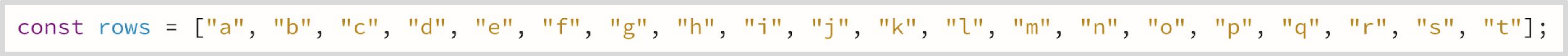
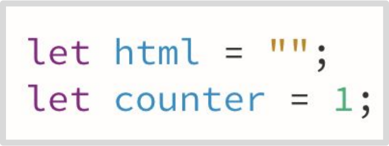
# Seat Reservation Program

Part 1 - Creating the Seats

# Variables Needed



```
const rows = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t"];
```

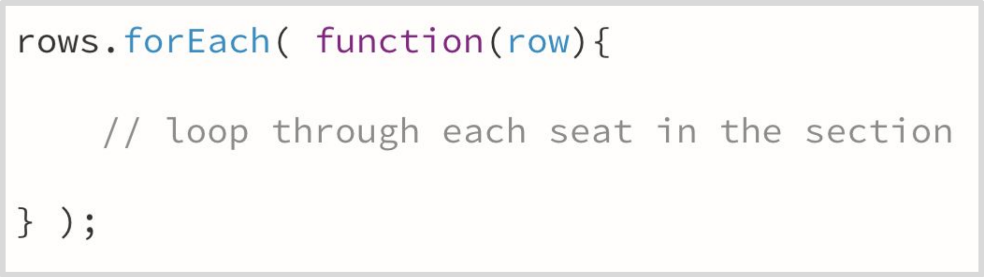


```
let html = "";
let counter = 1;
```

- First, an array for the rows.
- Then, a variable that will hold the HTML that is generated to add into each section.
- Then a counter that will hold each seat number.
- Use a loop inside a loop.

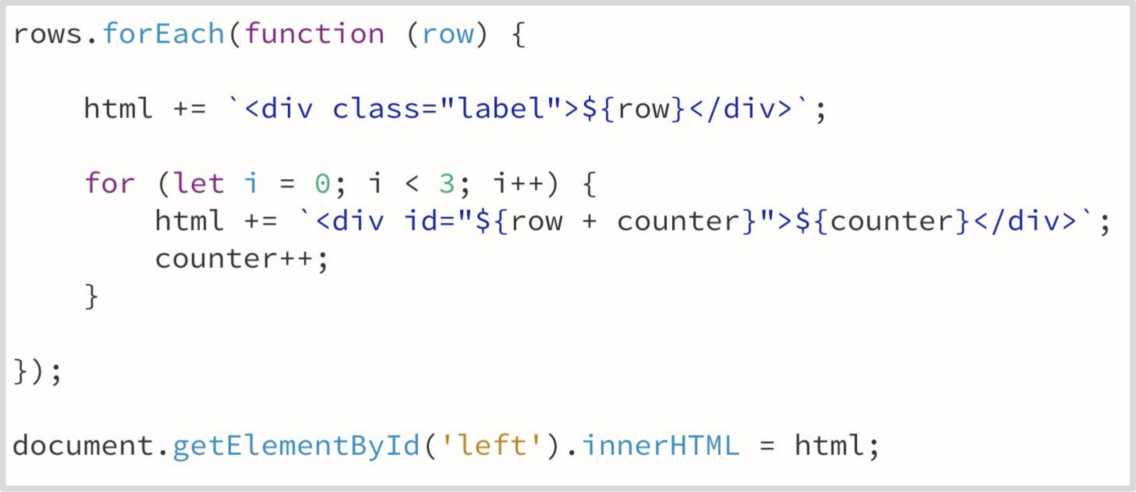
# Using forEach()

The forEach() method is an array method, meaning you run it on an array, such as “rows”. Then you can pass in a function, and that function takes an argument that represents each element in the array.



```
rows.forEach( function(row){

    // loop through each seat in the section

} );
```

# Working on the Left Section

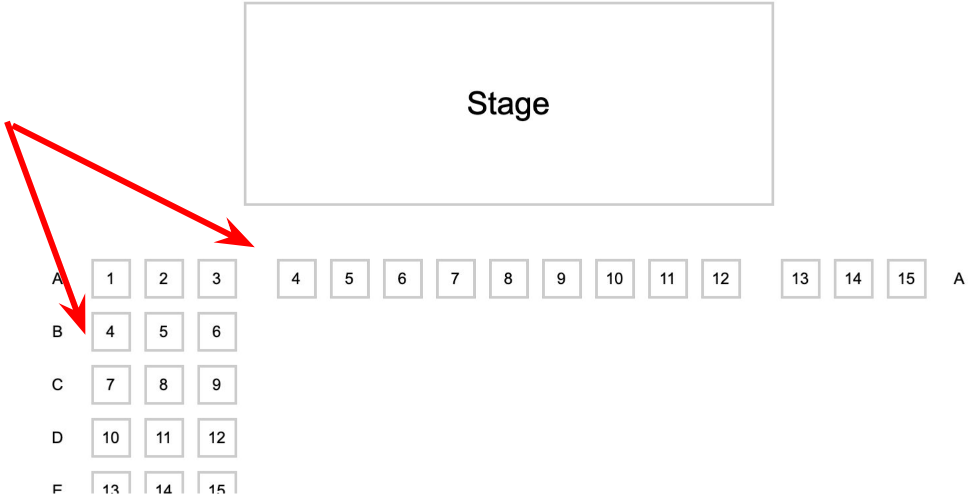


```
rows.forEach(function (row) {

    html += `<div class="label">${row}</div>`;

    for (let i = 0; i < 3; i++) {
        html += `<div id="${row + counter}">${counter}</div>`;
        counter++;
    }

});

document.getElementById('left').innerHTML = html;
```

To start, work on the left side. You need to add a label to the left side with the row letter, then loop through the seats. If you test this, you will see that there is a problem.

# Identifying the Problem

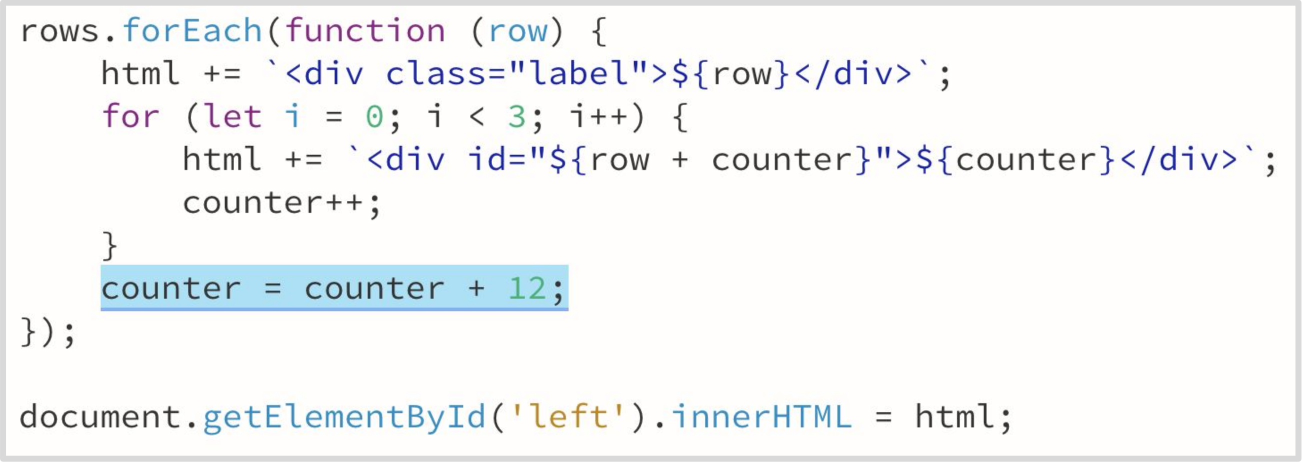
This is generating the seats for sure, but the numbering is wrong. Row “B” is starting with seat 4, but you really want it to start with seat 16.



Stage

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | A |
| B | 4 | 5 | 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| C | 7 | 8 | 9 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| D | 10 | 11 | 12 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| E | 13 | 14 | 15 |  |  |  |  |  |  |  |  |  |  |  |  |  |

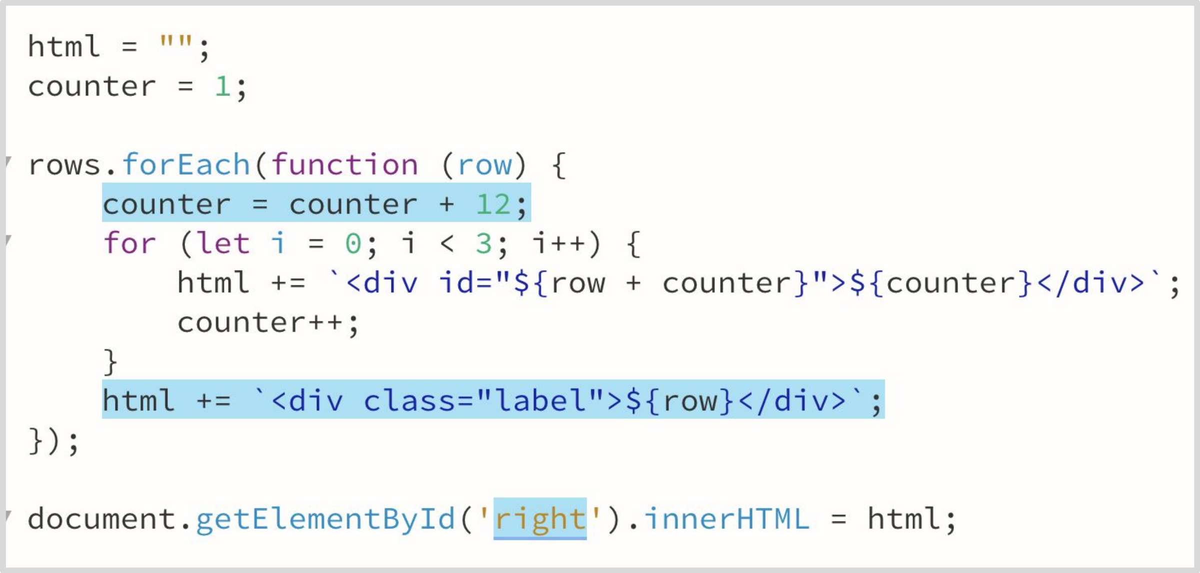
# Problem Fixed!



```
rows.forEach(function (row) {
    html += `<div class="label">${row}</div>`;
    for (let i = 0; i < 3; i++) {
        html += `<div id="${row + counter}">${counter}</div>`;
        counter++;
    }
    counter = counter + 12;
});

document.getElementById('left').innerHTML = html;
```

Here is the left section, with all the seats generated and with the correct numbers.

# The Right Side

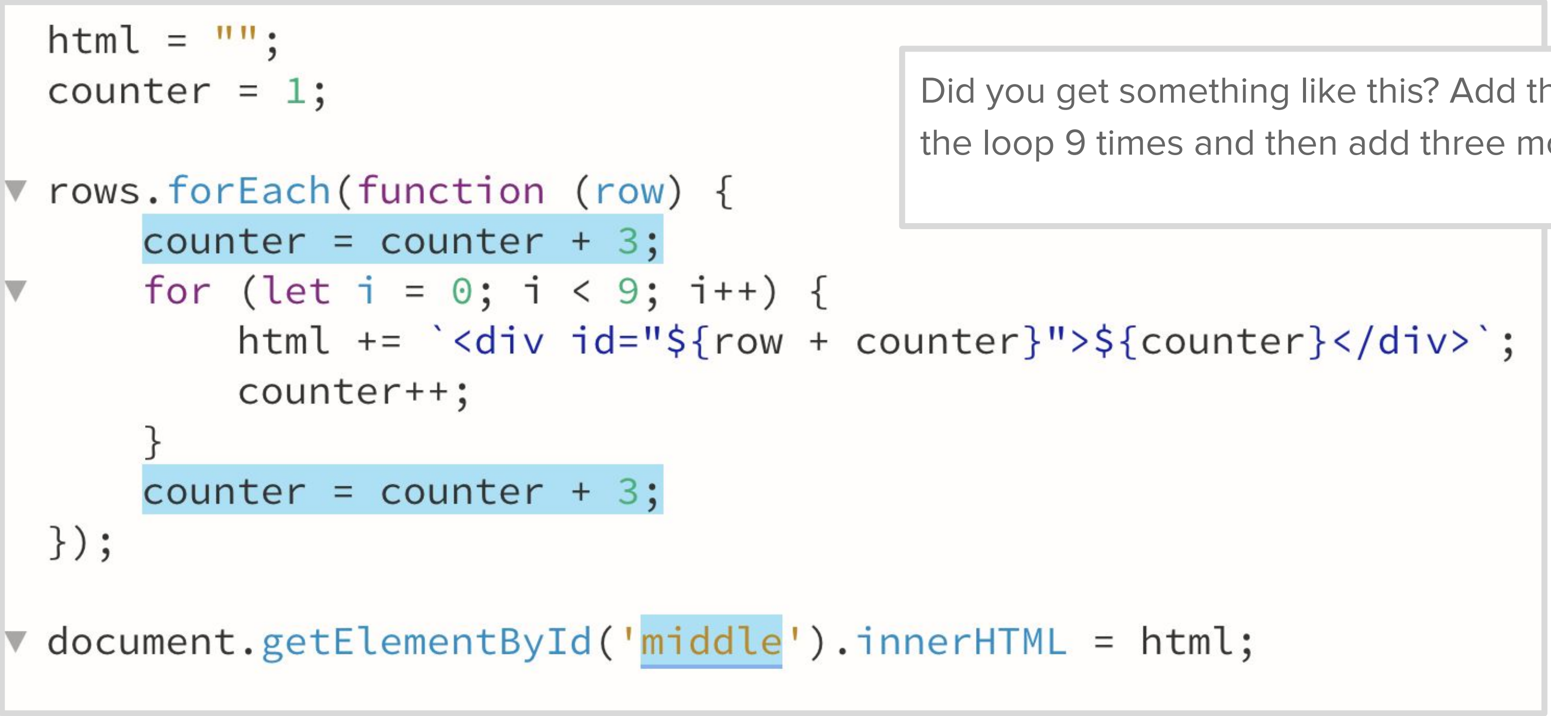


```
html = "";
counter = 1;

rows.forEach(function (row) {
    counter = counter + 12;
    for (let i = 0; i < 3; i++) {
        html += `<div id="${row + counter}">${counter}</div>`;
        counter++;
    }
    html += `<div class="label">${row}</div>`;
});

document.getElementById('right').innerHTML = html;
```

Here is the code for the right side. The highlighted lines have swapped places.

Can you do the middle section?

# Middle Section

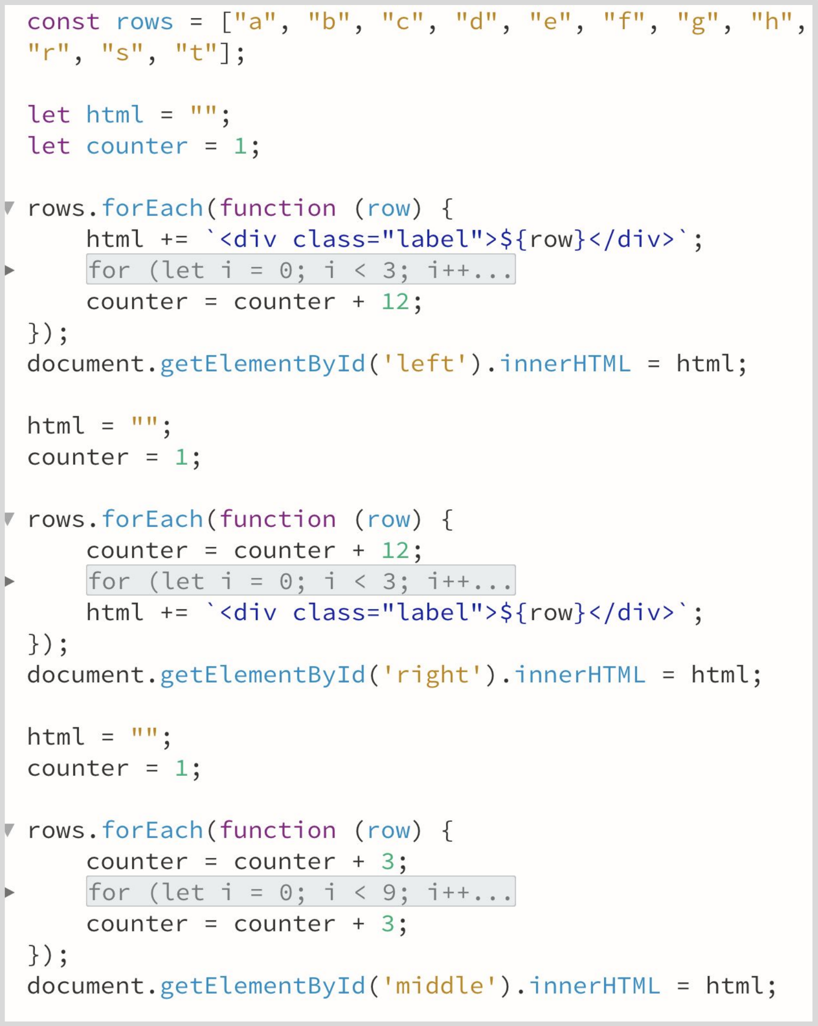


```
html = "";
counter = 1;

rows.forEach(function (row) {
    counter = counter + 3;
    for (let i = 0; i < 9; i++) {
        html += `<div id="${row + counter}">${counter}</div>`;
        counter++;
    }
    counter = counter + 3;
});

document.getElementById('middle').innerHTML = html;
```

Did you get something like this? Add three, do the loop 9 times and then add three more?

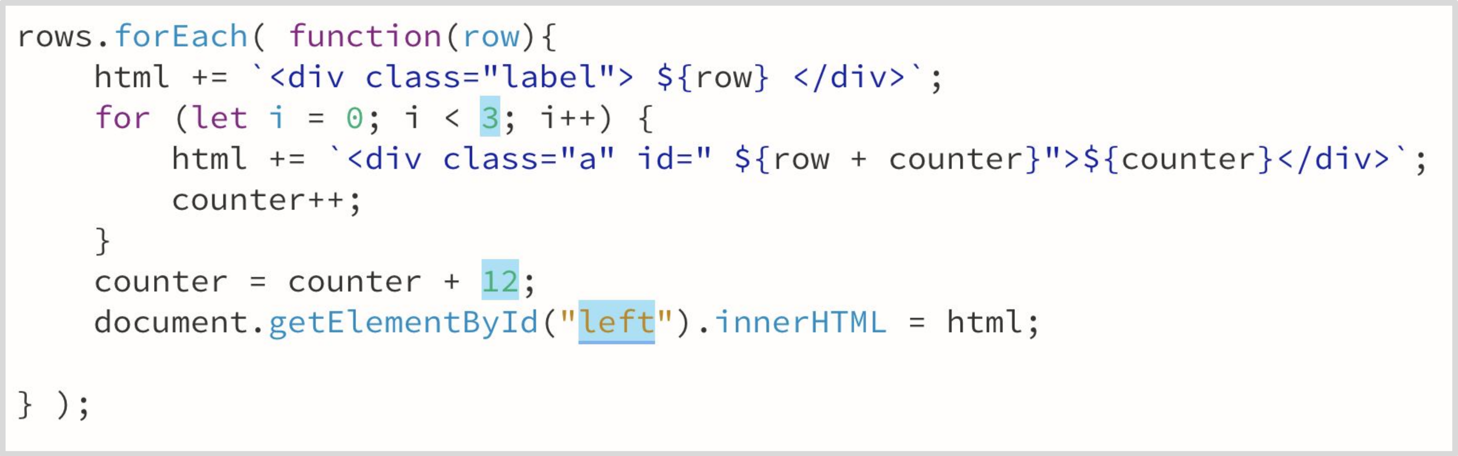
# So Far...

This is what you should have so far. It works, and that is fine, but it could be better.



```
const rows = ["a", "b", "c", "d", "e", "f", "g", "h",
"r", "s", "t"];

let html = "";
let counter = 1;

rows.forEach(function (row) {
    html += `<div class="label">${row}</div>`;
    for (let i = 0; i < 3; i++...
    counter = counter + 12;
});
document.getElementById('left').innerHTML = html;

html = "";
counter = 1;

rows.forEach(function (row) {
    counter = counter + 12;
    for (let i = 0; i < 3; i++...
    html += `<div class="label">${row}</div>`;
});
document.getElementById('right').innerHTML = html;

html = "";
counter = 1;

rows.forEach(function (row) {
    counter = counter + 3;
    for (let i = 0; i < 9; i++...
    counter = counter + 3;
});
document.getElementById('middle').innerHTML = html;
```

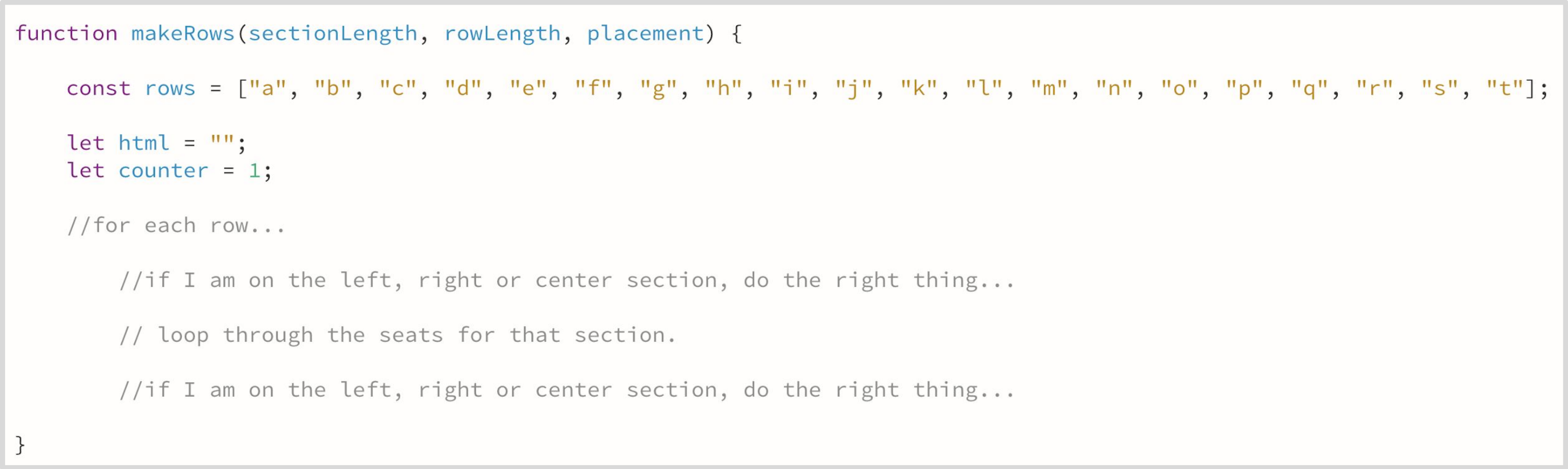
# Figuring out Variables



```
rows.forEach( function(row){
    html += `<div class="label"> ${row} </div>`;
    for (let i = 0; i < 3; i++) {
        html += `<div class="a" id=" ${row + counter}">${counter}</div>`;
        counter++;
    }
    counter = counter + 12;
    document.getElementById("left").innerHTML = html;

} );
```

If you look at the hard coded numbers in the code you already wrote, you can figure out what those numbers represent.

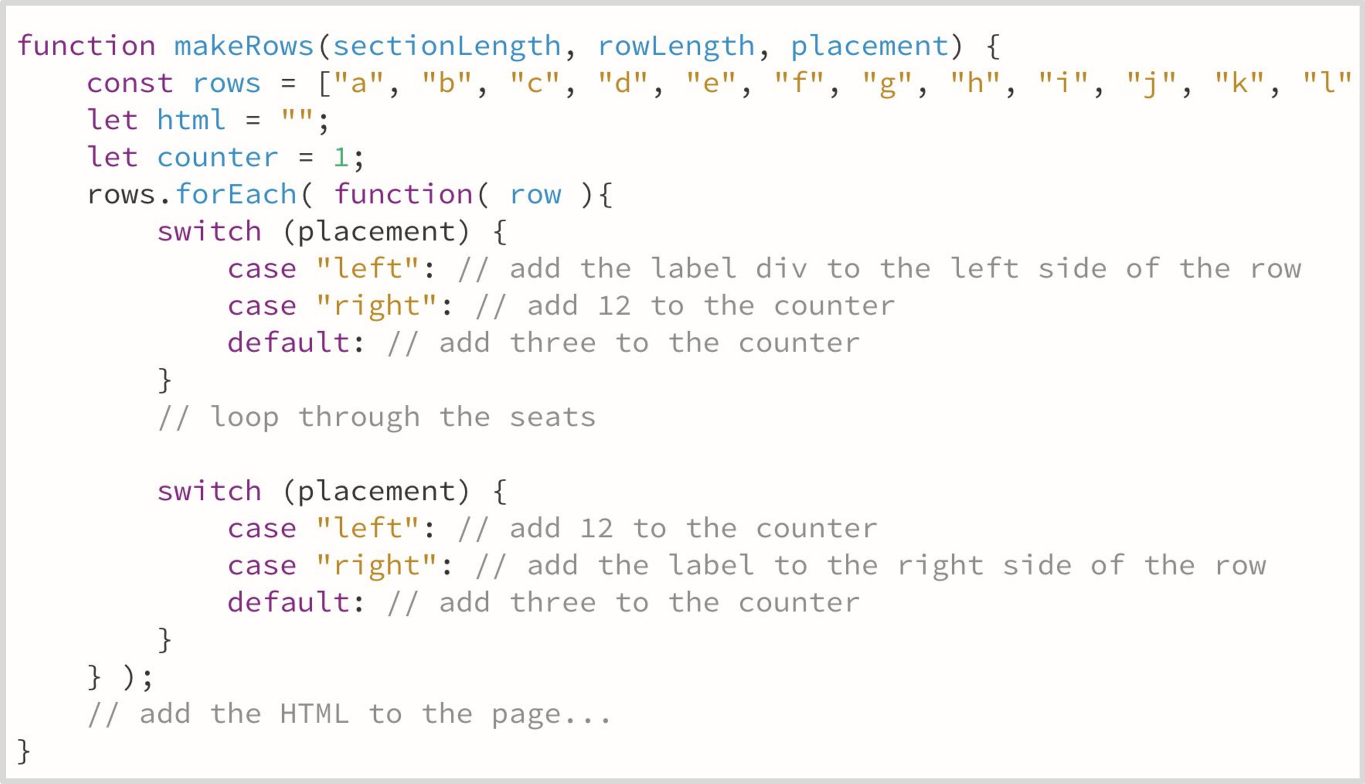
# Where We Are Going...



```
function makeRows(sectionLength, rowLength, placement) {

    const rows = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t"];

    let html = "";
    let counter = 1;

    //for each row...

        //if I am on the left, right or center section, do the right thing...

        // loop through the seats for that section.

        //if I am on the left, right or center section, do the right thing...

}
```

This is where you are going with this. Add this function, move the variables inside.

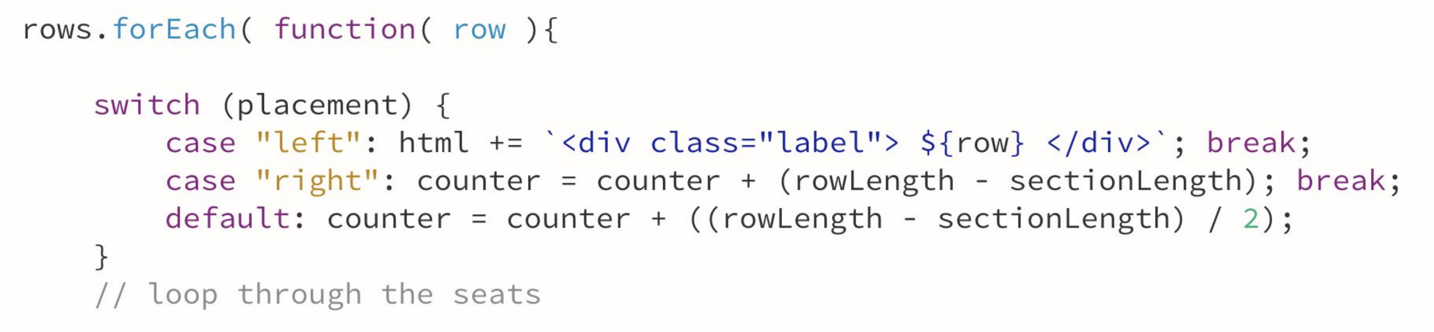
# Adding Switch Statements



```
function makeRows(sectionLength, rowLength, placement) {
    const rows = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l"
    let html = "";
    let counter = 1;
    rows.forEach( function( row ){
        switch (placement) {
            case "left": // add the label div to the left side of the row
            case "right": // add 12 to the counter
            default: // add three to the counter
        }
        // loop through the seats

        switch (placement) {
            case "left": // add 12 to the counter
            case "right": // add the label to the right side of the row
            default: // add three to the counter
        }
    } );
    // add the HTML to the page...
}
```

Making progress on this function. You could use if / else if statements, but the switch statement is a little more compact in this case.

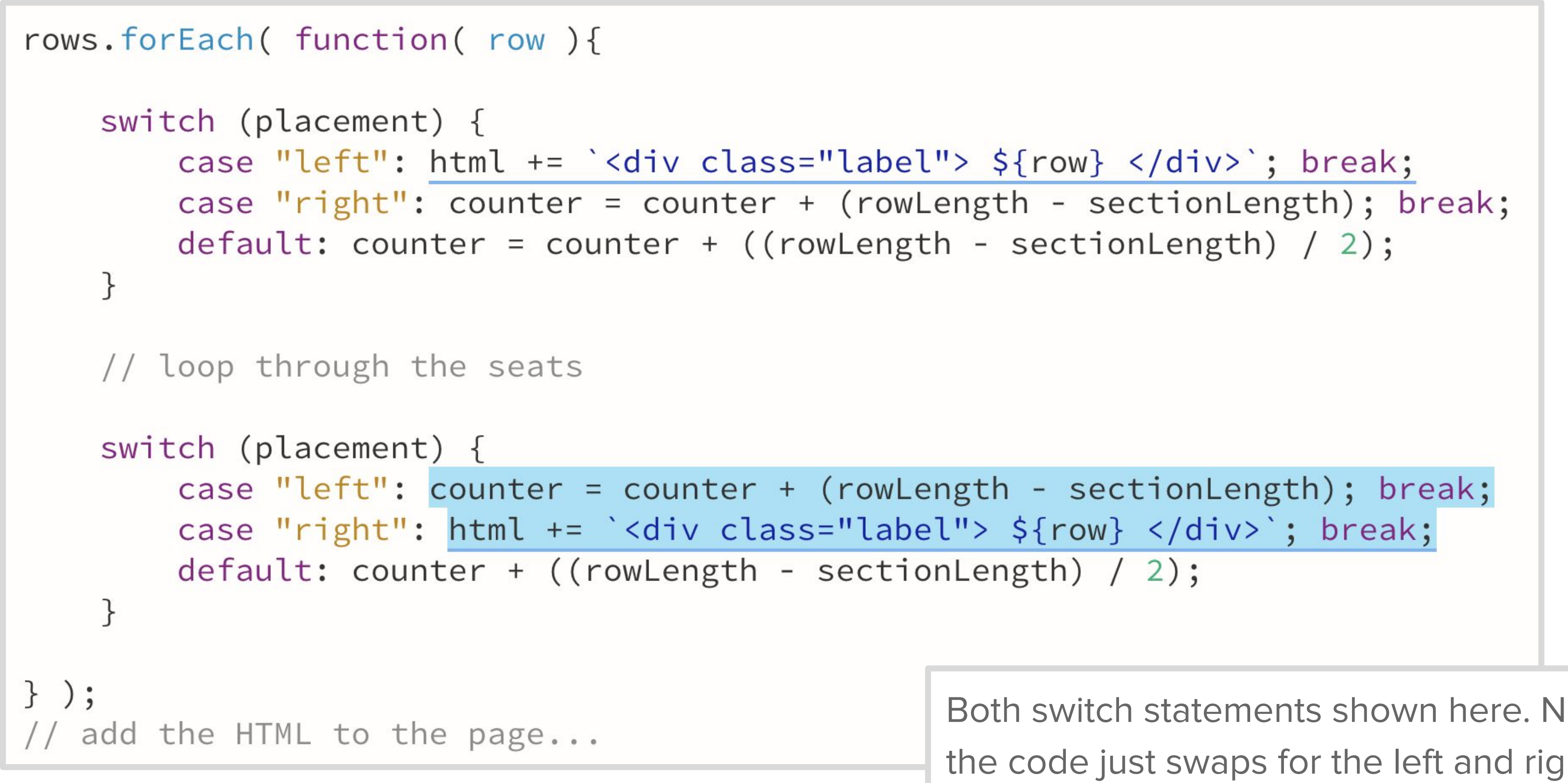
# First Switch Statement



```
rows.forEach( function( row ){

    switch (placement) {
        case "left": html += `<div class="label"> ${row} </div>`; break;
        case "right": counter = counter + (rowLength - sectionLength); break;
        default: counter = counter + ((rowLength - sectionLength) / 2);
    }
    // loop through the seats
```

Here is the first switch statement finished. The variable “placement” is going to be left, right or middle. If it’s set to “left”, just add the label div.

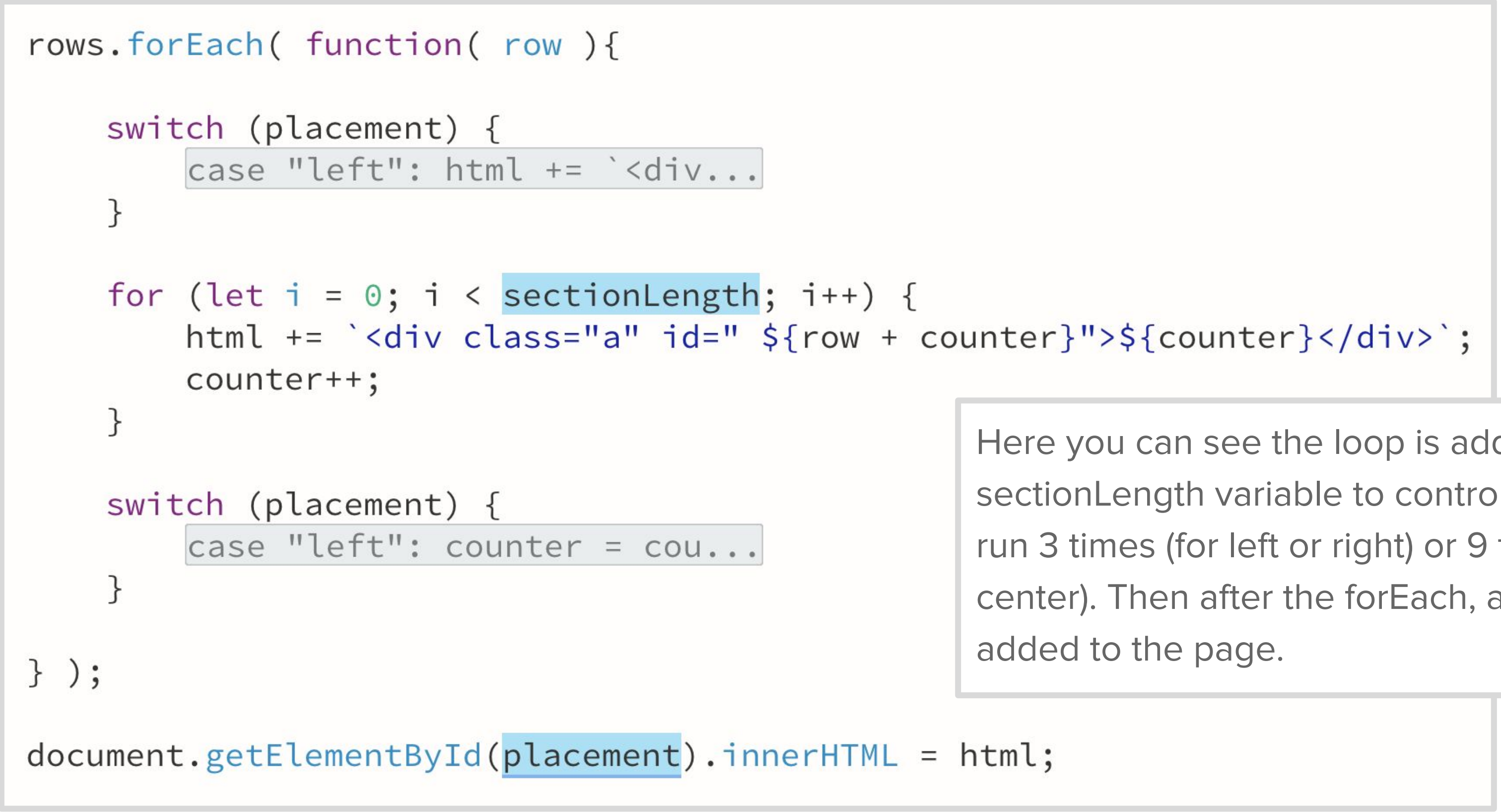
# Both Switch Statements



```
rows.forEach( function( row ){

    switch (placement) {
        case "left": html += `<div class="label"> ${row} </div>`; break;
        case "right": counter = counter + (rowLength - sectionLength); break;
        default: counter = counter + ((rowLength - sectionLength) / 2);
    }

    // loop through the seats

    switch (placement) {
        case "left": counter = counter + (rowLength - sectionLength); break;
        case "right": html += `<div class="label"> ${row} </div>`; break;
        default: counter + ((rowLength - sectionLength) / 2);
    }

} );
// add the HTML to the page...
```

Both switch statements shown here. Notice that the code just swaps for the left and right options.

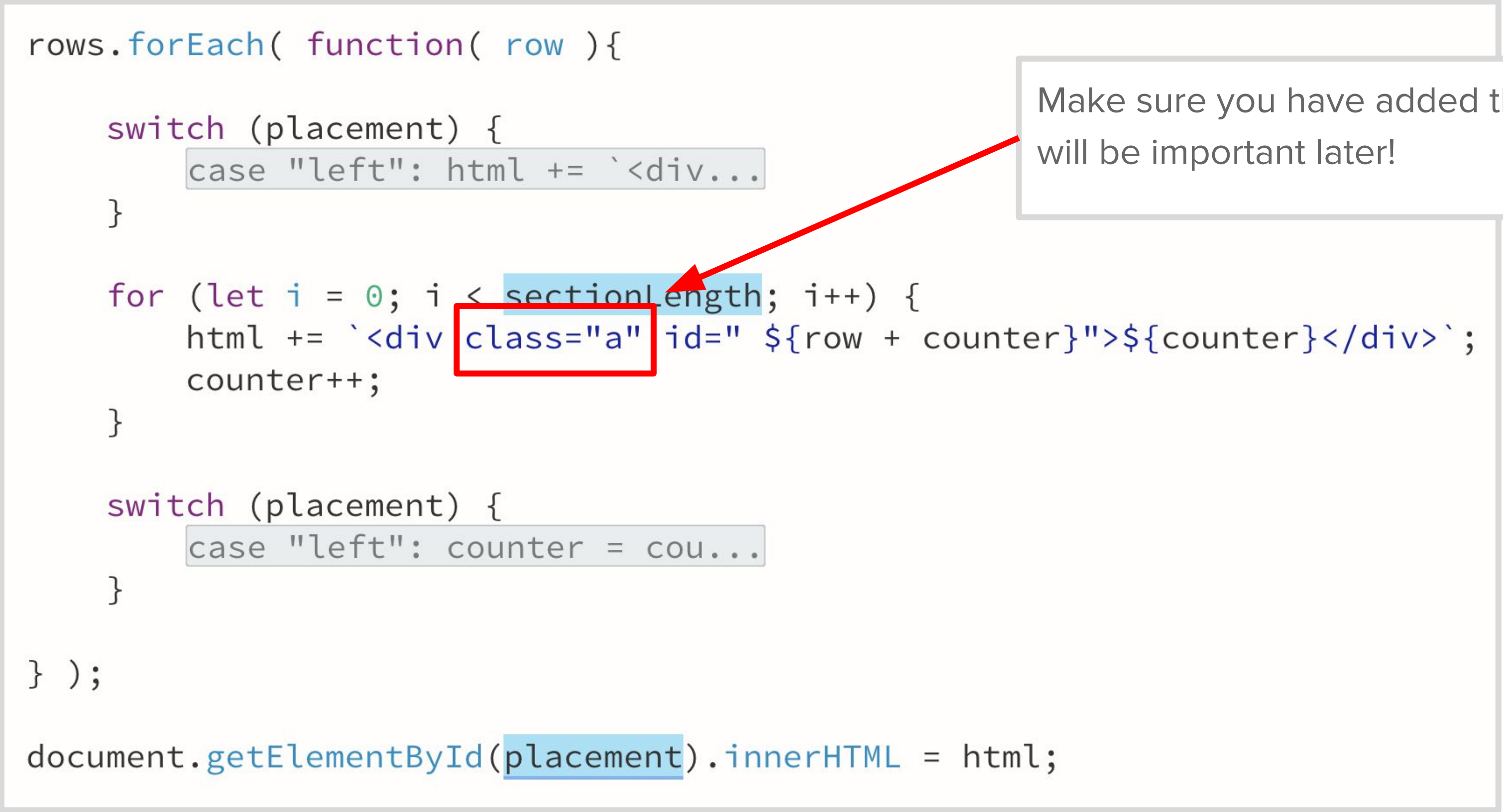
# Adding the Loop and the HTML



```
rows.forEach( function( row ){

    switch (placement) {
        case "left": html += `<div...
    }

    for (let i = 0; i < sectionLength; i++) {
        html += `<div class="a" id=" ${row + counter}">${counter}</div>`;
        counter++;
    }

    switch (placement) {
        case "left": counter = cou...
    }

} );

document.getElementById(placement).innerHTML = html;
```

Here you can see the loop is added, using the sectionLength variable to control if it is going to run 3 times (for left or right) or 9 times (for center). Then after the forEach, all the HTML is added to the page.

# Make Sure the Class is Set!

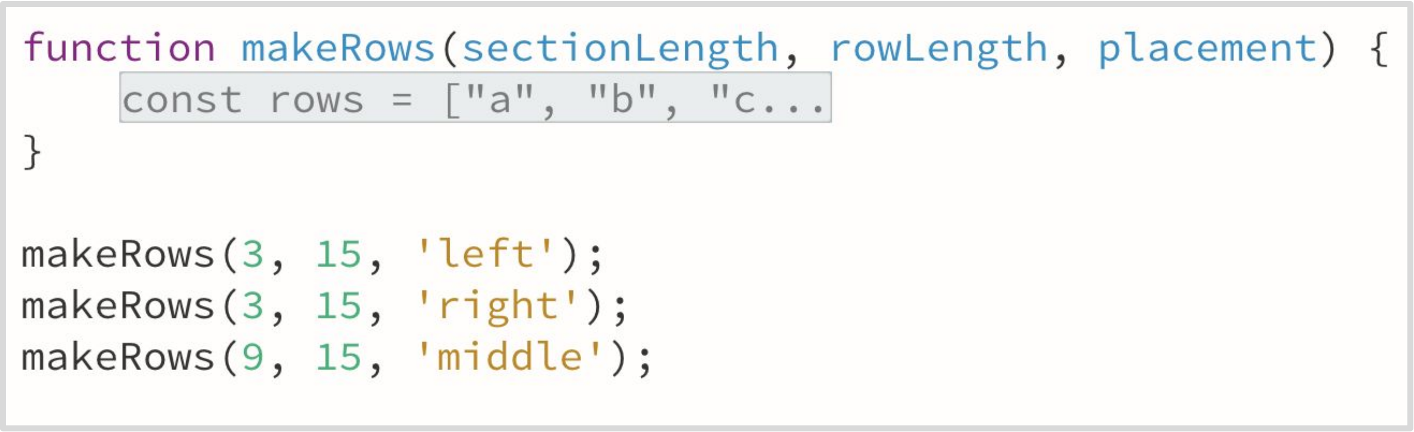


```
rows.forEach( function( row ){

    switch (placement) {
        case "left": html += `<div...
    }

    for (let i = 0; i < sectionLength; i++) {
        html += `<div class="a" id=" ${row + counter}">${counter}</div>`;
        counter++;
    }

    switch (placement) {
        case "left": counter = cou...
    }

} );

document.getElementById(placement).innerHTML = html;
```

Make sure you have added this class. It will be important later!

# Run the Function for Each Section

Run this function three times, with parameters set for each section and you will get all 300 seats showing up on your page.

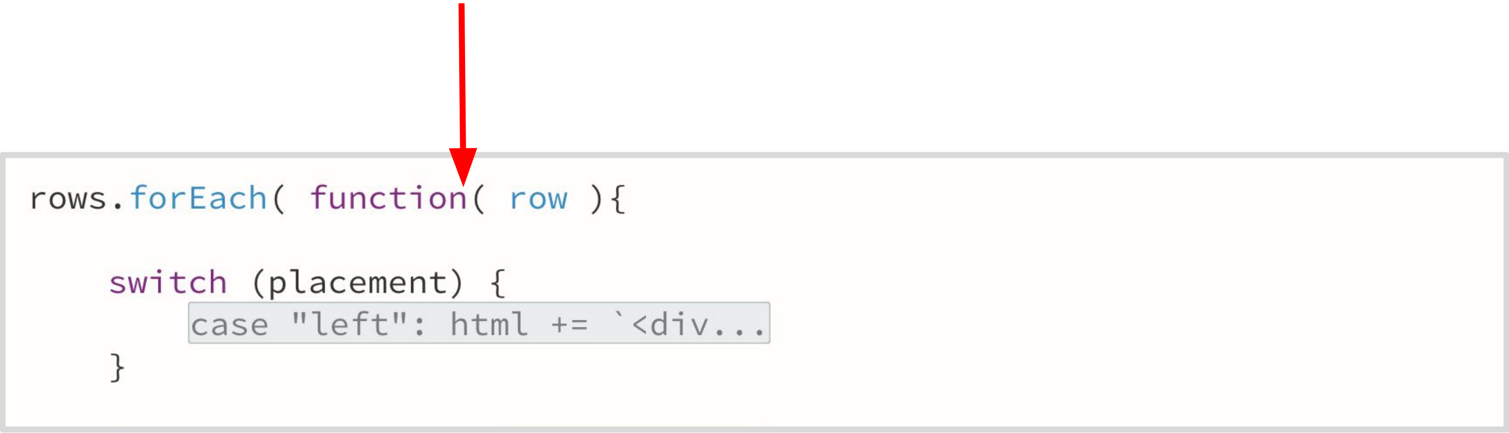


```
function makeRows(sectionLength, rowLength, placement) {
    const rows = ["a", "b", "c...
}

makeRows(3, 15, 'left');
makeRows(3, 15, 'right');
makeRows(9, 15, 'middle');
```

# Using an Arrow Function

The callback function in the forEach method is a good  candidate for an arrow function.

See if you can swap the syntax so that this function is an arrow function expression.



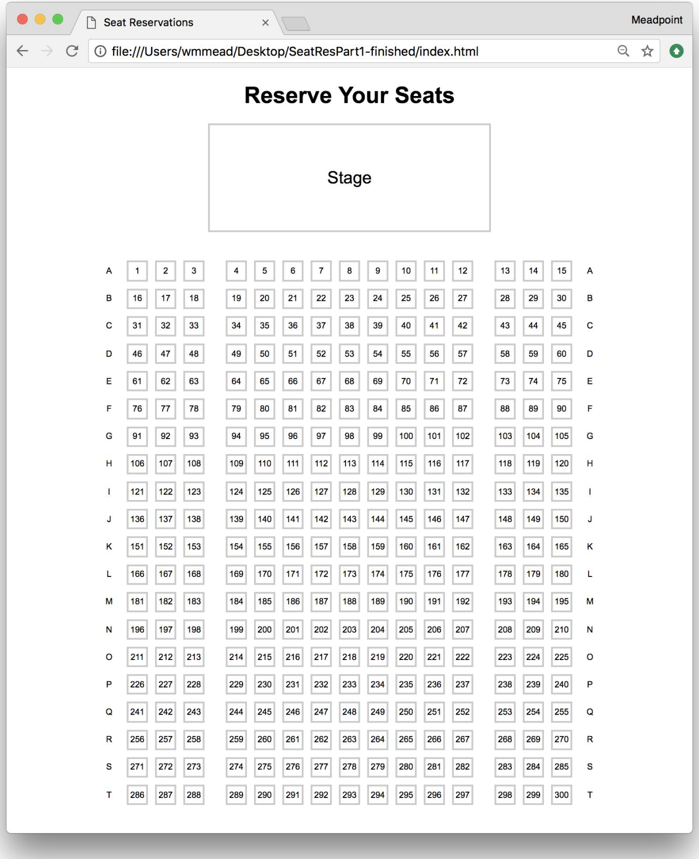
```
rows.forEach( function( row ){

    switch (placement) {
        case "left": html += `<div...
    }
```

# The Whole Function

```
function makeRows(sectionLength, rowLength, placement) {
  const rows = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t"];
  let html = ""; let counter = 1;
    rows.forEach(row => {
        switch (placement) {
            case "left": html += `<div class="label">${row}</div>`; break;
            case "right": counter = counter + (rowLength - sectionLength); break;
            default: counter = counter + ((rowLength - sectionLength) / 2);
        } for (let i = 0; i < sectionLength; i++) {
            html += `<div class="a" id="${row + counter}">${counter}</div>`;
            counter++;
        } switch (placement) {
            case "left": counter = counter + (rowLength - sectionLength); break;
            case "right": html += `<div class="label">${row}</div>`; break;
            default: counter = counter + ((rowLength - sectionLength) / 2);
        }
    });
    document.getElementById(placement).innerHTML = html;
}
```

# End of Part 1

That is how I made all the rows. You may have come up with a completely different strategy.


Seat Reservations
Meadpoint
file:///Users/wmmead/Desktop/SeatResPart1-finished/index.html

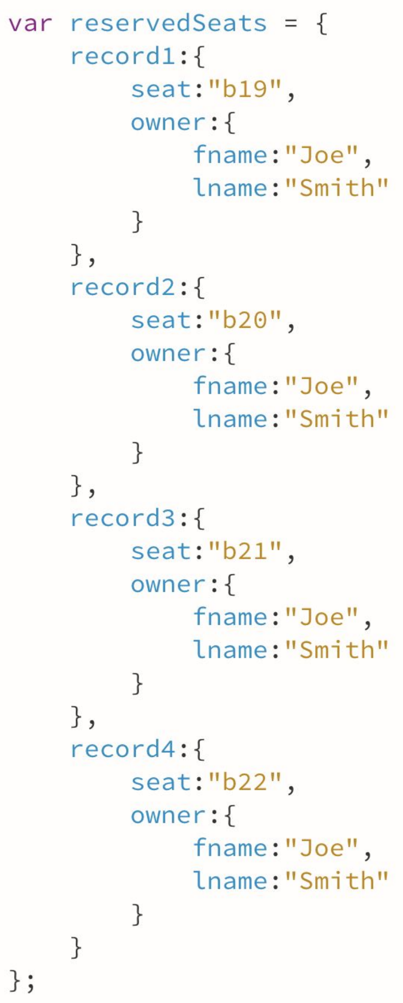
## Reserve Your Seats

Stage

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | A |
| B | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | B |
| C | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | C |
| D | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | D |
| E | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | E |
| F | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | F |
| G | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | G |
| H | 106 | 107 | 108 | 109 | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | H |
| I | 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 | I |
| J | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 | 144 | 145 | 146 | 147 | 148 | 149 | 150 | J |
| K | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 | 160 | 161 | 162 | 163 | 164 | 165 | K |
| L | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 | 176 | 177 | 178 | 179 | 180 | L |
| M | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 | 192 | 193 | 194 | 195 | M |
| N | 196 | 197 | 198 | 199 | 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 | 208 | 209 | 210 | N |
| O | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 | 219 | 220 | 221 | 222 | 223 | 224 | 225 | O |
| P | 226 | 227 | 228 | 229 | 230 | 231 | 232 | 233 | 234 | 235 | 236 | 237 | 238 | 239 | 240 | P |
| Q | 241 | 242 | 243 | 244 | 245 | 246 | 247 | 248 | 249 | 250 | 251 | 252 | 253 | 254 | 255 | Q |
| R | 256 | 257 | 258 | 259 | 260 | 261 | 262 | 263 | 264 | 265 | 266 | 267 | 268 | 269 | 270 | R |
| S | 271 | 272 | 273 | 274 | 275 | 276 | 277 | 278 | 279 | 280 | 281 | 282 | 283 | 284 | 285 | S |
| T | 286 | 287 | 288 | 289 | 290 | 291 | 292 | 293 | 294 | 295 | 296 | 297 | 298 | 299 | 300 | T |

# The reservedSeats Object

At the top of the script file in the start folder add the reservedSeats object found in the snippets file that goes with this lesson.



```
var reservedSeats = {
    record1:{
        seat:"b19",
        owner:{
            fname:"Joe",
            lname:"Smith"
        }
    },
    record2:{
        seat:"b20",
        owner:{
            fname:"Joe",
            lname:"Smith"
        }
    },
    record3:{
        seat:"b21",
        owner:{
            fname:"Joe",
            lname:"Smith"
        }
    },
    record4:{
        seat:"b22",
        owner:{
            fname:"Joe",
            lname:"Smith"
        }
    }
};
```

# The Next Challenge...

Can you write a function that does the following:

- Gets data from the reservedSeats object
- Updates the DOM by setting the class on each matching element from "a" (for available) to "r" (for reserved).
- Replaces the contents of each matching element with an "R" to show that seat has been reserved.