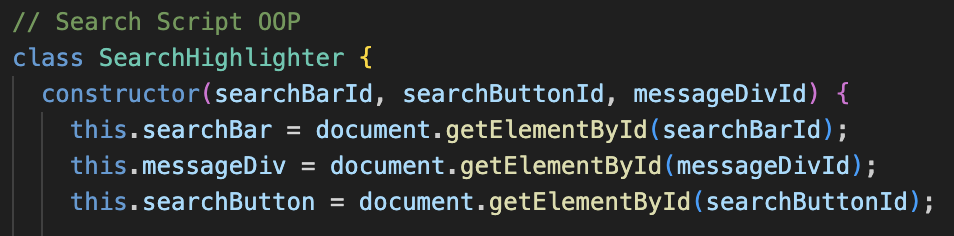
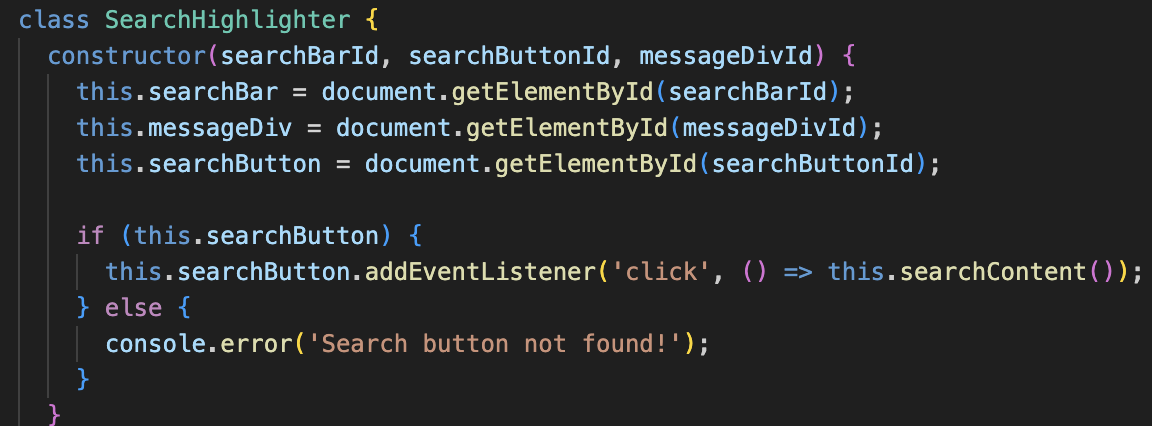
**My process of Implementing a search bar**

Throughout our Javascript studies, we have learned about Object Oriented Programming, such as encapsulation and instantiation. These are two details that I want to include in my search bar function.

I would like the search function to highlight results and show a message when no results are found, therefore, I included a message container within my index.html file for my website’s homepage.

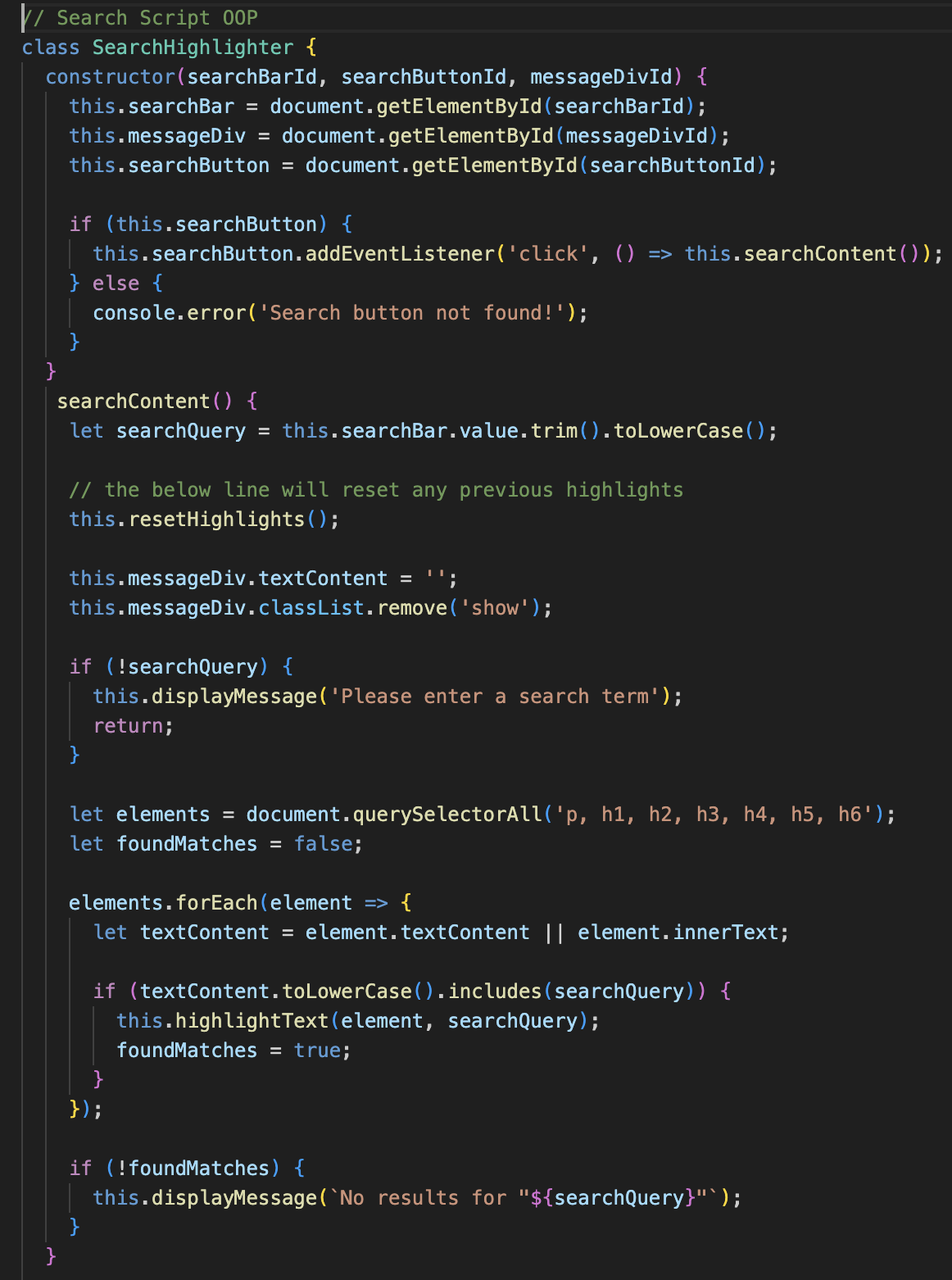
For the highlight function, I have done this through a class in JS called “SearchHighlighter”. This handles highlighting text and the search function.

The constructor uses the IDs as arguments and assigns them all to instance properties:

The 2nd part of my process involved adding in an event listener to handle the click listener on the search button. If the button is clicked, the search function kicks in. I added in a console message to display as I had issues with getting the search button working whilst I was testing my code. This turned out to be related to my searchButtonId being incorrect in my index file:

I also added in the code for the search functionality. I wanted to avoid case sensitivity and any unnecessary spaces or characters.

I did this by adding a trim to LowerCase argument. I also ensured that highlighted text was reset below this argument when a new search was initiated. Following this, any text in the message div is hidden so it is not visible in the next search. An ‘if’ statement is in place, so that in the event the search button is clicked and the search bar is empty, an error message will display.

I then added in a query selector so the search function searched all relevant elements on my homepage for the requested phrase/word.

Finally I added in the highlightText argument which creates a regular expression (regex) from the search query. It takes away case sensitivity for the search term and it uses “inner.HTML.replace()” to wrap each matching substring in a span with the class highlight. It also replaces any unwanted characters.

The resetHighlights function selects all elements with a highlight class and removes those classes, which remove the highlight from the text.

The displayMessage ensures that the error message shows if no search results are found on the page.

Finally an event listener is added to ensure all content on the page is loaded before the search functionality can take effect.