UNIVERSITY OF Baguio

SCHOOL OF INFORMATION TECHNOLOGY

HUMCOM1

HUMAN-COMPUTER INTERACTION - LECTURE

UNIVERSITY OF Baguio

Table of Contents

UNIVERSITY OF Baguio

Course Overview

Course Code: HUMCOM1

Course Description: Human-Computer Interaction 1

This course teaches students to design user interfaces based on the capabilities of computer technology and the needs of human factors. The course covers human capabilities, design principles, prototyping techniques, evaluation techniques, and the implementation of graphical user interfaces. Deliverables include short programming assignments and a semester-long individual/group project. Students design a user interface for a system and implement a prototype from a list of informal requirements. Students design a user interface by a design process based on current human–computer interaction principles.

Learning Competencies:

At end of the course, you are expected to

First Grading:

1. Setup the development environment with appropriate front-end development tools that will help in

Midterms:

2. Define cognition and its relevance in interaction design.
3. Identify means on how memory can be enhanced through technology aids
4. Enumerate ways on ways which attention affects people's ability to multitask.
5. Explain through multimedia (i.e. video, interactive presentation, etc.) what is meant by social interaction and its importance in interaction design
6. Describe how technologies can be designed to change people's attitudes and behavior.

Finals:

7. Explain the rationale and rules for an effective interface design methodology in order to establish a criteria for evaluating the quality of user interfaces.
8. Explain how to design user interfaces that anticipate what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate those actions.
9. Describe how to bring together concepts from interaction design, visual design, and information architecture to create an interface with elements that are consistent and predictable in their choices and their layout.

Module Requirements:

At the end of each module, you are expected to complete each:

1. Self-check (Quiz)
2. Assignments
3. Hands-on Activities
4. Recitation

*** All module quizzes and other activities shall be placed in a long brown envelope and submitted at the end of each module.

UNIVERSITY OF Baguio

Contact Information:

emartinez@e.ubaguio.edu
https://www.facebook.com/krejpinoy

Consultation Hours:

Course schedule

Course Schedule and Outline

	Lecture	Duration (Hours)	Requirements
Week 1	Development Environment Set-up Introduction to HTML5	3	✓ GitHub Account ✓ Published home page
Week 2	HTML Elements Part 1	3	✓
Week 4-5		3	✓
Week 6	First Grading Exam	2	
Week 7		2	✓
Week 8		2	✓
Week 9-10		3	✓
Week 10-11		3	✓
Week 11	Midterm Exam	2	
Week 13-14		4	✓
Week 15-17		6	✓
Week 18	Final Exam	2	

UNIVERSITY OF Baguio

LESSON 1: Development Environment Set-up

Duration: 2 hours

About this lesson: At the end of this lesson you are expected to:

- Setup and configure your development environment
- Setup a GitHub account and configure local and remote repositories.
- Familiarize the use of basic Git commands and use of code editors.

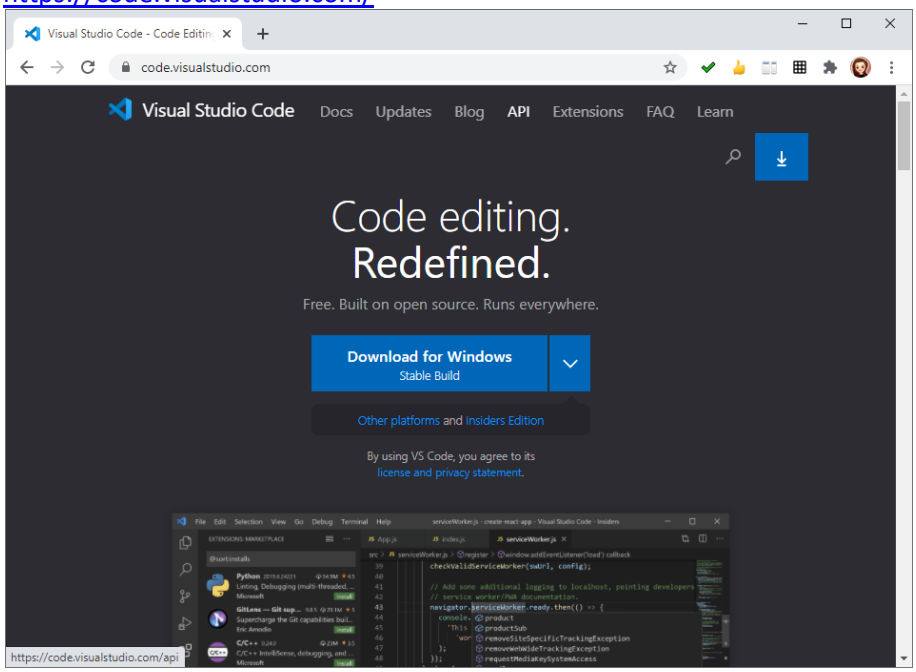
1.1 What you'll need

1. Code editor
    - Visual Studio Code or Notepad++ -> you can use any of these code edits. Both are free with support for various programming languages.
2. Git
3. Nodejs
4. Browser-sync
5. Github
6. Browsers
    - Chrome, Safari, Opera, and/or Edge.

1.2 Development environment setup

a. Visual Studio Code (or Notepad++)

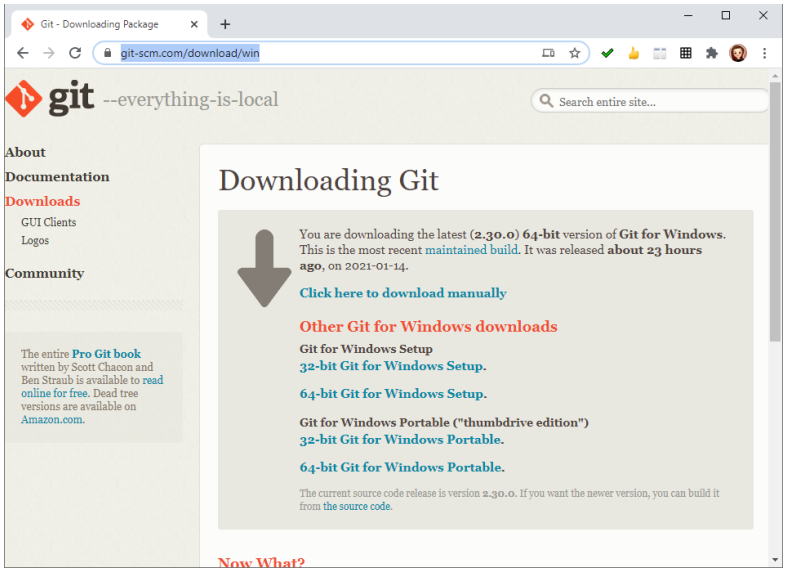
- Go to https://code.visualstudio.com/



- *Download* the *stable* Visual Studio Code installer
- *Install*. Follow the set-up (installation) wizard.
  **** You may use Notepad++ as your code editor. Notepad++ is a lightweight code editor (has lower system requirements) than Visual Studio Code, but is very efficient as a code editor.*

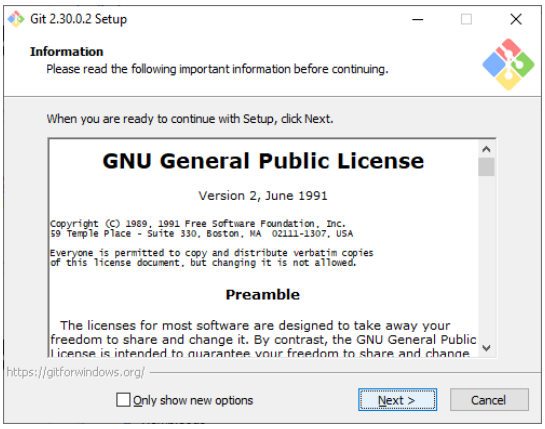
b. Git - is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. (Git and Software Freedom Conservancy, n.d.)

- Go to https://git-scm.com/download/win (for windows)

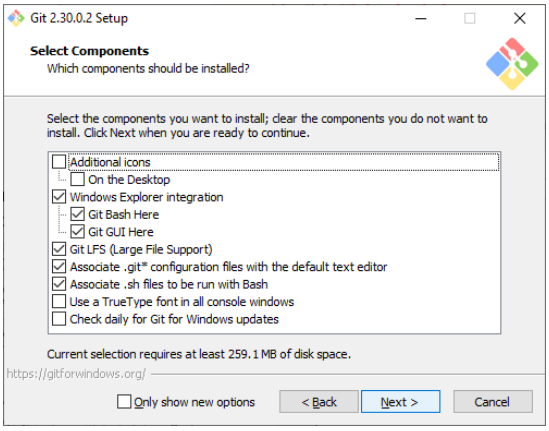


- Download either 32bit or 64bit for Windows, (make sure you download the version compatible for your computer)
- When completed, install the software. Follow the installation steps.

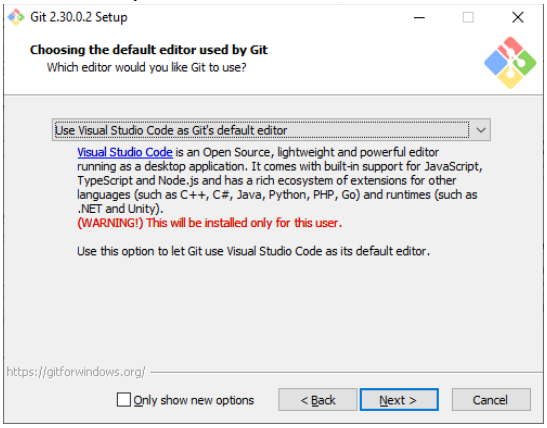
1. Just Click Next in this window

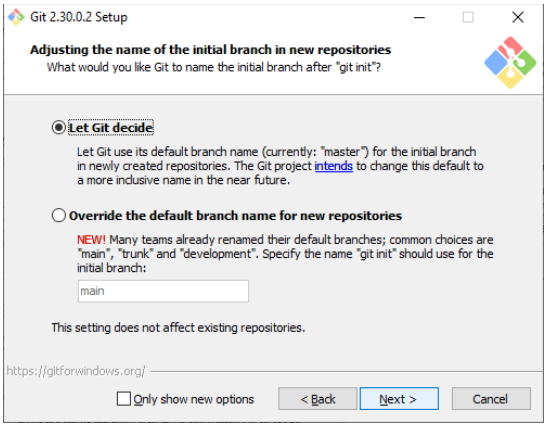


2. Just Click Next in this window

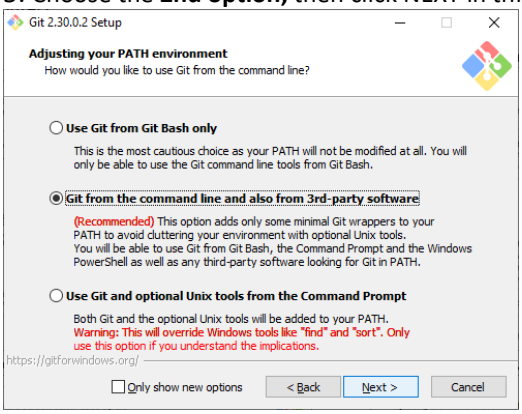


3. Choose your code editor from the list then click Next 4. Click Next in this Window

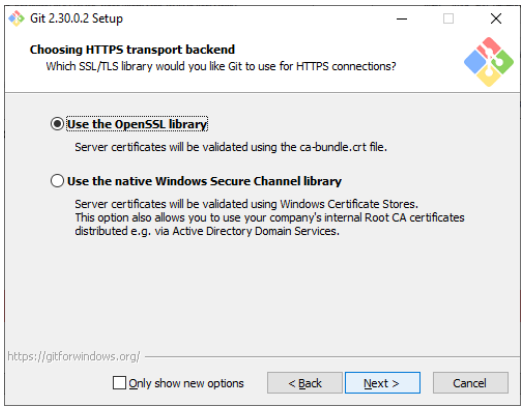




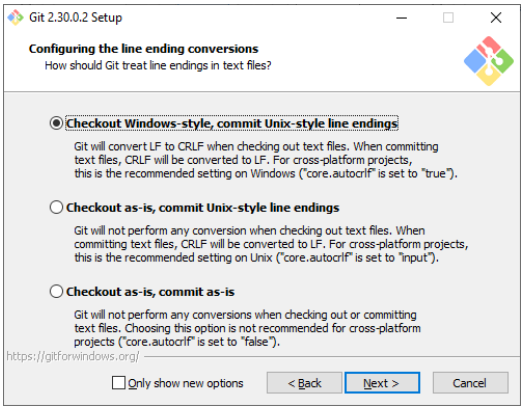
5. Choose the **2nd option,** then click NEXT in this window



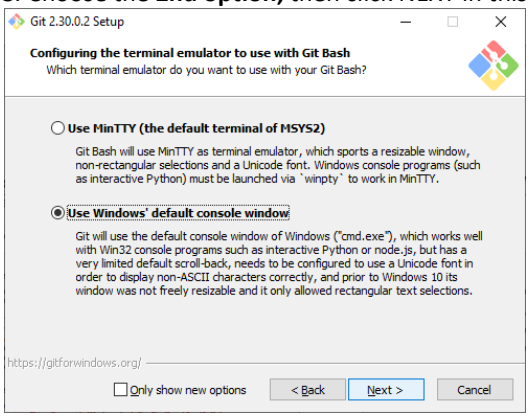
6. Click Next in this Window



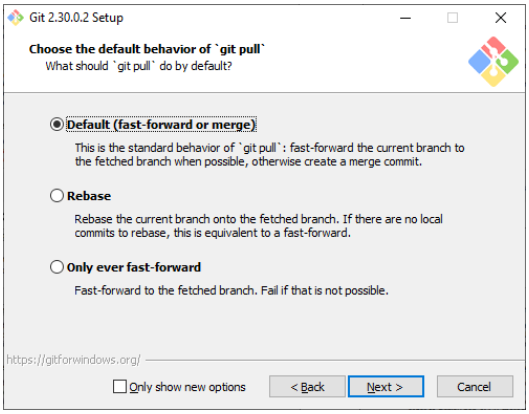
7. Click Next in this Window



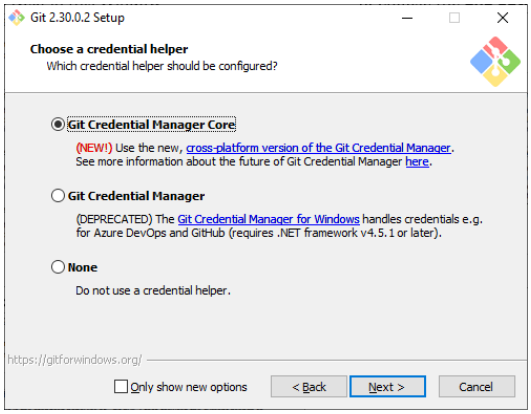
8. Choose the **2nd option,** then click NEXT in this window



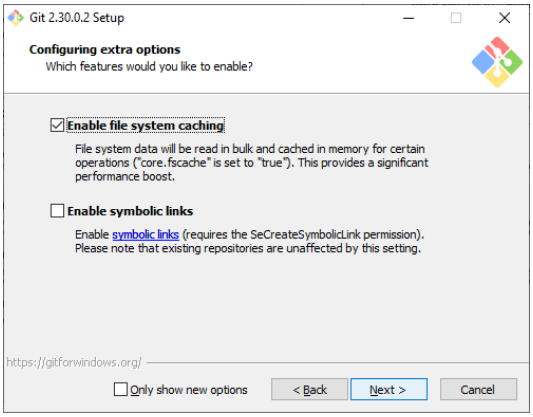
9. Click Next in this Window



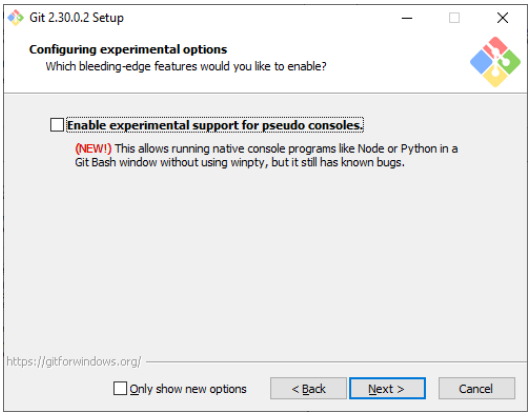
10. Click Next in this Window



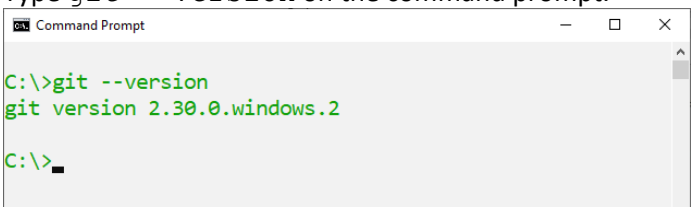
11. Click Next in this Window



12. Click Next in this Window

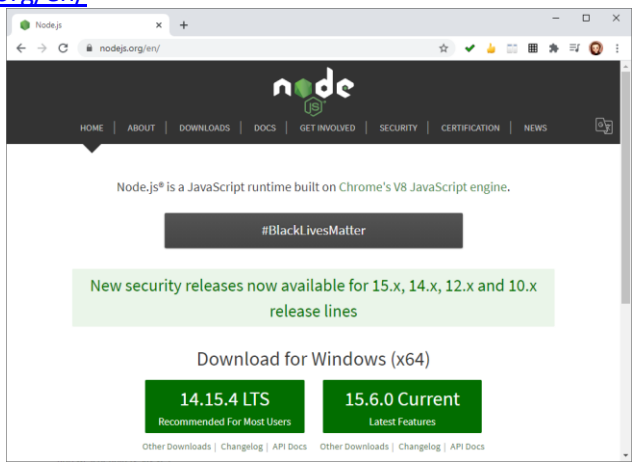


- Finish the installation process from there.
- Check if the application is installed through the Windows command prompt.
- Type `git --version` on the command prompt.

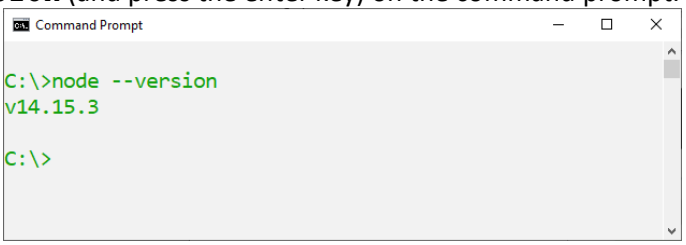


c. Nodejs - As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications. In the following "hello world" example, many connections can be handled concurrently.

- Go to https://nodejs.org/en/



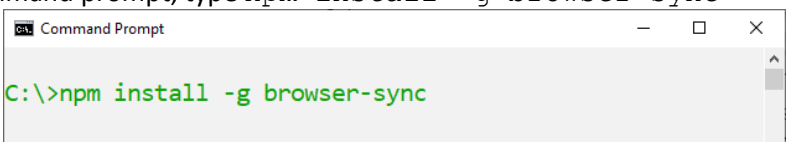
- Download and install the *Nodejs LTS* Windows installer.
- After installation, check if it was installed through the Windows command prompt.
- Type, `node --version` (and press the enter key) on the command prompt.



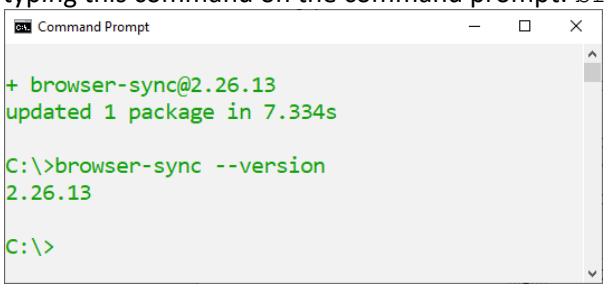
- Then, type, `npm -version`

d. Browser-sync - is an automation tool that makes web development faster by synchronizing file changes and interactions across many devices.

- This step should only be done after installing NodeJs.
- On the command prompt, type `npm install -g browser-sync`

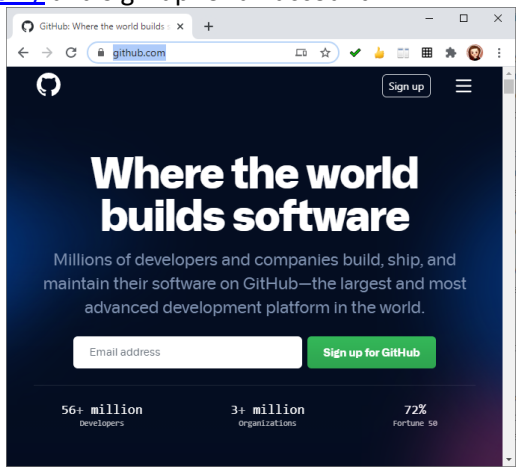


- Verify that it is installed by typing this command on the command prompt: `browser-sync --version`

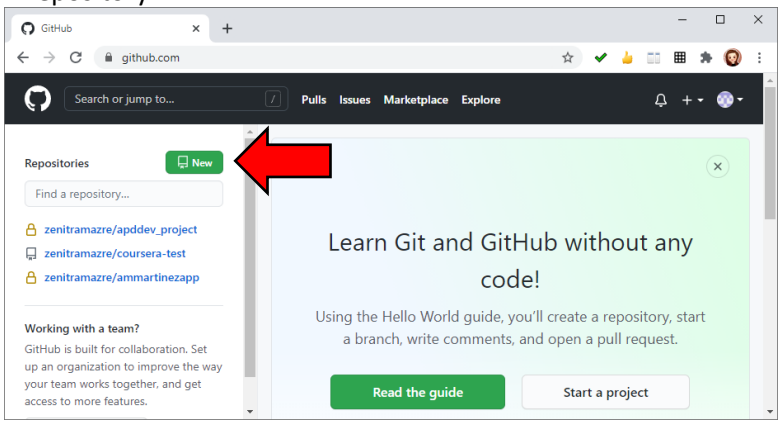


e. Github - GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere. You will be using pushing all your activities in your Github repository for this course.

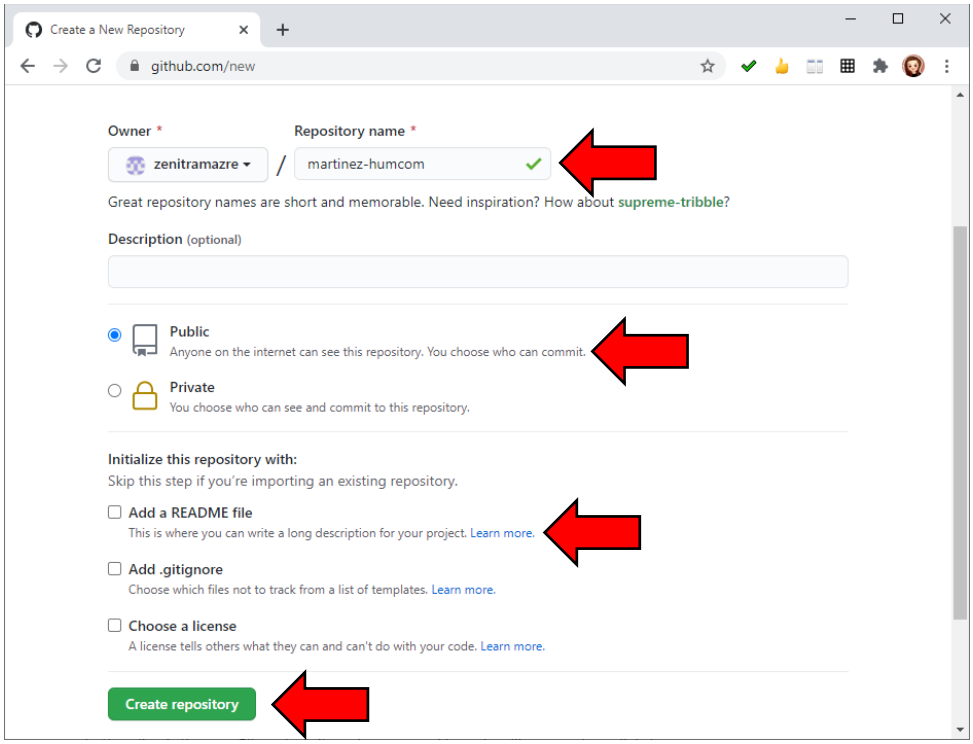
- Go to https://github.com/ and sign-up for an account



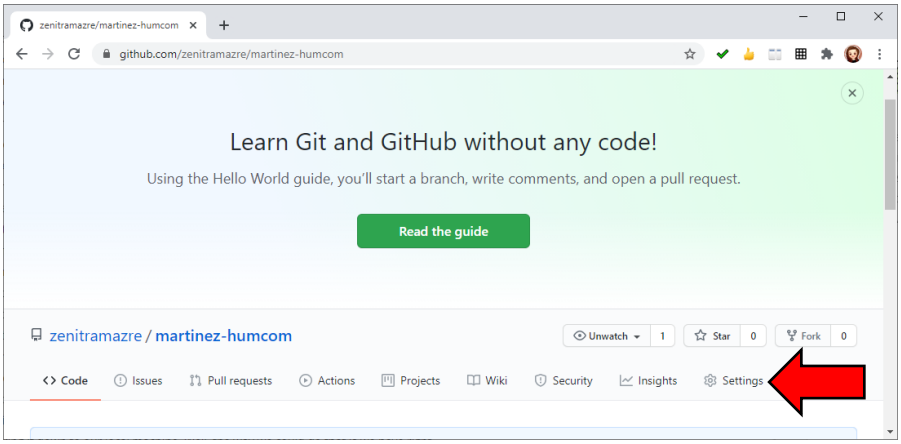
- After signing-up, sign-in to your account.
- Create a new repository



- Name the repository: lastname-humcom
- Set the repository to public. (Setting it to private requires a paid account)
- Enable 'Add a README file' for documentation purposes.
- Click 'Create repository'

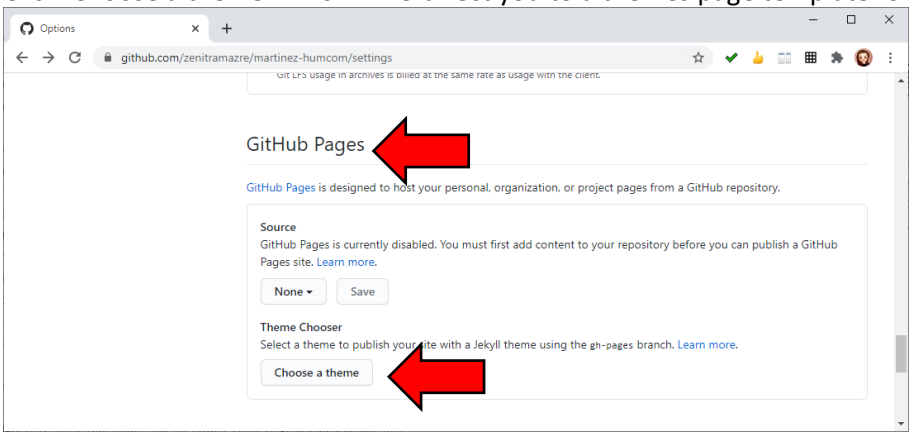


a. Go to the settings of your new repository

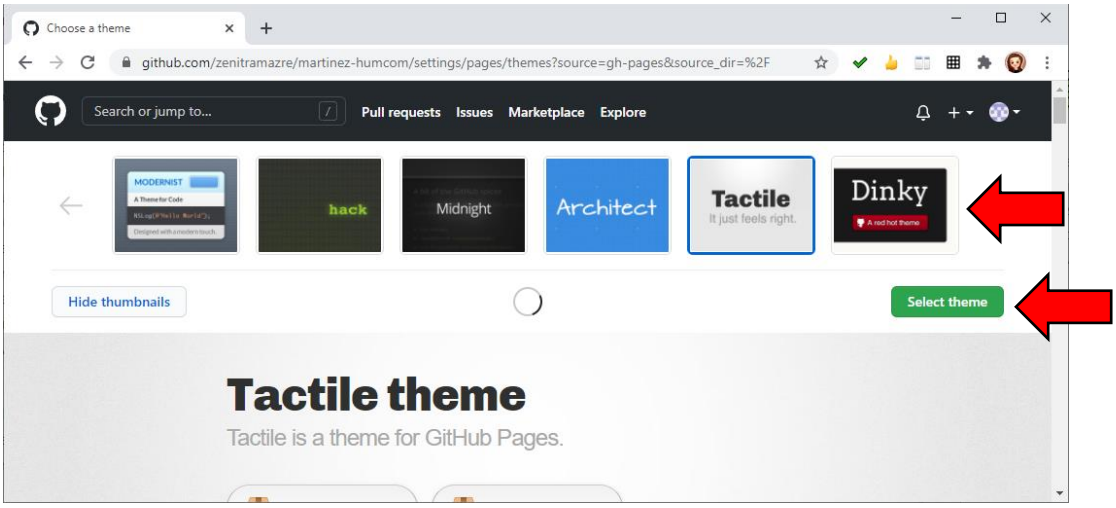


b. Scroll down to GitHub Pages

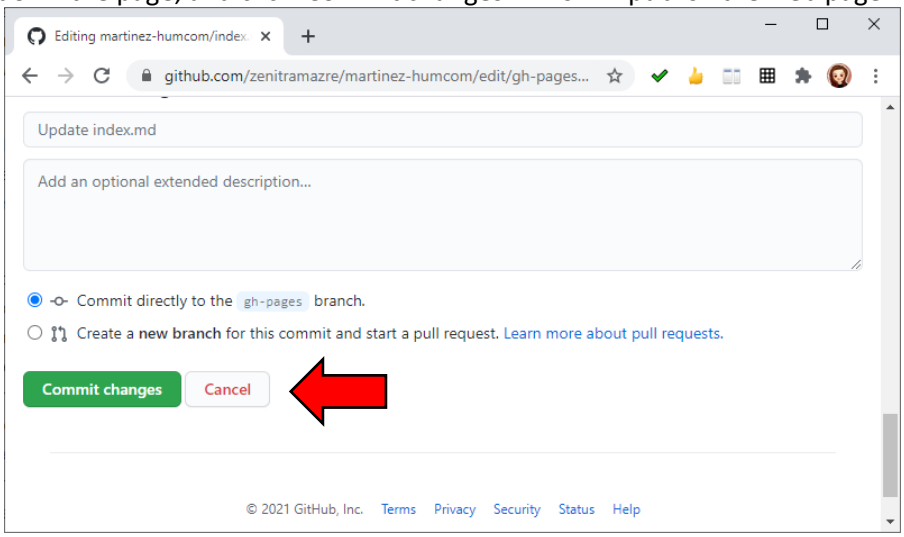
c. Click 'Choose a theme'. This will re-direct you to a themes page template for a homepage.



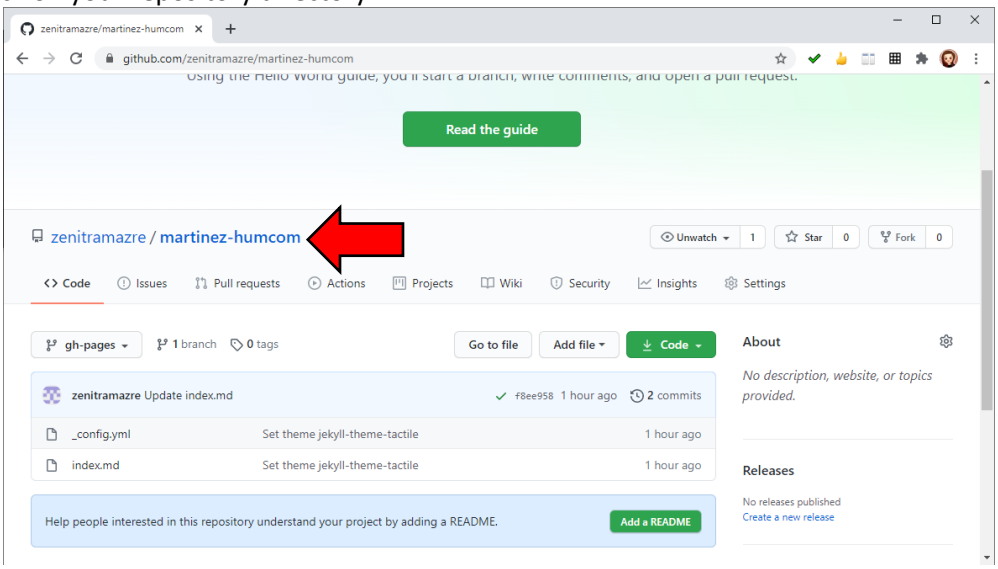
- Choose from any of the templates. Then click on 'Select theme'



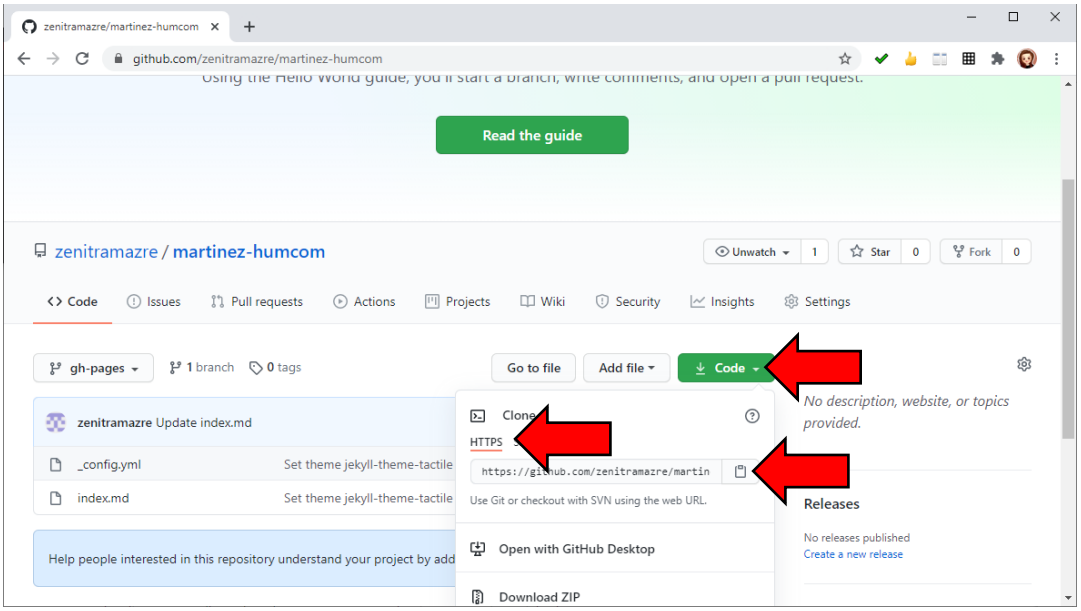
- You may edit this now (or later). If you want. This is not our purpose for now.
- Scroll down the page, and click 'Commit changes'. This will publish the web page.



- Click on your repository directory.

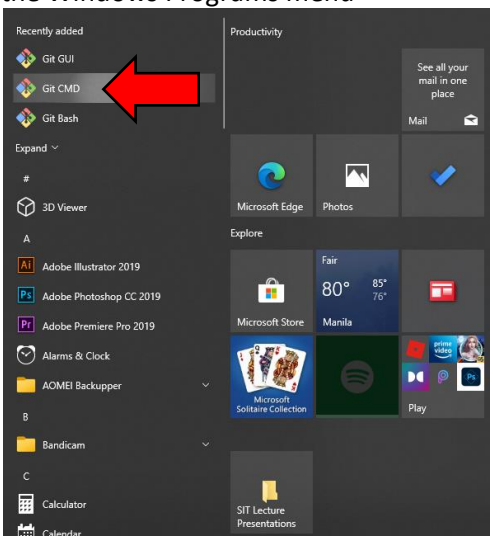


- Then click on the 'Code' button. Select HTTPS. Copy to clipboard. You will need this URL to clone the repository to your local machine.

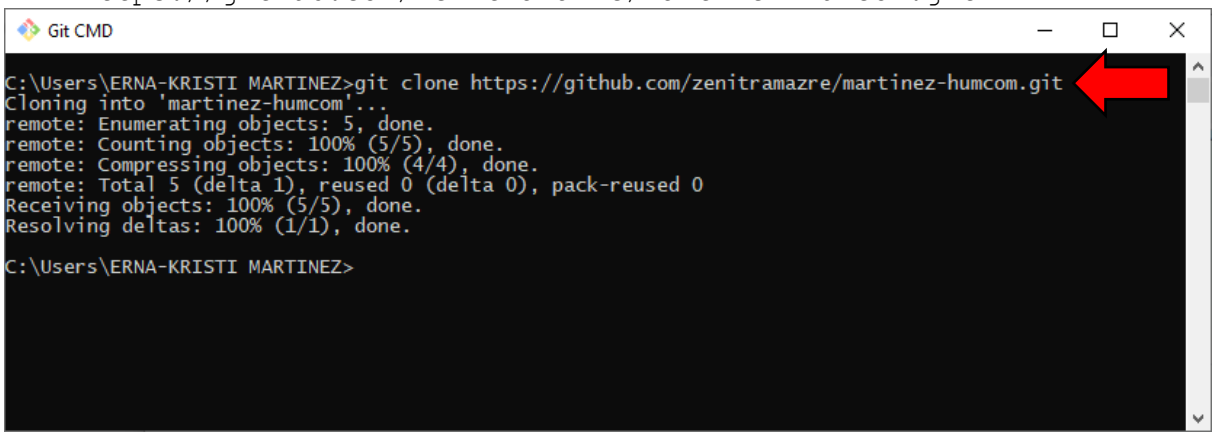


d. Setting up locally – cloning the repository on your pc to make it easier to sync your projects to the web repository.

- Open the Git cmd from the Windows Programs menu

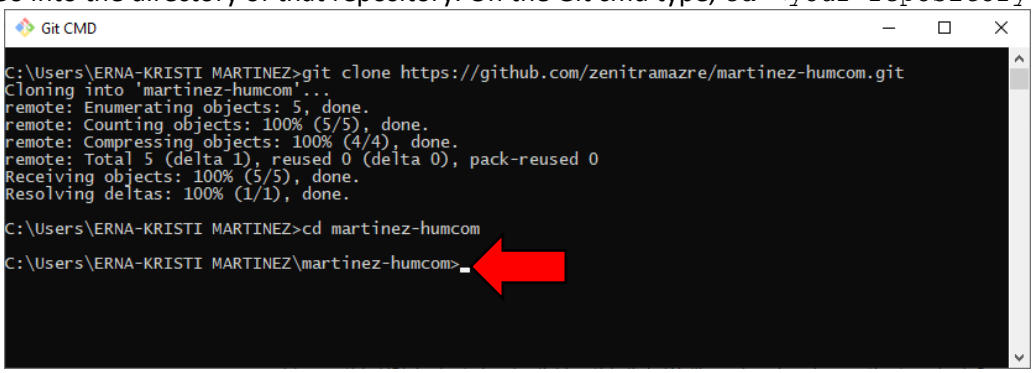


- Once it is open, on the command prompt type, `git clone (paste the copied URL) https://github.com/zenitramazre/martinez-humcom.git`

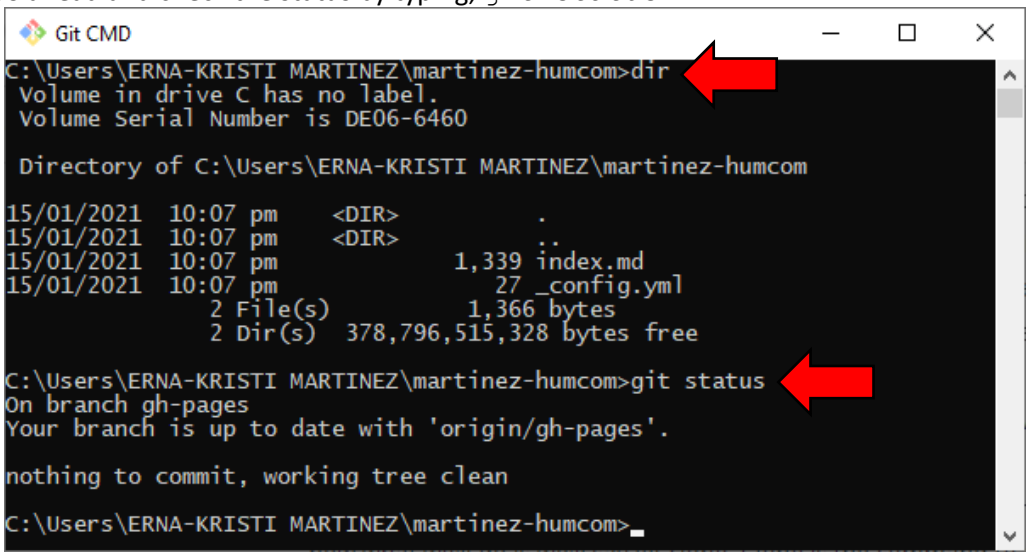


- The repository should be cloned in your local machine.

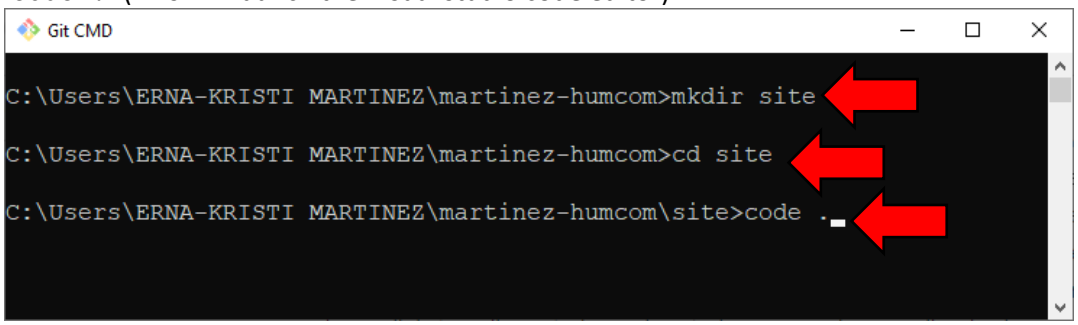
- Go into the directory of that repository. On the Git cmd type, `cd <your repository name>`



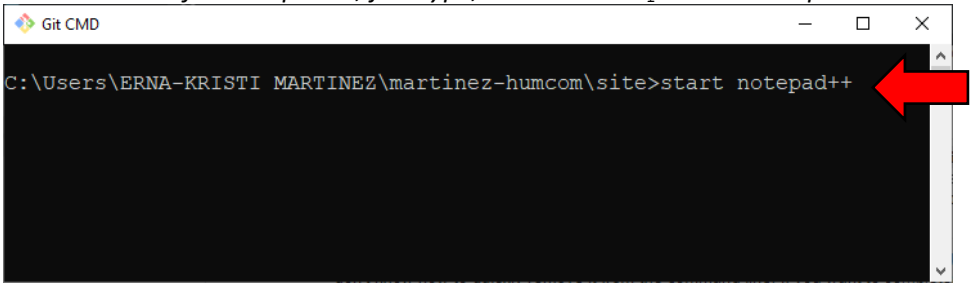
- You may check what is in the repository (or directory) by typing, `dir`
- Go ahead and check the status by typing, `git status`



- Create a new folder in the current directory by typing, `mkdir site`
- Go to that new folder by typing, `cd site`
- This step is only for Visual studio code users. On the command prompt type (and press enter after), `code .` (This will launch the Visual studio code editor)

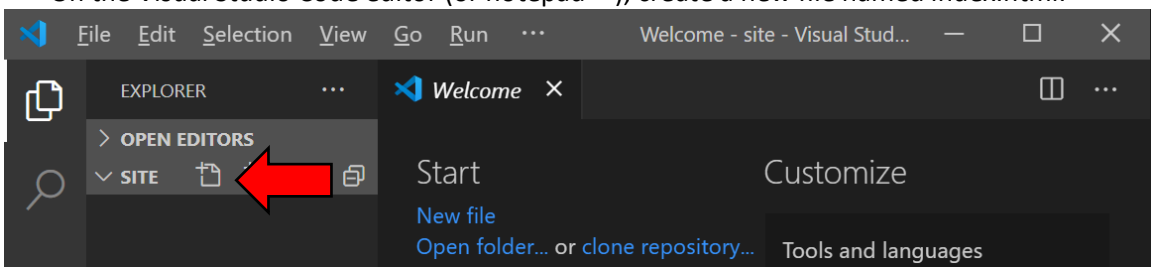


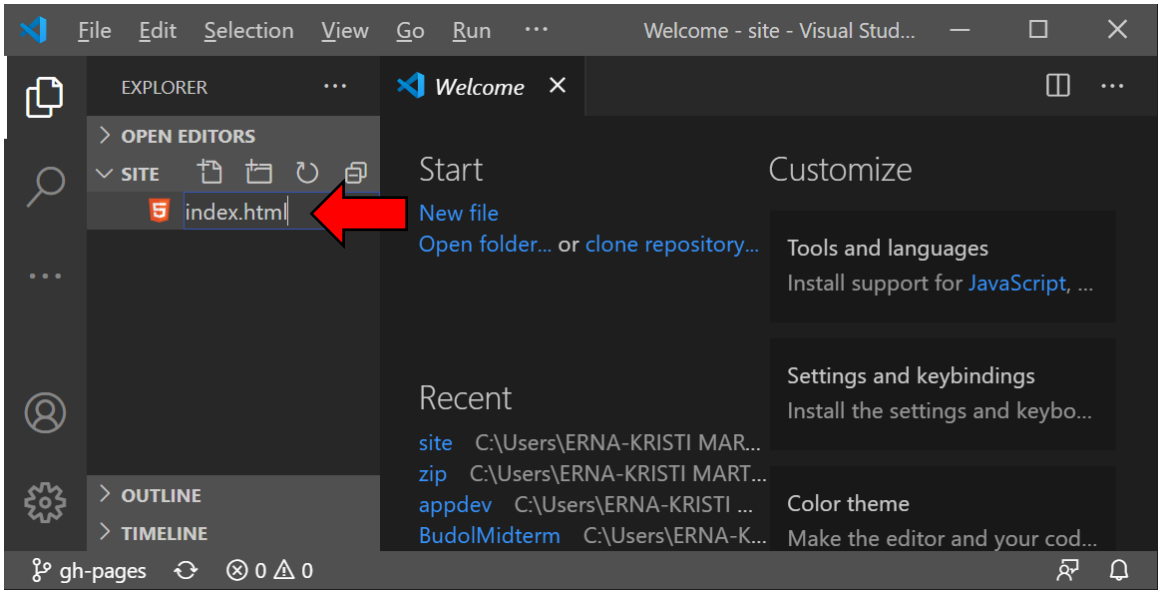
****Alternative for Notepad++, just type,* `start notepad++ and` *press enter.*



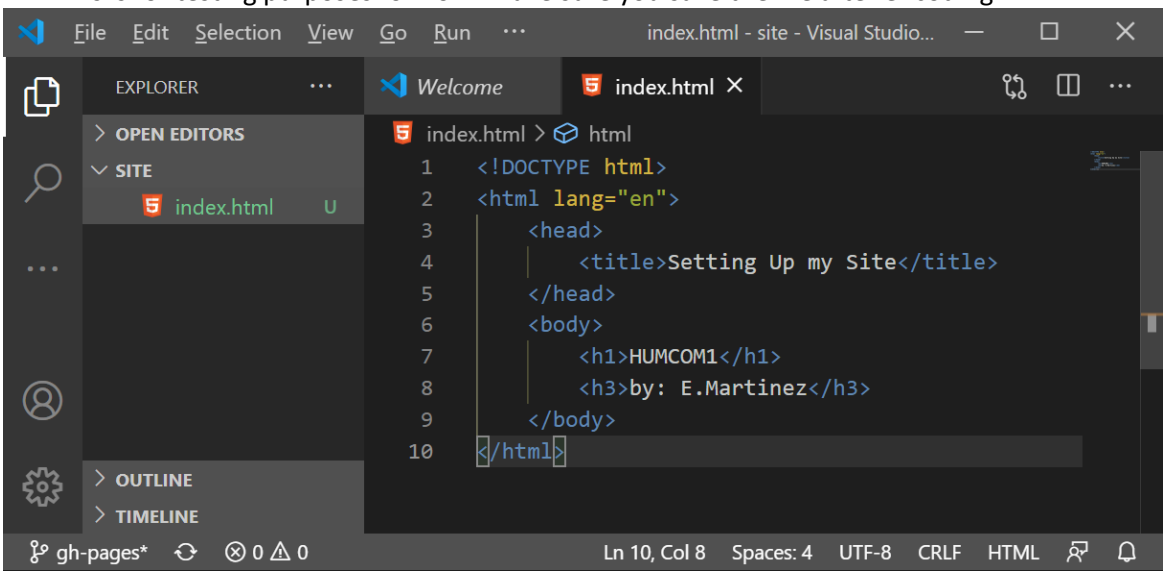
e. GitHub and Browser

- On the Visual Studio Code editor (or notepad++), create a new file named index.html.

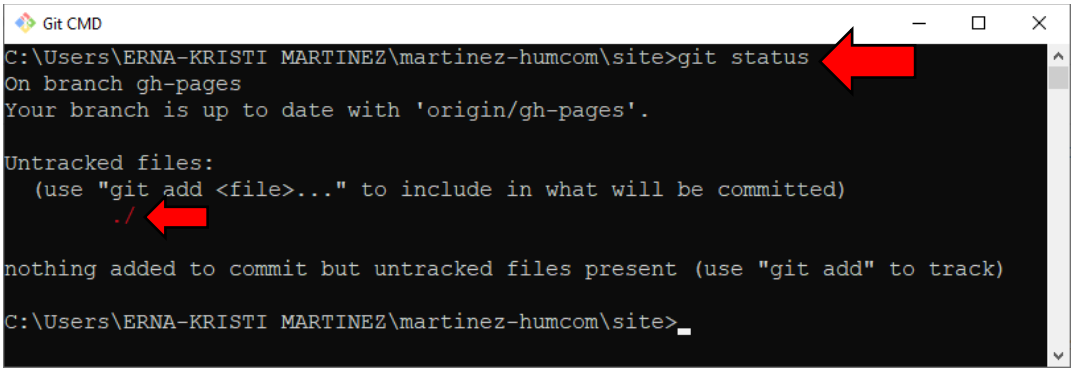




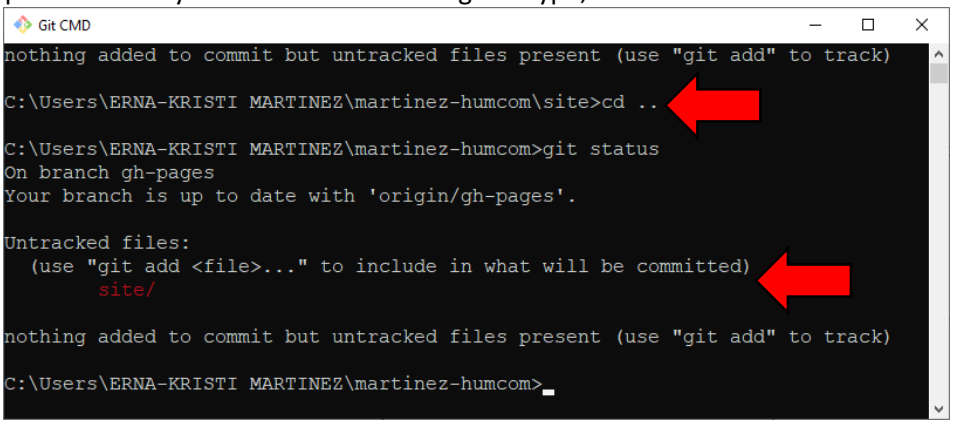
- On the page editor, type the following code for now. The context will be explained in the next lessons. This is for testing purposes for now. Make sure you save the file after encoding.



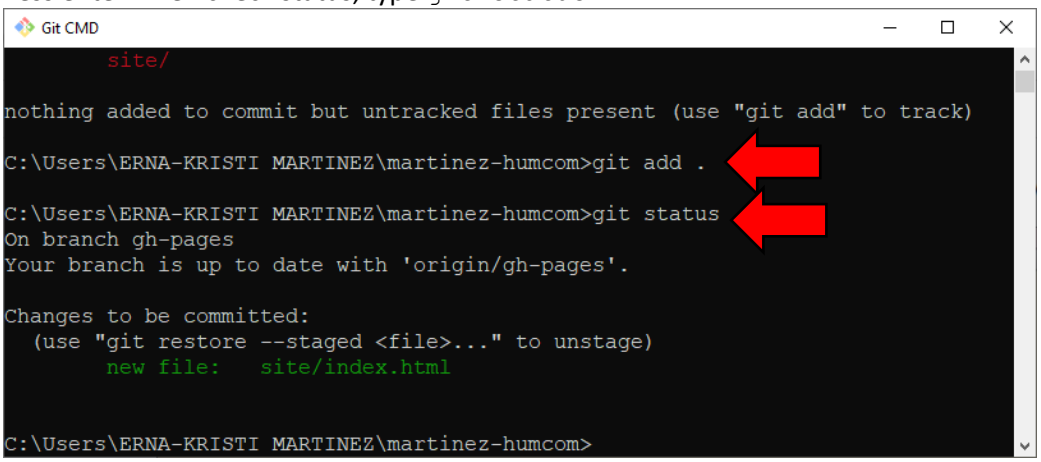
- Go back to your Git cmd and check the repository status. Type: `git status`
- It will indicate that a new file has been added in the site directory as indicated by the '**./**'. This means that the local repository is no longer 'in sync' with the web repository.



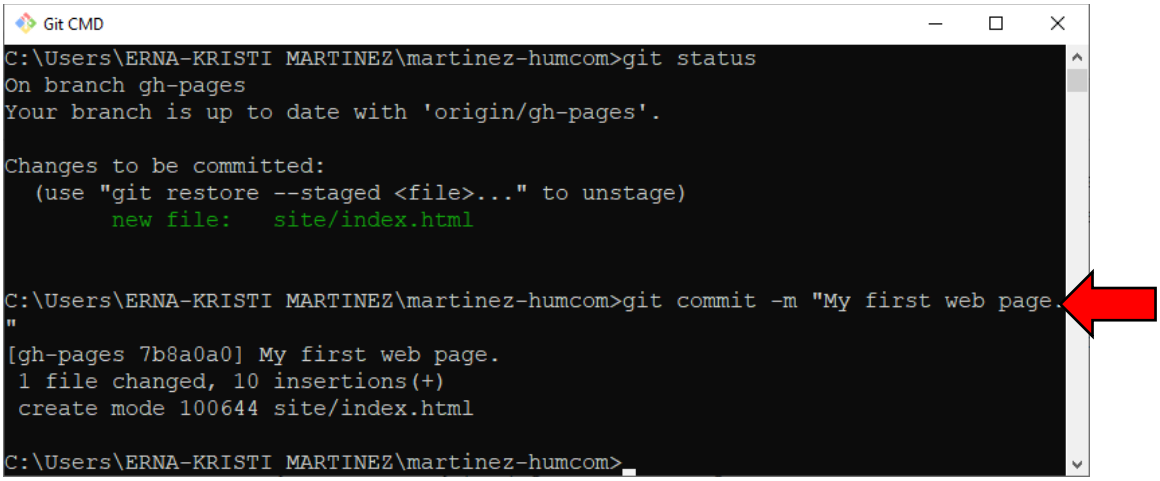
- Go up one directory and check the status again. Type, cd ..



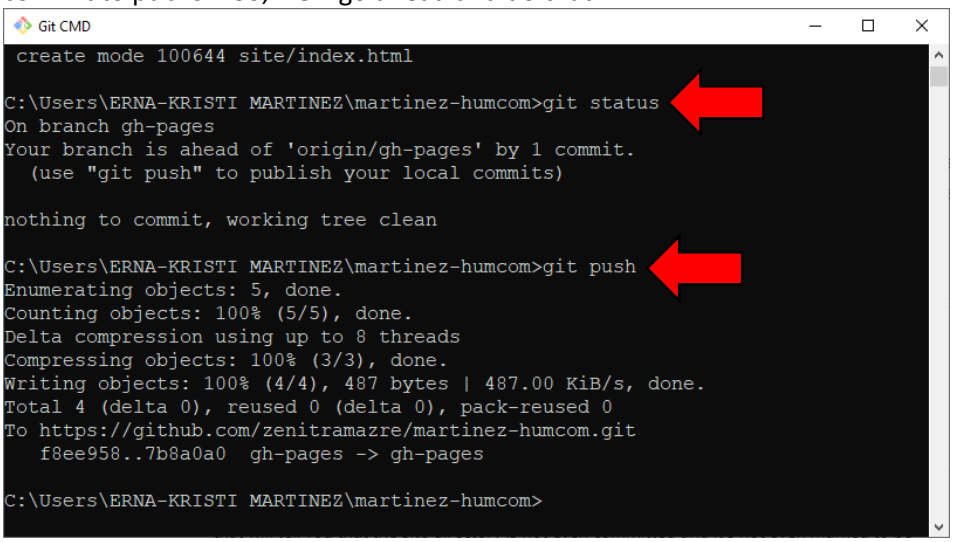
- To sync the local and remote repository, we need to mark something for commiting and push the changes in the repository.
- So, on the command line type, `git add .` (the dot includes everything in the local machine)
- Press enter. Then check status, type `git status`



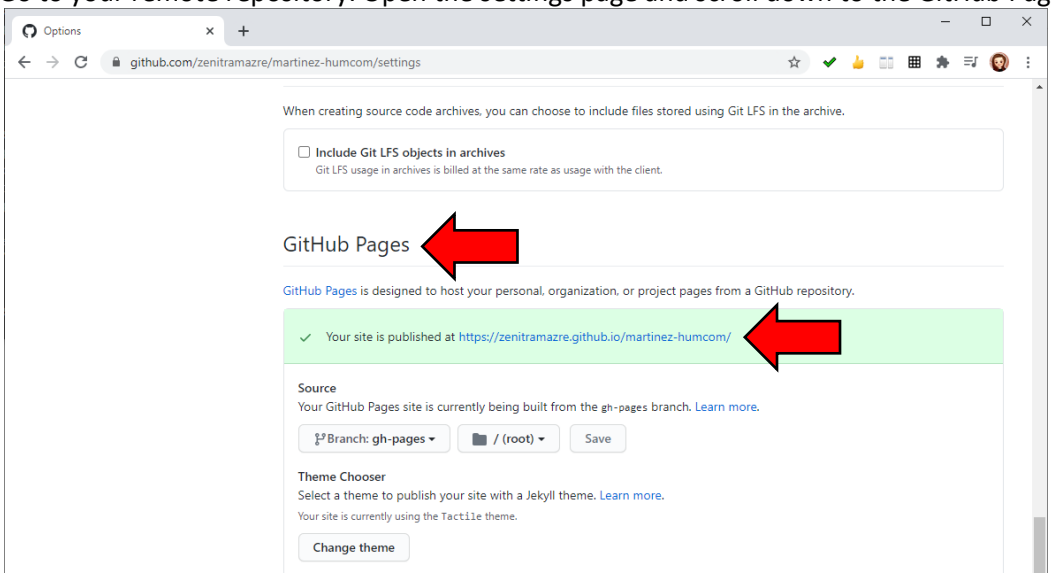
- It is now marked as committed. To commit the new file, type: `git commit –m "My first web page."` (the –m is require to attach a message for the commit)



- Now, all you need to do is to push this commit up on to the remote repository.
- Type, `git status` and it will tell you that the local repo is ahead by 1 commit and to 'push' the new commit to publish. So, we'll go ahead and do that.

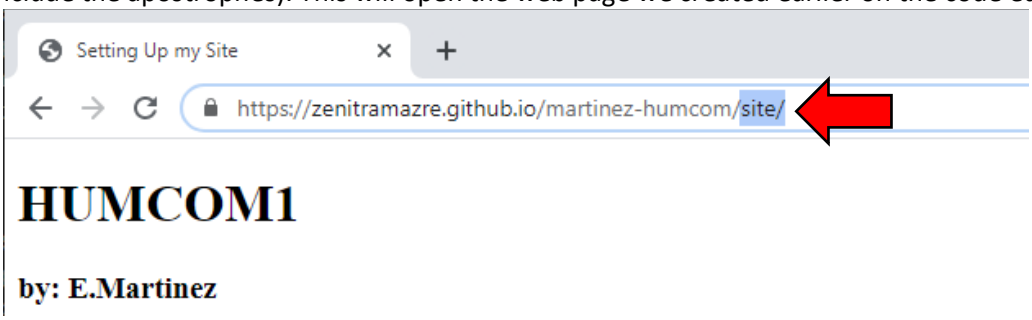


- Now the local and remote repo are in sync.
- Go to your remote repository. Open the settings page and scroll down to the GitHub Pages



- Click on the highlighted published URL shown above.

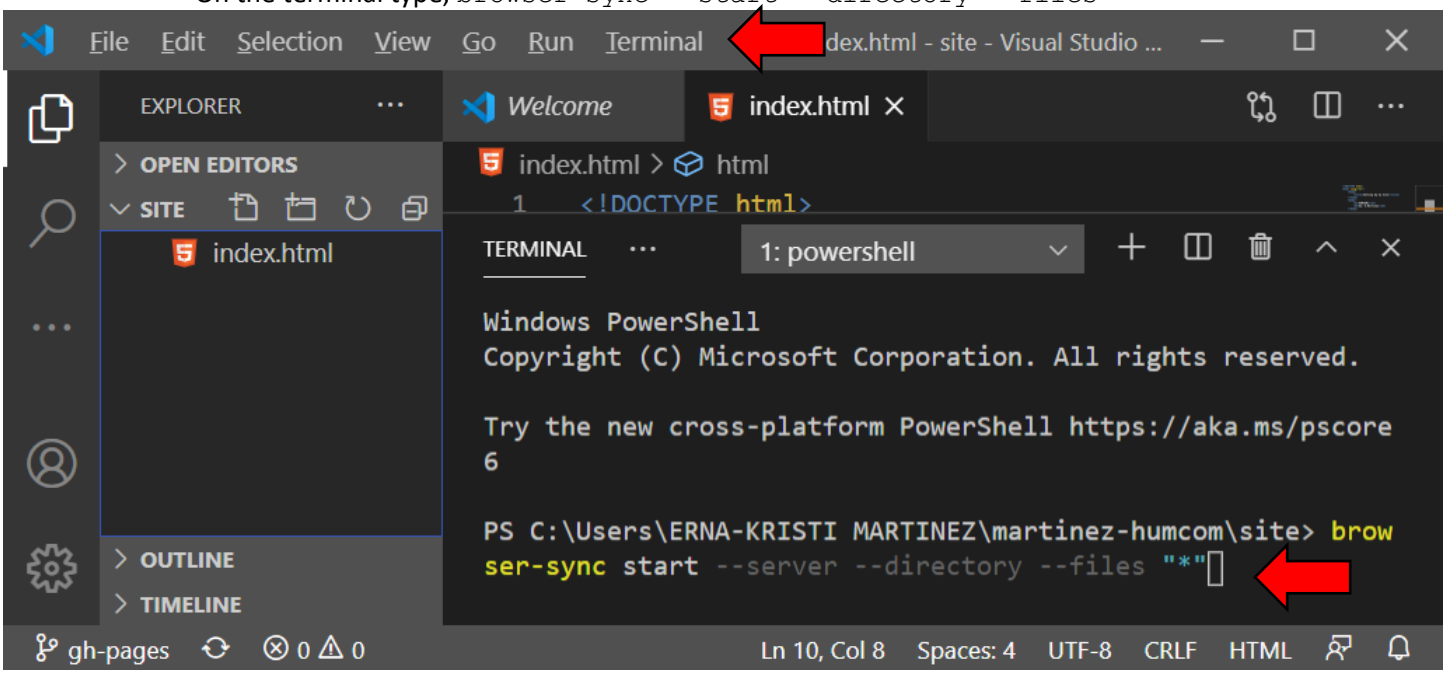
- On the address bar of the browser on the new page that is opened. Add 'site/' on the end. (Don't include the apostrophes). This will open the web page we created earlier on the code editor.



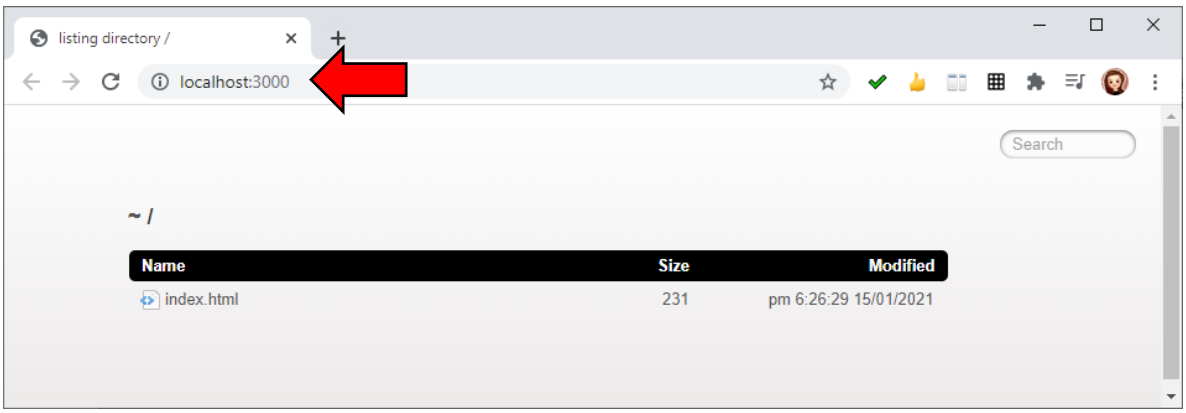
- Congratulations! You just published your first web page online.

f. Using Browser-sync

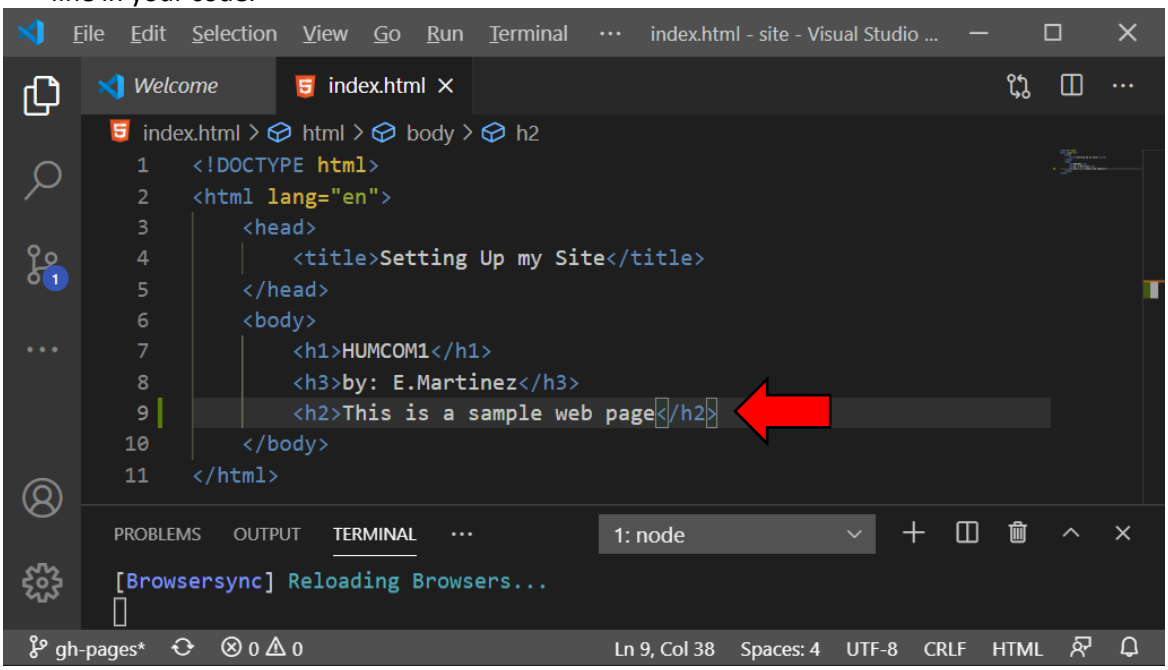
- On visual studio code, open the terminal from the menu (or press Ctrl+Shift+` on the keyboard). *** or just use the Windows cmd prompt (not the Git CMD)
- The Terminal is Visual studio code's built in command line interface.
- On the terminal type, `browser-sync --start --directory --files "*"`



- This will launch or open a new web page showing your directory listing (or repository directory) served on a localhost:3000.



- Click on 'index.html' file from the page. While the page is open, go back to your code editor and edit some of the content in the web page as shown below. Observe the web page as you are typing the new line in your code.



- So, `browser-sync --start --directory --files "*"`,
  - `--start`: starts the browser-sync
  - `--directory`: automatically syncs your directory, once you add any new files or folders in the directory
  - `--files`: automatically reloads any changes or modifications on local files

UNIVERSITY OF Baguio

Name:		Date:	
Development Environment Set-up		Lesson 1 **Activity**	Score: (HPS 20points)

Lesson 1 | Activity

1. Signup for GitHub account.
2. Publish your first sample web page as we have done in the lesson.
3. Submit your URL to your instructor.

Name:		Date:	
Introduction to HTML HTML Elements Part1		Lesson 2-3 **Activity 4 (Lists 2)**	Score:

References

Word Wide Web Consortium. (2015, October 23). 3.2.5 Content models — HTML5. Https://Www.W3.Org/. https://www.w3.org/TR/2011/WD-html5-20110525/content-models.html#flow-content