- A <div> element with the class buttons contains all the calculator buttons, and it has the onclick attribute set to call the handleButtonClick(event) function when any button inside it is clicked.

- Each button inside the buttons container has a data-value attribute that holds the value or operator associated with that button.

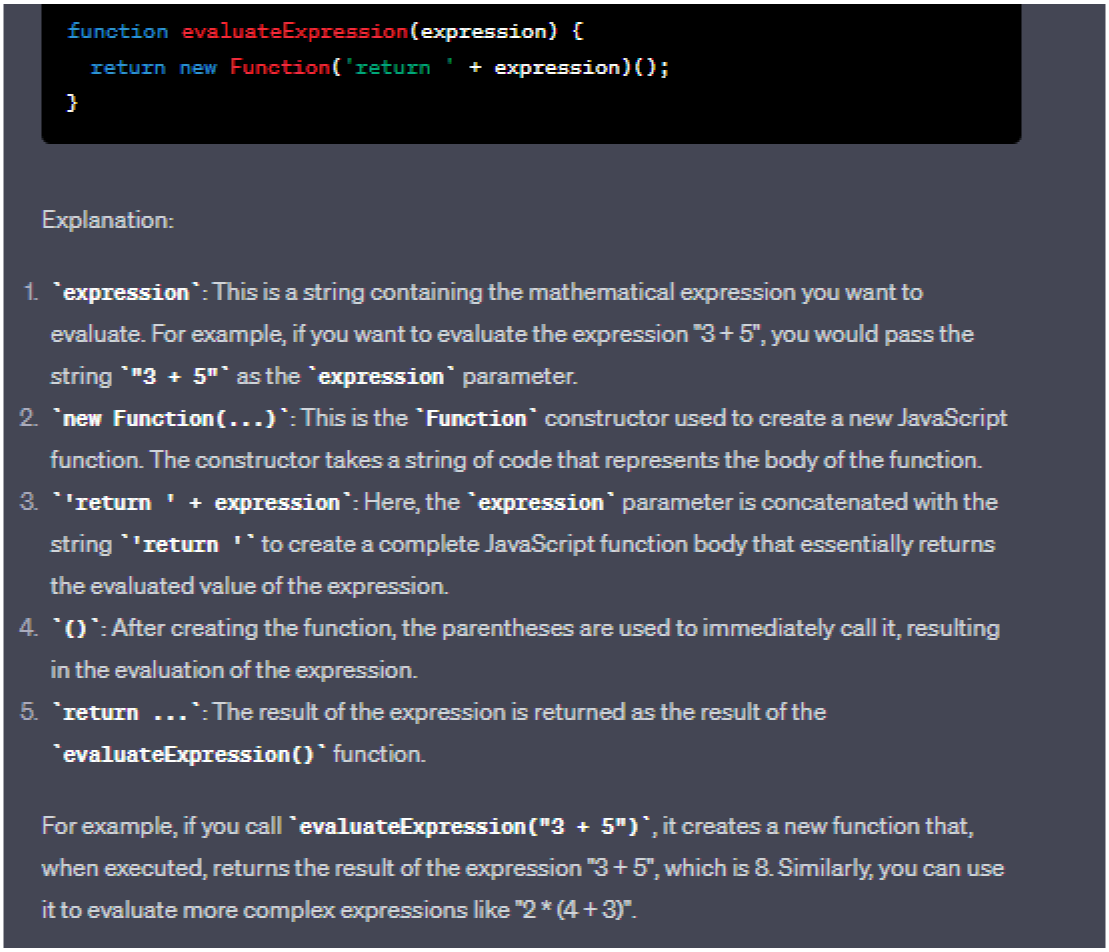
- The calculate() function uses the evaluateExpression(expression) function to evaluate the expression in the output and display the result.

- target: This is the variable that holds a reference to the HTML element that triggered the event. In this context, it's the button that was clicked.

- .matches("button"): The .matches() method is used to check if the element matches a specified CSS selector. In this case, it's checking if the clicked element matches the selector "button", which targets <button> elements.

- If the condition evaluates to true, it means that the clicked element is a <button> element, and the code inside the curly braces { ... } will be executed. If the condition is false, the code block will be skipped.

- The evaluateExpression(expression) function safely evaluates a mathematical expression by constructing a new function using the Function constructor.

- 

```
function evaluateExpression(expression) {
  return new Function('return ' + expression)();
}
```

Explanation:

1. `expression`: This is a string containing the mathematical expression you want to evaluate. For example, if you want to evaluate the expression "3 + 5", you would pass the string `"3 + 5"` as the `expression` parameter.
2. `new Function(...)`: This is the `Function` constructor used to create a new JavaScript function. The constructor takes a string of code that represents the body of the function.
3. `'return ' + expression`: Here, the `expression` parameter is concatenated with the string `'return '` to create a complete JavaScript function body that essentially returns the evaluated value of the expression.
4. `()`: After creating the function, the parentheses are used to immediately call it, resulting in the evaluation of the expression.
5. `return ...`: The result of the expression is returned as the result of the `evaluateExpression()` function.

For example, if you call `evaluateExpression("3 + 5")`, it creates a new function that, when executed, returns the result of the expression "3 + 5", which is 8. Similarly, you can use it to evaluate more complex expressions like "2 * (4 + 3)".

- grid-template-columns: This is the property that defines the sizing and layout of columns in the grid.

- repeat(4, 1fr): This part specifies the pattern of column sizing. In this case, it's repeating a column sizing pattern 4 times. The pattern is defined as 1fr, which means that each column occupies an equal share of the available space within the grid container.