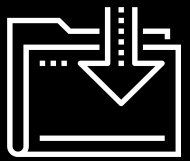
Citi Bike Project with Leaflet & Intro to Projects

Data Boot Camp

Lesson 15.3



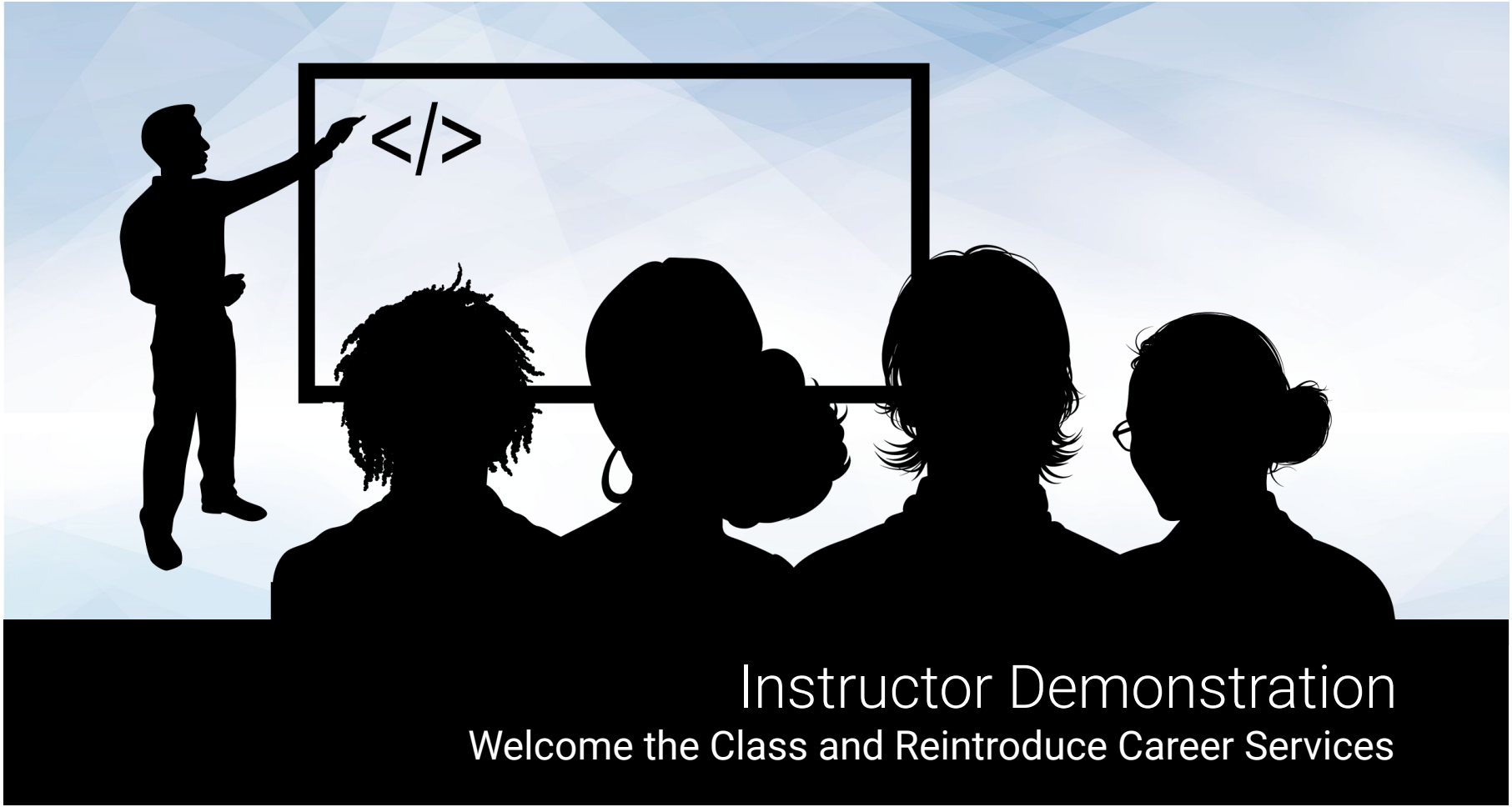
Class Objectives

By the end of this lesson, you will be able to:

Complete an in-class group project using leaflet.js.

Deploy data visualizations to GitHub Pages.

Draft a project proposal with team.

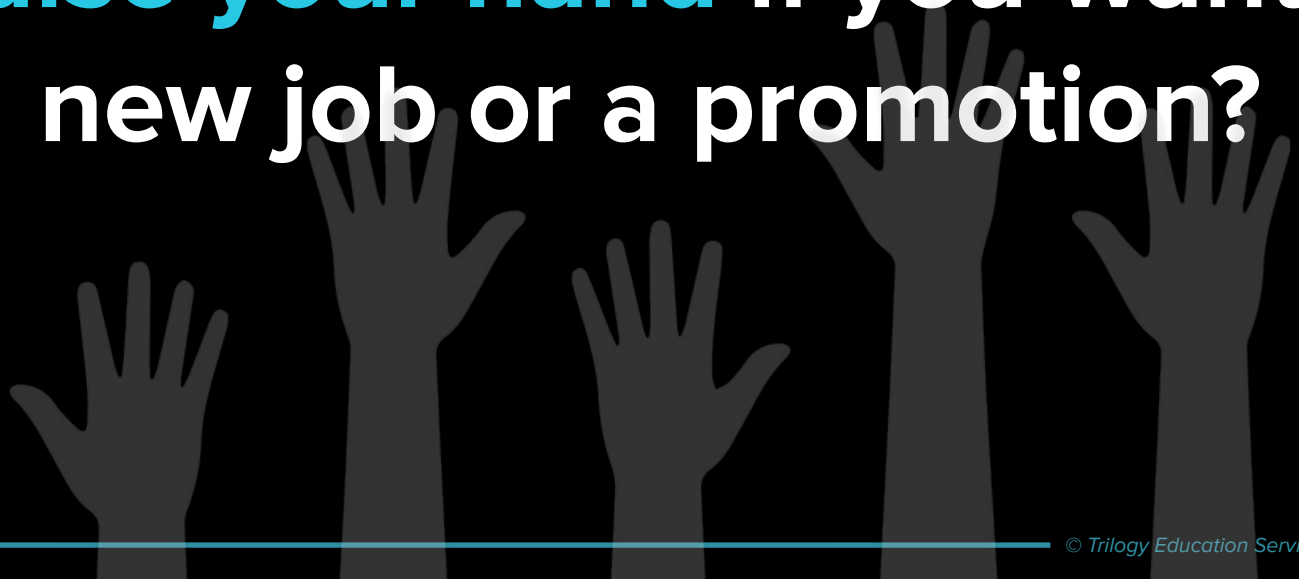


Instructor Demonstration

Welcome the Class and Reintroduce Career Services

UPCOMING CAREER SERVICES

Raise your hand if you want a new job or a promotion?



Why Discuss Career Services Now?

In the next few weeks, you will complete a Career Services Preferences survey on BootcampSpot

The options available:

Information about online career events and workshops

Access to JobTrack, a job search management tool

Feedback on your professional materials from your Profile Coach - Resume, Github, LinkedIn, etc.

1:1 scheduled career coaching sessions with your Career Director at least twice a month, OR a Career Director you can reach out to when you have questions

First Step: Becoming Employer Ready

You MUST HAVE polished professional materials to begin applying for jobs

REVIEW MILESTONE 1

Employer Ready vs. Employer Competitive for a refresher

Once you are Employer Ready, you'll unlock access to additional Career Services support

SMALL GROUP TECHNICAL INTERVIEW WORKSHOPS

Within one 90-minute session, 50% of students increase their scores by 50% on technical assessments.

JOB REFERRALS

We have a network of employers who regularly partner with us to hire tech talent. While we cannot guarantee a referral, you can only be referred when you are Employer Ready.

Introduction to Career Services

Working With Your Career Director

Your Career Director provides you with 1:1 coaching to help you be Employer Competitive in your job search.

Topics include: Applying, gaining traction to land interviews, conducting mock interviews, networking, salary negotiation, motivation and more!

YOU HAVE TWO OPTIONS

1. 1:1 scheduled bi-monthly recurring coaching calls
2. Reaching out to your Career Director when needed

We recommend scheduled Recurring Calls - Why?

The data shows that our students who are Employer Ready and participate in recurring calls are MUCH MORE LIKELY to secure the jobs they want

Career Services Next Steps

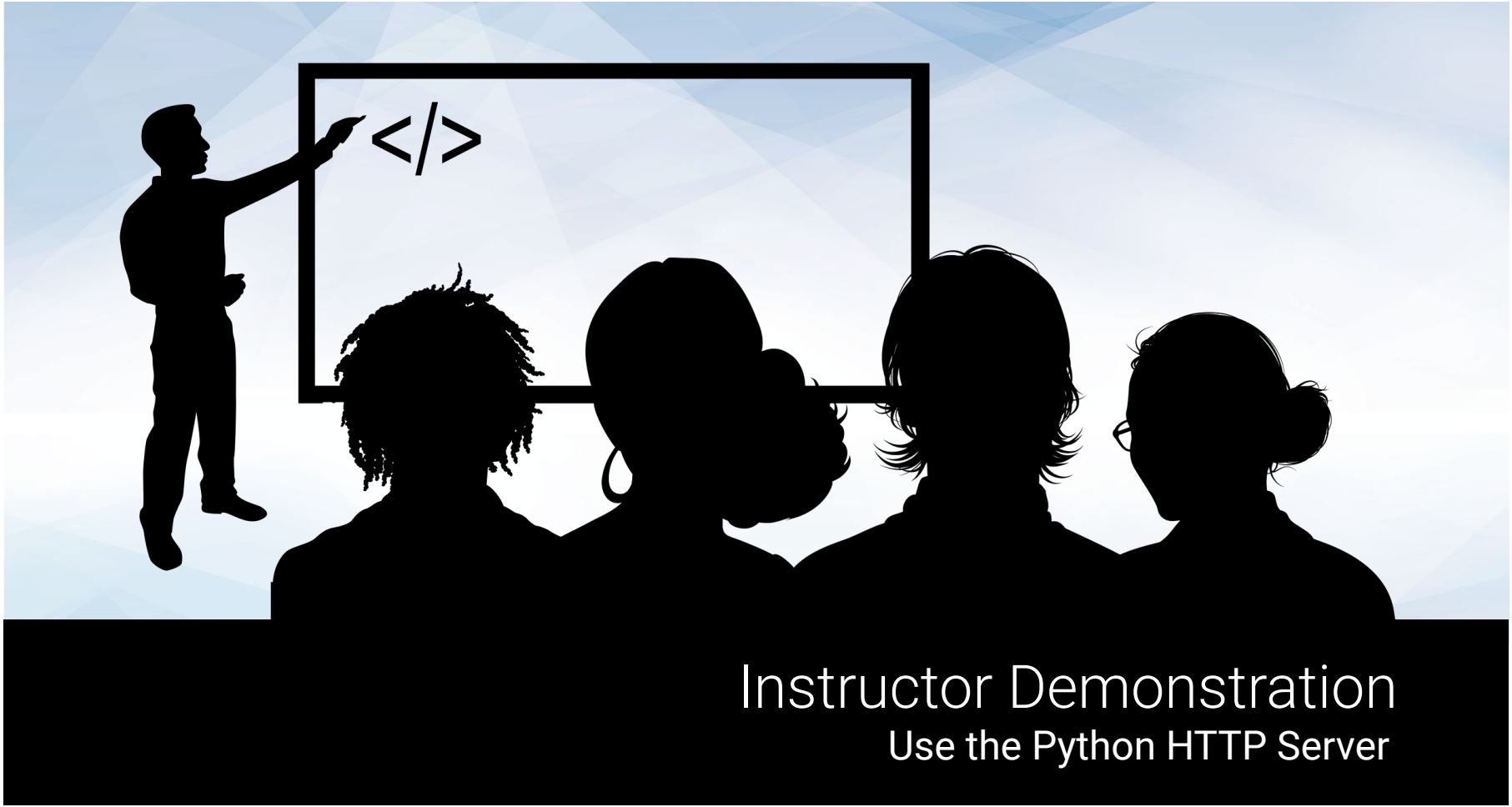
1. **Visit the Career Services page on BootcampSpot for resources on developing your Employer Ready professional materials**

   Submit your professional materials for review. You'll receive feedback from your Profile Coach within 5 days.

2. **Be on the lookout for an email from your Career Director inviting you to schedule your first coaching call in the next few weeks**

   Ready to meet with your Career Director now? They've sent you multiple emails throughout the course - respond to an email to get connected!

3. **Register for Career Service events and workshops**



Instructor Demonstration

Use the Python HTTP Server

Instructor Do: Use the Python HTTP Server

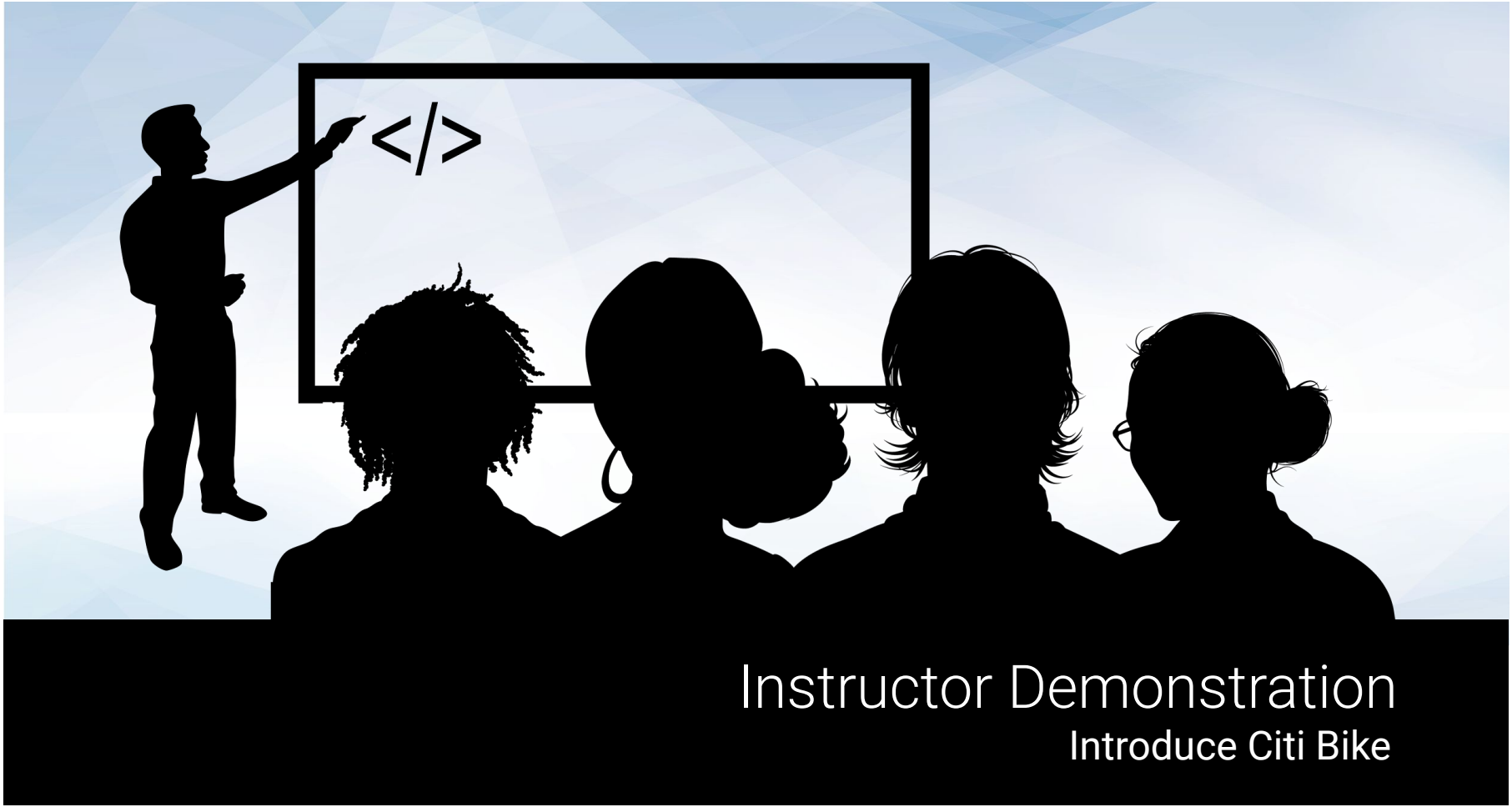
`<Time to Code>`

➔ Few things to note as we live code!

- A **server** is a program or device that performs actions such as processing and sharing data.
- **Cross-Origin Resource Sharing** (**CORS**) is a mechanism that tells browsers—through the HTTP headers in a web application—to access selected resources from a web server. CORS provides a way to allow cross-origin requests.

```
python -m http.server
```

Happy Coding!

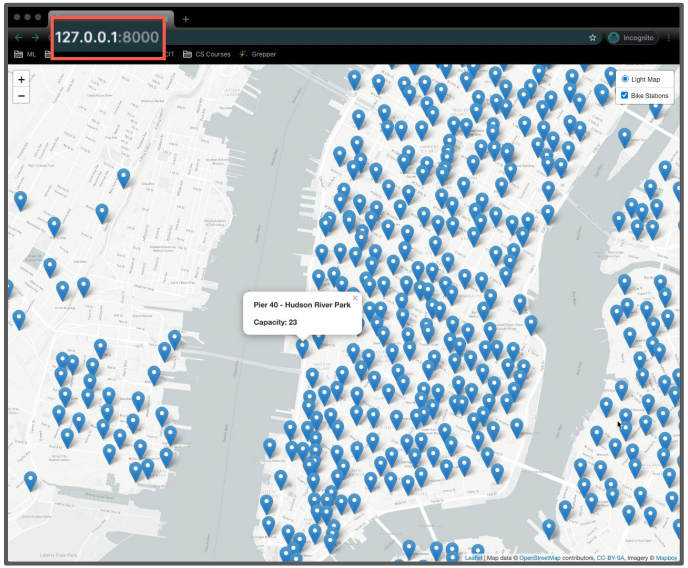


Instructor Demonstration

Introduce Citi Bike

Instructor Do: Introduce Citi Bike

Basic Version:



➔ Citi Bike API Station Information Endpoint.

```
d3.json("https://gbfs.citibikenyc.com/gbfs/en/station_information.json", createMarkers);
```

➔ One kvp (key value property) of the json.

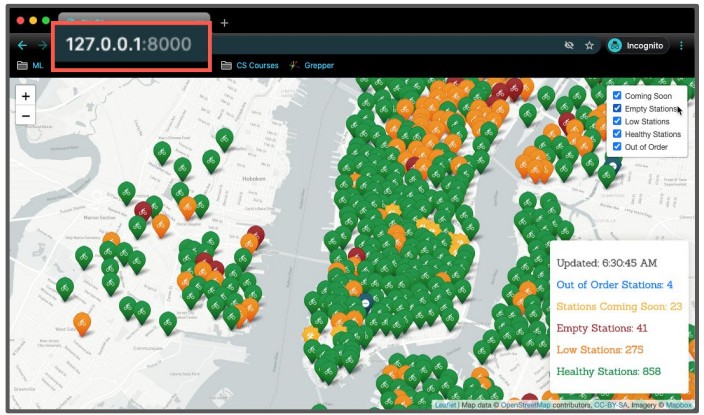

```
{"stations":[
    {"station_type":"classic",
    "lon":-73.99392888,
    "region_id":"71",
    "lat":40.76727216,
    "rental_url":"http://app.citibikenyc.com/S6Lr/IBV092JufD?station_id=72",
    "name":"W 52 St & 11 Ave",
    "short_name":"6926.01",
    "rental_methods":["CREDITCARD","KEY"],
    "electric_bike_surcharge_waiver":false,
    "external_id":"66db237e-0aca-11e7-82f6-3863bb44ef7c",
    "eighthd_station_services":[],
    "capacity":55,"has_kiosk":true,
    "legacy_id":"72",
    "station_id":"72",
    "eighthd_has_key_dispenser":false}
]}
```


- Each marker is placed at the latitude and longitude returned by request.
- When a marker is clicked, a popup appears displaying the station name and capacity.
- These responses includes: `name`, `station` and `capacity` of each station.

Instructor Do: Introduce Citi Bike

Advance Version:

➔ Citi Bike API Station Information + Status Endpoint.




```
d3.json("https://gbfs.citibikenyc.com/gbfs/en/station_information.json", function(infoRes) {
 d3.json("https://gbfs.citibikenyc.com/gbfs/en/station_status.json", function(statusRes) {
   var updatedAt = infoRes.last_updated;
   var stationStatus = statusRes.data.stations;
   var stationInfo = infoRes.data.stations;
   var stationCount = {
     COMING_SOON: 0,
     EMPTY: 0,
     LOW: 0,
     NORMAL: 0,
     OUT_OF_ORDER: 0
   };
```

- This version groups markers into layers according to station status.
- When a marker is clicked, a popup appears displaying the station name, capacity and bikes available.
- These responses includes: `name`, `station` and `capacity` of each station.

Groups Do: Create Citi Bike Maps

In this activity, you and your group will work with the Citi API to build a map of all the Citi Bike stations and their status.

Suggested Time:
30 Minutes

Instructions:

Groups Do: Create Citi Bike Maps

- **Basic Version**

  1. Use the Citi Bike station information endpoint to get information about the station names and locations. In your browser, take a moment to study the data that the endpoint sends back. Note the following:
     - Each object in the `stations` array has `station_id`, `name`, `capacity`, lat, and lon properties.
     - The logic.js file contains coordinates that you can use to position a Leaflet map over New York City.
  2. Create a function named `createMap` that takes `bikeStations` as an argument. This function will create both the tile layer and an overlay with the pins for each station.
  3. Create a second function named `createMarkers` that will take response as an argument. This function will do the following:
     - Using the response from a future d3 call, loop through the stations, and create a marker to represent each station.
     - Give each marker a popup to display the name and capacity of its station.
  4. In the `createMarkers` function, pass the resulting bike markers to the `createmap` function as a `layerGroup`.
  5. Using d3, retrieve json data from the Citi Bike station information endpoint, and call the `createMarkers` function.

Instructions:

Groups Do: Create Citi Bike Maps

- **Advance Version**
  1. Write code to perform a second API call to the Citi Bike station status endpoint. Take a few moments to study the data that the endpoint returns. In particular, notice `station_id`, `num_bikes_available`, `is_installed`, and `is_renting`.
  2. Using the data that you got from the second API call, try to add the following functionality:
     - In the popup for each marker, display the number of available bikes.
     - Add a layer control, and split the markers into the following layer groups:
       i. **Coming Soon:** This applies if a station isn't yet installed.
       ii. **Empty Stations:** This applies if a station has no available bikes.
       iii. **Out of Order:** This applies if a station is installed but not renting.
       iv. **Low Stations:** This applies if a station has less than five available bikes.
       v. **Healthy Stations:** This applies if a marker doesn't fall into any of the previous layer groups.
  3. Use a Leaflet plugin to create different types of markers to represent the layers. The following step shows an example map that uses Leaflet.ExtraMarkers. However, feel free to use another plugin if you prefer.
  4. Add a legend to your map to explain the different markers.
  5. When you complete the app, deploy it to GitHub Pages.

Instructions:

Groups Do: Create Citi Bike Maps

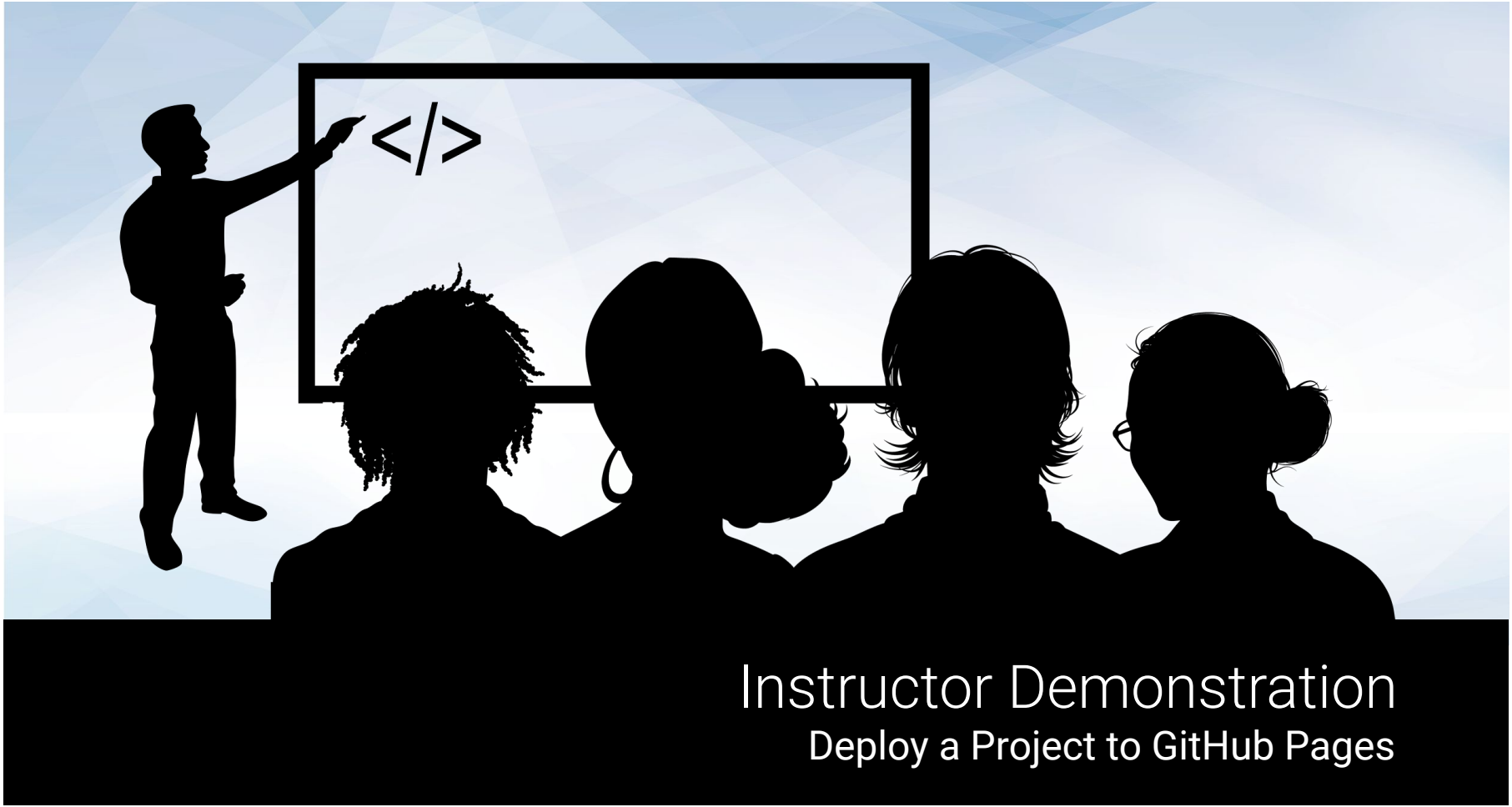
- **Hints**:
  - Make sure that you run `python -m http.server` in the folder that contains your files. Because you'll do all the work on the front end of your app, you won't need to restart the router for any changes that you make.
  - Here are some helpful links:
    - Leaflet map example
    - Citi Bike station information API endPoint
    - Leaflet popup documentation
    - Citi Bike station status API endPoint
    - Leaflet layer groups documentation
    - Leaflet.ExtraMarkers
    - Leaflet legend documentation

Let’s Review

Countdown timer

40:00

(with alarm)



Instructor Demonstration

Deploy a Project to GitHub Pages

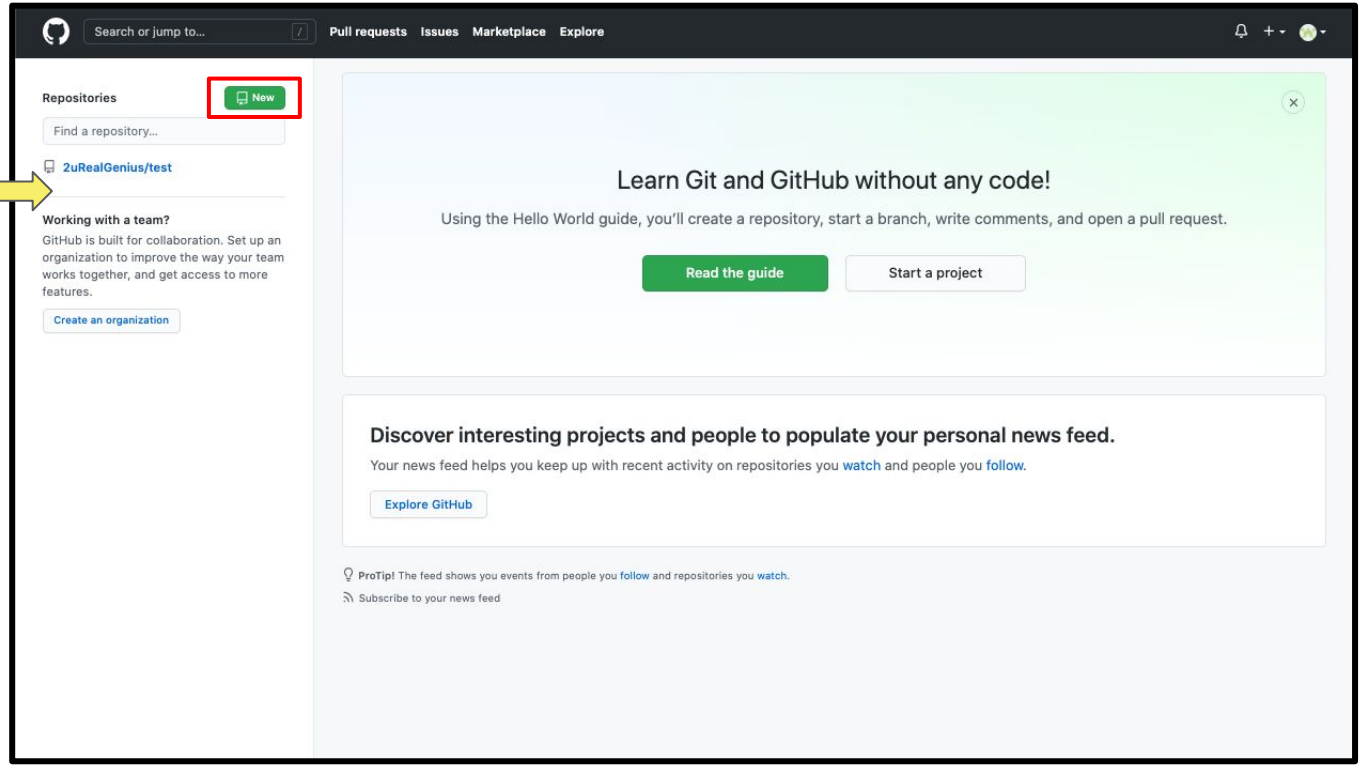
Instructor Do: Deploy a Project to GitHub Pages

1. GitHub

Navigate to http://github.com

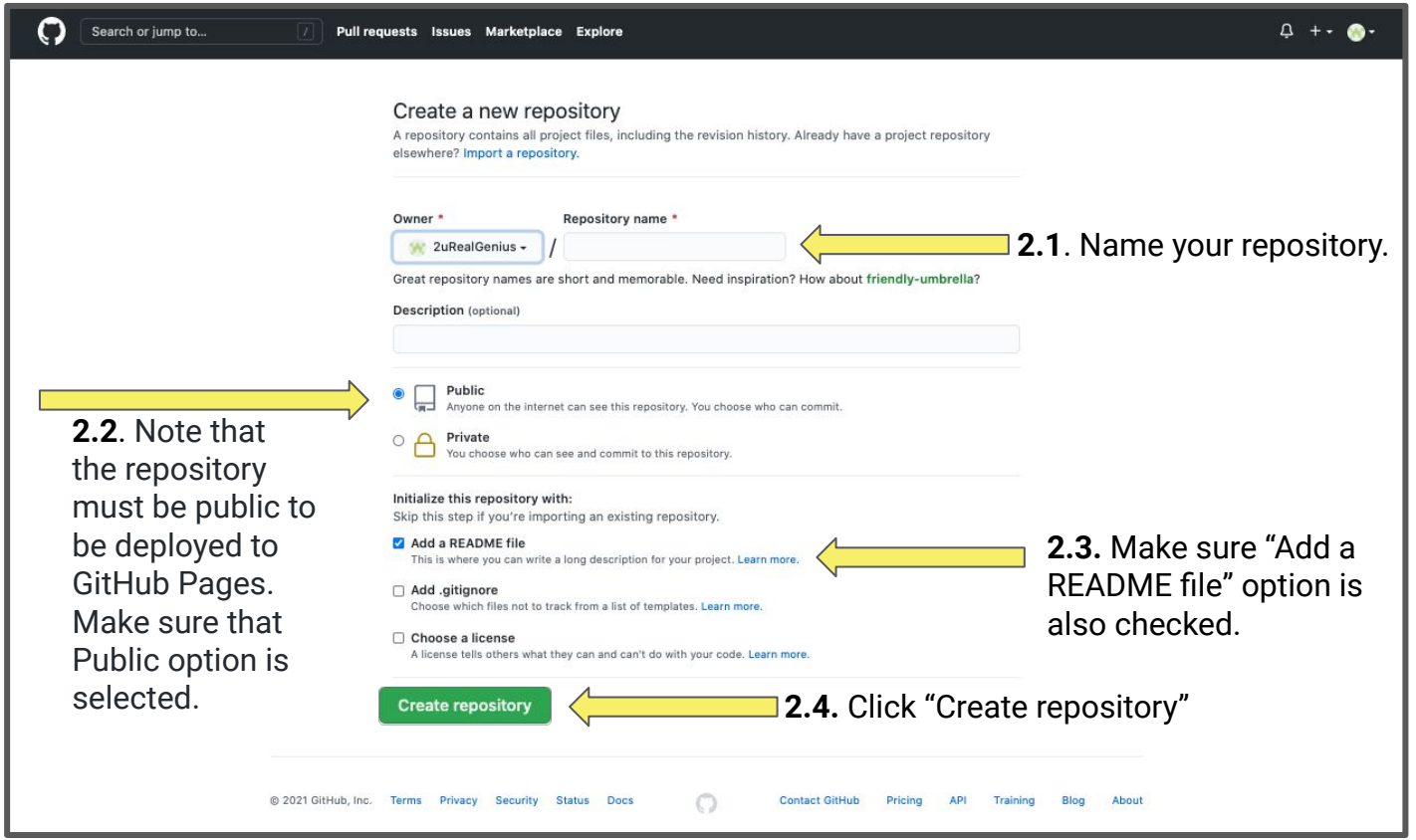
Then create a new repository

by clicking 



Instructor Do: Deploy a Project to GitHub Pages

2. GitHub



2.1. Name your repository.

2.2. Note that the repository must be public to be deployed to GitHub Pages. Make sure that Public option is selected.

2.3. Make sure “Add a README file” option is also checked.

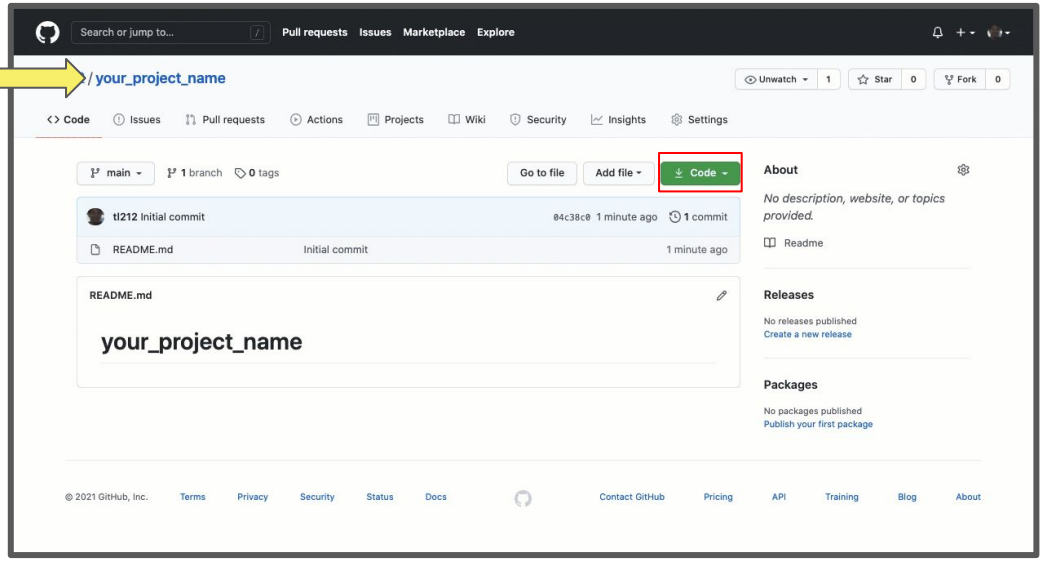
2.4. Click “Create repository”

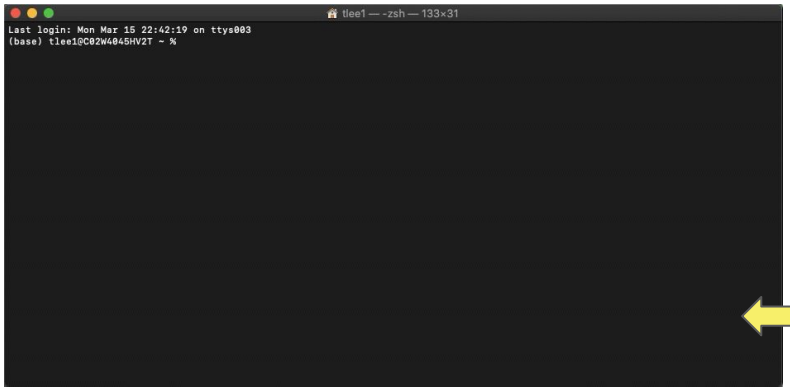
Instructor Do: Deploy a Project to GitHub Pages

3. GitHub

You will now be directed to your repository page.

Click on  to copy the URL of your repository.





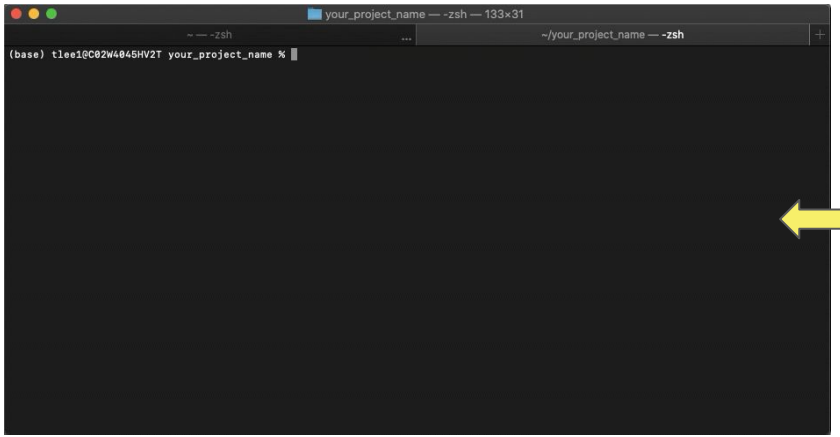
Next, open the command line and type:

```
git clone <url>
```

Instructor Do: Deploy a Project to GitHub Pages

4. GitHub

Now that we have your repository in GitHub and also cloned to your local machine, copy and paste the HTML, JavaScript, and JSON files from the `Solved` directory to local repository.



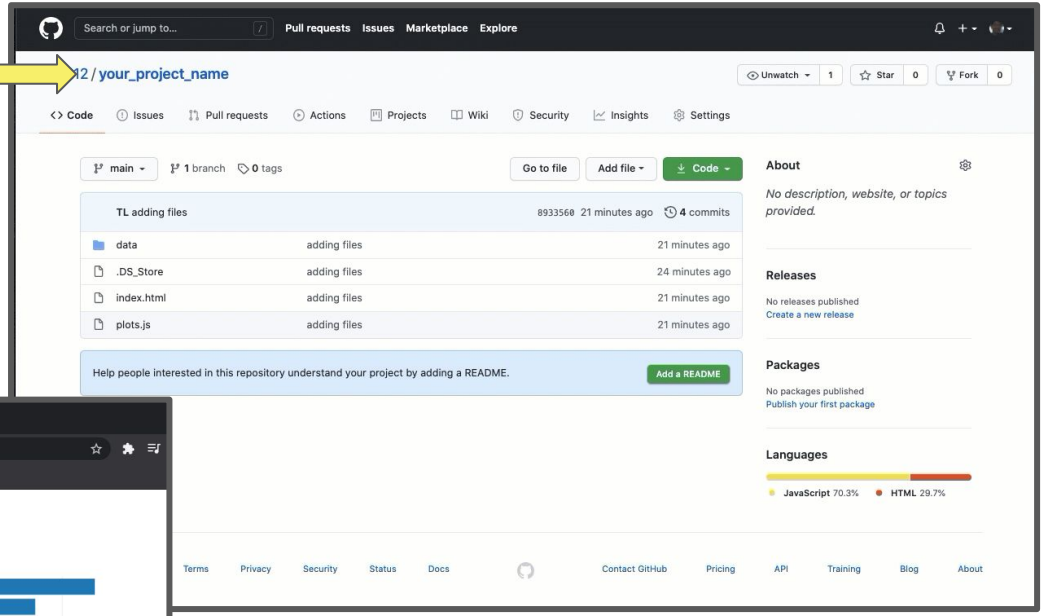
Once you had pasted the files to your local repository. Open CLI to push the changes made it by typing:

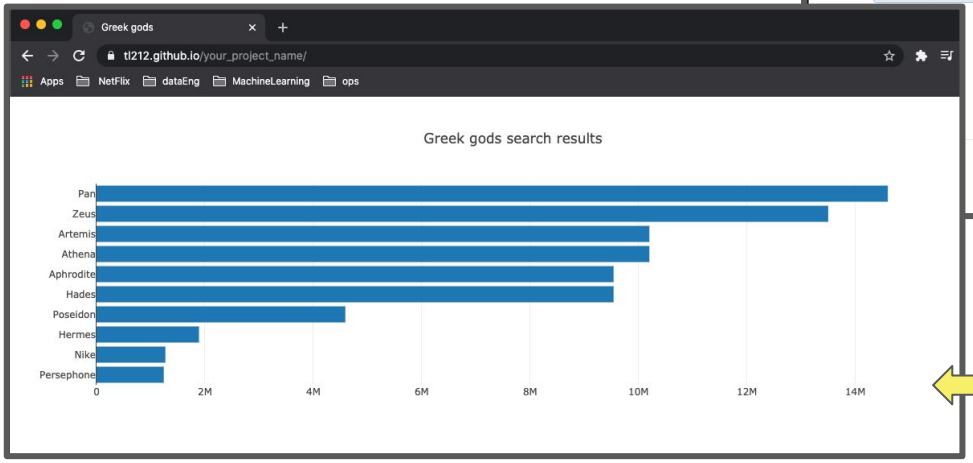
```
git add .
git commit -m ‘your commit msg’
git push origin main
```

Instructor Do: Deploy a Project to GitHub Pages

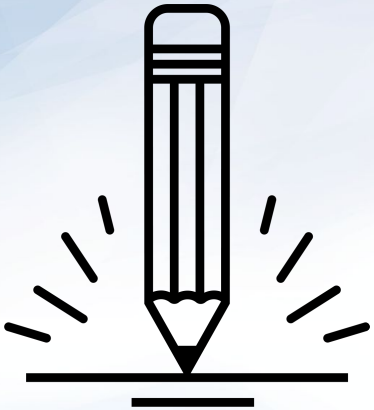
5. GitHub

Navigate back to your GitHub repository page. Under Settings, go to **GitHub Pages**, and then in the **Select source list**, select **main branch** and click save.





The project should now be deployed to GitHub Pages, as the following.



Activity: Deploy the Citi Bike Project

In this activity, you will deploy a Plotly project with a local data file to GitHub Pages.

Suggested Time:
20 Minutes

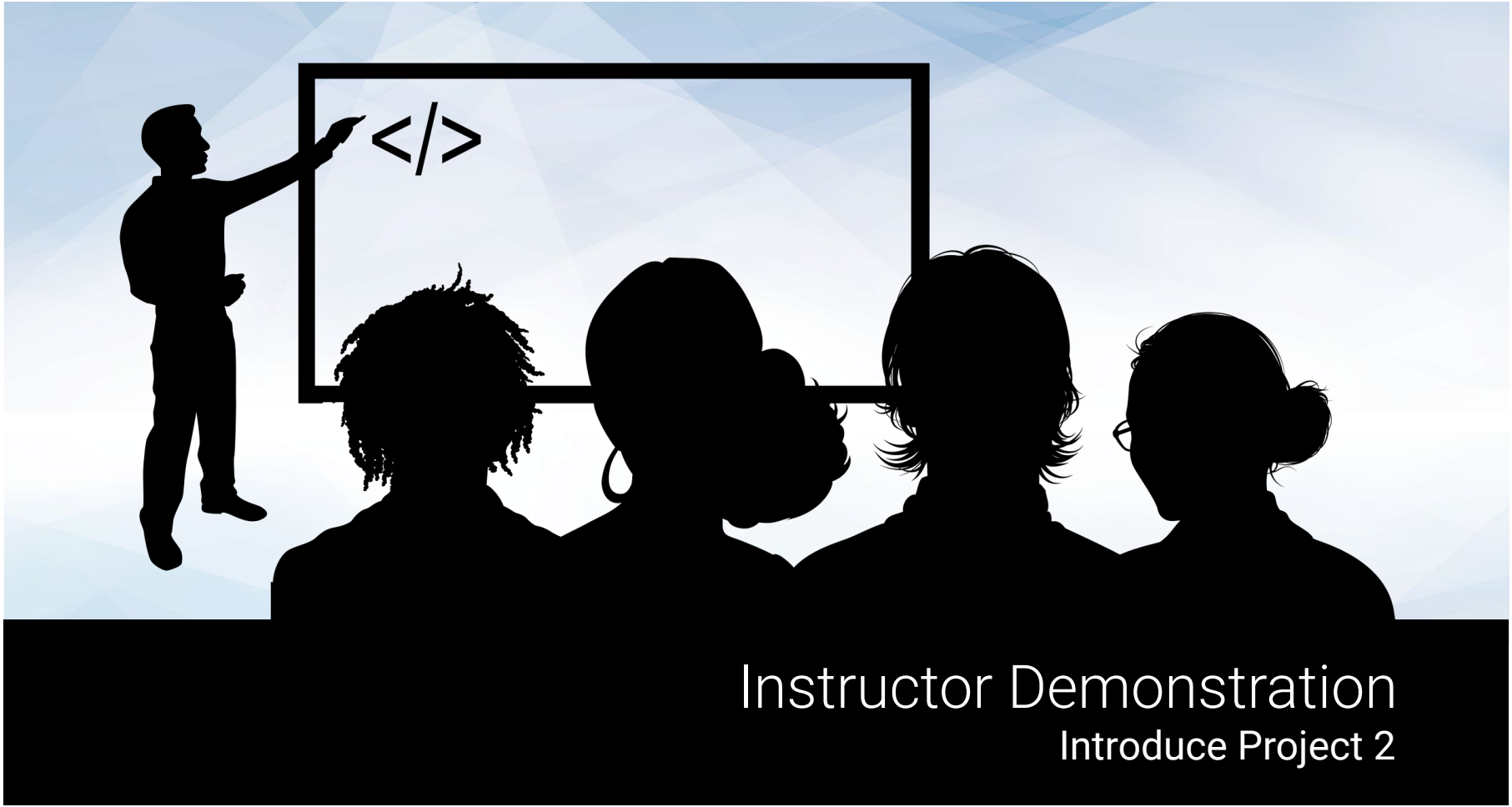
Instructions:

Activity: Deploy the Citi Bike Project

1. Note that you've been given a Plotly visualization project with `index.html`, `plot.js`, and `data.json`.
2. Deploy the project to GitHub Pages.

- **Hints**:
  - Consult GitHub Pages for reference. Be sure to select the Project Site and Start from Scratch options for instructions.

Let's Review



Instructor Demonstration

Introduce Project 2

➔ Project 3: Visualize Me, Captain!


Tell
a story
with data!

Project Requirements

Project Description

01 Your task is to **tell a story** data visualizations.

02 Focus on providing users an **interactive means** to explore data themselves.

03 Prepare a **10-minute presentation** that lays out your theme, coding approach, data munging techniques, and final visualization.

04 You may choose a project of any theme, but we encourage you to **think broadly**.

05 You will have **ample time in class** to work with your group, but expect to put in **hours outside of class** as well.

Specific Requirements

1. Your visualization must include a Python Flask–powered API, HTML/CSS, JavaScript, and at least one database (SQL, MongoDB, SQLite, etc.).
2. Your project should fall into one of the below four tracks:
    - A custom “creative” D3.js project (i.e., a nonstandard graph or chart)
    - A combination of web scraping and Leaflet or Plotly
    - A dashboard page with multiple charts that update from the same data
    - A “thick” server that performs multiple manipulations on data in a database prior to visualization (**must be approved**)
3. Your project should include at least one JS library that we did not cover.
4. Your project must be powered by a data set with at least 100 records.
5. Your project must include some level of user-driven interaction (e.g., menus, dropdowns, textboxes).
6. Your final visualization should ideally include at least three views.

Schedule

Weekly Schedule

Day 1 (Next Class):
Start brainstorming topics with your group and researching potential data sets. Your focus should center around:

- Selecting a topic
- Finding a data set
- Finding inspiration
- “Sketching” your ideal visuals
- Creating a 1-page proposal

Day 2:
You will need to create a 1-page proposal that includes:

- A brief articulation of your chosen topic and rationale
- A link to your data set(s) and a screenshot of the metadata if it exists.
- 3 or 4 screenshots of relevant, “inspiring” visualizations that frame your creative fodder
- A sketch of the final design
- A link to the primary GitHub repository you’ll be housing your work in

Day 3:
Project Work

Final Thoughts

01

Project week is a great time to tie up loose ends, both with your group and on your own.

02

If there are topics you'd like to review, shoot me and the TAs a message. We're happy to do (recorded) extra review sessions for small groups during these weeks.

03

Good luck and have fun!

Questions?

Project 2

Kickstart