**Calculator :**

* General instructions
  1. Your calculator is going to contain functions for all of the **basic math operators you typically find on simple calculators, so start by creating functions for the following items and testing them in your browser’s console :** ✅
     + Add
     + Subtract
     + Multiply
     + Divide
  2. Create a new function **operate** that takes an operator and 2 numbers and then calls one of the above functions on the numbers. ✅
  3. Create a basic HTML calculator **with buttons for each digit, each of the above functions and an “Equals” key** ✅
     + Do not worry about wiring up the JS yet.
     + There should also be a display for the calculator, go ahead and fill it with some dummy numbers so you can get it looking right
     + Add a “clear” button.
  4. Create the functions that populate the display when you click the number buttons … you should be storing the ‘display value’ in a variable somewhere for use in the next step. ✅
  5. Make the calculator work ! You’ll need to store the first number that is input into the calculator when a **user presses an operator, and also have which operation has been chosen and then operate() on them when the user presses the “=” key.** ✅
     + **You should** already have the code that can populate the display, so once **operate() has been called, update the display with the ‘solution’ to the operation;**
     + This is the hardest part of the project. You need to figure out how to store all the values and call the operate function with them. Don’t feel bad if it takes you a while to figure out the logic.
  6. Gotchas : watch out for and fix these bugs if they show up in your code
     + Users should be able to **string together several operations and get the right answer,** with each pair of numbers being evaluated at a time. **For example, 12 + 7 – 5 \* 3 = should yield 42.** An example of the behavior we’re looking for would be **this student solution.**

+

(span div thisNum)

444 (span div thisNum)

100 + 4