

Reporte de Diagnóstico Inicial / Initial Diagnosis Report.

ANTES

The image displays a code editor window showing JavaScript code for a calculator application. The code is a legacy script with various functions for handling numbers, symbols, and operations. Below the code editor, the ESLint Online Playground interface is visible, showing the same code with ESLint rules applied. The interface includes a file explorer on the left, a code editor in the center, and a console at the bottom displaying 14 errors: "Strings must use doublequote." for various lines in the code. The ESLint Online Playground header shows the URL and options for More Plugins and More Examples. The console also includes a "netlify" logo.

```
1 // ----- SCRIPT CALCULADORA LEGACY v1.2 -----
2 // NO TOCAR NADA - FUNCIONA (A VECES)
3 var buffer = "0";
4 var memoria = 0;
5 var ultimo_operador;
6 function handleNumber(numStr) {
7   if (buffer === "0") {
8     buffer = numStr;
9   } else {
10    buffer += numStr;
11  }
12  updateScreen();
13 }
14 function handleSymbol(symbol) {
15   switch (symbol) {
16     case 'C':
17       buffer = "0";
18       memoria = 0;
19       ultimo_operador = null;
20       break;
21     case '=':
22       if (ultimo_operador === null) {
23         return;
24       }
25       flushOperation(parseInt(buffer));
26       ultimo_operador = null;
27       buffer = "" + memoria;
28       memoria = 0;
29       break;
30     case '+':
31     case '-':
32     case '*':
33     case '/':
34       handleMath(symbol);
35       break;
36   }
37   updateScreen();
38 }
39 function handleMath(symbol) {
40   if (buffer === '0' && memoria === 0) {
41     return;
42   }
43   var intBuffer = parseInt(buffer);
44   if (memoria === 0) {
45     memoria = intBuffer;
46   } else {
47     flushOperation(intBuffer);
48   }
49   ultimo_operador = symbol;
50   buffer = "0";
51 }
52 function flushOperation(intBuffer) {
53   if (ultimo_operador === '+') {
54     memoria += intBuffer;
55   } else if (ultimo_operador === '-') {
56     memoria -= intBuffer;
57   } else if (ultimo_operador === '*') {
58     memoria *= intBuffer;
59   } else if (ultimo_operador === '/') {
60     memoria /= intBuffer;
61   }
62 }
63 function updateScreen(){
64   var laPantalla = document.getElementById("display");
65   laPantalla.innerText = buffer;
66 }
67 // INICIALIZADOR DE BOTONES
68 function init(){
```

ESLint Online Playground

More Plugins More Examples

calculador.js

example.js

eslint.config.js

package.json

Preview Apply Fix

PROBLEMS

CONSOLE

- Strings must use doublequote. (quotes) [17:6-9]
- Strings must use doublequote. (quotes) [22:6-9]
- Strings must use doublequote. (quotes) [31:6-9]
- Strings must use doublequote. (quotes) [32:6-9]
- Strings must use doublequote. (quotes) [33:6-9]
- Strings must use doublequote. (quotes) [34:6-9]
- Strings must use doublequote. (quotes) [41:16-19]
- Strings must use doublequote. (quotes) [54:25-28]
- Strings must use doublequote. (quotes) [56:32-35]
- Strings must use doublequote. (quotes) [58:32-35]
- Strings must use doublequote. (quotes) [60:32-35]
- Strings must use doublequote. (quotes) [60:32-35]
- Strings must use doublequote. (quotes) [60:32-35]
- Strings must use doublequote. (quotes) [60:32-35]
- Strings must use doublequote. (quotes) [60:32-35]

netlify

DESPUES

```
1
2 // ----- SCRIPT CALCULADORA LEGACY v1.2 -----
3 // NO TOCAR NADA - FUNCIONA A VECES
4 var buffer = "0";
5 var memoria = 0;
6 var ultimo_operador;
7 function handleNumber(numStr) {
8   if (buffer === "0") {
9     buffer = numStr;
10  } else {
11    buffer += numStr;
12  }
13  updateScreen();
14 }
15 function handleSymbol(symbol) {
16   switch (symbol) {
17     case "C":
18       buffer = "0";
19       memoria = 0;
20       ultimo_operador = null;
21       break;
22     case "+":
23       if (ultimo_operador === null) {
24         return;
25       }
26       flushOperation(parseInt(buffer));
27       ultimo_operador = null;
28       buffer = "" + memoria;
29       memoria = 0;
30       break;
31     case "*":
32     case "/":
33     case "%":
34       handleMath(symbol);
35       break;
36   }
37   updateScreen();
38 }
39 function handleMath(symbol) {
40   if (buffer === "0" && memoria === 0) {
41     return;
42   }
43   var intBuffer = parseInt(buffer);
44   if (memoria === 0) {
45     memoria = intBuffer;
46   } else {
47     flushOperation(intBuffer);
48   }
49   ultimo_operador = symbol;
50   buffer = "0";
51 }
52 function flushOperation(intBuffer) {
53   if (ultimo_operador === "+") {
54     memoria += intBuffer;
55   } else if (ultimo_operador === "-") {
56     memoria -= intBuffer;
57   } else if (ultimo_operador === "*") {
58     memoria *= intBuffer;
59   } else if (ultimo_operador === "/") {
60     memoria /= intBuffer;
61   }
62 }
63 function updateScreen(){
64   var document;
65   var laPantalla = document.getElementById("display");
66   laPantalla.innerText = buffer;
67 }
68
69 // INICIALIZADOR DE BOTONES
70 function init(){
71   var document;
72   var console;
73   console.log("Calculadora inicializada...");
74   document.querySelector(".buttons").addEventListener("click", function(event) {
75     handleClick(event.target.innerText);
76   });
77 }
78 function handleClick(value) {
79   if (isNaN(parseInt(value))) {
80     handleSymbol(value);
81   } else {
82     handleNumber(value);
83   }
84 }
85 init();
```

ESLint Online Playground

More PluginsMore Examples

+

calculador.js

calculador.js

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

calculador.js

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

Preview

PROBLEMS

CONSOLE

ESLint

ESLint es una herramienta de análisis estático de código (conocida como linter) para JavaScript y TypeScript. Su principal objetivo es identificar y reportar problemas en tu código antes de que se ejecute.

Esto incluye:

Errores potenciales: Detecta patrones de código que probablemente sean errores (por ejemplo, usar una variable antes de definirla).

Buenas prácticas: Ayuda a seguir un conjunto de reglas y buenas prácticas de programación.

Problemas de estilo: Puede señalar inconsistencias en el formato (aunque para corregir el formato automáticamente, Prettier es más específico).

Prettier

Prettier es un formateador de código (o code formatter). Su único propósito es reescribir tu código para que cumpla con un conjunto de reglas de estilo consistentes y predefinidas.

A diferencia de ESLint (que busca errores), Prettier solo se preocupa por la apariencia del código:

- Ajusta la indentación (espacios o tabulaciones).
- Maneja los saltos de línea para respetar un ancho máximo.
- Estandariza el uso de comillas (simples o dobles).
- Añade o quita espacios, puntos y comas, etc., según sus reglas.