jQuery Calculator

The Coding Bootcamp

No Presentation Today!

Warm-up

```
Write a function that takes in an array of
numbers and outputs the maximum number.

Ex:
Input: [ 1, 2, 3 ]
Output: 3

Input: [ 3, 6, 4, 5, 2, 1 ]
Output: 6

Input: [ 3, 3, 3 ]
Output: 3
```

Class Objectives

- Poll
- Scope Quiz
- 1 on 1 options!
- To build a semi-complex jQuery calculator application **in teams**

Poll Everywhere!

https://pollev.com/chriswoolcot778

or

Text **CHRISWOOLCOT778** to **37607**

then **1,2,3,4,5**

Find Nested Item in Object

```
var garage = {
        carsInGarage: {
          carMakes: ["Maza", "Ford", “Honda"]
        }
  }

  //Show last car type in the garage
  console.log(garage.carsInGarage.carMakes[2])
```

Partners Do: Scope Quiz (17-ScopeQuiz)

- 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.

Partners Do: “This” Example (18-ThisExample)

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 try to explain to your partner how "this" works, focus on the first three examples.

Students Do: Cobwebs (10 mins 19-Cobweb)

- Create the missing the code such that you can retrieve the requested item from the `theCobWeb` object.

- Note: This exercise is actually very relevant to work as a web developer, as data is often relayed across websites in the form of deeply nested JavaScript objects like this one.

- Bonus: If you finish early, begin pondering the bonus item. This is a **very** challenging exercise. It's impossible to complete in the allotted time. If you're feeling valiant - complete it outside of class and come back to instructors/TAs to go over it. This will arm you for difficult interview questions in the future.

jQuery Calculator vs 1:1 option

jQuery Calculator Challenge Excercise

Phase 1: Getting Situated + Pseudocoding Stage (20 mins)

Phase 2: Begin Logic (30 mins)

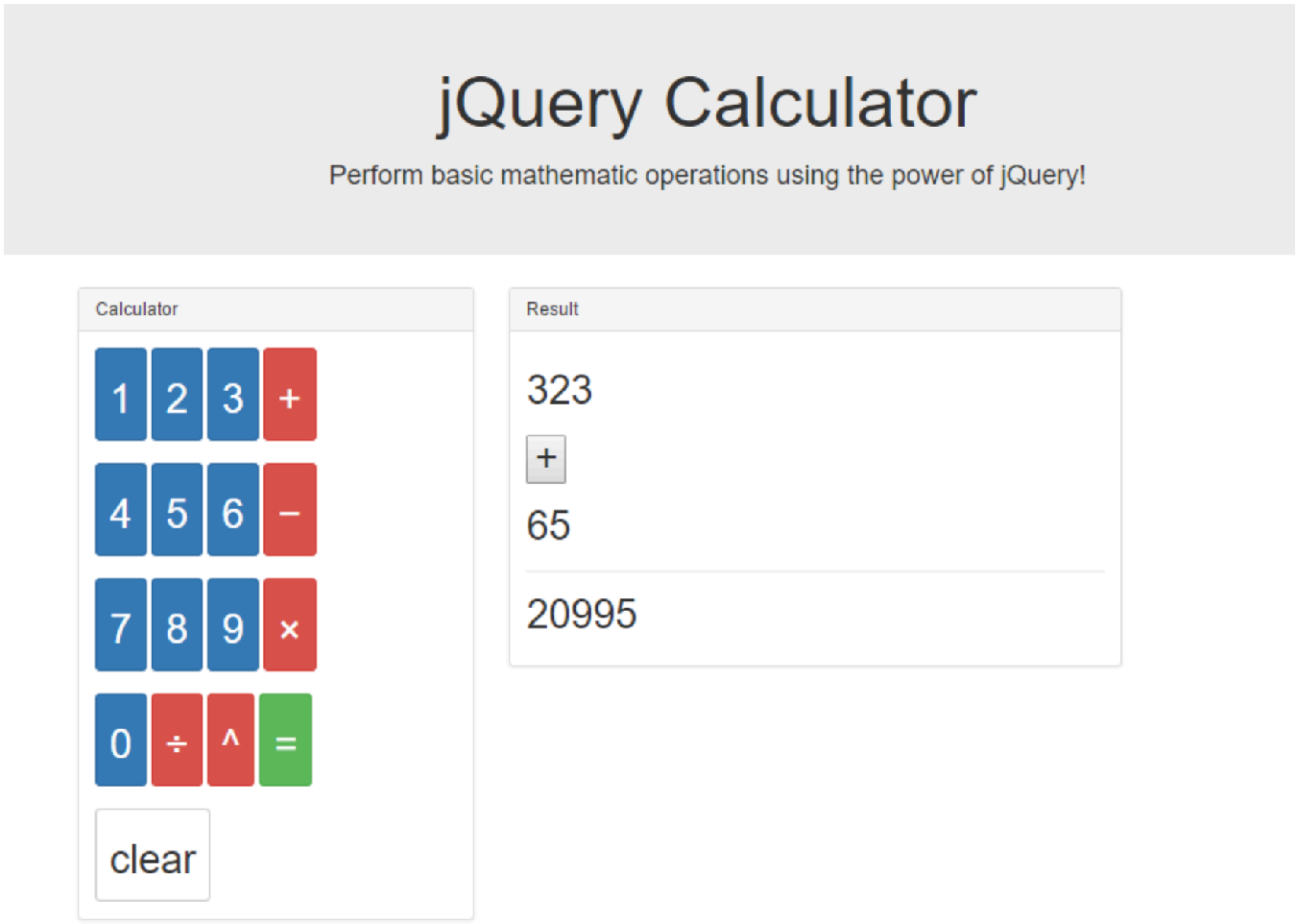
Lunch (35 mins)

Phase 3: Continue Logic Building (35 mins)

Phase 4: Refinement (25 mins)

1:1 Option!

Catch-up" sessions as “crash-courses" with a TA



jQuery Calculator

Perform basic mathematic operations using the power of jQuery!

Calculator

Result

323

+

65

20995

Calculator - Phase I Pseudocoding (20 Mins)

- For this first 20 minutes, your goals should be to:
    - Become familiar with the various elements of the HTML page as it is written now. (Identify the classes and IDs that matter!)
    - Create a general strategy for how you will accomplish the task:
        - How will you capture button clicks?
        - What will you do with the numbers clicked?
        - What will you do when the operator is clicked?
        - How will you differentiate numbers from operators?
        - How will you know the "value" of a number clicked?
        - How will you know when a user is done entering the first number?
    - Create a few test "on-click" events. These could just trigger Alert messages for now. If you get further along, then have these buttons alert their "value".

Calculator - Phase 2 Begin Logic (30 Mins)

Phase II Objectives

- For the next 30 minutes, your goals should be to:
    - Begin creating sets of variables that you think you will need. As a few suggestions: firstNumber, secondNumber, operator, result
    - Create code that captures the numbers from button clicks, and then displays them on the HTML.
    - Create code that captures an operator click, then immediately tells your code to begin recording the second number. (Example: If a user clicks the "plus" button, you know they are done typing the first number).
    - Create code that checks which operator the user has clicked, then saves this operator for later use in a conditional statement. (Example: If a user clicks the "plus" button, you know you will need to be adding two numbers. If a user clicks the "minus" button, you will need to be subtracting two numbers. Think: if-else statements!)

Lunch (35 mins)

YOU GET A LUNCH BREAK!
2004
AND YOU GET A LUNCH BREAK!
AND YOU GET A LUNCH BREAK!
EVERYONE GETS A LUNCH BREAK!

Calculator - Continue Logic Building (35 Mins)

Phase III Objectives

- For the next 40 minutes, your goals should be to:
    - Complete the functionality you are missing in your code.
    - Spend a few moments really assessing what you still have to do.
    - Get a TA/Instructor to help you set priorities if you are unsure as to how to proceed.

Calculator - Refinement (35 Mins)

Phase IV Objectives

- For the last 25 minutes, your goals should be to:
  - Complete any remaining functionality in your calculator
  - Handle bugs and edge cases (Example: What if a user tries to type in more numbers after getting the result? Will that mess up the screen?)
  - Create code to "restart" the calculator after a user clicks "clear".

Review Calculator

- We set a number of variables in the `initializeCalculator` function.

- We used JQuery to create separate on click listeners for numbers, operators, and equals.

- We used `this.value` in the callback function to determine what the value was for a given number or operator.

- We created code that would change the HTML content using `.html`

- We took numeric inputs for `firstNumber` **until** a user clicks an operator. Once a user clicks an operator, we changed the value of a boolean, called `isOperatorChosen`, and immediately began recording the `secondNumber`.

- Once a user clicks the "equal" button, we used conditionals to check which operation they had clicked, then ran that operation on the two numbers.

Questions