

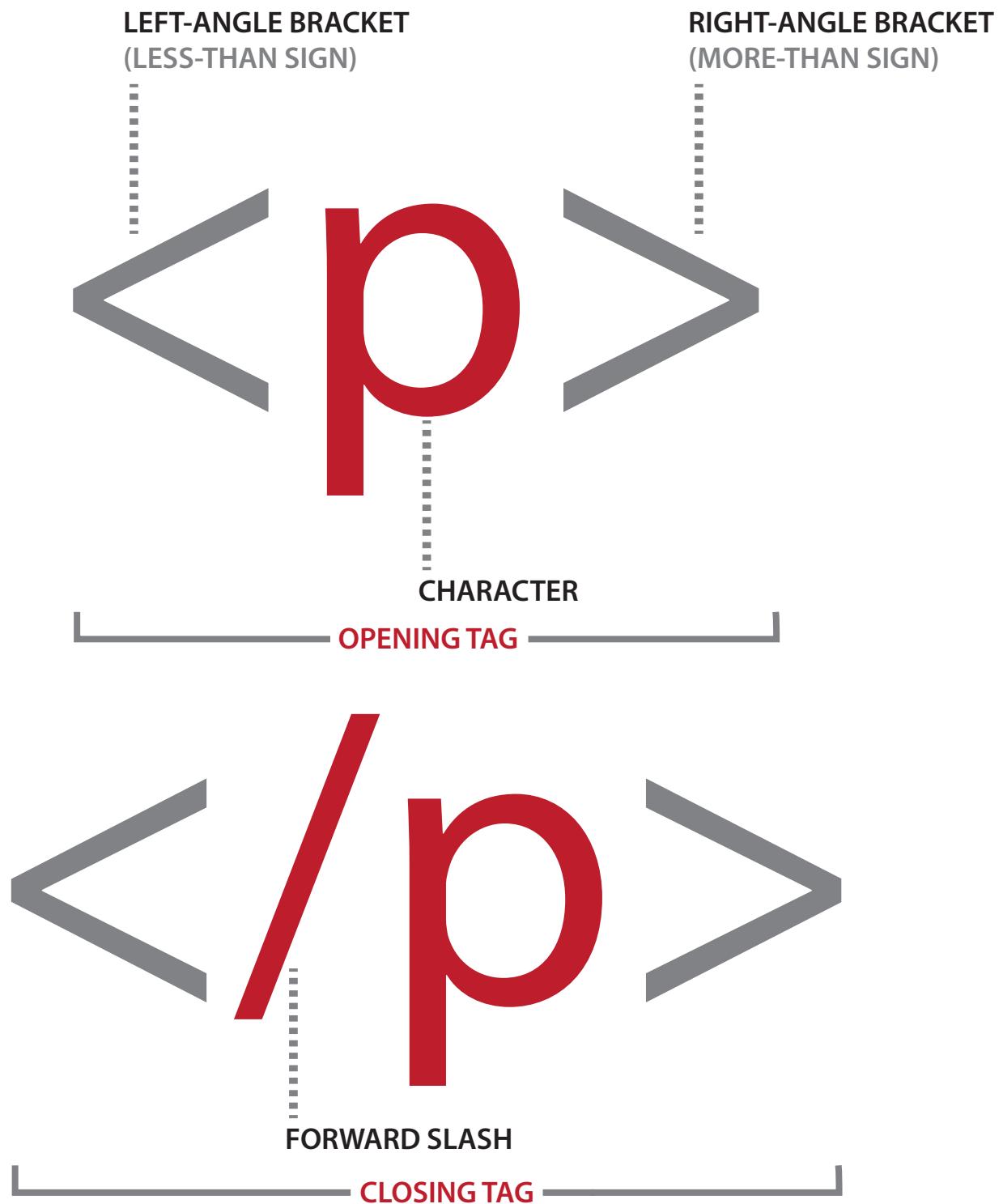


# **WELCOME TO CODE FELLOWS**

Code 101

Intro to Software Development  
& Careers in Tech

# TAG ANATOMY

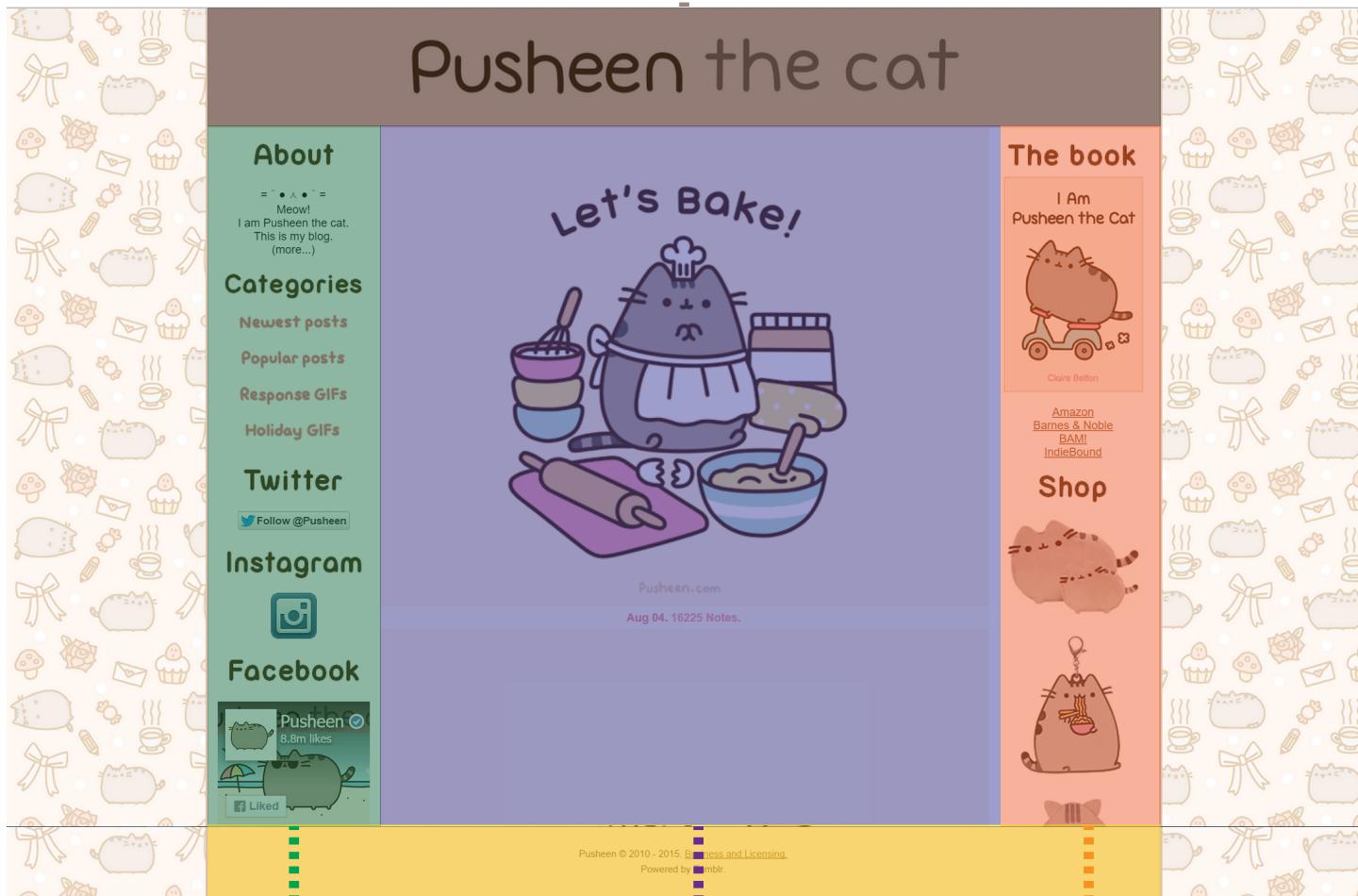


The **character** in each bracket indicates the tag's purpose.  
The content is written between the two tags.

# HTML SITE ANATOMY

## HEADER

<header></header>



## NAV

<nav></nav>

## FOOTER

<footer></footer>

## SECTION

<section></section>

## ASIDE

<aside></aside>

# HTML SITE ANATOMY

## HEADER

```
<header></header>
```

## Earthquake Preparedness

### LIST

```
<ul><li></li></ul>
```

LINK

```
<a href="url">
</a>
```

### IMAGE

```

```



### PARAGRAPH

```
<p></p>
```

#### North America

Look around places where you spend time. Identify safe place such as under a sturdy piece of furniture or against an interior wall in your home, office or school so that when the shaking starts, you Drop to the ground, Cover your head and neck with your arms, and if a safer place is nearby, crawl to it and Hold On.

When the shaking stops, look around. If there is a clear path to safety, leave the building and go to an open space away from damaged areas.

© Your Name Here 2015

## SECTION

```
<section></section>
```

## ARTICLE

```
<article></article>
```

# HTML CHEAT SHEET

<head></head>

This tag contains content **ABOUT** the page like the **title**.

<title></title>

This tag contains the title that shows up in the browser title bar.

<body></body>

This tag contains content shown **ON** the page like a **paragraph**.

<p></p>

Surround each paragraph of content with this tag.

<a href="url"></a>

Create links! The content between the tags is the display text.



This tag inserts an image. It doesn't need to be closed.

<strong></strong>

This tag **bolds** the content between it.

<br />

This tag creates a line break. It does not require a closing tag.

<h1></h1>

Use this tag with a number 1-6 to define a heading. <h1> is the largest and <h6> the smallest.

# CSS SITE ANATOMY

```
header {  
    color: white;  
    background-color: black;  
    text-align: center;  
    font-size: 20px;  
    display: block;  
}
```

```
body {  
    width: 800px;  
}
```

## Earthquake Preparedness

[North America](#)  
[East Asia](#)  
[Southeast Asia](#)



### North America

Look around places where you spend time. Identify safe place such as under a sturdy piece of furniture or against an interior wall in your home, office or school so that when the shaking starts, you Drop to the ground, Cover your head and neck with your arms, and if a safer place is nearby, crawl to it and Hold On.

When the shaking stops, look around. If there is a clear path to safety, leave the building and go to an open space away from damaged areas.

```
nav {  
    background-color: #eeeeee;  
    width: 125px;  
    height: 600px;  
    display: inline-block;  
    font-size: 18px;  
}
```

```
section {  
    width: 640px;  
    display: inline-block;  
    padding: 10px;  
}
```

© Your Name Here 2015

```
footer {  
    width: 800px;  
    background-color: black;  
    color: white;  
    display: block;  
    text-align: center;  
}
```

# CSS CHEAT SHEET

**width**

The content area's width (within the padding, border, and margin)

**margin**

The margin of the element on each of its four sides.

**padding**

The padding of the element on each of its four sides.

**background-color**

Set the background color of an element with a value or keyword.

**color**

Set the text color of an element with a value or keyword.

**font-size**

Specify the size of the font using a value or keyword.

**text-align**

Set the horizontal alignment of text using a keyword.

**vertical-align**

Set the vertical alignment of text using a keyword.

**display**

The way an element is rendered on screen. This setting can affect the position of other elements.

# GIT CHEAT SHEET

## Navigating the Terminal

**cd folder-name**

Type `cd` folder-name to navigate to a folder.<sup>1</sup> Type `cd ..` to go to the parent folder.

**ls -la**

Display ALL (including hidden) files in the folder. `-la` is a flag. Flags modify commands.<sup>2</sup>

## Using Git

**git clone url**

Replace `url` with the url of a repo you want to clone to your computer.

**git add --all**

This command will stage ALL untracked files.

**git status**

It shows which updated files are untracked (not git added) and which are staged.

**git commit -m "msg"**

Commit ALL staged files. Replace `msg` with a descriptive message.

**git push origin master**

This command will upload your commit to GitHub.

<sup>1</sup> Start typing the folder name and hit tab to auto-complete.

<sup>2</sup> Flag letters are abbreviations. `l` stands for list and `a` stands for all. You can mix and match these freely. Try `ls`, `ls -a`, and `ls -l` to see the differences.

# 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:

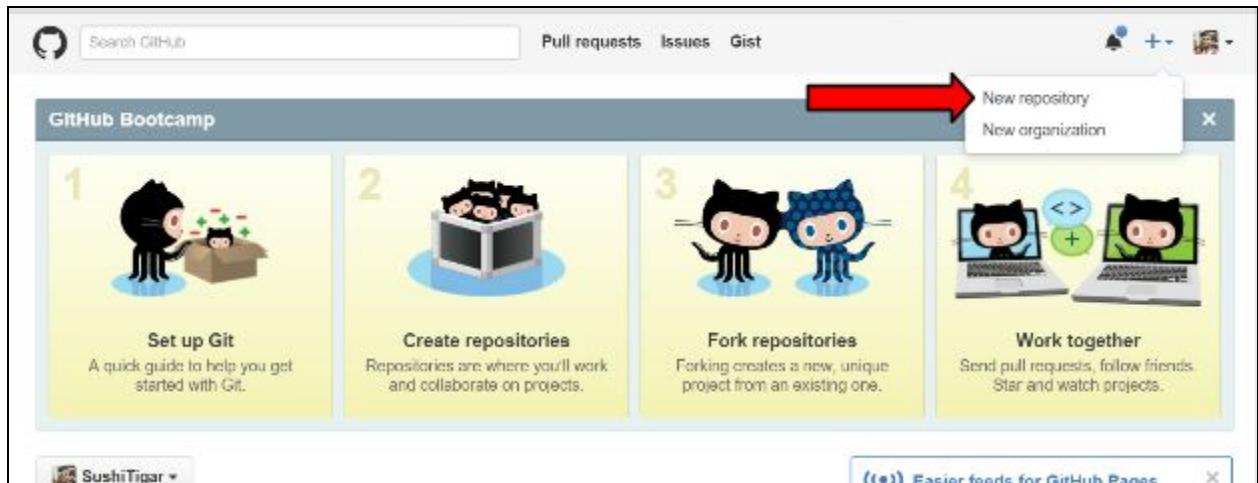
```
Sushiil@DASTSURFACE ~
$ pwd
/c/Users/sushiil
```

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.

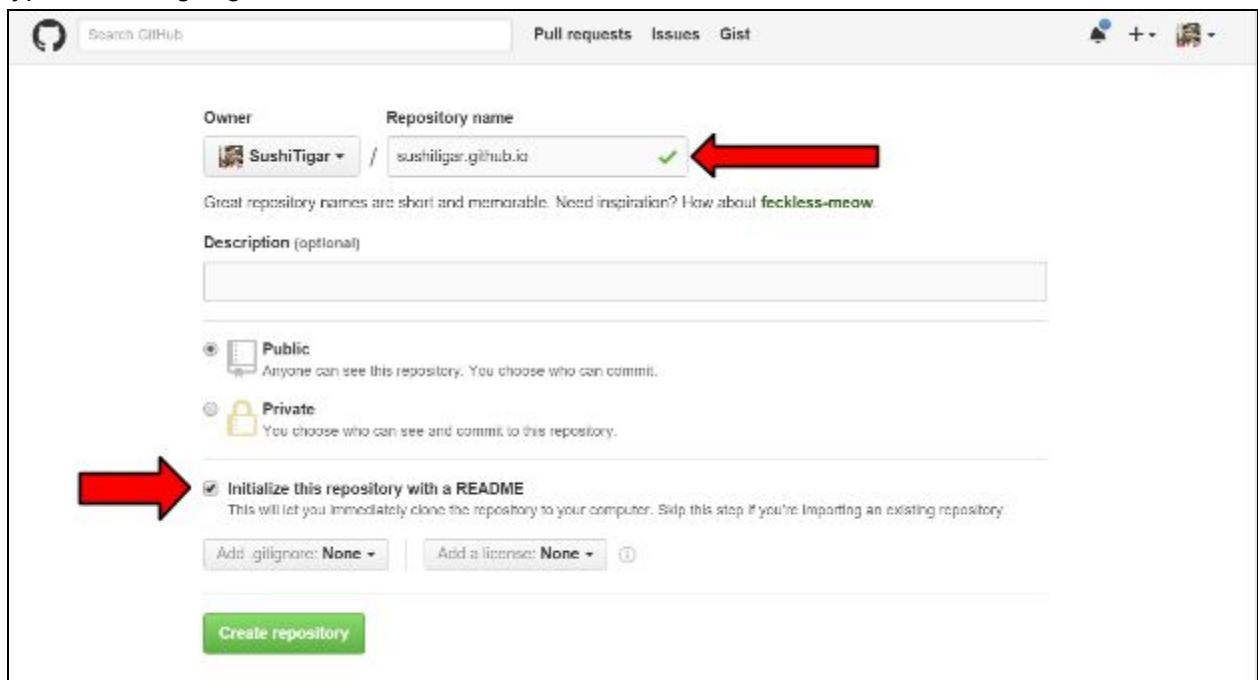
```
Sushiil@DASTSURFACE ~
$ cd Projects
```

# UPLOADING A PAGE TO GITHUB

- Now open up your web browser and go to [github.com](https://github.com). Login if you haven't already. Select the + icon in the top right and then select **New repository**.



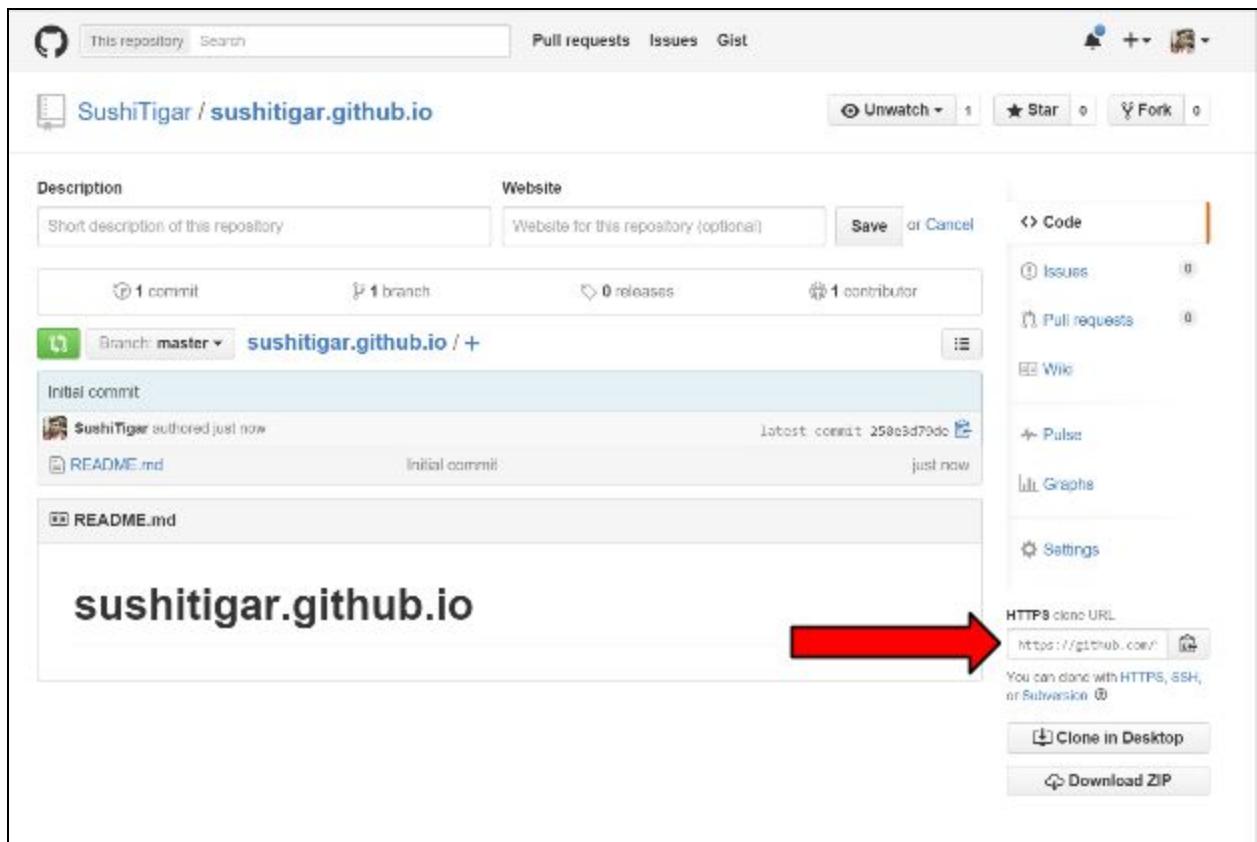
- In the field named "Repository name", type in **username.github.io** replacing "username" with your GitHub username. In this example below, my username is "sushitigar" so I typed "sushitigar.github.io".



- Ensure that you are creating "Public" repo. Add a Description if you like.
- IMPORTANT** select the checkbox next to **Initialize this repository with a README**.
- 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:

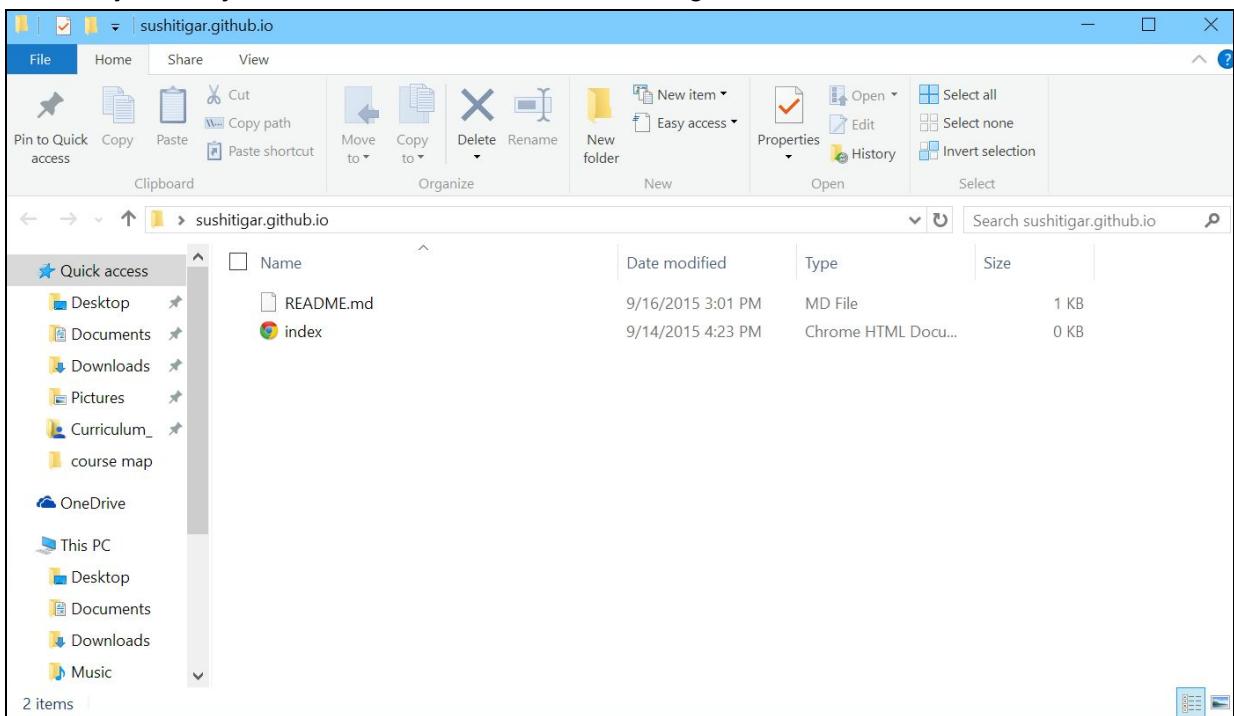
- Git Bash on Windows: right-click on the title bar, select **Edit**, and then select **Paste**.
- Mac: **Cmd-V**.
- Linux: press **Ctrl-Shift-V**.

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

```
Sushitigar@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.
```

- 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.



- 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.

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

- 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

```
Sushitigar@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:
  (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".

```
Sushitigar@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.

```
Sushitigar@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.

```
Sushitigar@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.

# Next Steps After Code 101

Congratulations on completing Code 101! You're about to enter an exciting industry and build some amazing things. So what's next in your journey? Here are some resources to help you as you continue to learn:

## Online Resources

### **Codecademy** - [www.codecademy.com](http://www.codecademy.com)

Codecademy provides step-by-step instructions and interactive tutorials, so you can practice writing the code yourself. Modules cover HTML/CSS, Python, JavaScript, jQuery, PHP, and Ruby.

### **Treehouse** - [www.teamtreehouse.com](http://www.teamtreehouse.com)

By combining quizzes, videos, and code challenges, Treehouse offers an interactive online education and specific paths to help you reach your goals. You can learn the basics of HTML, CSS, Ruby, JavaScript, design, iOS, Android, common development tools, and more.

## Books

### **HTML and CSS: Design and Build Websites** by Jon Duckett

### **JavaScript and JQuery: Interactive Front-End Web Development** by Jon Duckett

This set of books (also sold separately) is highly recommended for anyone interested in web design or development. The beautiful design and straightforward writing style makes it a great option for beginners and covers the pillars of web development. These are the texts for the Code Fellows Code 201 class.

## Code Fellows Classes

### **Code 201: Foundations of Software Development**

\$3,500

Daytime Track: 4 Weeks || Nights & Weekends Track: 8 Weeks

If you're ready to take the next course at Code Fellows, apply for Code 201 to get used to writing syntax and pushing code. You will learn the basics of web development through HTML, CSS, Javascript, and the tools and best practices used by software developers around the world.

Result: Training and experience suited for an internship-level role at a tech company.

Go to <https://www.codefellows.org/class-calendar> to see when the next Code 201 is scheduled!