

# JAVASCRIPT DEVELOPMENT

Sasha Vodnik, 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

---

## DEPLOYING YOUR APP

---

# WEEKLY OVERVIEW

### WEEK 9

Deploying your app / Final project lab

### WEEK 10

React / Graduation!

# ACTIVITY

---



## EXERCISE

### KEY OBJECTIVE

---

- Check in on final projects

### TYPE OF EXERCISE

---

- Groups of 2-4

### TIMING

---

*6 min*

1. Share what you have done so far on your final project (notes/outline, wireframe, pseudocode, basic functionality...)
2. Share your next step. If you're not sure, share where you are right now and brainstorm with your group what next steps might look like.

# **EXIT TICKET QUESTIONS**

1. When referencing the database, are there any methods or functions that will not be allowed by default?



---

**DEPLOYING YOUR APP**

---

# **FINALIZING YOUR CODE**

---

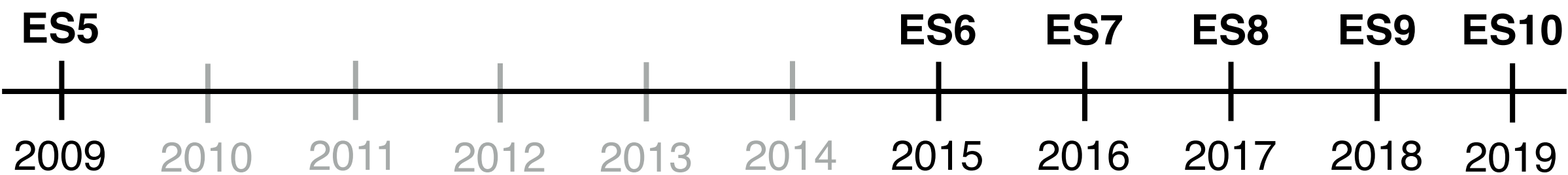
**DEPLOYING YOUR APP**

---

# **TRANSPILING**

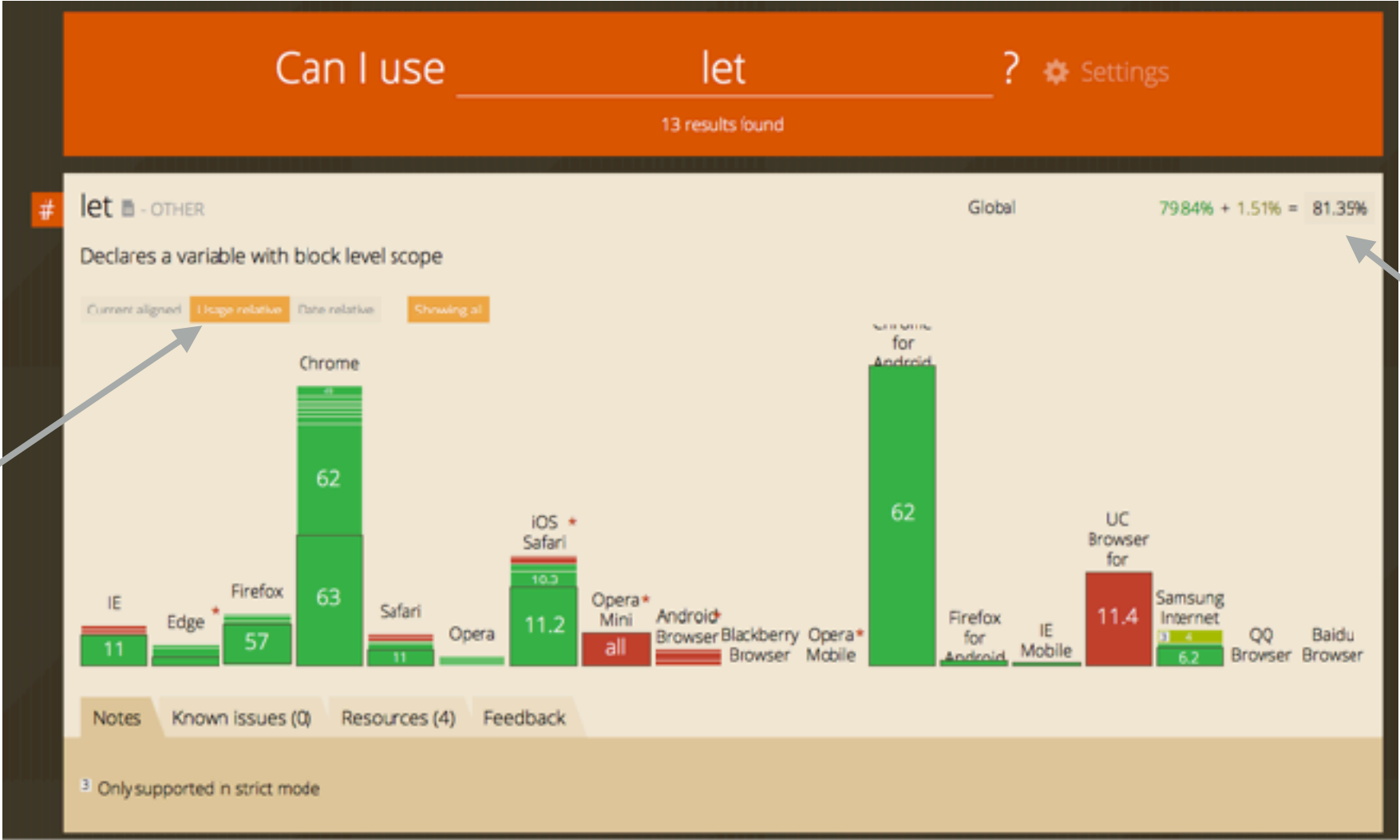
virtually all browsers  
in use support ES5

only modern browsers  
support ES6+



caniuse.com

“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



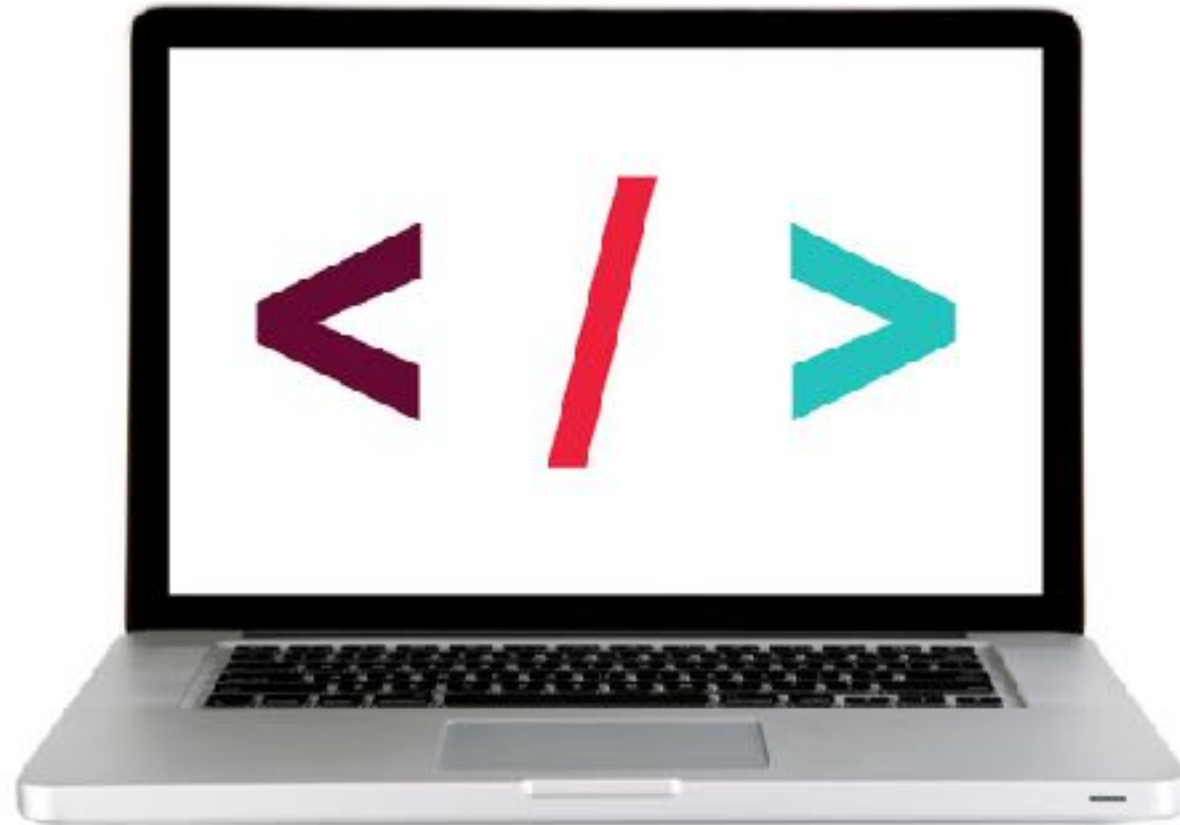
ES5

```
var taxRate = 0.0875;  
var items = [];  
  
function addToCart() {  
  // do something  
}
```

---

## LET'S TAKE A CLOSER LOOK

---



---

## 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.

---

**DEPLOYING YOUR APP**

---

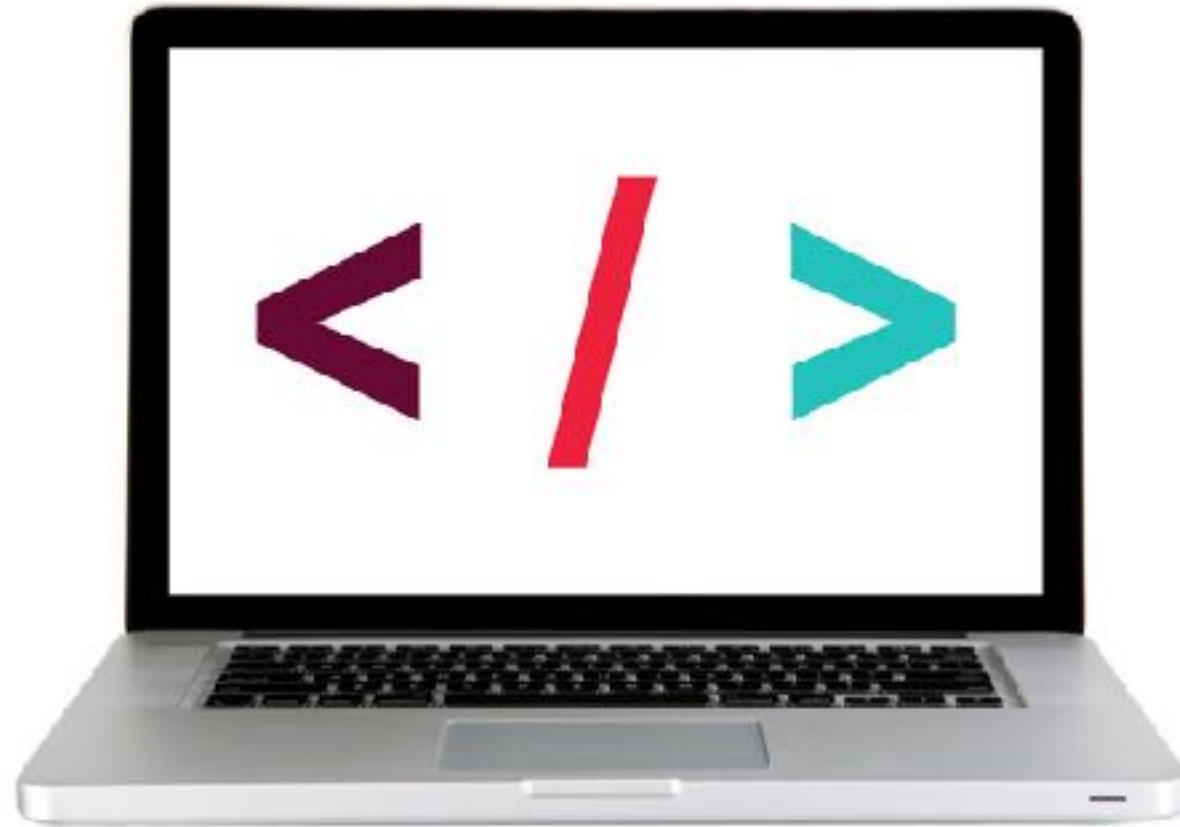
# **LINTING**



---

## LET'S TAKE A CLOSER LOOK

---



## 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.

---

**DEPLOYING YOUR APP**

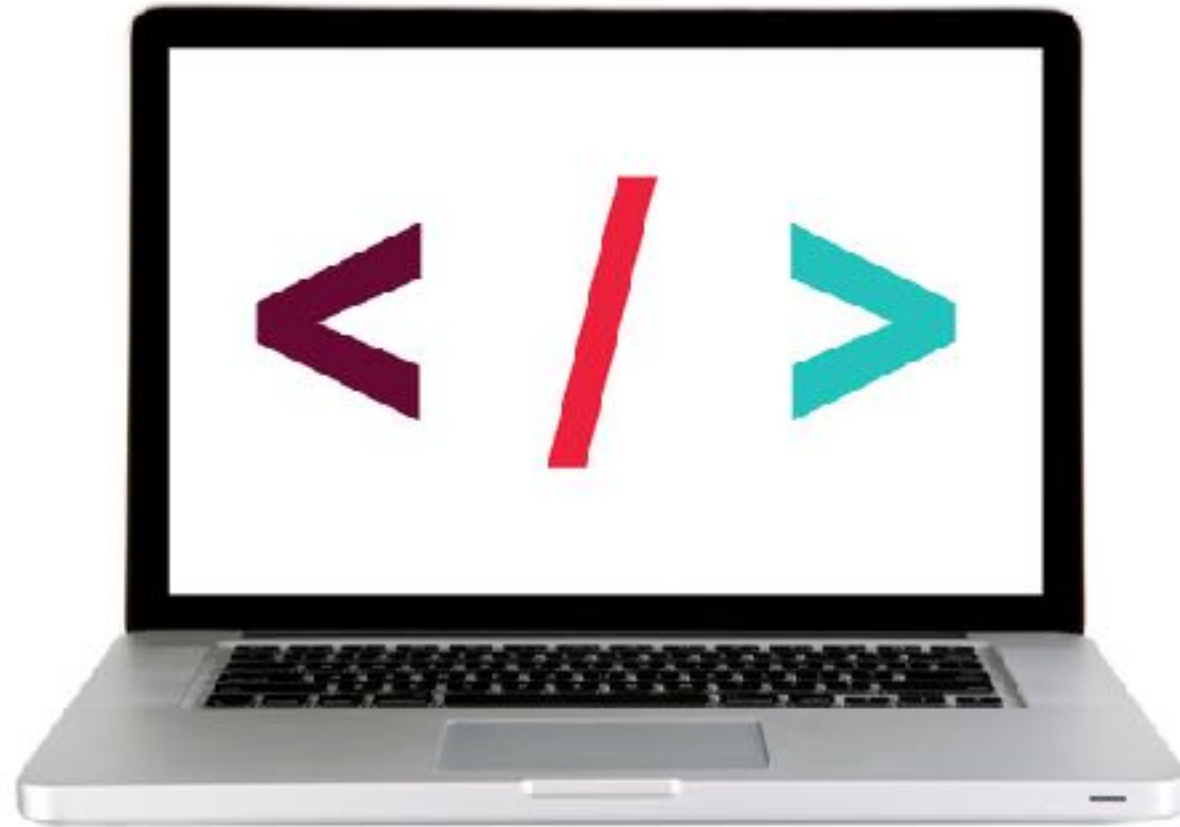
---

# **MINIFYING**

---

## LET'S TAKE A CLOSER LOOK

---



---

## 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.

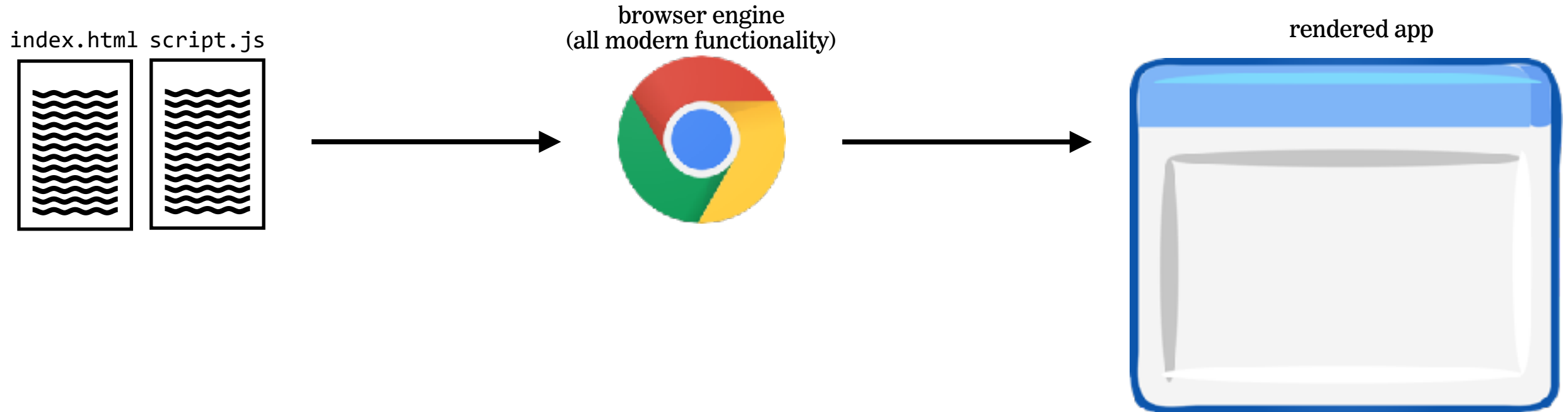
---

**DEPLOYING YOUR APP**

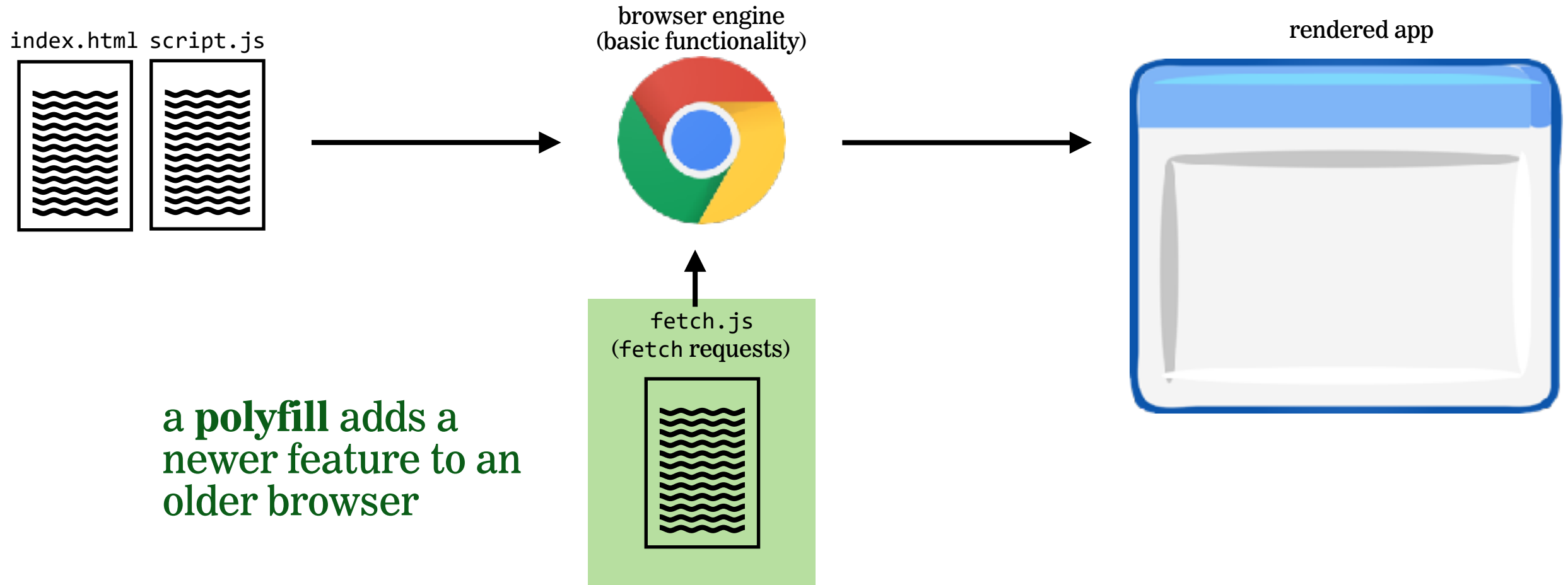
---

# **POLYFILLS**

# APP FUNCTIONALITY IN A MODERN BROWSER



# APP FUNCTIONALITY IN AN OLDER BROWSER

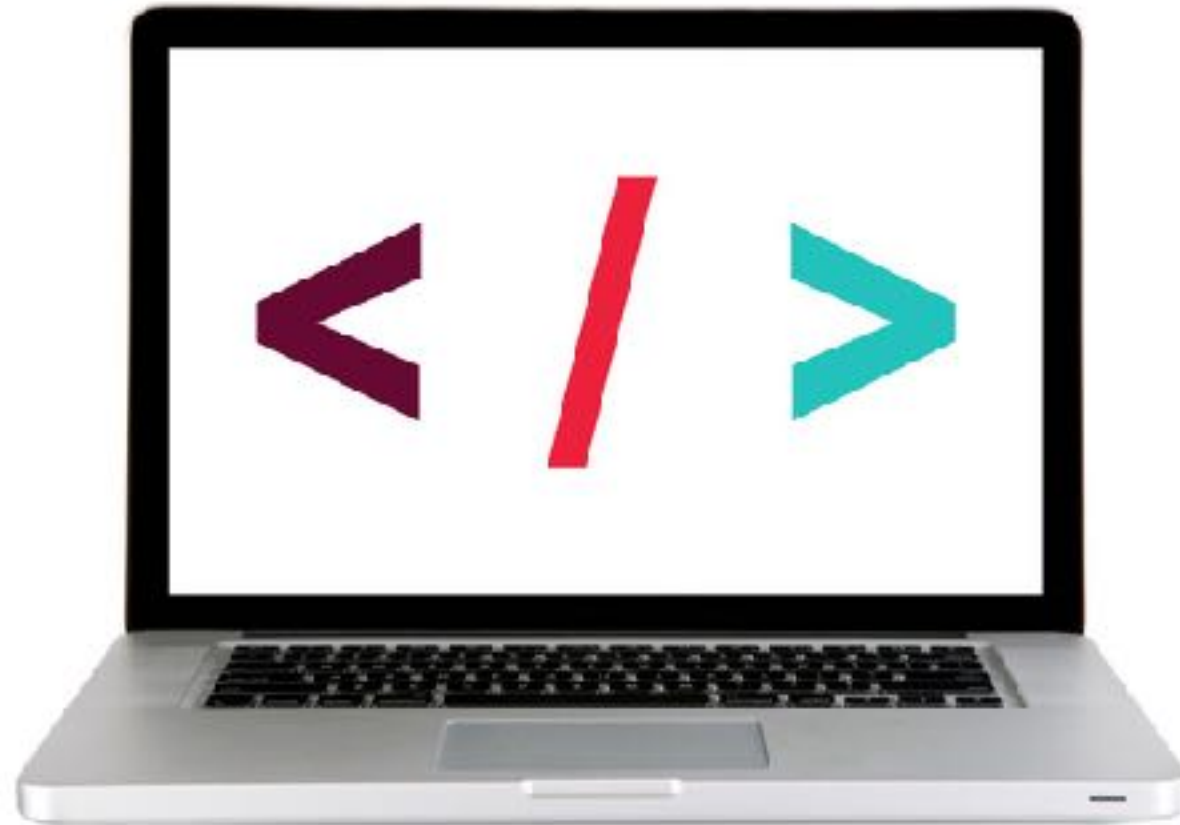




---

## LET'S TAKE A CLOSER LOOK

---



---

## EXERCISE — ADD POLYFILLS

---



### EXERCISE

#### KEY OBJECTIVE

---

- Optimize code for deployment.

#### TIMING

---

*5 min*

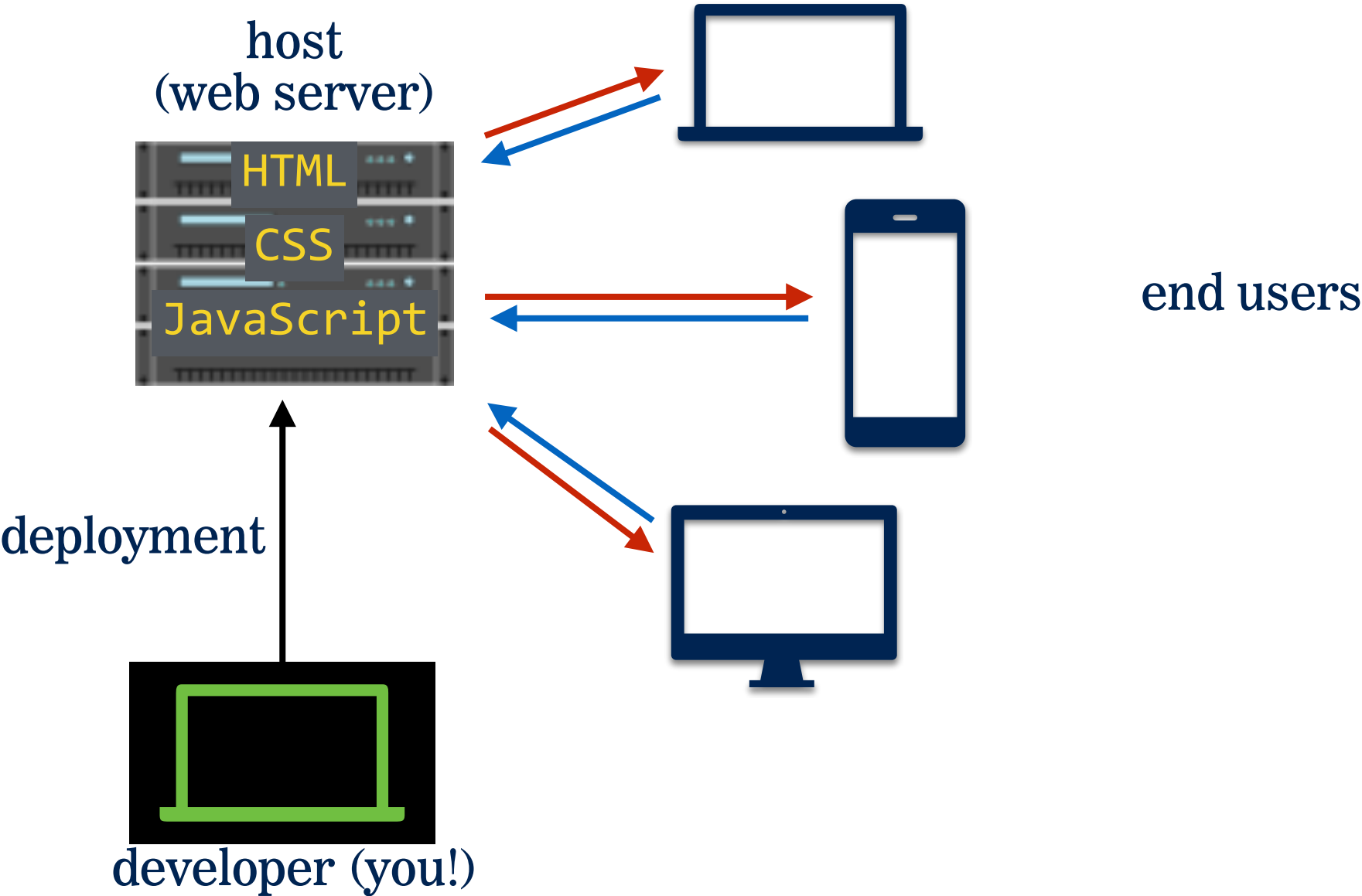
1. At the command line, navigate to the folder containing your Firebase project.
2. Add polyfills to enable Fetch in older browsers.
3. If you have access to a browser that does not support Fetch, test your app in that browser and make sure it works
4. Also test your app in a modern browser and ensure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

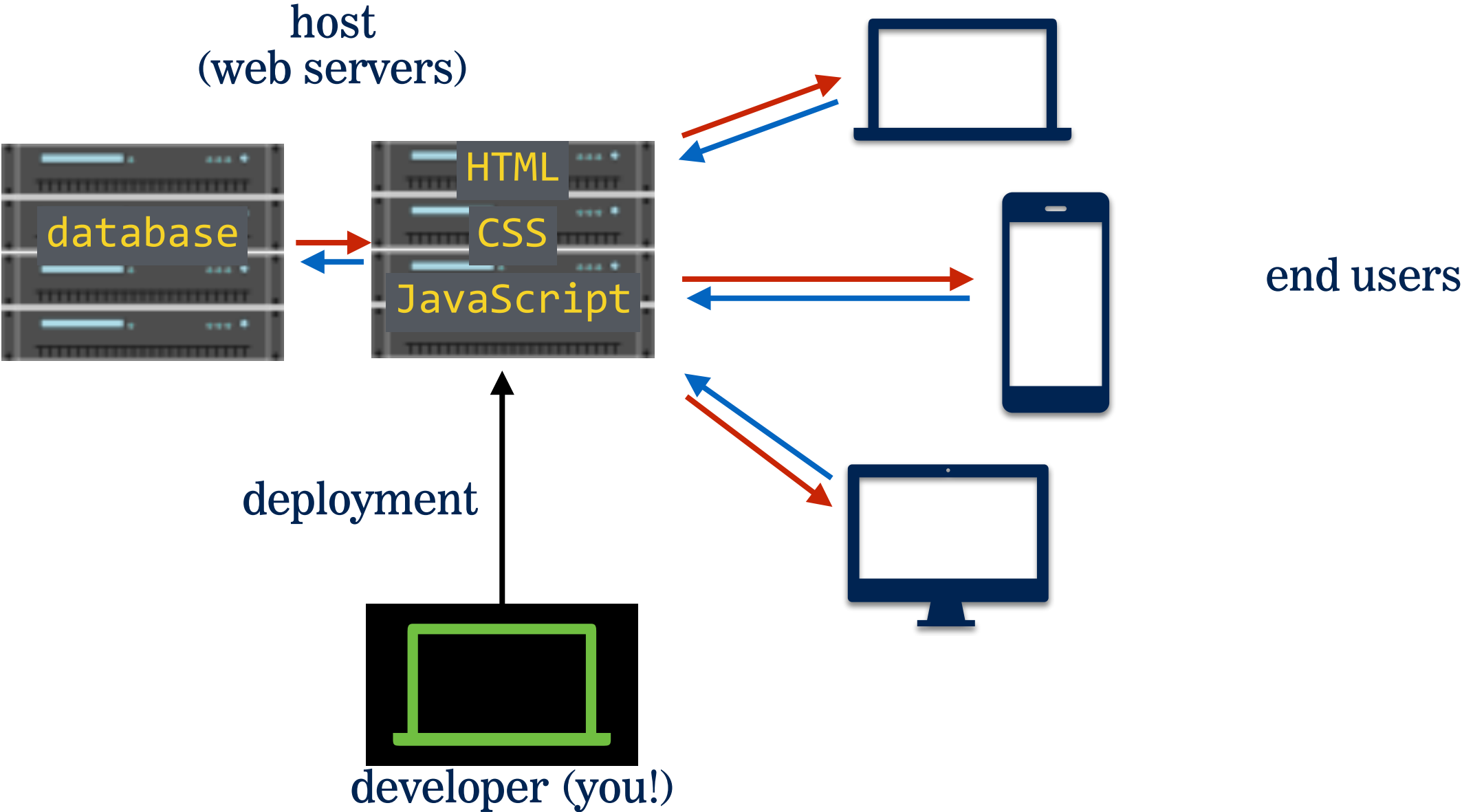
---

**DEPLOYING YOUR APP**

---







# **DEPLOYMENT**





## ALTERNATIVE “SERVERLESS” SERVICES

**THE MAJOR PROVIDERS**  
The biggest names in serverless, providing wide swaths of functionality.

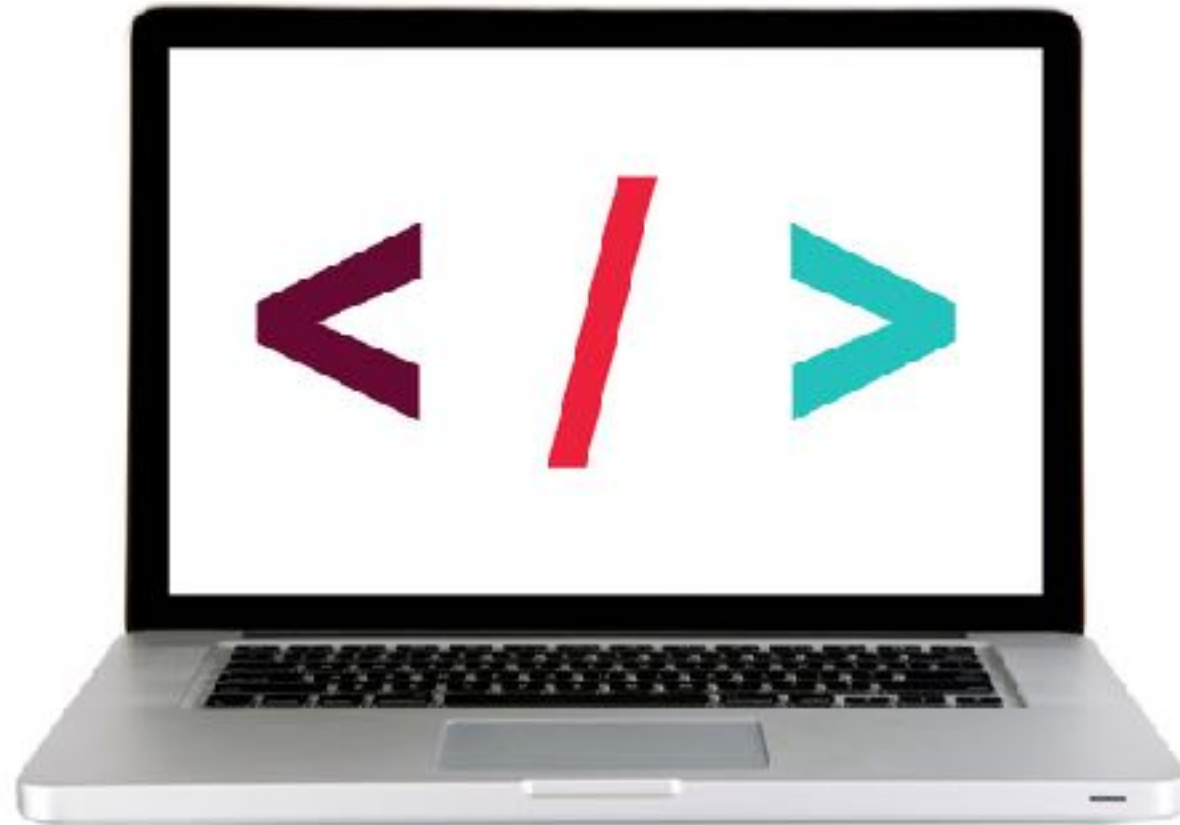
 <p><b>Amazon Web Services</b> <a href="#">Lambda</a> (Cloud Functions), <a href="#">S3</a> (File Storage), <a href="#">Amplify</a>, <a href="#">Console</a> (hosting with Cloud and HTTPS), <a href="#">AppSync</a> (Real-time GraphQL), and <a href="#">Cognito</a> (Auth) are probably the most relevant things to front-end developers. There are frameworks that help you deploy to Lambda, like <a href="#">Amplify</a>, <a href="#">Gatsby</a>, <a href="#">Function</a>, <a href="#">Fleet</a>, <a href="#">Serverless</a>, and <a href="#">Architect</a>.</p>	 <p><b>Google Cloud Platform</b> More of a major infrastructure provider in vein of Amazon Web Services than a toolkit for building out an app like Firebase is.</p>	 <p><b>Google Firebase</b> 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 stuff. It might be expensive to scale or though.</p>
 <p><b>Microsoft Azure</b> A major infrastructure provider with solutions for almost everything, and generally considered the least expensive. For working with cloud functions, there is an online editor, but it also allows GitHub sync and <a href="#">integrates directly with VS Code</a>. Data storage is through Cosmos DB.</p>	 <p><b>Netlify</b> 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.</p>	 <p><b>ZEIT</b> 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.</p>

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

---

## LET'S TAKE A CLOSER LOOK

---



---

## 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.firebaseio.com`

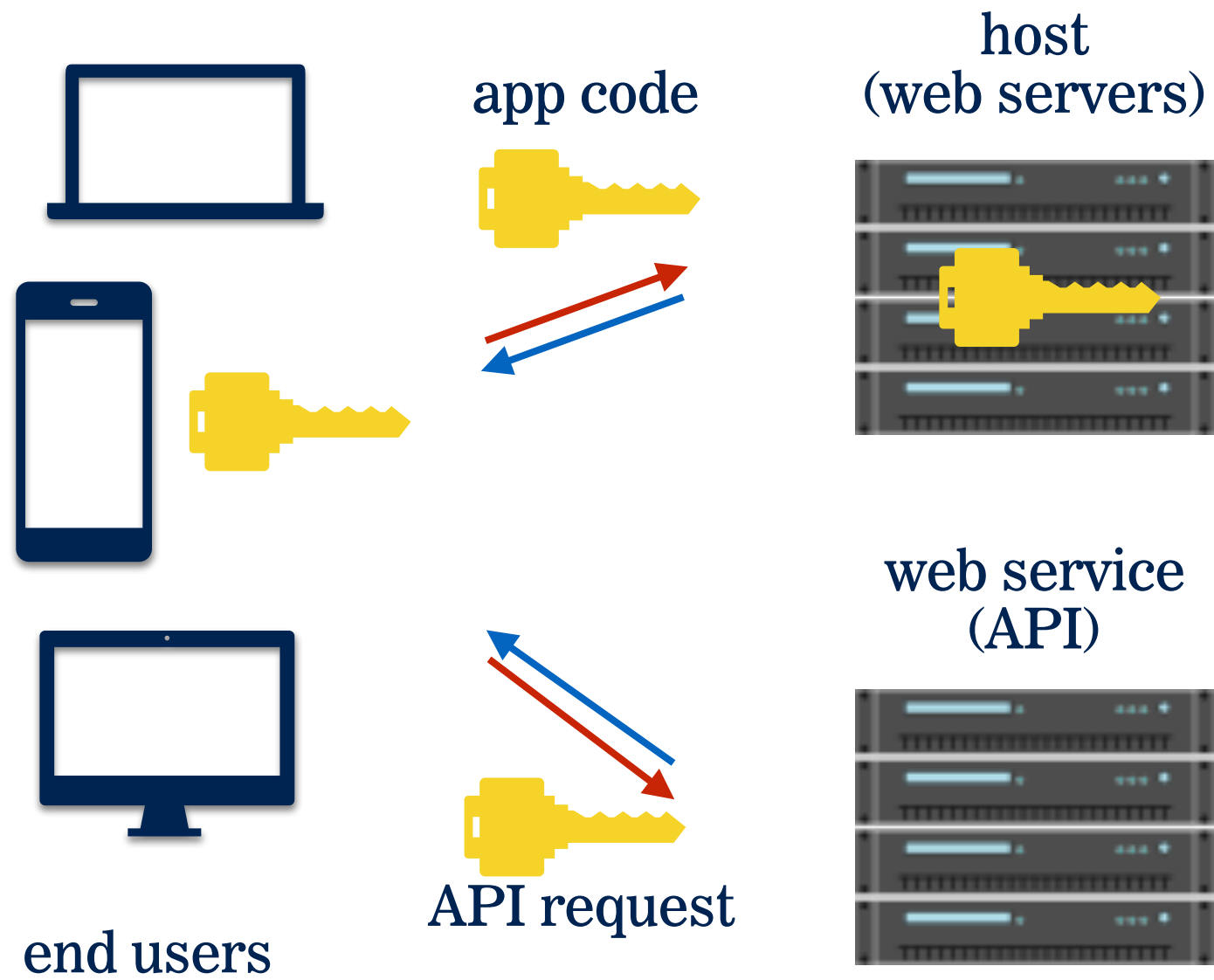


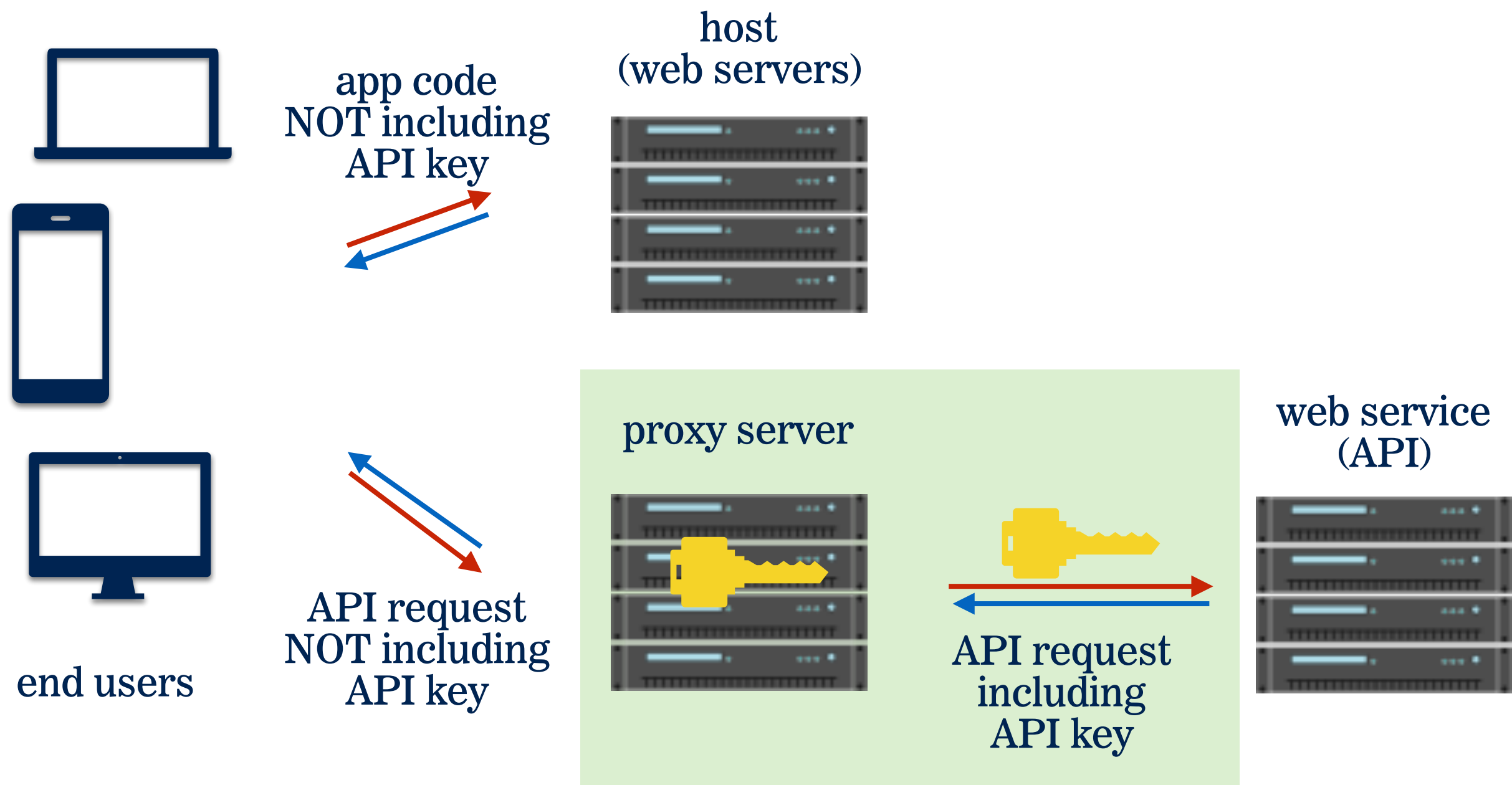
---

**DEPLOYING YOUR APP**

---

# **PROXY SERVERS**

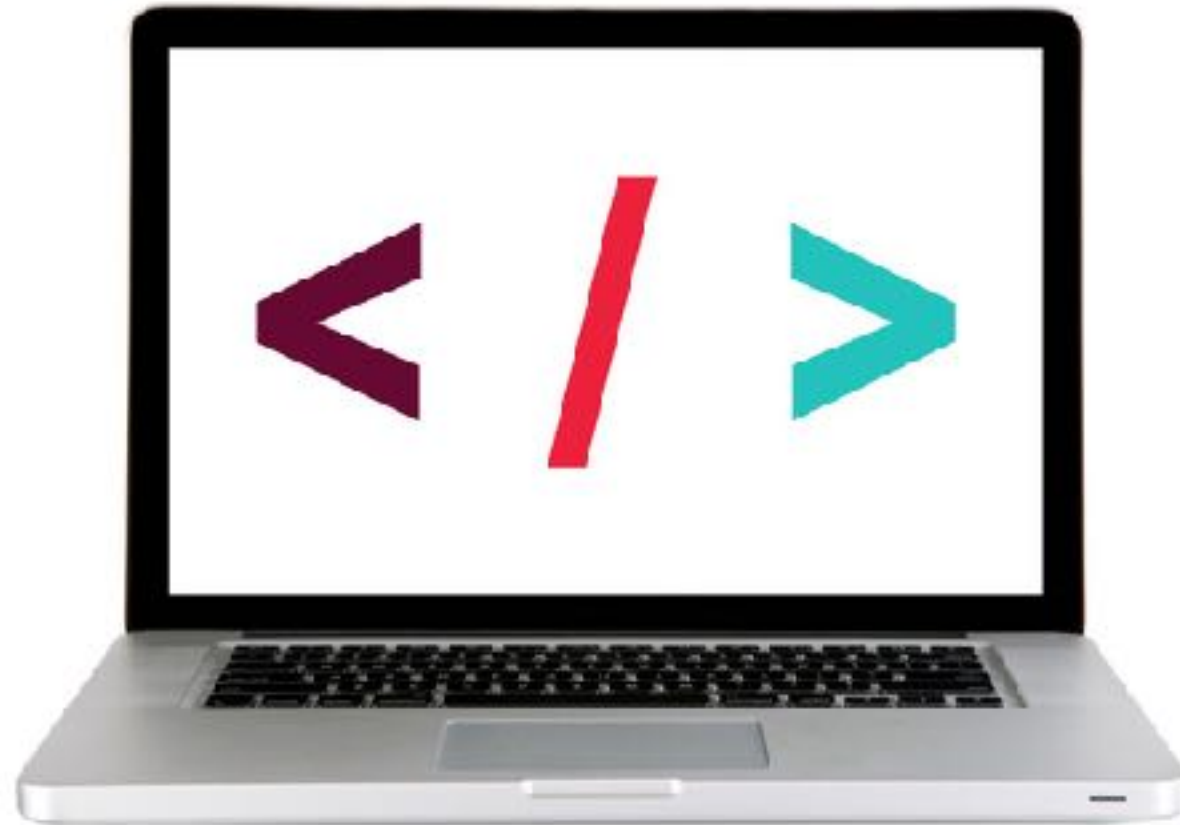




---

## LET'S TAKE A CLOSER LOOK

---



# **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.

# **NEXT CLASS PREVIEW**

## **Final Project Lab**

# Q&A