



# INFO 101 Week 2

## Thursday-Section AD

Today's Topic: Builders, basic HTML



# Quiz Time (20 min)

Please start any of your Thinker or Speaker quizzes

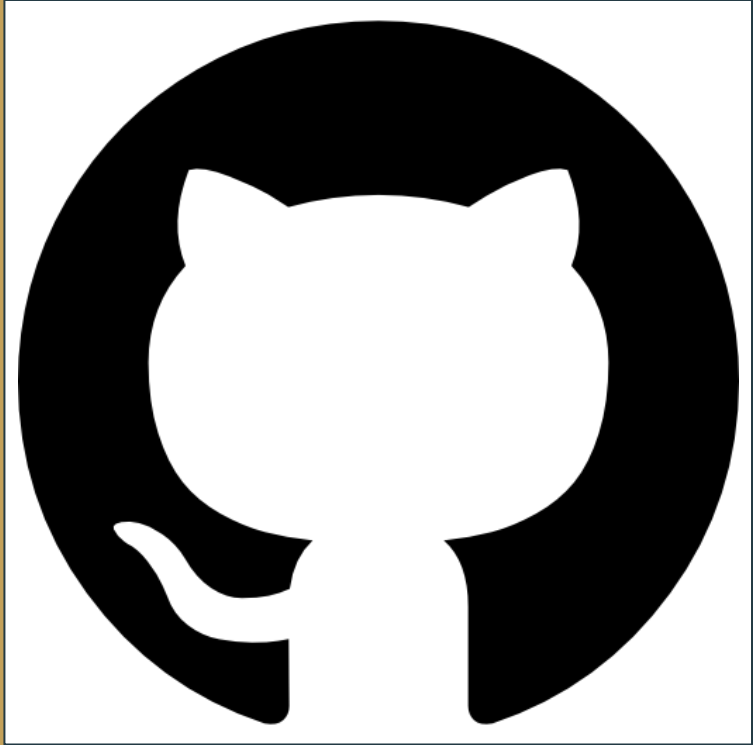
Any unfinished quizzes after the allotted time will be requeued and you will have to do them again during a different section.

# Builder Agenda

To complete this builder:

1. Create a Github account
2. Create a Github Repository
3. Make a hello world page
4. Upload your file to Github's web servers
5. Make enhancements (optional)

What is a Github?



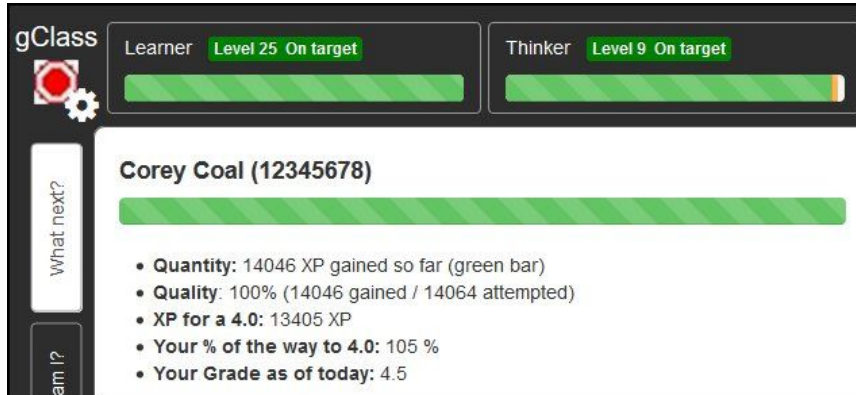
# Establishing a Web Site

First, we need to create a Github account!

We need you to make a new account with this specific name so we can find you.

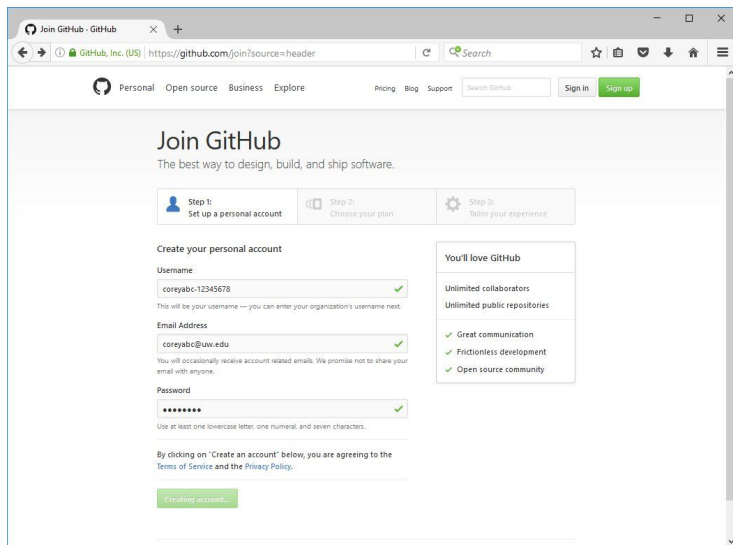
Your uw net id - your student number

Ex: coreya04-12345678



# Setting up Github

1. Go to Github: <https://github.com/join?source=header-home>
2. Fill out the join form and click create an account. Use a valid email account because Github will use it to verify your account



The screenshot shows the GitHub 'Join GitHub' page. The header includes navigation links: Personal, Open source, Business, Explore, Pricing, Blog, Support, and a search bar. The main heading is 'Join GitHub' with the tagline 'The best way to design, build, and ship software.' Below this is a progress bar with three steps: Step 1: Set up a personal account (active), Step 2: Choose your plan, and Step 3: Tailor your experience. The 'Create your personal account' section contains three input fields: Username (coreyabc-12345678), Email Address (coreyabc@uw.edu), and Password (masked with dots). Each field has a green checkmark indicating it is valid. Below the password field is a note: 'Use at least one lowercase letter, one numeral, and seven characters.' At the bottom, there is a link to the Terms of Service and Privacy Policy, and a green button labeled 'Create account'.

Join GitHub  
The best way to design, build, and ship software.

Step 1: Set up a personal account | Step 2: Choose your plan | Step 3: Tailor your experience

Create your personal account

Username  
coreyabc-12345678 ✓  
This will be your username — you can enter your organization's username next.

Email Address  
coreyabc@uw.edu ✓  
You will occasionally receive account related emails. We promise not to share your email with anyone.

Password  
•••••••• ✓  
Use at least one lowercase letter, one numeral, and seven characters.

By clicking on "Create account" below, you are agreeing to the Terms of Service and the Privacy Policy.

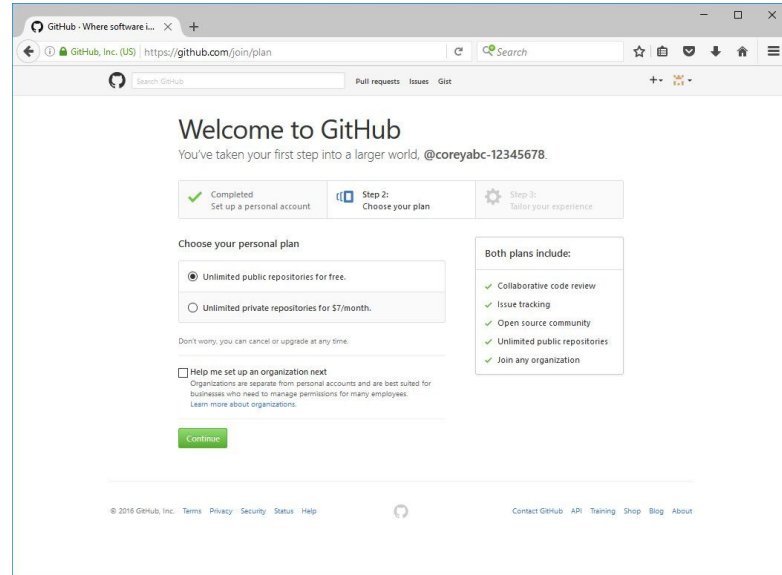
Create account

You'll love GitHub

- ✓ Unlimited collaborators
- ✓ Unlimited public repositories
- ✓ Great communication
- ✓ Frictionless development
- ✓ Open source community

# Setting up Github

## 3. Choose the free plan and click Continue



# Setting up Github

## 4. Fill out or skip the Welcome page

GitHub - Where software ...  
https://github.com/join/customize

Welcome to GitHub  
You'll find endless opportunities to learn, code, and create, @coreyabc-12345678.

Completed: Set up a personal account  
Step 2: Choose your plan  
Step 3: Tailor your experience

How would you describe your level of programming experience?

☐ Very experienced ☐ Somewhat experienced ☒ Totally new to programming

What do you plan to use GitHub for? (check all that apply)

☒ School projects ☐ Research ☐ Design  
☐ Project Management ☐ Development ☐ Other (please specify)

Which is closest to how you would describe yourself?

☐ I'm a professional ☐ I'm a hobbyist ☒ I'm a student  
☐ Other (please specify)

What are you interested in?

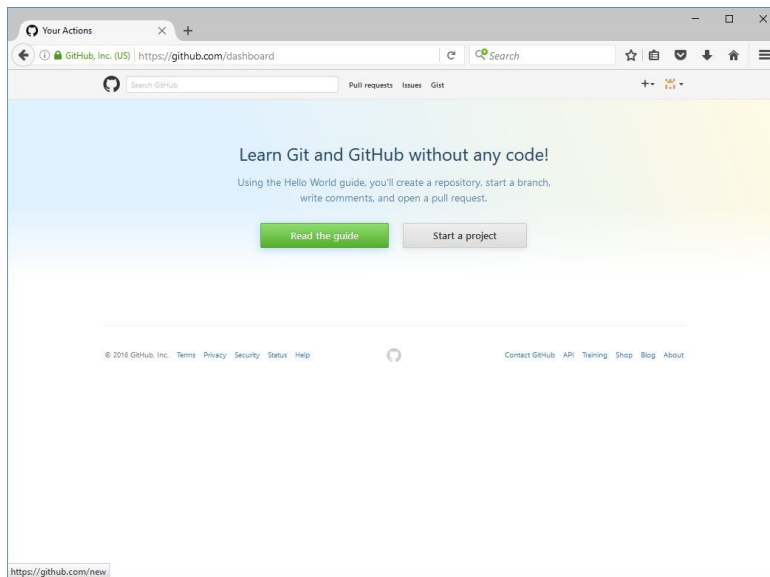
e.g. tutorials, android, ruby, web-development, machine-learning, open-source

Submit skip this step



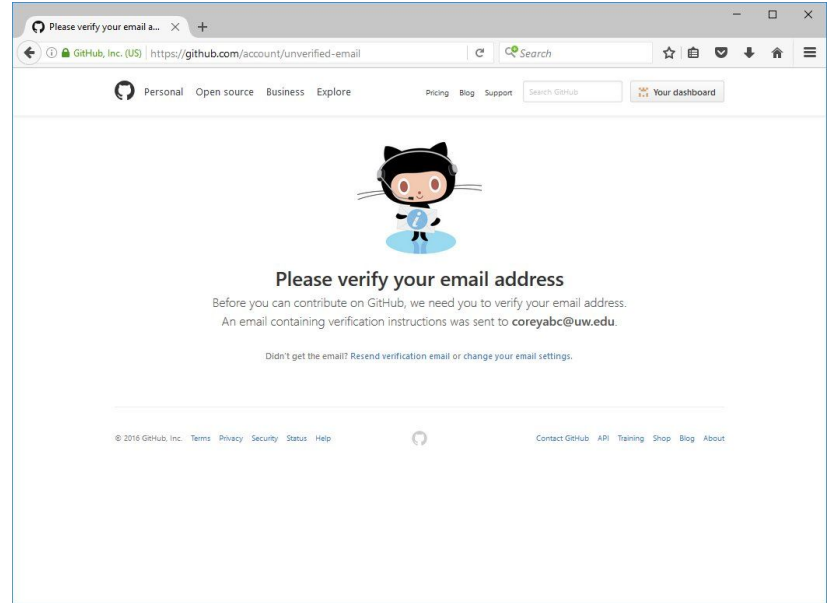
# Setting up Github

## 5. Click start a Project



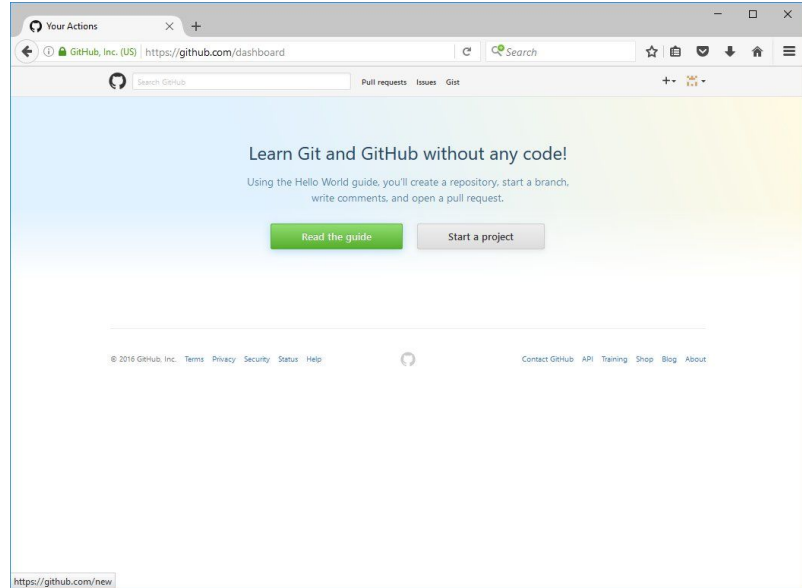
# Set up Github

6. Wait for the email to arrive and then click the link it sends you.



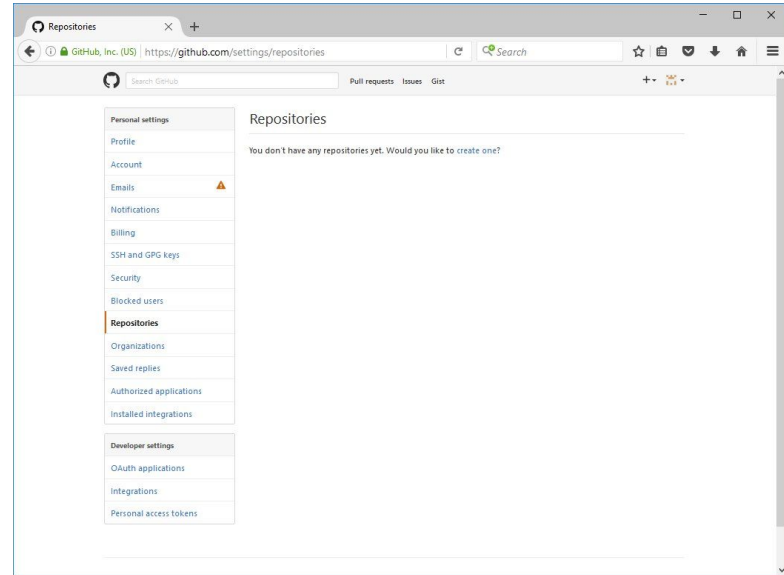
# Set up Github

7. Github should take you back to the Start  
A Project Page.



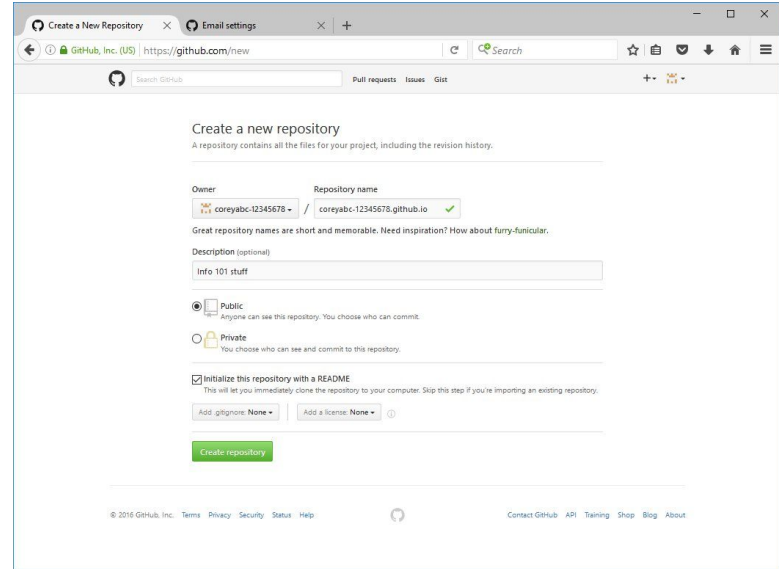
# Create a Github Repository

1. Click on Start a Project or find your way to the Repositories Page



# Create a Github Repository

2. Click “Create one” (but don’t fill in the info yet)



The screenshot shows the GitHub 'Create a new repository' page. The browser address bar displays 'https://github.com/new'. The page title is 'Create a new repository'. Below the title, a subtitle reads 'A repository contains all the files for your project, including the revision history.' The form includes fields for 'Owner' (set to 'coreyabc-12345678') and 'Repository name' (set to 'coreyabc-12345678.github.io' with a green checkmark). A description field contains 'Info 101 stuff'. The visibility is set to 'Public'. The 'Initialize this repository with a README' checkbox is checked. At the bottom, there are dropdowns for 'Add githgnore: None' and 'Add a license: None', followed by a green 'Create repository' button. The footer contains copyright information for 2016 GitHub, Inc. and links to Terms, Privacy, Security, Status, Help, Contact GitHub, API, Training, Shop, Blog, and About.

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: coreyabc-12345678 / Repository name: coreyabc-12345678.github.io ✓

Great repository names are short and memorable. Need inspiration? How about furry-funicular.

Description (optional): Info 101 stuff

☒ Public  
Anyone can see this repository. You choose who can commit.

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

☒ Initialize this repository with a README  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add githgnore: None Add a license: None

Create repository

© 2016 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub API Training Shop Blog About

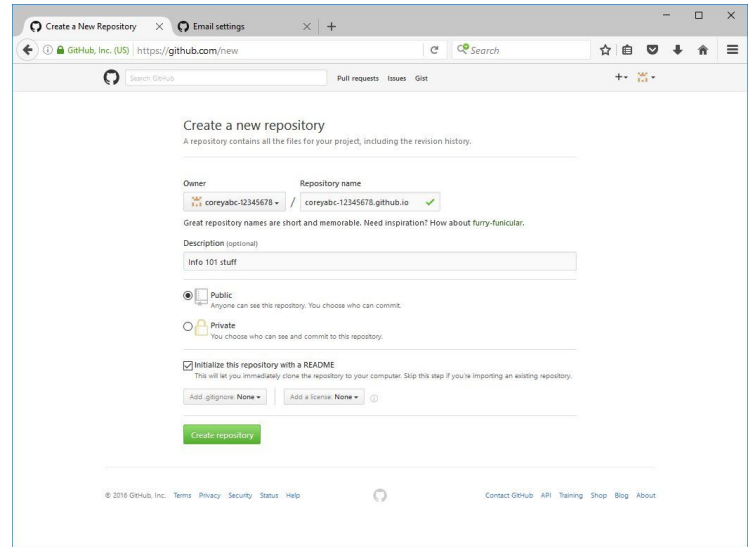
# Create a Github Repository

Be sure to name the repository with your username followed

by .github.io (for example, coreyabc-12345678.github.io)

Fill out the rest of the form as you see in the illustration below and

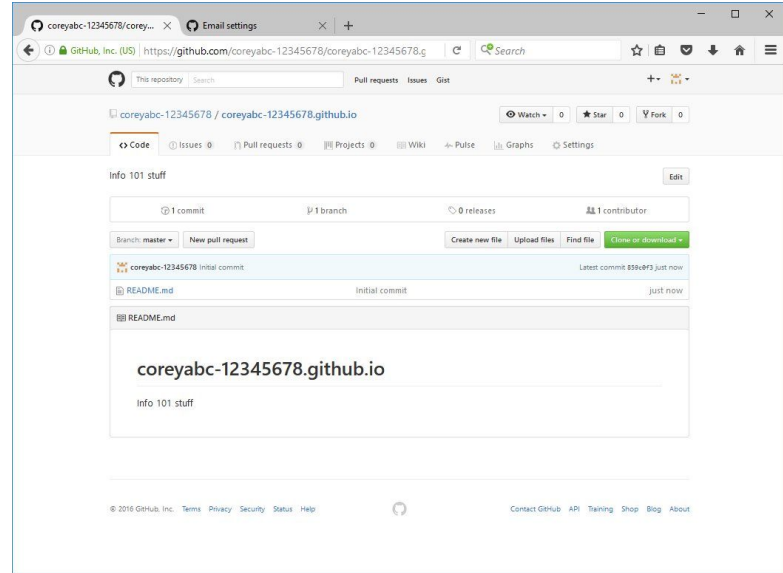
click Create Repository.



The screenshot shows the GitHub 'Create a new repository' page. The browser address bar shows 'https://github.com/new'. The page title is 'Create a new repository'. Below the title, it says 'A repository contains all the files for your project, including the revision history.' The form has two columns: 'Owner' and 'Repository name'. The 'Owner' dropdown is set to 'coreyabc-12345678' and the 'Repository name' dropdown is set to 'coreyabc-12345678.github.io'. Below these, there is a 'Description (optional)' field with the placeholder text 'Info 101 stuff'. The 'Public' radio button is selected, with the text 'Anyone can see this repository. You choose who can commit.' Below that, the 'Private' radio button is unselected, with the text 'You choose who can see and commit to this repository.' There is a checkbox labeled 'Initialize this repository with a README' which is checked. Below this, there are two dropdown menus: 'Add .gitignore: None' and 'Add a license: None'. At the bottom of the form is a green 'Create repository' button. The footer of the page shows '© 2016 GitHub, Inc.' and links for 'Terms', 'Privacy', 'Security', 'Status', 'Help', 'Contact GitHub', 'API', 'Training', 'Shop', 'Blog', and 'About'.

# Create a Github Repository

After pressing Create Repository you should land at a page that looks like this:



# Get an HTML Editor

If you are on a school computer, check that you have the programs below. If you are on your own computer download them as needed.

Download Notepad++ OR Visual Studio Code if you are on Windows

Download TextWrangler or Visual Studio Code if you are on OSX



# Make a Hello World Page

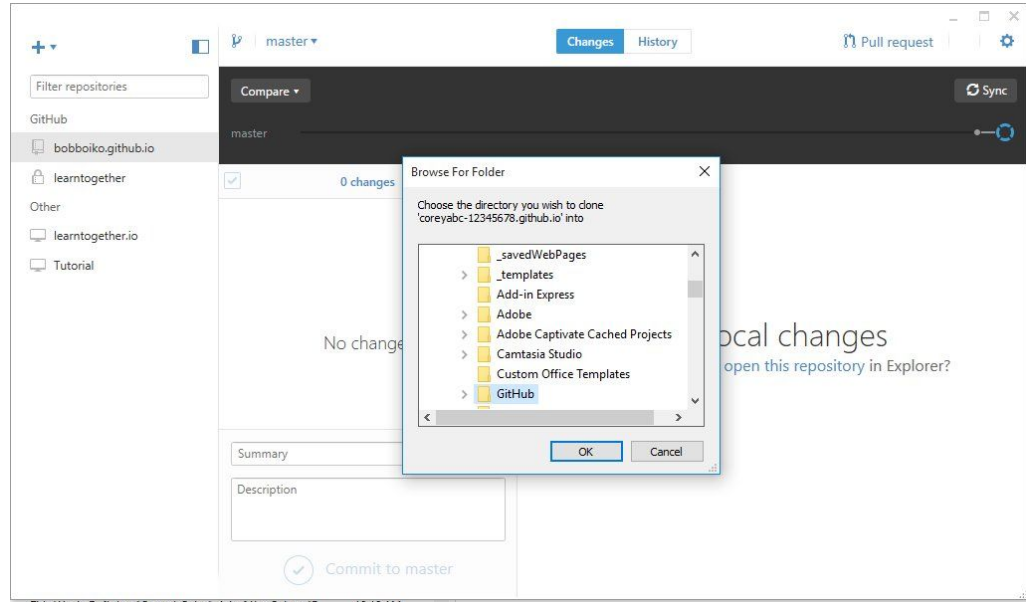
Following programmer tradition...

1. Open your text editor
2. Type in the following HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <h1>Hello World</h1>
  </body>
</html>
```

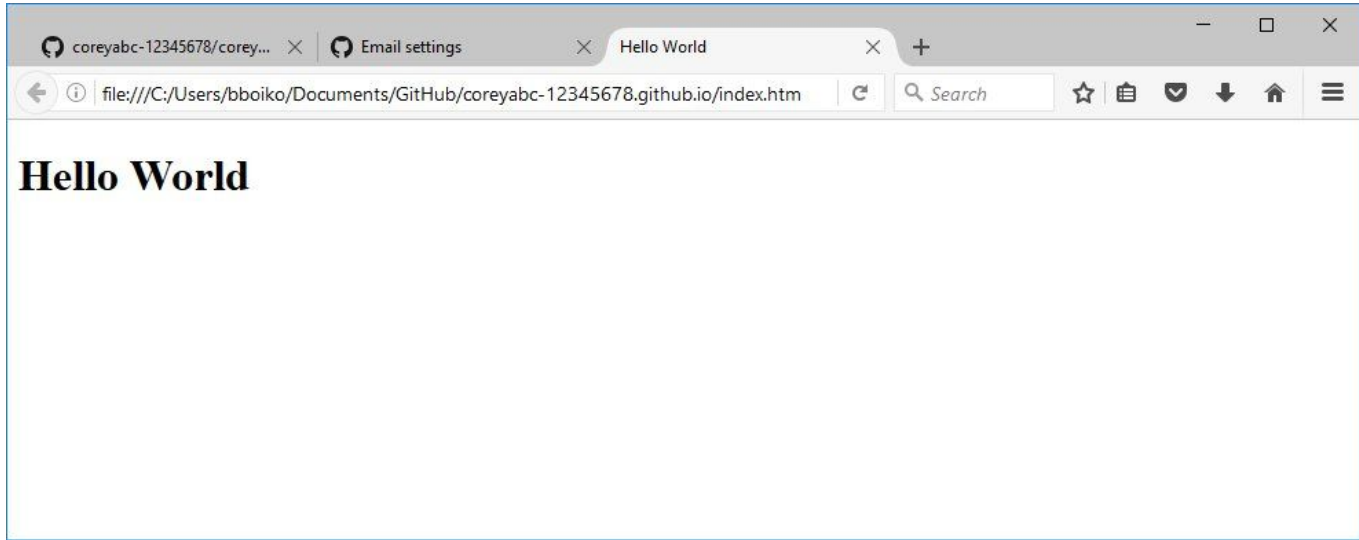
# Make a Hello World Page

3. Save your HTML file as index.html  
in the folder that you selected when  
you set up Github



# Make a Hello World Page

4. Open the file in a browser to make sure it looks like this:



# Upload your file to Github's web servers

Now you have a Web page and you have Github set up to put this and any other pages on the web for you.

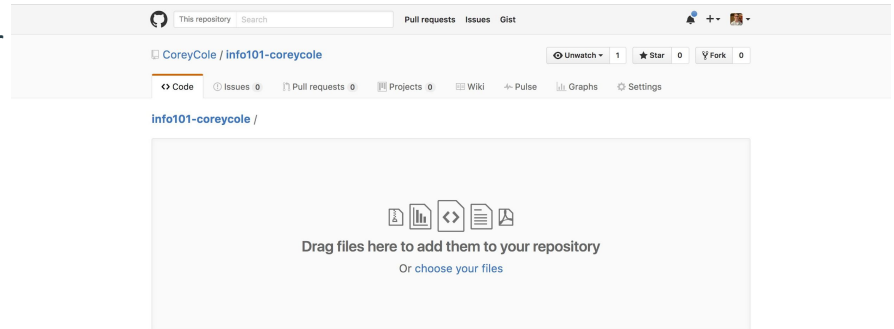
1. Go to <https://github.com> and navigate to the repository you created
2. Press the "Upload Files" button

# Upload your file to Github's web servers

3. Upload your index.html file.

Your first page should now be on the Web at [https://\[netId\]\[student number\].github.io/](https://[netId][student number].github.io/)

Don't forget to use your netId and student number



# Make Enhancements (Optional)

Improve your page or create other pages.

To work on your page

1. Make changes to your page in your text editor
2. Save the changes
3. Re-upload the changed file to github
4. Go back to [https://\[netId\]\[student number\].github.io/](https://[netId][student number].github.io/)

To create a new page

1. Create a new HTML file in your text editor
2. Upload the file to github.
3. Go back to Github, then commit and synch as before
4. Go to [https://\[netId\]\[student number\].github.io/\[new page name\]](https://[netId][student number].github.io/[new page name])

# Common Problems

1. If you are getting a 404 page when you click the link on gclass, make sure you gave your Github account and repository the correct names.
2. If you have made a change and Github has not updated, try a hard refresh to clear the cache (Hold Shift + click refresh). This can also fix 404 errors.
3. If your images/videos/audios are not showing up or working, make sure the file is saved in the same location as the HTML file on your computer, and that the HTML file specifies the correct file name(s) in the code.
4. If your images/videos/audios work on your local machine but not when pushed to GitHub, make sure the media files are uploaded to Github as well.

# Workshop: Embedding media

Learn how to embed features such as video and audio in your web page.

Some embeds are simply HTML tags that tell the browser to put something on the page

- `<img>` tag is one of the basic media tags
- In HTML 5 there is an `<audio>` tag that lets you embed a little audio player in your page. You can learn how to use it [here](#).
- Similarly there is a `<video>` tag that lets you embed a video on your page. You can learn how to use it [here](#)
- Flash and other similar stuff that the browser does not know how to render uses an `<object>` tag.



# Workshop: Embedding Media

`<iFrame>` is a way to insert someone else's web page inside yours.

You can find more information about iframes [here](#)

The `<iframe>` tag is an easy way to embed media from other websites.

- For Youtube videos, you just have to copy and paste the iFrame code from the “share” button of a video and click “Embed” to get the HTML code.

# Original Work: Show your embedding stuff

When you are satisfied with the files you have submitted, click the button below to certify that your files are uploaded.

Click to certify that all the above links Work

Here is what to create for the different original work levels.

- **Level 1: Newbie.** Show us a page that has:
  1. A Youtube video
  2. A google map
  3. One more thing that we did not demo in the workshop
- **Level 2: Apprentice.** Show us a page that has:
  1. A Youtube video that plays starting at minute 3
  2. A video that plays using the HTML5 video tag
  3. Something using the object tag
- **Level 3: Master.** Show us a page that has:
  1. A Youtube video that plays starting at minute 3 and another feature that you can't get from the basic embed code you paste from Youtube
  2. A video and audio file that plays using the HTML5 video and audio tags
  3. Something using the object tag that is not flash
  4. An RSS feed displayed

Toggle this on to let me know you are ready to be graded!

You can also toggle it off if you don't want me to grade.

Thank you