



# HTML, CSS, ET JAVASCRIPT

## RAPPORT

# Résumé du Workshop : Développement Web

OUARACH ABDERRAHMANE  
BABA SALMANE  
NAJMI YASSIN

*Encadrant :*  
Hassan FARSI

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Concepts Clés Abordés</b>	<b>2</b>
2.1	HTML . . . . .	2
2.2	CSS . . . . .	2
2.3	JavaScript . . . . .	2
<b>3</b>	<b>Activités du Workshop</b>	<b>2</b>
<b>4</b>	<b>Les codes</b>	<b>3</b>
4.1	Code HTML . . . . .	3
4.2	Code CSS (avec modification de l'arrière-plan) . . . . .	4
4.3	Code JavaScript . . . . .	6
<b>5</b>	<b>Exemple de Résultat</b>	<b>8</b>
<b>6</b>	<b>Conclusion</b>	<b>10</b>

# 1 Introduction

Ce workshop avait pour objectif principal de fournir aux participants une compréhension des bases du développement web front-end. Il a mis l'accent sur trois langages fondamentaux :

- **HTML** : Pour structurer les pages web.
- **CSS** : Pour concevoir et appliquer des styles aux pages web.
- **JavaScript** : Pour ajouter de l'interactivité et des comportements dynamiques.

L'approche pratique a permis aux participants d'acquérir des compétences concrètes pour construire et personnaliser des applications web.

## 2 Concepts Clés Abordés

### 2.1 HTML

HTML constitue la base de toute page web, en définissant sa structure à travers des balises comme `<div>`, `<p>`, et `<img>`. Points essentiels :

- Différence entre les éléments sémantiques et non sémantiques.
- Bonnes pratiques pour structurer le contenu web.

### 2.2 CSS

CSS ajoute une dimension visuelle en gérant les mises en page, les couleurs et les polices. Les participants ont appris :

- Comment lier des fichiers CSS à un document HTML.
- Les propriétés courantes comme `background-color`, `font-size`, et `position`.

### 2.3 JavaScript

JavaScript permet d'introduire de l'interactivité, comme la réponse aux actions de l'utilisateur. Notions couvertes :

- Variables, fonctions et gestion des événements.
- Liaison de fichiers JavaScript externes pour une meilleure organisation du code.

## 3 Activités du Workshop

Le point culminant était la création d'une calculatrice fonctionnelle. Les étapes comprenaient :

1. Conception de la mise en page avec HTML.
2. Stylisation de l'interface avec CSS.
3. Implémentation des opérations en JavaScript.

Les participants ont également exploré des personnalisations avancées, comme l'ajout de boutons de changement de thème.

article listings xcolor

## 4 Les codes

### 4.1 Code HTML

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Calculator</title>
7   <link rel="stylesheet" href="indexCSS.css">
8 </head>
9 <body>
10  <div class="calculator-container">
11    <!-- Dropdown for selecting number button color -->
12    <select id="colorSelectorNumbers">
13      <option value="#333">Default</option>
14      <option value="#007BFF">Blue</option>
15      <option value="#28A745">Green</option>
16      <option value="#FFC107">Yellow</option>
17      <option value="#DC3545">Red</option>
18    </select>
19    <!-- Dropdown for selecting other button color -->
20    <select id="colorSelectorOthers">
21      <option value="#f39c12">Default</option>
22      <option value="#007BFF">Blue</option>
23      <option value="#28A745">Green</option>
24      <option value="#FFC107">Yellow</option>
25      <option value="#DC3545">Red</option>
26    </select>
27
28    <div class="calculator">
29      <input type="text" class="calculator-screen" value="" disabled
30        placeholder="0" />
31      <div class="calculator-keys">
32        <button type="button" value="7">7</button>
33        <button type="button" value="8">8</button>
34        <button type="button" value="9">9</button>
35        <button type="button" class="operator" value="+">+</button>
36
37        <button type="button" value="4">4</button>
38        <button type="button" value="5">5</button>
39        <button type="button" value="6">6</button>
40        <button type="button" class="operator" value="-">-</button>
41
42        <button type="button" value="1">1</button>
43        <button type="button" value="2">2</button>
44        <button type="button" value="3">3</button>
45        <button type="button" class="operator" value="*">&times;</button>
46
47        <button type="button" value="0">0</button>
48        <button type="button" class="decimal" value=".">.</button>
49        <button type="button" class="all-clear" value="all-clear">AC</button>

```

```

49         <button type="button" class="operator" value="/">&divide;</
           button>
50
51         <button type="button" class="equal-sign_operator" value="=">=</
           button>
52     </div>
53 </div>
54 </div>
55 <script src="indexJs.js"></script>
56 </body>
57 </html>

```

## 4.2 Code CSS (avec modification de l'arrière-plan)

```

1  * {
2    box-sizing: border-box;
3    margin: 0;
4    padding: 0;
5  }
6
7  body {
8    display: flex;
9    flex-direction: column;
10   align-items: center;
11   justify-content: center;
12   height: 100vh;
13   background: linear-gradient(135deg, #1d2671, #c33764);
14   font-family: 'Roboto', sans-serif;
15   color: #fff;
16 }
17
18 /* Calculator container */
19 .calculator-container {
20   text-align: center;
21   margin-bottom: 20px;
22 }
23
24 select {
25   margin: 5px;
26   padding: 10px;
27   border-radius: 5px;
28   border: none;
29   font-size: 1rem;
30   cursor: pointer;
31 }
32
33 /* Calculator styles */
34 .calculator {
35   background-color: #2e3b4e;
36   border-radius: 15px;
37   overflow: hidden;
38   box-shadow: 0 10px 20px rgba(0, 0, 0, 0.3);
39   max-width: 400px;
40   width: 100%;
41 }

```

```
42
43 .calculator-screen {
44     width: 100%;
45     height: 100px;
46     border: none;
47     background-color: #1c2533;
48     color: #4bcffa;
49     text-align: right;
50     padding: 15px 20px;
51     font-size: 2.5rem;
52     border-bottom: 1px solid #1e293b;
53 }
54
55 .calculator-keys {
56     display: grid;
57     grid-template-columns: repeat(4, 1fr);
58     gap: 15px;
59     padding: 20px;
60     background-color: #1e2a38;
61 }
62
63 button {
64     height: 70px;
65     border-radius: 10px;
66     border: none;
67     font-size: 1.5rem;
68     font-weight: bold;
69     color: #fff;
70     background-color: #333;
71     cursor: pointer;
72     transition: all 0.2s ease;
73 }
74
75 button:hover {
76     transform: scale(1.1);
77     background-color: #555;
78 }
79
80 .operator {
81     background-color: #f39c12;
82 }
83
84 .operator:hover {
85     background-color: #d87a0d;
86 }
87
88 .equal-sign {
89     height: calc(100px + 15px);
90     grid-column: span 4;
91     font-size: 2rem;
92     background-color: #20bf6b;
93 }
94
95 .equal-sign:hover {
96     background-color: #26de81;
97 }
98
```

```

99 .all-clear {
100     background-color: #eb3b5a;
101 }
102
103 .all-clear:hover {
104     background-color: #fc5c65;
105 }
106
107 .decimal {
108     background-color: #9b59b6;
109 }
110
111 .decimal:hover {
112     background-color: #8e44ad;
113 }

```

### 4.3 Code JavaScript

```

1  let firstOperand = '';
2  let secondOperand = '';
3  let currentOperator = null;
4  let awaitingSecondOperand = false;
5
6  const display = document.querySelector('.calculator-screen');
7  const keys = document.querySelector('.calculator-keys');
8  const colorSelectorNumbers = document.getElementById('
    colorSelectorNumbers');
9  const colorSelectorOthers = document.getElementById('colorSelectorOthers
    ');
10
11 // Select number and non-number buttons
12 const numberButtons = document.querySelectorAll('.calculator-keys button
    :not(.operator):not(.equal-sign):not(.all-clear):not(.decimal)');
13 const otherButtons = document.querySelectorAll('.calculator-keys button.
    operator, .calculator-keys button.equal-sign, .calculator-keys button
    .all-clear, .calculator-keys button.decimal');
14
15 const updateDisplay = () => {
16     display.value = awaitingSecondOperand ? secondOperand : firstOperand;
17 };
18
19 const resetCalculator = () => {
20     firstOperand = '';
21     secondOperand = '';
22     currentOperator = null;
23     awaitingSecondOperand = false;
24     updateDisplay();
25 };
26
27 const inputNumber = (number) => {
28     if (awaitingSecondOperand) {
29         secondOperand += number;
30     } else {
31         firstOperand += number;
32     }

```

```
33     updateDisplay();
34 };
35
36 const inputDecimal = () => {
37     if (awaitingSecondOperand) {
38         if (!secondOperand.includes('.') ) {
39             secondOperand += '.';
40         }
41     } else if (!firstOperand.includes('.') ) {
42         firstOperand += '.';
43     }
44     updateDisplay();
45 };
46
47 const calculate = () => {
48     let result;
49     const first = parseFloat(firstOperand);
50     const second = parseFloat(secondOperand);
51
52     if (Number.isNaN(first) || Number.isNaN(second)) return;
53
54     switch (currentOperator) {
55         case '+':
56             result = first + second;
57             break;
58         case '-':
59             result = first - second;
60             break;
61         case '*':
62             result = first * second;
63             break;
64         case '/':
65             result = first / second;
66             break;
67         default:
68             return;
69     }
70
71     firstOperand = String(result);
72     secondOperand = '';
73     awaitingSecondOperand = false;
74     currentOperator = null;
75     updateDisplay();
76 };
77
78 const inputOperator = (operator) => {
79     if (!firstOperand) return;
80
81     if (secondOperand) {
82         calculate();
83     }
84
85     currentOperator = operator;
86     awaitingSecondOperand = true;
87 };
88
89 keys.addEventListener('click', (event) => {
```



```
90     const { target } = event;
91     const { value } = target;
92
93     if (!target.matches('button')) return;
94
95     switch (value) {
96       case 'all-clear':
97         resetCalculator();
98         break;
99       case '=':
100        calculate();
101        break;
102       case '.':
103        inputDecimal();
104        break;
105       case '+':
106       case '-':
107       case '*':
108       case '/':
109        inputOperator(value);
110        break;
111       default:
112        if (Number.isInteger(parseFloat(value))) {
113          inputNumber(value);
114        }
115    }
116  });
117
118  // Change number button colors dynamically
119  colorSelectorNumbers.addEventListener('change', (event) => {
120    const selectedColor = event.target.value;
121    numberButtons.forEach(button => {
122      button.style.backgroundColor = selectedColor;
123    });
124  });
125
126  // Change non-number button colors dynamically
127  colorSelectorOthers.addEventListener('change', (event) => {
128    const selectedColor = event.target.value;
129    otherButtons.forEach(button => {
130      button.style.backgroundColor = selectedColor;
131    });
132  });
133
134  document.addEventListener('DOMContentLoaded', updateDisplay);
```

## 5 Exemple de Résultat

Voici une capture d'écran montrant le résultat obtenu lors du workshop (on a modifié l'arrière-plan du page en modifiant le fichier indexCSS.css) :

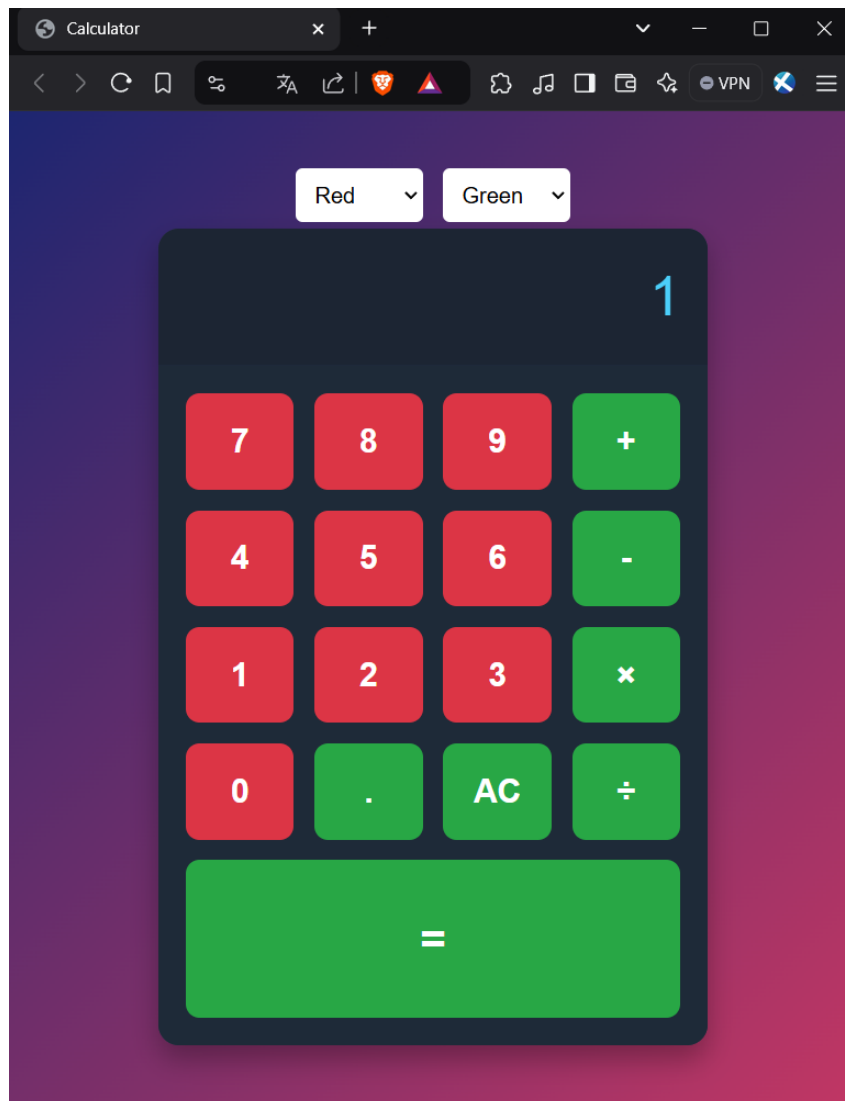
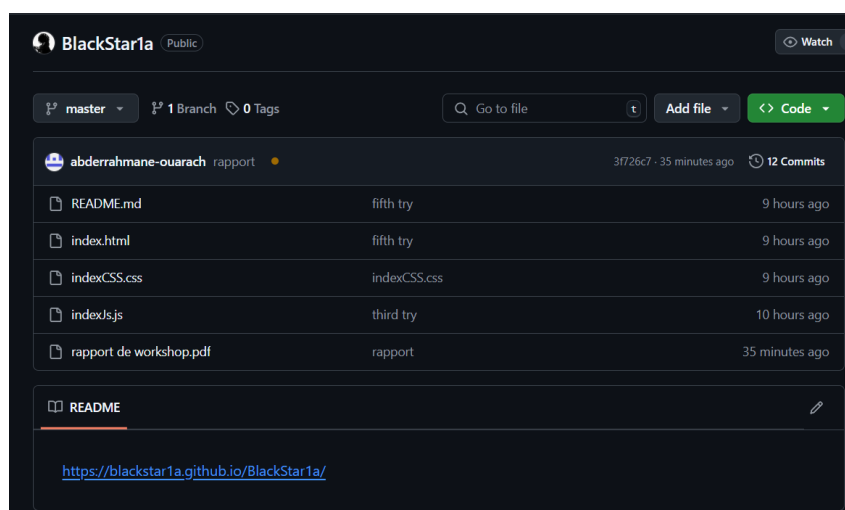


FIGURE 1 – Capture d'écran de la calculatrice fonctionnelle.

FIGURE 2 – Les fichiers de code sur GitHub  
Clique ici pour afficher le site web

## 6 Conclusion

Ce workshop a fourni une introduction complète au développement web. En mettant l'accent sur la pratique, les participants ont gagné en confiance pour créer et personnaliser des applications web. L'expérience a également renforcé les compétences en travail d'équipe et en résolution de problèmes, essentielles pour des projets futurs.