GENERAL ASSEMBLY

JAVASCRIPT DEVELOPMENT

Sasha Vodnik, Instructor

HELLO!

1. Pull changes from the `svodnik/JS-SF-7` repo to your computer
2. Open the `starter-code` folder in your code editor

SCOPE AND CLOSURES

LEARNING OBJECTIVES

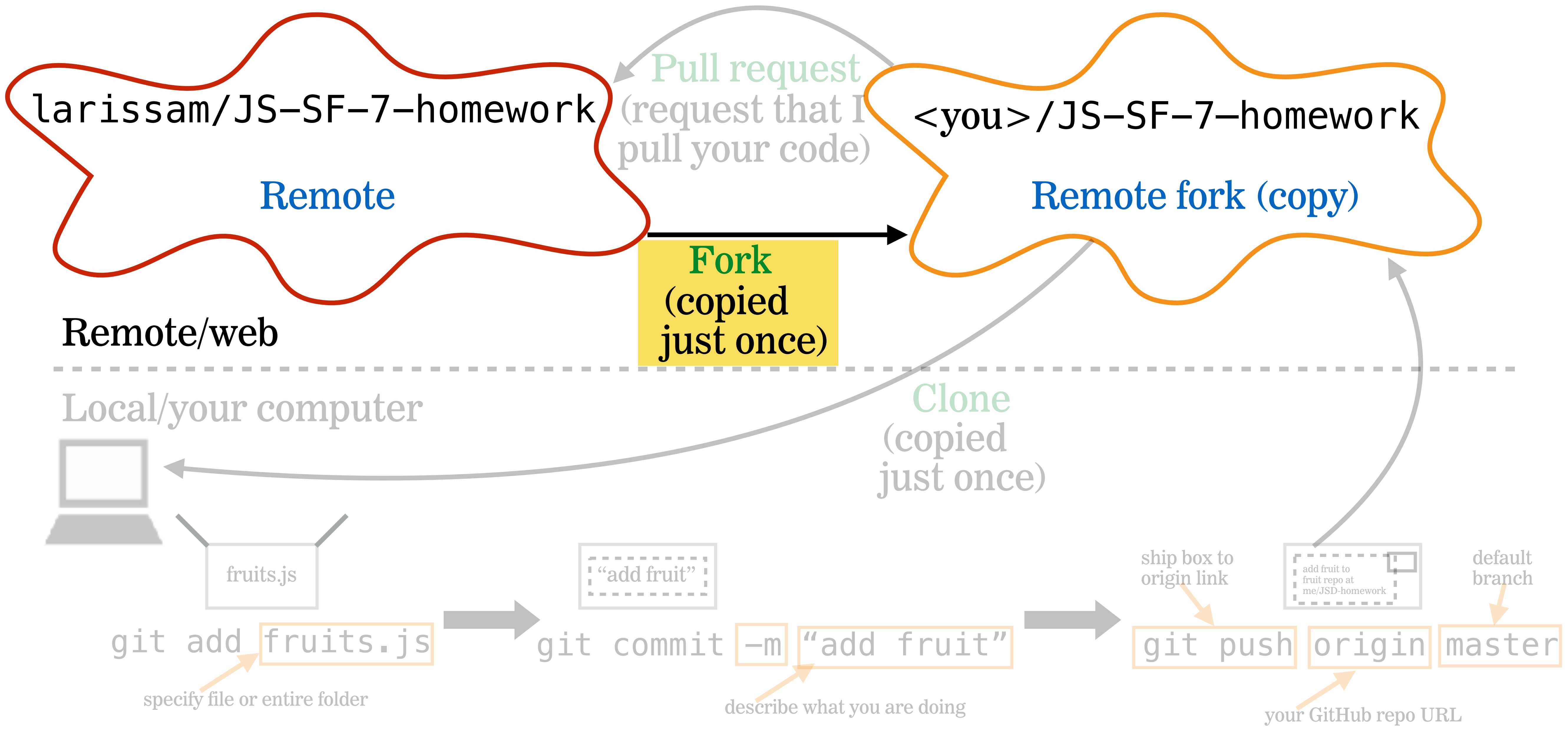
At the end of this class, you will be able to

- Determine the scope of local and global variables
- Create a program that hoists variables
- Understand and explain closures

AGENDA

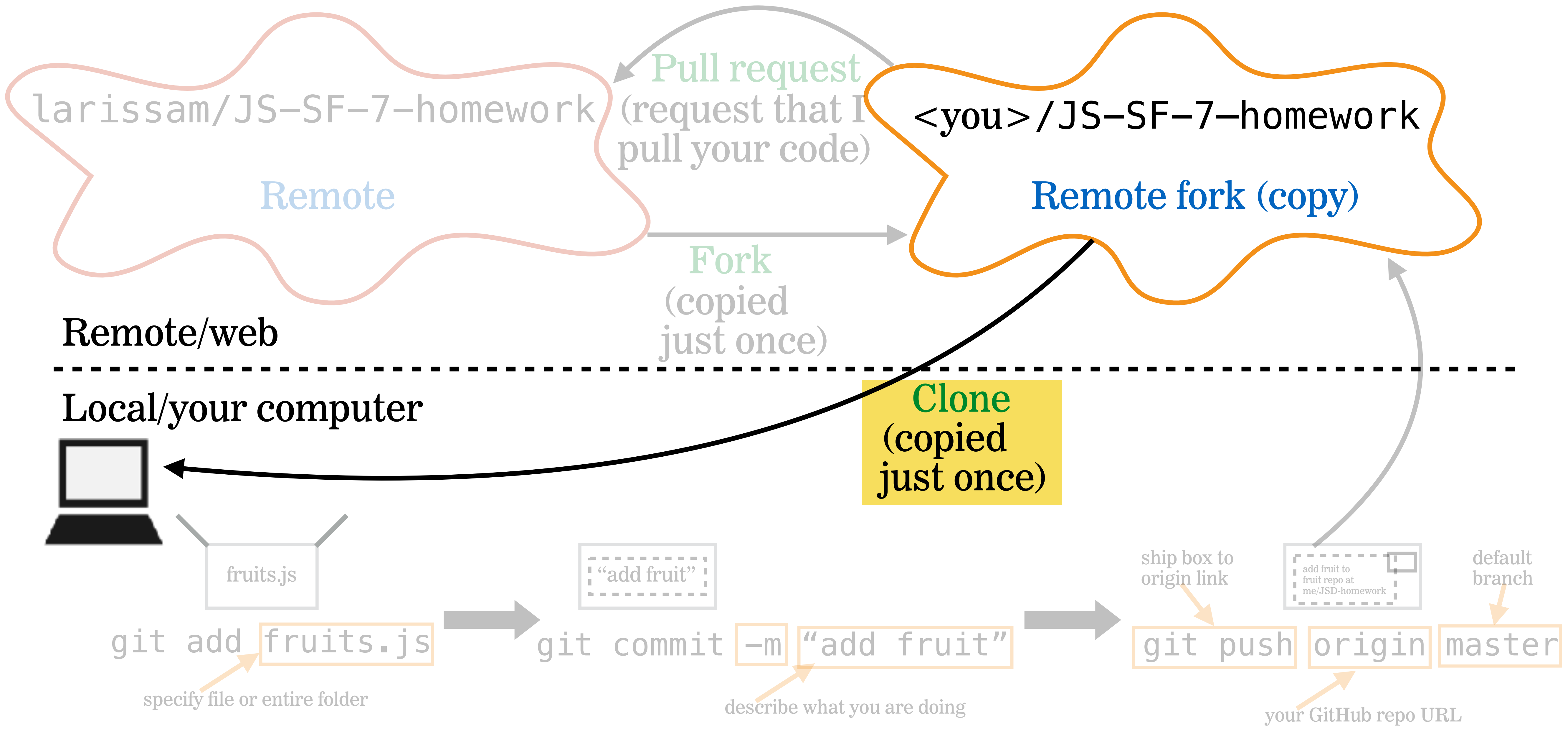
- Submit homework
- Variable scope
- The `var`, `let`, and `const` keywords
- Hoisting
- Closures

USING THE JS-SF-7-HOMEWORK REPO



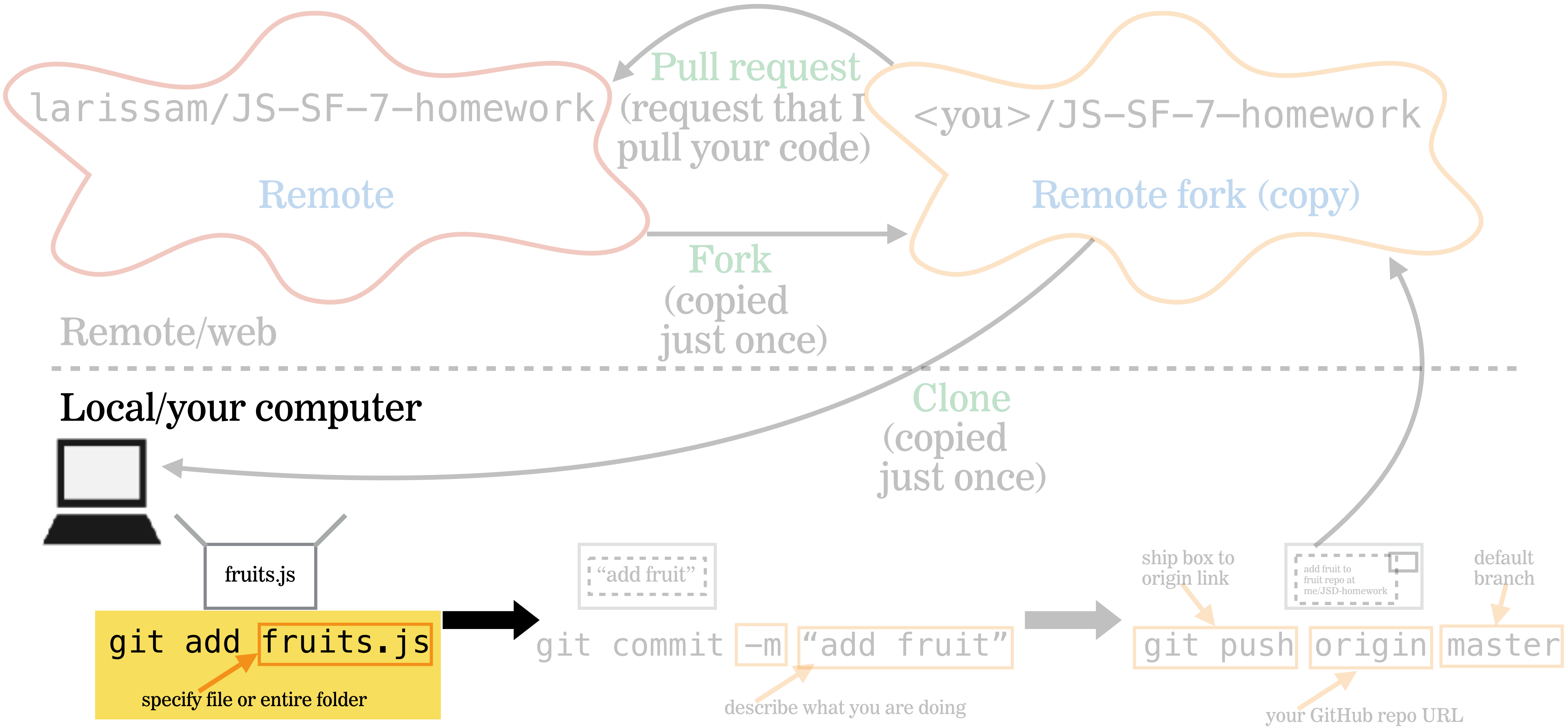
```
git add fruits.js
git commit -m "add fruit"
git push origin master
```

USING THE JS-SF-7-HOMEWORK REPO

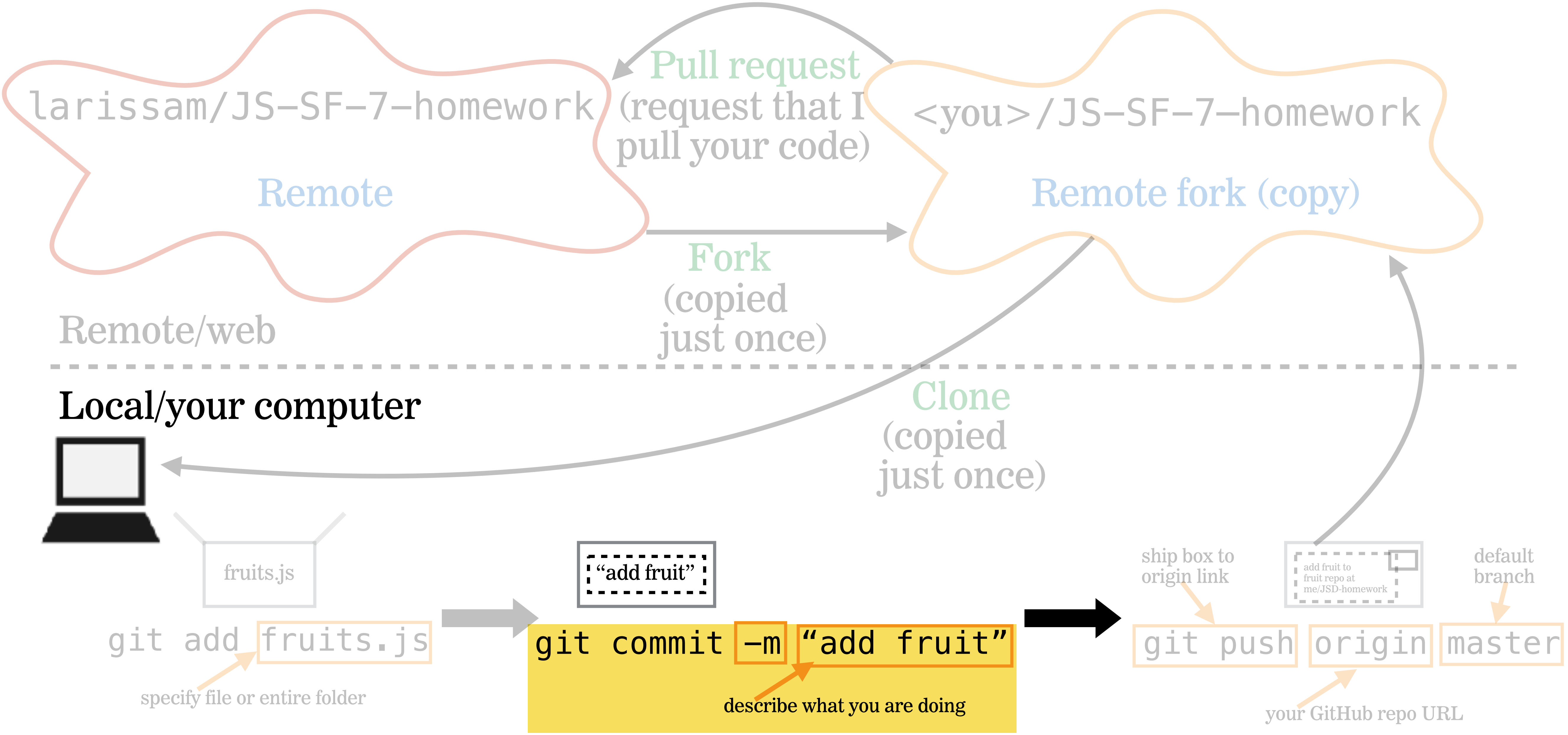


```
git add fruits.js
git commit -m "add fruit"
git push origin master
```

USING THE JS-SF-7-HOMEWORK REPO

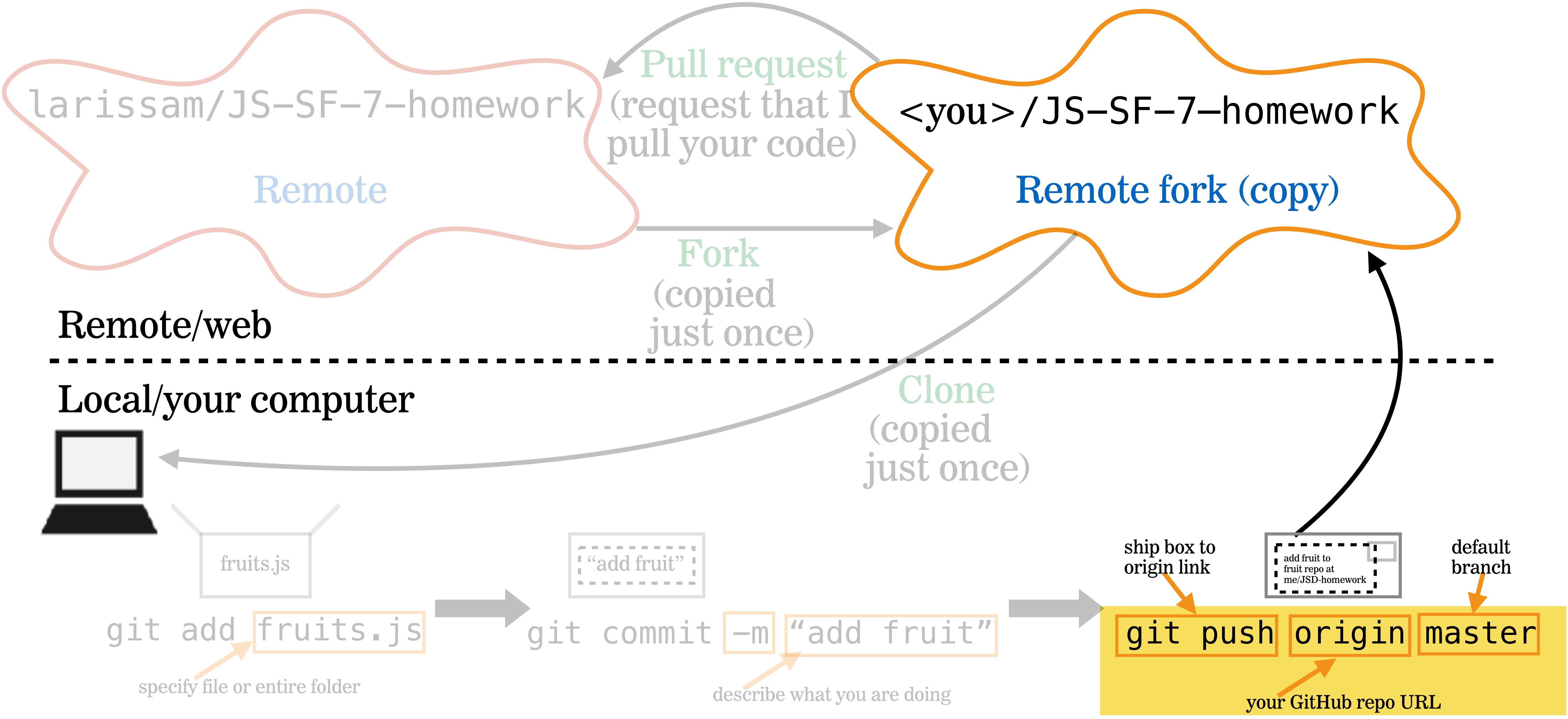


USING THE JS-SF-7-HOMEWORK REPO

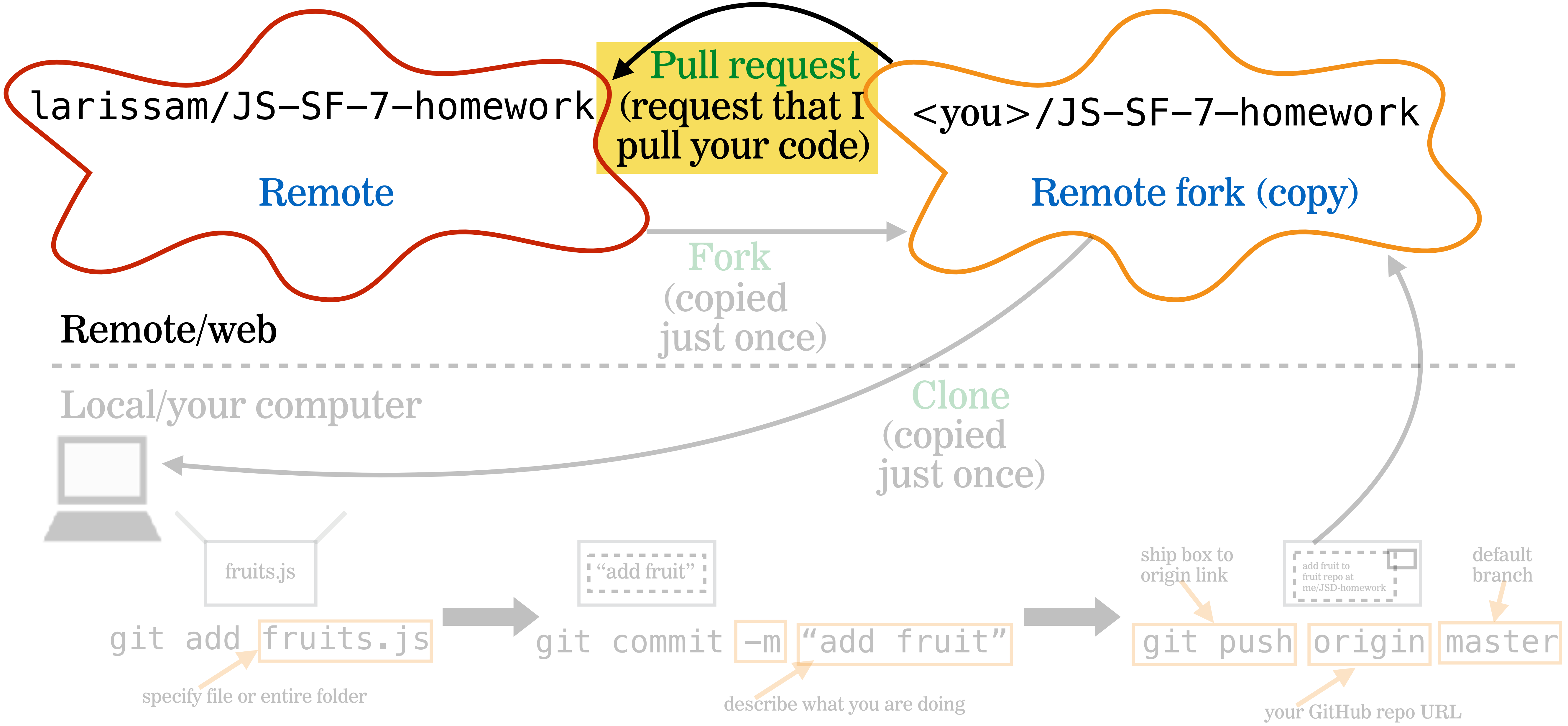


```
git add fruits.js
git commit -m "add fruit"
git push origin master
```

USING THE JS-SF-7-HOMEWORK REPO



USING THE JS-SF-7-HOMEWORK REPO



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

SUBMIT HOMEWORK: STEP 1

In Finder:

- navigate to *`firstname-username`* folder (example: Sasha-svodnik)
- create `HW_1` folder
- move your completed homework into `HW_1`

SUBMIT HOMEWORK: STEP 2

In Terminal:

- navigate to *`firstname-username`* folder
- `git add .`
- `git commit -m “submitting homework 1”`
- `git push origin master`

SUBMIT HOMEWORK: STEP 3

In Browser:

- Go to your fork of `JS-SF-7-homework` on github.com
- click **New pull request**
- click **Create pull request**
- click **Create pull request** (again)

HOMEWORK — GROUP DISCUSSION

EXERCISE

TYPE OF EXERCISE

- Groups of 3

TIMING

10 min

1. Pick someone to take notes for your group.
2. Share 1 thing you’re excited about being able to accomplish.
3. Have each person in the group note 1 thing they found challenging for the exercises and make note. Discuss as a group how you think you could solve that problem.
4. Did you complete the bonus exercise? Demonstrate it and show your group how you did it!

EXIT TICKET FEEDBACK

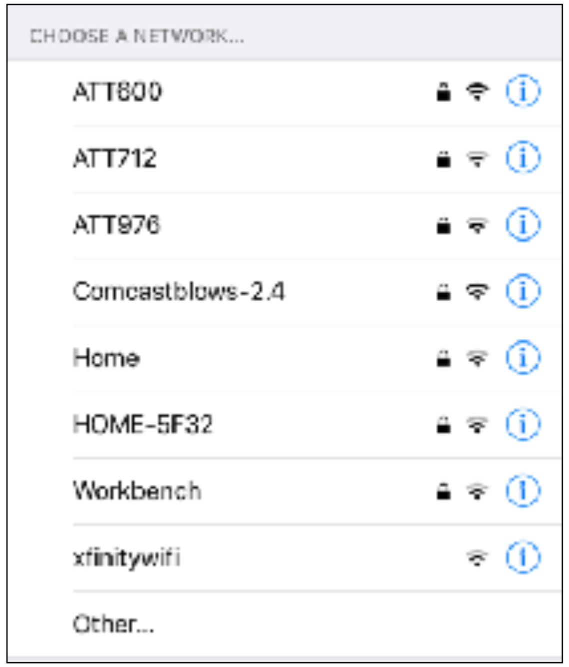
- For loops, why is command sometimes in brackets and sometimes in parentheses?
- Requests for more examples:
  - I had trouble following the lesson when we got into functions/objects. It's helpful for me to have visual examples of how the concepts relate to each other &/or real world examples.
  - I'd like more examples and be able to have you live code examples or provide code examples/solution code after class

Checkin and questions

- The **most significant thing I learned** about using Conditionals and Functions is ___________.
- My **biggest outstanding question** about using Conditionals and Functions is ___________.

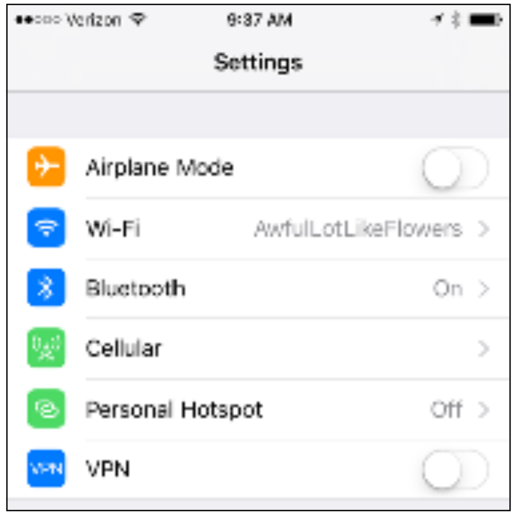
Why do we use different networks to connect to the Internet when we're in different places?

- home
- GA
- in a car/on MUNI



CHOOSE A NETWORK...

ATT800
ATT712
ATT976
Comcastblows-2.4
Home
HOME-5F32
Workbench
xfinitywifi
Other...



●●○○○ Verizon 9:37 AM

Settings

Airplane Mode	
Wi-Fi	AwfulLotLikeFlowers
Bluetooth	On
Cellular	
Personal Hotspot	Off
VPN	

SCOPE

SCOPE

- Describes the set of variables you have access to

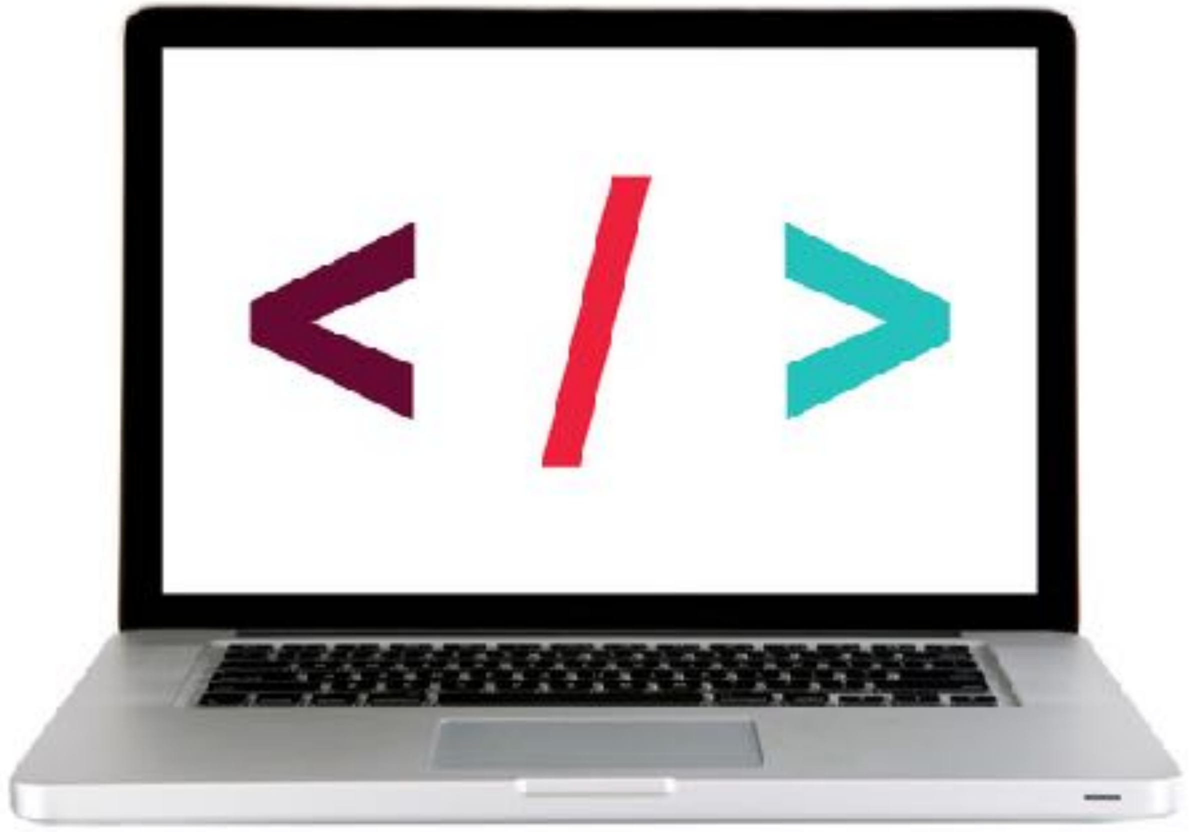
GLOBAL SCOPE

- A variable declared outside of a function is accessible everywhere, even within functions. Such a variable is said to have **global scope**.

LOCAL SCOPE

- A variable declared within a function is not accessible outside of that function. Such a variable is said to have **local scope**.

LET'S TAKE A CLOSER LOOK



EXERCISE — SCOPE

EXERCISE

KEY OBJECTIVE

- Determine the scope of local and global variables

TYPE OF EXERCISE

- Turn and Talk

EXECUTION

3 min

1. Describe the difference between global and local scope
2. Collaborate to write code that includes at least one variable with local scope and one variable with global scope

LAB — SCOPE

KEY OBJECTIVE

- Determine the scope of local and global variables

TYPE OF EXERCISE

- Pairs

LOCATION

- starter code > 1-scope-lab

EXECUTION

5 min

1. Open the index.html file in your browser, view the console, and examine the error.
2. Follow the instructions in js > main.js to complete parts A and B.

`var`, `let`, `const`, AND SCOPE

- `var` obeys the scoping rules we've just seen
    - "generic" way to create variables
- `let` and `const` are newer keywords with different scoping rules
    - local scope within functions **and** within any block (including loops and conditionals)

`let`

- used in the same situations as `var`, but with different scoping rules for code blocks

```
let results = [0,5,2];
```

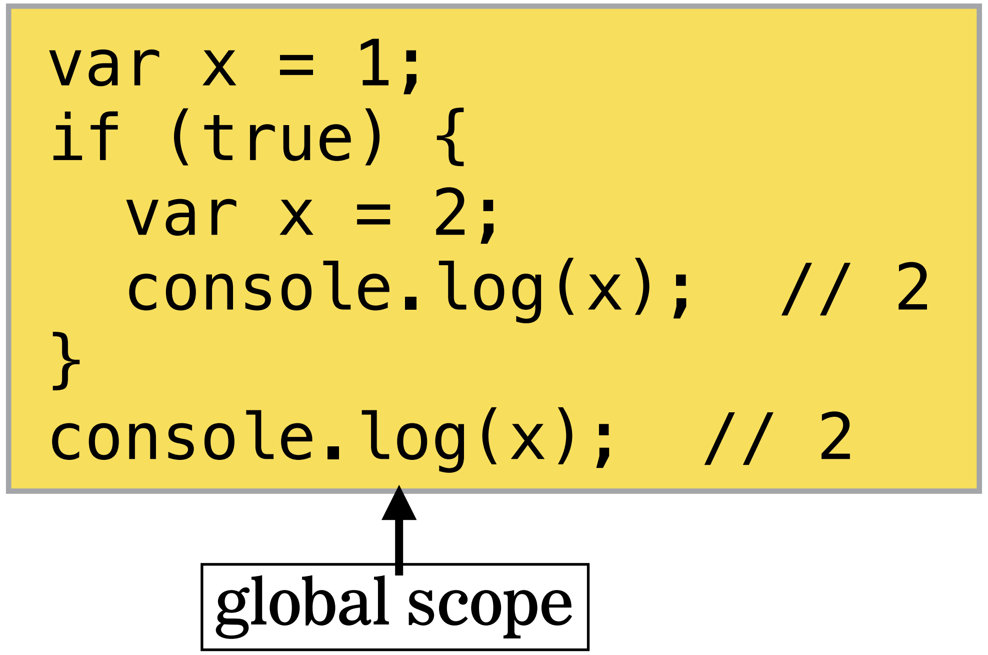
`const`

- used to declare constants
  - once you've declared a value using `const`, you can't change the value in that scope
- by convention, constant names use all capital letters

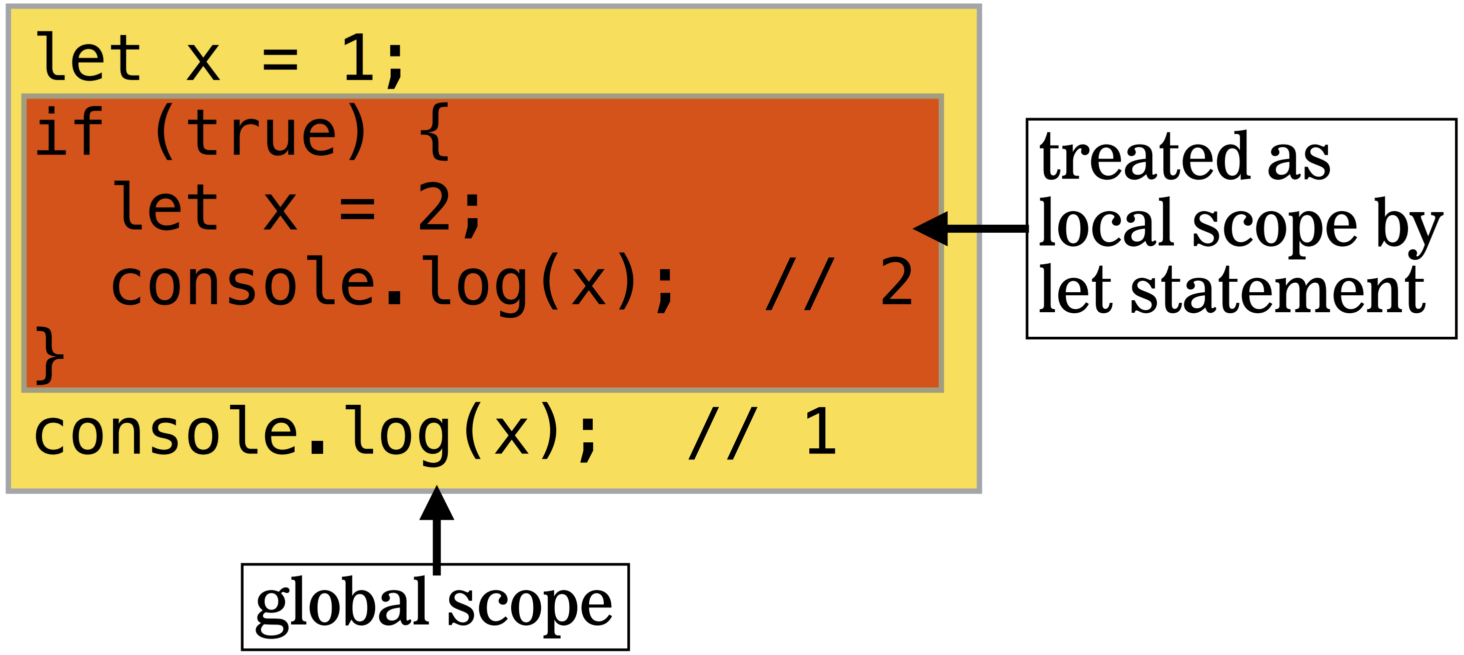
```
const SALESTAX = 0.0875;
```

`let/const vs var`

- `let` & `const` create local scope within any block (including loops and conditionals) but `var` does not



```
var x = 1;
if (true) {
  var x = 2;
  console.log(x);  // 2
}
console.log(x);  // 2
```

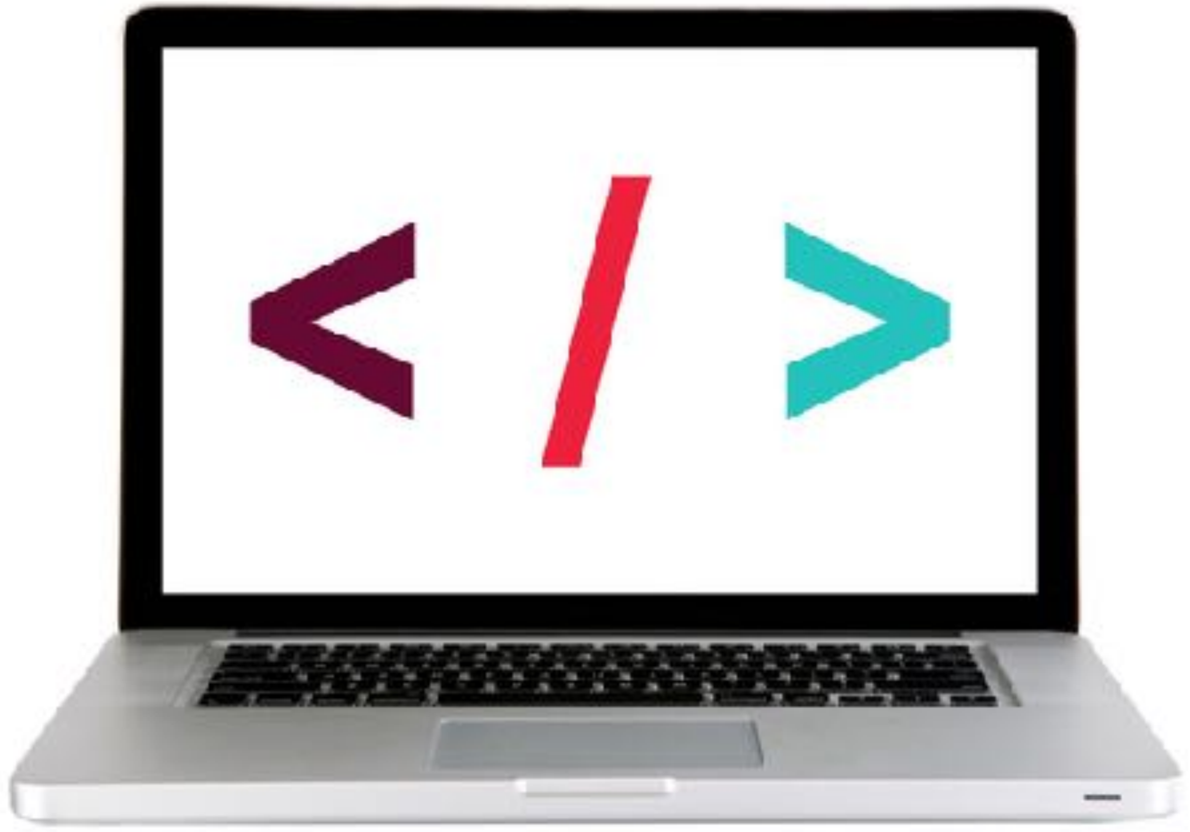


```
let x = 1;
if (true) {
  let x = 2;
  console.log(x);  // 2
}
console.log(x);  // 1
```

`var`, `let`, `const`, AND BROWSER SUPPORT

- `let` and `const` are not supported by older browsers
  - see caniuse.com, search on `let`
- babel.js (babeljs.io) allows you to transpile newer code into code that works with older browsers as well
- we will use `var` in class, but feel free to explore `let` and `const` on your own

LET'S TAKE A CLOSER LOOK



EXERCISE — VAR, LET, AND CONST

EXERCISE

KEY OBJECTIVE

- Distinguish between `var`, `let`, and `const`

TYPE OF EXERCISE

- Individual or pairs

EXECUTION

2 min

1. Draw the table shown on the whiteboard, which compares a few aspects of var, let, and const usage.
2. Complete the table.

`var`, `let`, AND `const`

keyword	local scope	can you change the value in the current scope?	browser support
`var`	within the code block of a **function** only	yes	all browsers
`let`	within any code block	yes	**only modern browsers**
`const`	within any code block	**no**	**only modern browsers**

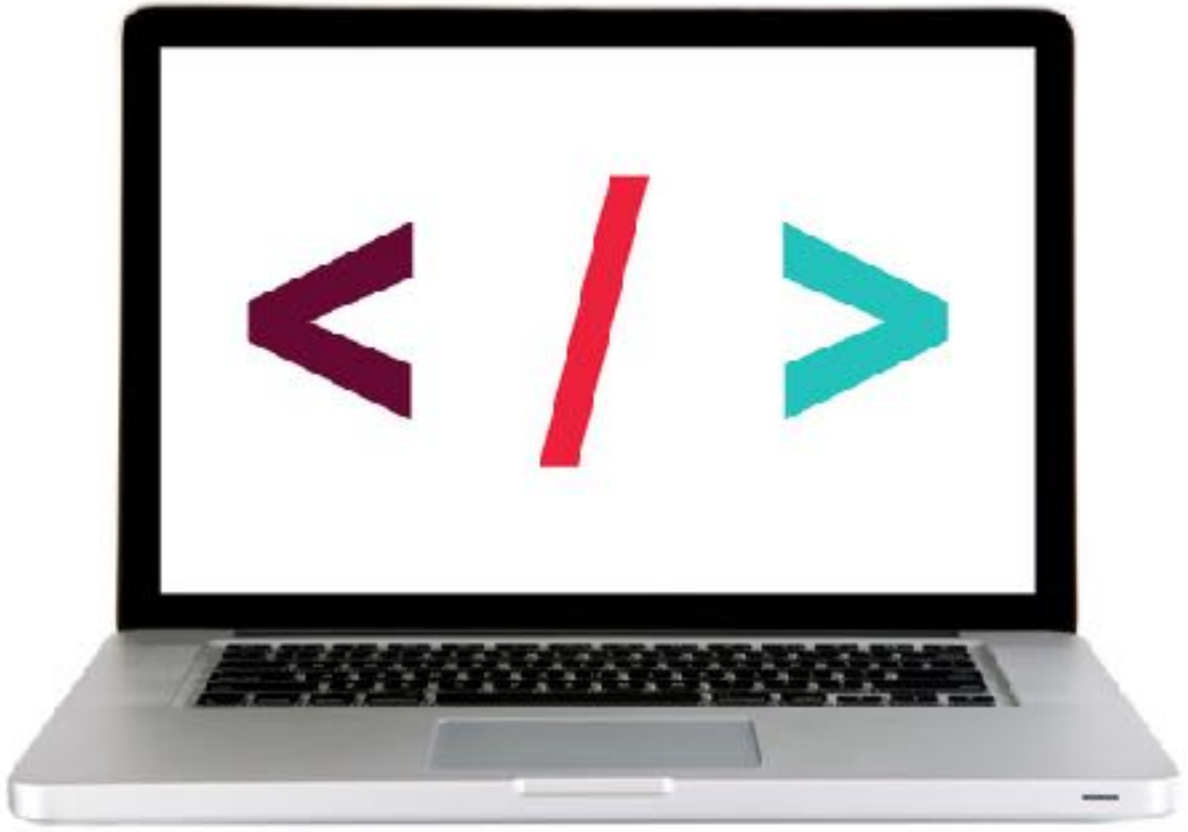
HOISTING

- JavaScript’s behavior of moving declarations to the top of a scope.
- This means that you are able to use a function or a variable before it has been declared.

FUNCTIONS AND HOISTING

- Function expressions are treated like other variables
  - only the name is hoisted, not the value
- Function declarations are treated differently
  - the code for the entire function is hoisted along with a function declaration

LET'S TAKE A CLOSER LOOK



EXERCISE — HOISTING

EXERCISE

KEY OBJECTIVE

- Create a program that hoists variables

TYPE OF EXERCISE

- Groups of 3

EXECUTION

2 min

1. Examine the code on the whiteboard.
2. Discuss with your group which parts of the code are hoisted.
3. Predict the result of each of the first four statements.

CLOSURES

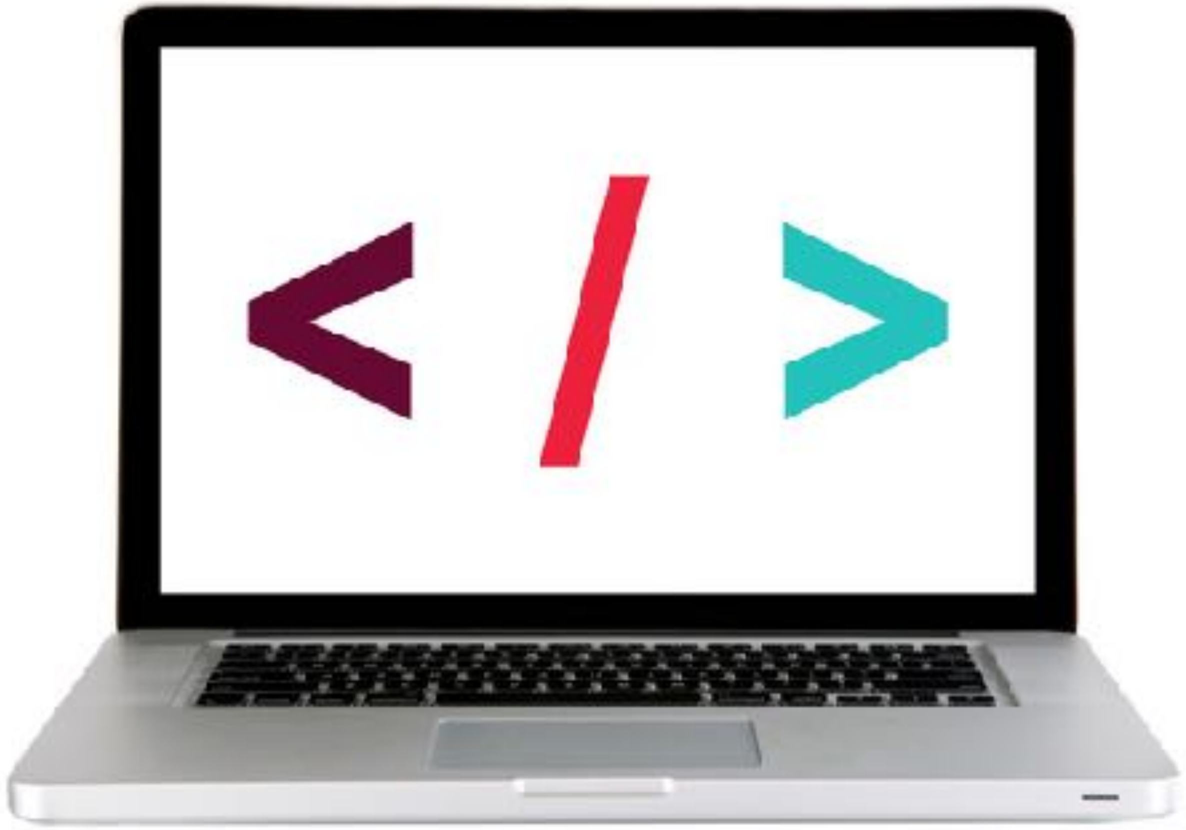
BUILDING BLOCKS OF CLOSURES

- nested functions
- scope
  - inner function has access to outer function’s variables
- `return` statements
  - inner function returning reference to outer function’s variables

CLOSURES

- A **closure** is an inner function that has access to the outer (enclosing) function’s variables.
- You create a closure by adding a function inside another function.
- A closure is also known as **lexical scope**

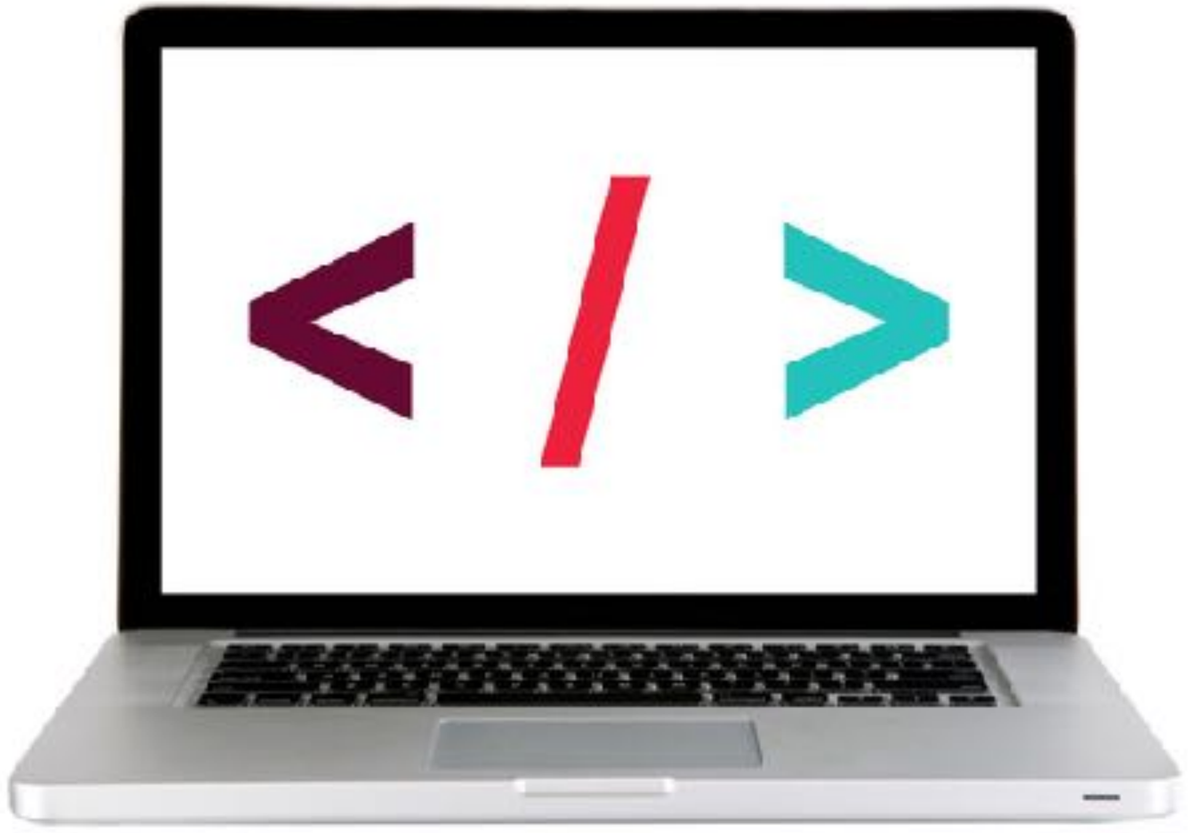
LET'S TAKE A CLOSER LOOK



CLOSURES — KEY POINTS

- Closures have access to the outer function's variables (including parameters) **even after the outer function returns**.
- Closures store **references** to the outer function's variables, not the actual values.

LET'S TAKE A CLOSER LOOK



WHAT ARE CLOSURES USED FOR?

- Turning an outer variable into a private variable
- Namespacing private functions

LAB — CLOSURES

KEY OBJECTIVE

- Understand and explain closures

TYPE OF EXERCISE

- Pairs

LOCATION

- starter-code > 3-closures-lab

EXECUTION

Until 9:20

1. Follow the instructions in app.js to build and test code that uses a closure.

LEARNING OBJECTIVES - REVIEW

- Determine the scope of local and global variables
- Create a program that hoists variables
- Understand and explain closures

NEXT CLASS PREVIEW

Hubot Lab

- Install and configure all utilities needed to run a Hubot
- Write scripts that allow your Hubot to interact with users of the class Slack organization

Exit Tickets!

Q&A