GENERAL ASSEMBLY

JAVASCRIPT DEVELOPMENT

Larissa Muramoto, Instructor

HELLO!

1. Pull changes from the `svodnik/JS-SF-15-resources` repo to your computer
2. Open the `16-deploying` folder in your code editor

JAVASCRIPT DEVELOPMENT

DEPLOYING YOUR APP

LEARNING OBJECTIVES

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

- Understand what hosting is.
- Identify a program’s needs in terms of host providers.
- Ensure backward compatibility by using Babel to transpile code.
- Optimize code before deployment
- Deploy to a web host.

AGENDA

- Transpile with Babel
- Lint with ESLint
- Minify with Uglify-JS
- Add a polyfill
- Deploy with Firebase

WEEKLY OVERVIEW

WEEK 9	Deploying your app / React
WEEK 10	Final project lab / Graduation!

FINALIZING YOUR CODE

FINALIZING YOUR CODE

Process	WHY
Transpiling	So you can use the most recent JS features
Linting	So your code is clean (check for syntax errors & formatting)
Minifying	So your code is more efficient
Polyfills	So you can use the most recent browser features

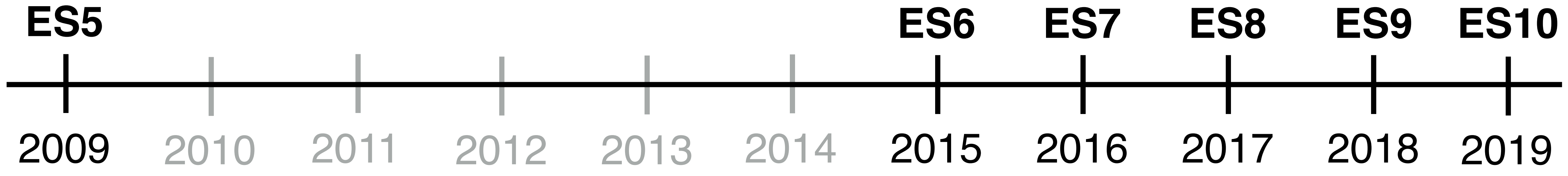
DEPLOYING YOUR APP

TRANSPILING

DEPLOYING YOUR APP

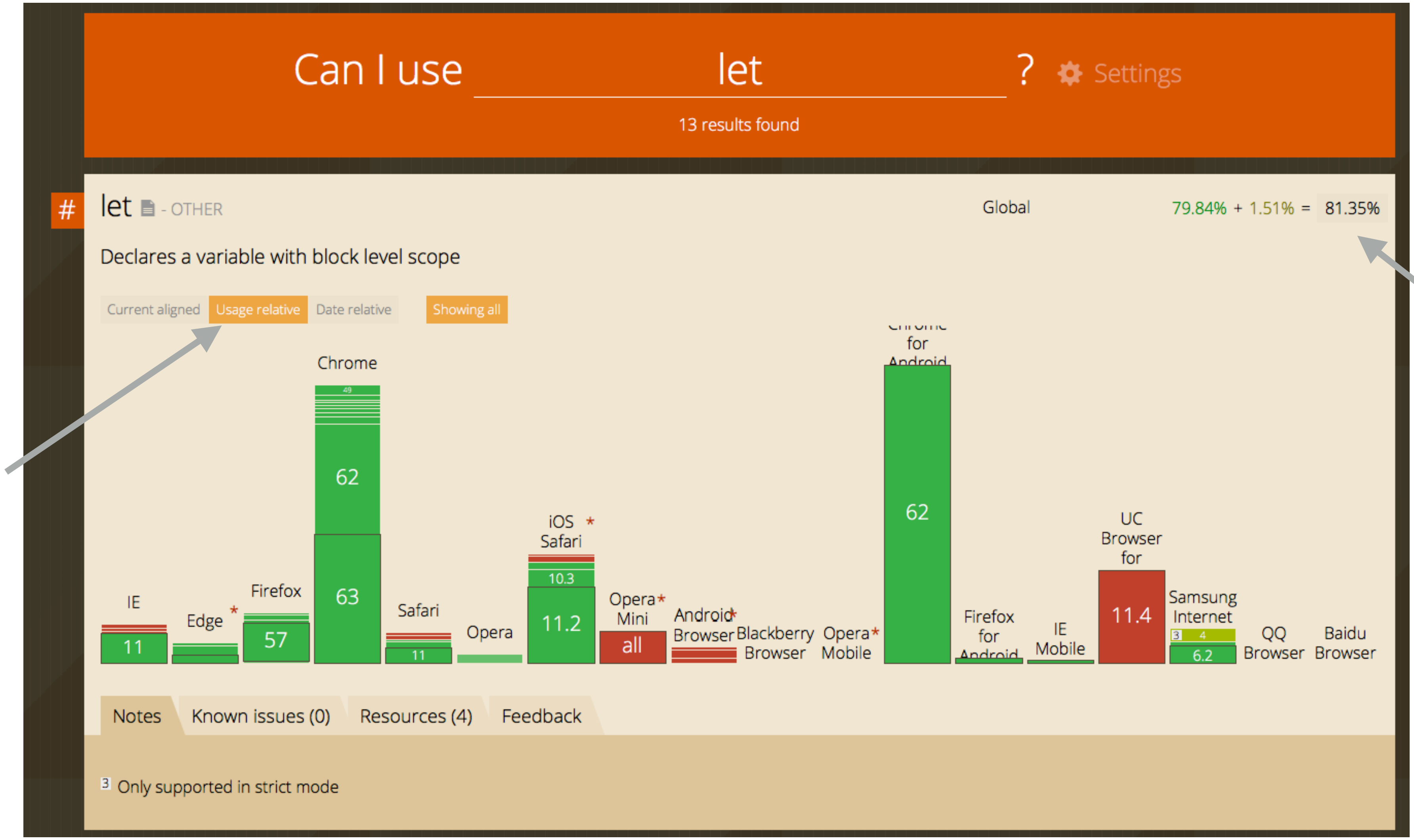
virtually all browsers
in use support ES5

only modern browsers
support ES6+



DEPLOYING YOUR APP

caniuse.com



Can I use let ? Settings

13 results found

let - OTHER

Global 79.84% + 1.51% = 81.35%

Declares a variable with block level scope

[3] Only supported in strict mode

"Usage relative" option shows proportional graph

Estimated percent of global browser traffic that can parse this feature

Transpiling involves rewriting code that uses ES6+ features to produce the same result using ES5 code

ES6

```
const taxRate = 0.0875;
let items = [];

let addToCart = () => {
  // do something
}
```

transpiling $\longrightarrow$

ES5

```
var taxRate = 0.0875;
var items = [];

function addToCart() {
  // do something
}
```

BABEL

https://babeljs.io/setup#installation

SETTING UP YOUR TRANSPILER

(Modified from: https://babeljs.io/setup#installation)

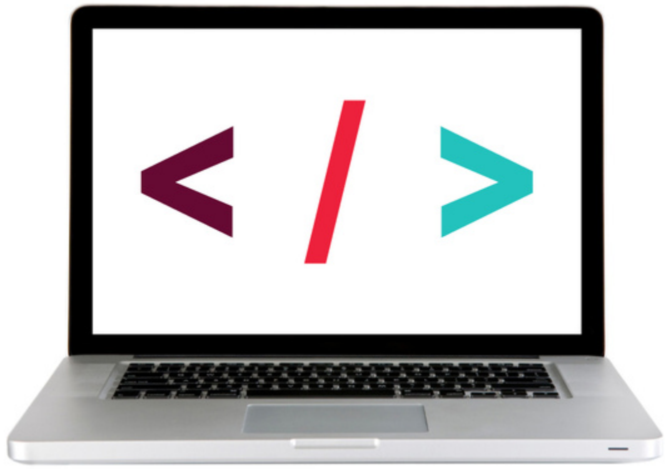
1. npm init
2. npm install --save-dev @babel/core @babel/cli
3. Add to package.json under “scripts”:

```
“transpile": "babel js -d lib"
```

4. npm install @babel/preset-env --save-dev
5. Create .babelrc file with the code inside it:

```
{
  "presets": ["@babel/preset-env"]
}
```

6. npm run transpile



0-transpiling-codealong

EXERCISE — TRANSPILE CODE USING BABEL

EXERCISE

KEY OBJECTIVE

- Ensure backward compatibility by using Babel to transpile code.

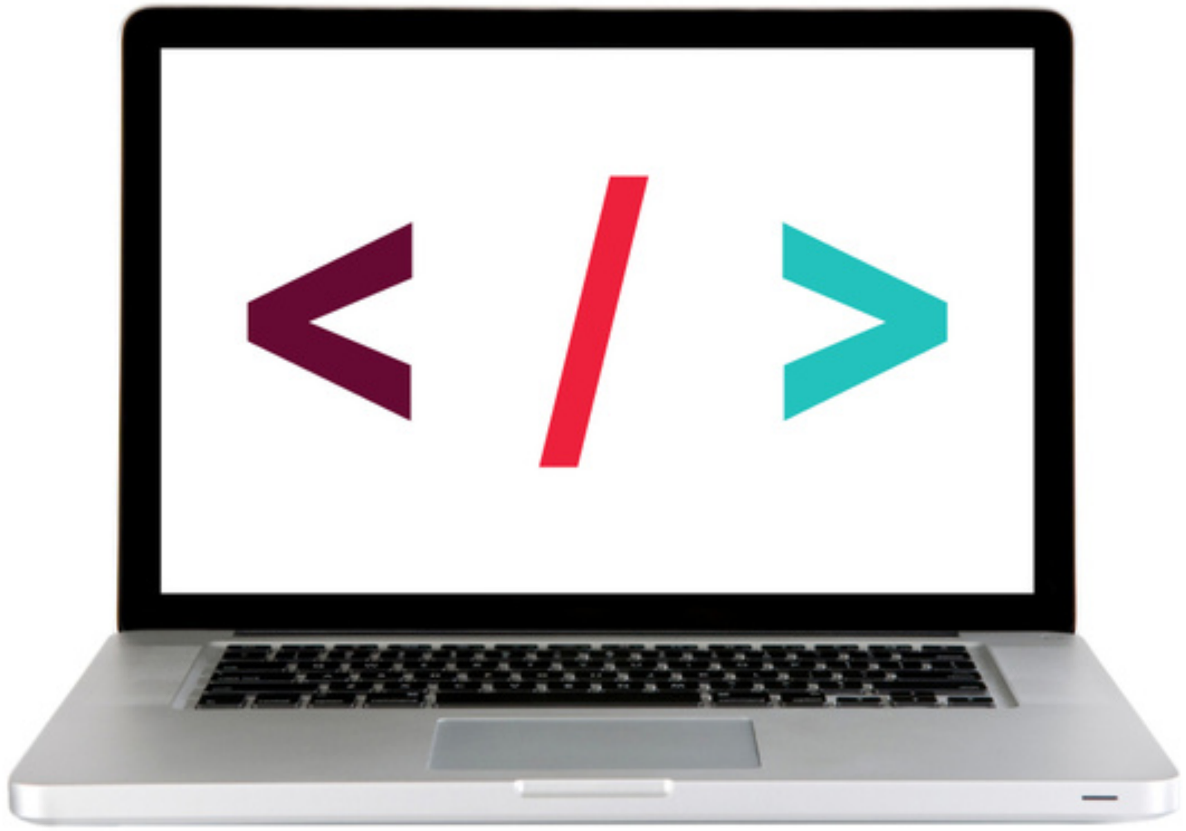
TIMING

5 min

1. Configure Babel for the Firebase app you created in the previous class.
   (If your code isn't quite working, use the code in the starter-code > 1-transpiling-exercise folder as a starting point.)
2. Run Babel to create an ES5-compatible version of your code.
3. Open the converted file in your editor and verify the code was transpiled.
4. Open index.html and change the source for the script element to the JavaScript file created by Babel.
5. Test your app in the browser and make sure it still works as it did previously.

LINTING

LET'S TAKE A CLOSER LOOK



https://eslint.org/demo

EXERCISE — LINT CODE USING ESLINT

EXERCISE

KEY OBJECTIVE

- Optimize code for deployment.

TIMING

3 min

1. In your browser, open https://eslint.org/demo.
2. Copy the contents of app.js from your Firebase project, paste in the left pane of the ESLint interface, and verify that no errors are shown.
3. If errors are flagged, fix them in the web interface, then when the code is error-free, copy the code from the web interface (click in the code and press command+A), then replace the code in app.js with the copied code. Save your changes.
4. Test your app in the browser and make sure it still works as it did previously.

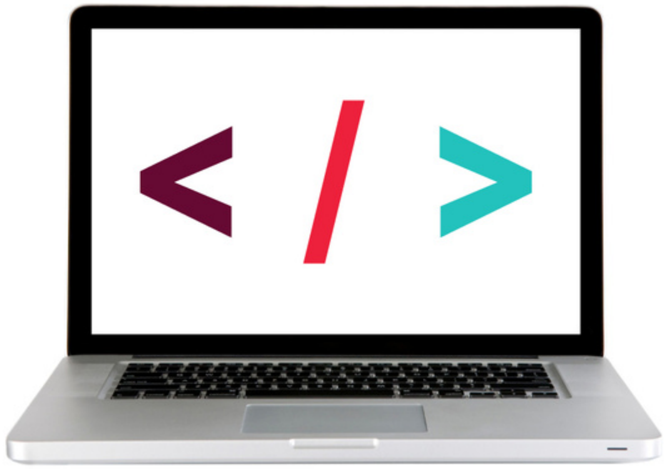
MINIFYING

DEPLOYING YOUR APP

https://www.npmjs.com/package/uglify-js

SETTING UP YOUR MINIFIER

(Modified from: https://www.npmjs.com/package/uglify-js)



1. npm install --save-dev uglify-js
2. Add to package.json under “scripts”:

   ```
   “minify": “uglifyjs lib/app.js —o lib/app.min.js“
   ```
3. npm run minify
4. Update your index.html to point to the minified version!

EXERCISE — MINIFY CODE

EXERCISE

KEY OBJECTIVE

- Optimize code for deployment.

TIMING

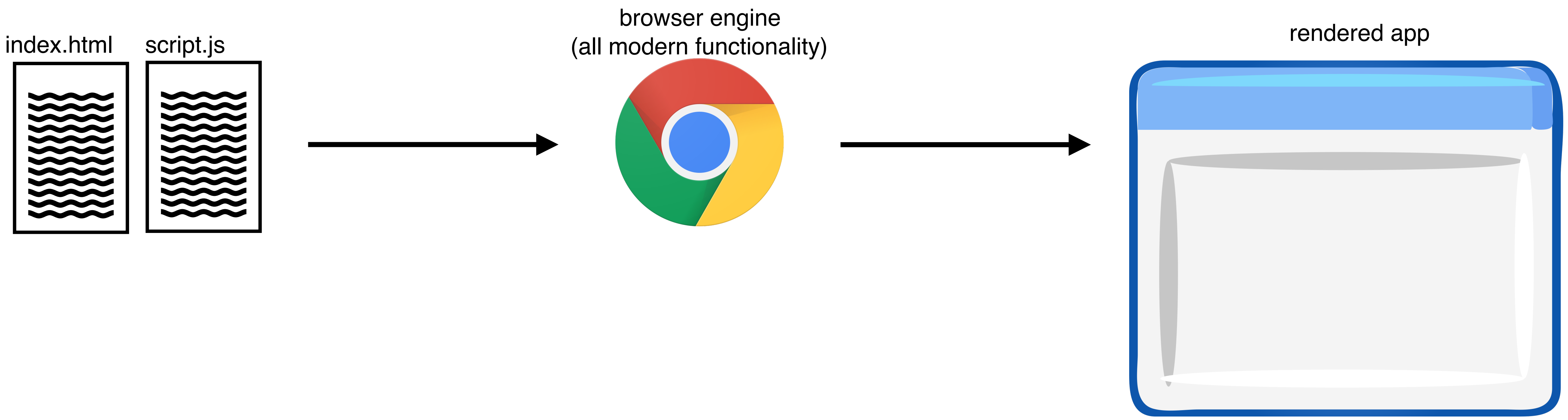
3 min

1. At the command line, navigate to the folder containing your Firebase project.
2. Use uglify to create a minified version of app.js, outputting to app.min.js.
3. Open index.html and change the source for the script element to app.min.js.
4. Test your app in the browser and make sure it still works as it did previously.

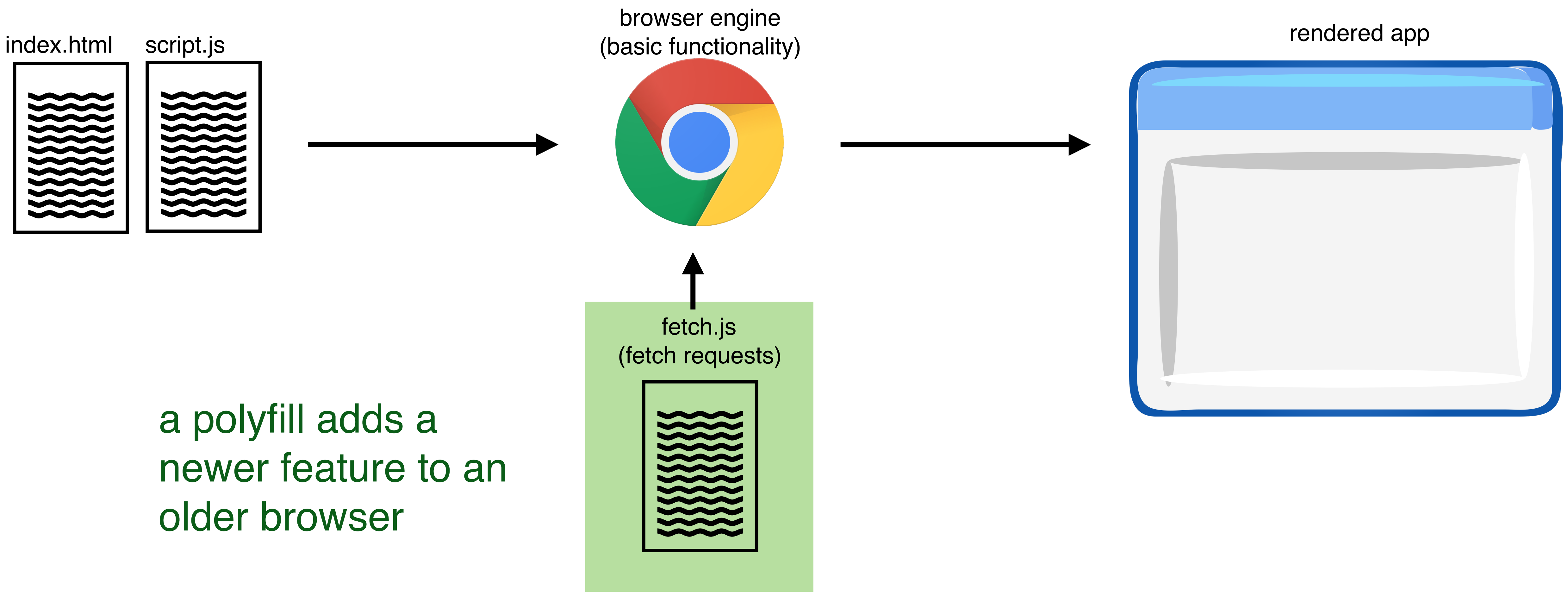
DEPLOYING YOUR APP

POLYFILLS

APP FUNCTIONALITY IN A MODERN



APP FUNCTIONALITY IN AN OLDER

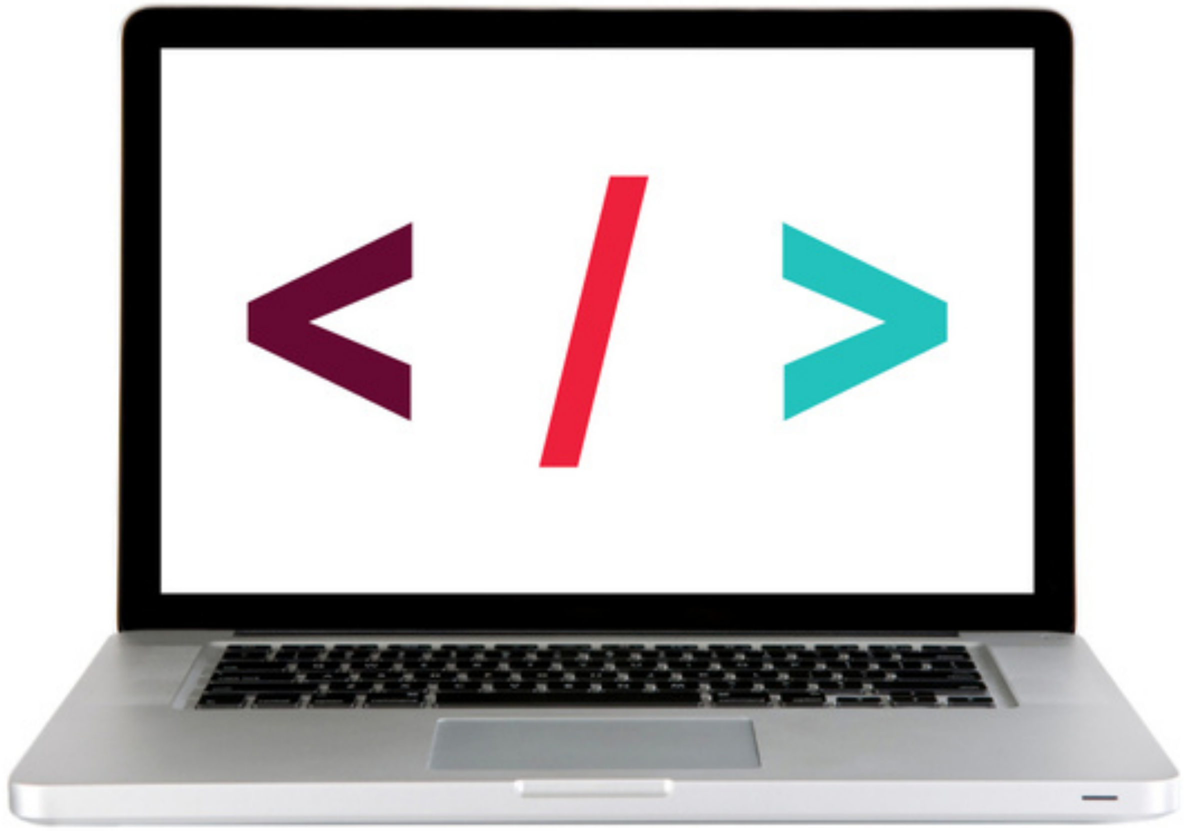


a polyfill adds a newer feature to an older browser

DEPLOYING YOUR APP

https://github.com/github/fetch

LET'S TAKE A CLOSER LOOK



2-polyfill-codealong

EXERCISE — ADD POLYFILLS

EXERCISE

KEY OBJECTIVE

- Optimize code for deployment.

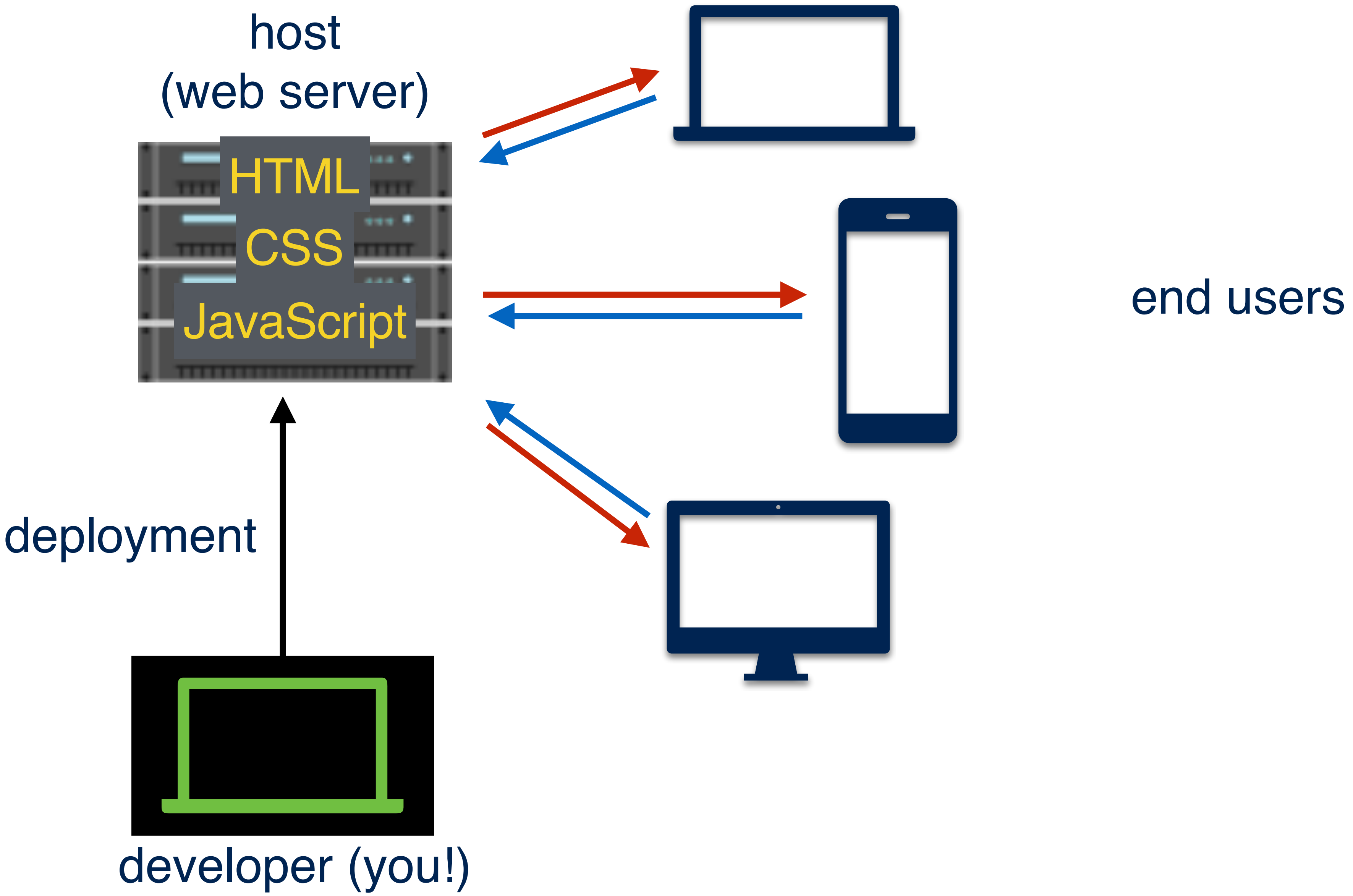
TIMING

5 min

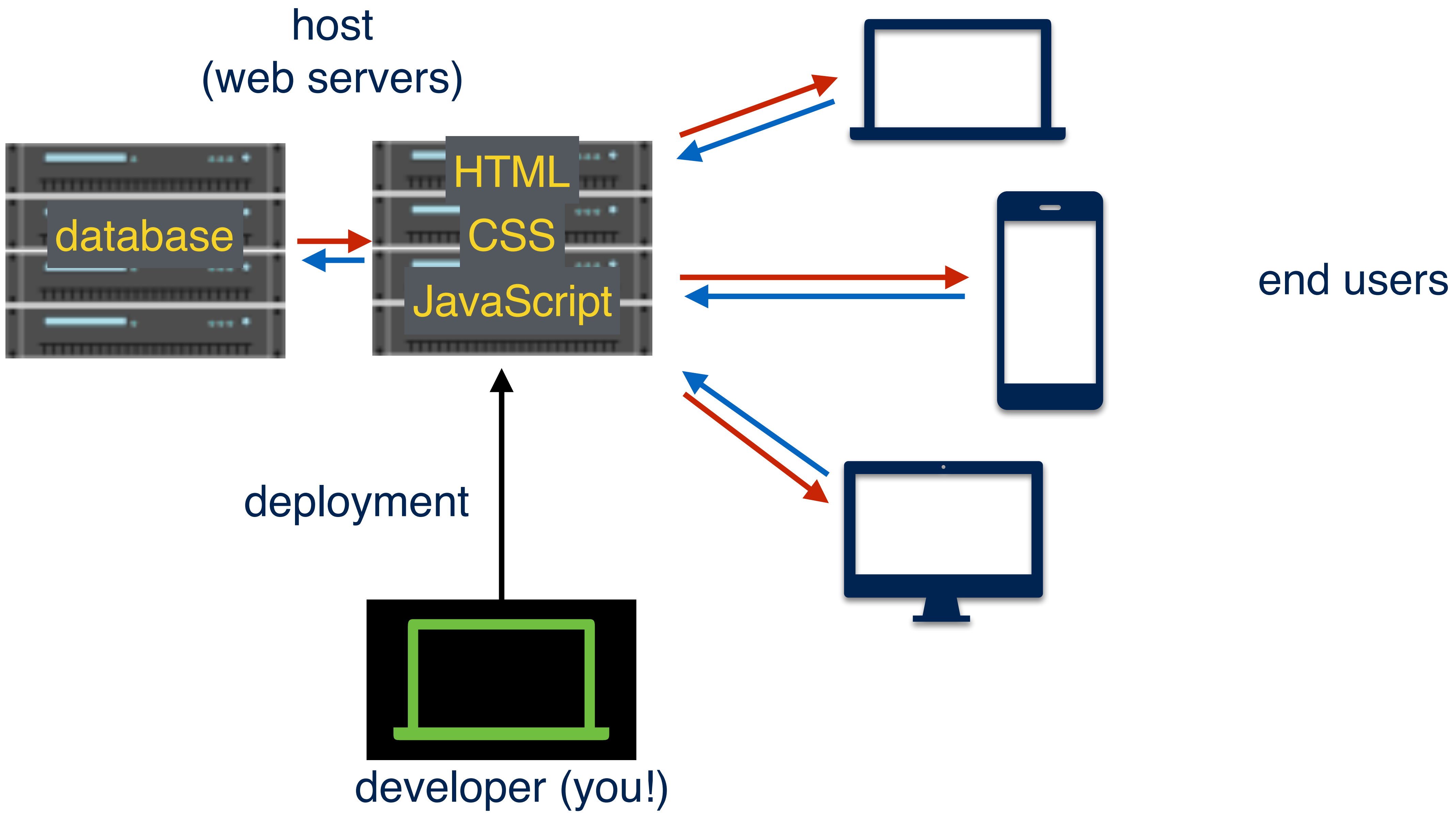
1. At the command line, navigate to the folder containing your Firebase project.
2. Add polyfills to enable Fetch in older browsers.
3. If you have access to a browser that does not support Fetch, test your app in that browser and make sure it works
4. Also test your app in a modern browser and ensure it still works as it did previously.

DEPLOYMENT

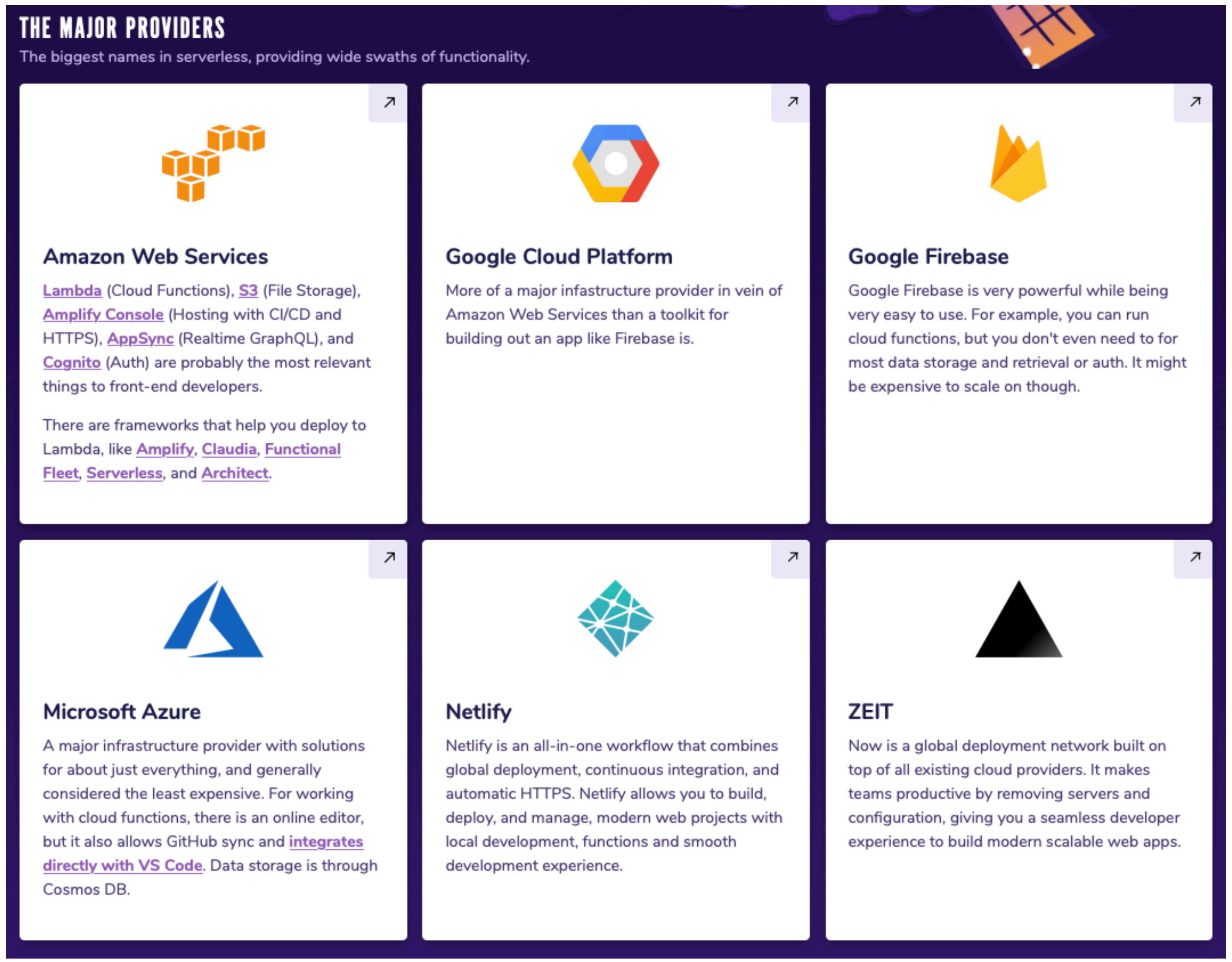
DEPLOYING YOUR APP



DEPLOYING YOUR APP



ALTERNATIVE “SERVERLESS”



THE MAJOR PROVIDERS

The biggest names in serverless, providing wide swaths of functionality.

Amazon Web Services

Lambda (Cloud Functions), S3 (File Storage), Amplify Console (Hosting with CI/CD and HTTPS), AppSync (Realtime GraphQL), and Cognito (Auth) are probably the most relevant things to front-end developers.

There are frameworks that help you deploy to Lambda, like Amplify, Claudia, Functional Fleet, Serverless, and Architect.

Google Cloud Platform

More of a major infrastructure provider in vein of Amazon Web Services than a toolkit for building out an app like Firebase is.

Google Firebase

Google Firebase is very powerful while being very easy to use. For example, you can run cloud functions, but you don't even need to for most data storage and retrieval or auth. It might be expensive to scale on though.

Microsoft Azure

A major infrastructure provider with solutions for about just everything, and generally considered the least expensive. For working with cloud functions, there is an online editor, but it also allows GitHub sync and integrates directly with VS Code. Data storage is through Cosmos DB.

Netlify

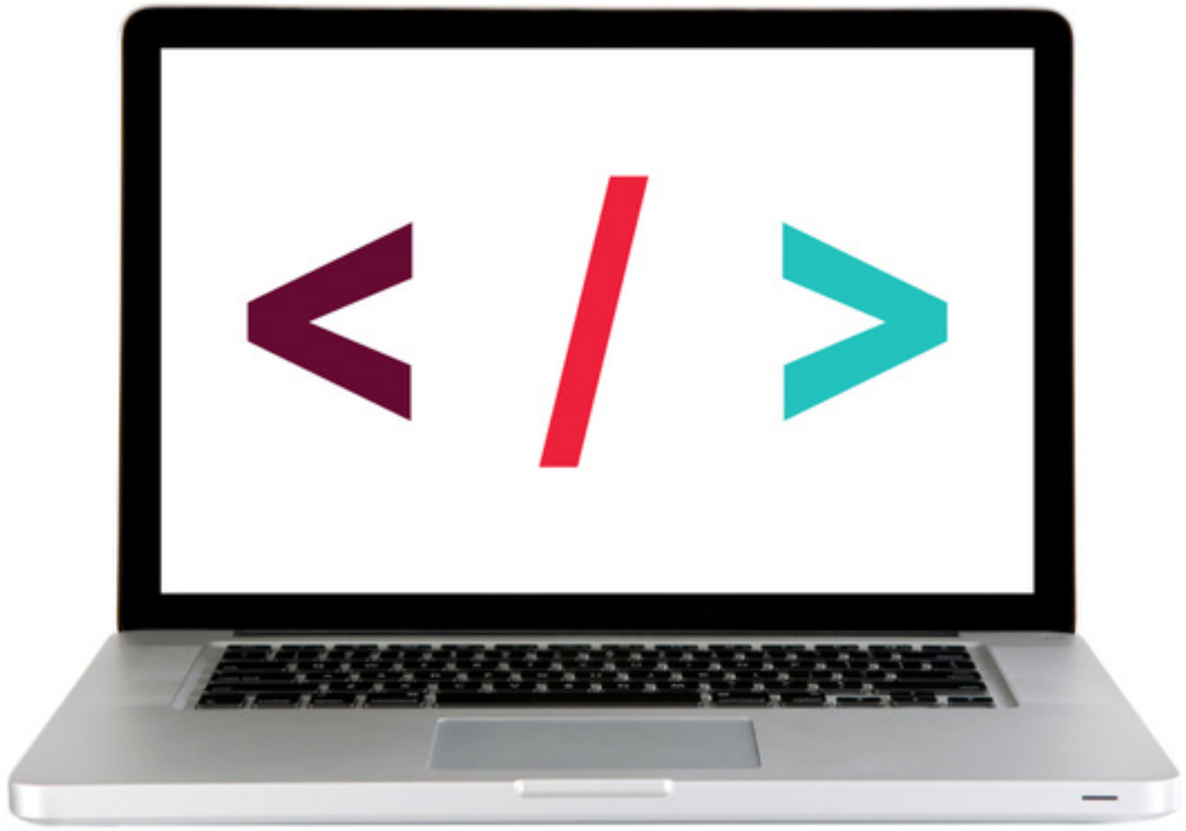
Netlify is an all-in-one workflow that combines global deployment, continuous integration, and automatic HTTPS. Netlify allows you to build, deploy, and manage, modern web projects with local development, functions and smooth development experience.

ZEIT

Now is a global deployment network built on top of all existing cloud providers. It makes teams productive by removing servers and configuration, giving you a seamless developer experience to build modern scalable web apps.

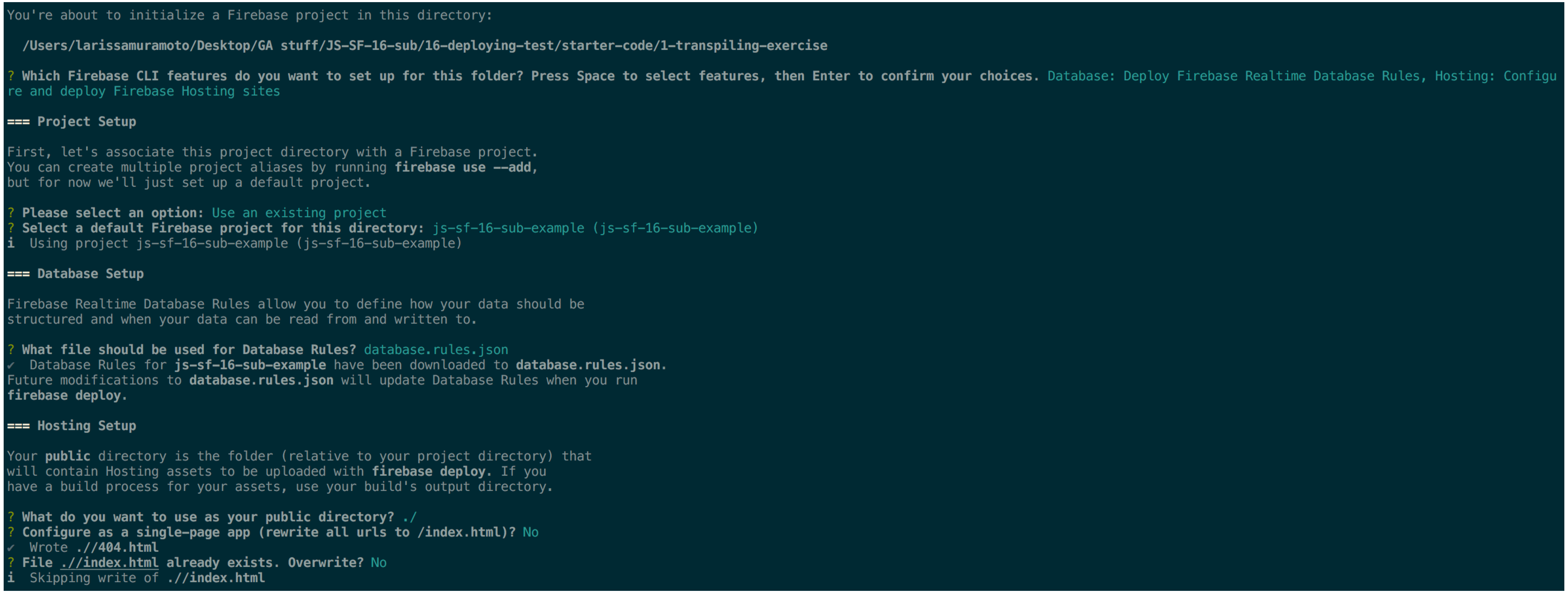
https://thepowerofserverless.info/services.html#major-providers

LET'S TAKE A CLOSER LOOK



firebase.google.com

LET'S TAKE A CLOSER LOOK



```
You're about to initialize a Firebase project in this directory:

  /Users/larissamuramoto/Desktop/GA stuff/JS-SF-16-sub/16-deploying-test/starter-code/1-transpiling-exercise

? Which Firebase CLI features do you want to set up for this folder? Press Space to select features, then Enter to confirm your choices. Database: Deploy Firebase Realtime Database Rules, Hosting: Configu
re and deploy Firebase Hosting sites

=== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

? Please select an option: Use an existing project
? Select a default Firebase project for this directory: js-sf-16-sub-example (js-sf-16-sub-example)
i  Using project js-sf-16-sub-example (js-sf-16-sub-example)

=== Database Setup

Firebase Realtime Database Rules allow you to define how your data should be
structured and when your data can be read from and written to.

? What file should be used for Database Rules? database.rules.json
✔  Database Rules for js-sf-16-sub-example have been downloaded to database.rules.json.
Future modifications to database.rules.json will update Database Rules when you run
firebase deploy.

=== Hosting Setup

Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? ./
? Configure as a single-page app (rewrite all urls to /index.html)? No
✔  Wrote .//404.html
? File .//index.html already exists. Overwrite? No
i  Skipping write of .//index.html
```

EXERCISE — PUSH CHANGES TO FIREBASE

EXERCISE

KEY OBJECTIVE

- Deploy to a web host.

TIMING

5 min

1. Make a change to the HTML, CSS, and/or JavaScript for the project you deployed to Firebase.
2. Push your changes to Firebase and verify that your updated code is what you see in your browser at appname.firebaseapp.com

Exit Tickets!

(Class #16)

LEARNING OBJECTIVES - REVIEW

- Understand what hosting is.
- Identify a program’s needs in terms of host providers.
- Ensure backward compatibility by using Babel to transpile code.
- Optimize code before deployment
- Deploy to a web host.

Q&A