

DOMの操作

前回までは他のプログラミング言語でもよくあるパターンでしたがDOMの操作はJavaScript特有です。

プログラミングではありますが、HTML/CSSも絡んできます。

DOMの取得

まずHTMLを用意し、下記のタグを書いてください。

代表的なものを2つ紹介します。

```
<h1 id="main-title">MainTitle</h1>
<p class="contents">contents</p>
```

idの取得

```
const mainTitle = document.getElementById("main-title")

console.log(mainTitle);
```

これをJSファイルに書いて読み込むと、上記で書いたh1の要素がすべて取得できています。

`document`はJSファイルを読み込んでいるHTMLが対象です。

そして`getElementById`でid名を取得しています。

`()`の中に書かれたid名がHTML上に存在したら取得できます。

class名の取得

```
const contents = document.getElementsByClassName("contents")

console.log(contents);
```

idの時と少し表示の仕方が違うと思います。

これはHTMLのid名とclass名の仕様にに基づいているのでclass名は配列で取得しています。

DOMの挿入

まず下記のコードをHTMLに貼り付けてください。

```
<div class="wrapper">
  <div class="box">子要素1</div>
  <div class="box">子要素2</div>
</div>
```

jsファイルに下記のコードを貼ってください。

```
// idで指定したwrapperを取得
const wrapper = document.getElementById("wrapper")
// 挿入するHTML要素
const newBox = `<div class="new-box">new-box要素</div>`

// setTimeoutはJavaScriptが用意している関数です・
setTimeout(function () {
  // wrapper要素内先頭にnewBoxに代入されたHTML要素を挿入
  wrapper.insertAdjacentHTML("afterbegin", newBox)
  // wrapper要素の直後にnewBoxに代入されたHTML要素を挿入
  wrapper.insertAdjacentHTML("afterend", newBox)
}, 3000)
```

できたらリロードして3秒程まってください。

新しい要素が追加されたかと思います。

このようにして取得したDOMを基準にして要素の追加が行えます。

DOMの削除

まず下記のコードをHTMLに貼り付けてください。

```
<div id="parent">
  <div id="child">取り除かれる要素</div>
</div>
```

jsファイルに下記のコードを貼ってください。

```
// idで指定したparentを取得
const parent = document.getElementById("parent")
// idで指定したchildを取得
const child = document.getElementById("child")

setTimeout(function () {
  // 3秒後に#child要素が削除される
  parent.removeChild(child)
}, 3000)
```

リロードして3秒待ってください。

表示してあった文字が消えたはずです。

このようにしてDOMの削除が行えます。