

**HECK YEA, IT'S  
CCLAB!**

**Vanilla Javascript Workshop**  
**Thursday from 5:30 to 6:50**

# LOCAL STORAGE

*how'd it go?*

LOCAL  
STORAGE  
*in* 11 LINES

# LOCAL STORAGE

after `var taskArray = [ ];`

```
//update local storage
var updateLocalStorage = function(){

    //stringify the task array and save it as 'taskList' in local storage
    var taskListString = JSON.stringify(taskArray);
    localStorage.setItem('taskList', taskListString);

};
```

# LOCAL STORAGE

updateTasks( )

```
//check if there is a taskList in localStorage  
var taskListString = localStorage.getItem('taskList');  
if(taskListString){  
    taskArray = JSON.parse(taskListString);  
};
```

# LOCAL STORAGE

saveTask( )

```
//add the value to the taskArray  
taskArray.push(newTask);
```

```
//update local storage  
updateLocalStorage();
```



```
//update your task view  
updateTasks();
```

# LOCAL STORAGE

deleteTask( )

```
//go to the taskNumber positiong of taskArray and remove one object  
taskArray.splice(taskNumber, 1);
```

```
//fire update local storage  
updateLocalStorage();
```



```
//update tasks to show new array  
updateTasks();
```



# LOCAL STORAGE

```
init( )
```

```
console.log("Hi! I'm Ready!");
```

```
//define "add" button
```

```
var addButton = document.getElementById("addButton");
```

```
//add event listener for click
```

```
addButton.addEventListener('click', function(e){...});
```

```
//update tasks
```

```
updateTasks();
```



# TERMINAL & GIT

**SHELL YEAH!**

# TERMINAL + GIT

```
git --version
```

# TERMINAL + GIT

```
$ git version 1.8.2.1
```

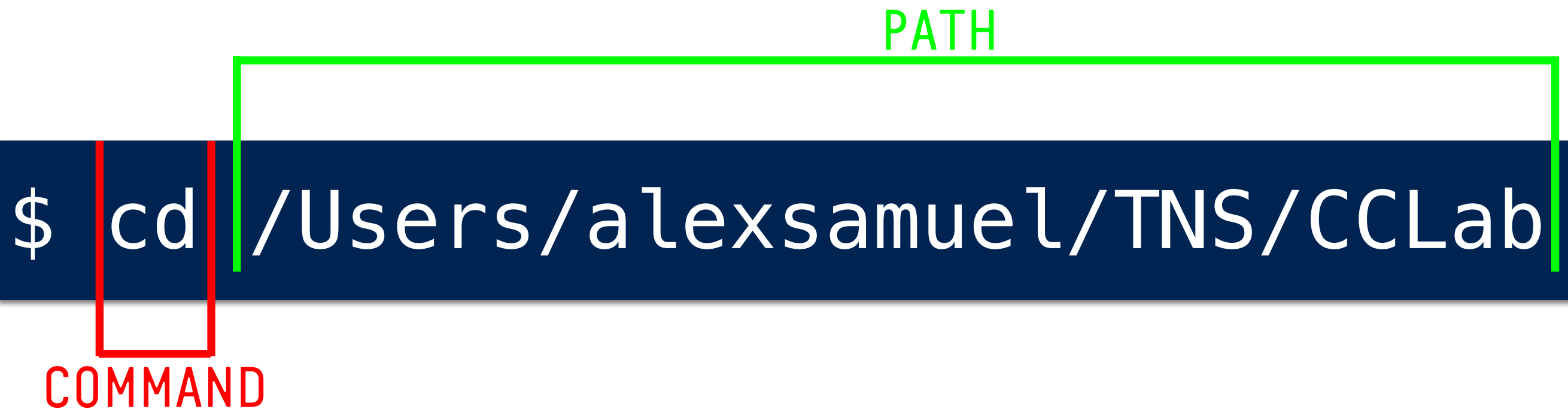
# TERMINAL + GIT

type **cd** and space

# TERMINAL + GIT

drag your CCLab folder  
to the terminal window

# TERMINAL + GIT



A diagram illustrating the components of a terminal command. The command is displayed on a dark blue background: `$ cd /Users/alexsamuel/TNS/CCLab`. A red bracket labeled "COMMAND" points to the `cd` part. A green bracket labeled "PATH" points to the directory path `/Users/alexsamuel/TNS/CCLab`.

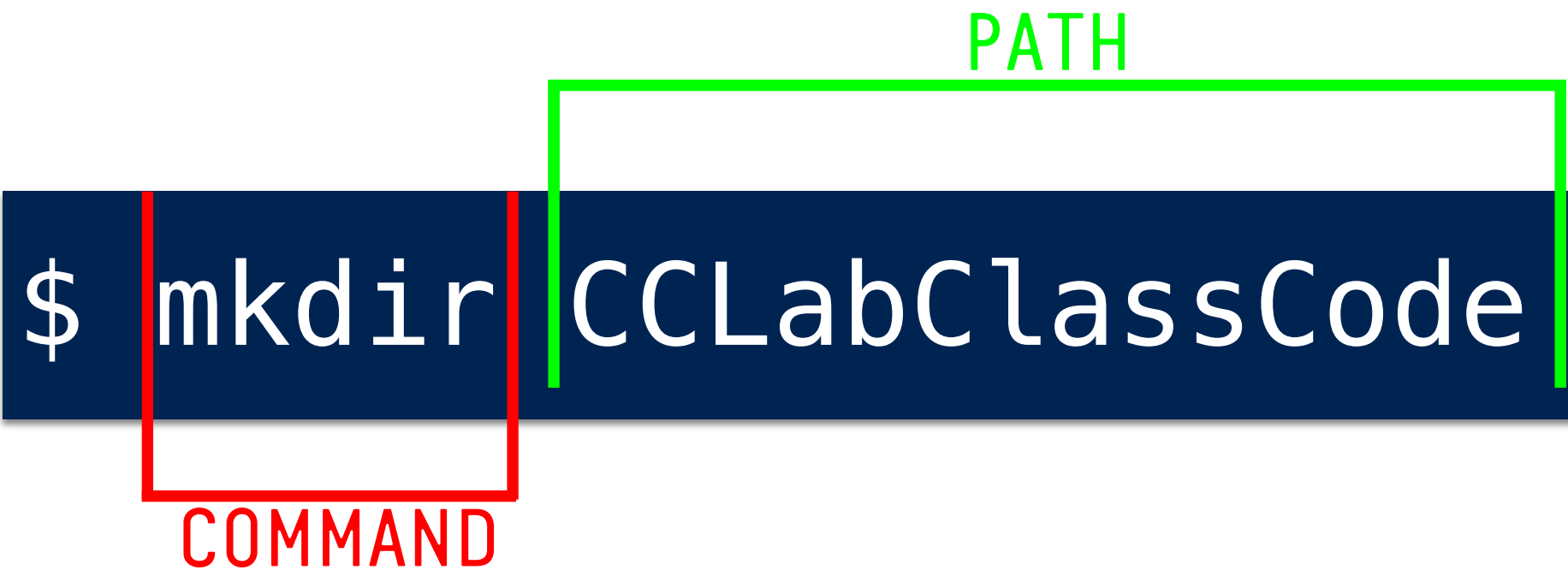
COMMAND

PATH

`cd` means “change directory”



# TERMINAL + GIT



A diagram illustrating the components of a terminal command. The command `$ mkdir CCLabClassCode` is displayed on a dark blue background. A red bracket below the word `mkdir` is labeled `COMMAND` in red text. A green bracket above the path `CCLabClassCode` is labeled `PATH` in green text.

```
$ mkdir CCLabClassCode
```

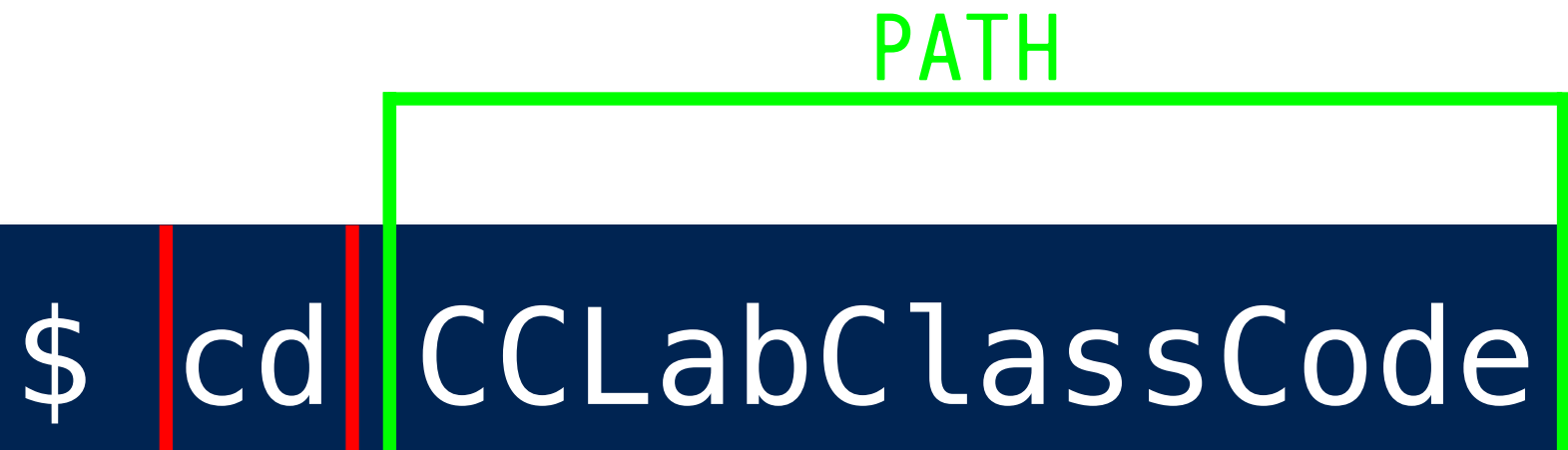
COMMAND

PATH

`mkdir` means “make directory”

# TERMINAL + GIT

\$ cd CCLabClassCode



PATH

COMMAND

# TERMINAL + GIT

```
$ pwd
```

pwd means “print working directory”

# TERMINAL + GIT

```
$ git init
```

# TERMINAL + GIT

you should see something like...

```
$ /Users/alexsamuel/TNS/CCLab/  
CCLabClassCode
```

# TERMINAL + GIT

COMMAND

REPOSITORY NICKNAME

```
$ git remote add classcode  
https://github.com/aesam/CCLAB-  
Fall2014
```

# TERMINAL + GIT

COMMAND

BRANCH

```
$ git pull classcode master
```

REPOSITORY NICKNAME

# TERMINAL + GIT

check the folder



# TERMINAL + GIT



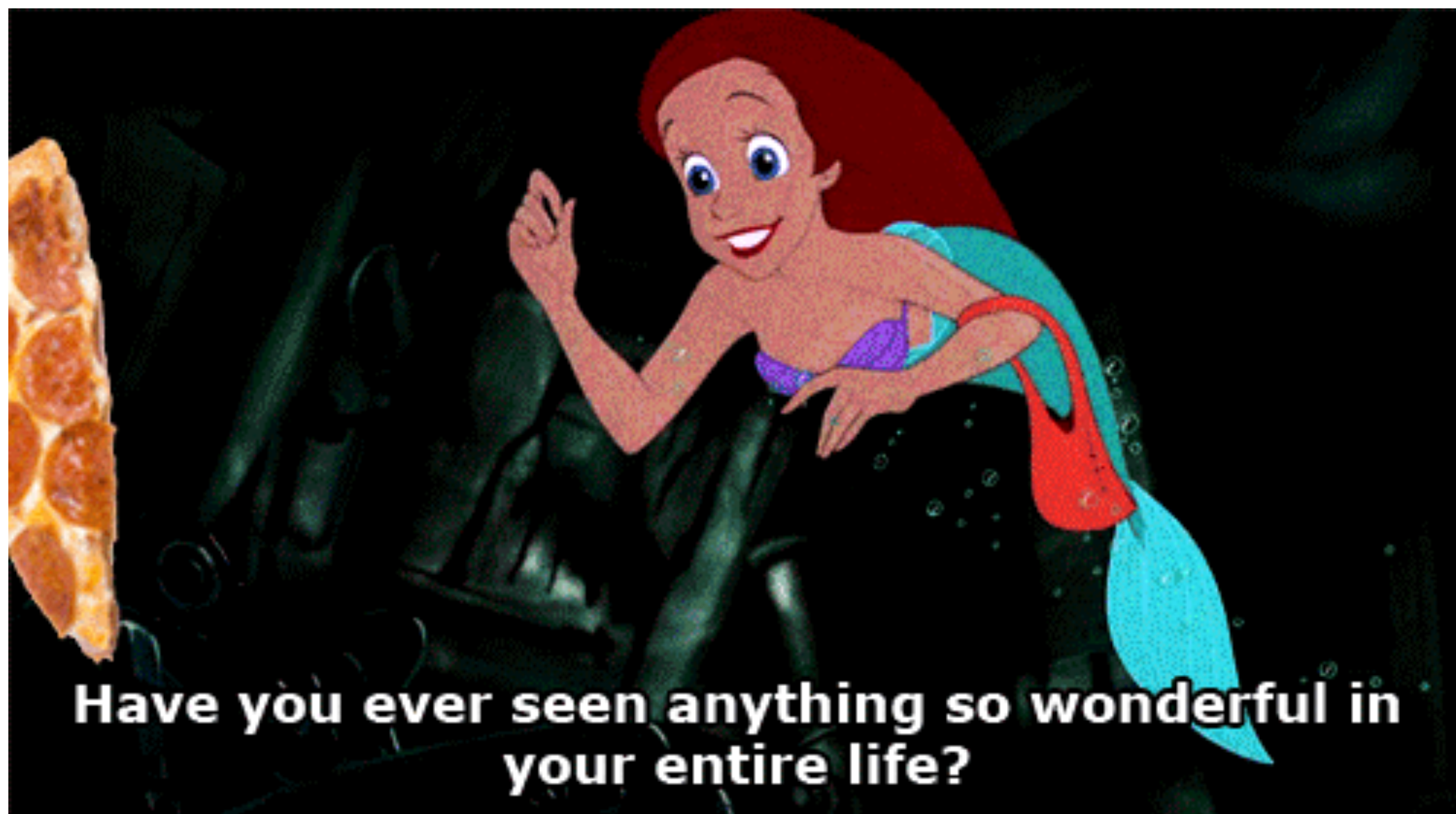
# JAVASCRIPT LIBRARIES

# LIBRARIES

A JavaScript library is a bunch of pre-written JavaScript files that make it easier to do cool stuff.

# LIBRARIES

Like pre-made pizza dough.



# LIBRARIES

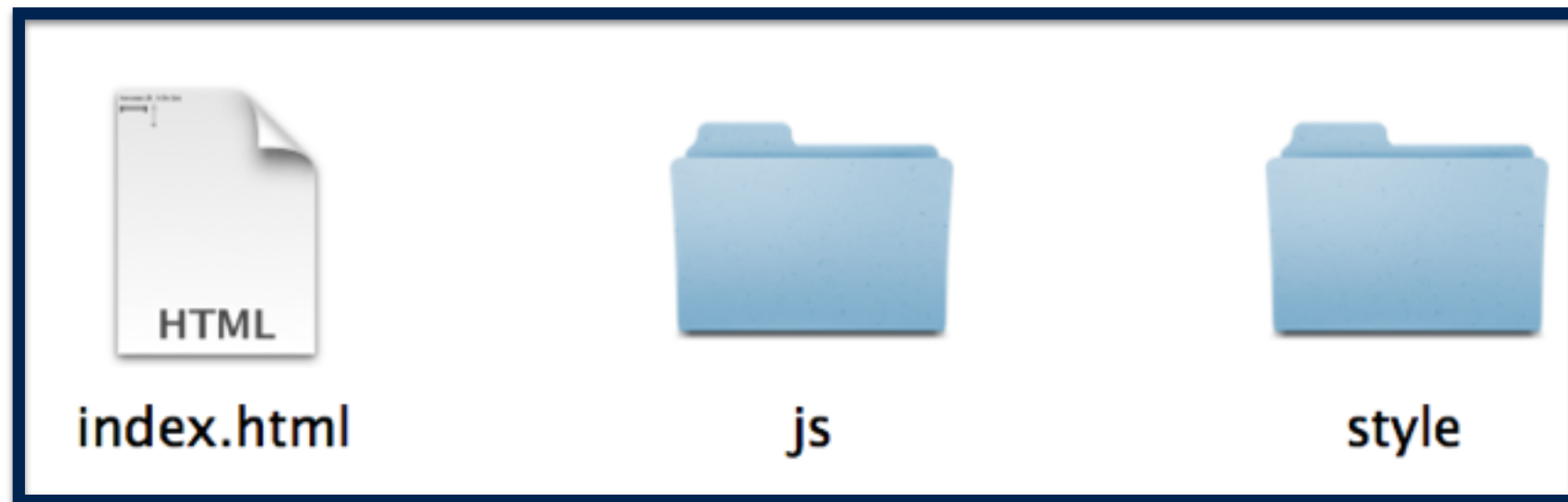
Libraries contain objects + functions that you can reference the same way you would reference objects + functions that you wrote.

# LIBRARIES

Libraries contain objects + functions that you can reference **ALMOST** the same way you would reference objects + functions that you wrote.

# LIBRARIES

## YOUR ROOT FOLDER





# LIBRARIES

## YOUR STYLE FOLDER



style.css

# LIBRARIES

## YOUR SCRIPTS FOLDER



library\_name



main.js

# LIBRARIES

## YOUR HTML FILE


```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>My Awesome Web App</title>
  <link rel="stylesheet" type="text/css" href="style/style.css">
  <script src="js/library_name/main_library_file.js"></script>
</head>
<body>

  <div id="myAwesomeWebApp">

    <!-- Your Site Content -->

  </div>

  <script src="js/main.js"></script>
</body>
</html>
```



**LET'S TRY IT!**

**YES, THERE'S  
HOMEWORK**

1. GET YOUR CODE UP AND RUNNING
2. GO TO JSDB.IO + CHOOSE A LIBRARY
3. ADD THAT LIBRARY TO THE CURRENT CODE, OR MAKE A NEW APP THAT USES JQUERY + YOUR NEW LIBRARY



**YOU GOT THIS!**