

Lab 13:

Loops

By Julian Fee, Matilda Krulder, and Becky Phillips

Lab 13: Loops

Submit Assignment

Due Thursday by 8:50am Points 10 Submitting a file upload File Types pdf

Big Idea

Working with your partner, experiment with loops.

Task 1: Create an index.html for your lab

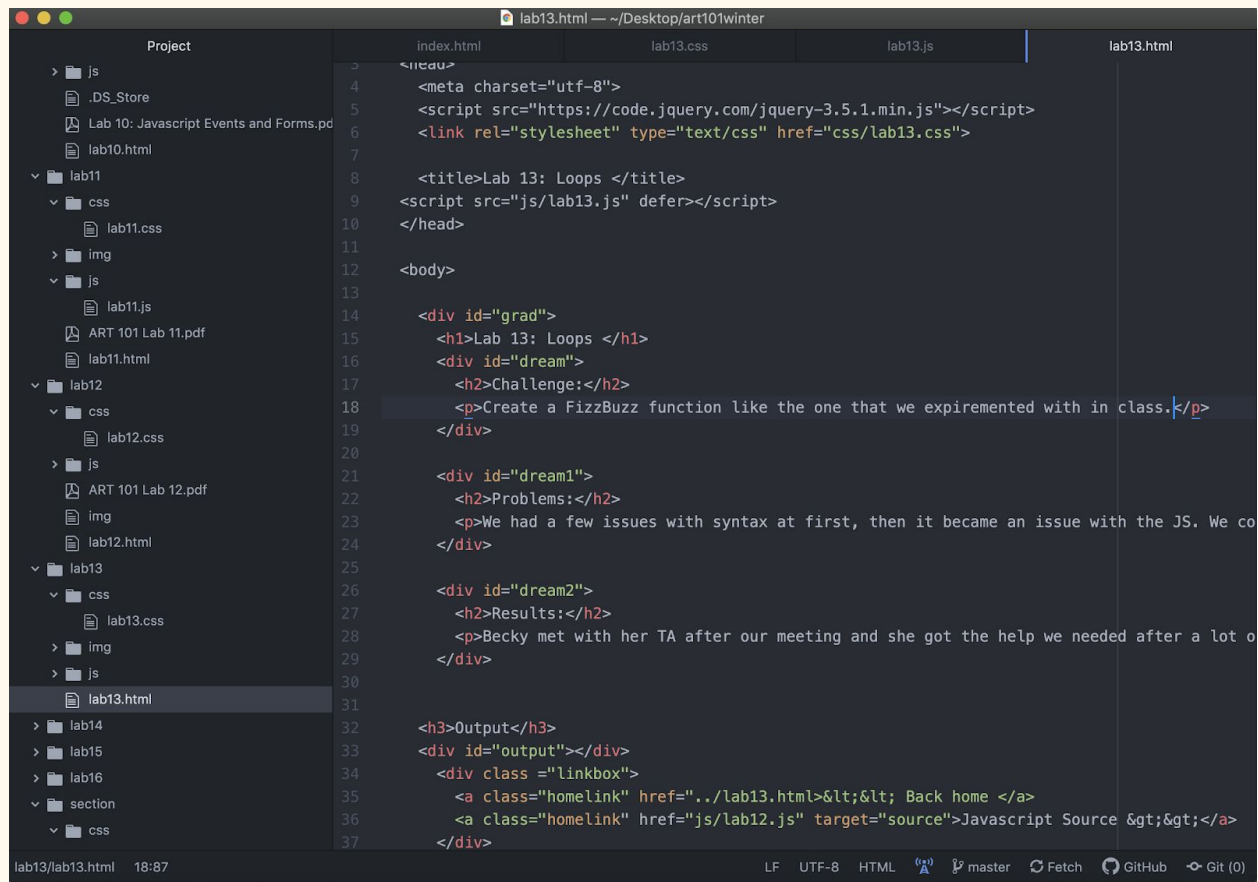
As always, create the proper folders and an index, css, and JavaScript files for this lab. And make sure you also put a link from your art101 homepage to this lab

1. Create an **index.html** with three organized sections: **Challenge, Problems, and Results**
2. Use heading, div, and paragraph tags to organize your page
3. As usual, link jQuery and a JavaScript file from the `<head>` section of your HTML like this:

```
<head>
  <title>Lab 13</title>
  <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
  <script src="js/lab.js" defer></script>
</head>
```

4. Add a CSS stylesheet in your css folder and link from **index.html**
5. Create an output div `<div id="output">`
6. Add a link to your JavaScript file in your index page.

Task 1: Create an index.html for Lab 13



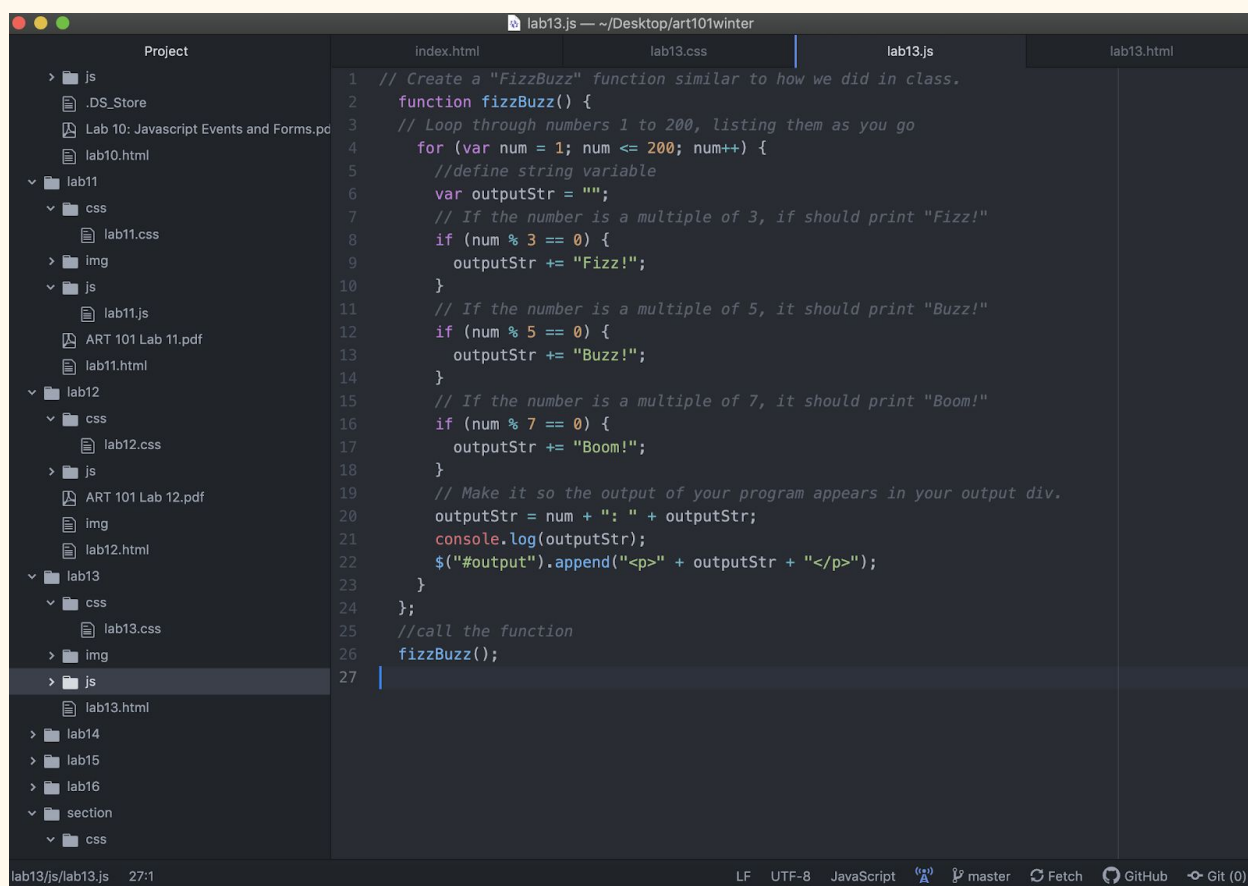
```

1 <!doctype>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
6     <link rel="stylesheet" type="text/css" href="css/lab13.css">
7
8     <title>Lab 13: Loops </title>
9     <script src="js/lab13.js" defer></script>
10  </head>
11
12  <body>
13
14    <div id="grad">
15      <h1>Lab 13: Loops </h1>
16      <div id="dream">
17        <h2>Challenge:</h2>
18        <p>Create a FizzBuzz function like the one that we expiremented with in class.</p>
19      </div>
20
21      <div id="dream1">
22        <h2>Problems:</h2>
23        <p>We had a few issues with syntax at first, then it became an issue with the JS. We co
24      </div>
25
26      <div id="dream2">
27        <h2>Results:</h2>
28        <p>Becky met with her TA after our meeting and she got the help we needed after a lot o
29      </div>
30
31
32    <h3>Output</h3>
33    <div id="output"></div>
34    <div class="linkbox">
35      <a class="homelink" href="../lab13.html">&lt;&lt; Back home </a>
36      <a class="homelink" href="js/lab12.js" target="source">Javascript Source &gt;&gt;</a>
37    </div>

```

(Our HTML in Atom.)

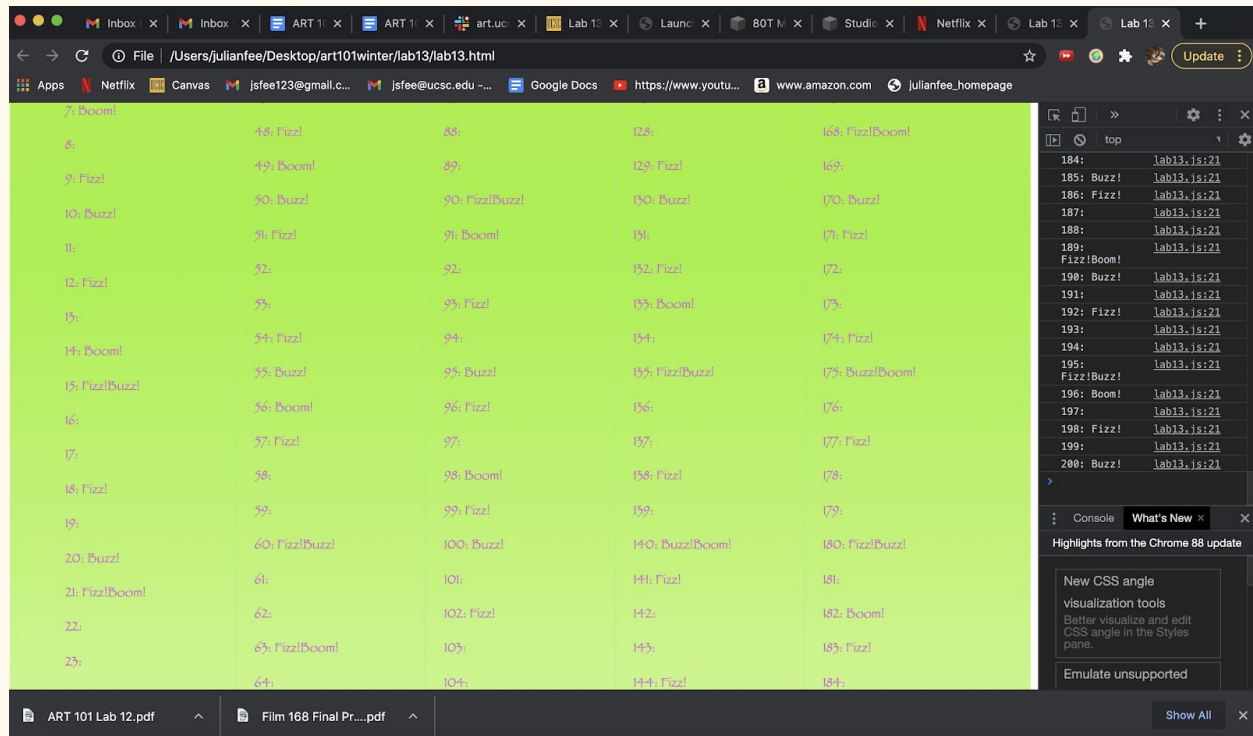
Task 2: Create a JavaScript file



```
1 // Create a "FizzBuzz" function similar to how we did in class.
2 function fizzBuzz() {
3   // Loop through numbers 1 to 200, listing them as you go
4   for (var num = 1; num <= 200; num++) {
5     //define string variable
6     var outputStr = "";
7     // If the number is a multiple of 3, it should print "Fizz!"
8     if (num % 3 == 0) {
9       outputStr += "Fizz!";
10    }
11    // If the number is a multiple of 5, it should print "Buzz!"
12    if (num % 5 == 0) {
13      outputStr += "Buzz!";
14    }
15    // If the number is a multiple of 7, it should print "Boom!"
16    if (num % 7 == 0) {
17      outputStr += "Boom!";
18    }
19    // Make it so the output of your program appears in your output div.
20    outputStr = num + ": " + outputStr;
21    console.log(outputStr);
22    $("#output").append("<p>" + outputStr + "</p>");
23  }
24 };
25 //call the function
26 fizzBuzz();
27
```

(Our JavaScript in Atom.)

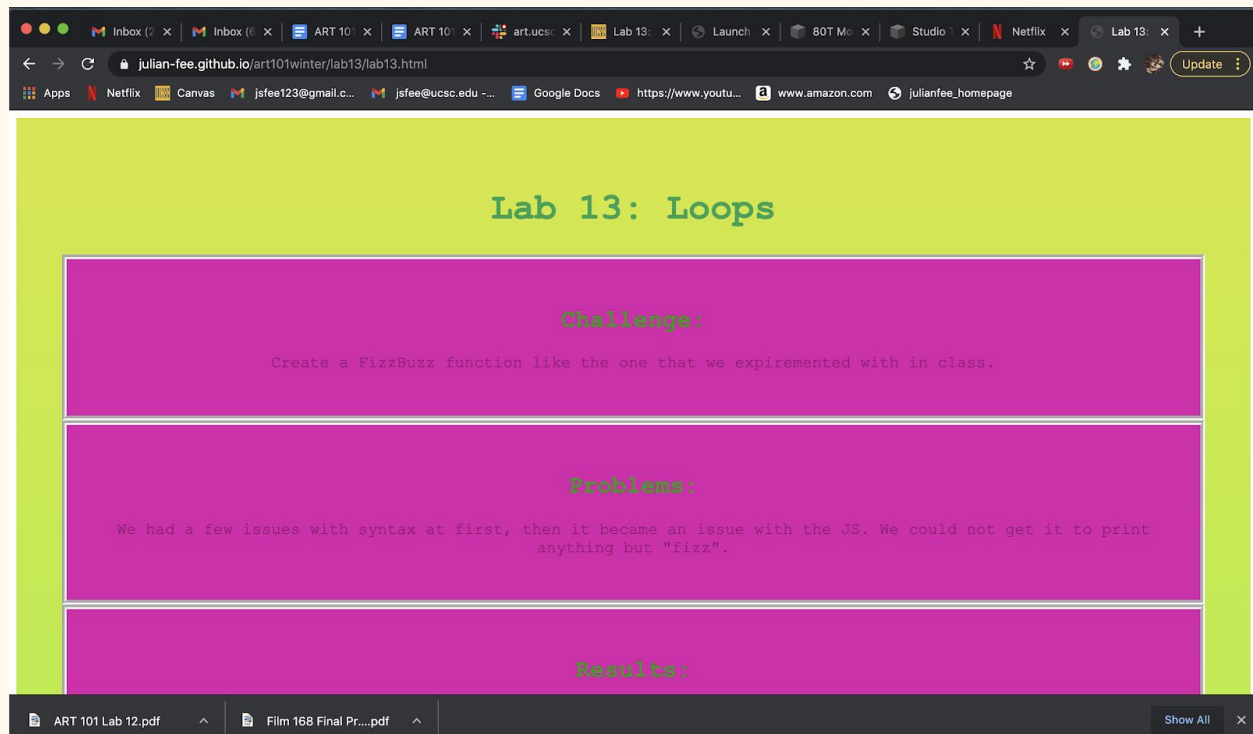
Task 3: Test, Debug, and Upload



(Our local file opened on a webpage.)

Link to published Lab 13 webpage:

<https://julian-fee.github.io/art101winter/lab13/lab13.html>



(Published Lab 13 webpage.)

Summary of Efforts:

We met to discuss the Lab on Wednesday morning. At first, we had a lot of trouble with the lab, and had a lot of difficulty getting the output to actually show up on the webpage, not just in the console. After a lot of messing around, and getting a bit of TA help from section, we were able to figure out that we had to make all the conditionals “if” statements in order for the number to have more than one word attached to it (ex. 70 would print Buzz!Boom! rather than just Buzz!). We also initially had an else statement for the numbers that didn’t fall into any category, but then got rid of it because it was messing up the rest of the function. Although we did have some problems with this lab, it was ultimately satisfying to see it when it finally worked. As a bonus, we figured out how to style the CSS of the output so that it would print in neat columns.

Bonus: Columns

Results				
Our lab is below! Nicely seperated out into columns too...				
1:	41:	81: Fizz!	121:	161: Boom!
2:	42: Fizz!Boom!	82:	122:	162: Fizz!
3: Fizz!	43:	83:	123: Fizz!	163:
4:	44:	84: Fizz!Boom!	124:	164:
5: Buzz!	45: Fizz!Buzz!	85: Buzz!	125: Buzz!	165: Fizz!Buzz!
6: Fizz!	46:	86:	126: Fizz!Boom!	166:
7: Boom!	47:	87: Fizz!	127:	167:
8:	48: Fizz!	88:	128:	168: Fizz!Boom!
9: Fizz!	49: Boom!	89:	129: Fizz!	169:
10: Buzz!	50: Buzz!	90: Fizz!Buzz!	130: Buzz!	170: Buzz!
11:	51: Fizz!	91: Boom!	131:	171: Fizz!
12: Fizz!	52:	92:	132: Fizz!	172:
13:	53:	93: Fizz!	133: Boom!	173:
14: Boom!	54: Fizz!	94:	134:	174: Fizz!
15: Fizz!Buzz!	55: Buzz!	95: Buzz!	135: Fizz!Buzz!	175: Buzz!Boom!
16:	56: Boom!	96: Fizz!	136:	176:
17:	57: Fizz!	97:	137:	177: Fizz!
18: Fizz!	58:	98: Boom!	138: Fizz!	178:

(Our output printed in 5 separate columns.)