**BUA 235 lab #4: hosting your website on an Internet web server**

In the last three labs, you learned how to create a basic HTML web site, and get started adding visual style with CSS. Naturally those tutorials didn’t do everything for you. You’ll have to spend a lot of time and trial and error to get the web site looking the way you want. Search engines are your friend as you’re looking for how-to advice.

So far, you’ve been working on these files on your personal computer. In order for other people to see the website on the Internet, you’ll now need to upload those files to a web server. This service is called “web hosting” and there are a variety of ways to obtain it.

Three main ways this is done in industry:

1. Rent traditional web hosting service from HostGator, GoDaddy, or another hosting provider. You would upload files using an FTP program like FileZilla. The cost is under $5/month and they make it easy to register a custom domain name. *This is the easiest approach*.
2. Set up a server on a cloud platform like Heroku or Amazon Web Services, and configure your own server. *This gives you lots of control and can be cheaper, but is the most difficult approach*.
3. Set up a web server on a computer you own, and plug it in to an internet connection with a dedicated IP address. *Only do this if you’re really interested in the technology, and have the internet connection for free*.

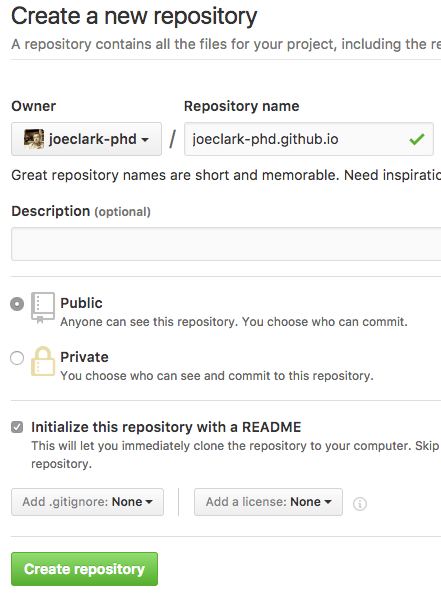
In this lab, I’m going to demonstrate an alternative using GitHub. GitHub provides free hosting for static websites for their users, at URLs like http://*username*.github.io.

I have chosen GitHub as the hosting method for your personal website assignment both because it is free and because it allows me to see your code in case I need to give you help, or partial credit. However, the GitHub option has a couple of limitations: (1) it doesn’t host dynamic application code or databases, and (2) you have to have a GitHub account.

**Getting Started**

**1.** First, go to <http://www.github.com> and create an account, if you don’t already have one. GitHub is a site where programmers share and show off their code; the account is free. To protect your privacy you don’t have to use your real name if you don’t want to.

**2. Create a repository.**  Each GitHub account contains any number of “repositories” of code; typically one per project. You can take a look at some of mine at <https://github.com/joeclark-phd> or see some public open-source projects by going to <https://github.com/explore>.

Create a new repository for your website’s code by going to the “+” sign at the top right corner of the page and selecting “New Repository” from the menu.

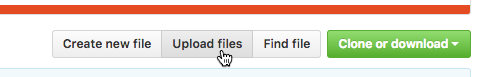
To take advantage of the free web hosting, you must name it *username.github.io*, with your own username, of course.

Check the box saying “Initialize this repository with a README”. If it’s not initialized, you’ll have to create the initial files later using the Git program, which is outside of the scope of this lesson. *For our purposes: if you create the repository without this, go to “settings” and delete the repository, then create it again.*

The other options can be left alone.

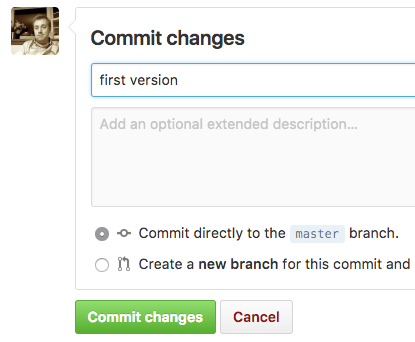
Click “Create repository”.

**3. Upload files manually.** GitHub is meant to work with a program called Git, which programmers use to keep track of changes to their code. However, we can ignore that for now and just use the “Upload files” button while viewing the repository.



Drag the files for your website (including the HTML, CSS, and image files) onto the screen where it says “Drag files here”, or click the “choose your files” link and select files using the pop-up.

Then scroll down. Each version of the code in GitHub is called a “commit”, and you may leave a comment that says what changed from the previous version. For example, since this is your first commit, you can label it “first commit” or something similar. (Later, you’ll be able to backtrack through the history of changes, so using good labels is helpful. But optional.)



Finally click “Commit changes”.

When you make future updates or changes to your files, re-upload them the same way.

**4. View the website at http://*username*.github.io.**  Your username, of course. This works if your main file is called “index.html”, which is the default file that a webserver looks for. If your file is named something else, like “home.html”, add the file name to the end of the URL like so: **http://*username*.github.io/*filename***

For the assignment due on October 13, you will submit to me the website’s URL for grading according to the assignment guidelines. Don’t plan on submitting your code files directly to me.

**Extras**

For more information about GitHub’s web hosting service called GitHub Pages, see this URL: <https://pages.github.com/>. At that link, you’ll find out how to add a custom domain name (i.e., www.*yourname*.com), add a Jekyll blog, or set some other features like custom 404 pages.