Day 4 – More JS Basics

Show students how to do an alert window.



Show students how to do a prompt



Explain that if they do set that to a variable, it will not be saved.



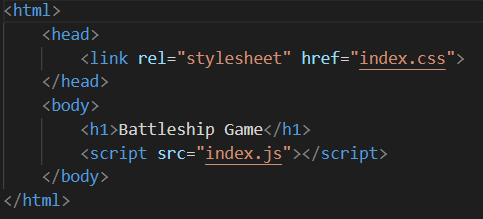
**Battleship!**

So, this is not like normal battleship, it is just something simple for the students. It is a little advanced and goes over some things we did not cover but that is ok, just explain as you go.

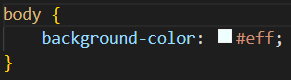
The rules of this battleship game are:

* There will be one ship
* The ship will occupy 4 parts of the grid
* The player will input the x and y coordinants at prompt
* The grid is 9x9.
* The play will put h or v for what way they want the ship to be set
* We will put . for the grid
* We will use the fire button to start the game.

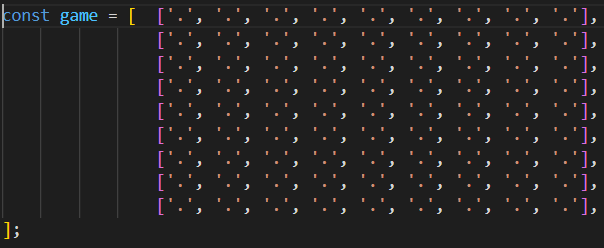
1. Create index.html, index.css, and index.js files in vcode.
2. Link the css and js files to the html file.
3. Add an <h1> element in the body of the html file.



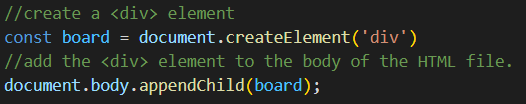
1. Add css for the body that changes the background color to #eff. (they can choose whatever color they want as long as it is a lighter color.)



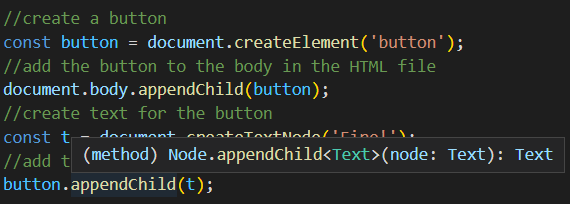
1. Create a two-dimensional array for the grid with 9 row and 9 columns. (briefly explain a two-dimensional array is)



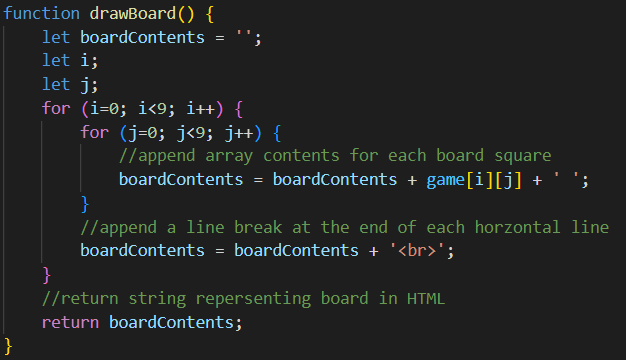
1. Use JavaScript to create a new div element that is going to hold our board and append it to the body of the html document.



1. Create a button, append it to the html, create a new textNode that says ‘Fire!’ and append that to the button.



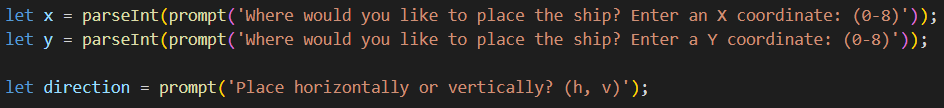
1. Make a function to draw the board and we are going to add in some new variables, boardContents, i, and j.
2. Loop through i and j using nested for loops to add the contents of the array to boardContents. Make sure to add a line break at the end and return boardContents.



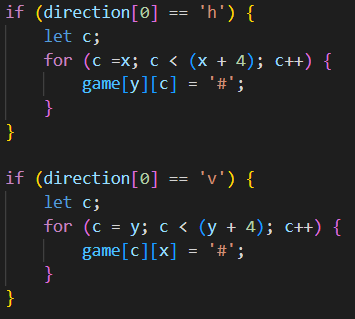
1. Add the board to the HTML (outside of the function we just made.)



1. Use prompt functions to ask the player to input the X and Y coordinates and if they want it horizontal or vertical. Make sure to change the type of X and Y to an integer.



1. We need to change the dots depending on the player’s inputs. We will do that by using if statements.



1. Redraw the board after the ship is placed.



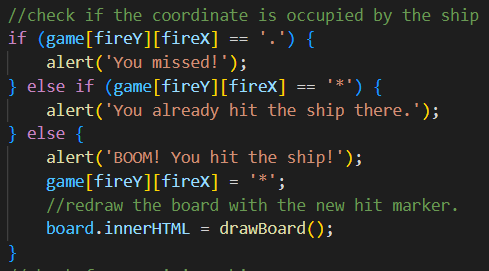
1. Add an event listener for when the button is clicked. We will be adding an anonyms function for the function.



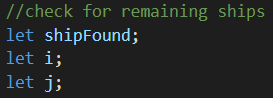
1. The function will prompt the user to enter an x and y coordinate. Make sure those are converted to an integer.



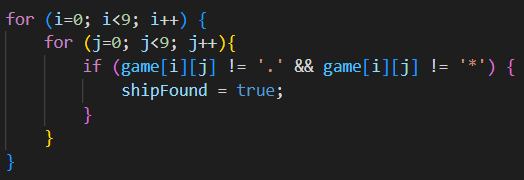
1. Now we will add if statements to check if we hit a ship. The first one will check if there is no ship (.). the next will check if we have already hit there (\*), and the last will run if we hit a ship.



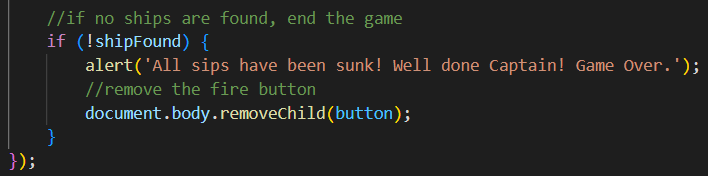
1. The last thing that the function needs to check for is if we have any remaining ships. Make 3 new variables called shipFound, i, and j.



1. Use nested for loops to check if there are any ships left. And set shipFound to true.

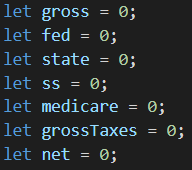


1. Lastly, if not ships are found, end the game and remove the fire button.



**JS Shell 3 - Taxes!**

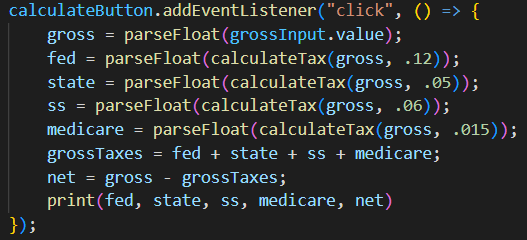
1. Add the script tag into the HTML
2. Add use strict
3. Declare variables (gross, fed, state, ss, medicare, net, grossTaxes)



1. Use the selector function ($) to reference the calculate taxes button and monthly salary input element from the HTML (show students how to use inspect on browser to find the ids)



1. Use the calculateTax function to calculate each of the taxes.
2. Add the parameters for the print function (fed, state, ss, medicare, net)

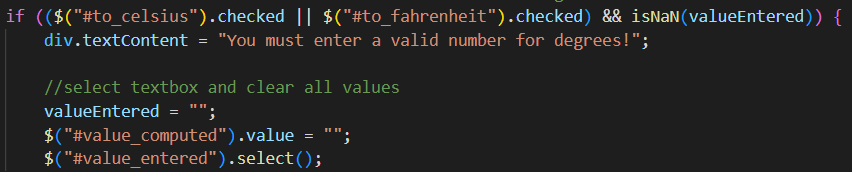


1. Check to make sure everything works.

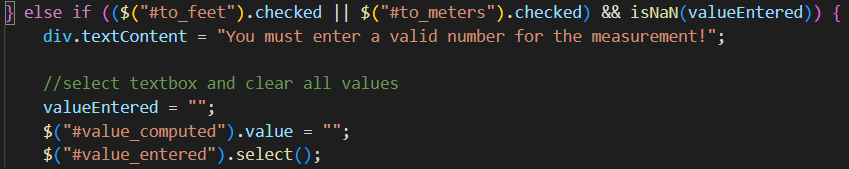
**Break**

**JS Shell 4 – The Converterlator**

1. Add the script tag into the HTML to link the JS file and the css file.
2. Add use strict
3. Get the value from the value entered textbox.
4. Use an if statement to check if to Celsius is checked, OR to Fahrenheit is checked AND if the input is not a number. Inside that, add the text to the div, clear value entered, value computed, and select the value entered textbox.



1. Add an else if checking if to feet is checked OR to meters is checked AND if the value entered is not a number. Inside that, Inside that, add the text to the div, clear value entered, value computed, and select the value entered textbox.



1. Add an else clause that is going clear any error messages and do our calculations for us.
2. Clear any error messages.
3. Add an if statement to see if to Celsius is checked. Inside that, set the value computed textbox to the function calculate Celsius with no decimal places. Select the value entered textbox.
4. Add an else if clause to see if to Fahrenheit is checked. Inside that, set the value computed textbox to the function calculate Fahrenheit with no decimal places. Select the value entered textbox.
5. Add an else if clause to see if to feet is checked. Inside that, set the value computed textbox to the function calculate feet with no decimal places. Select the value entered textbox.
6. Add an else clause. Inside that, set the value computed textbox to the function calculate meters with no decimal places. Select the value entered textbox.



**Go over what we will be working on next week – React**