

Your website address will be **username.github.io**. For this reason, you might want to give a bit of thought to what makes the most sense for a username. If you want to choose a new username the sooner you do it the better since you are going to want to spread links to this on all of your other online profiles.

One of the biggest benefits to having a website is that you can control your image online. This is particularly important as you apply to postdocs/faculty positions – pretty much anything you apply to one of the first things that people will do is search for you on google. To increase your ranking in google search returns you want to include links to your homepage on all online profile pages that you have. I keep a list of everywhere that my website is linked from so that I can update them if I ever change my website address:

- Google scholar
- LinkedIn
- Scholastica
- Pubfacts
- impactstory
- FigShare
- Github
- mendeley
- Research gate
- Departmental and PI websites

By changing all of these to point at the same place it is a strong signal to the ranking algorithms that your homepage is a good place to send someone who has searched your name and your site will quickly rise through the ranks to be the top return when someone searches your name.

The Basics

- 1) Go to github and create a new repository. The repository must be named **username.github.io**.
- 2) Go to the settings tab and scroll down to the github pages section and select a theme.
- 3) At this point you could simply use the web interface and edit the index.md file to act as your website. In fact, lets go to **username.github.io** and take a look at what that would look like.
- 4) Go back to the repository on the server and try changing some text. Once you have made a change go ahead and commit the change at the bottom of the screen. Once you have done this you can go back to the website that is publicly available and after 30-60 seconds you should be able to refresh the page and see the changed text.
- 5) This approach of using markdown to serve your site is very simple but also has some serious limitations. Most of you will likely want to do something a bit more fancy so we will spend the rest of the day looking at how you might set up a stylistically more impressive and flexible web presence using a bootstrap template. One big advantage here is that you will have a website that will look good across many different monitor resolutions and devices.

Bootstrap Approach

The first thing we are going to want to do is get ourselves setup with a local copy of our repository in this example I will assume you are using github desktop app to do this though you can use command line git tools to the same effect if you prefer.

- 1) download and install github desktop app.
- 2) open the github desktop app and link it to your github account and clone your repository onto your computer locally.
- 3) choose and install an editor for your html documents (for Mac I recommend brackets, when I'm on a linux system I use gedit) you can use any file that edits plain text files but one designed for html editing will make the coding a bit easier.
- 4) go to your repository folder on your computer and open the index.md file. Make a couple of simple changes to it and then save the file
- 5) Go back to the github desktop app and you should see that you have 1 file that has changed and on the left you can see the actual lines that you edited. Lets commit this change and the push.
- 6) After a minute or so these changes should be visible to anyone who visits your website (make sure you see the changes). If so you have successfully set up your repository and linked the one in the cloud to the one on your computer.
- 7) Lets pick out a template website that we will build our websites from. Here is a good source we will use today. **bootstrapmade.com**. Another option is **startbootstrap.com**. I am going to use the iPortfolio template and so I will click on download for that one.
- 8) Delete both of the existing files in your repository. Unzip your downloaded template and move all of the files it contains into your repository.
- 9) Go back to github desktop app and commit and push the changes that you have made. After pushing give it 60 seconds or so and check out your website in the web browser.

The fun part!

Now we begin editing our website to make it our own. You don't need a lot of coding knowledge for this and the website <https://www.w3schools.com/> can be super helpful.

10) Open up the index.html page in your html editor (or text editor). Below is a list of things that I would want to change and we can spend the rest of time looking at how we change each one of them.

- page title look for the tag `<title>`
- favicon <https://favicon.io/>

- pictures
- navigation bar buttons (<https://boxicons.com/cheatsheet/>)
- twitter timeline
- **google scholar link** - This is the first thing I often look for make it easy for your applicant reviewers
- CV include a PDF

Googling and looking at W3 school pages can let figure out most things.

- setup google analytics for your webpage