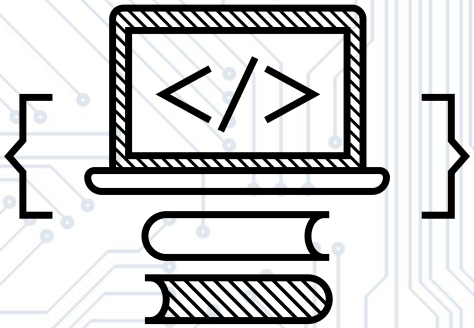
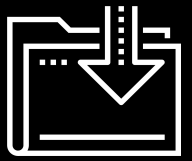


jQuery Jubilee

Web Development Boot Camp

Lesson 2.5



This shouldn’t be you:

Remember This:

> You can't tell whether you're learning something when you're learning it—in fact, learning feels a lot more like frustration.
>
> What I've learned is that during this period of frustration is actually when people improve the most, and their improvements are usually obvious to an outsider. If you feel frustrated while trying to understand new concepts, try to remember that it might not feel like it, but you're probably rapidly expanding your knowledge.
>
> —Jeff Dickey, author of *Write Modern Web Apps with the MEAN Stack: Mongo, Express, AngularJS, and Node.js*

Important Reminders

This course covers a lot of material quickly, so remember:

- Instructors and TAs are here to help.
- Feel encouraged to schedule a one-on-one during office hours.
- One-on-one sessions are a great way to identify weaknesses and outline a plan to get back on track.
- Office hours are held before and after class.

Today’s Class

Objectives

01 Pretend to learn scoping

02 Build a jQuery calculator

Lexical Scope

This next section is **heavy** on theory.

JavaScript Scope

In Javascript, curly **brackets { }** indicate blocks of code.

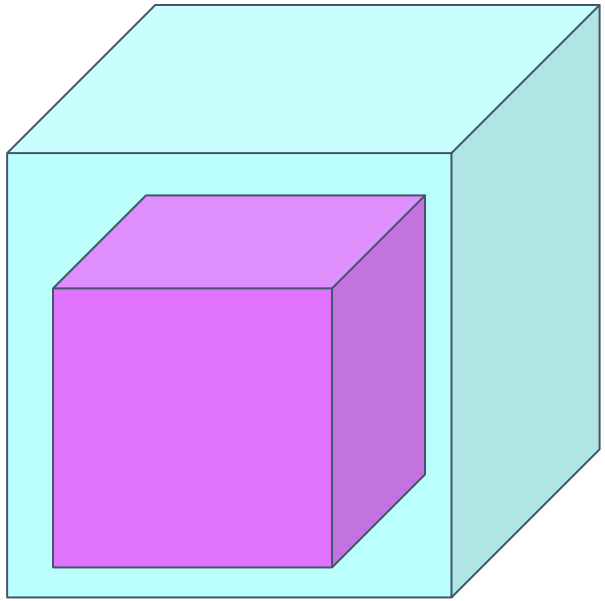
In order for the code inside the curly brackets to be executed, it must meet the condition or be called (example: functions).

These blocks of code can affect variables that were declared outside the curly brackets—so be careful!

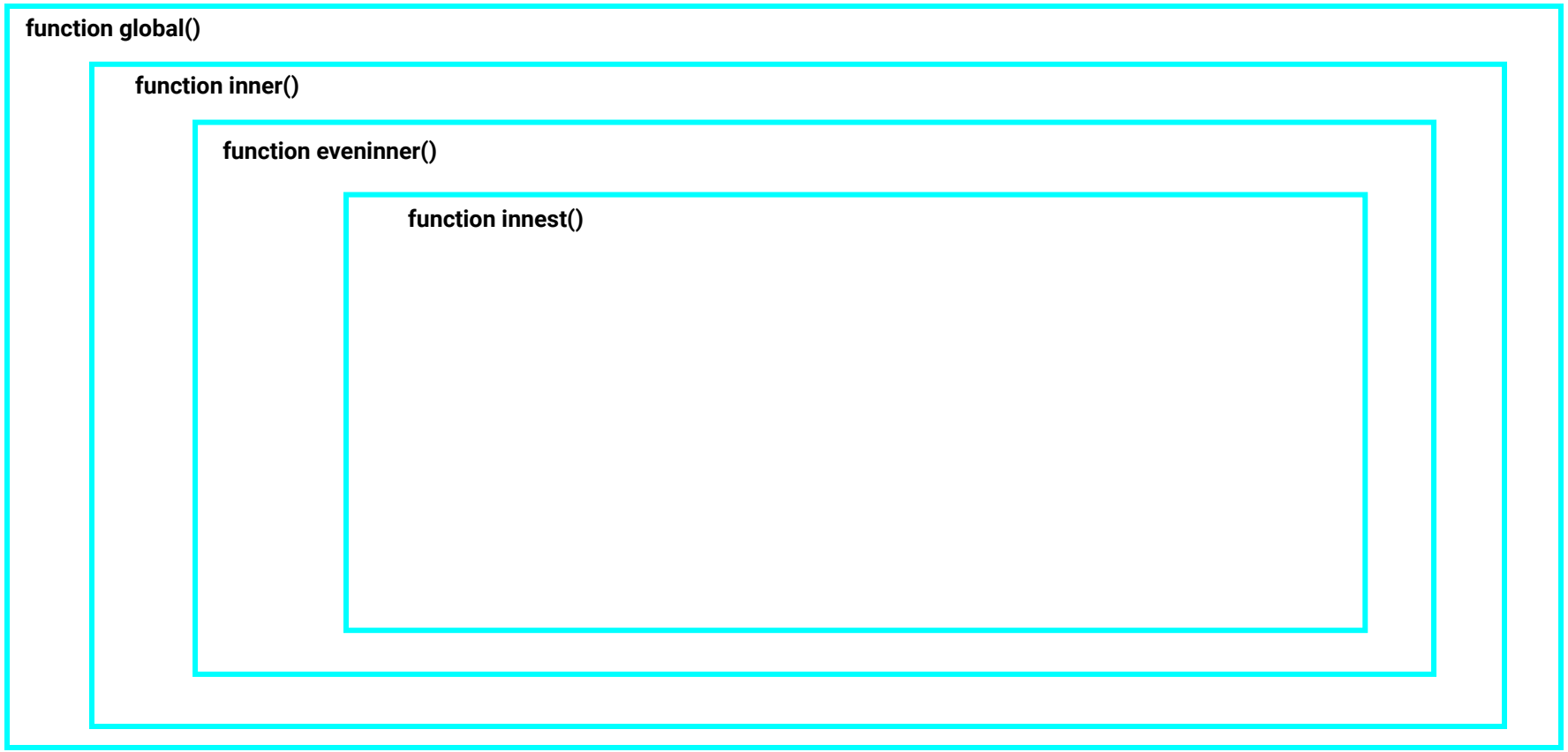
```
// Sets initial value of x
var x = 5;

// False Condition doesn't get run
if(1 > 2000) {
    x = 10
}

// Will print 5. X was unchanged.
console.log(x);
```

Scope = Boxes in Boxes

Scope impacts which variables can be accessed by which function.



Scope = Boxes in Boxes

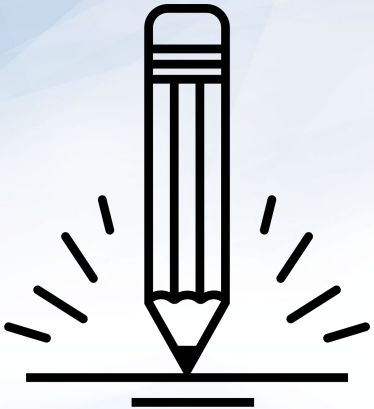


JavaScript Scope Example

Here, **inside** is clearly able to access the variables of its **parent function, outside**.

How does **insideOut** have access to **x**?

```
<script>

  function outside() {

    var x = 1;

    // what is the scope of this function and the scope of y?
    function inside(y) {

      console.log(x + y);
    }

    return inside;
  }

  var insideOut = outside();

  // What does this return?
  insideOut(2);

  // Uncaught ReferenceError: x is not defined.
  // How does insideOut have access to x?
  console.log("The value of 'x' outside 'outside()' is: " + x);

</script>
```



Activity:
Lexical Scope 1

Suggested Time:
5 minutes

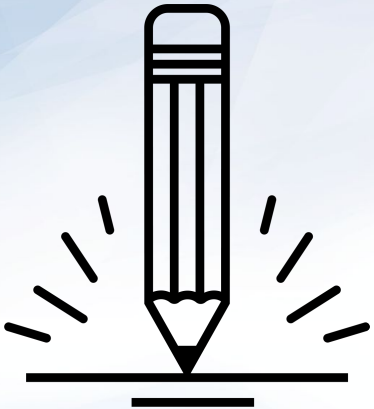
Activity: Lexical Scope 1

Review the file sent to you and explain the following to the person sitting next to you:

- What do the terms *parent function* and *child function* mean?
- Why can child functions access parent variables, but not vice versa?

Be prepared to share your answers!

Suggested Time: 5 minutes



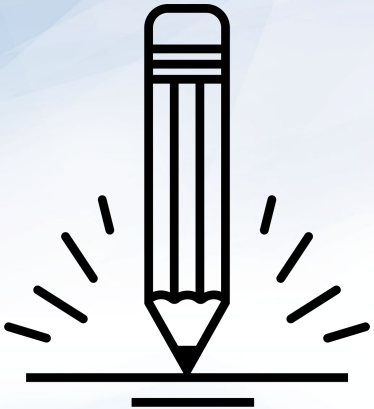
Activity:
Lexical Scope 2

Suggested Time:
5 minutes

Activity: Lexical Scope 2

- Take a few moments to dissect the code just sent to you.
- Try to predict what will be printed in each of the examples.
- Be prepared to share!
- **Note:** Pay attention to the unusual use of the keyword *this*.

Suggested Time: 5 minutes



Activity:
Lexical Scope 3

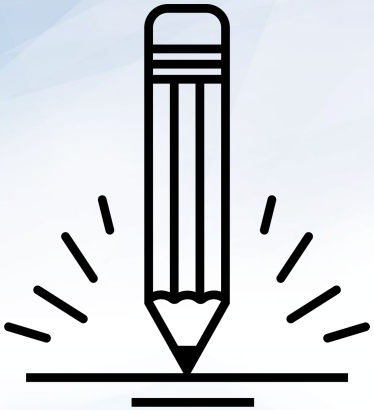
Suggested Time:
7 minutes

Activity: Lexical Scope 3

- Take a few moments to dissect the code just sent to you.
- Try to predict what will be printed in each of the examples.
- Be prepared to share!
- **Note:** Pay attention to the unusual use of the keyword *this*.

Suggested Time: 7 minutes

Time’s Up! Let’s Review.



Partner Activity: Scope Quiz

Instructions and file sent via Slack.

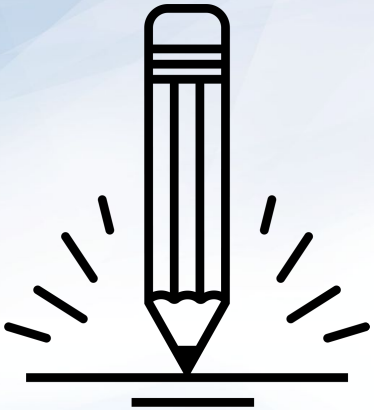
Suggested Time:
7 minutes

Partner Activity: Scope Quiz

- Spend a few moments studying the codefile with the person sitting next to you.
- Then run the program in the browser.
- Once you run the program, you'll find that Code Block 1 leads to different alerts than Code Block 2.
- Ask your partner which Code Block is behaving the way you would expect.
- Then work with your partner to try and identify the specific difference that is causing the issue with the faulty block.
- Once you spot the issue, try to explain to your partner why JavaScript is handling these Code Blocks differently.

Suggested Time: 7 minutes

Time’s Up! Let’s Review.



Partner Activity: This Example

Instructions and file sent via Slack.

Suggested Time:
10 minutes

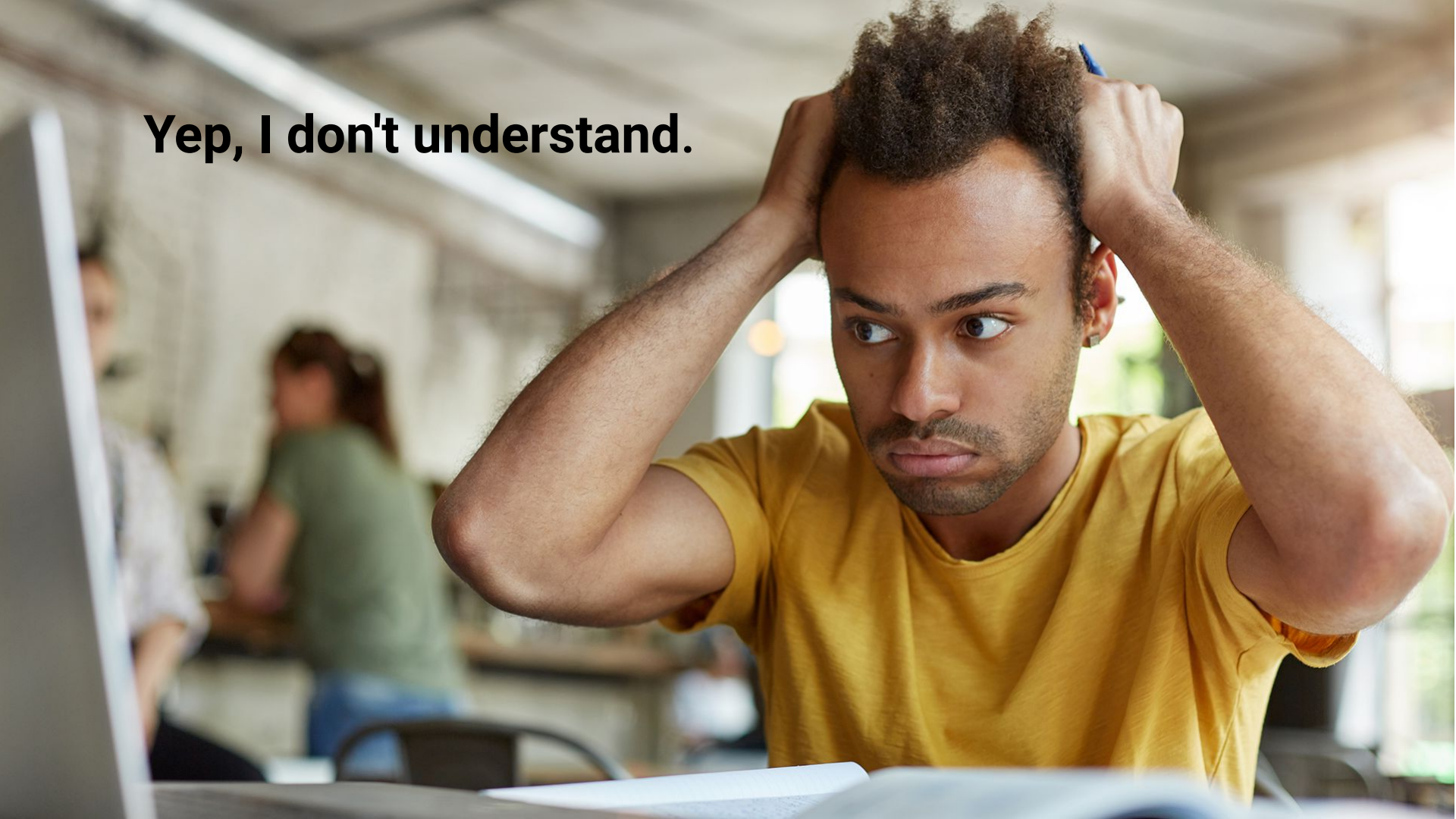
Partner Activity: This Example

Instructions:

- Using the comments in the guide answer each of the questions asked in the file.
- Focus your attention on trying to wrap your mind around the concept of "this" and the unique role it can play in code.
- Then run the program in the browser.
- Then try to explain to your partner how "this" works, focus on the first three examples.

Suggested Time: 10 minutes

Time’s Up! Let’s Review.


Yep, I don't understand.

If you'd like to learn more, here's a helpful article:

What You Should Already Know about JavaScript Scope

spin.atomicobject.com

Questions?