

🏠

🕒

📖

📄

🔍

✎

🔊

教材

🔍 検索

所属チーム ▼ 🔔¹ 👤

本文 目次 質問一覧 13件

ホーム 教材 JavaScriptの基礎を学ぼう イベント処理でボタンの表示・非表示を切り替えよう

19章 イベント処理でボタンの表示・非表示を切り替えよう

イベント処理の活用方法を解説します。

🕒120分 🏆 - 読了

19.1 本章の目標

本章では以下を目標にして学習します。

■ 画面のスクロール量に合わせてイベント処理を行う方法を知ること

以下のように、ある程度下にスクロールすると「TOPに戻る」ボタンが表示されるWebサイトを目にしたことはないでしょうか。

1ページ目

このようなボタンも、JavaScriptのイベント処理で作成することができます。

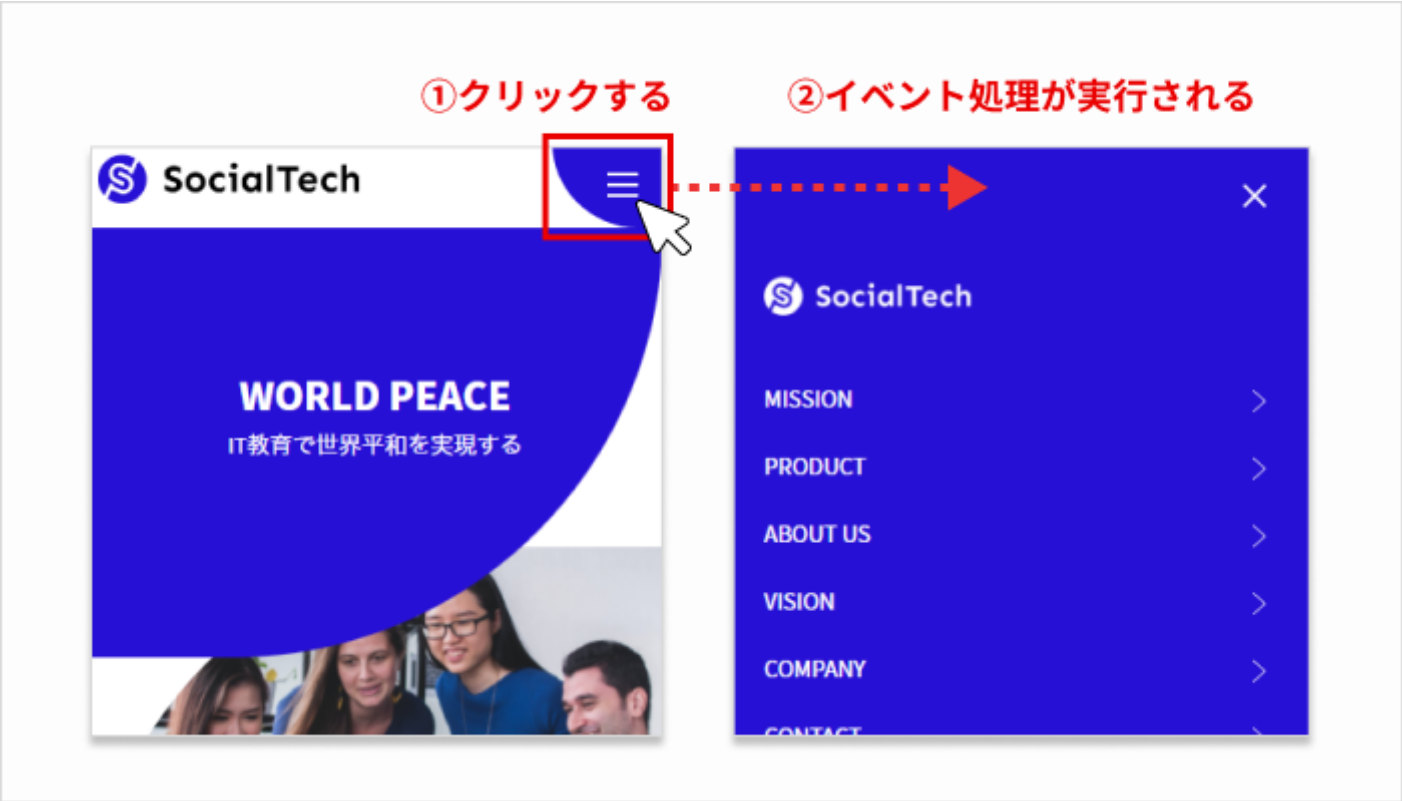
3章にわたるイベント処理の学習も本章で最後なので、頑張ってください。

19.2 イベント処理について復習しよう

まずはイベント処理について復習しておきましょう。

+ 質問する

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。




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

19.3 スクロール量に合わせてボタンの表示・非表示を切り替えよう

本章では以下のように、ある程度下にスクロールしたときに、非表示にしていた「TOPに戻る」ボタンを表示させるイベント処理を作成します。



🏠

🕒

📖

📄

🔍

✎

🔊

>

1ページ目

HTMLファイルを作成・編集する

まずはVisual Studio Codeを開き、 javascript-basic フォルダ直下（ index.html と同じ階層）に scroll1.html というファイルを作成してください。

続いて scroll1.html を以下のように編集し、スクロール用の div 要素5つと「TOPに戻る」ボタンを作成しましょう。なお、 link タグ内で読み込んでいる scroll1.css や script タグ内で読み込んでいる scroll1.js は、このあと作成します。

```
scroll1.html

1 + <!DOCTYPE html>
2 + <html lang="ja">
3 +
4 + <head>
5 +   <meta charset="UTF-8">
6 +   <title>イベント処理（スクロール）</title>
7 +   <link rel="stylesheet" href="css/scroll1.css">
8 + </head>
9 +
10 + <body>
11 +   <div class="blue-screen">1ページ目</div>
12 +   <div class="red-screen">2ページ目</div>
13 +   <div class="blue-screen">3ページ目</div>
14 +   <div class="red-screen">4ページ目</div>
15 +   <div class="blue-screen">5ページ目</div>
16 +   <a id="back-btn" href="#">TOPに戻る</a>
17 +
18 +   <script src="js/scroll1.js"></script>
19 + </body>
20 +
21 + </html>
22
```

表示結果

https://terakoya.sejuku.net/programs/60/chapters/685

3/10

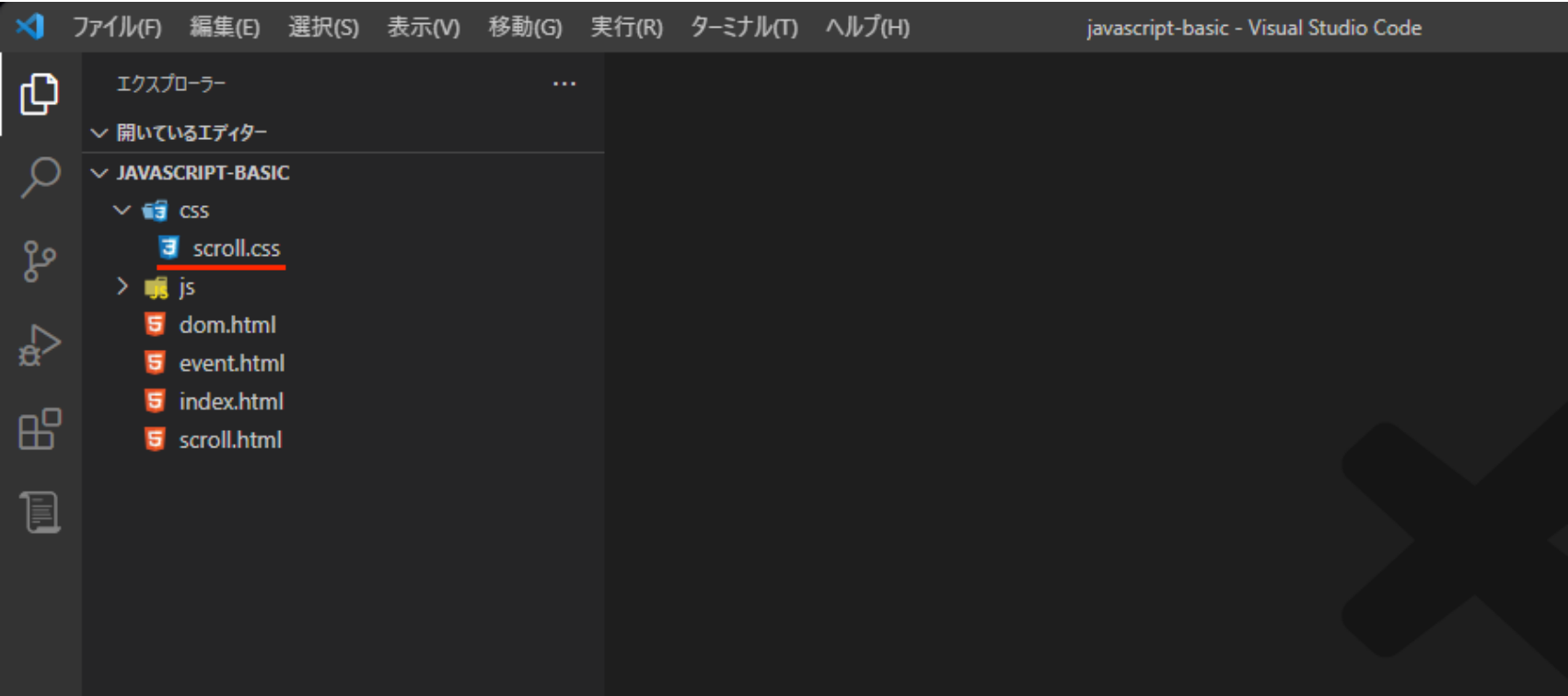


1ページ目
2ページ目
3ページ目
4ページ目
5ページ目
[TOPに戻る](#)

CSSファイルを作成・編集する

今回はdiv要素の背景や「TOPに戻る」 ボタンを装飾するため、CSSを記述します。

そこで、 javascript-basic フォルダ直下（ index.html と同じ階層）に、新しく css という名前のフォルダを作成してください。続いて作成した css フォルダ内に、新しく scroll.css というファイルを作成しましょう。



CSSファイルを編集する際のポイントは以下のとおりです。

- 色違いの背景を交互に表示し、スクロール量が視覚的にわかるようにする
- div 要素に height: 100vh; を指定し、1画面分の高さにする（スクロールできる分の高さを確保するため）
- 「TOPに戻る」 ボタンを非表示にしておく（スクロール量に合わせてJavaScriptから表示させるため）
- 「TOPに戻る」 ボタンに position: fixed; を指定し、表示位置を固定する

それでは scroll.css を以下のように編集してください。

scroll.css



```
1 + .blue-screen {
2 +   height: 100vh;
3 +   color: #fff;
4 +   font-size: 5em;
5 +   text-align: center;
6 +   background-color: #88f;
7 + }
8 +
9 + .red-screen {
10 +   height: 100vh;
11 +   color: #fff;
12 +   font-size: 5em;
13 +   text-align: center;
14 +   background-color: #f88;
15 + }
16 +
17 + #back-btn {
18 +   display: none;
19 +
20 +   padding: 40px 10px;
21 +   background-color: #fff;
22 +   color: #000;
23 +   text-decoration: none;
24 +   border-radius: 50%;
25 +
26 +   position: fixed;
27 +   bottom: 10px;
28 +   right: 10px;
29 + }
30
```

表示結果



イベント処理の内容

続いて、JavaScriptで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画面をスクロールしたときなので、 `addEventListener()` メソッドの1つ目の引数には `'scroll'` を指定します。





JSファイル（見本）

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

なお、今回イベント処理の実行対象となるのは特定のHTML要素ではなく、**画面全体**です。よって、上記の「HTML要素」の部分には**windowオブジェクト**を指定します。 window オブジェクトについて復習したい方は15章を参照してください。

JSファイル（見本）

```
1 window.addEventListener('scroll', () => {
2   イベント処理
3 });
4
```

イベント処理の内容は以下のとおりです。

1. 画面のスクロール量を取得する
2. スクロール量が 300px 以上であれば「TOPに戻る」ボタンを表示する
3. スクロール量が 300px 未満であれば「TOPに戻る」ボタンを非表示にする

なお、今回はスクロール量の分岐点を 300px にしていますが、「必ず 300px にしなければならない」という決まりはありません。Web 制作の現場では、実際にブラウザで動作を確認しながら適切な値を探っていくことが多いです。

それでは順番に解説します。

1. 画面のスクロール量を取得する

今回作成するイベント処理をもう一度確認しましょう。

- ある程度下にスクロールしたときに、非表示にしていた「TOPに戻る」ボタンを表示させる

では、「ある程度下にスクロールしたかどうか」をどのように判断すればよいのでしょうか。

そのようなときに使うのが document.scrollingElement.scrollTop プロパティです。 document.scrollingElement.scrollTop プロパティを使えば、スクロール量を px （ピクセル）数で取得できます。

JSファイル（見本）

```
1 // 画面のスクロール量をpx（ピクセル）数で取得する
2 const scrollValue = document.scrollingElement.scrollTop;
3
```

なお、イベント処理はスクロールする度に実行されるので、リアルタイムで画面のスクロール量を取得できます。

2. スクロール量が300px以上であれば「TOPに戻る」ボタンを表示する



取得したスクロール量を利用し、if文で「スクロール量が 300px 以上であれば」という条件式を作ります。

JSファイル（見本）

```
1 // 画面のスクロール量が300px以上であれば、処理を実行する
2 if (scrollValue >= 300) {
3   条件が成り立つときの処理
4 }
5
```

なお、「TOPに戻る」ボタンを表示するには、JavaScriptからCSSのスタイルを変更する必要があります。JavaScriptからCSSのスタイルを変更するには、以下のように記述します。

JSファイル（見本）

```
1 HTML要素.style.CSSプロパティ = 変更後の値;
2
```

例えば「TOPに戻る」ボタンの display プロパティの値を inline に変更するには、以下のように記述します。「TOPに戻る」ボタンの display プロパティの初期値は none なので（ scroll.css の #back-btn を参照）、 inline に変更することで「TOPに戻る」ボタンを表示できます。

JSファイル（見本）

```
1 // back-btnというidを持つHTML要素を取得し、定数に代入する
2 const backBtn = document.getElementById('back-btn');
3
4 // 取得したHTML要素のdiplayプロパティの値をinlineに変更する
5 backBtn.style.display = 'inline';
6
```

display プロパティの値が none 以外であれば block や inline-block でも表示はされますが、 a 要素（「TOPに戻る」ボタン）の display プロパティの初期値が inline なので、ここでは inline に変更しています。

ではここまでの内容をまとめます。「スクロール量が 300px 以上であれば『TOPに戻る』ボタンを表示する」という条件分岐を行うには、以下のように記述します。

JSファイル（見本）

```
1 // back-btnというidを持つHTML要素を取得し、定数に代入する
2 const backBtn = document.getElementById('back-btn');
3
4 // 画面のスクロール量をpx（ピクセル）数で取得する
5 const scrollValue = document.scrollingElement.scrollTop;
6
7 // 画面のスクロール量が300px以上であれば、「TOPに戻る」ボタンを表示する
8 if (scrollValue >= 300) {
9   backBtn.style.display = 'inline';
10 }
11
```



3. スクロール量が300px未満であれば「TOPに戻る」 ボタンを非表示にする

このままでは、一度スクロール量が 300px 以上になると、「TOPに戻る」ボタンが表示されたままになってしまいます。つまり、表示後にスクロール量を 300px 未満にしても、非表示になりません。

そこで、以下のようにelse文を使って非表示にする処理も追記します。

JSファイル（見本）

```
1 // back-btnというidを持つHTML要素を取得し、定数に代入する
2 const backBtn = document.getElementById('back-btn');
3
4 // 画面のスクロール量をpx（ピクセル）数で取得する
5 const scrollValue = document.scrollingElement.scrollTop;
6
7 // 画面のスクロール量が300px以上であれば、「TOPに戻る」ボタンを表示する
8 if (scrollValue >= 300) {
9   backBtn.style.display = 'inline';
10 }
11 // 画面のスクロール量がそれ以外（300px未満）であれば、「TOPに戻る」ボタンを非表示にする
12 else {
13   backBtn.style.display = 'none';
14 }
15
```

JSファイルを作成・編集する

では js フォルダ内に新しく scroll1.js というファイルを作成してください。 scroll1.js を作成したら、以下のように編集しましょう。

scroll1.js

```
1 + // back-btnというidを持つHTML要素を取得し、定数に代入する
2 + const backBtn = document.getElementById('back-btn');
3 +
4 + // 画面がスクロールされたときにイベント処理を実行する
5 + window.addEventListener('scroll', () => {
6 +   // 画面のスクロール量をpx（ピクセル）数で取得する
7 +   const scrollValue = document.scrollingElement.scrollTop;
8 +
9 +   // 画面のスクロール量が300px以上であれば、「TOPに戻る」ボタンを表示する
10 +   if (scrollValue >= 300) {
11 +     backBtn.style.display = 'inline';
12 +   }
13 +   // 画面のスクロール量がそれ以外（300px未満）であれば、「TOPに戻る」ボタンを非表示にする
14 +   else {
15 +     backBtn.style.display = 'none';
16 +   }
17 + });
18
```

実行結果を確認する

では scroll1.html をブラウザで開き、画面をスクロールしてみましょう。以下のように、ある程度下にスクロールしたときに「TOPに戻る」ボタンが表示されればOKです。



また、上にスクロールしたときに「TOPに戻る」ボタンが非表示になることも確認してください。



まとめ

本章では以下の内容を学習しました。

- イベントの種類に 'scroll' を指定する場合、`addEventListener()` メソッドの実行対象は `window` オブジェクトにする（`window.addEventListener('scroll', 関数);`）
- `document.scrollingElement.scrollTop` プロパティを使えば、スクロール量を `px`（ピクセル）数で取得できる
- `HTML要素.style.CSSプロパティ = 変更後の値;`（例：`backBtn.style.display = 'inline';`）と記述することで、JavaScriptからCSSのスタイルを変更できる

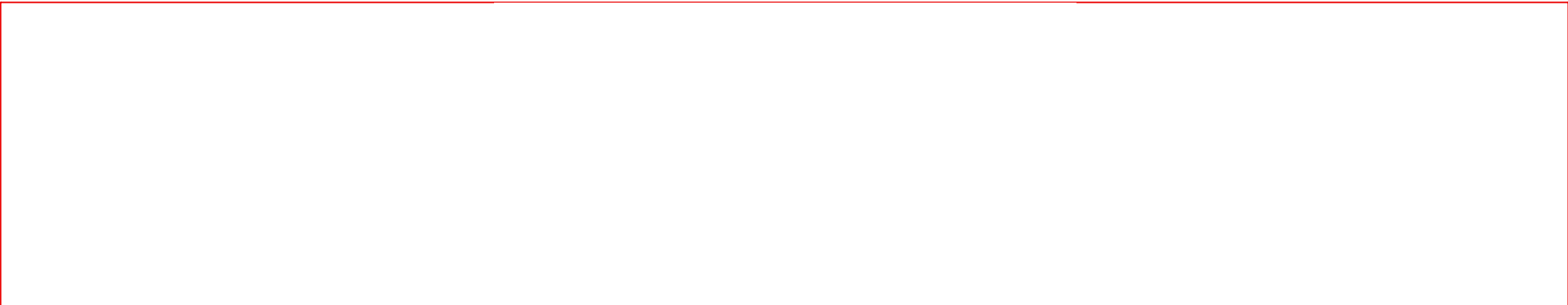
17章から続けてイベント処理について学んできましたが、「**ユーザーの行動に合わせてDOM操作を行う**」というイベント処理の特徴を少しでも理解できたでしょうか。

イベント処理を活用すれば、実際のWebサイトでは以下のようにリッチな表現を実現できます。

- 下にスクロールすると文字や画像がフェードインで表示される
- ページ内リンクをクリックしたときのスクロールをなめらかにする
- 画像をクリックしたときにモーダルで拡大表示する

今後、他の教材で学習する際や実際にWebサイトを制作する際は、ぜひ本教材を復習しながらさまざまなイベント処理に挑戦してみてください。

次章では、非同期処理について学びます。





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

～50%

50～80%

80～100%

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

23 / 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。