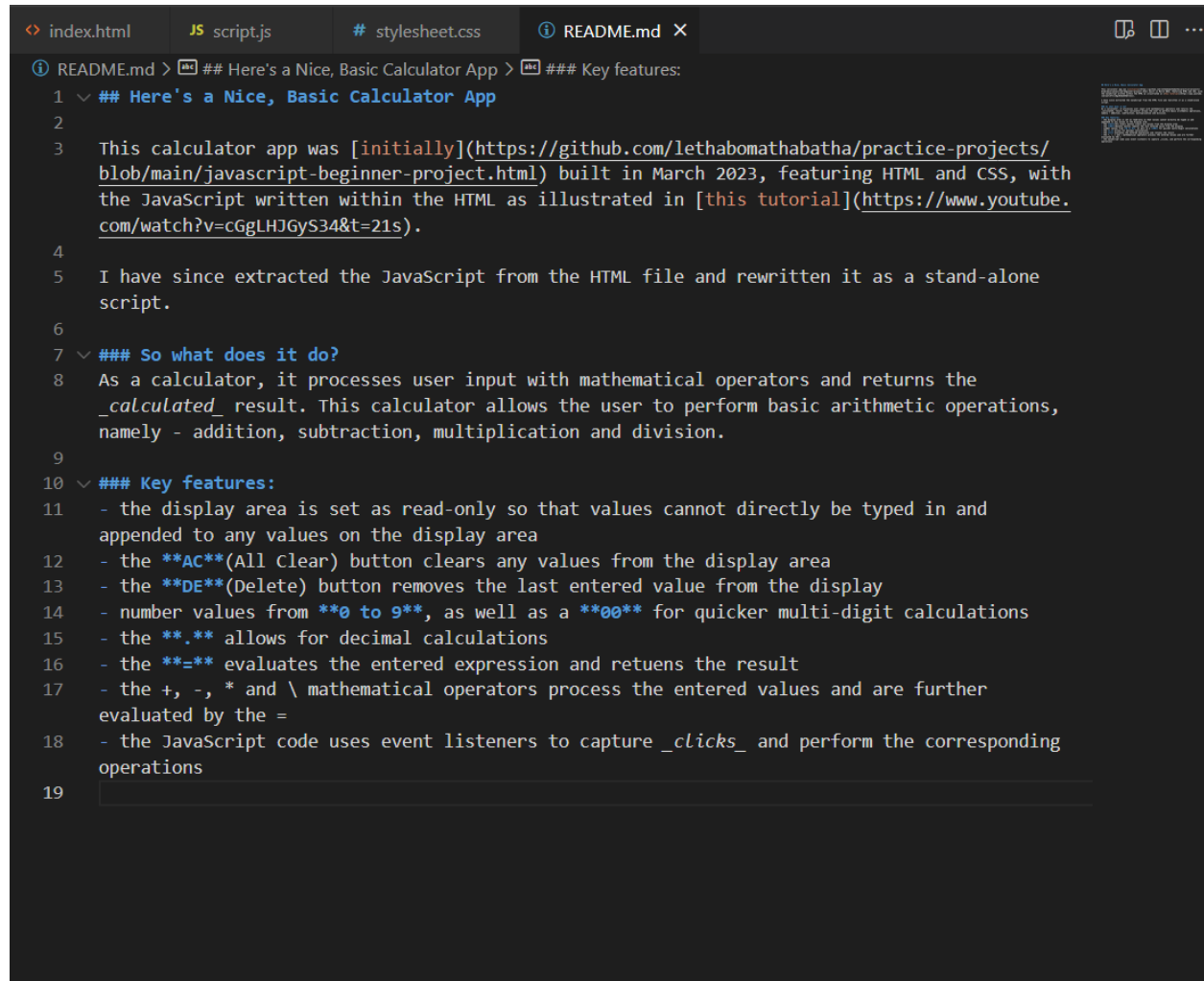


# DWA\_03.4 Knowledge Check\_DWA3.1

1. Please show how you applied a Markdown File to a piece of your code.



```
< index.html JS script.js # stylesheet.css ⓘ README.md X
ⓘ README.md > ## Here's a Nice, Basic Calculator App > ### Key features:
1  ✓ ## Here's a Nice, Basic Calculator App
2
3  This calculator app was [initially](https://github.com/lethabomathabatha/practice-projects/
   blob/main/javascript-beginner-project.html) built in March 2023, featuring HTML and CSS, with
   the JavaScript written within the HTML as illustrated in [this tutorial](https://www.youtube.
   com/watch?v=cGgLHJGyS34&t=21s).
4
5  I have since extracted the JavaScript from the HTML file and rewritten it as a stand-alone
   script.
6
7  ✓ ### So what does it do?
8  As a calculator, it processes user input with mathematical operators and returns the
   _calculated_ result. This calculator allows the user to perform basic arithmetic operations,
   namely - addition, subtraction, multiplication and division.
9
10 ✓ ### Key features:
11  - the display area is set as read-only so that values cannot directly be typed in and
   appended to any values on the display area
12  - the **AC**(All Clear) button clears any values from the display area
13  - the **DE**(Delete) button removes the last entered value from the display
14  - number values from **0 to 9**, as well as a **00** for quicker multi-digit calculations
15  - the **.** allows for decimal calculations
16  - the **=** evaluates the entered expression and returns the result
17  - the +, -, * and \ mathematical operators process the entered values and are further
   evaluated by the =
18  - the JavaScript code uses event listeners to capture _clicks_ and perform the corresponding
   operations
19
```

☰ README.md



## Here's a Nice, Basic Calculator App

This calculator app was [initially](#) built in March 2023, featuring HTML and CSS, with the JavaScript written within the HTML as illustrated in [this tutorial](#).

I have since extracted the JavaScript from the HTML file and rewritten it as a stand-alone script.

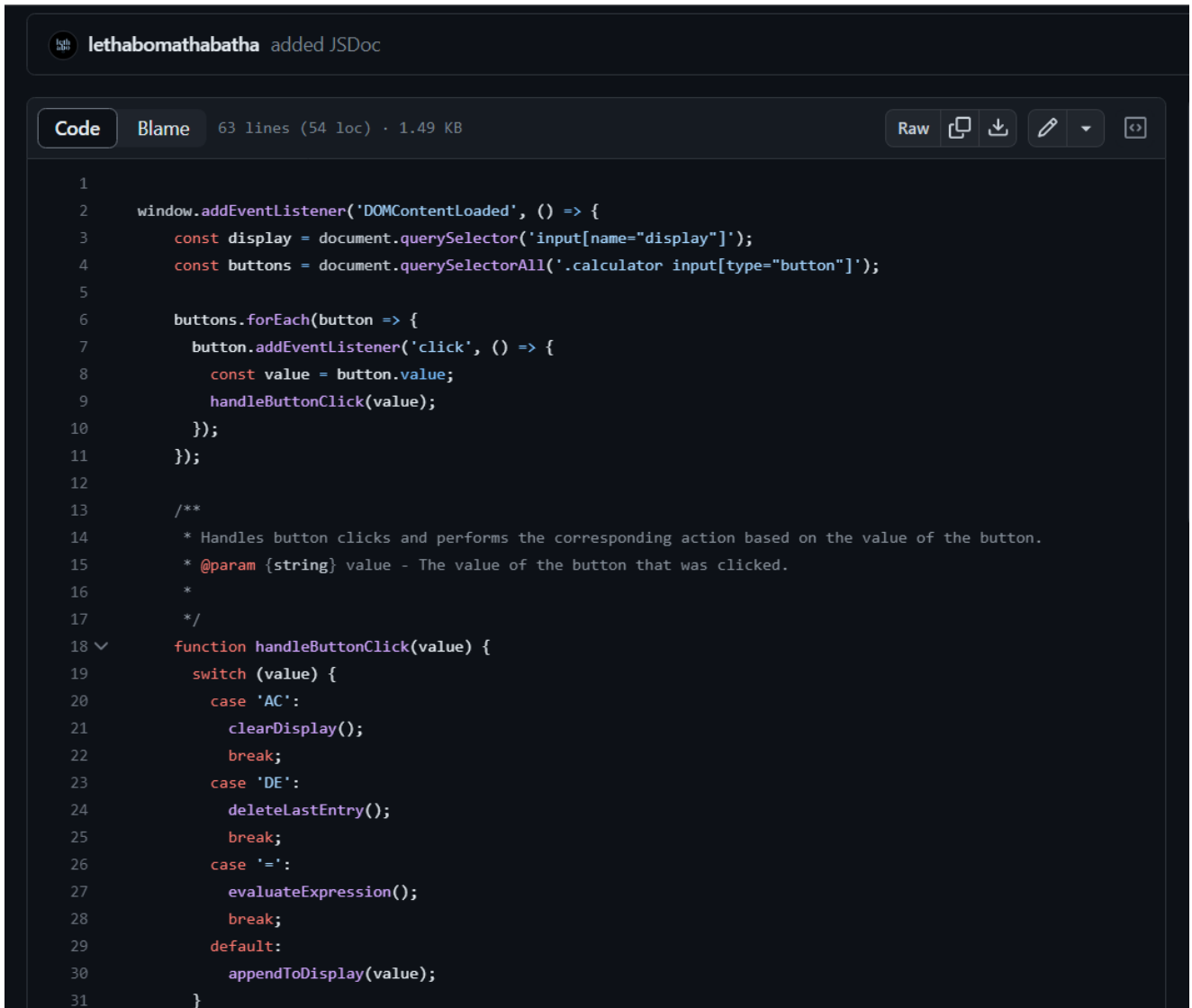
### So what does it do?

As a calculator, it processes user input with mathematical operators and returns the *calculated* result. This calculator allows the user to perform basic arithmetic operations, namely - addition, subtraction, multiplication and division.

### Key features:

- the display area is set as read-only so that values cannot directly be typed in and appended to any values on the display area
- the **AC**(All Clear) button clears any values from the display area
- the **DE**(Delete) button removes the last entered value from the display
- number values from **0 to 9**, as well as a **00** for quicker multi-digit calculations
- the **.** allows for decimal calculations
- the **=** evaluates the entered expression and returns the result
- the **+**, **-**, **\*** and **/** mathematical operators process the entered values and are further evaluated by the **=**
- the JavaScript code uses event listeners to capture *clicks* and perform the corresponding operations

2. Please show how you applied JSDoc Comments to a piece of your code.

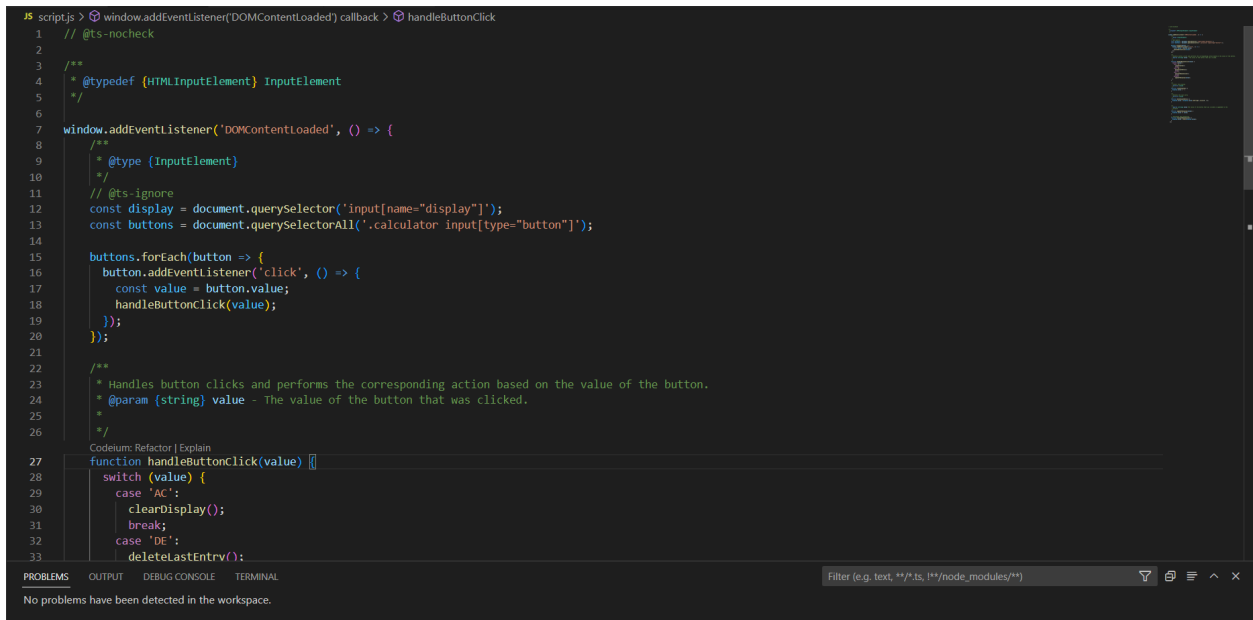


The screenshot shows a code editor with a dark theme. At the top, a notification bar indicates 'lethabomathabatha added JSDoc'. Below this, the editor has tabs for 'Code' and 'Blame', and a status bar showing '63 lines (54 loc) · 1.49 KB'. On the right side of the editor, there are icons for 'Raw', 'Copy', 'Download', 'Edit', and 'Toggle Full Screen'. The code is as follows:

```
1
2 window.addEventListener('DOMContentLoaded', () => {
3     const display = document.querySelector('input[name="display"]');
4     const buttons = document.querySelectorAll('.calculator input[type="button"]');
5
6     buttons.forEach(button => {
7         button.addEventListener('click', () => {
8             const value = button.value;
9             handleButtonClick(value);
10        });
11    });
12
13    /**
14     * Handles button clicks and performs the corresponding action based on the value of the button.
15     * @param {string} value - The value of the button that was clicked.
16     *
17     */
18    function handleButtonClick(value) {
19        switch (value) {
20            case 'AC':
21                clearDisplay();
22                break;
23            case 'DE':
24                deleteLastEntry();
25                break;
26            case '=':
27                evaluateExpression();
28                break;
29            default:
30                appendToDisplay(value);
31        }
32    }
33 }
```

```
32     }
33
34     /**
35      * Clears the display
36      * @returns {void}
37      */
38     function clearDisplay() {
39         display.value = '';
40     }
41
42     /**
43      * Deletes the last entry
44      * @returns {void}
45      */
46     function deleteLastEntry() {
47         display.value = display.value.toString().slice(0, -1);
48     }
49
50
51     /**
52      * @param {string} value The value of the button that was clicked is appended to the display
53      */
54     function appendToDisplay(value) {
55         display.value += value;
56     }
57
58     // Evaluates the expression
59     function evaluateExpression() {
60         display.value = eval(display.value);
61     }
62     });
```

3. Please show how you applied the @ts-check annotation to a piece of your code.



```
JS scripts.js > window.addEventListener('DOMContentLoaded') callback > handleButtonClick
1 // @ts-nocheck
2
3 /**
4  * @typedef {HTMLInputElement} InputElement
5  */
6
7 window.addEventListener('DOMContentLoaded', () => {
8   /**
9    * @type {InputElement}
10   */
11   // @ts-ignore
12   const display = document.querySelector('input[name="display"]');
13   const buttons = document.querySelectorAll('.calculator input[type="button"]');
14
15   buttons.forEach(button => {
16     button.addEventListener('click', () => {
17       const value = button.value;
18       handleButtonClick(value);
19     });
20   });
21
22   /**
23    * Handles button clicks and performs the corresponding action based on the value of the button.
24    * @param {string} value - The value of the button that was clicked.
25    */
26
27   function handleButtonClick(value) {
28     switch (value) {
29       case 'AC':
30         clearDisplay();
31         break;
32       case 'DE':
33         deleteLastEntry();
34     }
35   }
36
37   // ... (rest of the code)
38 }
```

4. As a BONUS, please show how you applied any other concept covered in the 'Documentation' module.

\*\*\*\*\*