

イベントとイベントハンドラー

イベントドリブンとイベントリスナー


ブラウザで表示されたページでは、様々なイベント（クリックされた、マウスカーソルが乗った、内容が変更されたなど）が発生します。JavaScript では、それらのイベントに応じて実行する処理を実装するプログラミングモデルのことをイベントドリブンモデルと呼び、様々なイベントに対応した処理内容を定義する関数のことをイベントリスナーまたは、イベントハンドラーといいます。


イベントの種類（一部抜粋）

load	ページや画像の読み込みが完了した時
click	クリックした時
dblclick	ダブルクリックした時
mousedown	マウスボタンを押した時
mouseup	マウスボタンを離した時
mousemove	マウスカーソルが移動した時
mouseover	マウスカーソルが要素に乗った時
mouseout	マウスカーソルが要素から外れた時
change	内容が変更された時
reset	リセットボタンを押した時
submit	サブミットボタンを押した時
focus	要素がフォーカスされた時
blur	要素からフォーカスが外れた時
resize	要素のサイズを変更した時
scroll	スクロールした時

イベントリファレンス | MDN

イベントは、コードの実行に影響を与える可能性のある「興味深い変化」をコードに通知するために発行されます。これは、マウス操作やウィンドウのサイズ変更などのユーザー操作や、環境の変化（バッテリー残量の低下や OS のメディアイベントなど）、その他の原因によ

 <https://developer.mozilla.org/ja/docs/Web/Events>

 mdn web docs

イベントハンドラー/イベントリスナーを定義する

イベントドリブンモデルの中心となるのは、イベントとイベントリスナーとなり、イベントドリブンモデルでは、以下の3つを定義しておく必要があります。

1. どの要素で発生した
2. どのイベントを
3. どのイベントリスナーに関連付けするか

JavaScriptでは、上記関連付けをおこなうためにいくつかの方法が提供されています。

1. タグ内の属性として宣言する（インラインJavaScriptハンドラーとなるため非推奨）

2. 要素オブジェクトのプロパティとして宣言する

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
  <title>Web演習2 - サンプル3</title>
</head>
<body>
  <h1>JavaScript</h1>
  <p>クライアントサイドの動的なウェブサイト構築が可能になります。</p>
  <div>
    <input type="text" id="todo_text" placeholder="予定を入力してください">
    <button type="button" id="btn_save">登録</button>
  </div>
  <h2>ToDo List</h2>
  <ul id="todo_list"></ul>
  <script src="sample03_1.js"></script>
</body>
</html>
```

```
{
  // 要素のプロパティでイベントの定義
  // #btn_saveのクリックイベント

  // ラムダ式でのfunction
  document.querySelector( '#btn_save' ).onclick = ( eve
    console.log( '#btn_save click' );
    console.log( event );
  )
  /*
  // function
  document.querySelector( '#btn_save' ).onclick = funct
    console.log( '#btn_save click' );
    console.log( event );
  }
  */
}
```


JavaScriptでの関数定義には、**function** と **アロー関数（ラムダ式）** での記述方法が用意されています。アロー関数は、**ES6（ECMAScript 2015）** から新たに用意された記述方法で、以前からある function とでは、後方互換性のために相違点があります。

上記サンプルでは、無名（匿名）関数をonclickプロパティに保存して定義しています。無名関数ではなく、関数名を使ってイベントハンドラーを定義することも可能ですが、イベントハンドラーを複数箇所を使い回すことはあまりないため、無名関数で定義するほうがシンプルに記述できると思います。

アロー関数式 - JavaScript | MDN

アロー関数式は、従来の 関数式の簡潔な代替構文ですが、制限があり、すべての場面で使用することができるわけではありません。

 https://developer.mozilla.org/ja/docs/Web/JavaScript/Reference/Functions/Arrow_functions

 mdn web docs

3. addEventListenerメソッドを使って宣言する

```

{

// 要素のプロパティでイベントの宣言
// #btn_saveのクリックイベント

// addEventListenerメソッドを使って宣言
document.querySelector( '#btn_save' ).addEventListener( 'click', ( event ) => {
  console.log( event );
});

/*
// ラムダ式でのfunction
document.querySelector( '#btn_save' ).onclick = ( event ) => {
  console.log( event );
}

// function
document.querySelector( '#btn_save' ).onclick = function( event ) {
  console.log( event );
}
*/


}

```

EventTarget.addEventListener() - Web API | MDN

addEventListener() は EventTarget インターフェイスのメソッドで、ターゲットに特定のイベントが配信されるたびに呼び出される関数を設定します。

 <https://developer.mozilla.org/ja/docs/Web/API/EventTarget/addEventListener>

 mdn web docs