

JAVASCRIPT DEVELOPMENT

Sasha Vodnik, Instructor

HELLO!

- 1. Pull changes from the JS-SF-15-resources repo to your computer
- 2. Open the 13-feedr-lab folder in your code editor

JAVASCRIPT DEVELOPMENT

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

Project 2 lab / Prototypal inheritance

WEEK 8

Closures & this / CRUD & Firebase

WEEK 9

Deploying your app / Final project lab

HOMEWORK REVIEW

HOMEWORK — GROUP DISCUSSION



TYPE OF EXERCISE

• Groups of 2-3

TIMING

4 min

- 1. Share your solutions for the Flickr project.
- 2. Share 1 thing you found challenging. If you worked it out, share how; if not, brainstorm with your group how you might approach it.
- 3. If you completed the bonus, demonstrate it and show how you coded it.
- 4. Share the APIs you plan to use for the Feedr project, and what you've learned about them from their documentation.

EXIT TICKET QUESTIONS

- 1. When working with different API's in the same app, will how we write the Fetch/Get requests verbage need to change in each part of the same code?
- 2. Couldn't really get the size suffixes to work. Any reason why only url_n works?

EXERCISE — PROJECT PLANNING



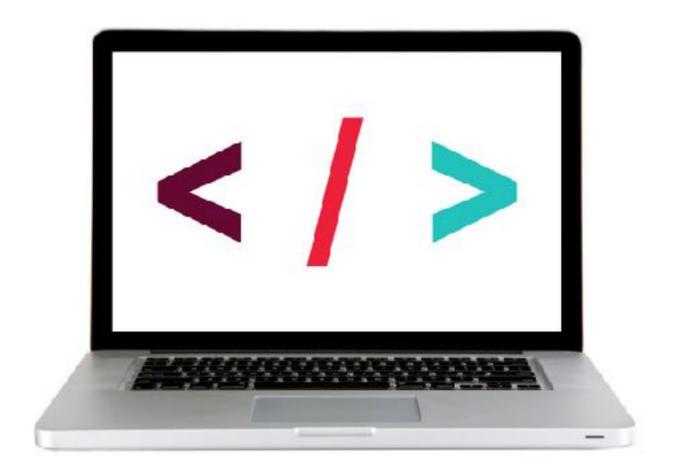
TYPE OF EXERCISE

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.



LET'S TAKE A CLOSER LOOK

EXERCISE — FEEDR PLANNING



TYPE OF EXERCISE

Individual, then pairs

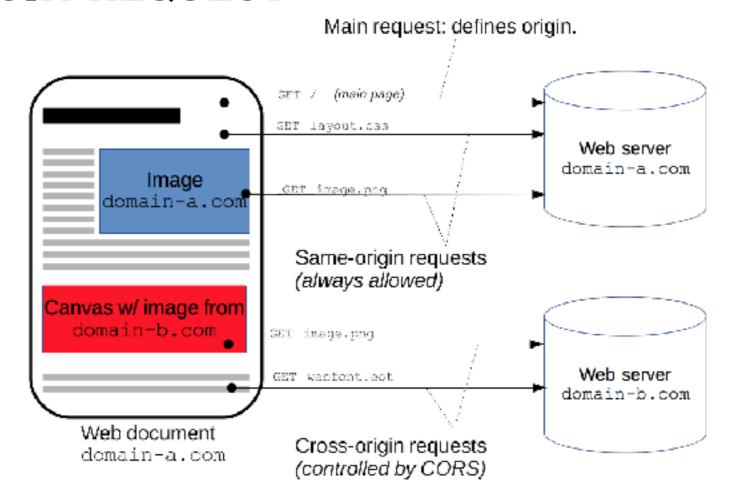
TIMING

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

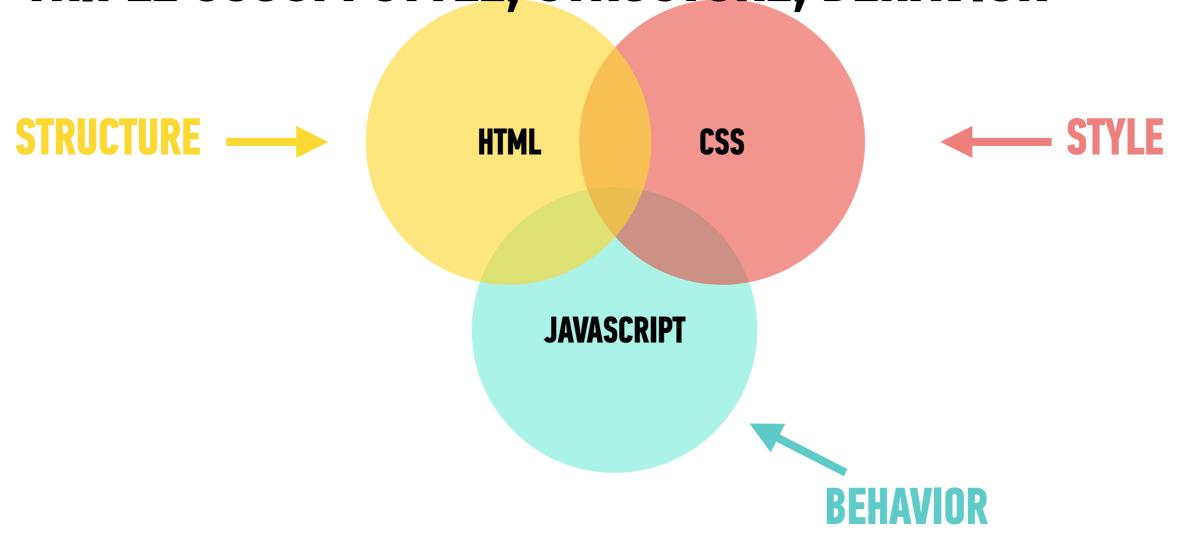
CROSS ORIGIN RESOURCE SHARING

Response header:

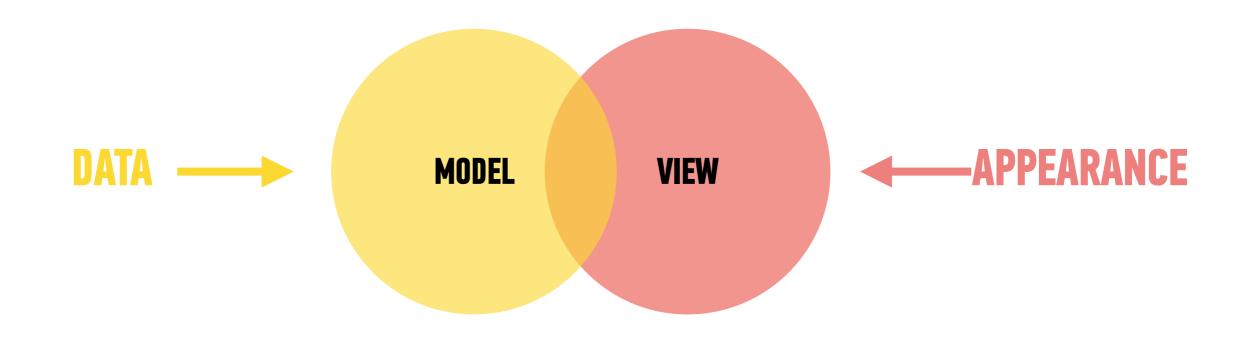
Access-Control-Allow-Origin: *

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
$\text(\${\degCInt} C / ${\degFInt} F');
```

TEMPLATE LITERALS

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

variable reference starts with a dollar sign

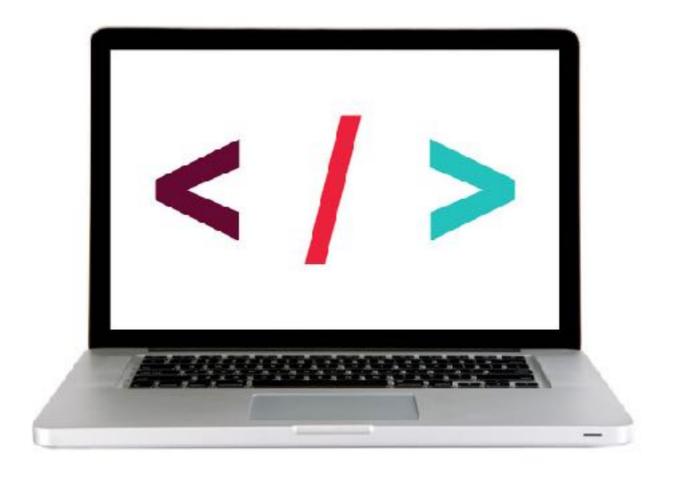
TEMPLATE LITERALS

```
$resultDiv.text(`${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



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



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.

QSA