

script.js

```
1 class Calculator {
2   constructor(previousOperandTextElement, currentOperandTextElement) {
3     this.previousOperandTextElement = previousOperandTextElement
4     this.currentOperandTextElement = currentOperandTextElement
5     this.clear()
6   }
7
8   clear() {
9     this.currentOperand = ''
10    this.previousOperand = ''
11    this.operation = undefined
12  }
13
14  delete() {
15    this.currentOperand = this.currentOperand.toString().slice(0, -1)
16  }
17
18  appendNumber(number) {
19    if (number === '.' && this.currentOperand.includes('.')) return
20    this.currentOperand = this.currentOperand.toString() + number.toString()
21  }
22
23  chooseOperation(operation) {
24    if (this.currentOperand === '') return
25    if (this.previousOperand !== '') {
26      this.compute()
27    }
28    this.operation = operation
29    this.previousOperand = this.currentOperand
30    this.currentOperand = ''
31  }
32
33  compute() {
34    let computation
35    const prev = parseFloat(this.previousOperand)
36    const current = parseFloat(this.currentOperand)
37    if (isNaN(prev) || isNaN(current)) return
38    switch (this.operation) {
39      case '+':
40        computation = prev + current
41        break
42      case '-':
43        computation = prev - current
44        break
45      case '*':
46        computation = prev * current
47        break
48      case '÷':
```

```
49         computation = prev / current
50         break
51     default:
52         return
53     }
54     this.currentOperand = computation
55     this.operation = undefined
56     this.previousOperand = ''
57 }
58
59 getDisplayNumber(number) {
60     const stringNumber = number.toString()
61     const integerDigits = parseFloat(stringNumber.split('.')[0])
62     const decimalDigits = stringNumber.split('.')[1]
63     let integerDisplay
64     if (isNaN(integerDigits)) {
65         integerDisplay = ''
66     } else {
67         integerDisplay = integerDigits.toLocaleString('en', { maximumFractionDigits: 0 })
68     }
69     if (decimalDigits !== null) {
70         return `${integerDisplay}.${decimalDigits}`
71     } else {
72         return integerDisplay
73     }
74 }
75
76 updateDisplay() {
77     this.currentOperandTextElement.innerText =
78         this.getDisplayNumber(this.currentOperand)
79     if (this.operation !== null) {
80         this.previousOperandTextElement.innerText =
81             `${this.getDisplayNumber(this.previousOperand)} ${this.operation}`
82     } else {
83         this.previousOperandTextElement.innerText = ''
84     }
85 }
86 }
87
88
89 const numberButtons = document.querySelectorAll('[data-number]')
90 const operationButtons = document.querySelectorAll('[data-operation]')
91 const equalsButton = document.querySelector('[data-equals]')
92 const deleteButton = document.querySelector('[data-delete]')
93 const allClearButton = document.querySelector('[data-all-clear]')
94 const previousOperandTextElement = document.querySelector('[data-previous-operand]')
95 const currentOperandTextElement = document.querySelector('[data-current-operand]')
96
97 const calculator = new Calculator(previousOperandTextElement, currentOperandTextElement)
98
```

```
99     numberButtons.forEach(button => {
100         button.addEventListener('click', () => {
101             calculator.appendNumber(button.innerText)
102             calculator.updateDisplay()
103         })
104     })
105
106     operationButtons.forEach(button => {
107         button.addEventListener('click', () => {
108             calculator.chooseOperation(button.innerText)
109             calculator.updateDisplay()
110         })
111     })
112
113     equalsButton.addEventListener('click', button => {
114         calculator.compute()
115         calculator.updateDisplay()
116     })
117
118     allClearButton.addEventListener('click', button => {
119         calculator.clear()
120         calculator.updateDisplay()
121     })
122
123     deleteButton.addEventListener('click', button => {
124         calculator.delete()
125         calculator.updateDisplay()
126     })
```