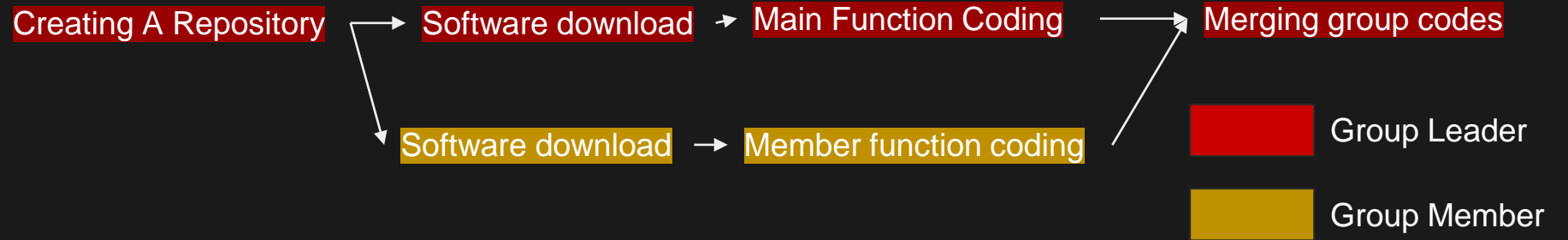


ENG1003

Tutorial 1 Walkthrough

Tutorial 1 Workflow



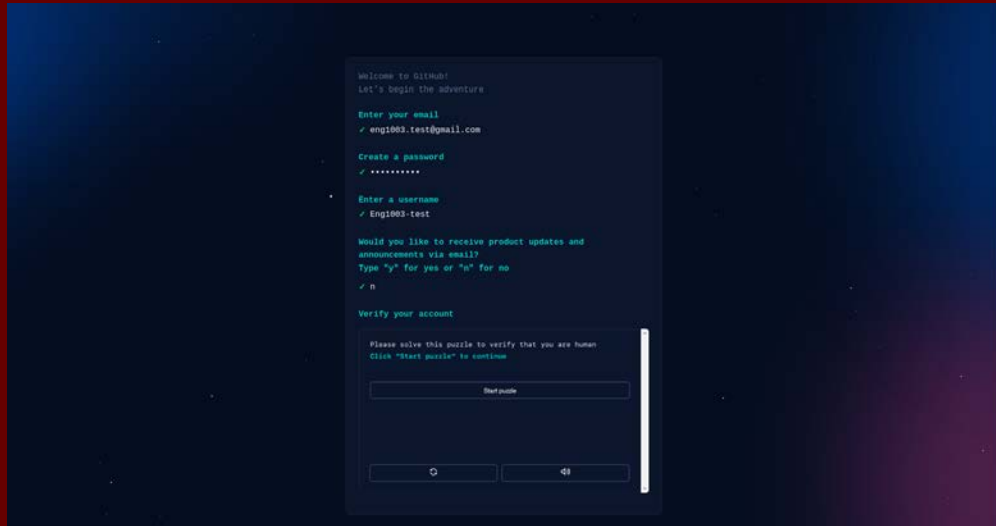
1. Initialize your GitHub project repository
2. Coding of the programming task
3. Combining the codes into one programme

- Group Leaders: Follow the red slides
- Group Members: Follow the yellow slides
- Everyone: Follow the blue slides

Initializing the GitHub Repository

1. Create your own GitHub account
2. Create a Repository for Tutorial 1
3. Invite everyone to the repository
4. Create a .py file for the repository

Create your own GitHub account



The screenshot shows the GitHub account creation interface. It starts with a welcome message and a series of input fields for email, password, and username. Below these is a checkbox for email notifications. The final step is a CAPTCHA verification where the user is asked to solve a puzzle by clicking a button. At the bottom, there are links for 'Log in' and 'Forgot your password?'. The interface is dark-themed with a blue gradient background.

Welcome to GitHub!
Let's begin the adventure

Enter your email
✓ eng1003.test@gmail.com

Create a password
✓

Enter a username
✓ Eng1003-test

Would you like to receive product updates and announcements via email?
Type "y" for yes or "n" for no
✓ n

Verify your account

Please solve this puzzle to verify that you are human.
Click "Start puzzle" to continue

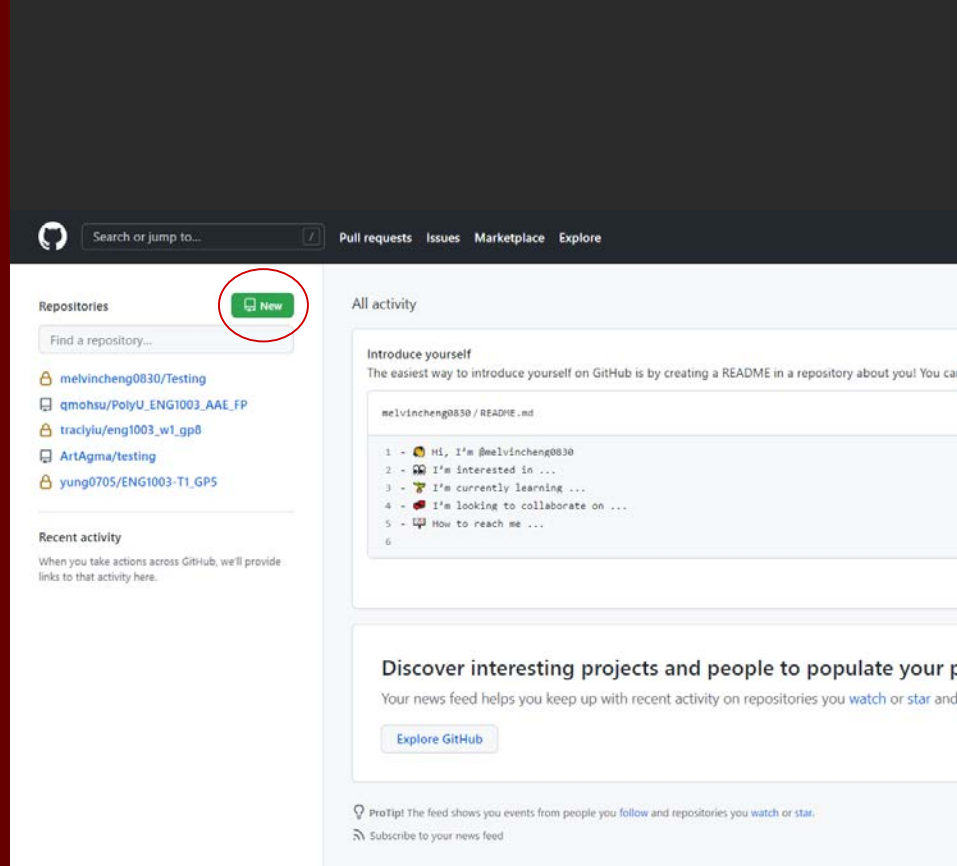
Start puzzle

Log in Forgot your password?

- Search 'GitHub on Google
- Create your own personal GitHub account
- Receive an email confirmation
- Activate your account using the email received

Create a Repository for Tutorial 1

- Click the new repository button once your logged into GitHub




Create a Repository for Tutorial 1

- Create a new repository
- Set it public
- Give the repository a correct name

Create a new repository


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


Owner * Repository name *

 melvincheng0830

Great repository names are short and unique. [ENG1003_w1_11 is available.](#) How about [miniature-bassoon](#)?

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.](#)

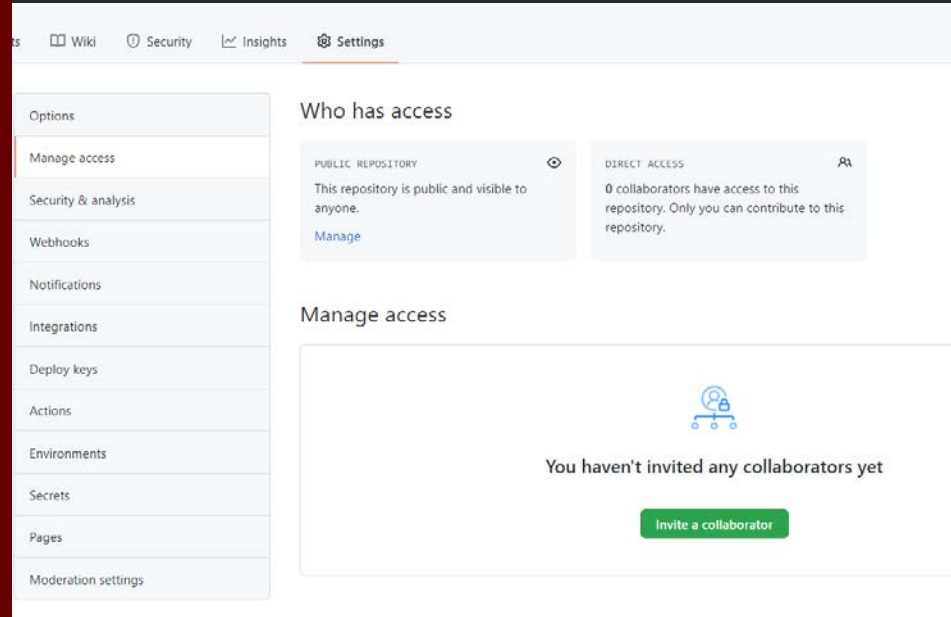
☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

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

[Create repository](#)

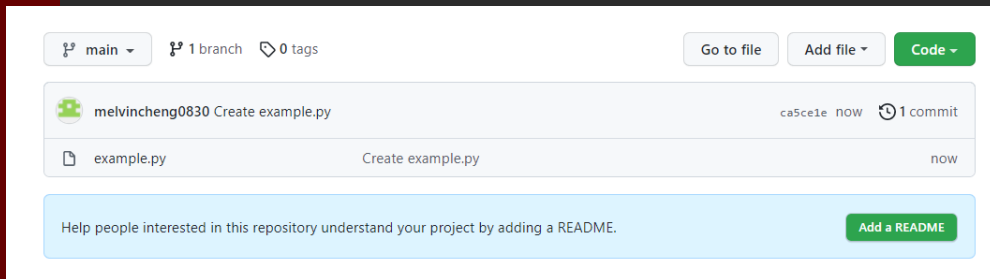
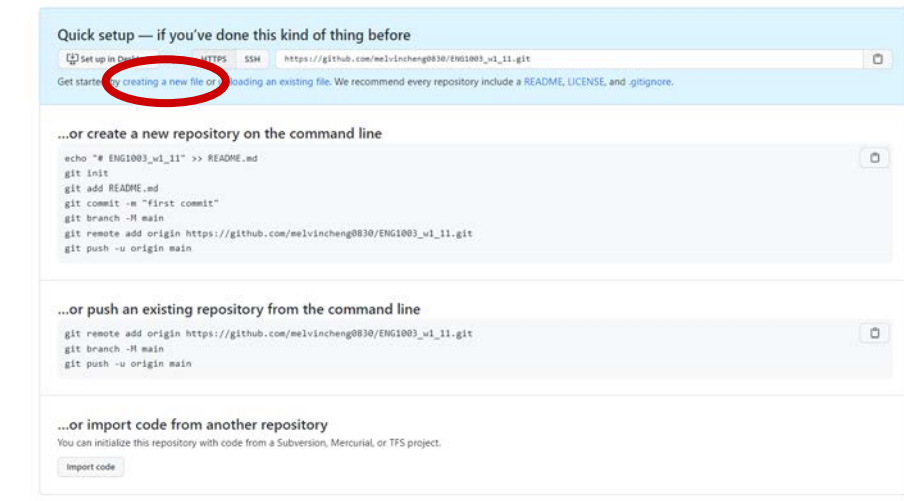
Invite everyone to your repository

- In settings -> manage access
- Use 'invite a collaborator to invite a new account to the repository
- Invite 4 group members and the assessor (Lecturer)
- You can find your corresponding assessor in the ENG1003's GitHub page



Create a .py file for the repository

- Click 'create a new file'
- Name the file with .py as ending

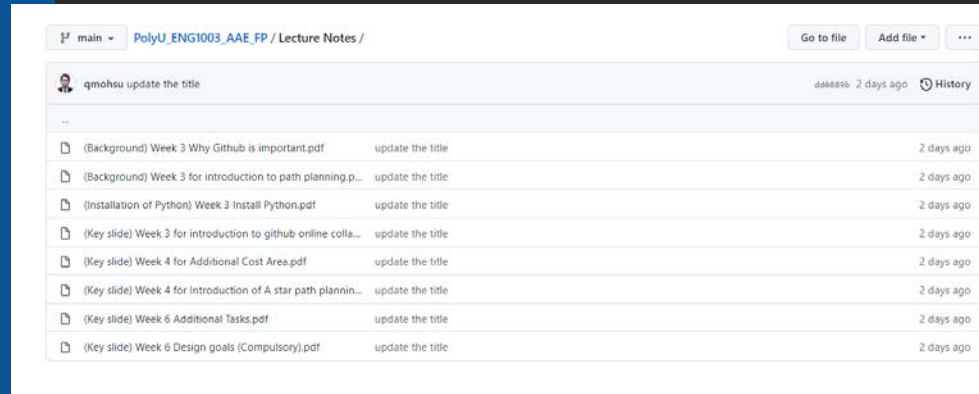


Coding of the programming Task

1. Software download
2. Create your own branch
3. Link GitHub with your local repository
4. Start Coding:
 - a. For group leaders
 - b. For group members
5. Create pull request

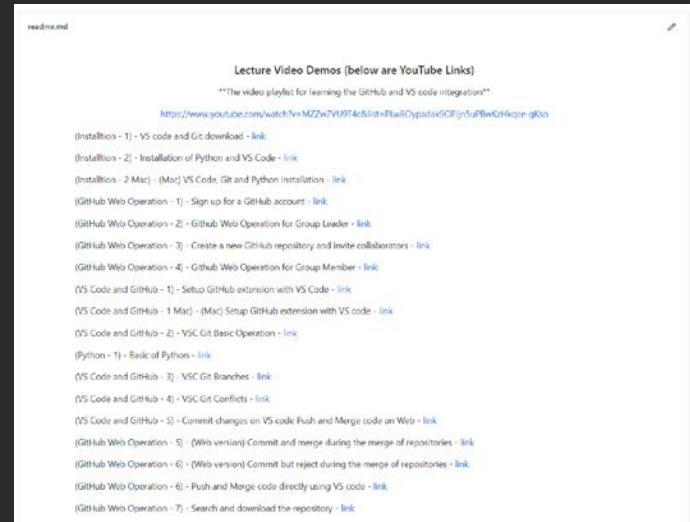
Software Download

- You have to first download the following Softwares:
 - Python
 - VS Code
 - Git
- You can find all installation tutorial under 'Lecture Notes' and 'Lecture Videos'



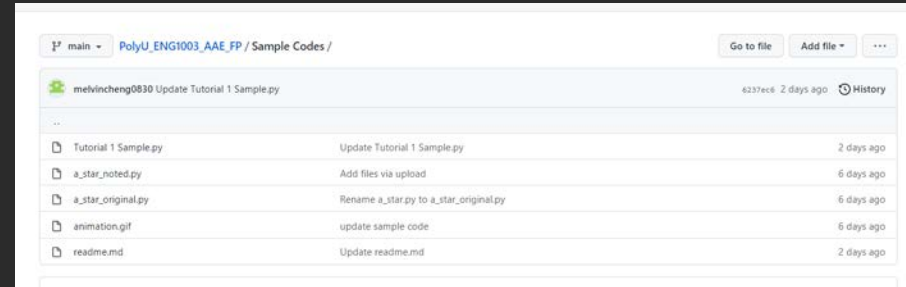
The screenshot shows a GitHub repository interface. At the top, the repository name is 'PolyU_ENGI003_AAE_FP / Lecture Notes'. Below this, there's a commit history table. The table has columns for the commit message, the user who made the commit, and the time since the commit. The commit messages are all 'update the title', and the user is 'qmohsu'. The times range from '2 days ago' to '2 days ago'.

Commit Message	User	Time
qmohsu update the title	qmohsu	2 days ago
(Background) Week 3 Why Github is important.pdf	qmohsu	2 days ago
(Background) Week 3 for introduction to path planning.p...	qmohsu	2 days ago
(Installation of Python) Week 3 Install Python.pdf	qmohsu	2 days ago
(Key slide) Week 3 for introduction to github online colla...	qmohsu	2 days ago
(Key slide) Week 4 for Additional Cost Area.pdf	qmohsu	2 days ago
(Key slide) Week 4 for introduction of A star path plannin...	qmohsu	2 days ago
(Key slide) Week 6 Additional Tasks.pdf	qmohsu	2 days ago
(Key slide) Week 6 Design goals (Compulsory).pdf	qmohsu	2 days ago



Coding

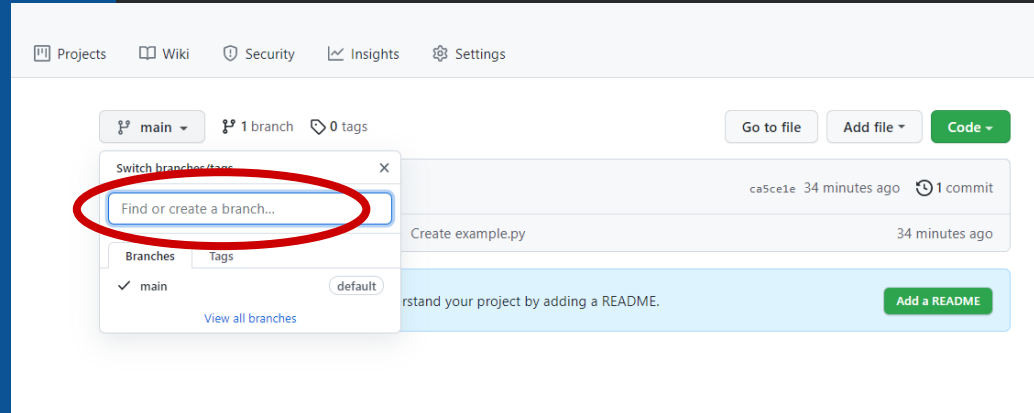
- Group Leaders:
 - Make a main function that can run 4 different functions made by members
- Group members:
 - Choose and make **ONE** function for the group leader
 - There are 4 members so there should be 4 functions made



- You can find a sample of the code 'Tutorial 1 Sample.py' under 'Sample Code'

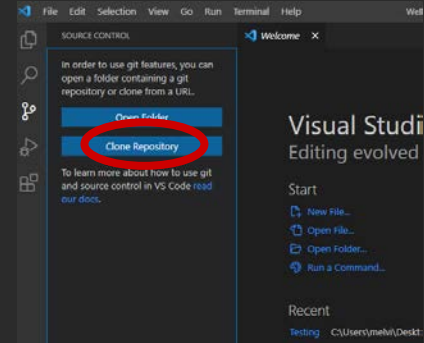
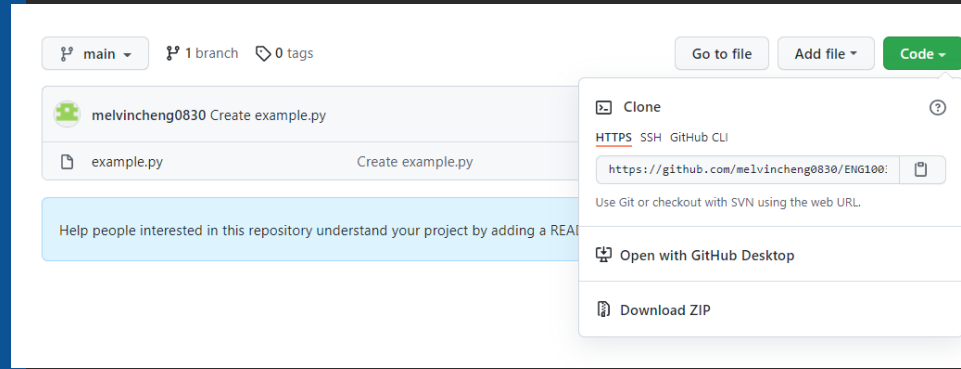
Create your own branch

1. Go back to GitHub
2. Create your own branch



Link GitHub with your local repository

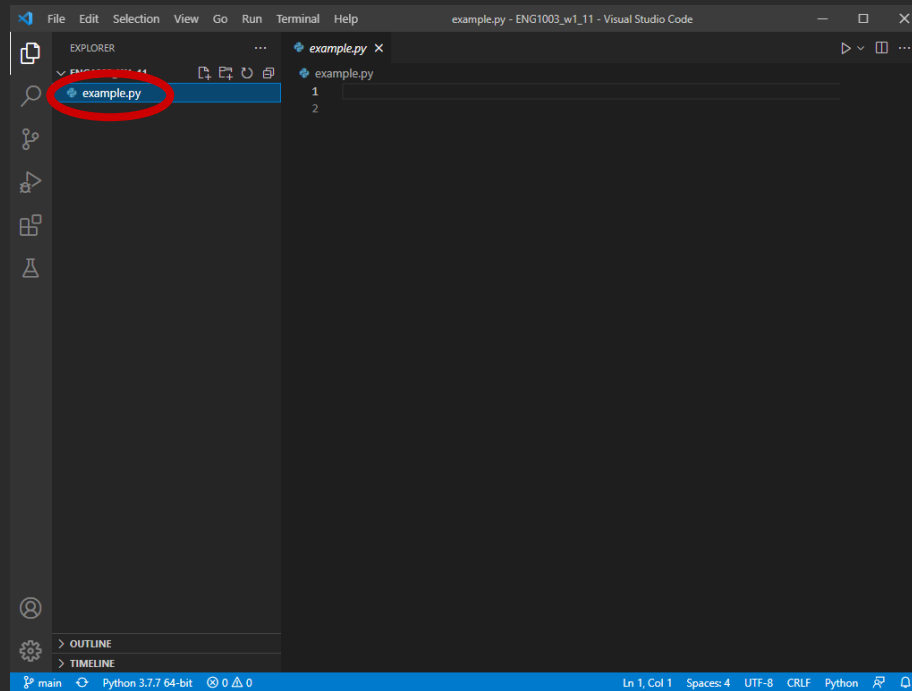
1. Copy the repository link from GitHub
2. Clone your repository to VS Code using Git Clone



- You can find additional tutorial under 'Lecture Notes' and 'Lecture Videos'

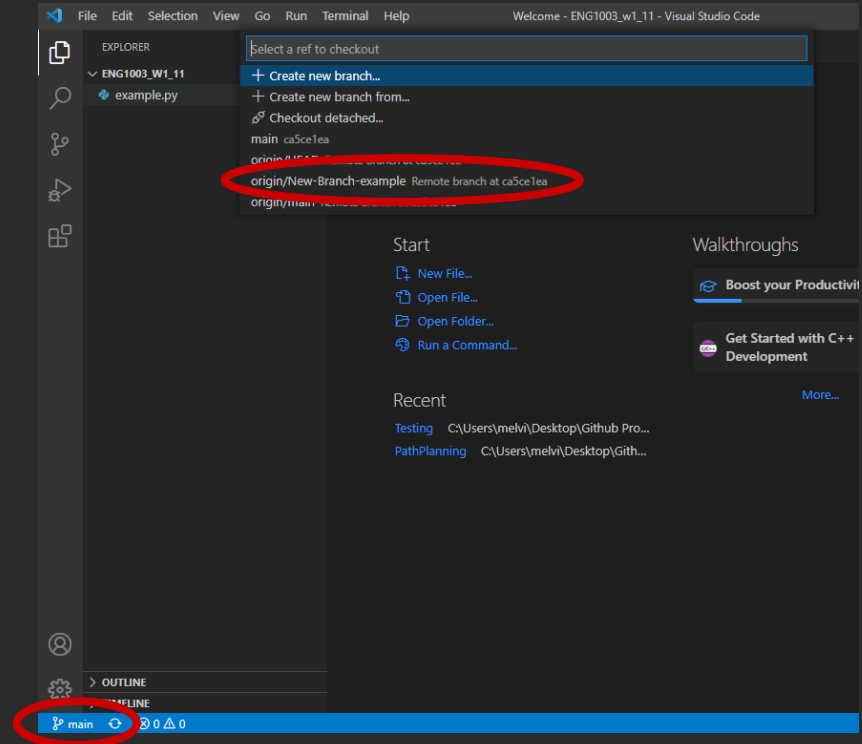
Link GitHub with your local repository

1. Cloning is successful if:
 - a. You can see the file you created in GitHub
 - b. You see you are in the main branch
 - c. You see the repository name is the same as the one in GitHub



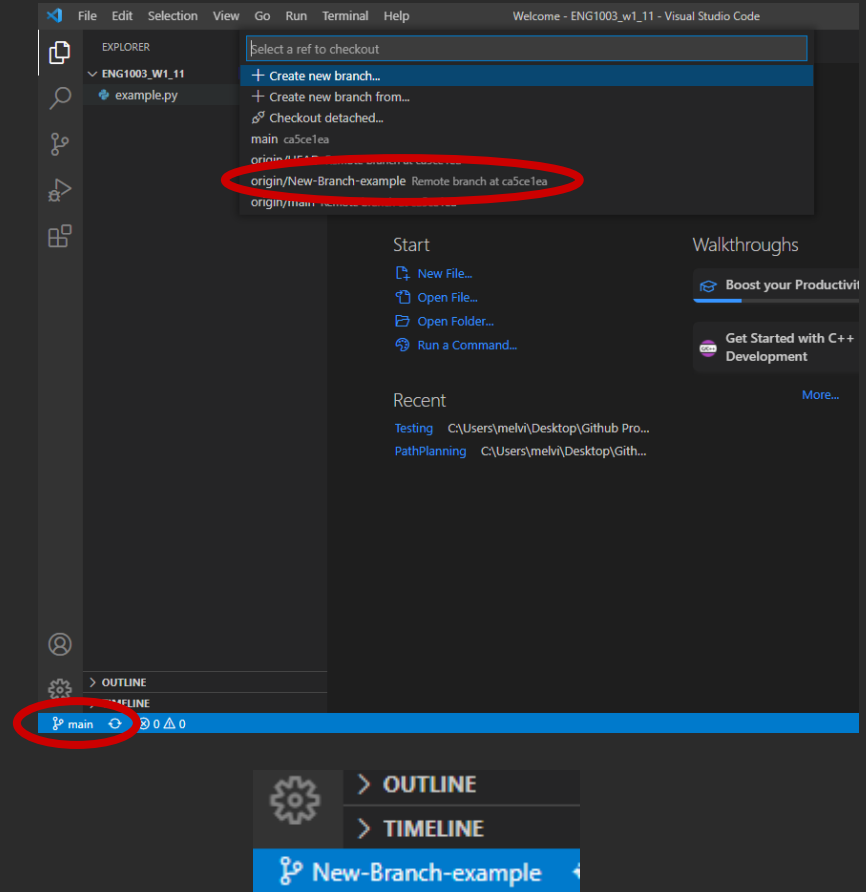
Switch to your own branch

1. Click the 'main' icon on the bottom left corner
2. Select the branch you created for yourself
3. You will see the 'main' icon replaced by the name of your own branch



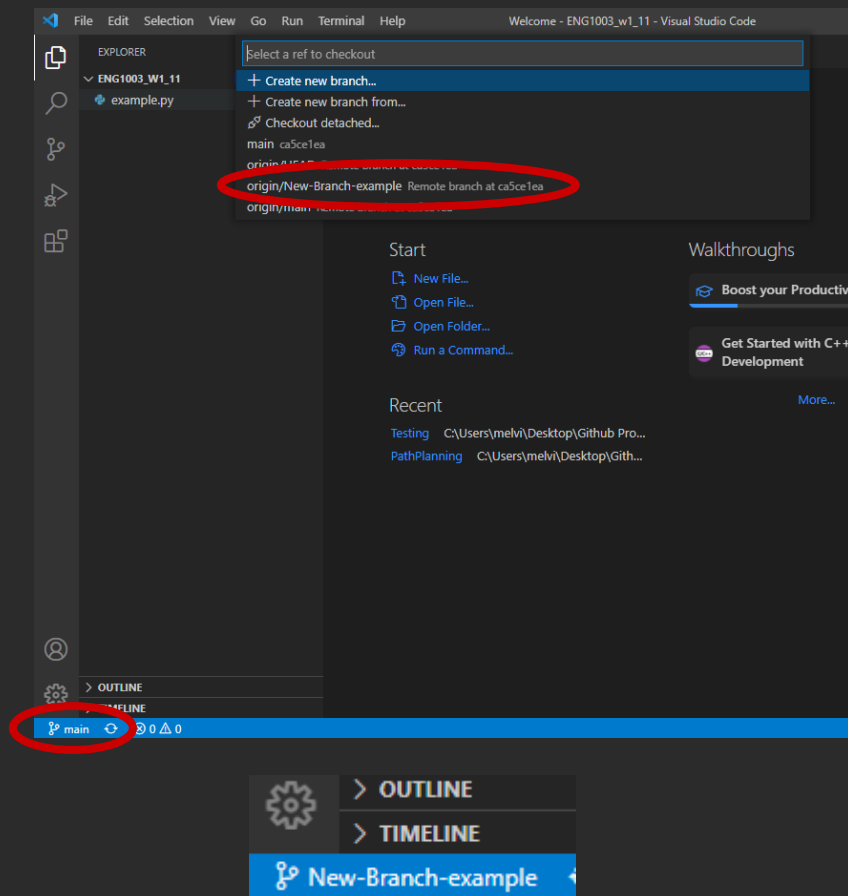
Switch to your own branch

1. Click the 'main' icon on the bottom left corner
2. Select the branch you created for yourself
3. You will see the 'main' icon replaced by the name of your own branch



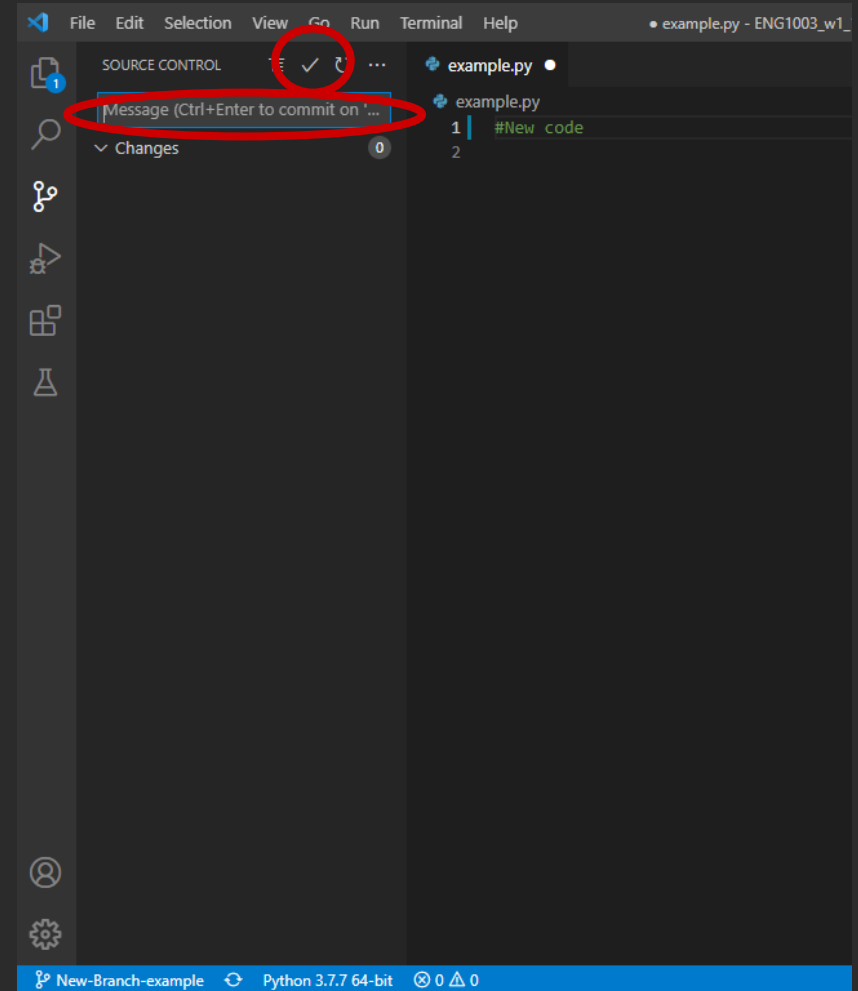
Coding

1. Edit the code once you are in your own branch



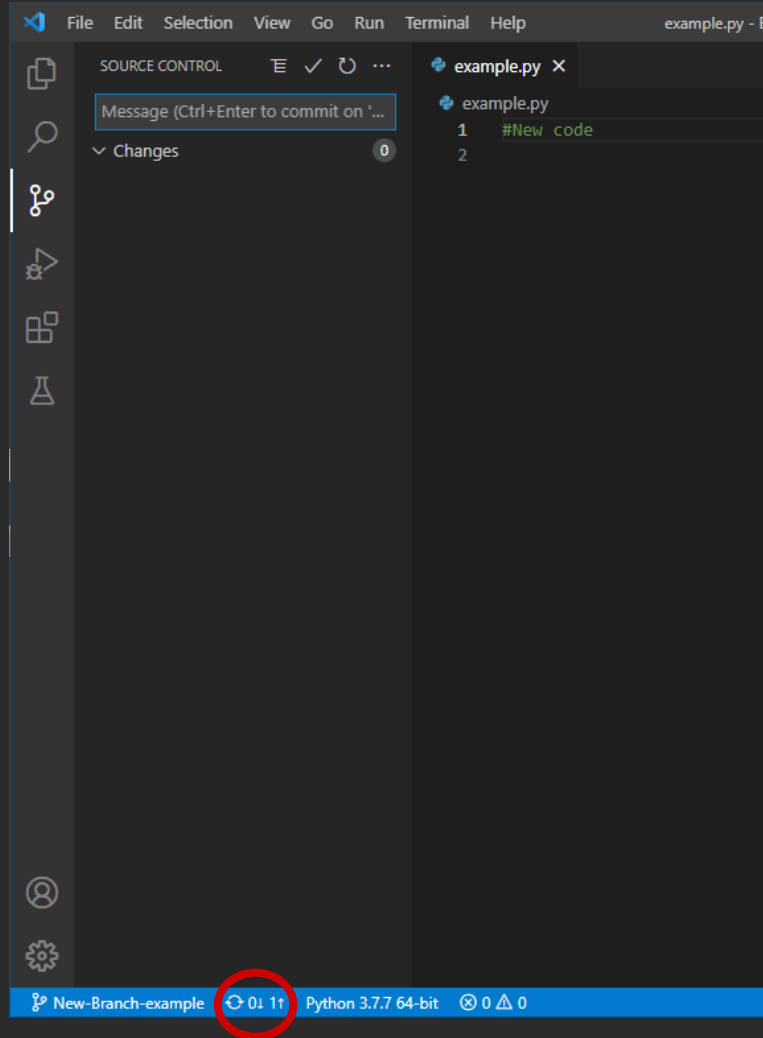
Commit Changes

1. Once you finish editing the code:
 - a. Type in a commit message
 - b. Click the tick button



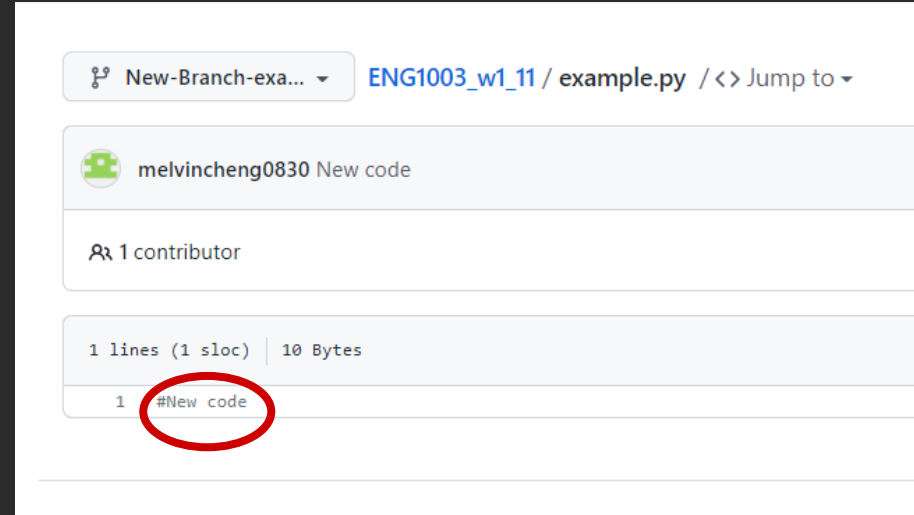
Push

1. Once you committed your changes to the local repository:
 - a. Click the push button
(Shown on the right)



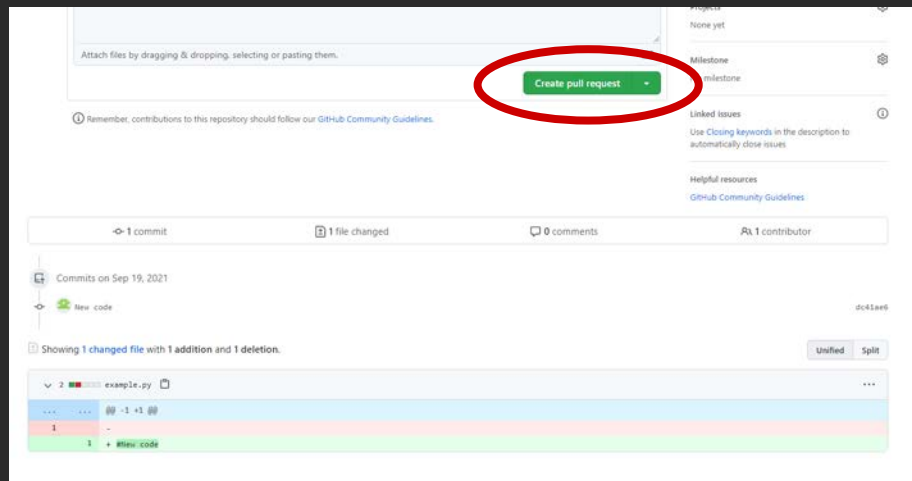
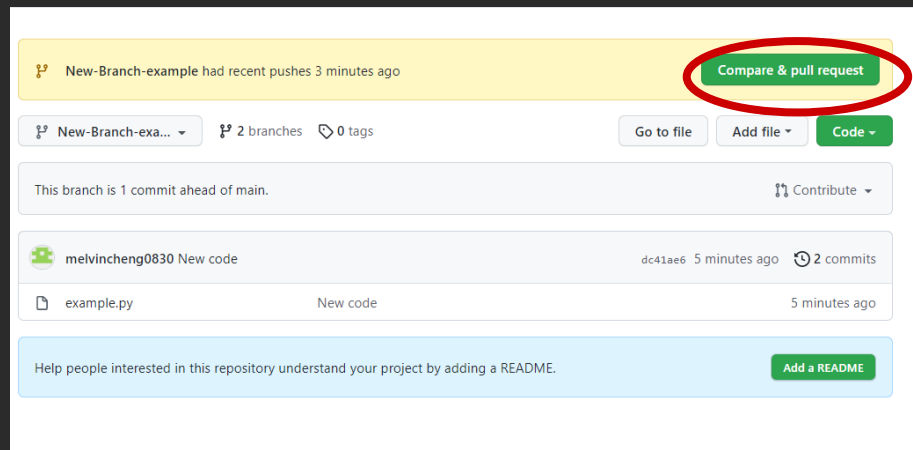
Check if push is finished

1. Go back to GitHub
2. Go to your own branch
3. See if the changes were made



Create pull request

1. Once you completed your part:
 - a. Go to your branch
 - b. Click compare and pull request
 - c. Create a pull request



Combining the codes into one programme

1. Go to 'pull request'
2. Manage all pull requests created for the repository
3. Press 'merge pull request' if you think the code is good
4. Review with your groupmates if there is any problems with the code

