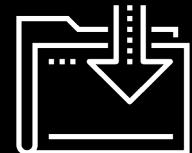




Citi Bike Project with Leaflet and Intro to Projects

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 visualisations to GitHub Pages.



Draft a project proposal in a team setting.

Overview Of Your Career Resources



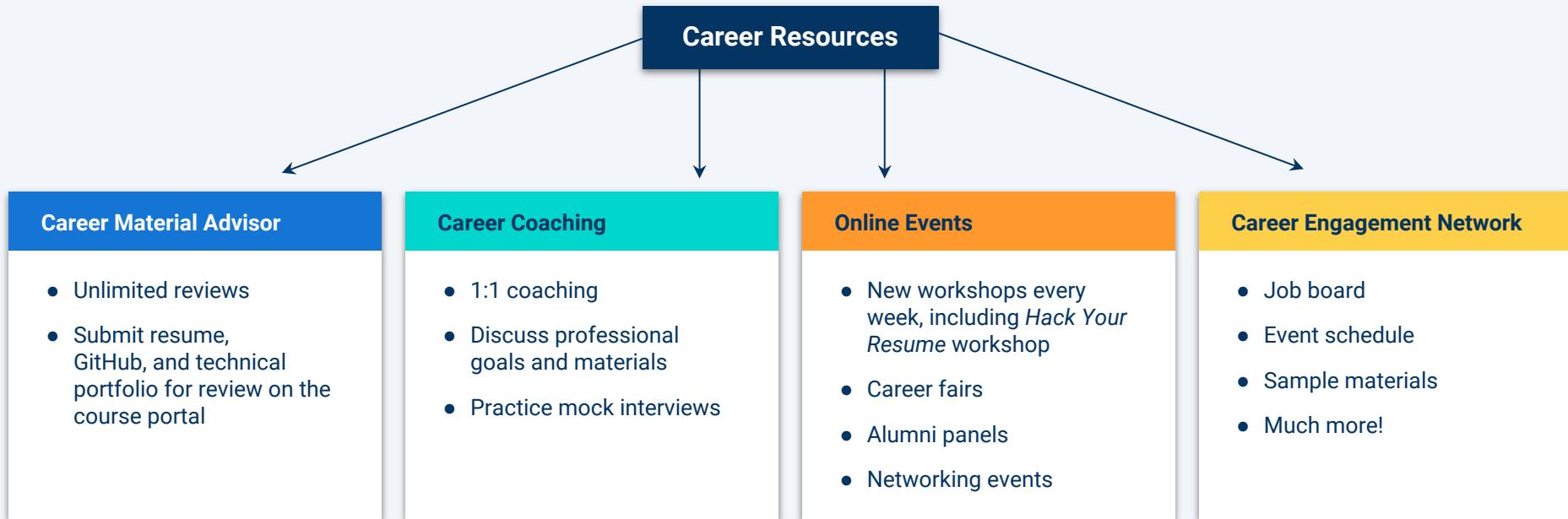
What are your career goals?

In the chat indicate your post-bootcamp career goals.

- +1 If you want to find a new job.
- +2 If you want a promotion or salary increase.
- +3 If you want to start a business.

Suggested Time:
1 minute

Your Career Resources



After your resume is approved by a Career Material Advisor you will be matched with your Career Coach. Submit your resume via the Career Services tab on the course portal.

Working With Your Career Coach

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

Topics include:



Applying and networking



Salary negotiation



Gaining traction to land interviews



Motivation and more!



Conducting mock interviews

Working With Your Career Coach

You have two options:

01

1:1 scheduled bi-monthly recurring coaching calls

02

Reaching out to your Career Coach when needed



We recommend scheduled Recurring Calls. Why?

The data shows that our students who have professional application materials and participate in recurring calls are much more likely to secure the jobs they want.

Working With Your Career Coach

Next Steps:

01

Visit the Career Engagement Network (careernetwork.2U.com) and explore the resources available to you.

Definitely check out the virtual workshops and events!

02

Get your resume approved by a Career Material Advisor by submitting it via the Career Services tab in the course portal.

This will grant you access to your Career Coach!

03

Schedule a 1:1 meeting with your Career Coach!



Instructor Demonstration

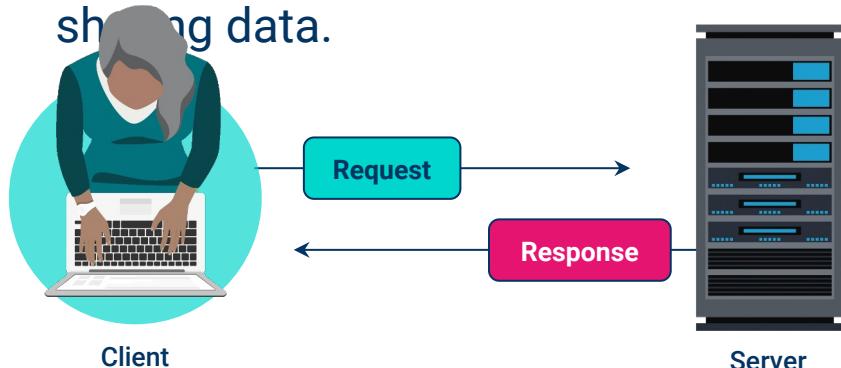
Use the Python HTTP Server

Use the Python HTTP Server

Here are some things to note as we live-code:

A server

A server is a program or device that performs actions such as processing and sharing data.



Cross-Origin Resource Sharing

Cross-Origin Resource Sharing (CORS) is a mechanism that tells browsers to access selected resources from a web server through information in the HTTP headers in a web application.

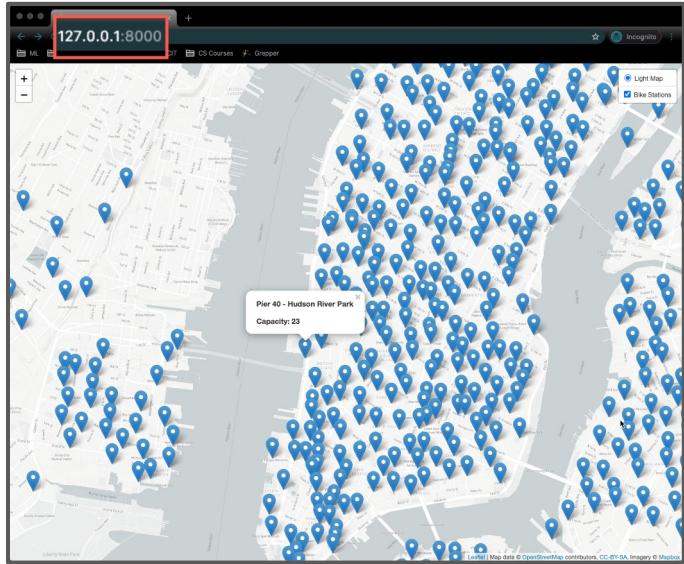
CORS provides a way to allow cross-origin requests.

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

Create Citi Bike Maps

Instructor Do: Introduce Citi Bike

Basic Version



→ Citi Bike API Station Information Endpoint

```
3d.json("https://gbfs.citibikenyc.com/gbfs/en/station_information.json", createMarkers);
```

→ One KVP (key value property) of the JSON

```
{"stations": [
    {"station_type": "classic",
     "lon": -73.9932888,
     "region_id": "71",
     "lat": 40.76727216,
     "rental_url": "http://app.citibikenyc.com/S6Lr/IBV092JufD?station_id=72",
     "name": "52 St & 11 Ave",
     "short_name": "626.01",
     "rental_methods": ["CREDITCARD", "KEY"],
     "electric_bike_surcharge_waiver": false,
     "external_id": "660b237e-0ca1-11e7-82f6-3863bb44ef7c",
     "eightd_station_services": [],
     "capacity": 55,
     "has_kiosk": true,
     "legacy_id": "72",
     "station_id": "72",
     "eightd_has_key_dispenser": false}
]}
```

- Each marker is placed at the latitude and longitude returned by the request.
- When someone clicks a marker, a popup displays the station name and capacity.
- These responses include the 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 statuses.

Suggested Time:
30 minutes



Groups Do: Create Citi Bike Maps

Instructions:

- Basic Version

1. Use the [Citi Bike station information endpoint](#) to get information about the station names and locations. Take a moment to study the data that the endpoint sends back in your browser. 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.
 - 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.

Groups Do: Create Citi Bike Maps

Instructions:

- **Advanced 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 returned by the second API call, 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.

Groups Do: Create Citi Bike Maps

Instructions

- **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 after making changes.
- 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)

Break



Instructor Demonstration

Deploy a Project to GitHub

Pages

Instructor Do: Deploy a Project to GitHub Pages



Navigate to <http://github.com>,

then create a new repository

by clicking



The screenshot shows the GitHub homepage. At the top, there is a search bar and navigation links for "Pull requests", "Issues", "Marketplace", and "Explore". Below the search bar, there is a "Repositories" section with a "New" button highlighted with a red box. A yellow arrow points from the "New" button on the left towards the highlighted "New" button on the page. To the right of the "Repositories" section, there is a large callout box with the text "Learn Git and GitHub without any code!" and "Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request." It includes two buttons: "Read the guide" and "Start a project". Further down the page, there is another section titled "Discover interesting projects and people to populate your personal news feed." with a "Explore GitHub" button and a "ProTip!" note about news feeds.

Instructor Do: Deploy a Project to GitHub Pages

2. GitHub

The screenshot shows the GitHub interface for creating a new repository. At the top, there's a search bar and navigation links for Pull requests, Issues, Marketplace, and Explore. Below that, the main title is "Create a new repository". A sub-instruction says: "A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository." The form fields include "Owner" (set to "2uRealGenius") and "Repository name" (empty). A yellow arrow points from the text "2.1. Name your repository." to the "Repository name" field. The "Description (optional)" field is empty. The "Visibility" section shows "Public" (selected) and "Private" (unchecked). A yellow arrow points from the text "2.2. Note that the repository must be public to be deployed to GitHub Pages. Make sure that the Public option is selected." to the "Public" radio button. Below that, under "Initialize this repository with:", there are three options: "Add a README file" (checked), "Add .gitignore" (unchecked), and "Choose a license" (unchecked). A yellow arrow points from the text "2.3. Make sure the 'Add a README file' option is also checked." to the "Add a README file" checkbox. At the bottom is a green "Create repository" button. A yellow arrow points from the text "2.4. Click the Create repository button." to the "Create repository" button. The footer includes copyright information (© 2021 GitHub, Inc.) and links for Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.

2.1. Name your repository.

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

2.3. Make sure the "Add a README file" option is also checked.

2.4. Click the Create repository button.

Instructor Do: Deploy a Project to GitHub Pages



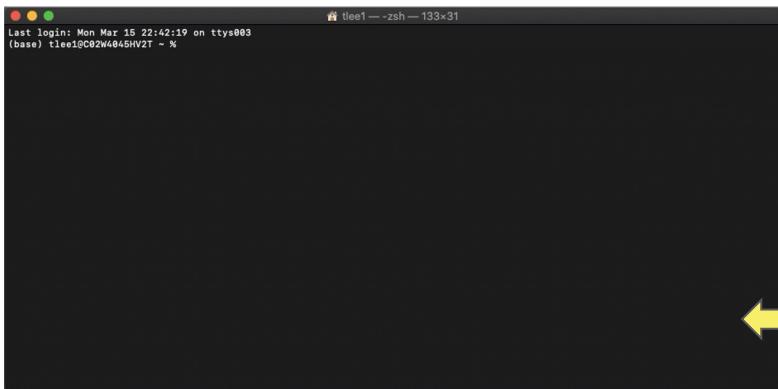
You will now be directed to your repository page.

Click on **Code** to copy the URL of your repository.

A screenshot of a GitHub repository page for a project named 'your_project_name'. The URL in the address bar is '/your_project_name'. A yellow arrow points from the text above to the repository name. A red box highlights the 'Code' dropdown menu in the top right corner of the page. The page shows one commit, 'tl1212 Initial commit', and a README.md file. The repository has 1 branch, 0 tags, and 1 commit. The 'About' section notes 'No description, website, or topics provided.' and 'Readme'. The 'Releases' section says 'No releases published' and 'Create a new release'. The 'Packages' section says 'No packages published' and 'Publish your first package'. The bottom of the page includes links for GitHub, Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.

Next, open the command line and type:

```
git clone <url>
```



Instructor Do: Deploy a Project to GitHub Pages



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

```
your_project_name -- zsh -- 133x31
(base) tlee1@C02W4045HV2T your_project_name %
```

Once you have pasted the files to your local repository, open CLI to push the changes by typing:

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

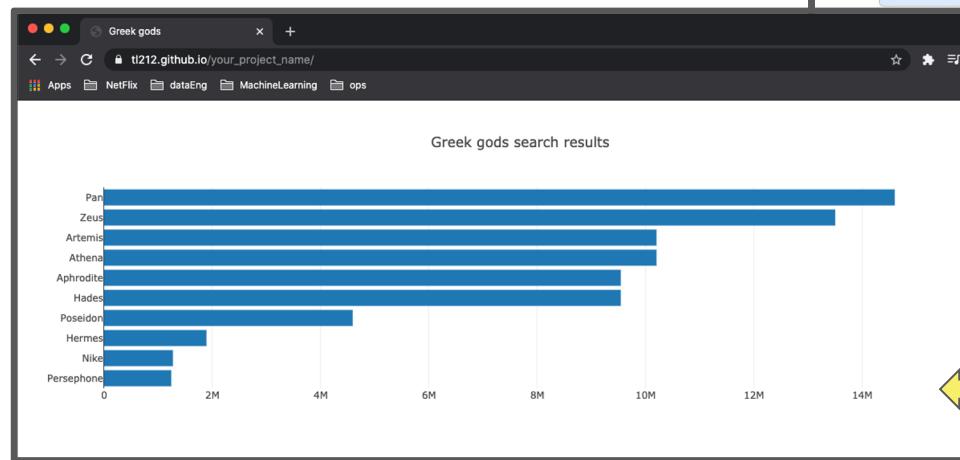


Instructor Do: Deploy a Project to GitHub Pages

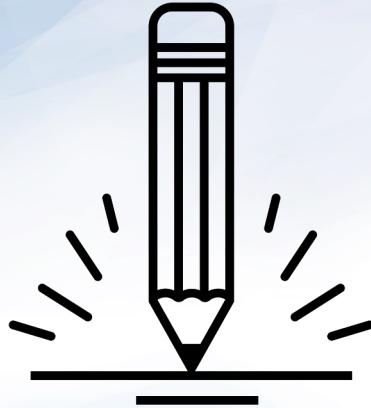
5. GitHub

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.

A screenshot of a GitHub repository page for 'tl212/your_project_name'. A yellow arrow points from the text above to the repository URL in the address bar. The page shows recent commits by 'TL adding files' and a file tree with 'index.html' as the root. On the right, there's an 'About' section with a message about no description, website, or topics provided, and a 'Releases' section with a message about no releases published. Below the repository details, there's a 'Languages' bar showing JavaScript at 70.3% and HTML at 29.7%.



The project should now be deployed to GitHub Pages, as in 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



Activity: Deploy the Citi Bike Project

-
1. Note that you've been given a Plotly visualisation 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 3

Project Requirements

Project Description

01

Your task is to **tell a story** with data visualisations.

02

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

03

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

04

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

05

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

Specific Requirements

1. Your visualisation must include a Python Flask-powered API, HTML/CSS, JavaScript, and at least one database (PostgreSQL, MongoDB, SQLite, etc.).
2. Your project should fall into one of the below three tracks:
 - 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 visualisation (**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 dataset with at least 100 records.
5. Your project must include some level of user-driven interaction, such as menus, dropdowns, and textboxes.
6. If possible, your final visualisation should include at least three views.

Schedule

Project Schedule

Day 1 (Next Class)

Start brainstorming topics with your group and researching potential datasets. Your focus be:

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

Day 2

You will need to create a one-page proposal that includes:

- A brief articulation of your chosen topic and rationale
- A link to your dataset(s) and a screenshot of the metadata, if it exists.
- Three or four screenshots of relevant, “inspiring” visualisations that show your creative ideas
- A sketch of the final design
- A link to the primary GitHub repository where you’ll be housing your work

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, send 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?