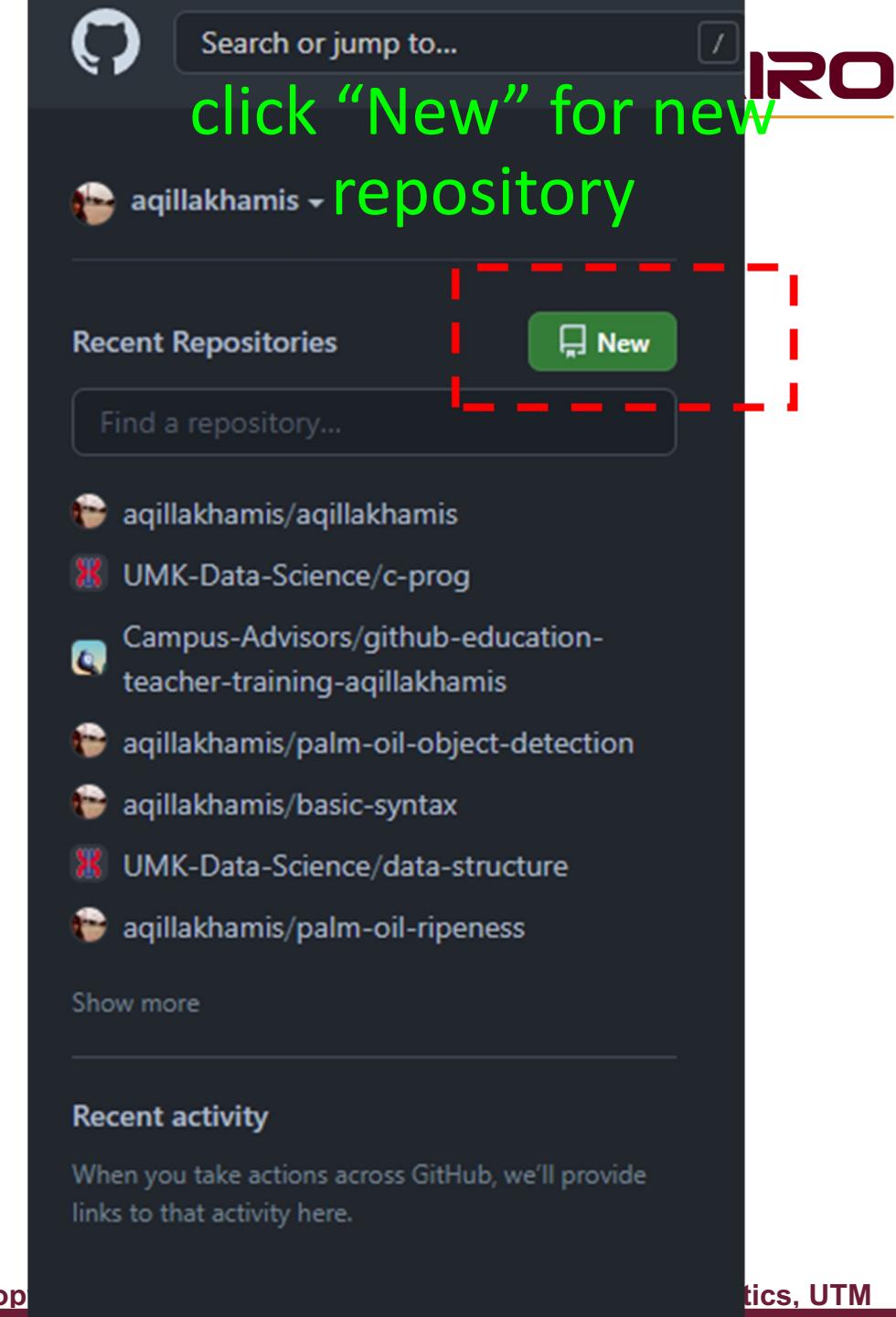
Your Desktop

CREATE NEW REPOSITORY (GITHUB)



The GitHub homepage features a large blue globe with a dotted map of the world. A cartoon astronaut in a white spacesuit with a blue helmet is standing on a small green patch of grass at the bottom. The background is dark blue. At the top, there's a navigation bar with links for Product, Team, Enterprise, Explore, Marketplace, and Pricing. Below the navigation is a search bar labeled "Search GitHub". To the right of the search bar are "Sign in" and "Sign up" buttons. The main headline reads "Where the world builds software". Below it, a sub-headline says "Millions of developers and companies build, ship, and maintain their software on GitHub—the largest and most advanced development platform in the world." There are two buttons: "Email address" and "Sign up for GitHub". At the bottom, there are four statistics: "73+ million Developers", "4+ million Organizations", "200+ million Repositories", and "84% Fortune 100".

www.github.com

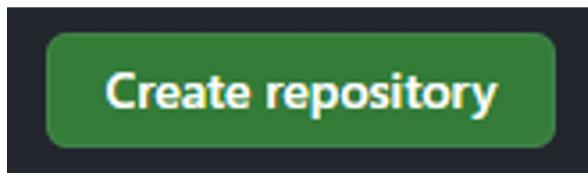


A screenshot of a GitHub user profile page for "aqillakhamis". The top navigation bar includes a search bar, a user icon, and a "New" button. The user's name "aqillakhamis" is displayed with a dropdown arrow. Below the header, a red dashed box highlights the "New" button. The main content area shows a list of "Recent Repositories" with the following items:

- aqillakhamis/aqillakhamis
- UMK-Data-Science/c-prog
- Campus-Advisors/github-education-teacher-training-aqillakhamis
- aqillakhamis/palm-oil-object-detection
- aqillakhamis/basic-syntax
- UMK-Data-Science/data-structure
- aqillakhamis/palm-oil-ripeness

Below the repository list is a "Show more" link. At the bottom, a section titled "Recent activity" is shown with the text: "When you take actions across GitHub, we'll provide links to that activity here."

1. Repository name: Create similar project name from your local machine repo - “GitHubProject”
2. Description – Optional
3. Public @ Private Repo? (Your Choice)
4. Click “Create repository”



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner *  aqillakhamis /

Great repository names are short and memorable. Need inspiration? How about [fuzzy-computing-machine?](#)

Description (optional)

 **Public**
Anyone on the internet can see this repository. You choose who can commit.

 **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

Add a README file
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template:

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License:

 You are creating a public repository in your personal account.

[Pin](#)[Unwatch](#)

Quick setup — if you've done this kind of thing before

[Set up in Desktop](#)

or

[HTTPS](#)[SSH](#)<https://github.com/aqillakhamis/GitHubProject.git>

Get started by creating a new file or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# GitHubProject" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/aqillakhamis/GitHubProject.git
git push -u origin main
```



...or push an existing repository from the command line

```
git remote add origin https://github.com/aqillakhamis/GitHubProject.git
git branch -M main
git push -u origin main
```



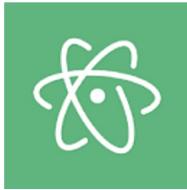
...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

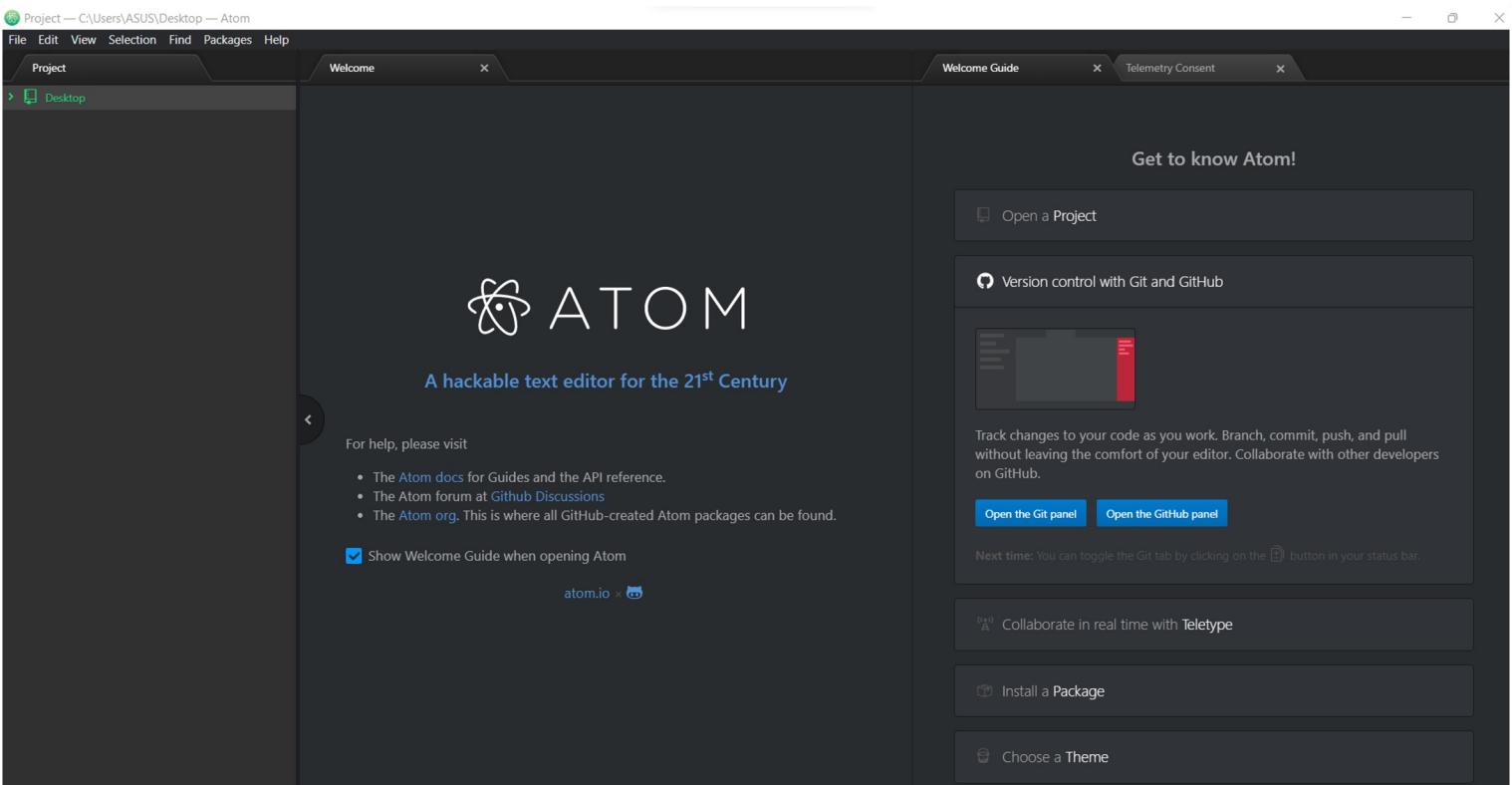
[Import code](#)

CREATE NEW FILE (“README.MD”)

1. Click “Atom” (source-code editor)
2. Click File > New File



Atom - free open source code editor support any programming languages 😊



```
1  
#include<stdio.h>  
  
int main(){  
    printf("Hello World!");  
}
```

Note: You will get an empty canvas.

We will write our code in here! 🎨

Instruction:

1. After complete write the code, save the file by click **File > Save As (Ctrl + s)**
2. Save this file as “**helloworld.c**” inside your “GitHubProject” repo/folder

[Pin](#)[Unwatch](#)

Quick setup — if you've done this kind of thing before

[Set up in Desktop](#)

or

[HTTPS](#)[SSH](#)<https://github.com/aqillakhamis/GitHubProject.git>

Get started by creating a new file or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# GitHubProject" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/aqillakhamis/GitHubProject.git
git push -u origin main
```



...or push an existing repository from the command line

```
git remote add origin https://github.com/aqillakhamis/GitHubProject.git
git branch -M main
git push -u origin main
```



...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

INITIALIZE PROJECT IN GIT



1. Open “Git Bash” (type at your search bar)
2. Change the path file that we want to access. Point it to your working repo.
3. Type “cd desktop/your_folder_name”

```
MINGW64:/c/Users/ASUS/Desktop/GitHubProject
ASUS@NKhamis-PC MINGW64 ~
$ cd desktop/GitHubProject
ASUS@NKhamis-PC MINGW64 ~/Desktop/GitHubProject (master)
$ |
```

Now it will change to
your repo path 😮



INITIALIZE PROJECT IN GIT



1. Initialize “git init” - this will tell Git to get ready watching your files for every changes that occurs
2. Git init - information about your local machine & path where the folder exists

```
MINGW64:/c/Users/ASUS/Desktop/GitHubProject
ASUS@NKhamis-PC MINGW64 ~/Desktop/GitHubProject (master)
$ git init
Initialized empty Git repository in c:/Users/ASUS/Desktop/GitHubProject/.git/
ASUS@NKhamis-PC MINGW64 ~/Desktop/GitHubProject (master)
$
```

A red arrow points from the word "exists" in the second list item down to the final line of the terminal output, which shows the creation of a new Git repository in the specified directory.

ADD FILE FROM LOCAL MACHINE TO GITHUB REPOSITORY



1. **git add README.md @ git add helloworld.c** (if you want to add specific single file)
2. **git add -A @ git add .** (the period or dot that comes after add means all the files that exist in the repository)

```
ASUS@NKhamis-PC MINGW64 ~/desktop/GitHubProject (master)
$ git add -A
```

COMMIT FILES IN GIT

1. To commit our file, we use **git commit -m “write your message”**

```
ASUS@NKhamis-PC MINGW64 ~/desktop/GitHubProject (master)
$ git commit -m "first update"
[master (root-commit) a4125a0] first update
 2 files changed, 8 insertions(+)
 create mode 100644 README.MD
 create mode 100644 helloworld.c
```

PUSH FILES IN GITHUB



1. **git remote add origin [repo URL name]** - create connection between your local repo and central repo on GitHub
2. **git branch -M main**
3. **git push -u origin main** : push your repo from local to GitHub

```
ASUS@NKhamis-PC MINGW64 ~/desktop/GitHubProject (main)
$ git remote add origin https://github.com/aqillakhamis/GitHubProject.git

ASUS@NKhamis-PC MINGW64 ~/desktop/GitHubProject (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 380 bytes | 380.00 kib/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/aqillakhamis/GitHubProject.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

PUSH FILES IN GITHUB

A screenshot of a GitHub repository page for "aqillakhamis / GitHubProject". The repository is public, has 1 branch, and 0 tags. The main branch is selected. The commit history shows three commits from "aqillakhamis" made 19 seconds ago. The first commit pushed "README.MD" and "helloworld.c".

aqillakhamis / GitHubProject Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

aqillakhamis first update f18f413 19 seconds ago 3 commits

README.MD first update 1 hour ago

helloworld.c first update 19 seconds ago

README.MD

Tadaa! 😊

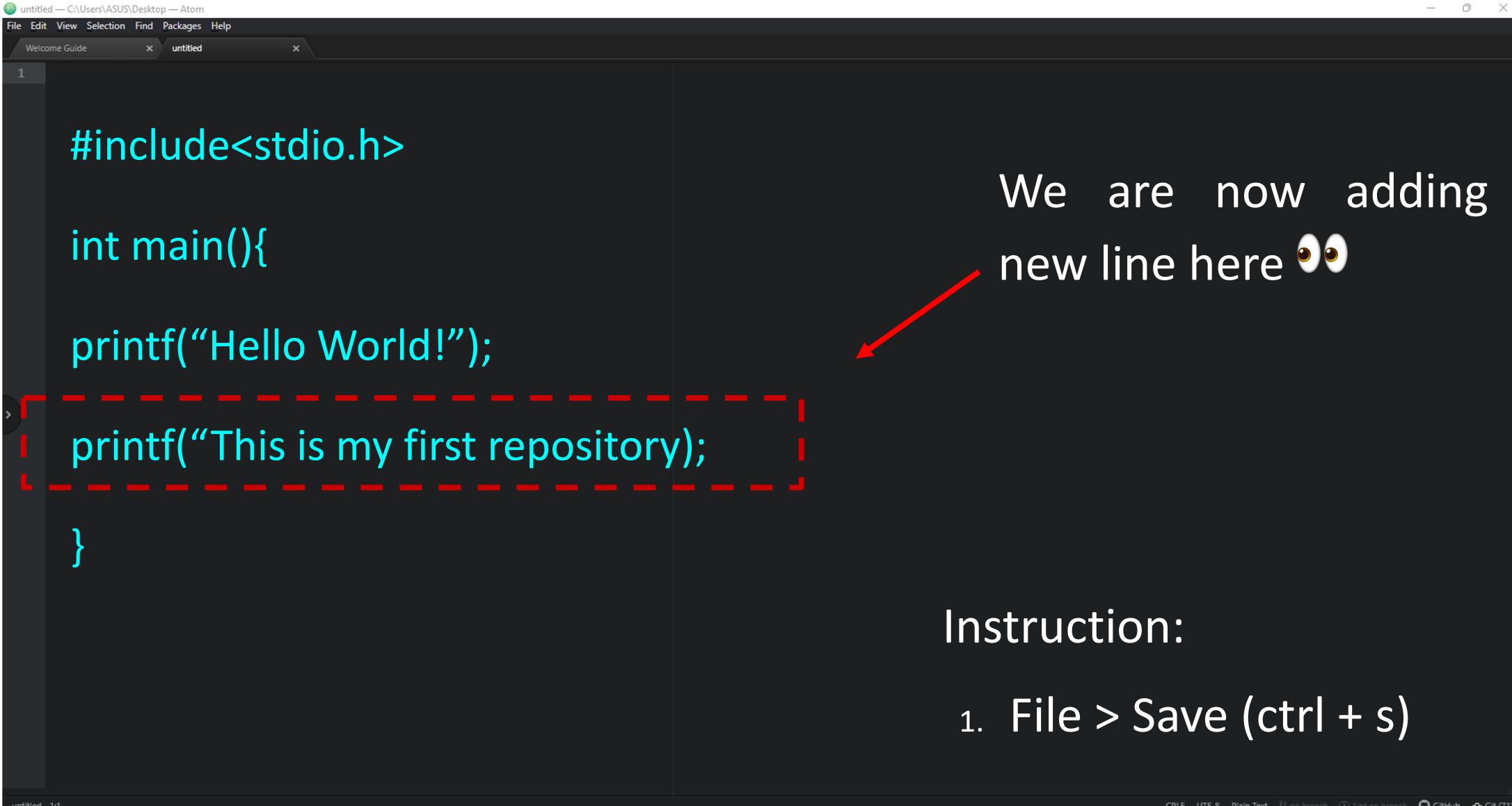
GitHub Project

This is my first tutorial creating repository on GitHub.

You can see your files from local repo now inside GitHub repository

Copyright for Center for Artificial Intelligence and Robotics, UTM

UPDATE ANY CHANGES (FILES)



A screenshot of the Atom code editor showing a C program. The code includes a standard Hello World print statement and a new printf statement that has been added:

```
#include<stdio.h>
int main(){
    printf("Hello World!");
    printf("This is my first repository");
}
```

A red dashed rectangular box highlights the second printf statement. A red arrow points from the text "new line here" to the line separator between the two printf statements.

We are now adding
new line here 😊

Instruction:

1. File > Save (ctrl + s)

LET'S SEE WHAT GIT HAS FOUND? 😊



- **git status** - every changes you made can be trace by type “**git status**”

MINGW64:/c/Users/ASUS/Desktop/GitHubProject

```
ASUS@NKhamis-PC MINGW64 ~/Desktop/GitHubProject (main)
$ git status
on branch main
Your branch is up to date with 'origin/main'.
```

Changes not staged for commit:

```
(use "git add <file>..." to update what will be committed)
(use "git restore <file>..." to discard changes in working directory)
  modified:   helloworld.c
```

```
no changes added to commit (use "git add" and/or "git commit -a")
```

Gotcha! got modification in our file, and git can trace 😎

I WANT TO UPDATE CHANGES TO MY CENTRAL REPO (GITHUB). HOW CAN I DO THAT? 😢



```
ASUS@NKhamis-PC MINGW64 ~/desktop/GitHubProject (main)
$ git add helloworld.c

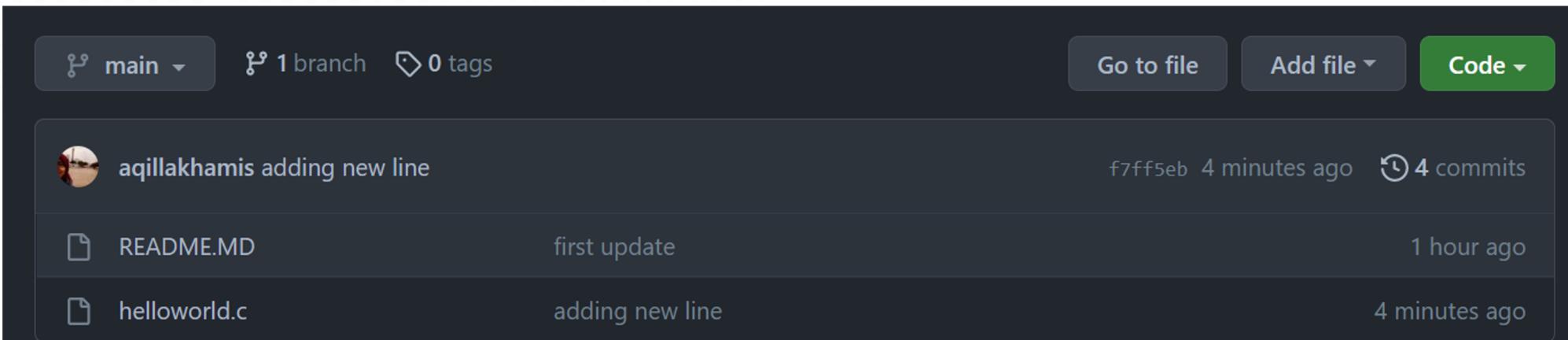
ASUS@NKhamis-PC MINGW64 ~/desktop/GitHubProject (main)
$ git commit -m "adding new line"
[main f7ff5eb] adding new line
 1 file changed, 1 insertion(+)

ASUS@NKhamis-PC MINGW64 ~/desktop/GitHubProject (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 363 bytes | 363.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/aqillakhamis/GitHubProject.git
 f18f413..f7ff5eb main -> main

ASUS@NKhamis-PC MINGW64 ~/desktop/GitHubProject (main)
$
```

1. Specify which file to update? **git add filename**
2. Put message before push, **git commit -m “message”**
3. Ready to push to your GitHub 😊! **git push**

Let's Check!

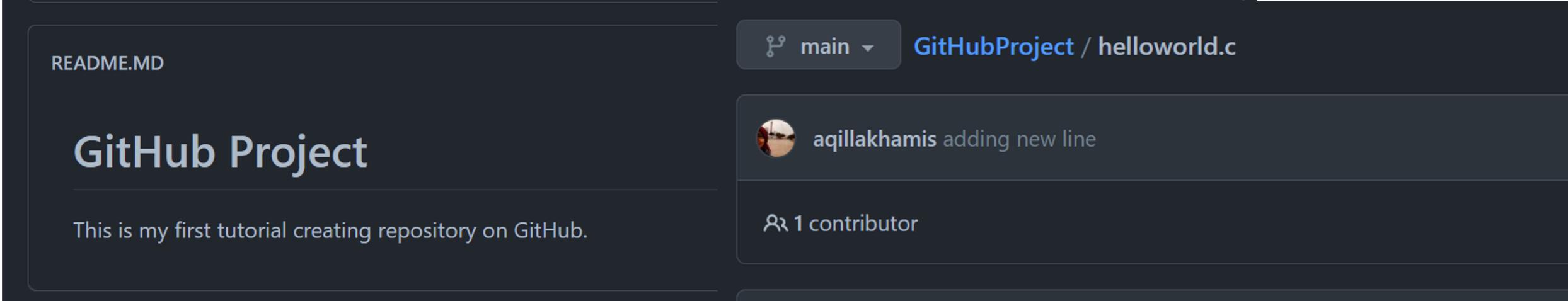


aqillakhamis adding new line

f7ff5eb 4 minutes ago 4 commits

README.MD first update 1 hour ago

helloworld.c adding new line 4 minutes ago



main ▾ GitHubProject / helloworld.c

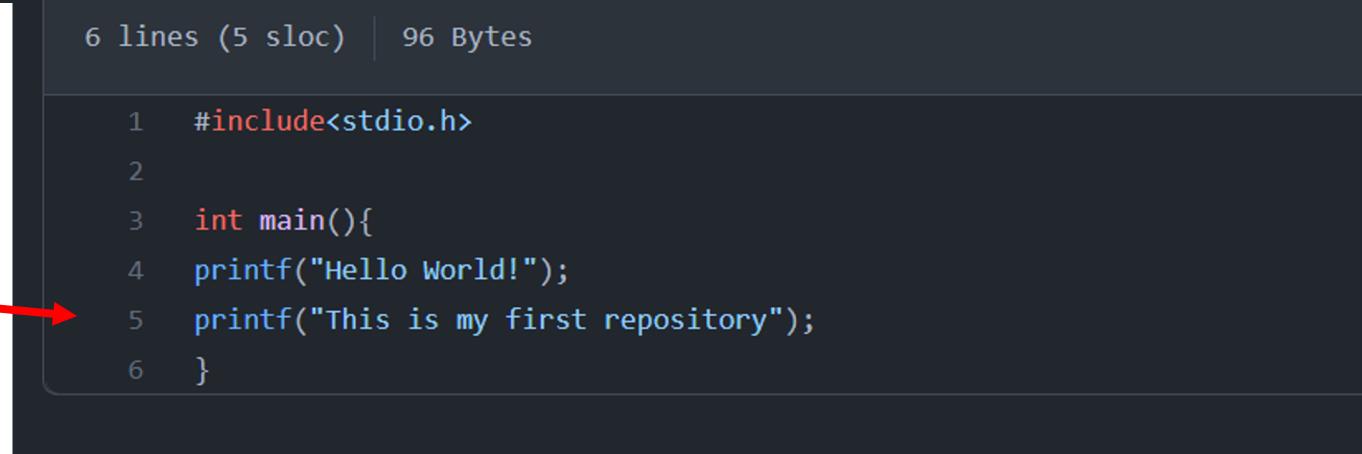
aqillakhamis adding new line

1 contributor

GitHub Project

This is my first tutorial creating repository on GitHub.

1. Refresh your GitHub repo
2. Click “helloworld.c” file
3. Everything update!



6 lines (5 sloc) | 96 Bytes

```
1 #include<stdio.h>
2
3 int main(){
4     printf("Hello World!");
5     printf("This is my first repository");
6 }
```

HANDS-ON INDIVIDUAL REPOSITORY



HANDS-ON (1)



1. Create Folder/Repo Project in your Local Machine
2. Create New Repository (GitHub)
3. Create New File (README, project file)
4. Initialize Project in Git
5. Add File from Local Machine to GitHub Repository
6. Commit Files in Git
7. Push Files in GitHub
8. Update Any Changes (Files)

TAKE-HOME-MESSAGE



1. Create Folder/Repo Project in your Local Machine ✓
2. Create New Repository (GitHub) ✓
3. Create New File (README, project file) ✓
4. Initialize Project in Git ✓
5. Add File from Local Machine to GitHub Repository ✓
6. Commit Files in Git ✓
7. Push Files in GitHub ✓
8. Update Any Changes (Files) ✓

**Practice
Makes
Perfect!**





THANK YOU



Centre for Artificial Intelligence and Robotics



research.utm.my/cairo/



cairo@utm.my / trainer's email