CS472 Lab W1D1  Create a homepage and publish it on a web server.

Today you will create a home page where you will put all your assignments for the first part of the course.

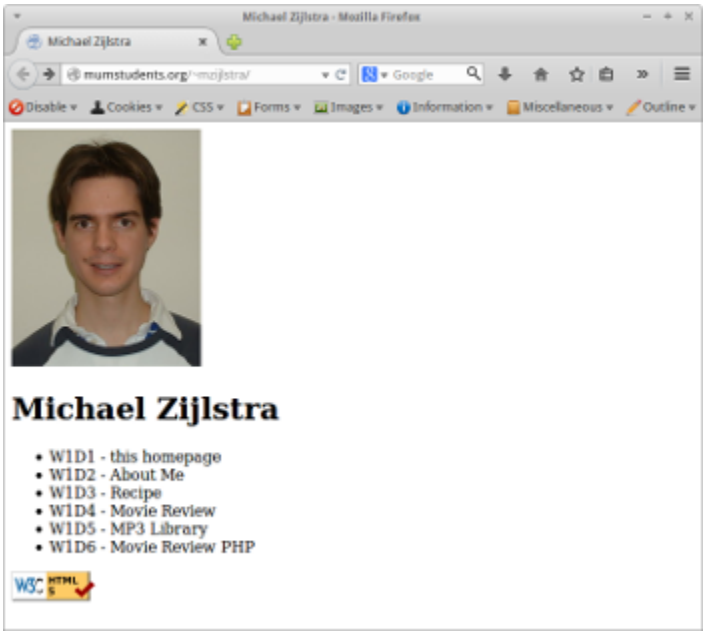
1. Create your homepage.
2. Create a github account and publish a test repository.
3. Create a repository for a github pages website and put your homepage there.
4. Validate your HTML
5. Submit your assignment on Sakai

# 1. Create your homepage

Your new homepage should have a couple of different items:

- Your name
- Your picture
- An unordered list with an item for each days assignment

Below is a basic example and a more sophisticated example.




# 2. Create a GitHub account and publish a test repository

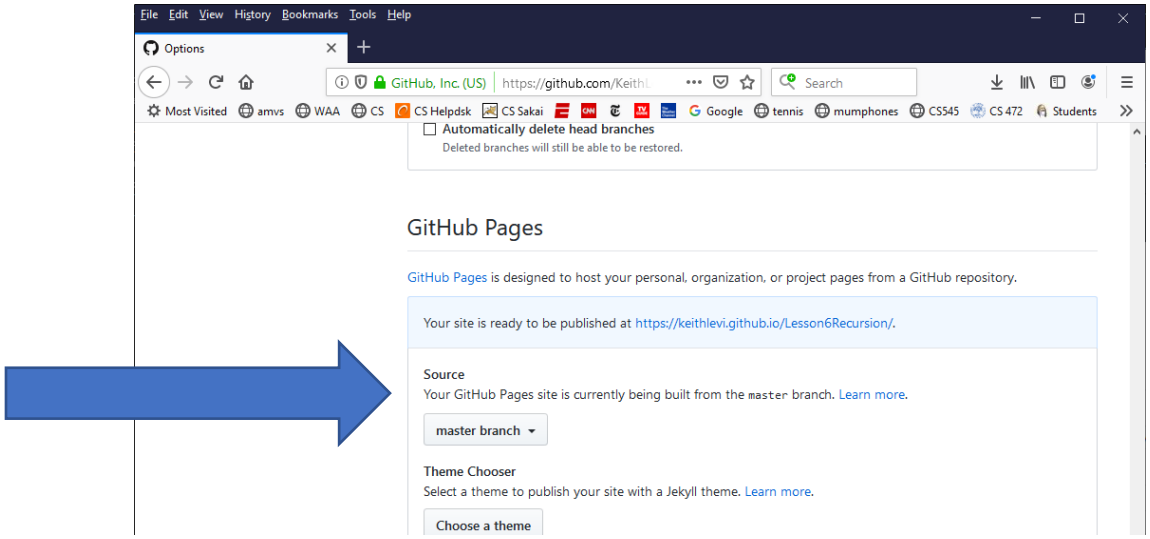
- Create a GitHub account if do not already have one
  - https://github.com
- Create repository for cs472
  - Download and install GitHub Desktop
    - https://desktop.github.com/
  - Login to your github.com account on the Github Desktop app
    - File > options > accounts
  - File > create new repository, e.g. 'myFirstRepo'
  - Publish the new repository (push to github.com)
- As a test add a simple textfile to your repository
  - Add a textfile, e.g., 'test.txt' to the folder that is the repository you created on your local file system, e.g. myFirstRepo.
  - In your Github Desktop application you will now see there is a change to the repository, the addition of the file test.txt
    - Commit the changes to the local repository (there is a commit button on Github Desktop)

        - After committing the change to the local disk repository there will be a button showing on Github Desktop that will push your changes to the github.com remote repository, 'push origin'. Click the button.
    - Go to your account on github.com and find the repository you created and published, myFirstRepo. Your test.txt file should be there.

# 3. Create a repository for a GitHub Pages website and put your homepage there.

GitHub Pages provides a simple web server to serve your html pages. You will have one website on the server with the domain name: [your user name].github.io. You will be able to put multiple projects on the site, with the url: https://[your user name].github.io/[repo name]. The instructions below should suffice for us. If you have any difficulties ask me or you can also look at https://pages.github.com/, which has quite good instructions.

a. first, create a new repository named *username*.github.io, here *username* is your username on github.com. You can do this on github.com or Github Desktop as described above.
    i. If the first part of the repository doesn't exactly match your username, it won't work, so make sure to get it right
    ii. Copy your homepage to the local file containing this new local repository. Name it 'index.html' so the web server identifies this as the homepage of your github pages site.
b. After creating the new repository find it on github.com and do the following
    i. settings > enable github pages (towards bottom of settings page)
    ii. if you see a message that says your site is currently being built from the master branch then you may need to make a push to the repository to get it to complete the build. This means make any small change to your index.html page and commit/push the changes to the repository.

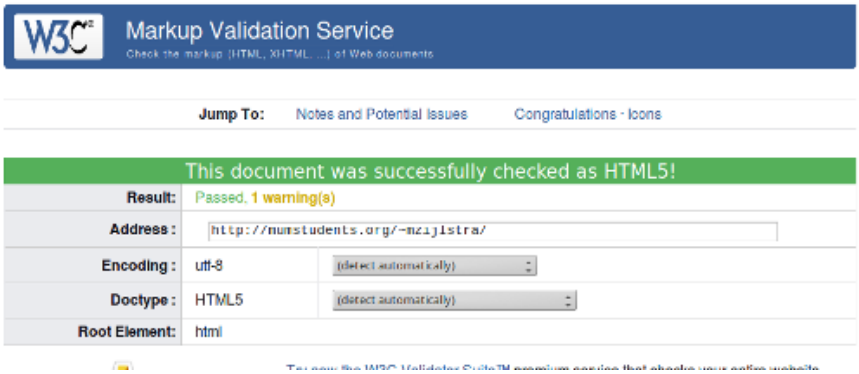



C. If this is done successfully then you will see a message that you can view your page https://[your user name]].github.io

## 4. Validate your HTML.

At the bottom of your page, add the following image link. Doing so will allow you to validate that there are no errors in your HTML (we will talk about this more tomorrow) `<a href="http://validator.w3.org/check/referer"> <img src="http://mumstudents.org/cs472/2014-09/images/w3c-html.png" alt="html validator"/></a>`

This link will work when you click it from a page in a browser that was loaded from a web server. It will not work if the page came from local file. If everything goes well you should see a page something like this:




## 5. Submit to Sakai.

When you're done, login to Sakai and submit the url to your home page along with your description of how things went as the homework for today for example: Hi Professor, I completed the homework for today; you can see the results at: http://myname.github.io . It took me about half an hour, I had some

trouble figuring out how the img src attribute worked, but everything worked in the end. The assignment is complete except that I still have a syntax error in the HTML validator report.

The comment is very important. To get full credit for the lab you must include

- a descriptive comment about something of interest in the lab,
- a status description of whether the lab is complete or not,
- the URL to your home page.

It is important to submit this comment by 9:30PM of the assignment due date. Even if the assignment is not complete if you submit the comment as described you will get almost full credit. In general, I will provide solutions that you can check the following morning. It is best to get to bed on time even if the assignment is not 100% complete. Be sure to take time to do the readings and answer the study questions. If you run out of time for the entire lab then check the solution the next day.