GENERAL ASSEMBLY

JAVASCRIPT DEVELOPMENT

Sasha Vodnik, Instructor

HELLO!

1. Pull changes from the `JS-SF-14-resources` repo to your computer
2. Open the `13-feedr-lab` folder in your code editor

IN-CLASS LAB: FEEDR

LEARNING OBJECTIVES

At the end of this class, you will be able to

- Familiarize yourself with the API documentation for news sources.
- Fork and clone your starter code.
- Strategize ways to hide the loader and replace the content of the `#main` container with that of the API.
- Integrate string and variable values using template literals

AGENDA

- Project 2 overview
- Template literals
- Project 2 lab time

WEEKLY OVERVIEW

WEEK 7	Advanced APIs / Project 2 lab
WEEK 8	Prototypal inheritance / Closures & `this`
WEEK 9	CRUD & Firebase / Deploying your app

EXIT TICKET QUESTIONS

1. Can API's be used for simpler code?
2. (for webcast) Might be better to have two feeds, one for whats on the projector computer and another for the normal feed.
3. iterating through the photo api, a bit more in depth

EXERCISE — PROJECT PLANNING

EXERCISE

TYPE OF EXERCISE

- Group

TIMING

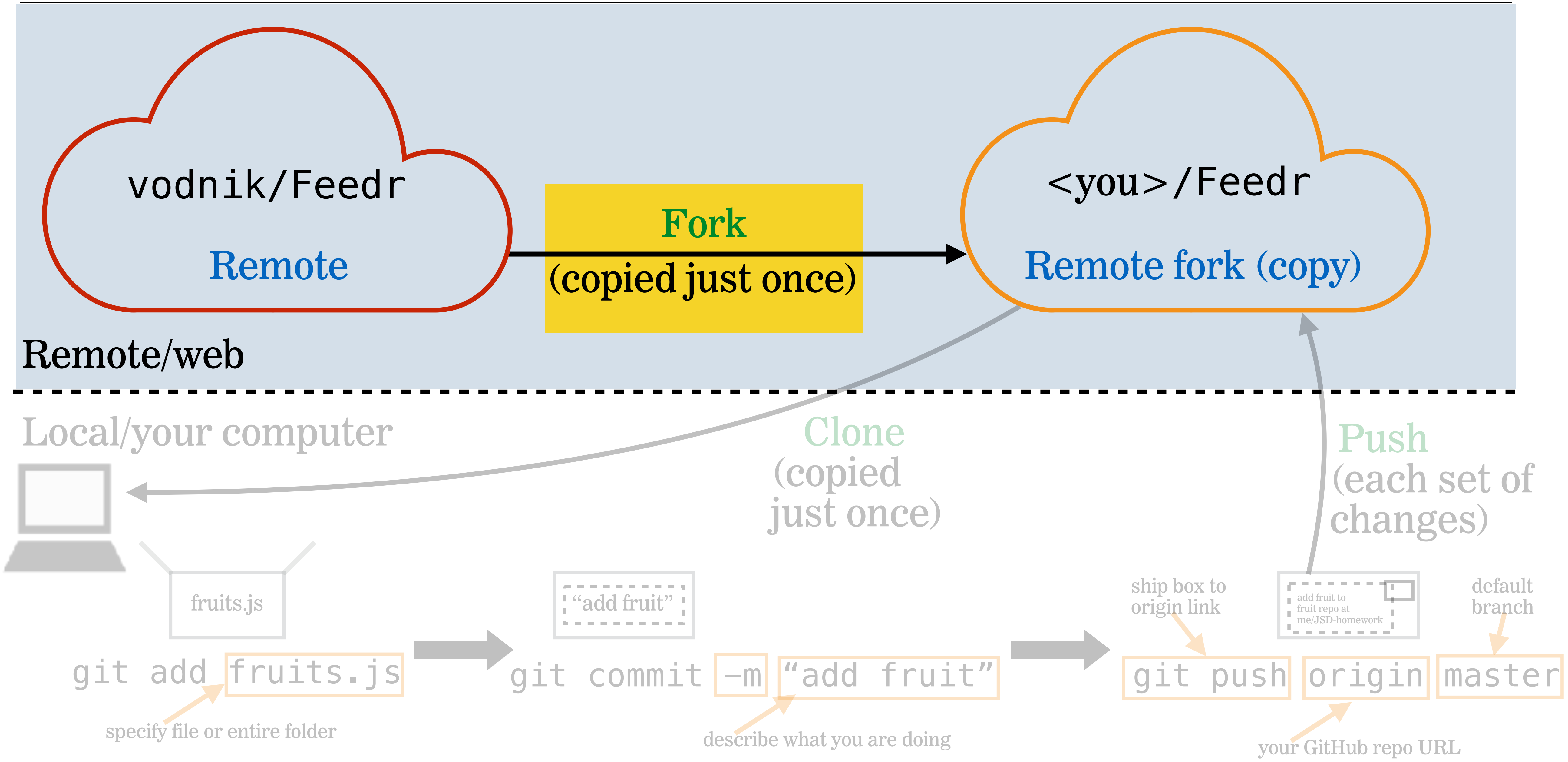
3 min

1. Think about how you approach a task with a lot of parts and steps. Jot down a list of ideas.
2. After everyone has had a chance to brainstorm individually, you will have a chance to share your ideas with the rest of the class.

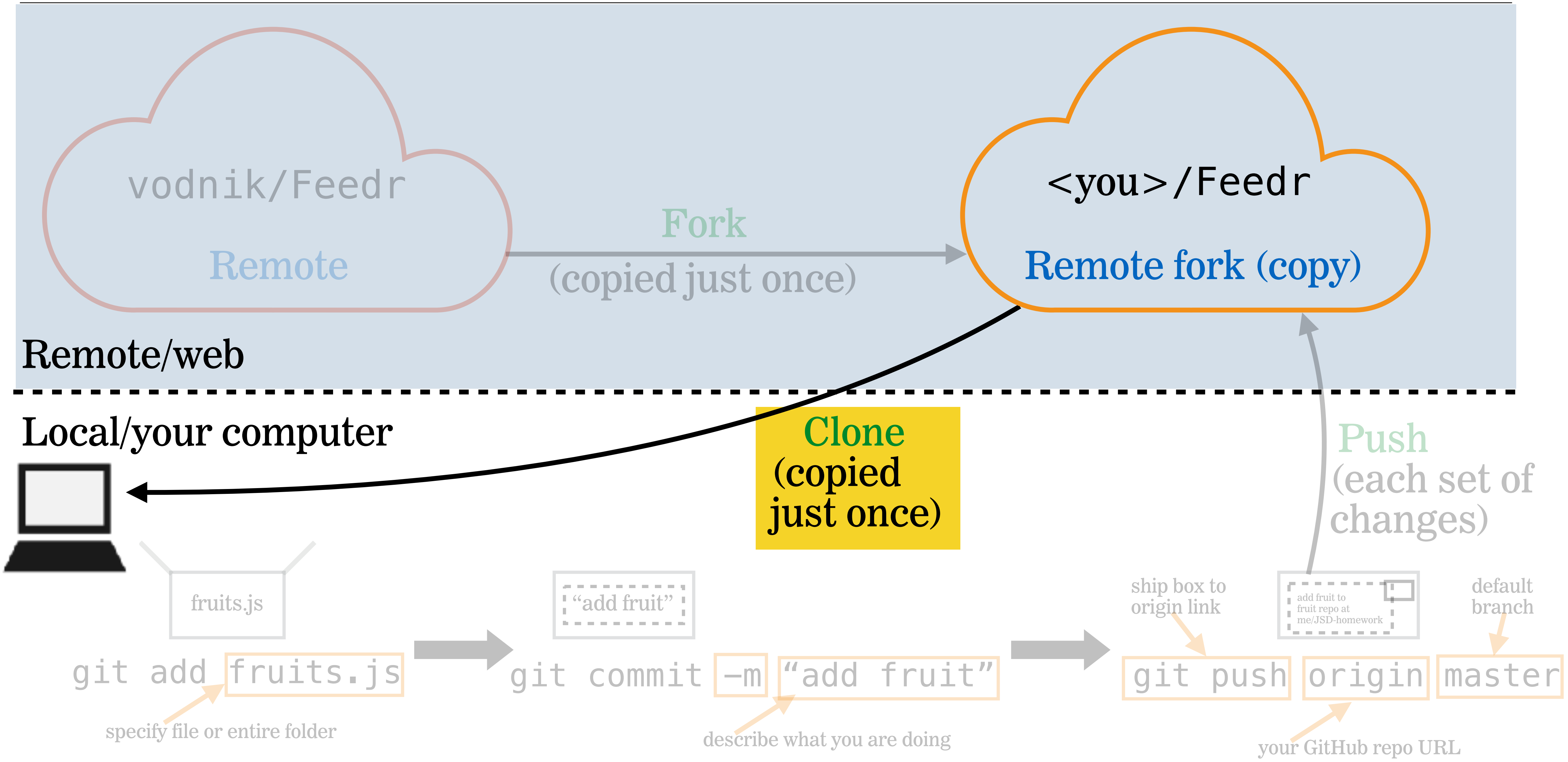
Project 2: Feedr

- **GitHub repo to fork**: https://git.generalassemb.ly/vodnik/feedr
- **Project overview & instructions**: https://pages.git.generalassemb.ly/vodnik/JSD14/pages/feedr.html

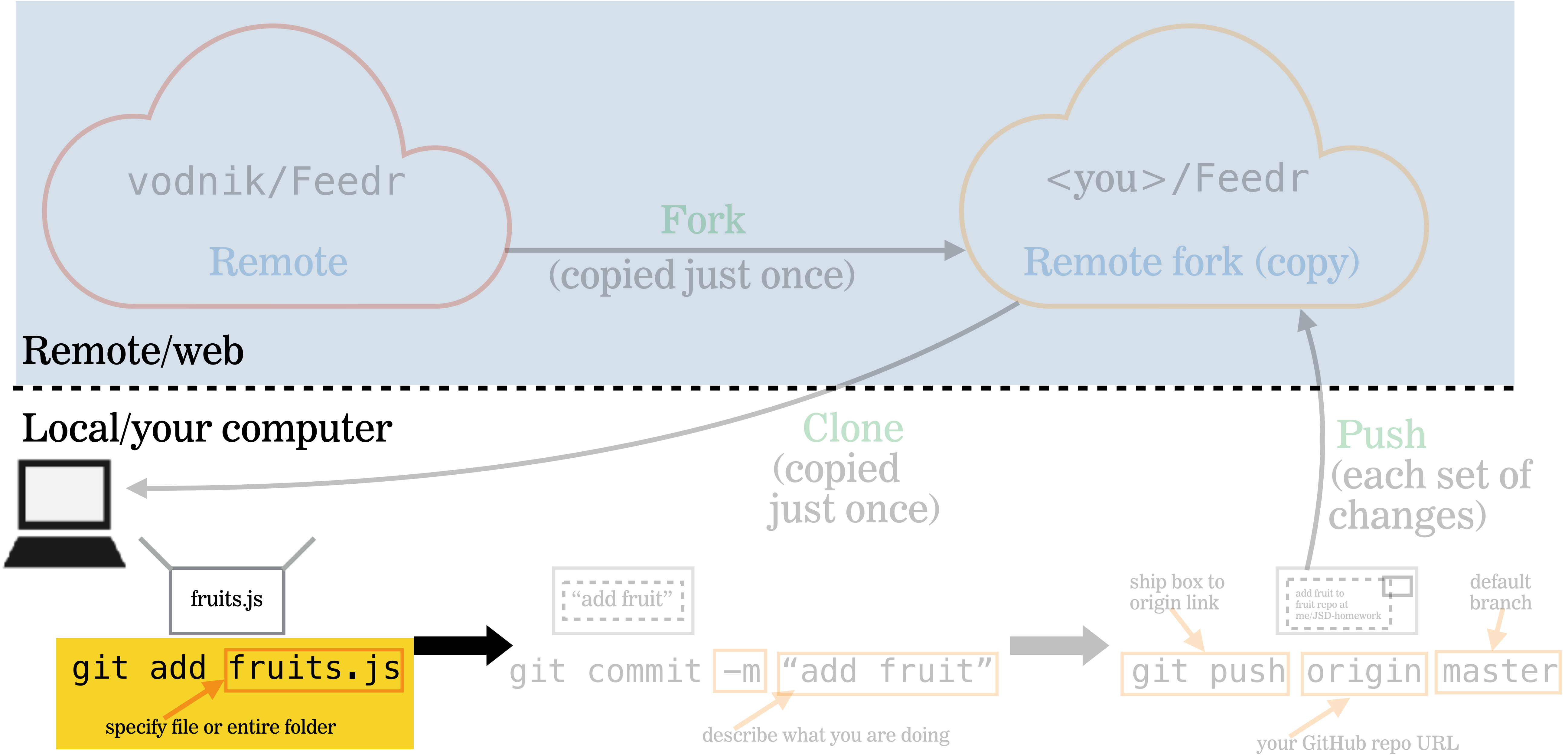
IN-CLASS LAB: FEEDR



IN-CLASS LAB: FEEDR



IN-CLASS LAB: FEEDR

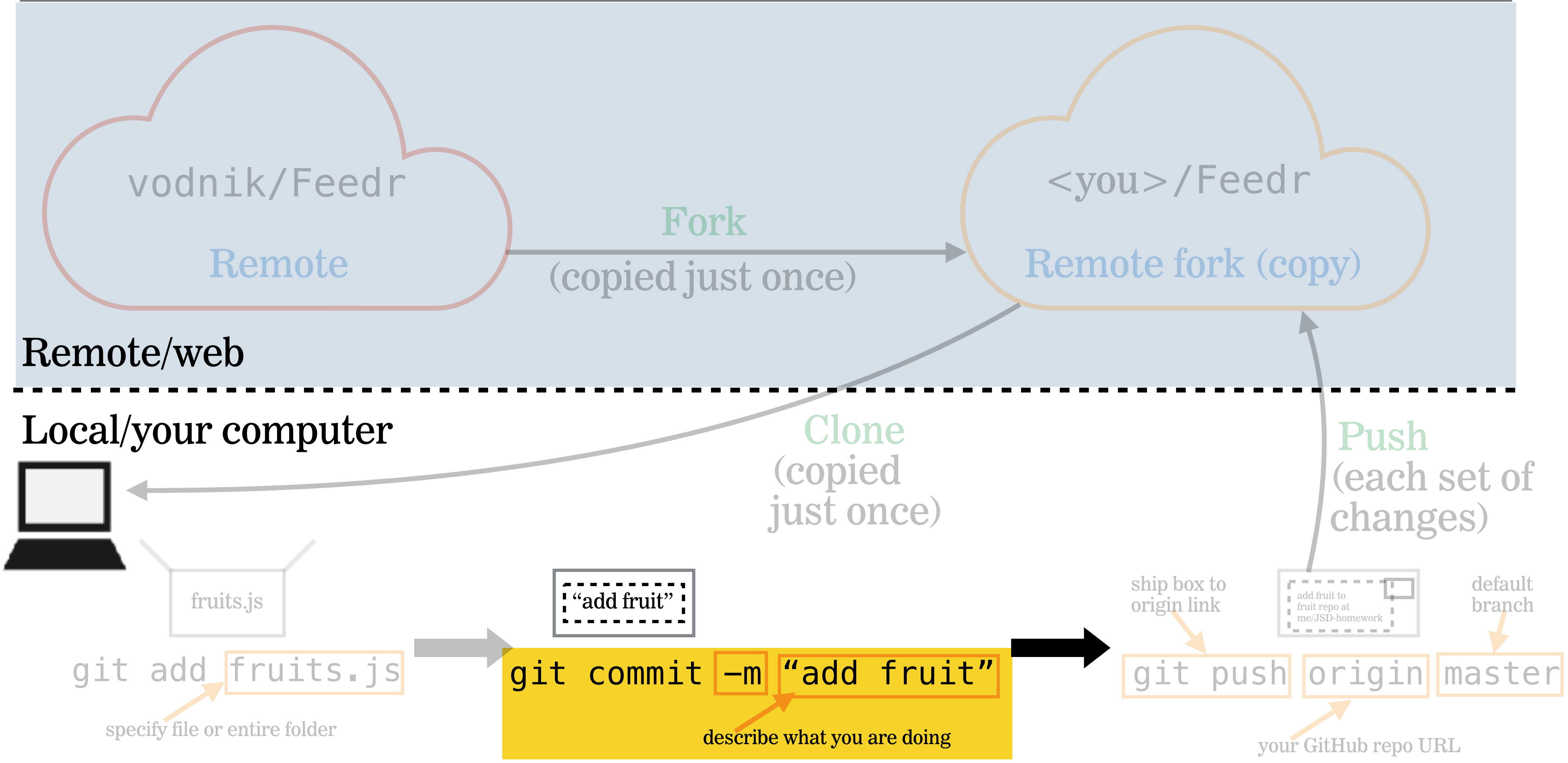


```
git add fruits.js
```

```
git commit -m “add fruit”
```

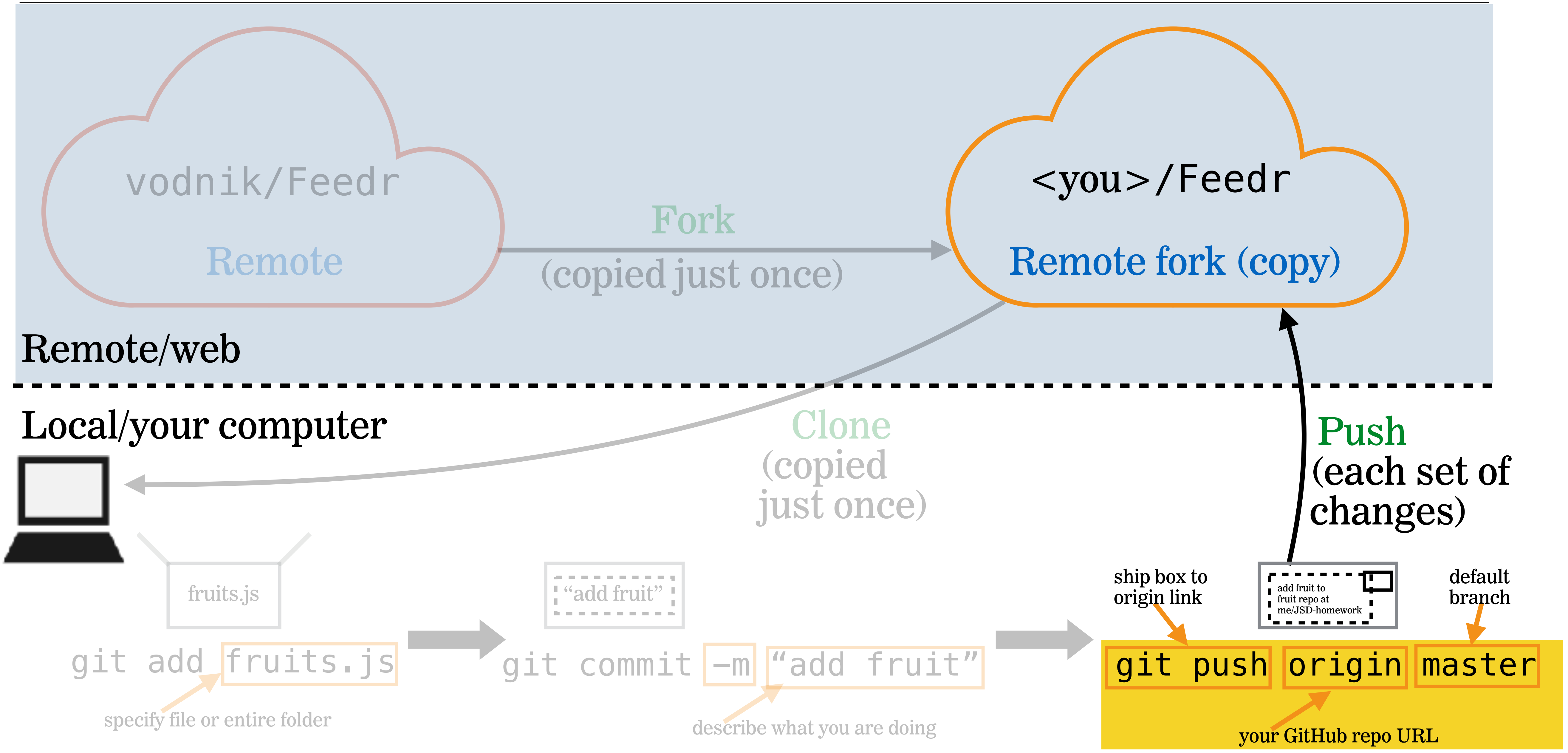
```
git push origin master
```

IN-CLASS LAB: FEEDR



```
git add fruits.js
git commit -m "add fruit"
git push origin master
```

IN-CLASS LAB: FEEDR

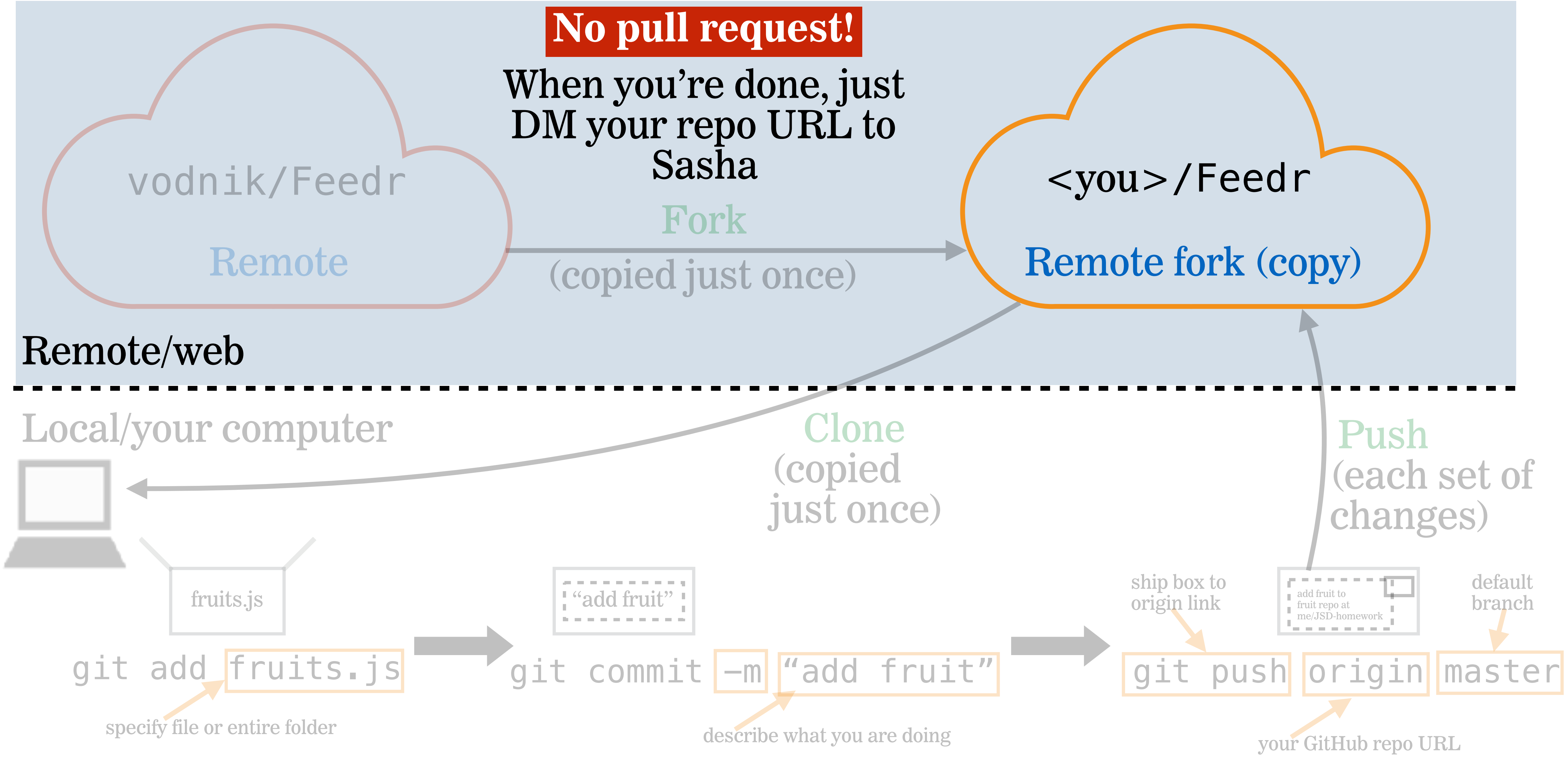


```
git add fruits.js
git commit -m "add fruit"
git push origin master
```

IN-CLASS LAB: FEEDR

No pull request!

When you're done, just DM your repo URL to Sasha



UPDATED FOLDER HIERARCHY

- JSD
  - feedr ← new folder for Project 2 is a sibling of existing folders
  - JS-SF-14-homework
  - JS-SF-14-resources
  - myhubot
  - *username.git.generalassemb.ly*

EXERCISE — FEEDR PLANNING

EXERCISE

TYPE OF EXERCISE

- Individual, then pairs

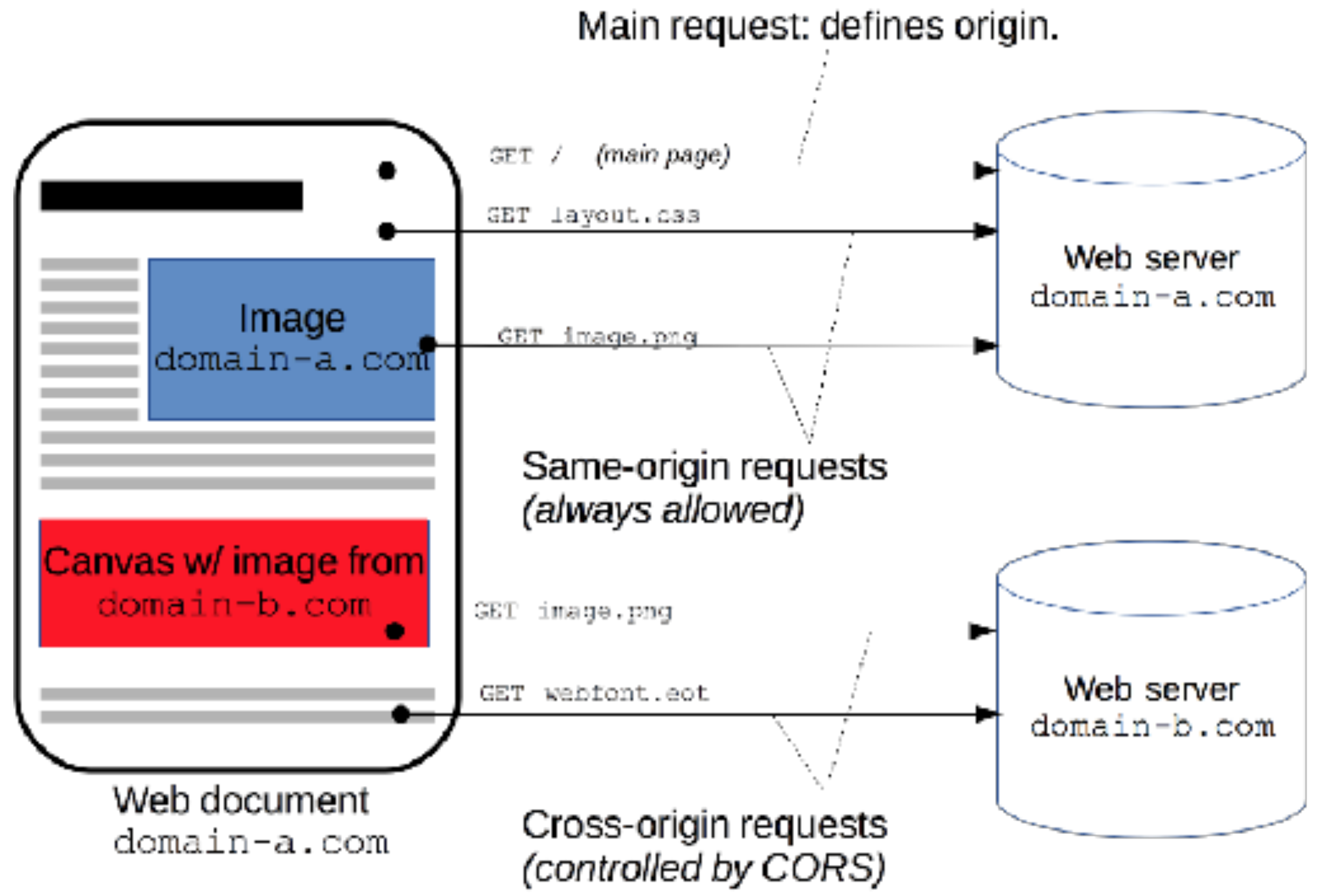
TIMING

6 min

1. Take a minute or two to decide on your next step for your Feedr project. (It’s okay to have a few possible next steps at this point.)
2. Share your next step(s) with one or two classmates. If you have different approaches, talk about how you decided on your approach.
3. Share the list of news sources you’ve selected for your project, and any pseudocode you’ve written, with your group, and discuss.

SAME-ORIGIN POLICY

CROSS ORIGIN REQUEST



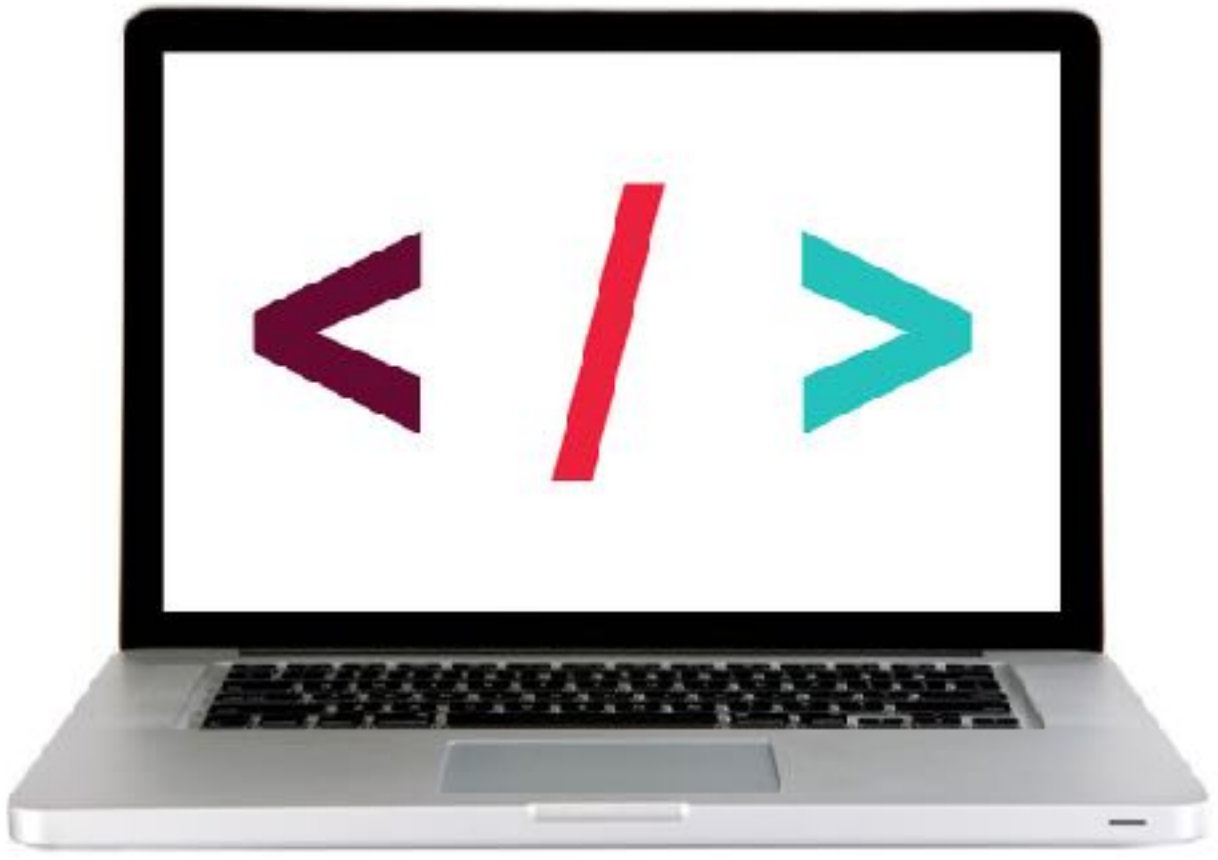
https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS

CROSS ORIGIN RESOURCE SHARING

Response header:

```
Access-Control-Allow-Origin: *
```

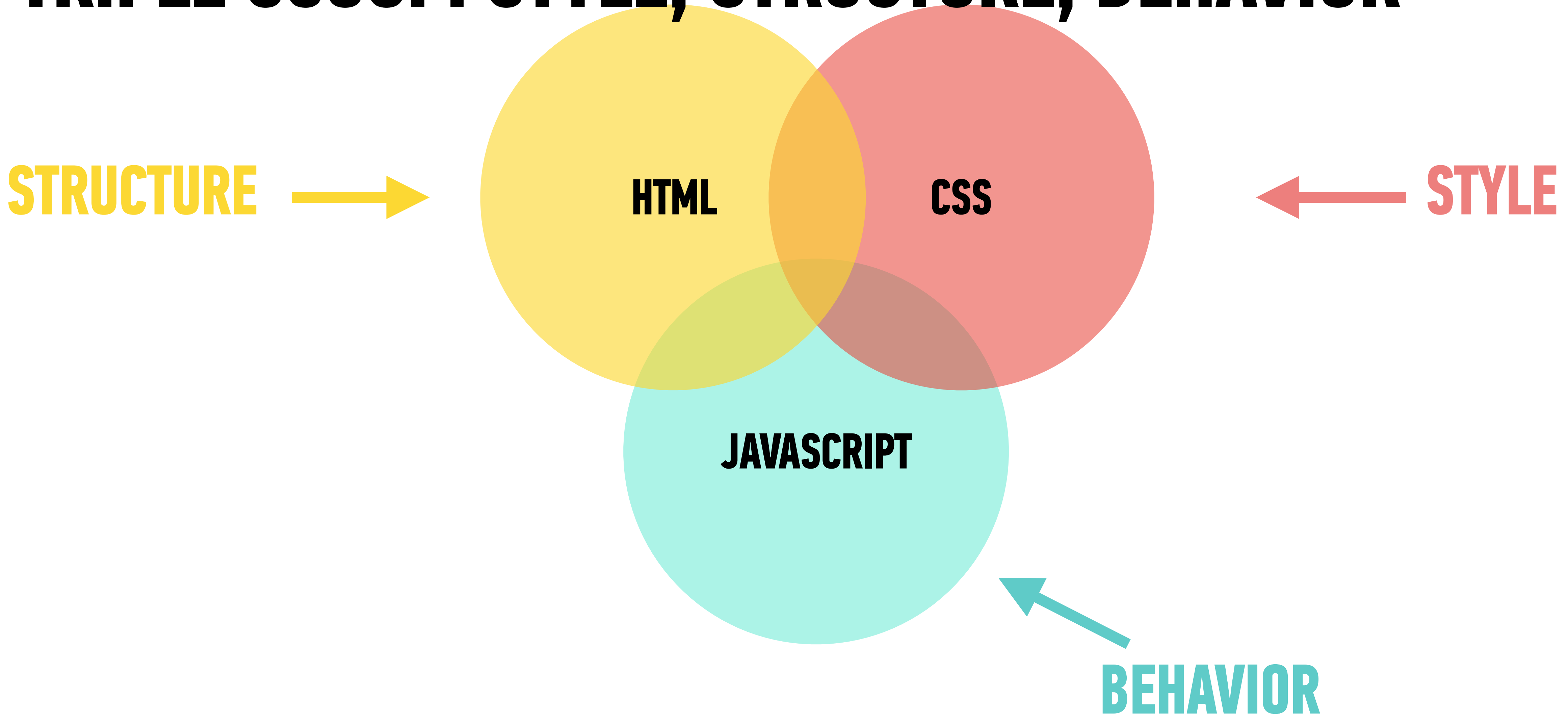
IN-CLASS LAB: FEEDR



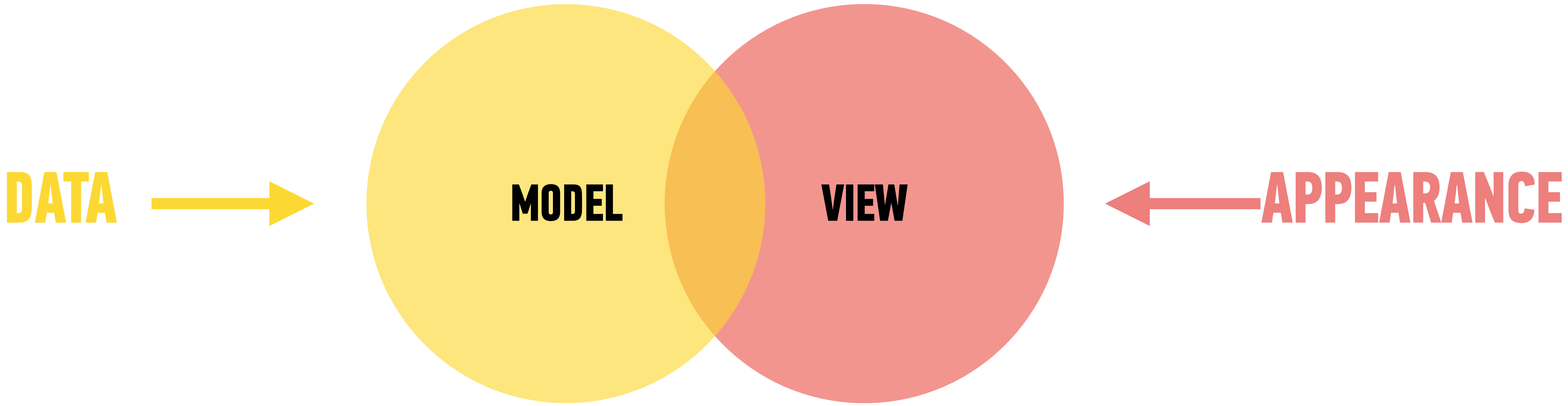
LET'S TAKE A CLOSER LOOK

TEMPLATE LITERALS

TRIPLE SCOOP: STYLE, STRUCTURE, BEHAVIOR



MODEL VS VIEW



DOM MANIPULATION

```
$resultDiv.text(degCInt + ' C / ' + degFInt + ' F');
```

TEMPLATE LITERALS

template literal starts and ends with a backtick

```
$resultDiv.text(`${degCInt} C / ${degFInt} F`);
```

TEMPLATE LITERALS

```
$resultDiv.text(`${degCInt} C / ${degFInt} F`);
```

variable reference starts with a dollar sign

TEMPLATE LITERALS

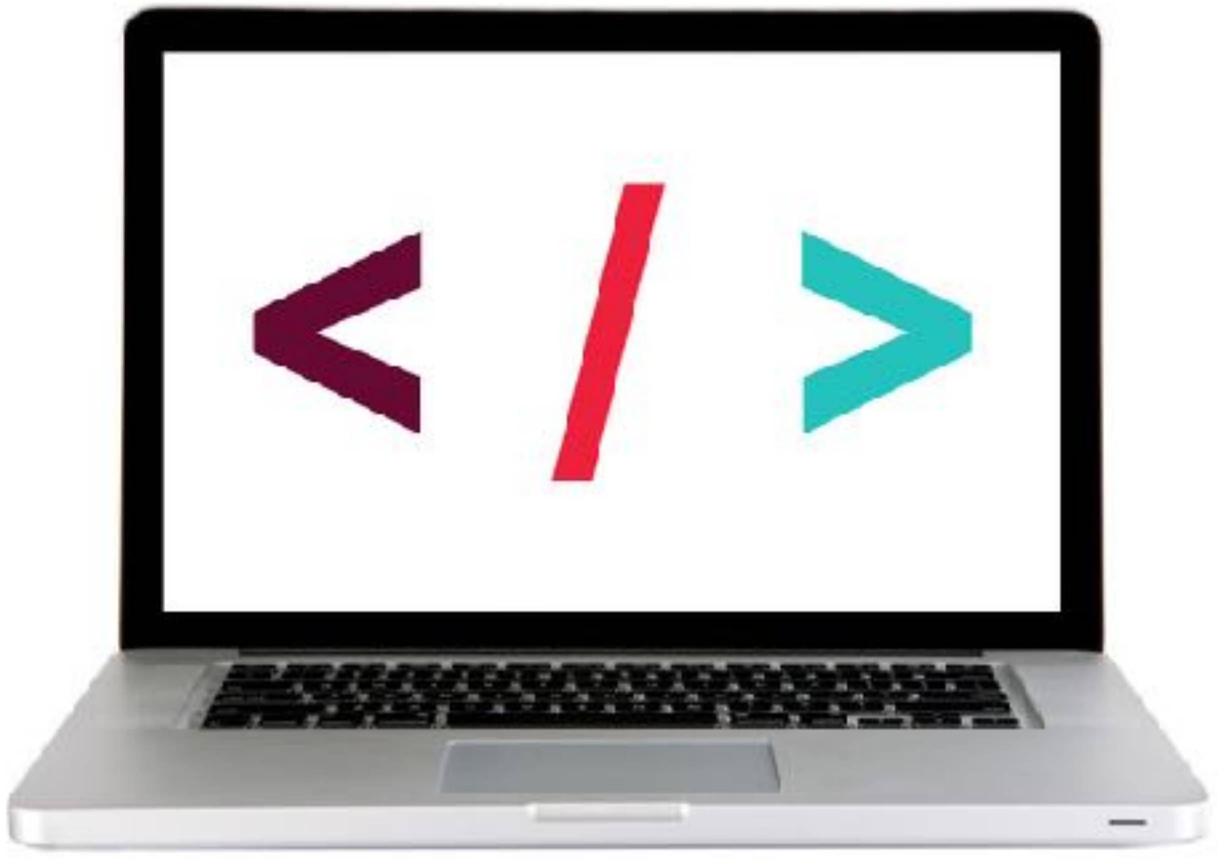
```
$resultDiv.text(`${degCInt} C / ${degFInt} F`);
```

variable reference enclosed in curly braces

BUILDING A TEMPLATE LITERAL

1. IDENTIFY THE DATA YOU WANT TO INTEGRATE
2. BUILD THE HTML STRUCTURE FOR A SAMPLE ELEMENT (INCLUDING CLASSES!)
3. CONVERT THE HTML CODE TO A TEMPLATE LITERAL

IN-CLASS LAB: FEEDR



LET'S TAKE A CLOSER LOOK

BUILDING A TEMPLATE FUNCTION

1. IDENTIFY THE DATA YOU WANT TO INTEGRATE
2. BUILD THE HTML STRUCTURE FOR A SAMPLE ELEMENT (INCLUDING CLASSES!)
3. CHOOSE A LOOP AND CREATE IT (FOREACH, FOR/IN, FOR)
4. CONVERT THE HTML CODE TO A TEMPLATE LITERAL, AND RETURN IT FROM THE LOOP

EXERCISE - TEMPLATE LITERALS

EXERCISE

LOCATION

- `starter-code > 2-templating-lab`

TIMING

10 min

1. Create a template literal and use it to display the data in the `favorite` object. Use the HTML structure shown in the comment in index.html.
2. Create a template literal that displays the contents of the `favorites` object at the bottom of main.js.

Exit Tickets!

(Class #12)

LEARNING OBJECTIVES - REVIEW

- Familiarize yourself with the API documentation for news sources.
- Fork and clone your starter code.
- Strategize ways to hide the loader and replace the content of the `#main` container with that of the API.
- Integrate string and variable values using template literals

NEXT CLASS PREVIEW

Prototypal inheritance

- Explain the difference between literal and constructed objects.
- Write a constructor for a JavaScript object.
- Explain prototypal inheritance and its purpose.
- Recognize the difference between prototypal and classical inheritance.
- Create and extend prototypes.

Q&A