



Introduction

Hello! We hope you are excited as we are about ChicTech! The point of this guide is to help get you set-up with some of the things we will be using before the retreat. We highly recommend that you take some time out of your day to read through this guide (we know it is lengthy) and try to go through each step. If you do not complete it all, that is perfectly okay! We will be spending the morning the day of helping you finish up any steps you may not have completed.

Please keep in mind that if you ever run into an error where the solution is not given in this guide, feel free to look it up and try to figure out how to fix the error or send us an email at illinoiswcs.outreach@gmail.com and we will try to help you out!

Things We Will Go Over and Goals

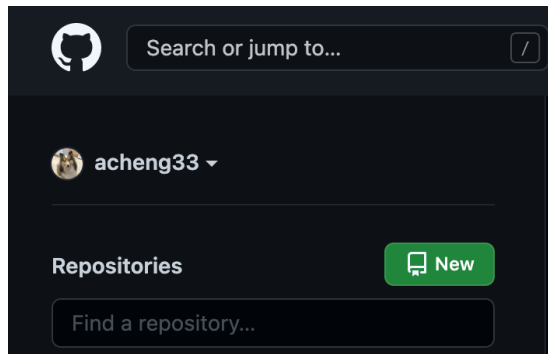
Here is the list of things you will be learning over the course of the day and this guide:

- 1) Github
- 2) Visual Studio Code (VSCode)
- 3) Git / Github Pages
- 4) HTML and CSS
- 5) D3.js

After going through this guide and day of ChicTech workshops, we hope to have you set up with a basic webpage about yourself, and a simple visualization that you can easily show off to your friends and family.

Github and Your First Repository!

- 1) Go to <https://github.com/> and sign up for an account
 - a) **Recommendation:** Use a more professional email so you can continue to use this account for any future personal projects
- 2) Sign into your newly created GitHub account, on the left hand side, look for a button that reads “New”



- 3) You should be directed a page that looks like the following:

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *

Repository name *

 acheng33 ▾ /

Great repository names are short and memorable. Need inspiration? How about [super-duper-octo-meme?](#)

Description (optional)

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**

Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**

A license tells others what they can and can't do with your code. [Learn more.](#)

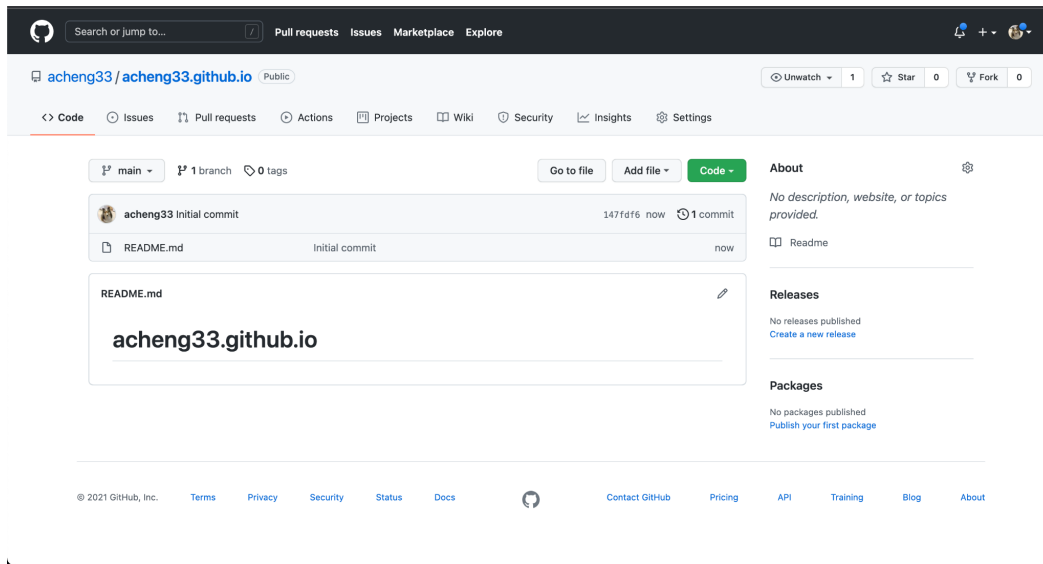
Create repository

- a) Fill in the following for **Repository Name**: *username.github.io*

- i) For this account, it would be *acheng33.github.io*

- b) Check the **Add a README file** box

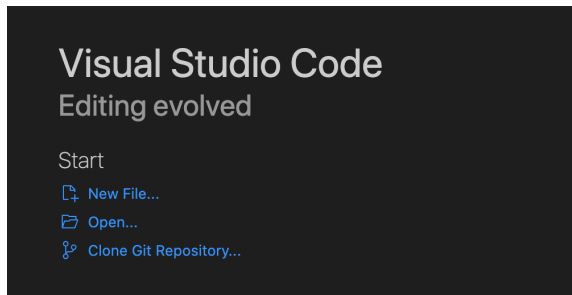
4) After clicking **Create Repository**, it should take you to a page similar to the following



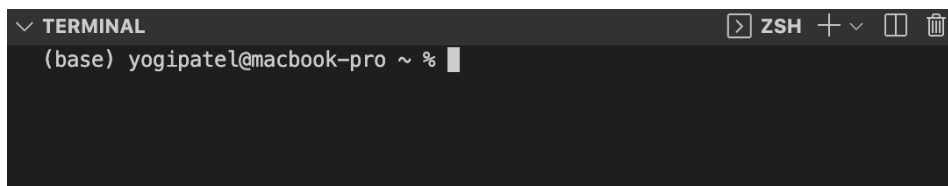
Once you have your repository, move on to downloading git and visual studio code portions of the guide!

VS Code Installation Setup

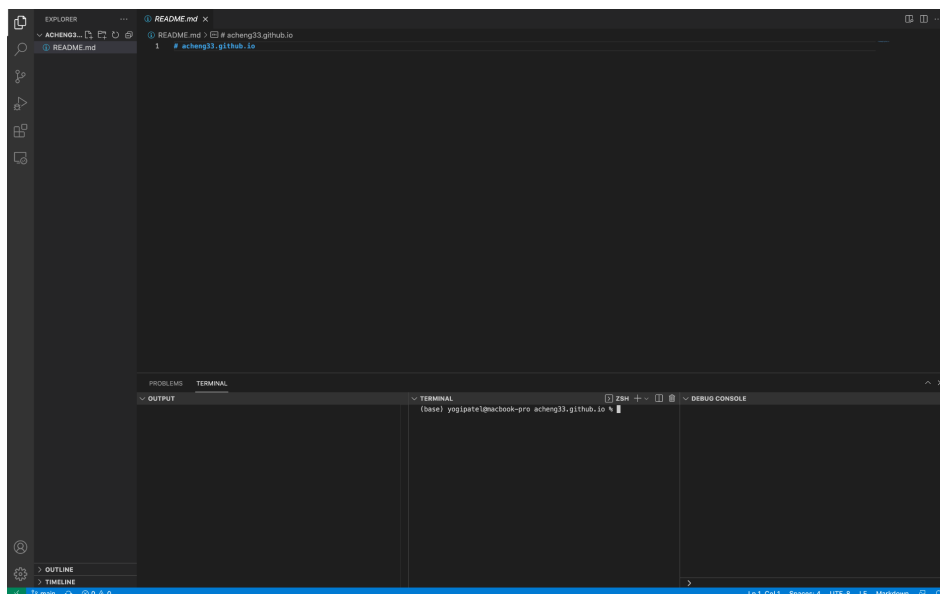
- 1) Go to <https://code.visualstudio.com/download> and download the Visual Studio Code according to the device you are using (Mac or Windows)
- 2) Open the downloaded application and continue with the installation process
- 3) Open Visual Studio Code and create a new project by clicking “New File”



- 4) To open terminal on VS Code, press “control” or “command” and ` (under the ~ key)



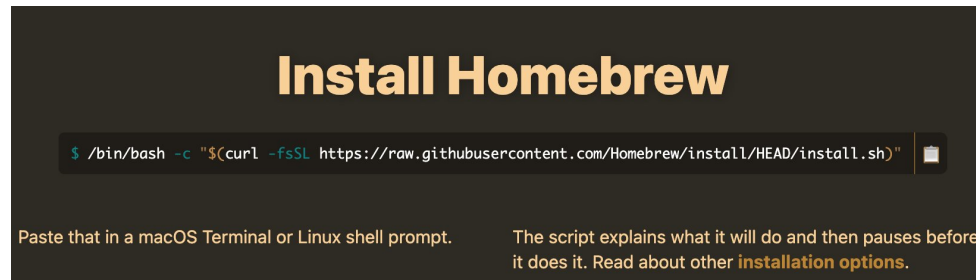
- 5) To clone a repository click “Clone git repository” and enter the link to your repository along with a location on your laptop to clone it to
- 6) Once you clone your repository into [directory], click open and select the folder where you cloned the repository to open it
- 7) You can then open files from the left hand “explorer” side
 - a) What it looks like having a file and terminal open



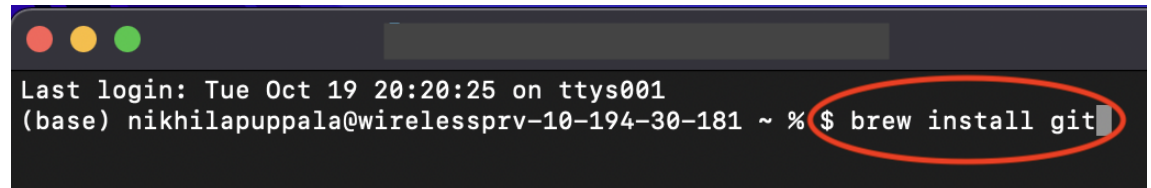
Git Setup Guide for Mac

1) How to download Git:

- a) Open terminal and copy and paste the following command into terminal to install homebrew: `/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"`

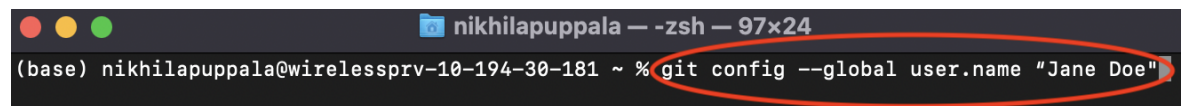


- b) Then enter the following command to install git: `$ brew install git`

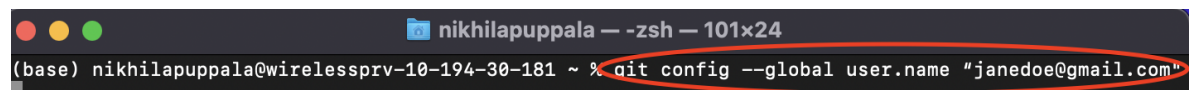


2) Git setup:

- a) Enter the following command into terminal: `git config --global user.name "[firstname lastname]"`
 - i) You should enter a first and last name that is identifiable for later use e.g.



- b) Enter this command to setup user email: `git config --global user.email "[valid-email]"`
 - i) Use the same email that is tied to your github account!



- c) Enter this command to set up automatic command line coloring for Git for easy reviewing: `git config --global color.ui auto`

3) Other helpful commands!

- a) `git status`: Shows the modified files in the working directory and the staging area when you enter `git status` into the terminal

```
(base) nikhilapuppala@wirelessprv-10-194-30-181 acheng33.github.io % git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    nikhilap.html

nothing added to commit but untracked files present (use "git add" to track)
```

- b) git commit: **git commit -m "[descriptive message]"** allows you to commit your staged content as a new commit snapshot

```
(base) nikhilapuppala@wirelessprv-10-194-30-181 acheng33.github.io % git add nikhilap.html
(base) nikhilapuppala@wirelessprv-10-194-30-181 acheng33.github.io % git status
On branch main
Your branch is up to date with 'origin/main'.

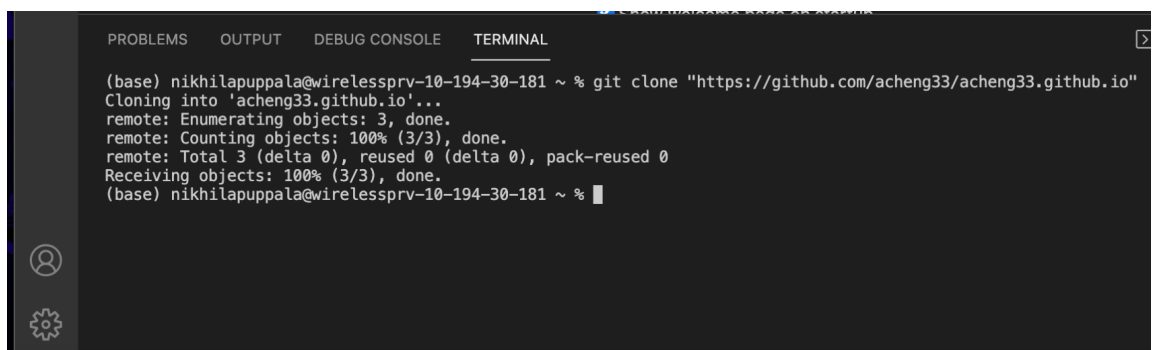
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   nikhilap.html

(base) nikhilapuppala@wirelessprv-10-194-30-181 acheng33.github.io %
```

- c) git add: **git add [file]** allows you to add a file as it looks now to your next commit
- d) git push: **git push [alias] [branch]** allows you to transmit local branch commits to the remote repository branch

```
→ acheng33.github.io git:(main) git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 294 bytes | 147.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/acheng33/acheng33.github.io.git
   147fdf6..2692330  main -> main
→ acheng33.github.io git:(main)
```

- e) git clone: **git clone [url]** allows you to retrieve an repository from a hosted location via URL



```
(base) nikhilapuppala@wirelessprv-10-194-30-181 ~ % git clone "https://github.com/acheng33/acheng33.github.io"
Cloning into 'acheng33.github.io'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
(base) nikhilapuppala@wirelessprv-10-194-30-181 ~ %
```

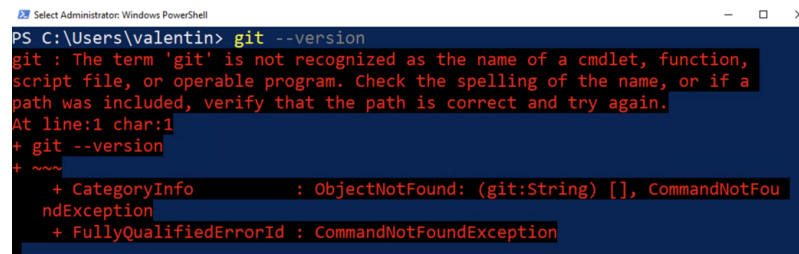
Git Installation for Windows

- 1) Open your terminal (Command Prompt/PowerShell) and check whether you have Git already installed: `git --version`

- a) If you see “git version #.#.#”, you are all set!

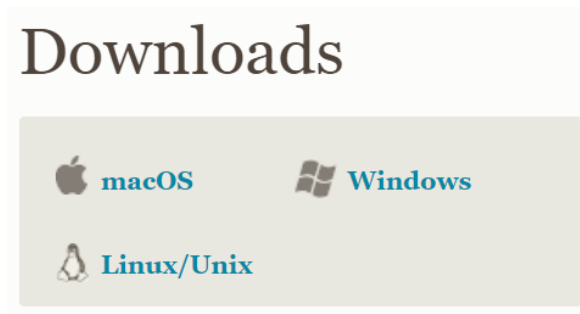
```
PS C:\Users\Jessi> git --version
git version 2.33.0.windows.2
```

- b) If you see an error message, continue with the following steps.



```
Select Administrator: Windows PowerShell
PS C:\Users\valentin> git --version
git : The term 'git' is not recognized as the name of a cmdlet, function,
script file, or operable program. Check the spelling of the name, or if a
path was included, verify that the path is correct and try again.
At line:1 char:1
+ git --version
+ ~~~
+ CategoryInfo          : ObjectNotFound: (git:String) [], CommandNotFou
ndException
+ FullyQualifiedErrorId : CommandNotFoundException
```

- 2) Go to: [Git - Downloads \(git-scm.com\)](https://git-scm.com) and choose the Windows option



- 3) Open the installer you downloaded and continue with the installation process. Leave all settings on default
- 4) (Optional) If you want Git to remember your user name and email address that you used to create your account on Git, enter the following in your terminal:
 - a) `git config --global user.name "Your Name"`
 - b) `git config --global user.email "you@example.com"`

Git commands:

- f) git clone: **git clone [url]** allows you to retrieve an repository from a hosted location via URL

```
PS C:\Users\Jessi> git clone https://github.com/acheng33/acheng33.github.io.git
Cloning into 'acheng33.github.io'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
PS C:\Users\Jessi> █
```

- g) git status: Shows the modified files in the working directory and the staging area when you enter **git status** into the terminal

```
PS C:\Users\Jessi\acheng33.github.io> git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    jessica.html

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\Jessi\acheng33.github.io> █
```

- h) git add: **git add [file]** allows you to add a file as it looks now to your next commit

```
PS C:\Users\Jessi\acheng33.github.io> git add jessica.html
PS C:\Users\Jessi\acheng33.github.io> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   jessica.html

PS C:\Users\Jessi\acheng33.github.io> █
```

- i) git commit: **git commit -m "[descriptive message]"** allows you to commit your staged content as a new commit snapshot

```
PS C:\Users\Jessi\acheng33.github.io> git commit -m "Add jessica.html file"
[main 0d50906] Add jessica.html file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 jessica.html
PS C:\Users\Jessi\acheng33.github.io> █
```


- j) **git push**: **git push [alias] [branch]** allows you to transmit local branch commits to the remote repository branch

```
→ acheng33.github.io git:(main) git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 294 bytes | 147.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/acheng33/acheng33.github.io.git
  147fdf6..2692330  main -> main
→ acheng33.github.io git:(main) █
```

git pull: **git pull** allows you to fetch and merge any commits from the tracking remote branch

HTML/CSS

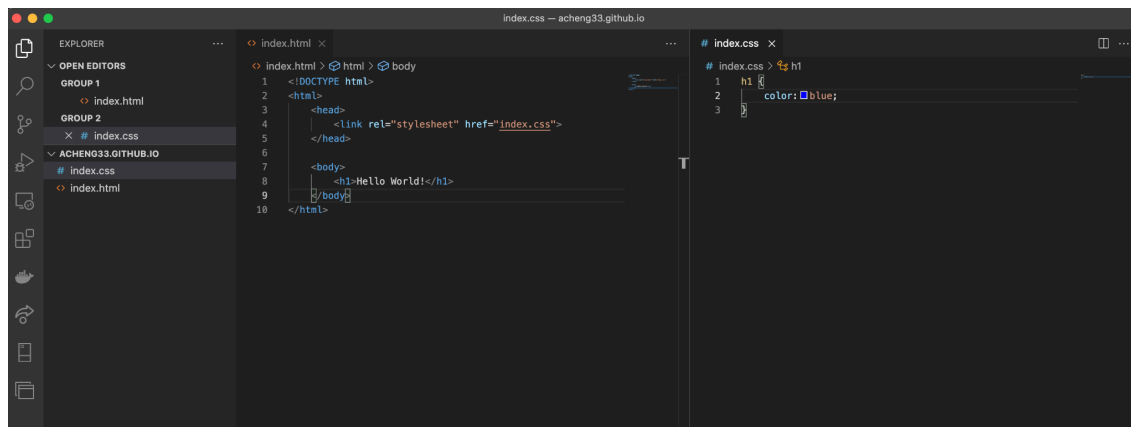
HTML and CSS are often used together to build very simple static websites. You may have heard of tools such as React.js, Jekyll or possibly others well! The reason why we are using HTML and CSS is that it's easy for setup and unlike some other tools for building websites, there is no differences in setup when it comes to which operating system (Mac/Windows/Linux) you may be using.

There are many many resources online for using HTML and CSS and we recommend trying some out. One of the resources you can look at is from W3 schools which can be found here: <https://www.w3schools.com/html/default.asp>

We'll provide a very very simple tutorial on how to have "Hello World" show up on your website and from there we encourage you to add new pages, new sections, whatever you feel like doing to make your website what you want it to be. Also feel free to try out other ways to build website such as React and Jekyll, though it will not be necessary for ChicTech. We will also be spending the morning working on some goals for your websites and have our staff helping you with adding anything you may want to add onto your website!

Open up VSCode and create a file called **index.html** and a file called **index.css**

In your **index.html** and **index.css** files, add the following code respectively:



Once you have, open up a terminal in VSCode and add and push these two files. Once you have, you can go to [https://\[githubusername\].github.io](https://[githubusername].github.io) to see your website! In my case, I went to <https://acheng33.github.io>

D3.js

(info from <https://betterprogramming.pub/a-beginners-guide-to-d3-js-962a3234f76>)

D3.js is a JavaScript library for manipulating documents based on data. It is a great tool for creating interactive data visualizations.

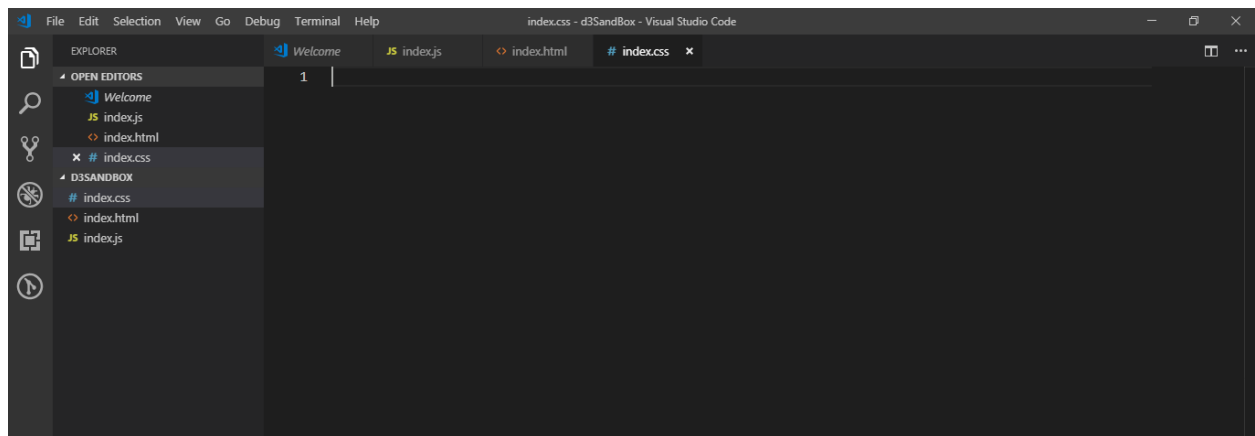
Some elements of D3.js that we will go through:

- Selection
- Data binding and loading
- Data visualization
- Scaling
- Axis

Setting up

Create a directory and open it in your code editor. Create 3 files: **index.js**, **index.css** and **index.html**

*Note: You can use the same **index.html** and **index.css** from the earlier portion*



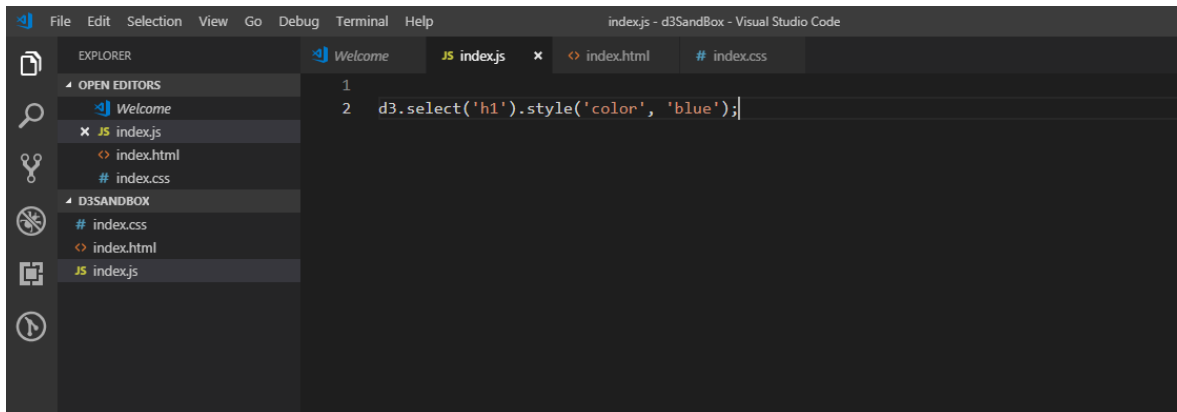
In your index.html, make sure you are loading the D3 library as well as your index.js file (and make sure you are using the [most recent link to the library](#)).

```
<script src="https://d3js.org/d3.v5.min.js"></script>
```

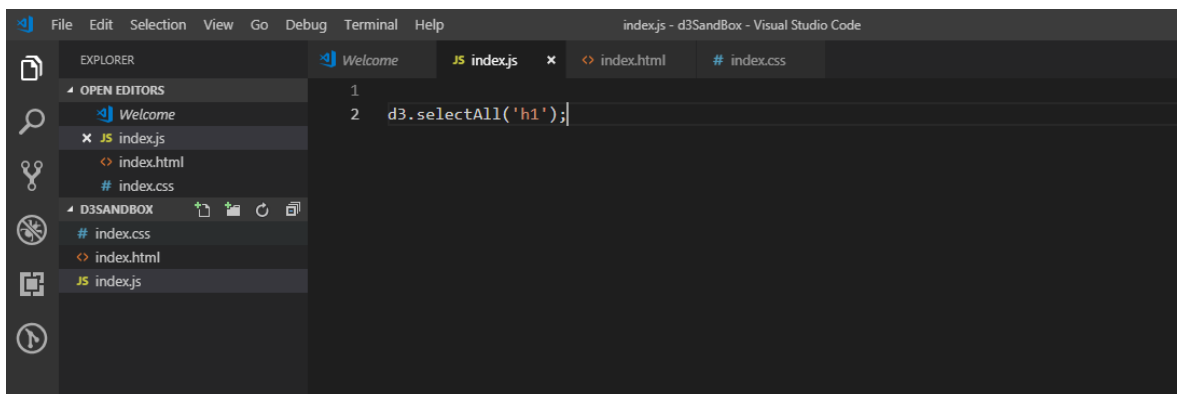
Selection

D3 allows you to identify DOM elements using their name (e.g p,h1) or their CSS selector. How then do we select them for manipulation after identifying them? D3 provides us with 2 methods for selection: `d3.select()` and `d3.selectAll()`. They accept either the name of the

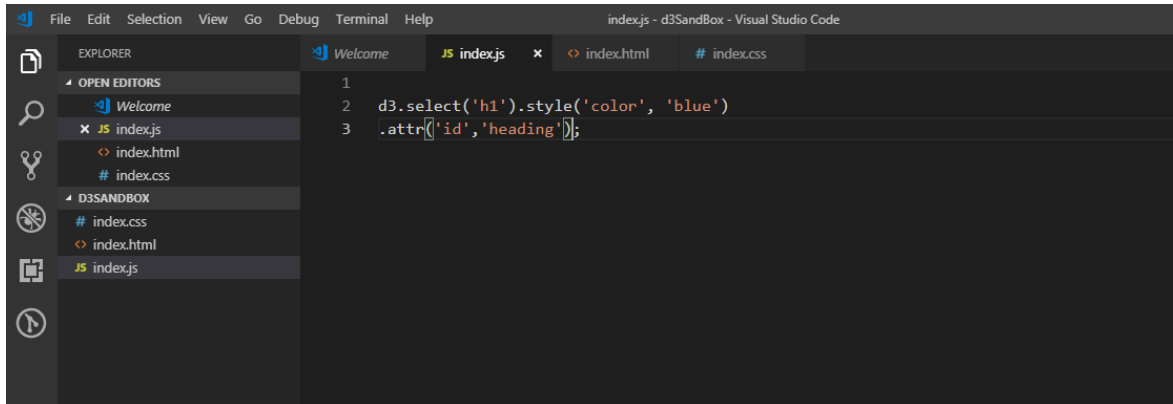
element or the CSS selector (let's call this the 'identifier') as an argument and return the selection of the element.



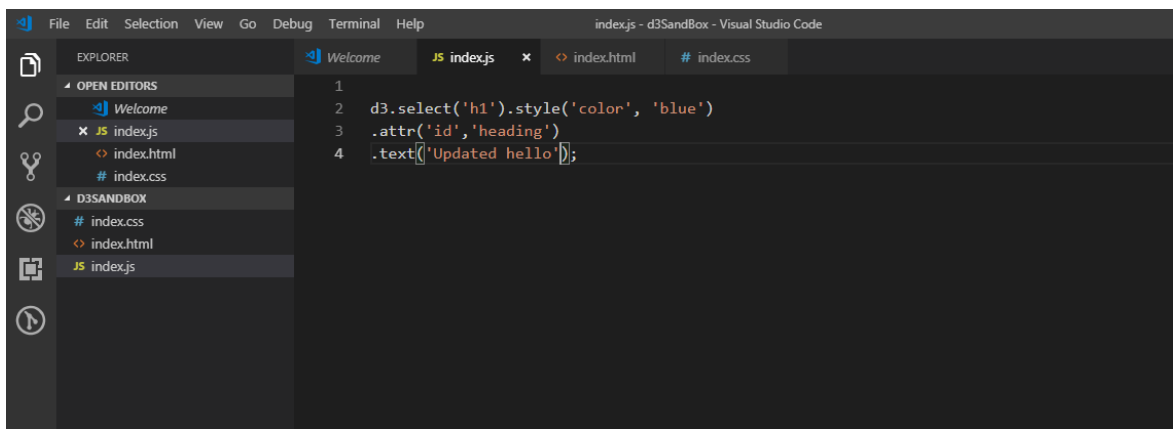
This will return the first `<h1></h1>` it finds and colours it blue. If it does not find any h1 tags, it will return an empty selection.



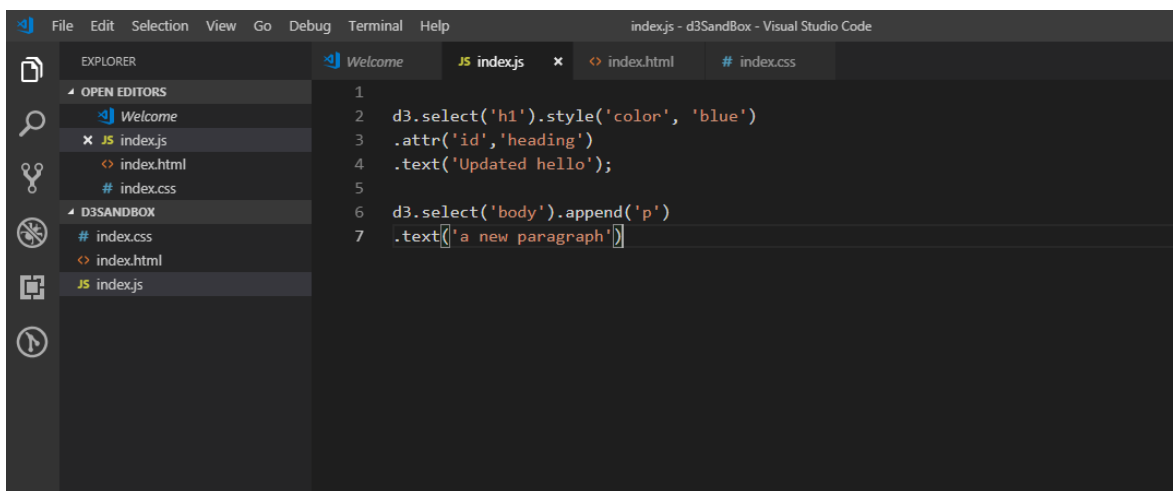
This will return all the `<h1></h1>` it finds. If it does not find any h1 tags, it will return an empty selection.



With the `.attr()` method, you can add attributes to the selected element. It accepts two string parameters: the name of the attribute and the value.

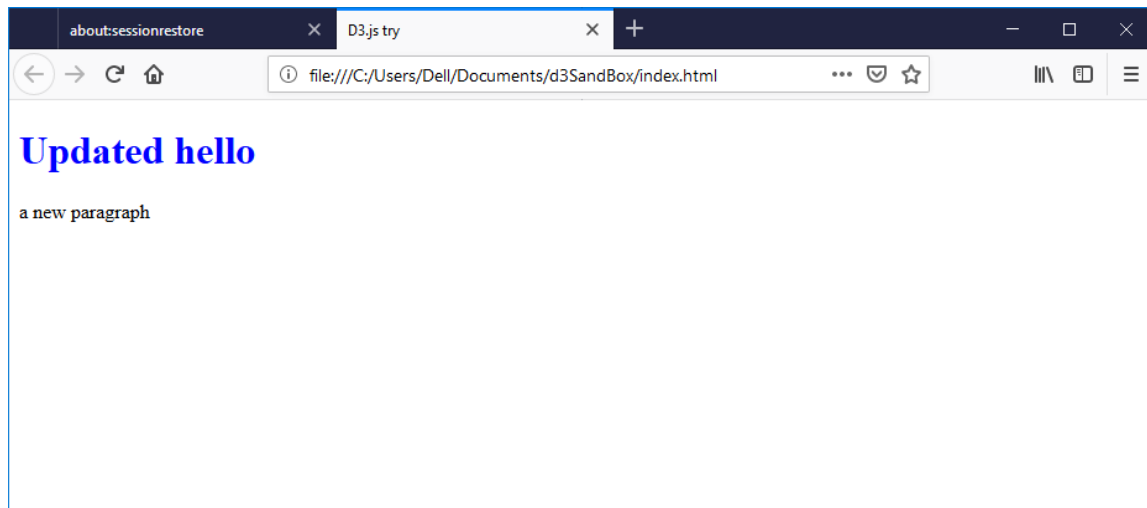


With the `.text()` method, you can update the value of the selected element. It accepts one string parameter: the value.



With the `.append()` method, you can add a new element to the DOM. It accepts one string parameter: the name of the tag.

This is what it should look like in your browser.



Task 1: add more `p` tags with the `append()` method and set their value to your favourite food with the `text()` method; then change their colours to orange using the `selectAll()` and the `style()` methods. Please do this task to solidify your knowledge. Don't worry if you don't get it in the first try—go through the Selection section again and you'll get the hang of it.

Data Binding and Loading

With D3, you can load data into your DOM element with the `.data()` method which accepts your dataset as an argument. With the help of the `.enter()` method, you can perform a 'for each element'-like operation on the dataset passed into `.data()`.

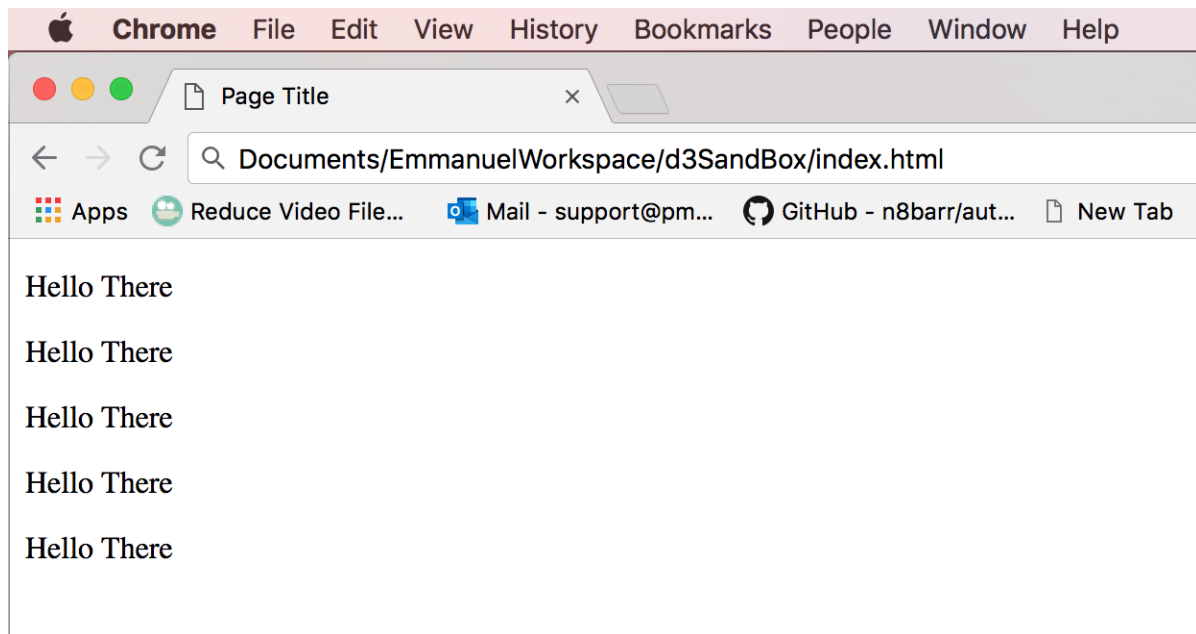
```

1 let dataset = [1,2,3,4,5]
2 d3.selectAll('p')
3   .data(dataset)
4   .enter()
5   .append('p')
6   .text('Hello There');

```

dataBinding.js hosted with ❤ by GitHub

[view raw](#)



Task 2: Change the dataset to contain your favourite dishes and the value of the p tag to the values of the elements from the dataset.

Hint: replace line six with this:

```

1 .text(function(d){return d})
2 or .text(d=>d)//Arrow function😄

```

taskTwo.js hosted with ❤ by GitHub

[view raw](#)

Data Visualization

Now that we know a little of the basics, we are going to create a simple bar chart with an arbitrary dataset.

Include an svg tag in your HTML file:

```
<svg class="bar-chart"></svg>
```

Add this to your index.js file:

```
1  let dataset = [30,12,103,170,45,95,190,75];
2  let svgHeight= 200, svgWidth = 500, barSpacing = 5;
3  let totalBarWidth = (svgWidth/dataset.length);
4  let barWidth = totalBarWidth-barSpacing
5  let svg = d3.select('svg')
6      .attr('width', svgWidth)
7      .attr('height', svgHeight);
8  let barchart = svg.selectAll('rect')
9      .data(dataset)
10     .enter()
11     .append('rect')
12     .attr('y',d=>svgHeight-d)
13     .attr('height',d=>d)
14     .attr('width',barWidth)
15     .attr('transform', (d,i)=>{
16         let translate = [totalBarWidth*i, 0];
17         return `translate(${translate})`;
18     });
```

index.js hosted with ❤ by GitHub

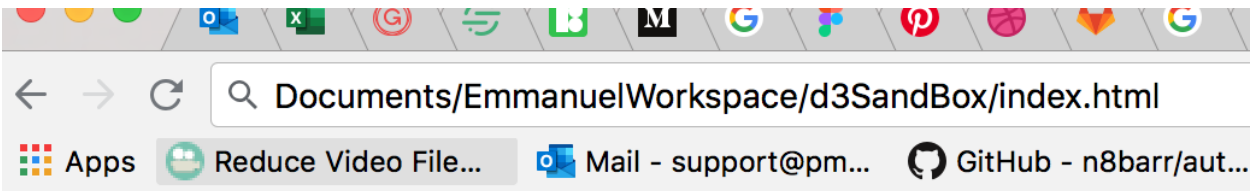
[view raw](#)

That looks like a lot of code, but don't be alarmed. Go over it again. The width and height of the svg tag are specified in svgWidth and svgHeight respectively, the spacing between the bar is set in barSpacing, and totalBarWidth is the width of the bar(barWidth) and the barSpacing

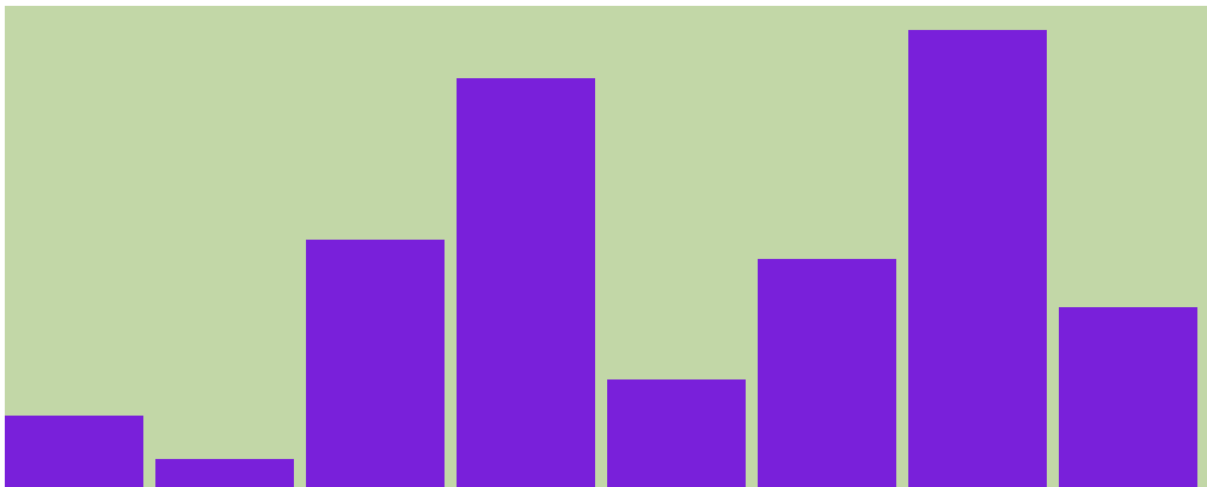

```
1  /*index.css*/
2  rect{
3      fill: rgb(122, 2, 221);
4  }
5  svg{
6      background: rgb(193, 216, 165);
7  }
```

index.css hosted with ❤ by GitHub

[view raw](#)



Simple D3 Barchart



Goals for your website

- 1) Add your name
- 2) Add any picture (Yourself, pet, etc.)
- 3) Add a title to describe yourself
- 4) Add some colors/Change background color
- 5) Make another tab on your website and add any video to it
- 6) Add a button that takes you to your new tab, alternatively, add a navigation bar!