

Webプログラミング 第12回



前回まで

- FizzBuzzを作ってみる
 - 関数、if文、繰り返しの復習
- 項目リストを表示する（配列）
 - 配列のデータを作成、1個だけ表示してみる
 - 配列
 - テンプレートリテラル
 - 項目をHTMLに出力する

今回の内容

■ オブジェクト

- 複数のプロパティを持つデータ

- for ... inを用いて表示

- HTMLに出力

■ フォームの入力とイベント

- イベント

- フォーム、テキストフィールド

- HTMLに出力

オブジェクト

■ オブジェクトとは

- 複数のプロパティを持つデータのまとめ
- 各種データをひとまとめにして、1つの変数として扱える

■ アイテムの価格と在庫を表示する

Webプログラミング 演習テンプレート		
Webプログラミング	2500円	3

オブジェクト

■ アイテムの価格と在庫を表示する

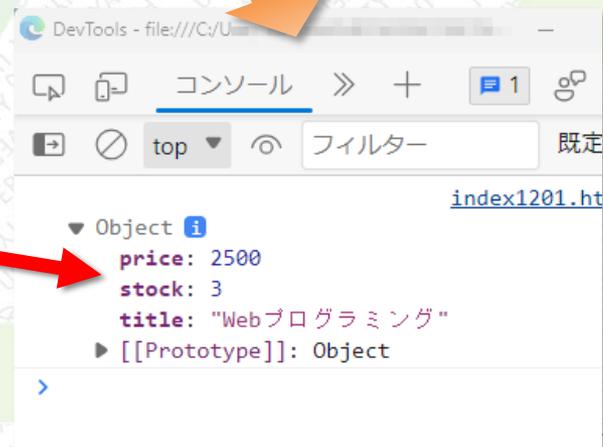
- テンプレートの「`index.html`」をコピーして、ファイル名を「`index1201.html`」に変更、下記を追記する
- テンプレートの「`style.css`」もコピーする

index1201.html

```
27     </footer>
28     <script>
29         'use strict';
30         let jsbook = {title: 'Webプログラミング', price: 2500, stock: 3};
31         console.log(jsbook);
32     </script>
33 </body>
```

複数の（異なる型も可）からなる**オブジェクト**を変数 `jsbook` に代入する

オブジェクトの**プロパティ**をすべて表示



オブジェクト

■ アイテムの価格と在庫を表示する

■ オブジェクト作成の書式

- ・ プロパティ、プロパティ名、データ（値）

```
let jsbook = {title: 'Webプログラミング', price: 2500, stock: 3};
```

Diagram illustrating the structure of the object literal: `{title: 'Webプログラミング', price: 2500, stock: 3};`. Red brackets above the code group each key-value pair as a "プロパティ" (Property).

プロパティ名

データ（値）

プロパティ名

データ

プロパティ名

データ

プロパティが0個のオブジェクトを作成

```
let 変数名 = {};
```

プロパティが1個以上あるオブジェクトを作成

```
let 変数名 = {プロパティ名1:データ, プロパティ名2:データ, ...,  
             ,プロパティ名N:データ};
```

オブジェクト

- アイテムの価格と在庫を表示する
 - プロパティを指定して表示してみる
 - 「index1201.html」に下記を追記する

```
27 </footer>
28 <script>
29   'use strict';
30   let jsbook = {title: 'Webプログラミング', price: 2500, stock: 3};
31   console.log(jsbook);
32   console.log(jsbook.title);
33   console.log(jsbook['price']);
34 </script>
```

index1201.html

プロパティ title と price
の値を表示する

コンソール

index1201

```
{title: 'Webプログラミング', price: 2500, stock: 3}
k: 3
  price: 2500
  stock: 3
  title: "Webプログラミング"
  [[Prototype]]: Object
```

Webプログラミング	index1201
2500	index1201

オブジェクト

■ アイテムの価格と在庫を表示する

■ プロパティを指定して参照する書式は2通り

1. ピリオドでつなぐ

オブジェクト名.プロパティ名

→ `jsbook.title`

2. []で囲む

オブジェクト名 ['プロパティ名']

→ `jsbook['price']`

※この場合の ' は、シングルクォーテーション
であることに注意

プロパティのデータを書き換えることもできる

```
jsbook.title = 'Javaプログラミング';  
jsbook['price'] = 3800;
```

オブジェクト

■ アイテムの価格と在庫を表示する

■ for...in を使って、すべてのプロパティを表示する

■ 「index1201.html」を下記のように修正・追記する

```
28 <script>
29   'use strict';
30   let jsbook = {title: 'Webプログラミング', price: 2500, stock: 3};
31   for(let p in jsbook) {
32     console.log(p + ' = ' + jsbook[p]);
33   }
34   /*
35   console.log(jsbook);
36   console.log(jsbook.title);
37   console.log(jsbook['price']);
38   */
39 </script>
```

index1201.html

※コメントアウト

すべてのプロパティが
表示される

title = Webプログラミング	index1201_3.html:32
price = 2500	index1201_3.html:32
stock = 3	index1201_3.html:32

オブジェクト

■ アイテムの価格と在庫を表示する

■ for...in の書式

```
for(let 変数名 in オブジェクト名) {  
  ...処理...  
}
```

※プロパティの順番通りに処理される保証はないので注意！

p に順番にプロパティが代入されて繰り返し処理される

```
let jsbook = {title: 'Webプログラミング', price: 2500, stock: 3};  
for(let p in jsbook) {  
  console.log(p + ' = ' + jsbook[p]);  
}
```

```
title = Webプログラミング  
price = 2500  
stock = 3  
>
```

p	→	jsbook[p]
title	jsbook[title]	'Webプログラミング'
price	jsbook[price]	2500
stock	jsbook[stock]	3

プロパティ

データ (値)

オブジェクト

■ アイテムの価格と在庫を表示する

■ HTMLに出力する

■ 「index1201.html」を下記のように修正・追記する

index1201.html

```
18 <section>
19   <table>
20     <tr>
21       <td id="title"></td>
22       <td id="price"></td>
23       <td id="stock"></td>
24     </tr>
25   </table>
26 </section>
```

```
34 <script>
35   'use strict';
36   let jsbook = {title: 'Webプログラミング', price: 2500, stock: 3};
37   for(let p in jsbook) {
38     console.log(p + ' = ' + jsbook[p]);
39   }
40   /*
41   console.log(jsbook);
42   console.log(jsbook.title);
43   console.log(jsbook['price']);
44   */
45 </script>
```

(1)削除する

```
34 <script>
35   'use strict';
36   let jsbook = {title: 'Webプログラミング', price: 2500, stock: 3};
37   document.getElementById('title').textContent = jsbook.title;
38   document.getElementById('price').textContent = jsbook.price + '円';
39   document.getElementById('stock').textContent = jsbook.stock;
40 </script>
```

(2)書き換える

オブジェクト

■ アイテムの価格と在庫を表示する

■ 表示されたが見にくい

Webプログラミング 演習テンプレート
Webプログラミング 2500円 3

■ 表の罫線を追加する、「index1201.html」を下記のように追記する

```
7 <link href="style.css" rel="stylesheet">
8 <style>
9   table {
10     border-collapse: collapse;
11   }
12   td {
13     padding: 4px 8px;
14     border: 1px solid #000000;
15   }
16 </style>
17 </head>
```

index1201.html

Webプログラミング 演習テンプレート		
Webプログラミング	2500円	3

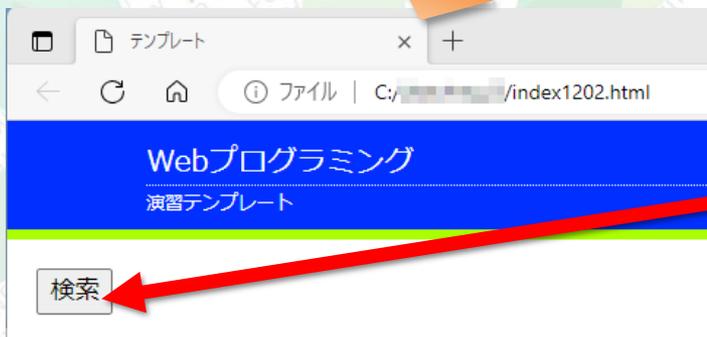
フォームの入力とイベント

- イベントとフォームのテキスト入力を使う
- まずはイベントをテストしてみる
 - テンプレートの「`index.html`」をコピーして、ファイル名を「`index1202.html`」に変更、下記を追記する

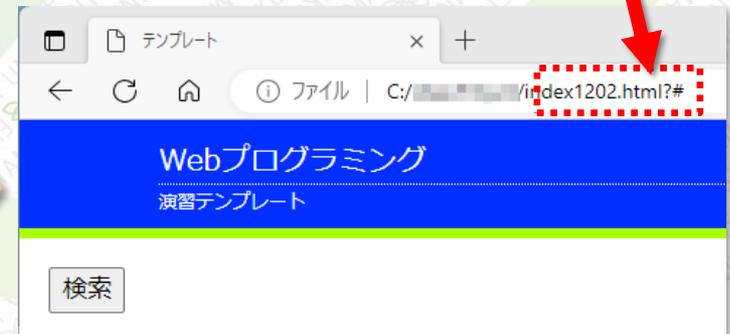
```
18 <section>
19   <form action="#" id="form">
20     <input type="submit" value="検索">
21   </form>
22 </section>
23 </div><!-- /.container -->
```

index1202.html

URLの一番後ろに「#」や「?#」がつく



クリックすると



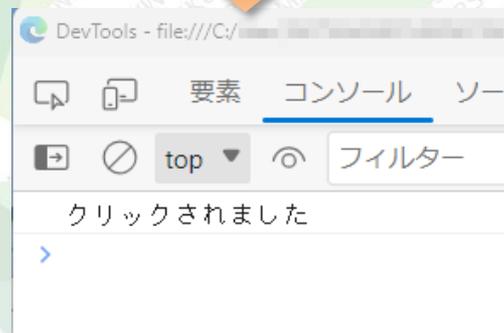
フォームの入力とイベント

- イベントとフォームのテキスト入力を使う
- まずはイベントをテストしてみる
 - [検索]ボタンが押されたらコンソールに「クリックされました」と表示する、「index1202.html」に下記を追記する

```
29 </footer>
30 <script>
31   'use strict';
32   document.getElementById('form').onsubmit = function() {
33     console.log('クリックされました');
34   };
35 </script>
```

index1202.html

[検索]ボタンをクリックすると ...



フォームの入力とイベント

■ イベントのタイミング

- ボタンをクリックする
- キーボードを操作する
- ページの読み込み、完了、切り替え など

■ フォーム

- `<form>~</form>` で囲まれている部分がフォーム

action属性には通常データを送信する先のURLを指定する
どこにも送信しない場合は「#」を指定するのが一般的

```
<form action="#" id="form">  
  <input type="submit" value="検索">  
</form>
```

検索 (submit) ボタンがクリックされたら、action属性のURLに入力内容が送信される

フォームの入力とイベント

■ onsubmitイベント

- 送信ボタンが**クリックされた直後**のタイミングで、入力内容がサーバに**送信される直前**に発生する

```
document.getElementById('form').onsubmit = function() {
```

(1)

<form id="form"> ~ </form>
を取得する

(2)

取得した<form>要素のonsubmit
イベントに関数を代入する

要素にイベントを設定する書式

```
取得した要素.onsubmit = function() {  
    ...処理...  
};
```

なお、イベントの関数は

- 関数名をつけない
- 戻り値がない

ので注意する

※onsubmitの部分は、イベントによって書き換える

フォームの入力とイベント

■ 入力内容を読み取って出力する

- [検索]ボタンをクリックしたタイミングでテキストフィールドの内容を読み取る、「index1202.html」に下記を追記する

```
18 <section>
19   <form action="#" id="form">
20     <input type="text" name="word">
21     <input type="submit" value="検索">
22   </form>
23   <p id="output"></p>
24 </section>
```

index1202.html

Webプログラミング
演習テンプレート

テキストフィールドが追加されている

検索

フォームの部品にname属性の名前がないと、受信したサーバ側のプログラムで処理ができないので注意

フォームの入力とイベント

■ 入力内容を読み取って出力する

- [検索]ボタンをクリックしたタイミングでテキストフィールドの内容を読み取る、「index1202.html」に下記を追記する

```
32 <script>
33   'use strict';
34   document.getElementById('form').onsubmit = function() {
35     const search = document.getElementById('form').word.value;
36     document.getElementById('output').textContent = `「${search}」の検索中...`;
37   };
38 </script>
```

index1202.html

`console.log('クリックされました');` は削除する

Webプログラミング
演習テンプレート

[検索]ボタンをクリックすると、一瞬表示されてすぐ消えてしまう・・・



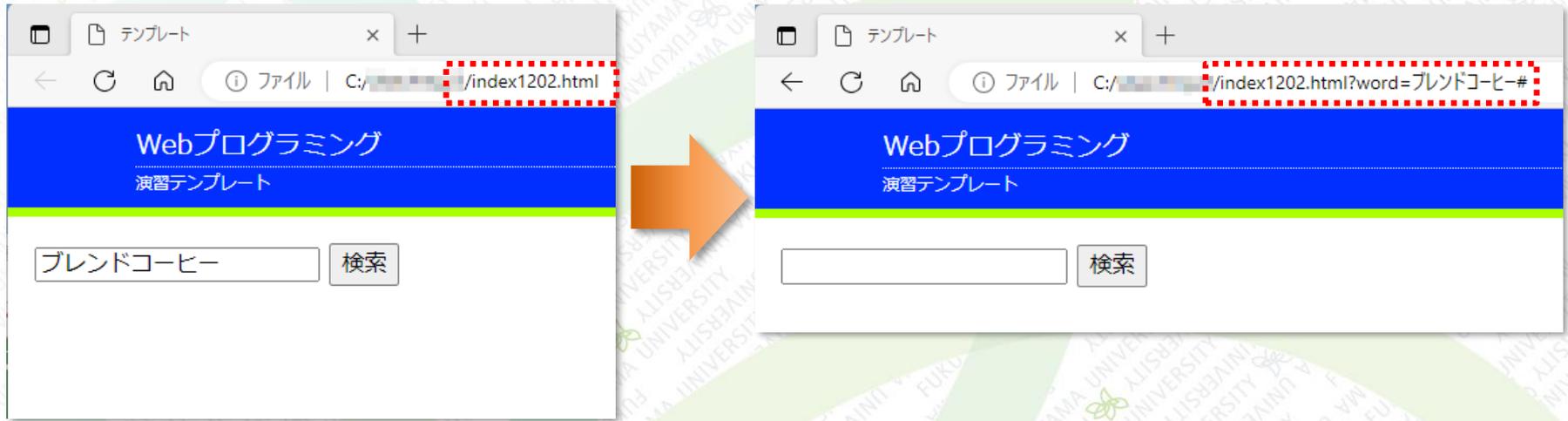
一瞬、“「ブレンドコーヒー」の検索中...”と表示されているはず

Webプログラミング
演習テンプレート

フォームの入力とイベント

■ 入力内容を読み取って出力する

- [検索]ボタンをクリックすると、URLが変わる



- URLに変更があると、ブラウザは「再読み込み」しようとする
- 『「ブレンドコーヒー」を検索中...』を出力した直後に表示しているページの再読み込みが行われたことが原因

フォームの入力とイベント

■ 入力内容を読み取って出力する

- 表示したページを再読み込みしないようにする、
「index1202.html」に下記を追記する

```
32     <script>
33         'use strict';
34         document.getElementById('form').onsubmit = function(event) {
35             event.preventDefault();
36             const search = document.getElementById('form').word.value;
37             document.getElementById('output').textContent = `「${search}」の
38         };
39     </script>
```

index1202.html

基本イベントをキャンセルするメソッド

```
event.preventDefault();
```

Webプログラミング

演習テンプレート

ブレンドコーヒー

検索

「ブレンドコーヒー」の検索中...

フォームの入力とイベント

■ 入力内容を読み取って出力する

■ 表示のしくみ

```
const search = document.getElementById('form').word.value;
```

<form id="form"> ~ </form>を取得する

```
<input type="text" name="word">
```

このフォームのテキストフィールドを指定している

フォームのテキストフィールドの中身である
"「ブレンドコーヒー」の検索中..."が格納されている

「`」はバッククォートであることに注意

```
document.getElementById('output').textContent = `「${search}」の検索中...`;
```

テンプレートリテラルの表示形式

演習 1

テンプレートの「[index.html](#)」をコピーして、ファイル名「[ensyu01.html](#)」に変更しなさい。次に、田中さんの英・数・国の点数を示す

```
let data = {shimei: '田中太郎', eng: 82, math: 75, japan:98};
```

のオブジェクトを下記の実行例と同じように表示しなさい。また、表の罫線も描画すること。

Webプログラミング			
演習テンプレート			
田中太郎	82 点	75 点	98 点

<実行例>

演習 2

テンプレートの「`index.html`」をコピーして、ファイル名「`ensyu02.html`」に変更しなさい。次に、テキストフィールドを2個作成し、そこに数値を入力して、[足し算]ボタンをクリックすると、2つの数値の足し算をして結果を表示したい。実行例のように表示されるようにしなさい。（テキストフィールドから入力された値は文字列形式なので、`parseInt()`で変換する必要がある）

Webプログラミング
演習テンプレート



Webプログラミング
演習テンプレート

3 5

(1)数値を入力

(2)クリック



Webプログラミング
演習テンプレート

3 5

3 + 5 = 8

(3)計算結果を表示

<実行例>