GENERAL ASSEMBLY

JAVASCRIPT DEVELOPMENT

Sasha Vodnik, Instructor

HELLO!

1. Pull changes from the `JS-SF-16-resources` repo to your computer
2. Open the `12-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	Project 2 lab / Prototypal inheritance
WEEK 8	Closures & `this` / CRUD & Firebase
WEEK 9	Deploying your app / React

HOMEWORK REVIEW

HOMEWORK — GROUP DISCUSSION

TYPE OF EXERCISE

- Groups of 2-3

TIMING

5 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. What are some common errors with accessing APIs that aren't always apparent?
2. When should we use fetch versus get for APIs?
3. What's an SDK? What's the difference between an SDK and an API?

EXERCISE — PROJECT PLANNING

TYPE OF EXERCISE

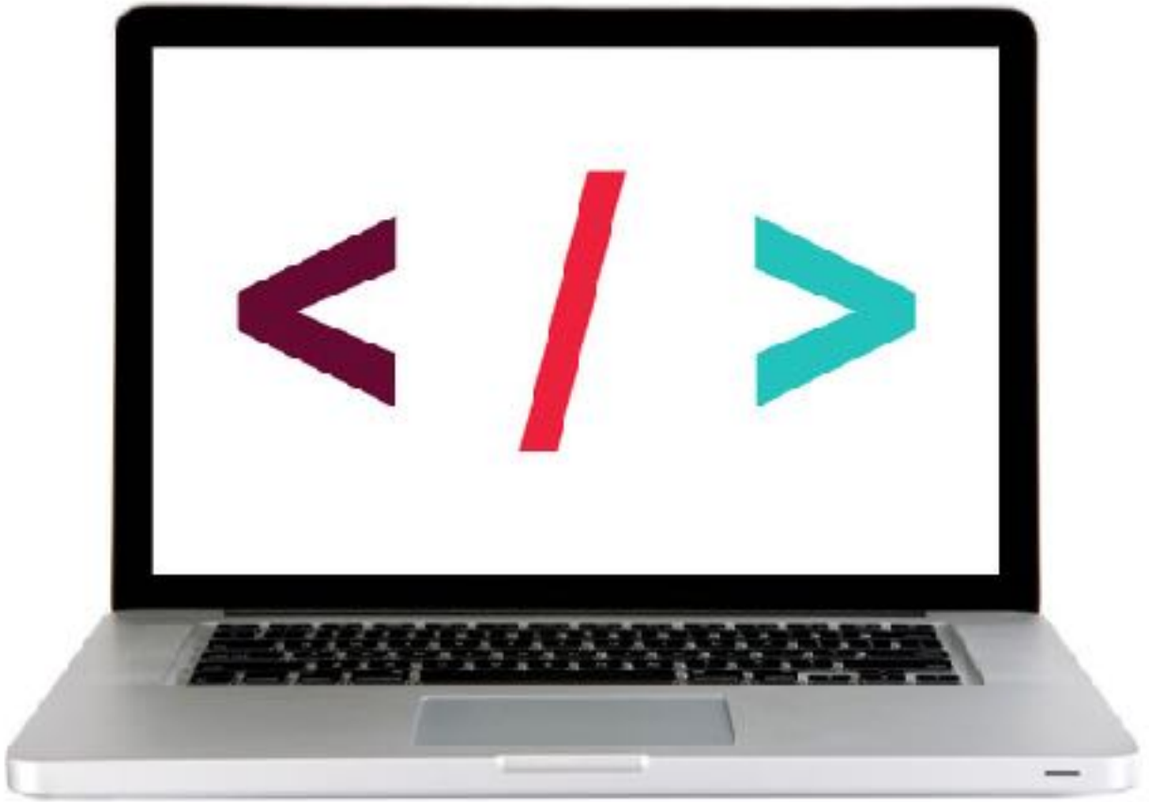
- Group

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.

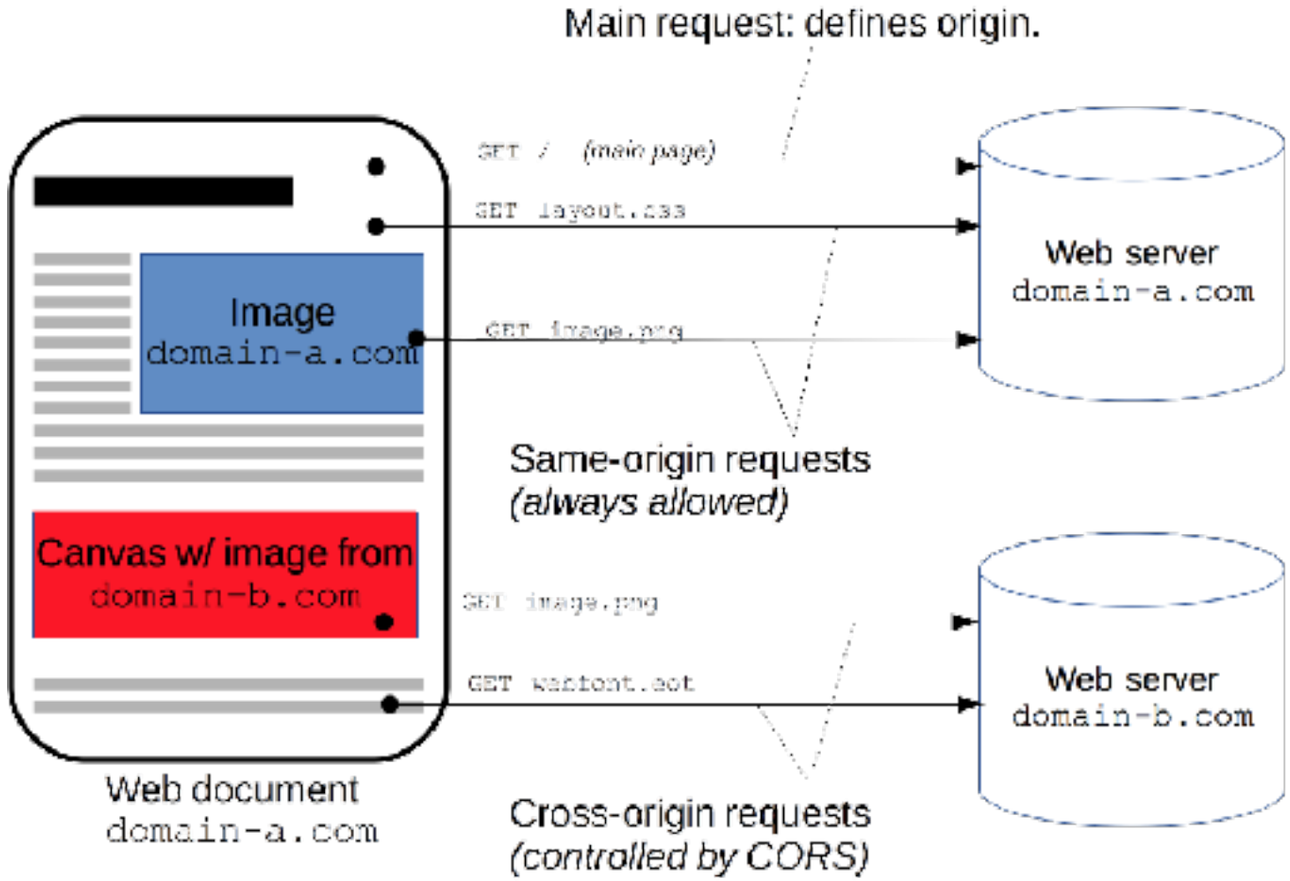
IN-CLASS LAB: FEEDR



LET'S TAKE A CLOSER LOOK

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: *
```

EXERCISE — FEEDR PLANNING

TYPE OF EXERCISE

- Individual, then pairs

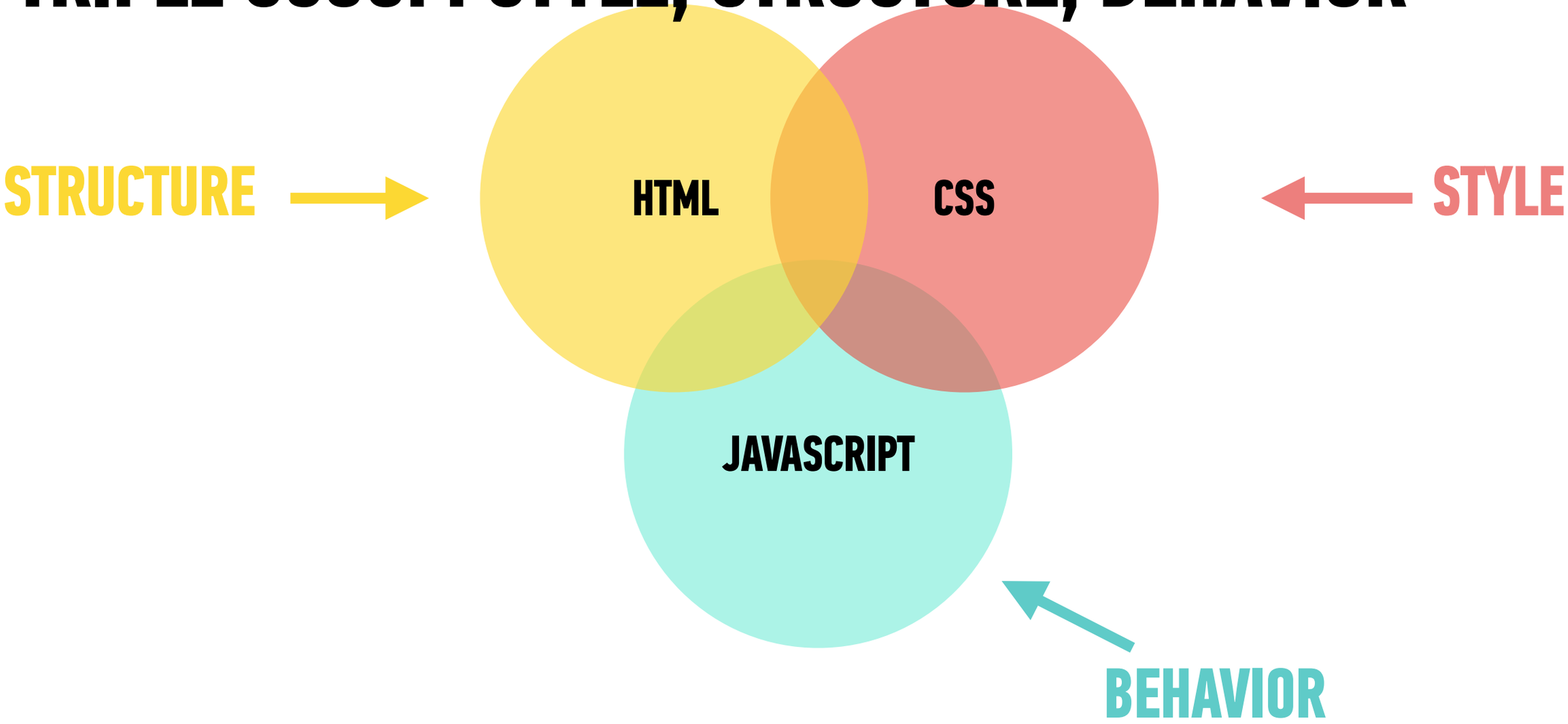
TIMING

6 min

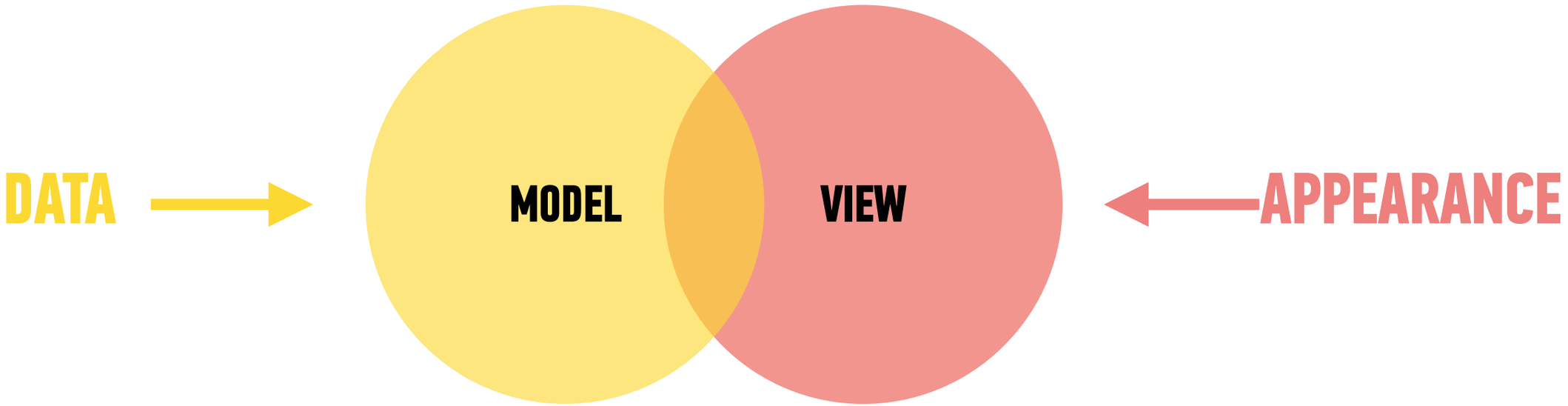
1. Take a minute or two to decide on your next step for your Feedr project. (It's okay to have a few possible next steps at this point.)
2. Share your next step(s) with one or two classmates. If you have different approaches, talk about how you decided on your approach.
3. Share the list of news sources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

TEMPLATE LITERALS

TRIPLE SCOOP: STYLE, STRUCTURE, BEHAVIOR



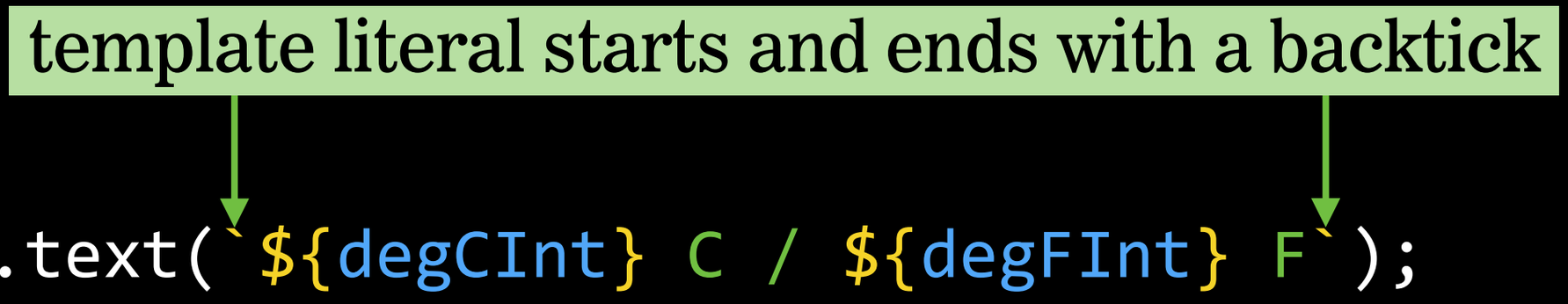
MODEL VS VIEW



DOM MANIPULATION

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

TEMPLATE LITERALS



TEMPLATE LITERALS

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

variable reference
starts with a dollar sign

TEMPLATE LITERALS

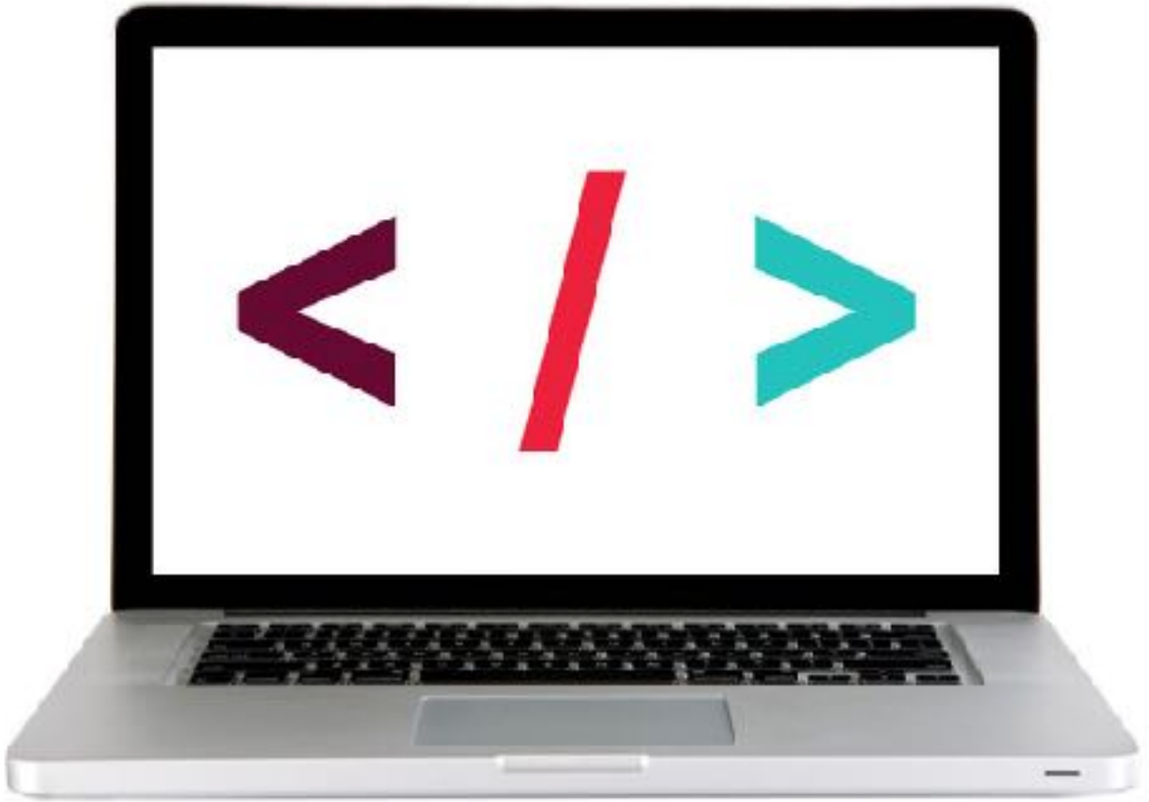
```
$resultDiv.text(`${degCInt} C / ${degFInt} F`);
```

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

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