To link your .js file to your .html file, you add a script element at the bottom of the HTML body. This could look like:

```
    <script src=”./folderName/fileName.js”></script>
  </body>
</html>
```

You can use the in-Chrome inspector as a console to see the effect of your code. The keyboard shortcut is command+option+I

**Control flow** is the order in which the computer executes code. It executes the code from top to bottom, so order is important! You can’t successfully put a code block above that is dependent on the code block below it–while it’s executing the top code block, the computer doesn’t yet know that the code block below it exists. You can interrupt the control flow by using things like conditional statements and loops.

## <u>Some basic terms:</u>

The **console** is where we can see the output of our code via console.log()

A **variable** is a named container for data.

An **array** is contained within [square brackets] and allows us to store a group of related data in a variable, which can then be accessed/used individually.

The **index** is the numbered position of a data item within an array, beginning with 0.

**Conditional statements** like “if”, “if…else”, and “else if” can interrupt the control flow. They allow for different outputs depending on whether the condition is met, or “truthy.”

**Loops** also interrupt the control flow. Loops will run until a stopping condition is met. Loops contain the following 3 statements within the parentheses, separated by semicolons:

-First statement = starting point for the loop. Define a new variable x, set it equal to 0 (first index position) → `var x = 0;`

-Second statement = condition. This determines how long the loop can continue. To set it to run until it has gone through the whole array once → `x < arrayName.length`

-Third statement = tells the array to iterate over each item incrementally: `x++`

Summary of three statements: (starting point; ending point; how to loop). This is followed by curly brackets containing the code we want to be executed: `{console.log(arrayName[x]);}`

A **function** is a set of instructions that tells the computer how to execute a certain task. Unlike conditional statements and loops, which execute automatically when the program is run, a function must be called (or activated) specifically. Function syntax:

```
function functionName(parameters) {
      //function body
}
```

To **call** the function, you just enter: `functionName();`