

JSのデバッグを試みよう

複雑な計算アプリ

数値を入力してください: 操作を選択してください:

Result: 27

- Input: 3, Operation: square, Result: 9
- Input: 3, Operation: cube, Result: 27

この教材でできること

- HTMLについて理解できる
- JSについて理解できる

どんな教材？

JSに起きているバグを修正して、求めている処理にしよう

目次

- ①コードを書く前の準備をしよう
- ②完成系を確認してみよう
- ③コードの間違いを修正しよう
- ④答え合わせをしてみよう

①コードを書く前の準備をしよう

このページでは使うコードの準備などを行うよ

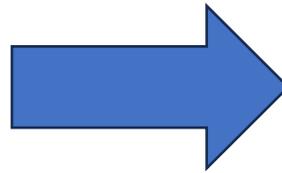
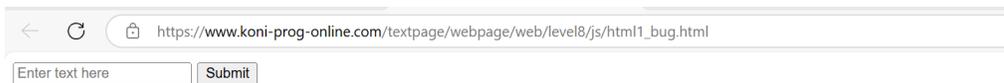
①コードを書く前の準備をしよう

以下のURLからコードを取ってこよう

HTML	JS
Buggy Website	koni-prog-online.com/textpage/webpage/web/level8/js/js1_bug.js
New Buggy Website	koni-prog-online.com/textpage/webpage/web/level8/js/js2_bug.js
修正済み複雑なウェブサイト	koni-prog-online.com/textpage/webpage/web/level8/js/js3_bug.js

①コードを書く前の準備をしよう

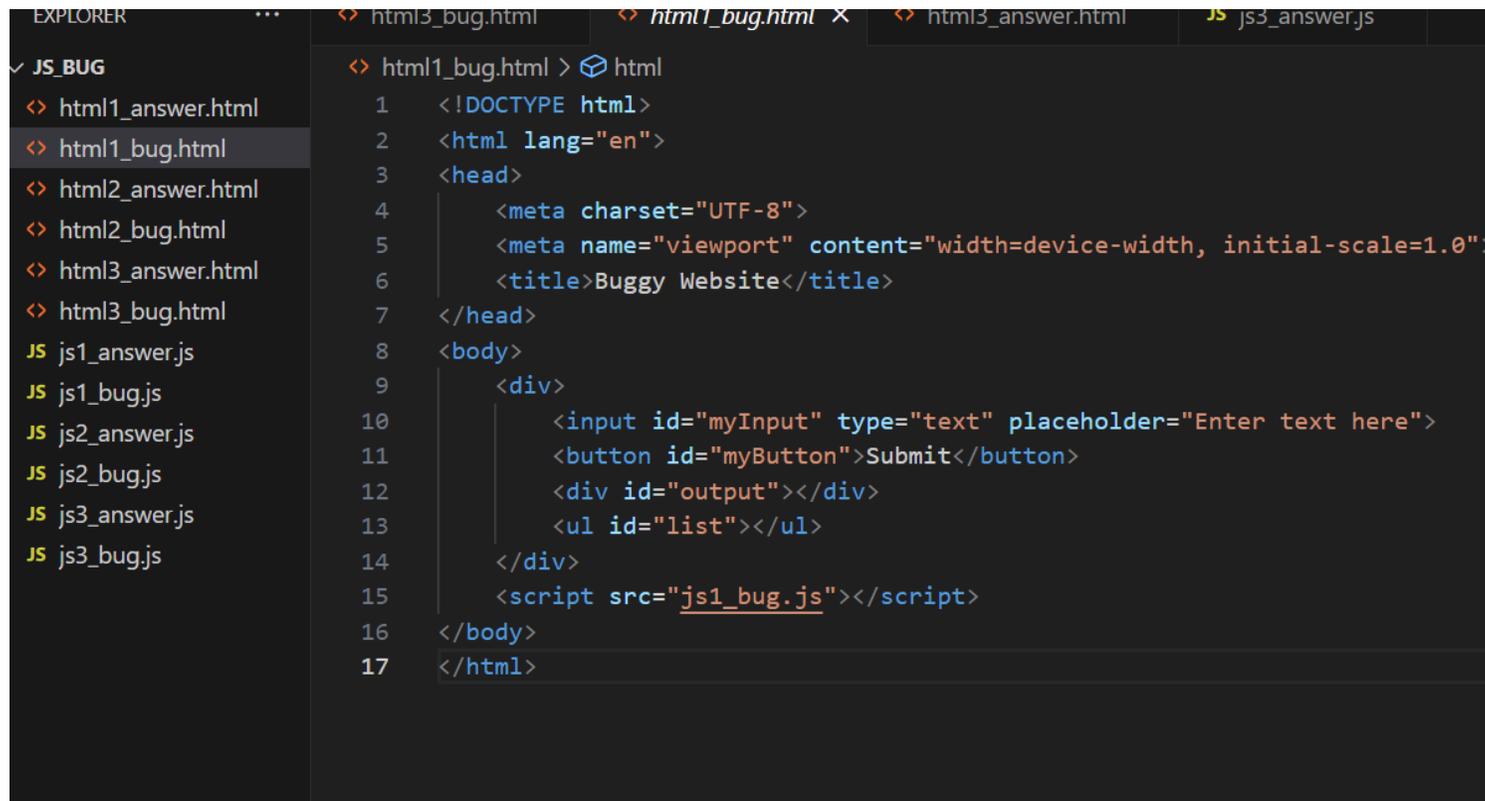
ページに行ったら「右クリック」->「ページのソースを表示」からコードをコピーしよう



```
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>Buggy Website</title>
7 </head>
8 <body>
9   <div>
10    <input id="myInput" type="text" placeholder="Enter text here">
11    <button id="myButton">Submit</button>
12    <div id="output"></div>
13    <ul id="list"></ul>
14  </div>
15  <script src="js1_bug.js"></script>
16 </body>
17 </html>
```

①コードを書く前の準備をしよう

Vscodeにhtmlファイルを作って、コピーしたコードを貼り付けよう



The screenshot shows the VS Code interface with a project named 'JS_BUG'. The Explorer sidebar on the left lists several files: 'html1_answer.html', 'html1_bug.html', 'html2_answer.html', 'html2_bug.html', 'html3_answer.html', 'html3_bug.html', 'js1_answer.js', 'js1_bug.js', 'js2_answer.js', 'js2_bug.js', 'js3_answer.js', and 'js3_bug.js'. The main editor area displays the content of 'html1_bug.html', which is a standard HTML5 boilerplate with a form and a script reference.

```
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>Buggy Website</title>
7 </head>
8 <body>
9   <div>
10     <input id="myInput" type="text" placeholder="Enter text here">
11     <button id="myButton">Submit</button>
12     <div id="output"></div>
13     <ul id="list"></ul>
14   </div>
15   <script src="js1_bug.js"></script>
16 </body>
17 </html>
```

①コードを書く前の準備をしよう

他のwebページもコードをコピーしよう

目次

- ①コードを書く前の準備をしよう
- ②完成系を確認してみよう**
- ③コードの間違いを修正しよう
- ④答え合わせをしてみよう

②完成系を確認してみよう

このページでは完成系がどのようなになっているか確認できるよ

複雑な計算アプリ

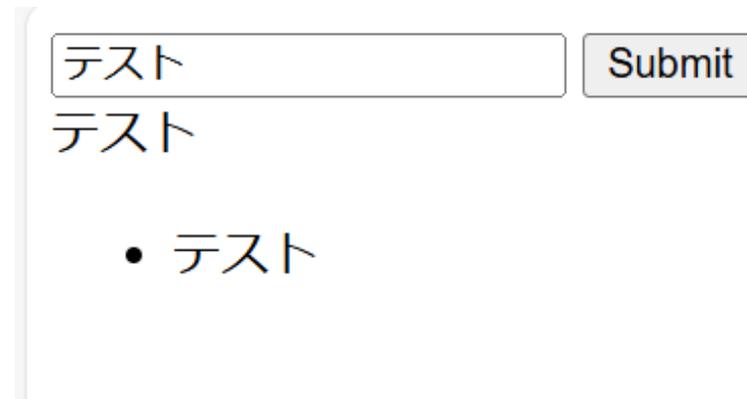
数値を入力してください: 操作を選択してください:

Result: 27

- Input: 3, Operation: square, Result: 9
- Input: 3, Operation: cube, Result: 27

②完成系を確認してみよう

まずは1つ目のwebサイトの完成写真だよ [Corrected Website](#)



テスト

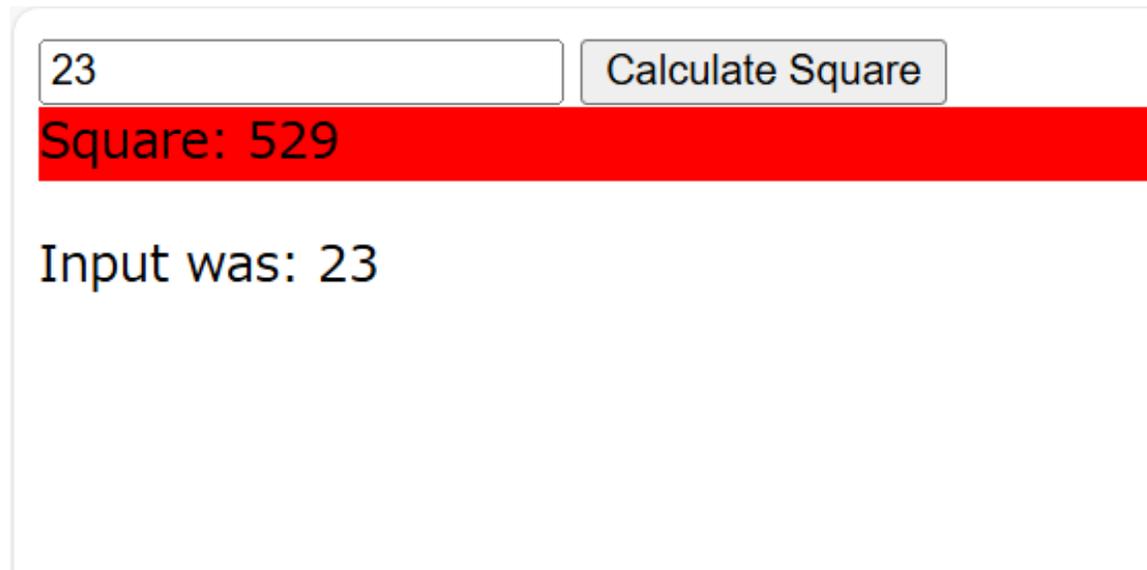
Submit

テスト

- テスト

②完成系を確認してみよう

2つ目のwebサイトの完成写真[Corrected Website](#)



A screenshot of a web application interface. At the top, there is a text input field containing the number '23' and a button labeled 'Calculate Square'. Below the input field, the text 'Square: 529' is displayed and highlighted with a red background. Underneath this, the text 'Input was: 23' is shown.

②完成系を確認してみよう

3つ目のwebサイトの完成写真修正済み複雑なウェブサイト

複雑な計算アプリ

数値を入力してください: 操作を選択してください:

Result: 27

- Input: 3, Operation: square, Result: 9
- Input: 3, Operation: cube, Result: 27

目次

- ①コードを書く前の準備をしよう
- ②完成系を確認してみよう
- ③コードの間違いを修正しよう
- ④答え合わせをしてみよう

③コードの間違いを修正しよう

このページではコードのどの部分にエラーがあるかヒントが書かれているよ

ヒントなしで済みたい場合はこの章はスキップしていいよ

③コードの間違いを修正しよう

コード1

- ・未定義変数の使用
- ・リストの初期化を忘れる
- ・条件式の代入ミス
- ・重複するイベントリスナー

③コードの間違いを修正しよう

コード2

- ・ 入力値の検証がない
- ・ ログの無制限な追加
- ・ 誤ったクリアロジック
- ・ スタイルのプロパティ名のタイポ

③コードの間違いを修正しよう

コード3

- ・ 階乗計算のオーバーフロー
- ・ 非数値入力の検証不足
- ・ 履歴リストの無制限成長
- ・ スタイリングの不整合
- ・ フォームのリセット時の不具合

目次

- ①コードを書く前の準備をしよう
- ②完成系を確認してみよう
- ③コードの間違いを修正しよう
- ④答え合わせをしてみよう

②完成系を確認してみよう

このページでは正解となるコードを書いているよ

```
JS js3_answer.js > ...
1
2 document.addEventListener('DOMContentLoaded', function() {
3   const form = document.getElementById('calcForm');
4   const resultDiv = document.getElementById('result');
5   const historyList = document.getElementById('history');
6
7   form.addEventListener('submit', function(event) {
8     event.preventDefault();
9
10    const input = document.getElementById('inputNumber').value;
11    const operation = document.getElementById('operation').value;
12
13    // Validate input
14    const number = parseInt(input, 10);
15    if (isNaN(number) || number < 0) {
16      resultDiv.innerHTML = "Invalid input. Please enter a positive number.";
17      resultDiv.style.color = "red";
18      return;
19    }
20
21    let result;
22    switch (operation) {
23      case 'factorial':
24        result = factorial(number);
25        break;
26      case 'square':
27        result = number ** 2;
28        break;
29      case 'cube':
```

```
30        result = number ** 3;
31        break;
32      default:
33        resultDiv.innerHTML = "Unknown operation.";
34        return;
35    }
36
37    // Display result
38    resultDiv.innerHTML = `Result: ${result}`;
39    resultDiv.style.color = "green";
40
41    // Update history with a cap of 10 entries
42    const historyItem = document.createElement('li');
43    historyItem.textContent = `Input: ${number}, Operation: ${operation}, Result: ${result}`;
44    historyList.appendChild(historyItem);
45    if (historyList.childElementCount > 10) {
46      historyList.removeChild(historyList.firstChild);
47    }
48  });
49
50  // Clear history and reset result on reset
51  form.addEventListener('reset', function() {
52    historyList.innerHTML = '';
53    resultDiv.innerHTML = "History cleared.";
54    resultDiv.style.color = "black";
55  });
```

②完成系を確認してみよう

コード1

```
1 document.addEventListener('DOMContentLoaded', function() {
2   const button = document.getElementById('myButton');
3
4   button.addEventListener('click', function() {
5     const input = document.getElementById('myInput').value;
6     const output = document.getElementById('output');
7     const list = document.getElementById('list');
8
9     output.innerText = input;
10
11    list.innerHTML = '';
12    const listItem = document.createElement('li');
13    listItem.textContent = input;
14    list.appendChild(listItem);
15
16    if (input === "error") {
17      output.style.color = "red";
18    } else {
19      output.style.color = "black";
20    }
21  });
22
23  console.log("Event listeners properly configured.");
24 });
```

②完成系を確認してみよう

コード2

```
JS js2_answer.js > document.addEventListener('DOMContentLoaded') callback > button.addEventListener
1 document.addEventListener('DOMContentLoaded', function() {
2   const button = document.getElementById('calculateButton');
3   const logDiv = document.getElementById('log');
4
5   button.addEventListener('click', function() {
6     const input = document.getElementById('textInput').value;
7     const resultDiv = document.getElementById('result');
8
9     const number = parseInt(input);
10    if (isNaN(number)) {
11      resultDiv.innerHTML = "Please enter a valid number.";
12      return;
13    }
14
15    resultDiv.innerHTML = `Square: ${number * number}`;
16
17    const logEntry = document.createElement('p');
18    logEntry.textContent = `Input was: ${input}`;
19    logDiv.appendChild(logEntry);
20    if (logDiv.childElementCount > 10) {
21      logDiv.removeChild(logDiv.firstChild);
22    }
23
24    if (input.toLowerCase() === "clear") {
25      while (logDiv.firstChild) {
26        logDiv.removeChild(logDiv.firstChild);
27      }
28      resultDiv.innerHTML = "Log cleared.";
29      return;
30    }
31
32    resultDiv.style.backgroundColor = "yellow";
33    if (number > 100) {
34      resultDiv.style.color = "blue";
35    } else {
36      resultDiv.style.backgroundColor = "red";
37    }
38  });
39 });
40
```

②完成系を確認してみよう

コード3その1

```
JS js3_answer.js > ...
1
2 document.addEventListener('DOMContentLoaded', function() {
3   const form = document.getElementById('calcForm');
4   const resultDiv = document.getElementById('result');
5   const historyList = document.getElementById('history');
6
7   form.addEventListener('submit', function(event) {
8     event.preventDefault();
9
10    const input = document.getElementById('inputNumber').value;
11    const operation = document.getElementById('operation').value;
12
13    // Validate input
14    const number = parseInt(input, 10);
15    if (isNaN(number) || number < 0) {
16      resultDiv.innerHTML = "Invalid input. Please enter a positive number.";
17      resultDiv.style.color = "red";
18      return;
19    }
20
21    let result;
22    switch (operation) {
23      case 'factorial':
24        result = factorial(number);
25        break;
26      case 'square':
27        result = number ** 2;
28        break;
29      case 'cube':
```

```
30        result = number ** 3;
31        break;
32      default:
33        resultDiv.innerHTML = "Unknown operation.";
34        return;
35    }
36
37    // Display result
38    resultDiv.innerHTML = `Result: ${result}`;
39    resultDiv.style.color = "green";
40
41    // Update history with a cap of 10 entries
42    const historyItem = document.createElement('li');
43    historyItem.textContent = `Input: ${number}, Operation: ${operation}, Result: ${result}`;
44    historyList.appendChild(historyItem);
45    if (historyList.childElementCount > 10) {
46      historyList.removeChild(historyList.firstChild);
47    }
48  });
49
50  // Clear history and reset result on reset
51  form.addEventListener('reset', function() {
52    historyList.innerHTML = '';
53    resultDiv.innerHTML = "History cleared.";
54    resultDiv.style.color = "black";
55  });
```

②完成系を確認してみよう

コード3その2

```
57 // Factorial function to handle large numbers safely
58 function factorial(num) {
59     if (num === 0 || num === 1) return 1;
60     let result = 1;
61     for (let i = 2; i <= num; i++) {
62         result *= i;
63         // Prevent overflow
64         if (result > Number.MAX_SAFE_INTEGER) {
65             return "Value too large to compute.";
66         }
67     }
68     return result;
69 }
70 });
71
```

お疲れさまでした

テキストは終了です。
あとは自分なりにアレンジを付け加えていこう！