

Github Pages

Hunter Jorgensen

Intro

Hosting a website typically requires a server, something that can respond to requests with the contents of a website. While setting up a server is pretty easy, we're going to use a functionality of Github known as Github Pages.

Github Pages is an easy to setup system to serve any CSS, HTML or images that we would want, without needing to manage or host a server. Best of all, it's free!

Also, I'm assuming you've already setup your Github integration with your local machine.

Hint: this will be helpful in your first assignment.

Setup: Personal Github Account

If you are using the Github account provided by NEU (or any Enterprise Github account), please skip forward a few slides.

Create a new repo with the following name: “*{username}*.github.io” (ignore the quotes and brackets and make sure that the username is your Github username

Keep this repo public

Setup: Personal Github Account

```
git clone https://github.com/{username}/{username}.github.io
```

```
cd {username}.github.io
```

```
echo "Hello World" > index.html
```

```
git add *
```

```
git commit -m "Initial commit"
```

```
git push -u origin master
```

Open a browser and go to “<https://{username}.github.io/>” (this can sometimes take a minute to propagate)

Setup: Enterprise (NEU) Github Account

create repo (can be named whatever you want, but copy it somewhere!)

```
git clone https://github.ccs.neu.edu/{username}/{repo name}
```

```
cd {repo name}
```

```
echo "Hello World" > index.html
```

```
git add *
```

```
git commit -m "Initial commit"
```

```
git push -u origin master
```

Setup: Enterprise (NEU) Github Account

In your browser, go to: <https://github.ccs.neu.edu/{username}/{repo name}> (the page for your repo)

click settings

scroll down to "Github Pages"

set "Source" to "master" (page will reset)

go to <https://pages.github.ccs.neu.edu/{username}/{repo name}/> (this can sometimes take a minute or 2 to update)

Setup: Adding a 2nd Page (Personal or Enterprise)

In your local repo, *mkdir help //* or however you create a new folder

```
echo "Help me" > help/index.html
```

```
git add --all
```

```
git commit -m "Adding simple help page"
```

```
git push
```

In your browser go to *{username}.github.io/help* or

https://pages.github.ccs.neu.edu/{username}/{repo name}/help

The magic of index.html

We can think a website or a server as a set of directories. (i.e., '/', '/help/', etc.) If it helps, you can think of this as opening a folder on your laptop and accessing all the contents inside (i.e., clicking on a folder called 'help').

For most web servers, the index.html acts as the default resource to load. So, when we went to the '/help' route, the server automatically returns the index.html file, even though we can potentially have other files there.

If this isn't clear, that's ok! We'll return to this concept more in module 3.