

JavaScript 研修

WEB 業界向いている人
 自分で問題解決できる人。
 興味をもって自分で色々調べることができる人。

学習の目標：自分で JavaScript の勉強ができるようにすること
 本を買うなら必ず後ろを見る。いつ刊行されたのか。できるだけ新しい本を買う。(1-2 年)

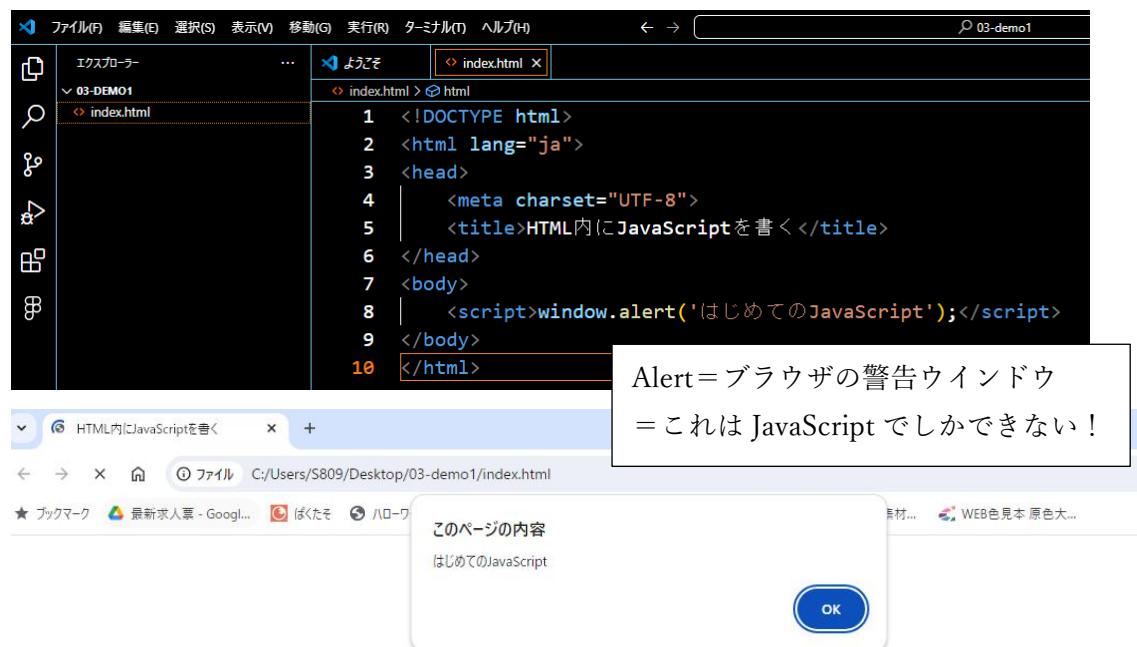
JavaScript の効果→ブラウザ上でもアプリみたいに使える。

JavaScript と JAVA は違う！目的も書き方も違う！メロンとメロンパンぐらい違う！
 JavaScript=JS

JavaScript の始め方

教科書 p 33- 参照

HTML に書く方法



```

<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="UTF-8">
<title>HTML内にJavaScriptを書く</title>
</head>
<body>
<script>window.alert('はじめてのJavaScript');</script>
</body>
</html>

```

Alert = ブラウザの警告ウインドウ
 = これは JavaScript でしかできない！

JavaScript ファイルに作成する方法

The screenshot shows a code editor interface with two tabs: 'index.html' and 'first.js'. The 'index.html' tab is active, displaying the following HTML code:

```

1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4   <meta charset="UTF-8">
5   <title>JavaScriptファイルを読み込む</title>
6 </head>
7 <body>
8   <script src="first.js"></script>
9 </body>
10 </html>

```

The screenshot shows a code editor interface with two tabs: 'index.html' and 'first.js'. The 'first.js' tab is active, displaying the following JavaScript code:

```

1 alert('はじめてのJavaScript');

```



p 39 指示の内容という項目をまずはおさえる

p 40

The screenshot shows a code editor interface with two tabs: 'index.html' and 'first.js'. The 'index.html' tab is active, displaying the following HTML code:

```

9 </body>
10 </html> | |

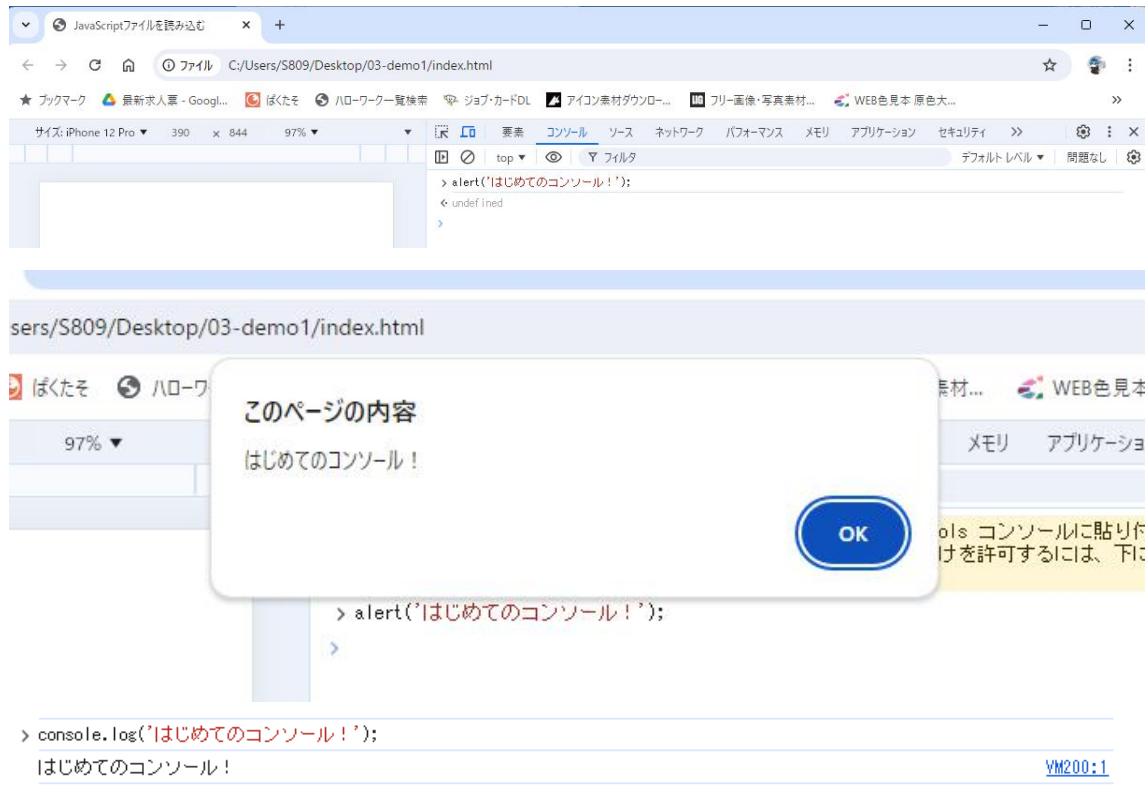
```

A callout box points to the blank line after '</html>' with the text '←全角スペースを入力するとこの表示'.

エスケープシーケンス

Mac→＼＼‘

Windows→¥‘



ファイルの書き換えはできない！

エラーメッセージが出てきたら、そのメッセージをコピペして検索→解決策がいっぱい出てくる

P48-90 本を読みながら各自で進める。

Point !

HTML に defer 属性を付ける！ p 54 参照

8 <script src="js/script.js" defer></script>

```
js> JS script.js > ...
1  console.log(document.querySelector('#colorPicker').value);
2  document.querySelector('#colorText').textContent='カラーコード:${document.querySelector('#colorPicker').value}';
3  const text=document.querySelector('#colorText');
4  const color=document.querySelector('#colorPicker');
5  text.textContent='カラーコード${color.value}';|_
6  color.addEventListener('input',colorBg);
```

const=定数=ラベルのようなもの。

text と書いたらそれは document.querySelector('#colorText') を指す。

color と書いたらそれは document.querySelector('#colorPicker') を指す。

3 const text=document.querySelector('#colorText');
4 const color=document.querySelector('#colorPicker');

//何がどうなったらどうなるか=.addEventListener.

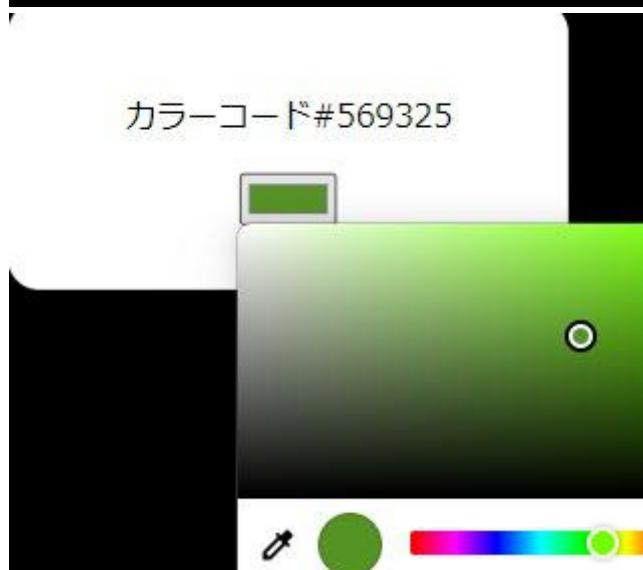
今回は色が input されたら、カラーコードを表示する

5 text.textContent='カラーコード\${color.value}';
6 color.addEventListener('input',colorBg);

関数=動作をひとまとめにする。ここでも const を使う。

const 関数名=()=>{関数名が呼び出されたときに何をするか}

```
6 ~ const colorBg=()=>{
7   text.textContent=`カラーコード${color.value}`;
8 }
9
10 //カラーピッカーが変更されたらcolorBgを発動させる
11 color.addEventListener('input',colorBg);
12 }
```



カラーピッカーを入力したら
(=色を選択したら)
colorBg 関数を発動させる
(=カラーコードをテキストで
表示する)

Point !

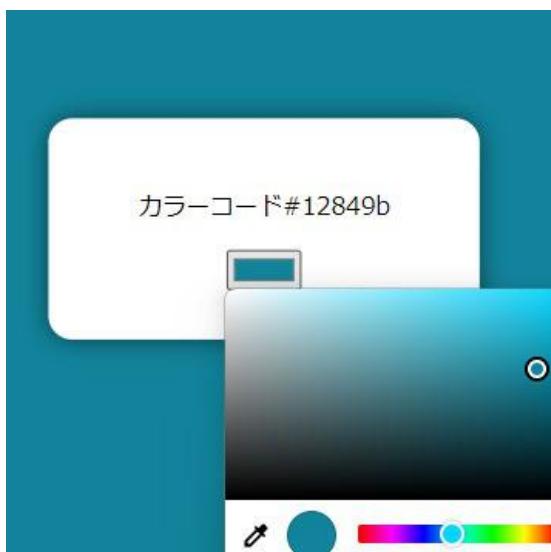
//はコメントアウト。

HTML の/ ! ---/や CSS の/**/のようなもの。

関数等は日本語で全部先にコメントを書いちゃう。

→そこからプログラムを書いていく。

```
6 //カラーピッカーを操作したときの一連の動作
7 const colorBg=()=>{
8 //選択した色を背景色に設定
9   document.body.style.backgroundColor=color.value;
10
11 //カラーコードを表示
12   text.textContent='カラーコード${color.value}';
13 }
14
15 //カラーピッカーが変更されたらcolorBgを発動させる
16 color.addEventListener('input',colorBg);
```



条件分岐

```

11 //カラーコードを表示
12 if(color.value === '#ffffff'){
13   text.textContent=^カラーコード${color.value}(white)^;
14 }
15 }
```

カラーコード#ffffff(white)



白を選んだ時は(white)と書いて

```

10
11 //カラーコードを表示
12 if(color.value === '#ffffff'){
13   text.textContent=^カラーコード${color.value}(white)^;
14 }else if(color.value === '#000000'){
15   text.textContent=^カラーコード${color.value}(black)^;
16 }else{
17   text.textContent=^カラーコード${color.value}^;
18 }
19 }
```

カラーコード#000000(black)



カラーコード#4ac5de



黒選んだら (black) と表示、 白・黒以外の色を選んだらカラーコードを表示