Webプログラミング
第８回

前回まで

- フォームを使うページ
  - フォームとは
  - 「お問い合わせページ」の作成
    - セレクトリストの追加
    - ラジオボタン、チェックボックス、テキストフィールド、テキストエリア
  - 送信ボタンの追加
  - ラベル
  - フォームのスタイル

今回の内容

- JavaScriptとは
- コンソールにアウトプット
- JavaScriptを記述する場所
- ダイアログボックスを表示する
- HTMLを書き換える
- JavaScriptの基本的な機能
  - 確認ダイアログボックス、if文

JavaScriptとは

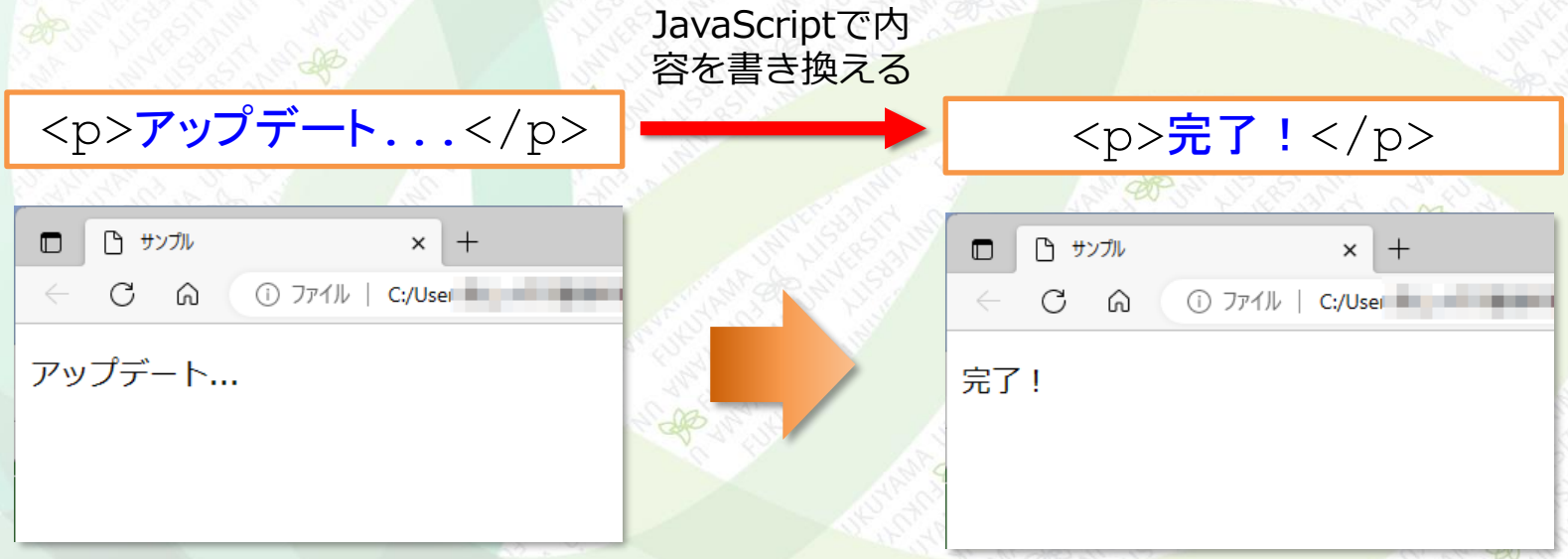
- JavaScriptは「ブラウザを操作するためのプログラミング言語」
- HTMLやCSSではできないことをする
- ブラウザ … Edge, Chrome, FireFox, Safariなど
- HTML&CSSは、一度ブラウザに読み込まれたら、もう変化することはない
- JavaScriptは、読み込み後も表示を動的に変更することができる

JavaScriptとは

- JavaScriptのWebページ変更のパターンは大きく分けて４つある

パターン１）タグに囲まれたテキストを書き換える

JavaScriptによって、<p>のテキストを書き換える



JavaScriptとは

- JavaScriptのWebページ変更のパターンは大きく分けて４つある

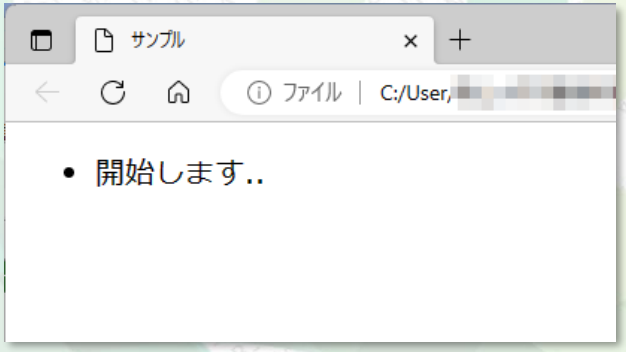
パターン２）要素を追加・削除する

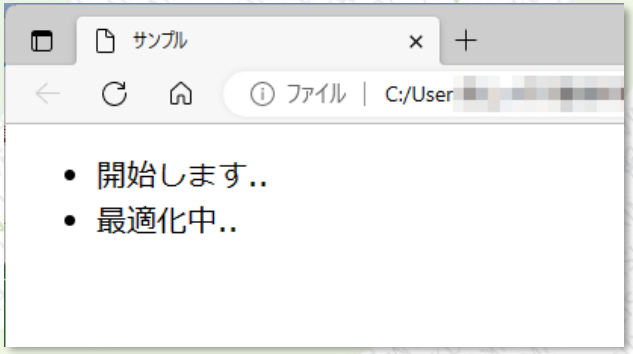
JavaScriptによって<li>を追加

```
<ul>
  <li>開始します..</li>
</ul>
```

JavaScriptで内容を書き換える

```
<ul>
  <li>開始します..</li>
  <li>最適化中..</li>
</ul>
```



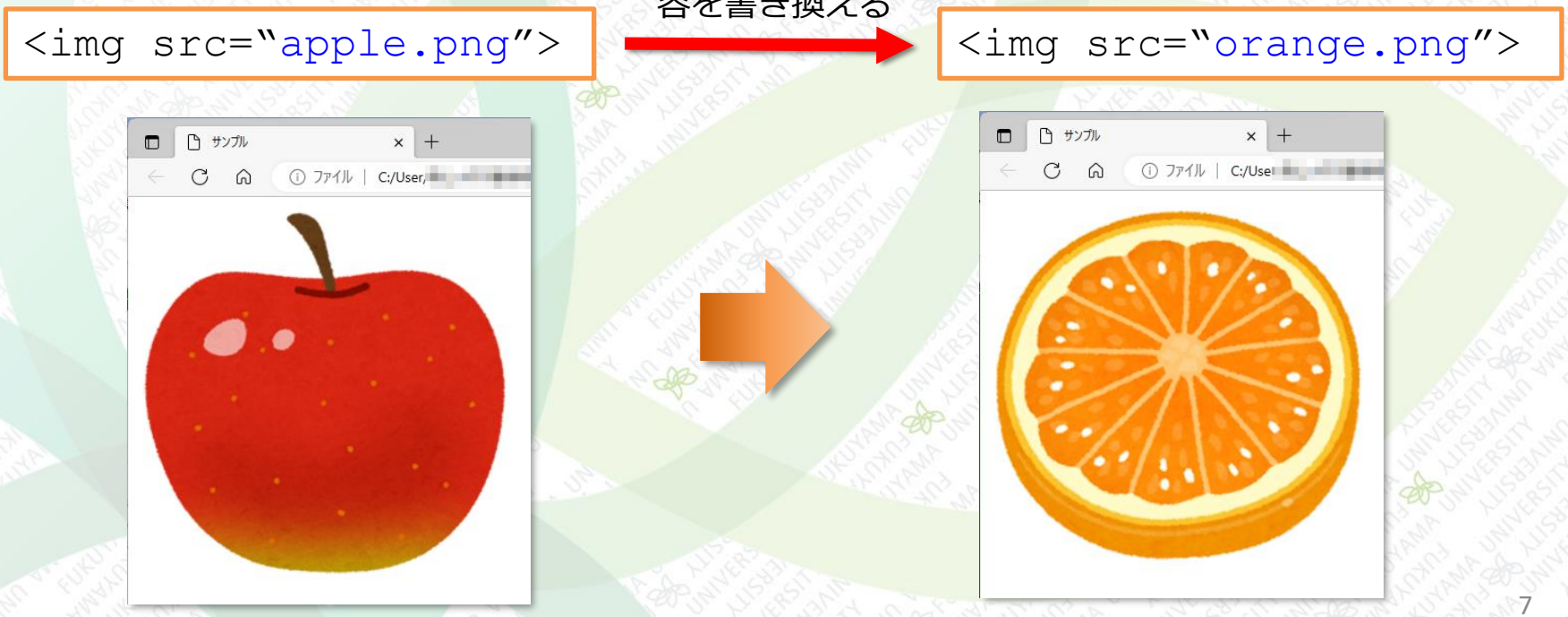


JavaScriptとは

- JavaScriptのWebページ変更のパターンは大きく分けて４つある

パターン３）タグの属性の値を変更する

class属性、id属性、href属性、src属性などの属性の値を変更できる



JavaScriptとは

- JavaScriptのWebページ変更のパターンは大きく分けて４つある

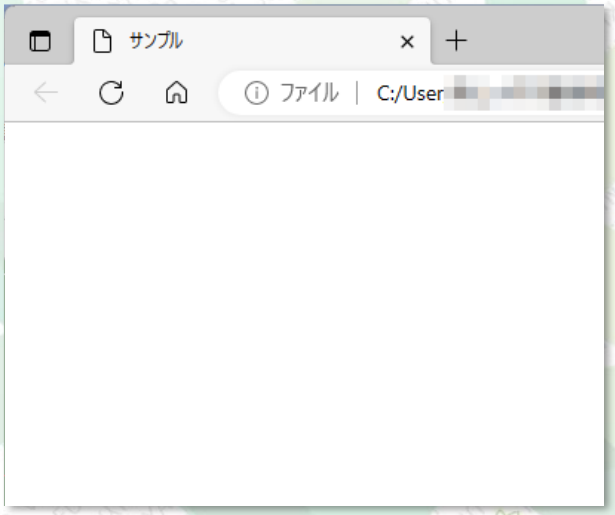
パターン４）CSSの値を変更する

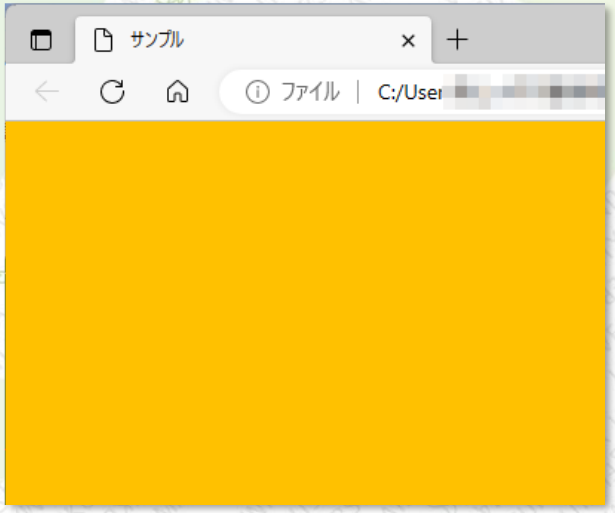
JavaScriptによって<body>の背景色を変更

```
body{
 background=“#ffffff”>
}
```

JavaScriptでCSSの内容を書き換える

```
body{
 background=“#ffc100”>
}
```





JavaScriptとは

- HTMLやCSSをリアルタイムに書き換えることができる
- JavaScriptでHTMLやCSSが書き換えられると、その変更がブラウザの表示に即座に反映される
- 画面が書き換わるのは、変更があった箇所のみ
- ページ全体の再読み込みは発生しない
- 変更の待ち時間もなし

JavaScriptとは

- ES2015(ES6)は「新しいJavaScript」
  - 2015年にJavaScriptの仕様が大きく改訂された
  - 正式名称は「ECMAScript」(通称ES2015またはES6)
    ちなみにInernet ExplorerはES6に対応していない

- JavaScriptプログラミングの基本的な構成は、以下のとおり

JavaScriptの作成に必要なツール

- JavaScriptプログラミングに必要なツール
  - ブラウザ
    - Edge, FireFox, Chrome, Safari など
  - テキストエディタ
    - Visual Studio Code(Windows/Mac)
    - Brackets(Windows/Mac)

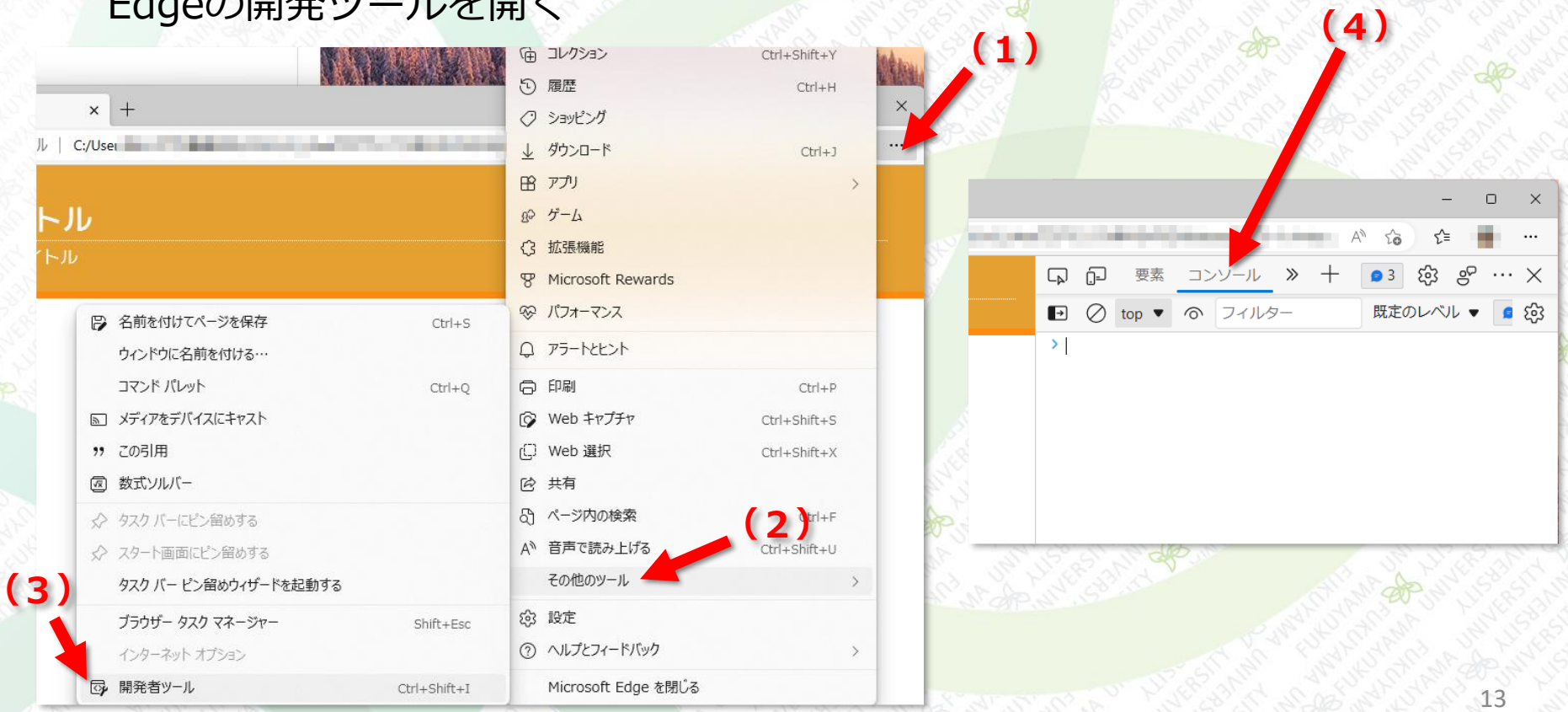
演習用のhtmlテンプレート

- JavaScript単体ではブラウザで動作を確認できない
  - 演習では、JavaScriptを表示するhtmlのテンプレートを用意して利用する
  - 「index.html」と「style.css」をダウンロードしておく

コンソールにアウトプット

- JavaScriptではブラウザのコンソールにテキストなどを表示できる
  - コンソールはブラウザの「開発ツール」を使う

Edgeの開発ツールを開く



コンソールにアウトプット

- 他のブラウザの「開発ツール」を開く手順
  - FireFox

    [メニュ]->[その他のツール] ->[ウェブ開発ツール]

  - Chrome

    [Google Chromeの設定]->[その他のツール]
    ->[デベロッパーツール]

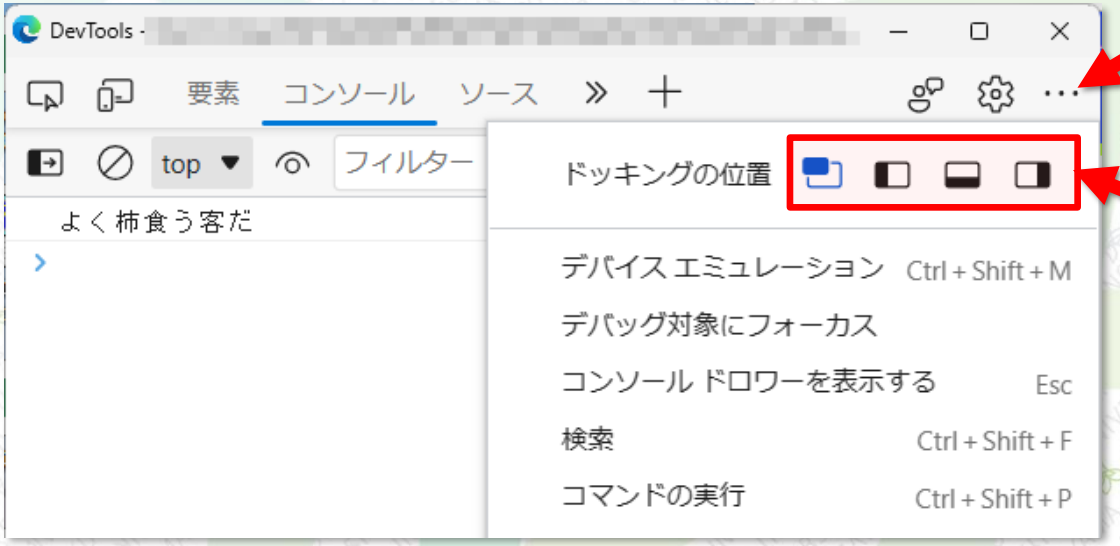
  - Safari
    - 最初に１度だけ、環境設定を変更する必要がある

    [環境設定]ダイアログ->[詳細]
    ->[メニューバーに“開発”メニューを表示]にチェック
    - Safariにて

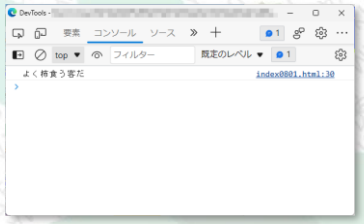
    メニューバーの[開発]->[Webインスペクタを表示]

コンソールにアウトプット

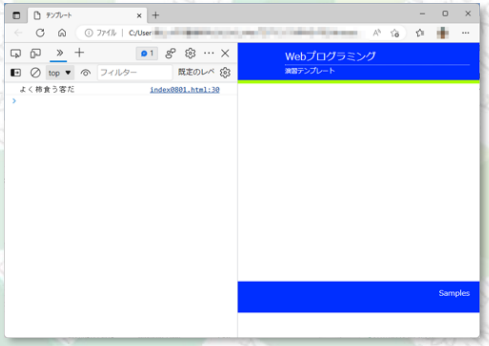
- コンソールの表示位置を変更できる
  - 別ウィンドウ、左にドッキング、下に‥、右に‥



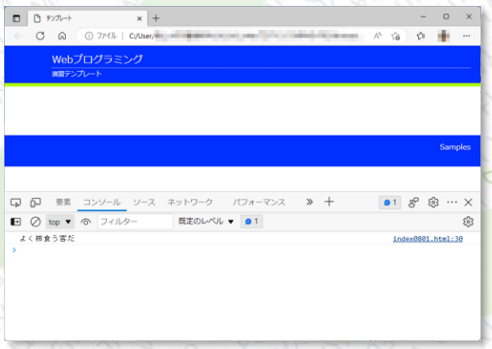
別ウィンドウ



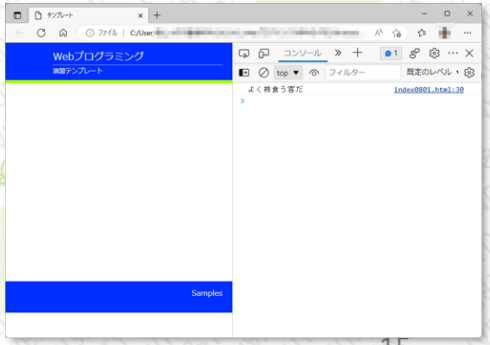
左にドッキング



下にドッキング



右にドッキング



コンソールにアウトプット

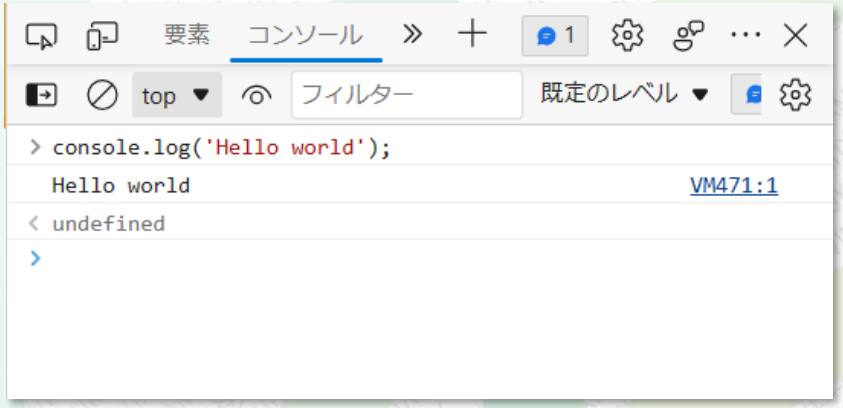
- 主なアウトプットは３種類
  - コンソールへのアウトプット
    - プログラムの動作を確認するなど
  - ダイアログボックスへのアウトプット
    - JavaScriptでダイアログボックスを表示して、テキストや数字を出力する
  - HTMLやCSSへのアウトプット
    - HTMLやCSSのタグに囲まれたテキストを書き換える、新たな要素を追加するなど
    - この操作で動的なWebページ作成を可能にしている

コンソールにアウトプット

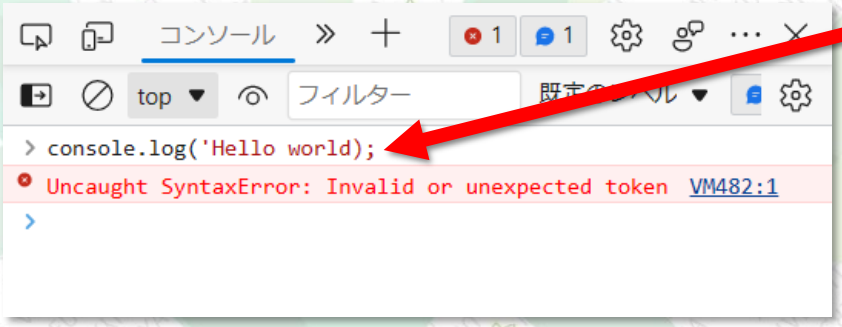
コンソールを使ってみる

コンソールに、以下を入力してみる

```
console.log('Hello world');
```



入力間違いがあると、エラーが表示される



シングルクォーテーションが抜けている

```
console.log('Hello world);
```

コンソールにアウトプット

さらにコンソールを使ってみる

・コンソールに、以下を入力してみる

```
console.log(2+3);
```

```
console.log(123-35);
```

```
console.log('2+3');
```

・ダブルクォーテーションで囲っても同じ

```
console.log("2+3");
```

・ダブルクォーテーションを表示文字列に含むことも可能

```
console.log('続けるには"C"キーを押す');
```

※以下の書き方はダメ

```
console.log("続けるには"C"キーを押す");
```

JavaScriptを記述する場所

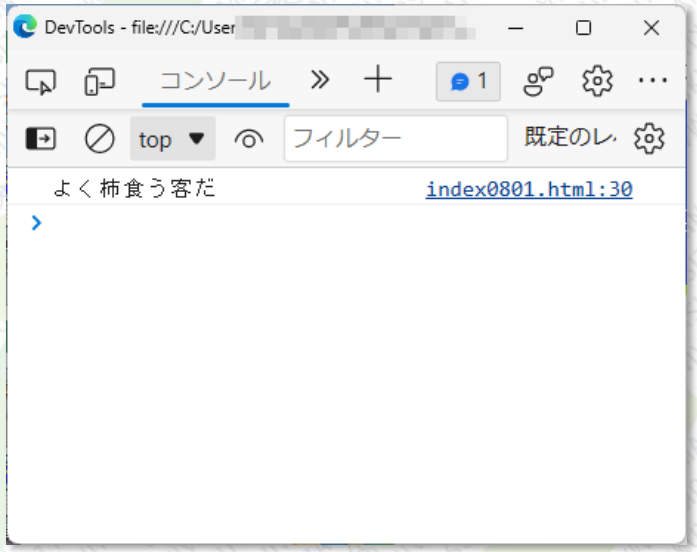
- 実際のWebページで利用するJavaScriptを記述するところは、以下の２つ
  - HTMLファイルに直接記述する
  - JavaScript専用のファイルをHTMLファイルとは別に作成し、HTMLから読み込むようにする

JavaScriptを記述する場所

- HTMLファイルに直接記述する
  - まずはじめに、テンプレートの「index.html」をコピーして、ファイル名を「index0801.html」に変更、下記を追記する

index0801.html

```
27          </footer>
28          <script>
29              'use strict';
30              console.log('よく柿食う客だ');
31          </script>
32      </body>
33  </html>
```



‘use strict’ について

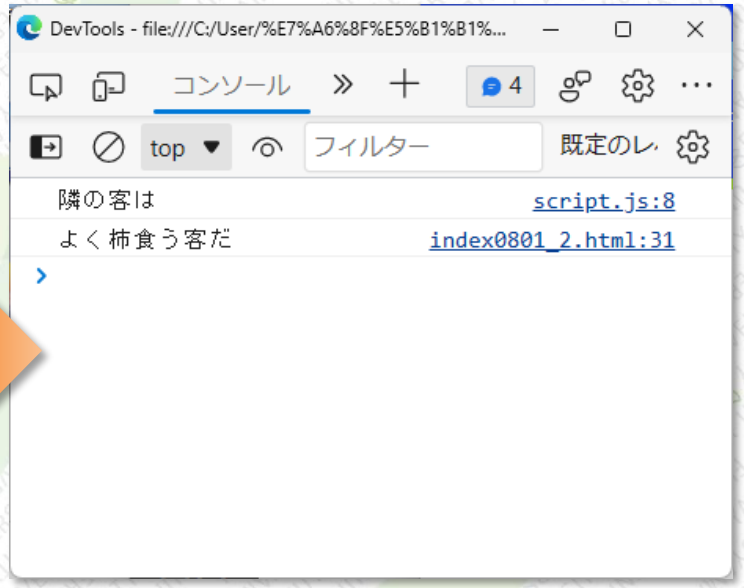
ブラウザには
「古いJavaScriptを実行するモード」
「新しいJavaScriptを実行するモード」… strictモード
がある
通常は strictモード で実行する、そのとき ‘use strict’を記述する

JavaScriptを記述する場所

- JavaScriptを別ファイルで作成する
  - 引き続き「index0801.html」に下記を追記する
  - さらにファイル「script.js」を新規作成して、下記を記述する

index0801.html

```
27      </footer>
28      <script src="script.js"></script>
29      <script>
30          'use strict';
31          console.log('よく柿食う客だ');
32      </script>
```



script.js

```
1   'use strict';
2
3   // 外部JavaScriptファイル
4   /*
5   外部JavaScriptファイルは
6   読み込まれたらすぐに実行されます。
7   */
8   console.log('隣の客は');
```

コメント文は /*　*/ で囲む
または、行頭に //

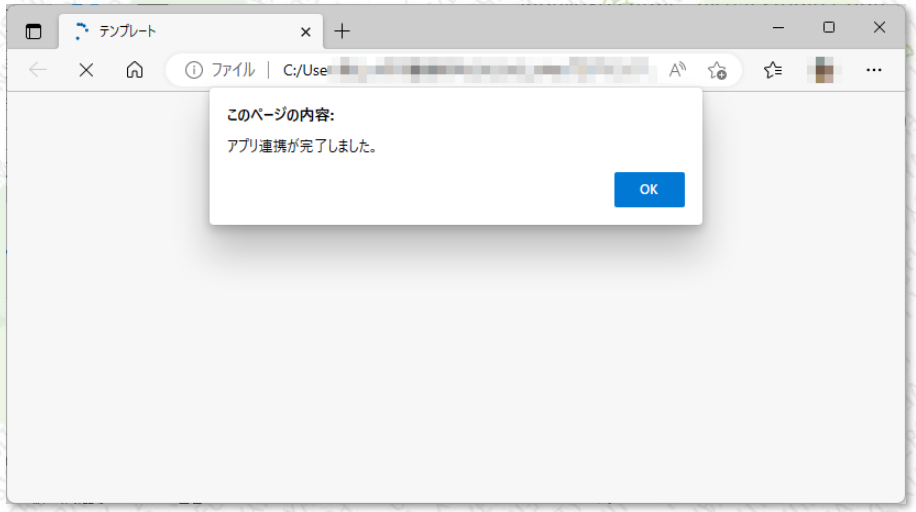
ダイアログボックスを表示する

- ダイアログボックスを表示、テキストを出力する
  - テンプレートの「index.html」をコピーして、ファイル名を「index0802.html」に変更、下記を追記する

index0802.html

```
27        </footer>
28        <script>
29            'use strict';
30            window.alert('アプリ連携が完了しました。');
31        </script>
32    </body>
```

ダイアログボックスが表示されて、テキストが出力される

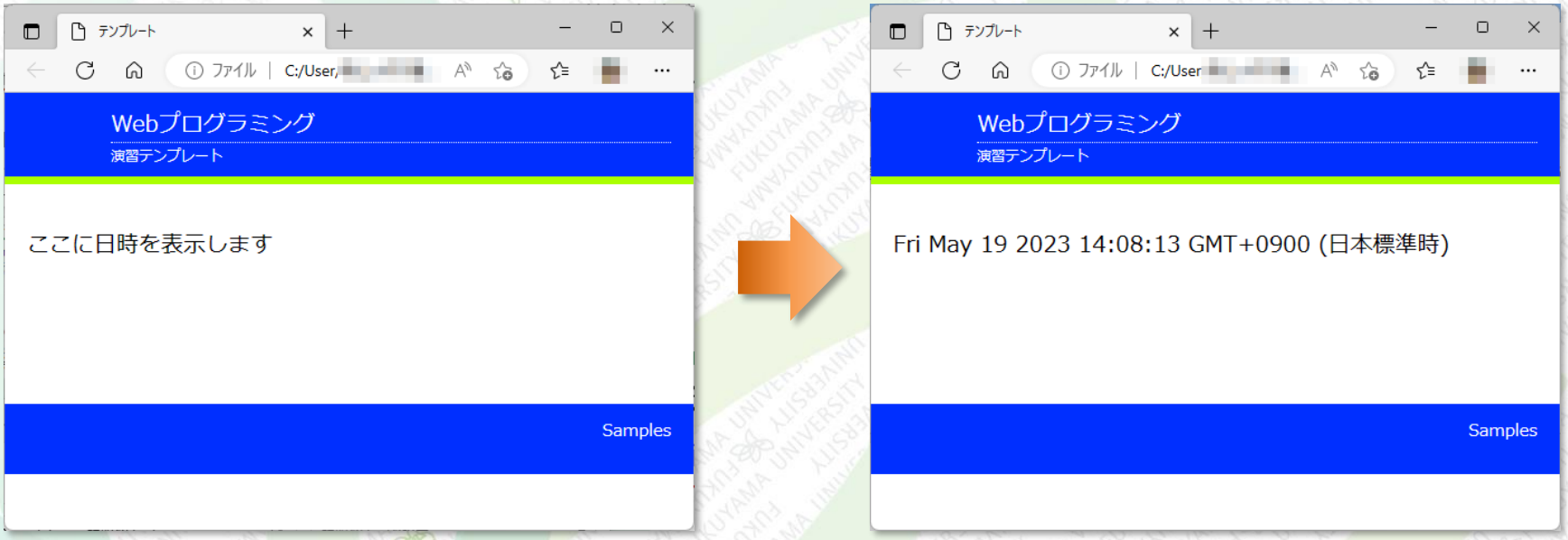


ダイアログボックス表示の書き方

```
window.alert(出力内容);
```

HTMLを書き換える

- JavaScriptで表示されているHTMLを書き換える
  - HTMLの中の「ここに日付を表示します」の部分を現在の日時表示に書き換える



- 以下、２段階の処理を行う
  - 書き換えたい部分の要素（HTMLとコンテンツ）を取得
  - 取得した要素のコンテンツを書き換える

HTMLを書き換える

- まず、書き換える前のHTMLを作成する
  - テンプレートの「index.html」をコピーして、ファイル名を「index0803.html」に変更、下記を追記する

index0803.html

```
16          <main>
17              <div class="container">
18              <section>
19                  <p id="choice">ここに日時を表示します</p>
20              </section>
21              </div><!-- /.container -->
22          </main>
23          <footer>
24              <div class="container">
25              <p>Samples</p>
26              </div><!-- /.container -->
27          </footer>
28          <script>
29              'use strict';
30              console.log(document.getElementById('choice'));
31          </script>
32      </body>
```

HTMLを書き換える

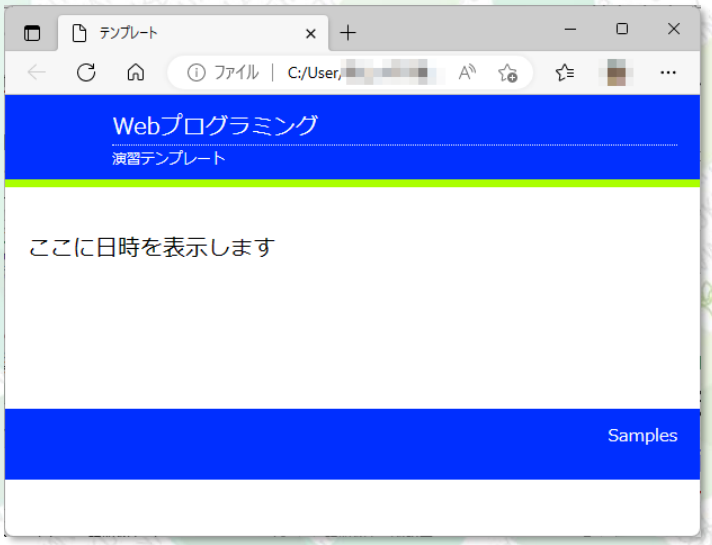
- 「index0803.html」では
  - 書き換えたい部分の要素を取得してコンソールに表示している
  - ここでは id属性で指定する

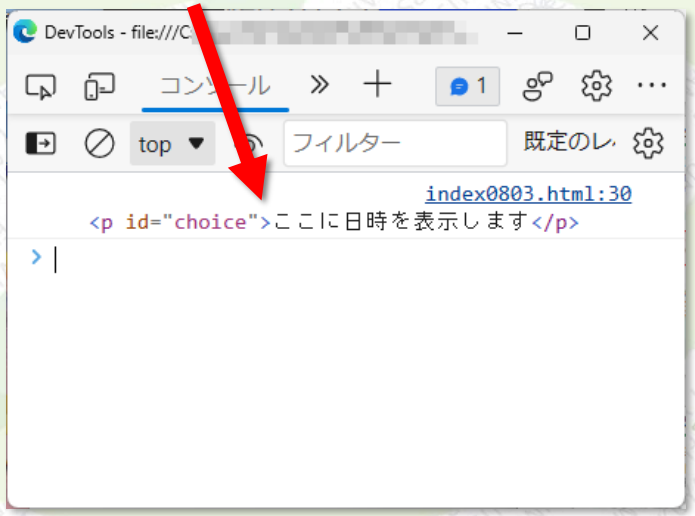
```
<p id="choice">ここに日時を表示します</p>
```

の要素を取得して、コンソールに表示するのは下記

```
console.log(document.getElementById('choice'));
```

取得できた要素が表示される





HTMLを書き換える

- document.getElementById() メソッド
  - documentオブジェクトにはHTMLやCSSを操作するための機能が多数用意されている
  - getElementById()メソッドは、()に指定されたid名を持つ要素を丸ごと取得する

```
document.getElementById('id名')
```

- 大文字・小文字の区別に注意する
  - JavaScriptはアルファベットの大文字・小文字を区別する
  - EとeBとbは別の文字として認識する

○ `document.getElementById('choice')`

× `document.getElementByid('choice')`

× `Document.getElementById('choice')`

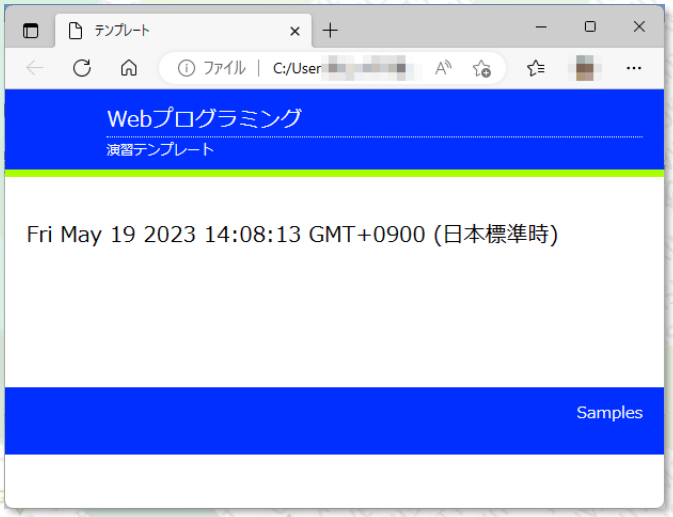
HTMLを書き換える

- 取得した要素のコンテンツを書き換える
  - 「index0803.html」に下記の変更を加える

index0803.html

```
28          <script>
29              'use strict';
30              console.log(document.getElementById('choice'));
31          </script>
32      </body>
```

```
28          <script>
29              'use strict';
30              document.getElementById('choice').textContent = new Date();
31          </script>
32      </body>
```



テキストが現在の日時に書き換えられた

HTMLを書き換える

- textContent
  - textContent は documentオブジェクト のプロパティ
  - 要素のコンテンツを書き換えるときに上書きする

```
document.getElementById('id名').textContent = '文字列';
```

記述例

```
document.getElementById('choice').textContent
                                    = new Date();
```

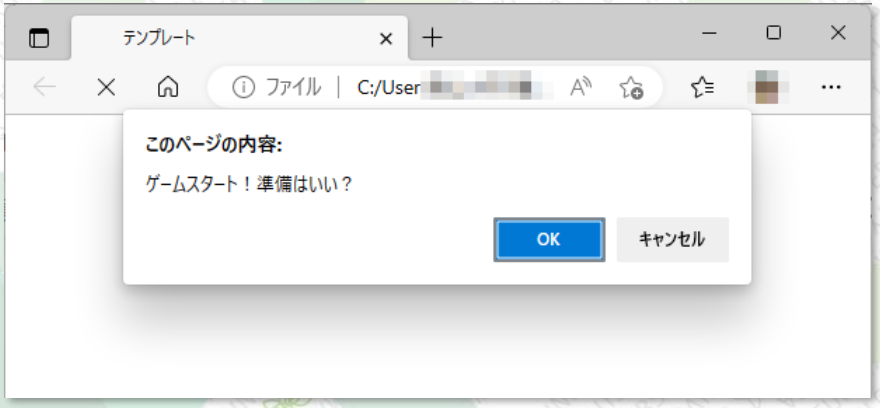
```
document.getElementById('choice').textContent
                                    = '通知を受け取りますか？';
```

JavaScriptの基本的な機能

- 確認ダイアログボックスを表示する
  - 条件分岐（if文）を用いて[OK] [キャンセル]を把握する
  - テンプレートの「index.html」をコピーして、ファイル名を「index0804.html」に変更、下記を追記する

```
27          </footer>
28          <script>
29              'use strict';
30
31              console.log(window.confirm('ゲームスタート！準備はいい？'));
32          </script>
33      </body>
```

index0804.html



確認ダイアログボックスが表示される

JavaScriptの基本的な機能

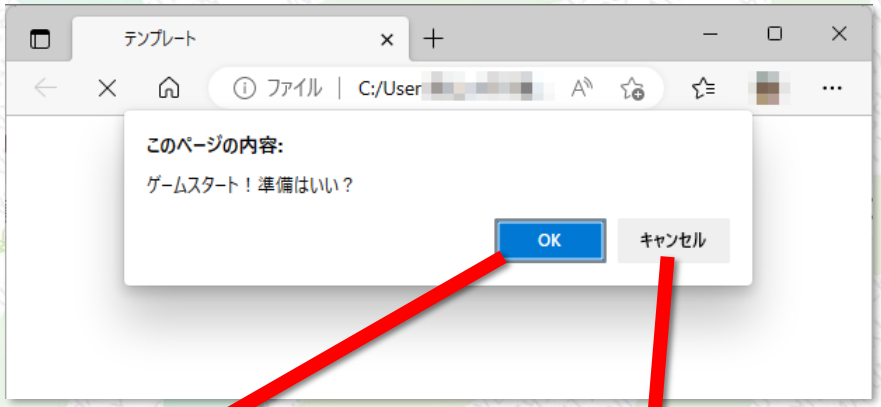
- 確認ダイアログボックスを表示する

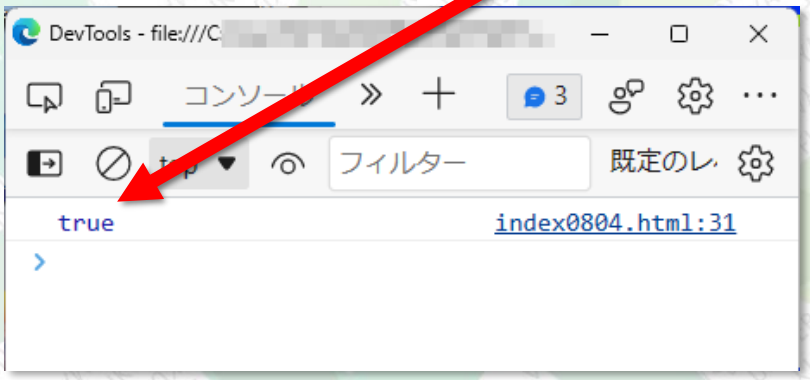
```
window.confirm(メッセージ)
```

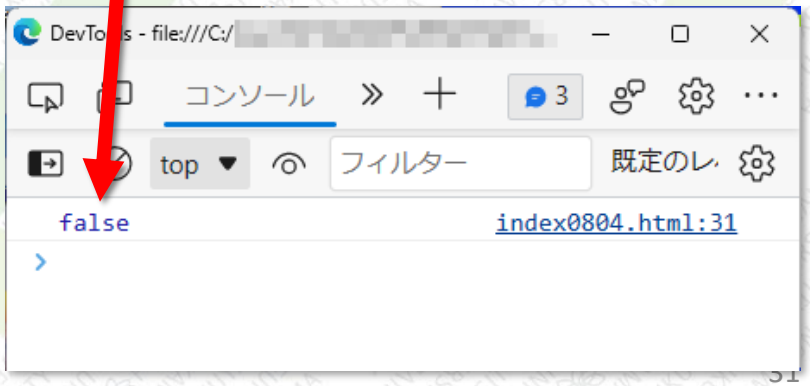
- リターン（return）
  - confirm()メソッドは、alert()メソッドとは異なり、リターン（戻り値・返り値）がある
- true と false
  - confirm()メソッドのリターンは ture か false のどちらか
  - true ・・・ 真（成り立つ）
  - false ・・・ 偽（成り立たない）
  - ブール値、ブーリアン値 と呼ぶ

JavaScriptの基本的な機能

- コンソールを確認
  - [OK] を押すと **ture** が表示される
  - [キャンセル]を押すと **false** が表示される







JavaScriptの基本的な機能

- [OK]と[キャンセル]でメッセージを変える
  - 条件分岐（if文）を用いて表示メッセージを変更する
  - 「index0804.html」に以下の変更を加える

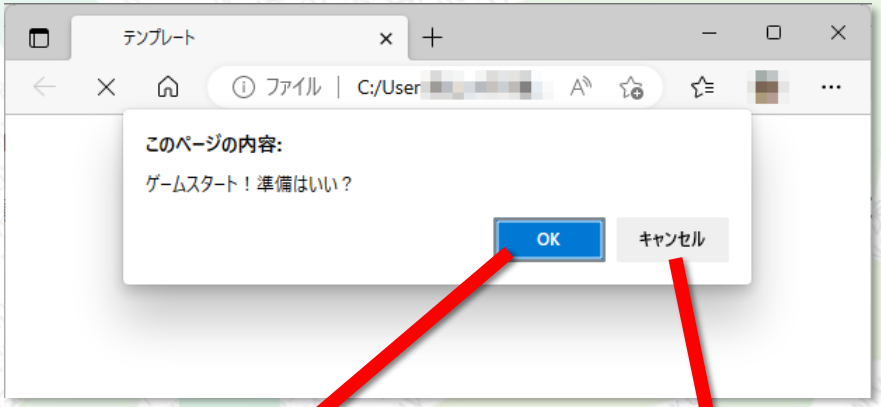
index0804.html

```
27          </footer>
28          <script>
29              'use strict';
30
31              console.log(window.confirm('ゲームスタート！準備はいい？'));
32          </script>
33      </body>
```

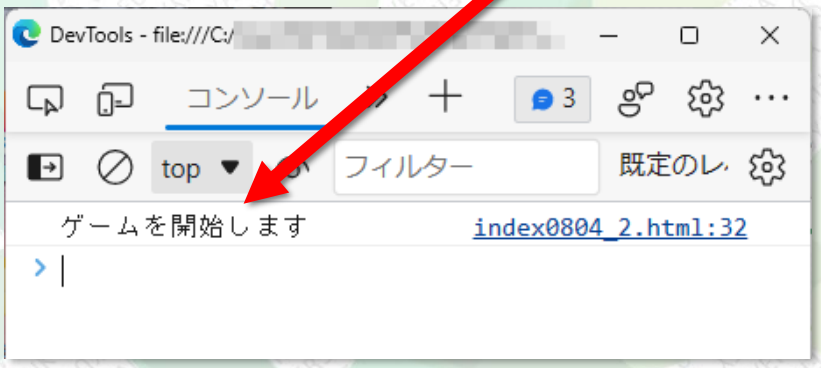
```
27          </footer>
28          <script>
29              'use strict';
30
31              if(window.confirm('ゲームスタート！準備はいい？')){
32                  console.log('ゲームを開始します')
33              }else{
34                  console.log('ゲームを終了します')
35              }
36          </script>
37      </body>
```

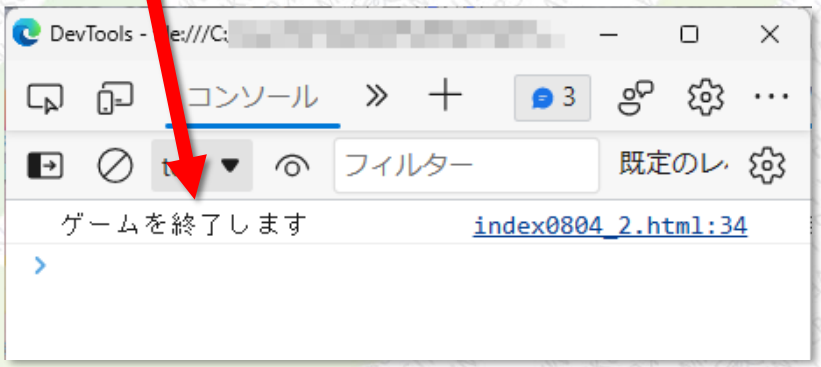
JavaScriptの基本的な機能

- [OK]と[キャンセル]でメッセージを変える
  - 条件分岐（if文）を用いて表示メッセージを変更する



やり直すときは、ブラウザで再読み込みをする





JavaScriptの基本的な機能

- if文
  - if文はJavaとほぼ同じ使い方

```
if(条件式){
   条件式が true になるときに実行される
}else{
   条件式が false になるときに実行される
}
```

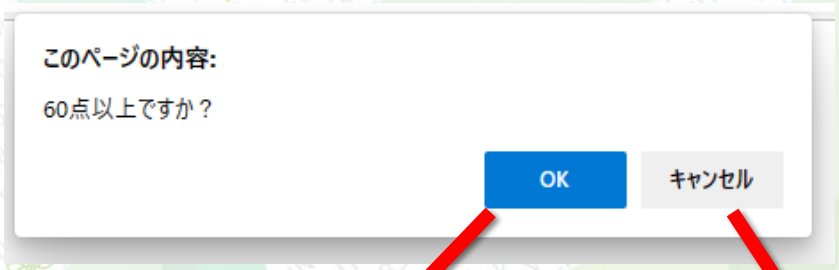
※ else{…}は省略可

  - if文のネスト（入れ子構造）も可能

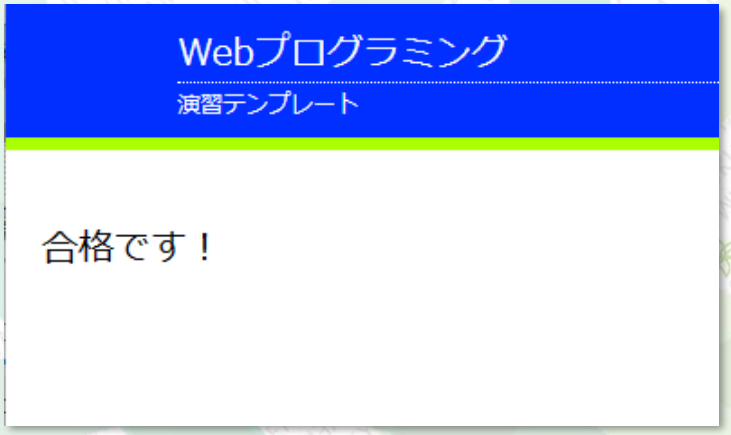
```
if(条件式){
    if(条件式){
       ・・・
    }else{
       ・・・
    }
}
```

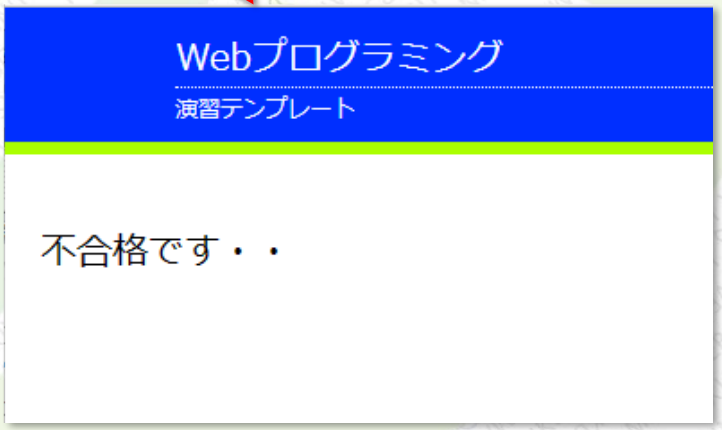
演習１

まずはじめに、テンプレートの「index.html」をコピーして、ファイル名を「ensyu01.html」に変更しなさい。
次に、下図のような確認ダイアログボックスを表示して、[OK]と[キャンセル]で表示が切り替わるように「ensyu01.html」を修正しなさい。



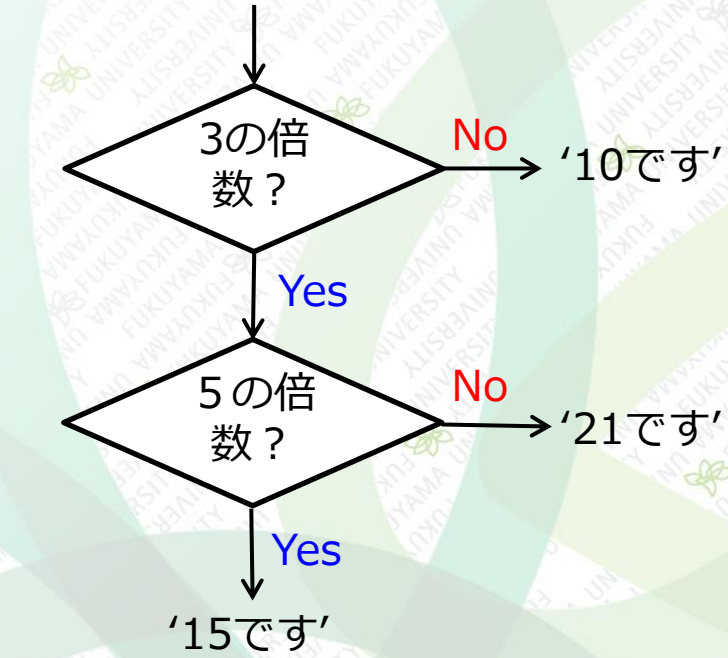
「HTMLを書き換える」のところも見直すこと





演習２

まずはじめに、テンプレートの「index.html」をコピーして、ファイル名を「ensyu02.html」に変更しなさい。
確認ダイアログボックスを利用して 10, 15, 21 のどの数なのかを当てるようにしたい。[OK]と[キャンセル]でif文の条件分岐を設定し、必要に応じて if文をネスト（入れ子）にすることで処理をする。「ensyu02.html 」を修正しなさい。なお、結果はすべてコンソールに出力すること。



<簡単なフローチャート>

コンソールの出力

※実行例は、分岐のときにnの倍数かどうかも同時に表示している