

JavaScript 入門講座

JavaScript 第 4 回 / 全 6 回

練習 1 のヒント

setTimeout 関数を利用して、5秒後に文字が書き換わるようにしてみましょう。

setTimeout 関数の使い方：

ほぼ setInterval と同じです。ただし、関数は1回しか呼ばれません。

```
function display() {  
    const element = document.getElementById('hello');  
    element.innerText = "Good Morning";  
}  
  
setTimeout(display, 5000);
```

配列

たくさんの変数を組として扱う仕組み

添字（出席番号）	0	1	2	3	4	5	6	7
値	3	1	4	1	5	9	2	6

作り方：

```
const pi = [3, 1, 4, 1, 5, 9, 2, 6];  
console.log(pi[2]);
```

配列と繰り返し

```
const list = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37];  
for (let i = 0; i < list.length; i++) {  
    console.log("配列の" + i + "番目は" + list[i] + "です。");  
}
```

配列への代入

配列の中身を変更したくなったら、直接代入も出来ます。

```
let list = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37];  
list[2] = 4;  
for (let i = 0; i < list.length; i++) {  
    console.log("配列の" + i + "番目は" + list[i] + "です。");  
}
```

やってみよう

```
let list = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37];
```

上の数値配列について、以下のプログラムを書いてください。

1. 各値を並べられた順番に表示する
2. 合計値を算出する
3. 平均値を算出する

※出来るところまでで構いません。

ブラウザをコントロール 1

ブラウザの情報を取得する。

```
<html>
  <body>
    <div>Hello!</div>
  </body>
  <script type="text/javascript">
    console.log(location.href);
  </script>
</html>
```

ブラウザをコントロール 2

ブラウザをコントロールする。

```
<html>
  <body>
    <div>Hello!</div>
  </body>
  <script type="text/javascript">
    location.href = "http://www.google.co.jp/";
  </script>
</html>
```


イベント 1

- あるイベント（マウスでクリックなど）が発生したときに、JavaScript プログラムを起動させることができます。
- これをイベントの登録などと表現し、プログラムが起動することを発火などと表現します。

```
<script type="text/javascript">  
    function buttonClick(){  
        alert('Click');  
    }  
</script>
```

```
<input type="button" value="button" id="myid2" onclick="buttonClick()">
```

イベント 2

HTML と JavaScript ファイルは、別ファイルに分けておきたい。

```
function buttonClick(){  
    alert('Click');  
}  
  
const button = document.getElementById('myid2');  
button.addEventListener('click', buttonClick);
```

練習

1. マウスをクリックしたら、 "Hello, World" から "Good Morning" へ変化するページを作ってみましょう。
2. マウスでクリックするたびに、 "Hello, World" と "Good Morning" が切り替わるページを作ってみましょう。

ファイルへの保存

```
<a href="#" id="weather" download="sample.html">ダウンロード</a>
```

```
document.getElementById('weather')
  .addEventListener('click', function(event) {
    // 保存する文字列の Blob オブジェクトを作成
    const blob = new Blob(["<html><body><h1>hello</h1></body></html>"],
      {type: 'text/html'});
    // a 要素の href 属性に Object URL を セット
    event.currentTarget.href = window.URL.createObjectURL(blob);
  });
```

ファイルの読み込み

今ダウンロードしたファイルを読み出してみましょう。
ファイルの読み書きには通常 File API を使用します。

```
<input id="myfile" type="file">
```

```
const f = document.getElementById('myfile');  
f.addEventListener('change', function(evt) {  
  const input = evt.target;  
  const file = input.files[0];  
  const reader = new FileReader();  
  reader.onload = function() {  
    // 読み出し結果の表示  
    console.log(reader.result);  
  };  
  reader.readAsText(file); // 読み込み開始  
});
```

CSV ファイル

- Excel みたいな表計算ライクな構造をもったデータ
- 1行で一つのデータの塊を表し、各データは記号「`,`」で区切ります。

タイトル	著者	発行年
博士の愛した数式	小川 洋子	2003
円周率 π の不思議	堀場 芳数	1989
超幾何関数入門	木村 弘信	2007

タイトル, 著者, 発行年
博士の愛した数式, 小川 洋子, 2003
円周率 π の不思議, 堀場 芳数, 1989
超幾何関数入門, 木村 弘信, 2007

CSV ファイルの読み込み

```
// 配列を定義
const csvArray = [];

// 改行ごとに配列化
const lines = body.split(/\r\n|\n/);

// 1行ごとに処理
for (let i = 0; i < lines.length; ++i) {
    let cells = lines[i].split(",");
    csvArray.push(cells);
}

console.log(csvArray);
```

※画面表示などで使用する制御文字エスケープシーケンスについては、JavaScript 本格入門 P73 を参照