UPLOADING A PAGE TO GITHUB

Host your web site on GitHub

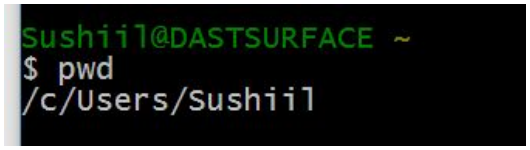
“Deployment” is the process of moving your code from your computer to a web server where anyone can access it with a browser. Follow along with these steps to deploy the web site that your team built.

1. The pair in your team that is working on the *home* page of your website MUST name it *index.html*. If this is not the case, they should rename it now. *Note to all team members: Anytime you change the filename of your HTML files, you must update the links(<a>) in your code so they reference the new filename.*

2. Make sure the file is saved into a folder named “Projects” in your your home directory.

3. Open up Terminal (Mac/Linux) or GitBash (Windows).

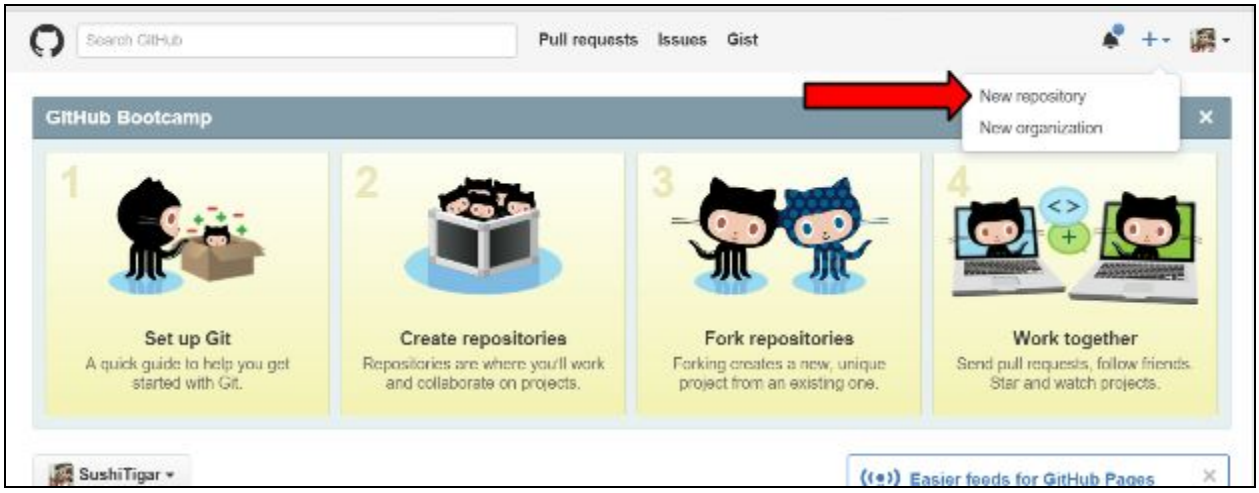
4. Type **pwd** and hit enter to verify that you are in your home directory. It should show something like this:



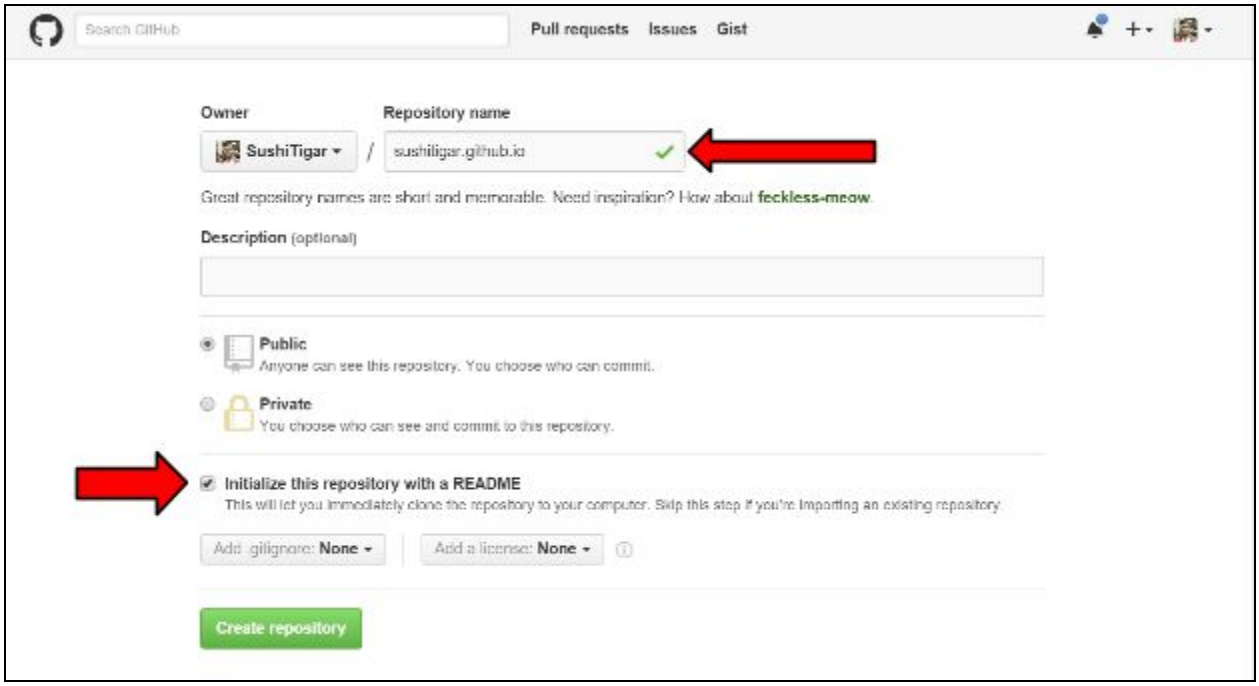
5. Now type **cd Projects** (if you named your project folder something different, change that name here) and hit enter. The “cd” command tells the system to move to a different folder. “Projects”, in this case, is the folder that you want to move to.

UPLOADING A PAGE TO GITHUB

6. Now open up your web browser and go to github.com. Login if you haven't already. Select the **+** icon in the top right and then select **New repository**.



7. In the field named "Repository name", type in ***username*.github.io** replacing "username" with your GitHub username. In this example below, my username is "sushitigar" so I typed "sushitigar.github.io".



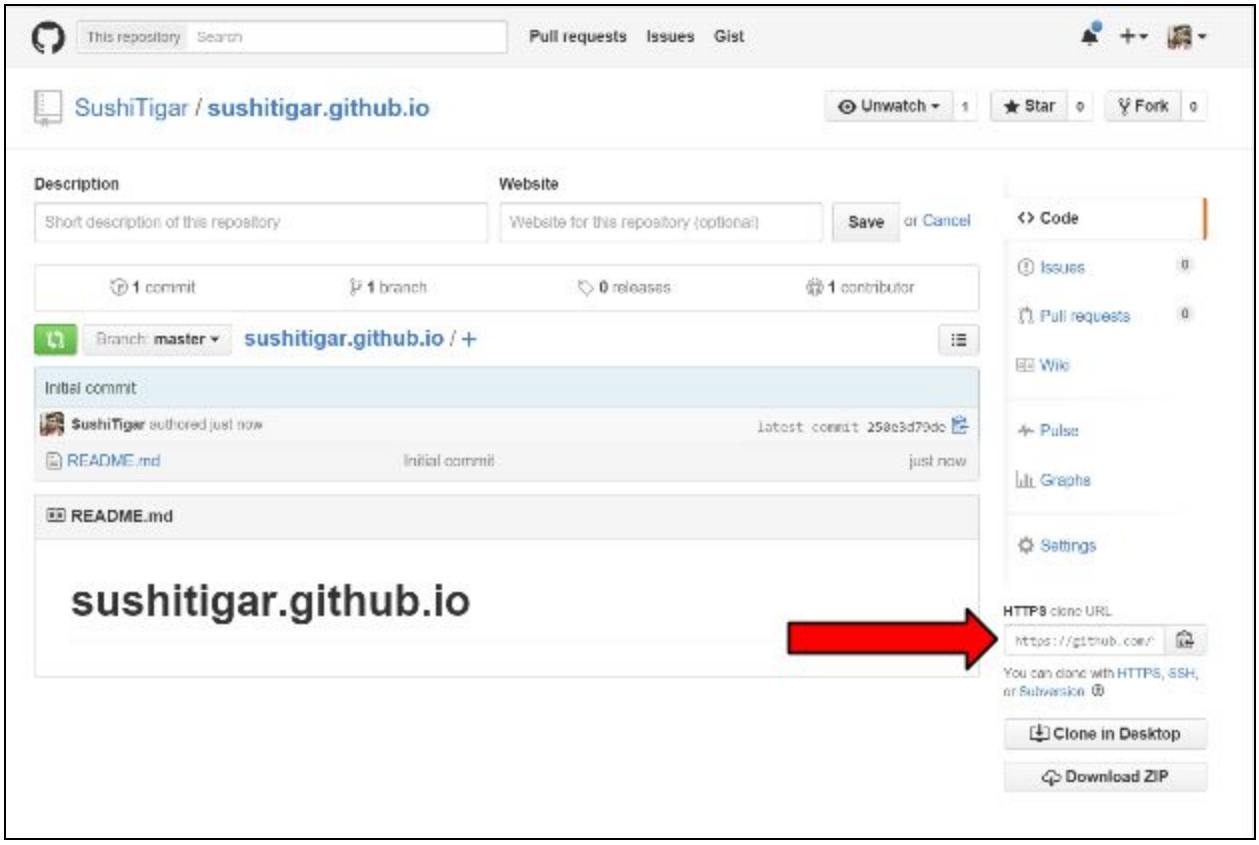
8. Ensure that you are creating "Public" repo. Add a Description if you like.

9. **IMPORTANT** select the checkbox next to **Initialize this repository with a README**.

10. Finally, select the green **Create repository** button at the bottom.

UPLOADING A PAGE TO GITHUB

11. On this next page, on the bottom of the right column (pictured below), highlight and copy the text in the field named **HTTPS clone URL**.



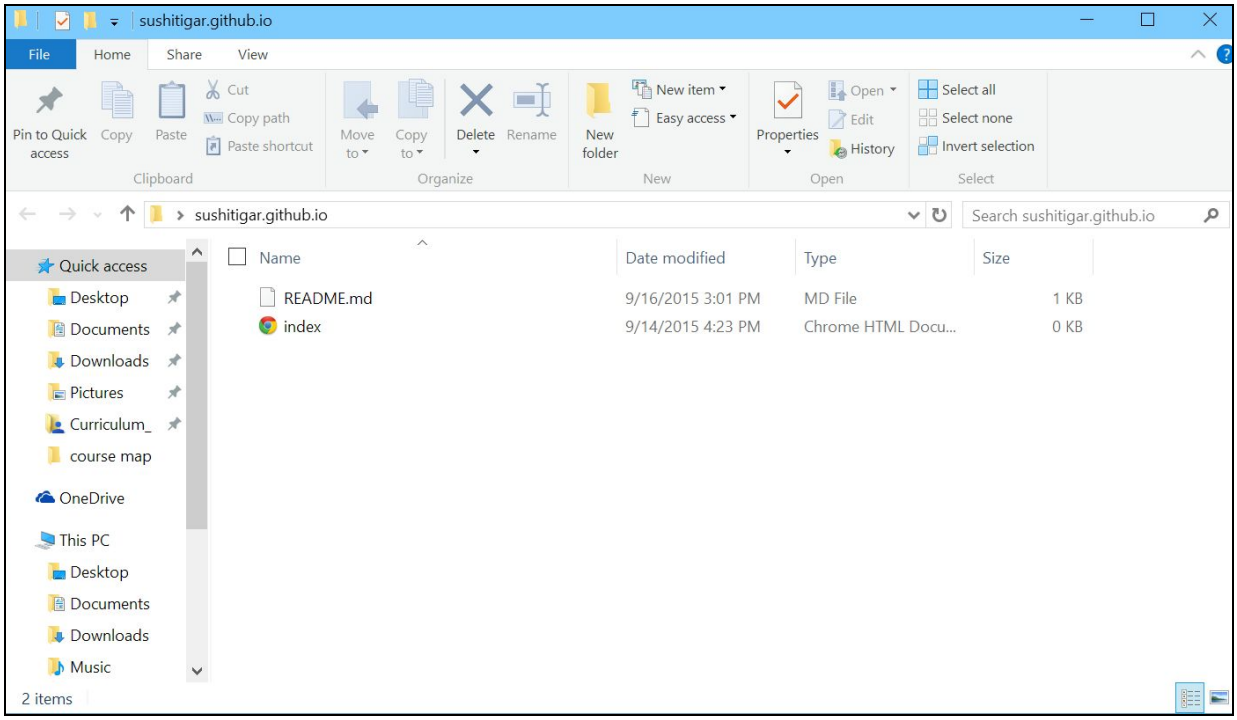
12. Now go back to your Terminal or Git Bash program. Before you go further, you need to know how to paste text into your terminal:
    a. Git Bash on Windows: right-click on the title bar, select **Edit,** and then select **Paste**.
    b. Mac: **Cmd-V**.
    c. Linux: press **Ctrl-Shift-V**.

13. Next you're going to **clone** that repository you just made on GitHub onto your computer. When you clone something, you are making a copy of that repository ("repo") on your computer, and linking your computer's repo to the one on GitHub. To clone, type **git clone** into your terminal, then paste in the link you copied from GitHub. It should look like this: (replacing "username" with your username)
    git clone https://github.com/*username*/*username*.github.io.git.
    Hit enter and you should get a message back similar to the image on the next page.

UPLOADING A PAGE TO GITHUB

```
Sushiil@DASTSURFACE ~/Projects/sushitigar.github.io (master)
$ git clone https://github.com/SushiTigar/sushitigar.github.io.git
Cloning into 'sushitigar.github.io'...
remote: Counting objects: 15, done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 15 (delta 5), reused 7 (delta 0), pack-reused 0
Unpacking objects: 100% (15/15), done.
Checking connectivity... done.
```

14. Now if you look in your Projects folder, there should be a new folder there named **username.github.io** (“username” being your GitHub username). Copy your index.html file from your Projects folder into the new “username.github.io” folder.



15. Now type **cd *username*.github.io** (As always, replace “username” with your username) and then hit enter. To make typing easier, start typing your username, then hit **Tab**; if the folder is there, the terminal will fill in the rest of the name for you.

```
Sushiil@DASTSURFACE ~/Projects
$ cd sushitigar.github.io
```

16. Type **git status**. Git will show you what files on your computer have changed since you last made a **commit**. Commits are a snapshot of what your files look like at a specific point in time. In this case, you haven’t done one yet, so it will tell you what’s changed since you cloned the repo. Here, it’s telling you that you’ve added the index.html file.

UPLOADING A PAGE TO GITHUB

```
Sushiil@DASTSURFACE ~/Projects/sushitigar.github.io (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.

Changes not staged for commit:
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        index.html

no changes added to commit (use "git add" and/or "git commit -a")
```

17. Now type **git add index.html** and hit enter. This tells git that you're going to want to take a snapshot of this file soon. Repeat the **git add** command for every other file indicated by Git (in blue) as "untracked".

```
Sushiil@DASTSURFACE ~/Projects/sushitigar.github.io (master)
$ git add index.html
```

18. Next, type **git commit -m "first commit"** and hit enter. **commit** tells git to take the snapshot. The **-m** bit tells it that you want to save a message with that snapshot (think of it like a caption). The text in quotes is the **commit message** (the caption). You should see a response back similar to below.

```
Sushiil@DASTSURFACE ~/Projects/sushitigar.github.io (master)
$ git commit -m "first commit"
[master 2aca5b8] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 index.html
```

19. Now type **git push origin master** and hit enter. You'll get asked for your GitHub username and password, and then it will give you a response similar to the one below. **git push** sends your code to GitHub, and will make your GitHub repo have the same files, with the same changes, as the commit you just made.

```
Sushiil@DASTSURFACE ~/Projects/sushitigar.github.io (master)
$ git push -u origin master
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 283 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/SushiTigar/sushitigar.github.io.git
   d99d577..2aca5b8  master -> master
Branch master set up to track remote branch master from origin.
```

20. If everything went right, you can enter this address in your browser and see your website, live on the Internet!

    (Replace "username" with your username)
    http://*username*.github.io

    Share this address with your friends and family and they'll be able to see your handiwork too.