



18章 イベント処理でフォームを操作しよう

イベント処理の活用方法を解説します。

🕒 120分 🏆 - 📖 読了

18.1 本章の目標

本章では以下を目標にして学習します。

- イベント処理でフォームを操作する方法を知ること

本章では、イベント処理を使ってフォームを操作する方法を学びます。具体的には、以下3つのイベント処理を順番に作成します。

1. フォームに入力された文字数をカウントする
2. 選択されたラジオボタンの値を取得する
3. 選択されたチェックボックスの値を取得する

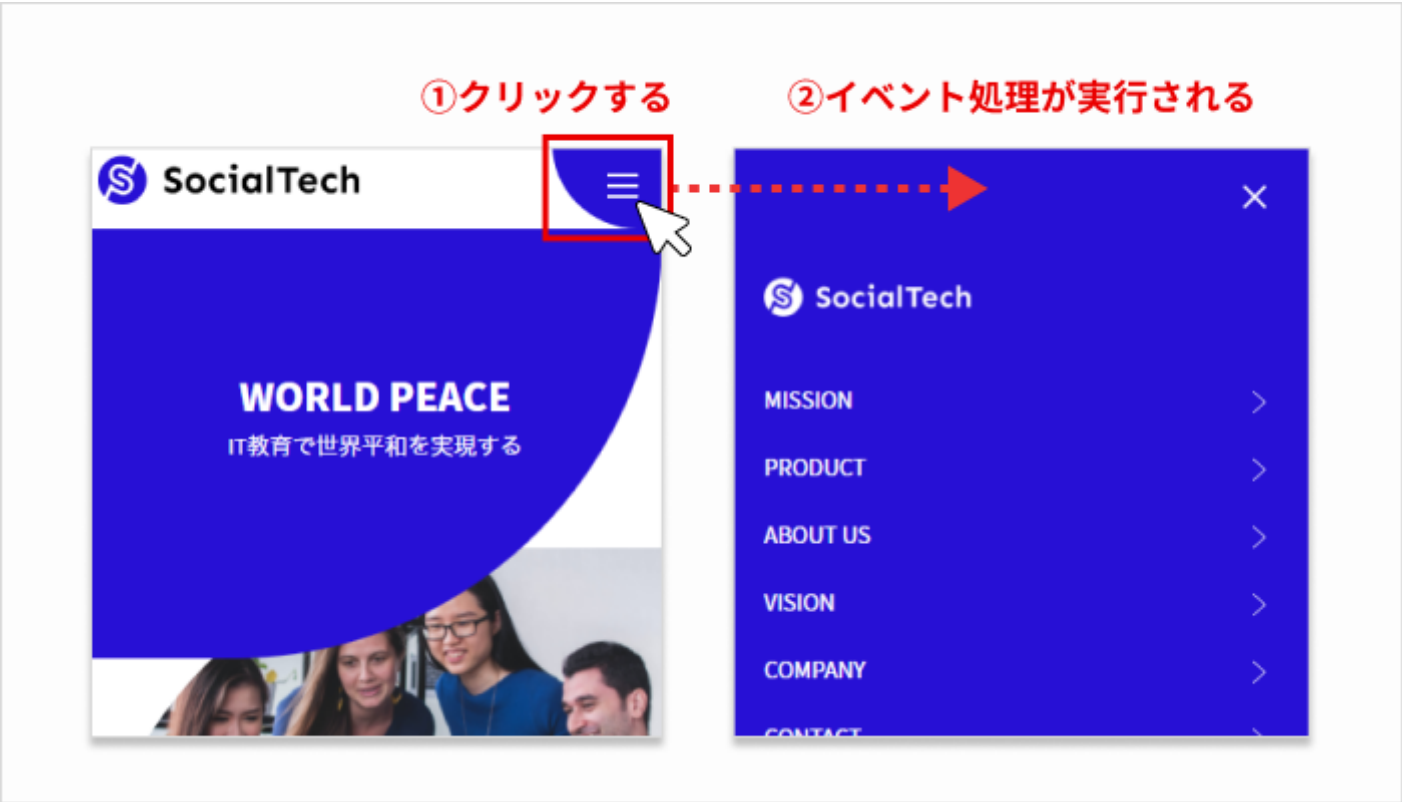
これらの処理は実際のWeb制作でもよく使われるので、一緒に練習していきましょう。

18.2 イベント処理について復習しよう

まずは前章で学んだイベント処理について復習しておきましょう。

イベント処理とは、**ユーザーの行動に合わせてDOM操作を行うこと**でした。

+ 質問する



またイベント処理は、主に `addEventListener()` メソッドを使って以下のように作成することも学びました。

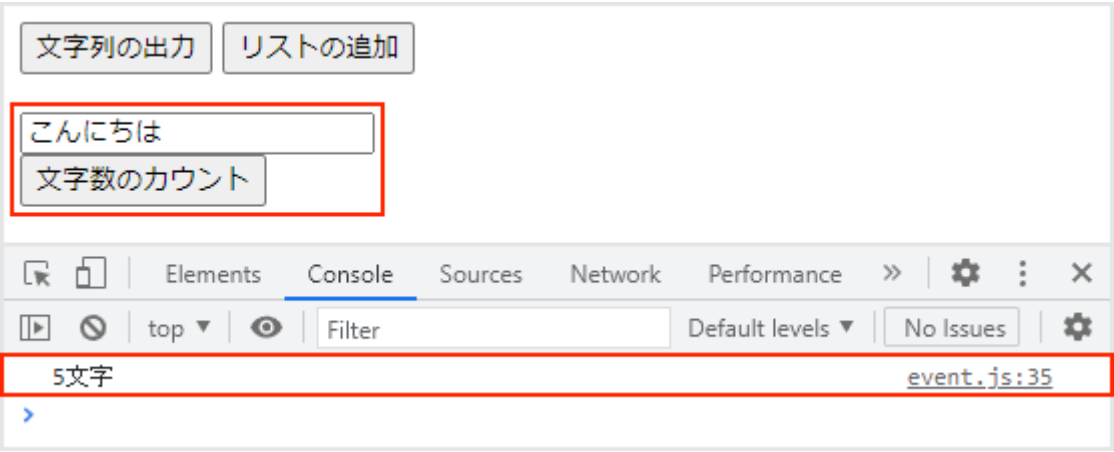
```
1 HTML要素.addEventListener('イベントの種類', () => {
2   イベント処理
3 });
4
```

イベントの種類	処理が実行されるタイミング
click	クリックしたとき（マウスボタンを押して離れたとき）
mousedown	マウスボタンを押したとき
mouseup	マウスボタンを離れたとき
mousemove	マウスカーソルを移動したとき
keydown	キーボードのキーを押したとき
keyup	キーボードのキーを離れたとき
submit	フォームを送信したとき
focus	HTML要素にフォーカスしたとき
scroll	画面をスクロールしたとき

18.3 フォームに入力された文字数をカウントしてみよう

1つ目は、フォームに入力された文字数をカウントするイベント処理を作成します。実際のWeb制作で使われる場面としては、「文字数制限を設け、指定した文字数を超えた場合に警告メッセージを表示する」などが考えられます。

今回は、以下のようにシンプルに文字数を出力する処理を作成します。



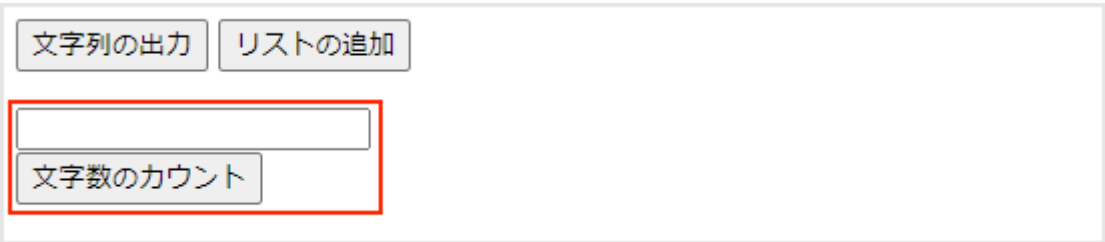
HTMLファイルを編集する

まずは event.html を以下のように編集し、テキストボックスを作成しましょう。

event.html

```
1 <!--===== 前略 =====>
2
3 <button id="add-btn">リストの追加</button>
4 <ul id="parent-list"></ul>
5
6 + <form name="textForm">
7 +   <input type="text" name="textBox" />
8 + </form>
9 + <button id="count-btn">文字数のカウント</button>
10
11 <script src="js/event.js"></script>
12 </body>
13
14 </html>
15
```

表示結果



イベント処理の内容

続いて、JavaScriptで以下のイベント処理を作成します。処理が実行されるタイミングは「文字数のカウント」ボタンをクリックしたときです。

1. テキストボックスに入力された文字列を取得する
2. 取得した文字列の文字数を出力する

1. テキストボックスに入力された文字列を取得する

1つ目ですが、JavaScriptからテキストボックスの値を取得するには、まず form 要素を取得する必要があります。











これまでのように `getElementById()` メソッドなどを使って取得することもできますが、フォームの場合は `name` 属性の値を使うことで簡単に取得できます。

JSファイル（見本）

```
1 // textFormというname属性を持つフォームを取得する
2 document.forms.textForm;
3
```

また、続けてテキストボックスの `name` 属性の値を記述すれば、テキストボックス（ `input` 要素）も取得できます。

JSファイル（見本）

```
1 // textBoxというname属性を持つテキストボックスを取得する
2 document.forms.textForm.textBox;
3
```

入力された値を取得するには、続けて `value` プロパティを記述します。

JSファイル（見本）

```
1 // textBoxというname属性を持つテキストボックスの値を取得する
2 document.forms.textForm.textBox.value;
3
```

なお、先ほど `event.html` を編集した際に、「なぜ `name` 属性の値がキャメルケースなのか」と疑問を抱いた方もいると思いますが、その理由はここで使うためです（キャメルケースについては4章を参照）。もし `id` のようにハイフン `-` を使ってしまうと、JavaScriptでエラーが発生してしまいます。

JSファイル（見本）

```
1 // ハイフンを使うとエラーが発生する
2 document.forms.text-form;
3
```

2. 取得した文字列の文字数を出力する

取得した文字列の文字数を出力するには、 `length` プロパティを使います。復習になりますが、 `length` プロパティは文字列や配列の長さを取得するプロパティです。

JSファイルを編集する

では `event.js` を以下のように編集してください。

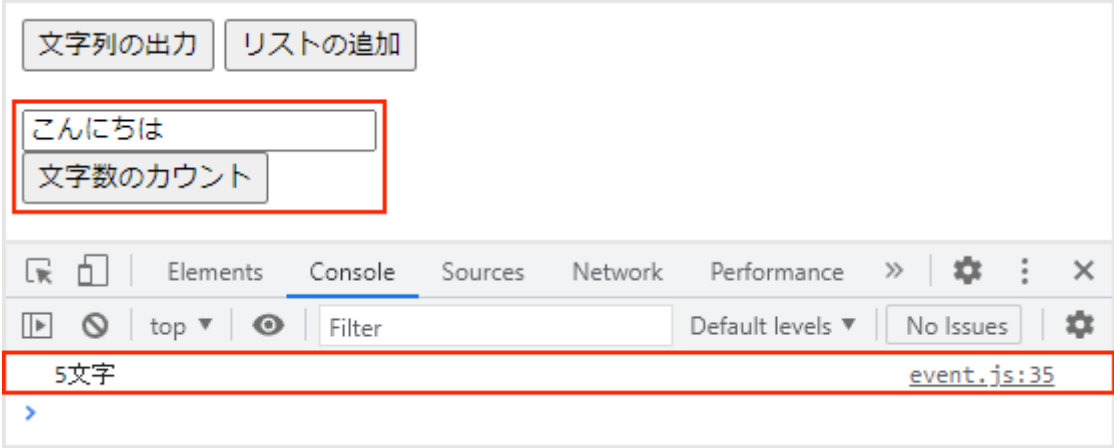
event.js



```
1 //===== 前略 =====
2
3 // add-btnというidを持つHTML要素を取得し、定数に代入する
4 const addBtn = document.getElementById('add-btn');
5 // parent-listというidを持つHTML要素を取得し、定数に代入する
6 const parentList = document.getElementById('parent-list');
7
8 // HTML要素がクリックされたときにイベント処理を実行する
9 addBtn.addEventListener('click', () => {
10   // li要素を新しく作成する
11   const childList = document.createElement('li');
12
13   // 作成したli要素に「リストが追加されました」というテキストを追加する
14   childList.textContent = 'リストが追加されました';
15
16   // 作成したli要素をul要素の子要素として末尾に追加する
17   parentList.appendChild(childList);
18 });
19
20 + // count-btnというidを持つHTML要素を取得し、定数に代入する
21 + const countBtn = document.getElementById('count-btn');
22 +
23 + // HTML要素がクリックされたときにイベント処理を実行する
24 + countBtn.addEventListener('click', () => {
25 +   // テキストボックスに入力された文字列を取得する
26 +   const text = document.forms.textForm.textBox.value;
27 +
28 +   // 取得した文字列の文字数を出力する
29 +   console.log(text.length + '文字');
30 + });
31
```

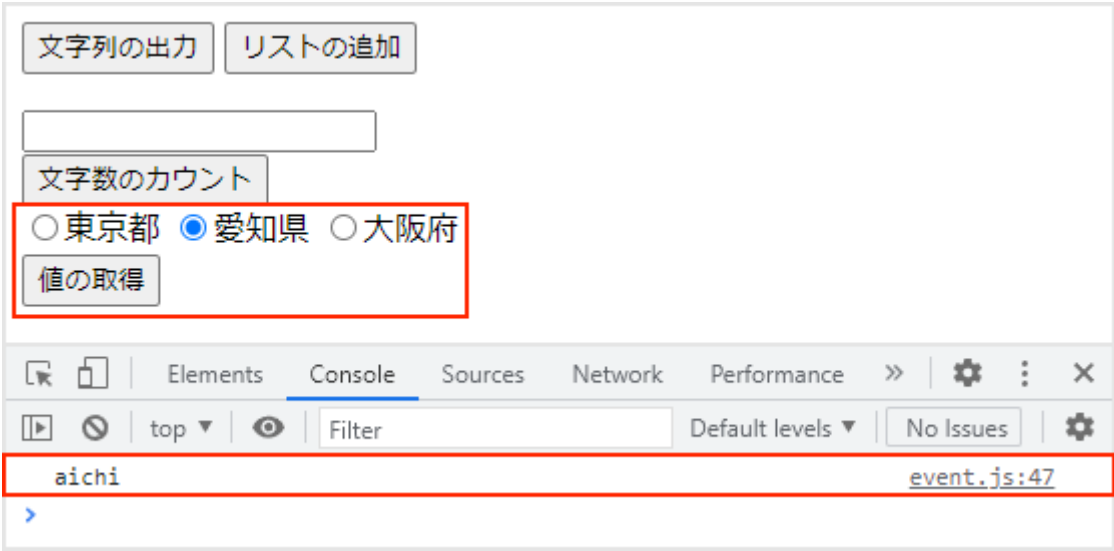
実行結果を確認する

では event.html をブラウザで開き、テキストボックスに好きな文字列を入力してみましょう。以下のように、入力した文字列の文字数がコンソールに出力されればOKです。



18.4. 選択されたラジオボタンの値を取得してみよう

2つ目は、選択されたラジオボタンの値を取得するイベント処理を作成します。



HTMLファイルを編集する

まずは event.html を以下のように編集し、ラジオボタンを作成しましょう。

event.html

```
1  <!--===== 前略 =====>
2
3  <form name="textForm">
4    <input type="text" name="textBox" />
5  </form>
6  <button id="count-btn">文字数のカウント</button>
7
8 +  <form name="areaForm">
9 +    <input type="radio" name="area" value="tokyo" />東京都
10 +    <input type="radio" name="area" value="aichi" />愛知県
11 +    <input type="radio" name="area" value="osaka" />大阪府
12 +  </form>
13 +  <button id="area-btn">値の取得</button>
14
15  <script src="js/event.js"></script>
16 </body>
17
18 </html>
19
```

表示結果



イベント処理の内容

続いて、JavaScriptで以下のイベント処理を作成します。処理が実行されるタイミングは「値の取得」ボタンをクリックしたときです。

1. 選択されたラジオボタンの値を取得する
2. 取得した値を出力する





1. 選択されたラジオボタンの値を取得する

先ほどテキストボックスの値を取得したときと同様に、ラジオボタンの値も取得できます。

JSファイル（見本）

```
1 document.forms.form要素のname属性の値.input要素のname属性の値.value;
2
```

2. 取得した値を出力する

取得した値を出力するには `console.log()` を記述するだけですが、取得した値は一度変数や定数に代入しておくことで、コードがスッキリして見やすくなります。

JSファイル（見本）

```
1 // 1. 変数や定数に代入しないパターン
2 // 引数が長いので見づらい
3 console.log(document.forms.form要素のname属性の値.input要素のname属性の値.value);
4
5 // 2. 変数や定数に代入しておくパターン
6 const 定数名 = document.forms.form要素のname属性の値.input要素のname属性の値.value;
7 // 引数が短くスッキリしていて見やすい
8 console.log(定数名);
9
```

JSファイルを編集する

上記を参考に、まずは自分で `event.js` を編集してみてください。編集したら、以下のコードと見比べて答え合わせをしましょう。

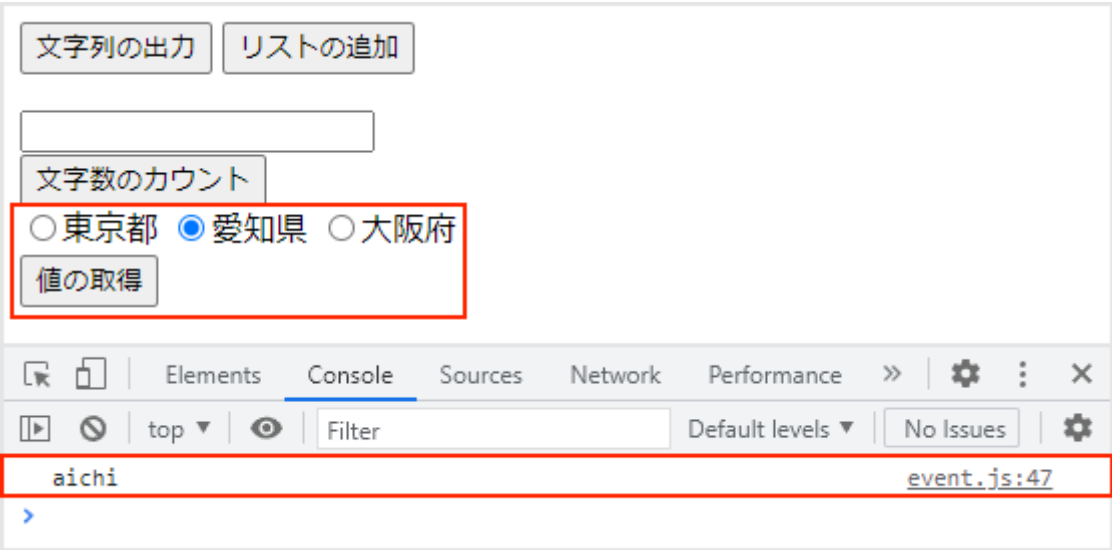
event.js



```
1 //===== 前略 =====
2
3 // count-btnというidを持つHTML要素を取得し、定数に代入する
4 const countBtn = document.getElementById('count-btn');
5
6 // HTML要素がクリックされたときにイベント処理を実行する
7 countBtn.addEventListener('click', () => {
8   // テキストボックスに入力された文字列を取得する
9   const text = document.forms.textForm.textBox.value;
10
11   // 取得した文字列の文字数を出力する
12   console.log(text.length + '文字');
13 });
14
15 + // area-btnというidを持つHTML要素を取得し、定数に代入する
16 + const areaBtn = document.getElementById('area-btn');
17 +
18 + // HTML要素がクリックされたときにイベント処理を実行する
19 + areaBtn.addEventListener('click', () => {
20 +   // 選択されたラジオボタンの値を取得する
21 +   const area = document.forms.areaForm.area.value;
22 +
23 +   // 取得した値を出力する
24 +   console.log(area);
25 + });
26
```

実行結果を確認する

では event.html をブラウザで開き、ラジオボタンをどれか1つ選択してみましょう。以下のように、選択したラジオボタンの値がコンソールに出力されていればOKです。



18.5. 選択されたチェックボックスの値を取得してみよう

最後に、選択されたチェックボックスの値を取得するイベント処理を作成します。



文字列の出力

リストの追加

文字数のカウント

○東京都

○愛知県

○大阪府

値の取得

利用したことのあるOSを選択してください

☒Windows

☒macOS

☐Linux

☒iOS

☐Android

値の取得

Elements

Console

Sources

Network

Performance

top

Filter

Default levels

No Issues

windows

event.js:61

macos

event.js:61

ios

event.js:61

>

HTMLファイルを編集する

まずは event.html を以下のように編集し、チェックボックスを作成しましょう。

event.html

```
1  <!--===== 前略 =====>
2
3  <form name="areaForm">
4    <input type="radio" name="area" value="tokyo" />東京都
5    <input type="radio" name="area" value="aichi" />愛知県
6    <input type="radio" name="area" value="osaka" />大阪府
7  </form>
8  <button id="area-btn">値の取得</button>
9
10 + <form name="osForm">
11 +   <label>利用したことのあるOSを選択してください</label><br />
12 +   <input type="checkbox" name="os" value="windows" />Windows
13 +   <input type="checkbox" name="os" value="macos" />macOS
14 +   <input type="checkbox" name="os" value="linux" />Linux
15 +   <input type="checkbox" name="os" value="ios" />iOS
16 +   <input type="checkbox" name="os" value="android" />Android
17 + </form>
18 + <button id="os-btn">値の取得</button>
19
20   <script src="js/event.js"></script>
21 </body>
22
23 </html>
24
```

表示結果



文字列の出力

リストの追加

文字数のカウント

○東京都

○愛知県

○大阪府

値の取得

利用したことのあるOSを選択してください

☐ Windows

☐ macOS

☐ Linux

☐ iOS

☐ Android

値の取得

イベント処理の内容

続いて、JavaScriptで以下のイベント処理を作成します。処理が実行されるタイミングは「値の取得」ボタンをクリックしたときです。

- すべてのチェックボックスを配列風のデータで取得する
- 繰り返し処理でチェックボックスを1つずつ取り出し、もし選択されていれば値を出力する

1. すべてのチェックボックスを配列風のデータで取得する

まずはすべてのチェックボックスを配列風のデータで取得します。値ではなく、チェックボックスを形作る `input` 要素そのものを取得します。

JSファイル（見本）

- ```
1 document.forms.form要素のname属性の値.input要素のname属性の値;
2
```

### 2. 繰り返し処理でチェックボックスを1つずつ取り出し、もし選択されていれば値を出力する

チェックボックスがラジオボタンと異なるのは、「複数の項目を選択できる」という点です。

よって、選択されたすべての値を出力するために、`for`文と`if`文を組み合わせ使います。以下のような流れです。

- `for`文を使い、取得したチェックボックス（`input` 要素）を1つずつ取り出す
- `for`文の中で`if`文を使い、各チェックボックスがもし選択されていれば値を出力する

`for`文は8章、`if`文は6章で学びました。書き方を復習しておきましょう。

JSファイル（見本）

- ```
1 for (カウンタ変数の初期値; 条件式; カウンタ変数の増減値) {  
2     カウンタ変数が決まった値になるまで繰り返す処理  
3 }  
4
```

JSファイル（見本）





```
1  if (条件式) {
2      条件が成り立つときの処理
3  }
4
```

チェックボックスの場合、ユーザーが選択する（チェックを入れる）たびに checked プロパティが true になる仕組みになっています。

この仕組みをif文の条件式に利用し、 チェックボックス.checked と記述すれば、チェックボックスが選択されていた場合にのみ処理を行うことができます。

なお、for文の条件式には length プロパティを使い、チェックボックスの数だけ繰り返し処理を行います。

JSファイルを編集する

今回はfor文やif文を組み合わせる使うので、難易度が高いです。よって、難しいと感じたら以下のコードを見ながら編集してOKです。可能であれば、まずは自分で event.js を編集してみましょう。

```
event.js

1  //===== 前略 =====
2
3  // area-btnというidを持つHTML要素を取得し、定数に代入する
4  const areaBtn = document.getElementById('area-btn');
5
6  // HTML要素がクリックされたときにイベント処理を実行する
7  areaBtn.addEventListener('click', () => {
8      // 選択されたラジオボタンの値を取得する
9      const area = document.forms.areaForm.area.value;
10
11      // 取得した値を出力する
12      console.log(area);
13  });
14
15 + // os-btnというidを持つHTML要素を取得し、定数に代入する
16 + const osBtn = document.getElementById('os-btn');
17 +
18 + // HTML要素がクリックされたときにイベント処理を実行する
19 + osBtn.addEventListener('click', () => {
20 +     // すべてのチェックボックスを配列風のデータで取得する
21 +     const items = document.forms.osForm.os;
22 +
23 +     // 繰り返し処理でチェックボックスを1つずつ取り出し、もし選択されていれば値を出力する
24 +     for (let i = 0; i < items.length; i++) {
25 +         if (items[i].checked) {
26 +             console.log(items[i].value);
27 +         }
28 +     }
29 + });
30
```

実行結果を確認する

では event.html をブラウザで開き、チェックボックスを複数選択してみましょう。以下のように、選択したチェックボックスの値がコンソールに出力されていればOKです。なお、チェックボックスを選択しなかった場合はボタンを押しても何も起こりません。





文字列の出力

リストの追加

文字数のカウント

○東京都

○愛知県

○大阪府

値の取得

利用したことのあるOSを選択してください

☐ Windows

☐ macOS

☐ Linux

☐ iOS

☐ Android

値の取得

まとめ

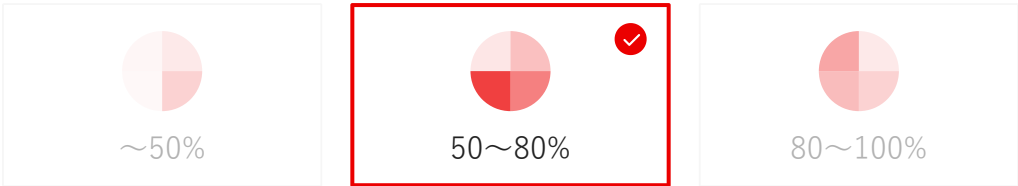
本章では以下の内容を学習しました。

- フォームの場合は name 属性の値を使うことで、簡単にHTML要素を取得できる
- フォームの値を取得するには value プロパティを使う
- 文字列の文字数を取得するには length プロパティを使う
- チェックボックスに対して checked プロパティを使うことで、選択されているかどうかを true または false で取得できる

次章では、イベント処理でボタンの表示・非表示を切り替えます。

理解度を選択して次に進みましょう

ボタンを押していただくと次の章に進むことができます



最後に確認テストを行いましょう

下のボタンを押すとテストが始まります。

教材をみなおす

テストをはじめる

前に戻る

22 / 26 ページ

次に進む

< 一覧に戻る

改善点のご指摘、誤字脱字、その他ご要望はこちらからご連絡ください。



