

JavaScript の e ラーニング教材の作成について

～資料編～

図 1 トップページ

JavaScriptオンライン教材

MENU

[トップページ](#)

[目次](#)

WORK MENU

- JavaScriptの概要と記述のルール
- イベントハンドラの利用
- 関数の基本
- 変数の宣言
- 変数の演算
- 配列
- 関数の引数
- 繰り返し処理
- 繰り返し処理と文字の表示
- 条件分岐-1

このサイトではJavaScriptについて勉強していきます。

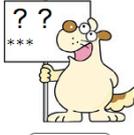
 このサイトではJavaScriptの概要や記述の仕方を説明します。STEP1~19では演習を含めながら徐々にJavaScriptを理解していきましょう。

※本教材を使用する前の注意点！

- ・ブラウザはSafariもしくはFirefox（推奨）を使用してください。Internet Explorerではうまく動作できない可能性があります。
- ・使用するOSによってはSafariかFirefoxのどちらかのみでしか動作できない場合があります。
- ・ブラウザのJavaScriptを有効にしてください。

↓こちらもJavaScriptを使用しています。↓

※ボタンをクリックすると占いができます。（おみくじができれば、JavaScriptは有効です。）
今日の運勢を占ってみよう！



[おみくじを引く](#)

図2 目次

JavaScriptオンライン教材

MENU

- トップページ
- 目次

WORK MENU

- 1. JavaScriptの概要と記述のルール
- 2. イベントハンドラの利用
- 3. 関数の基本
- 4. 変数の宣言
- 5. 変数の演算
- 6. 配列
- 7. 関数の引数
- 8. 繰り返し処理
- 9. 繰り返し処理と文字の表示
- 10. 条件分岐-1

~目次~

STEP01 JavaScriptの概要と記述のルール

- 1.1 JavaScriptの概要
- 1.2 JavaScriptを記述する場所
- 1.3 JavaScriptを記述する際のルール

STEP02 イベントハンドラの利用

- 2.1 イベントハンドラとは…?
- 2.2 クリックでJavaScriptを実行 onClick
- 2.3 マウスの移動でJavaScriptを実行 onMouseOver、onMouseOut
- 2.4 ページが表示された直後にJavaScriptを実行する onLoad
- 2.5 その他のイベントハンドラ

演習

STEP03 関数の基本

- 3.1 関数とは…?
- 3.2 関数の作成
- 3.3 関数名に指定できない文字
- 3.4 イベントハンドラから関数を呼び出す

演習

STEP04 変数の宣言

- 4.1 変数の宣言
- 4.2 変数に数値を代入する
- 4.3 変数に文字を代入する
- 4.4 変数の名前

図 3 STEP 1

STEP 01 JavaScriptの概要と記述のルール

JavaScriptは、一般的なプログラムのような動作をホームページ上で実現できるスクリプト言語です。JavaScript学習の第一歩となるステップ01では、JavaScriptの概要、およびJavaScriptの記述方法について学習します。

1.1 JavaScriptの概要

ホームページを作成するには、テキストエディタなどでHTMLを記述し、HTMLファイルを作成するのが一般的です。また、HTMLファイル内にCSSを記述し、ホームページ内の各要素について書式を指定することも可能です。

もちろん、これらの2言語だけでもホームページは十分に作成できます。ただし、このようにして作成されたホームページは“動きのないホームページ”にしかありません。一方、JavaScriptを利用してホームページを作成すると、閲覧者の操作に応じて“動きのあるホームページ”を作成できるようになります。

たとえば、『ボタンをクリックすると画像表示を変更する』などの演出をJavaScriptで実現することも可能です。

※それぞれのボタンをクリックすると表示される画像が変更されます。(図1-1)



川の画像 海の画像 空の画像 山の画像

また、「選択された項目について合計金額を算出する」など、実用的な仕組みをJavaScriptで実現する事も可能です。

※ボタンにチェックを入れると合計金額が自動的に算出されます。(図1-2)

図 4 STEP 2

JavaScriptオンライン教材

MENU

[トップページ](#)

[目次](#)

WORK MENU

- 1. JavaScriptの概要と記述のルール
- 2. イベントハンドラの利用
- 3. 関数の基本
- 4. 変数の宣言
- 5. 変数の演算
- 6. 配列
- 7. 関数の引数
- 8. 繰り返し処理
- 9. 繰り返し処理と文字の表示
- 10. 条件分岐-1

STEP 02 イベントハンドラの利用

マウスなどに応じてJavaScriptを実行するには、「どのタイミングでJavaScriptを実行するか?」をHTMLファイル内で指定しておく必要があります。このタイミングを指定するのがイベントハンドラです。ステップ02では、イベントハンドラの指定方法について学習します。

2.1 イベントハンドラとは…?

HTMLファイル内に記述したJavaScriptを正しく動作させるには、JavaScriptを実行するタイミングを指定しておく必要があります。このタイミング指定するのが**イベントハンドラ**です。イベントハンドラには、「マウスをクリックした時」「マウスポインタを移動したとき」など、操作内容ごとに個別のイベントハンドラ名が決められています。そして、HTMLのタグ内に以下のような形式でイベントハンドラを記述することにより、JavaScriptを実行するタイミング、および実行するJavaScriptを指定します。

イベントハンドラ名="実行するJavaScript"

2.2 クリックでJavaScriptを実行 onClick

ホームページにある要素をクリックした際にJavaScriptを実行させるには、**onClickイベント**を利用します。たとえば、画像をクリックした場合、そのimgタグにonClickイベントを追加し、その値にJavaScriptを記述します。

```

```

上の例では、「onClick」の部分がイベントハンドラ名、「alert('ねこの写真です!')」の部分がJavaScriptとなります。なお、**alert ()** はメッセージウィンドウを表示するJavaScript命令(メソッド)であり、メッセージとして表示する文字をカッコ内に「」(シングルコーテーション)で挟んで記述します。

図5 テキストエリア

※ここに直接入力できます。↓↓(テキストボックスのサイズは、右下の角をドラッグすると調節できます。)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja" lang="ja">

<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>繰り返し処理</title>

</head>

<body>
</body>

</html>
```

実行

図6 解答ページ

STEP08 繰り返し処理

演習(1)

以下のようなHTMLファイルをテキストボックスに作成します。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<title>繰り返し処理</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<script type="text/JavaScript">
<!--
function goukei0{
  var kai = 0;
  for (i=2 ; i<=100 ; i=i+2)           // 「i」の初期値は2、処理が実行されるたびに「i」を2ずつ増やす
    kai = kai + i;
  }
  alert("2、4、6、8、...、100の合計は" + kai + "です");
}
// -->
</script>
</head>

<body>
<h4>偶数の合計</h4>
以下のボタンをクリックすると、<br />
2、4、6、8、...、100を全て合計した結果を表示できます。<br />
<input type="button" value="計算結果" onClick="goukei0">
</body>
</html>
```

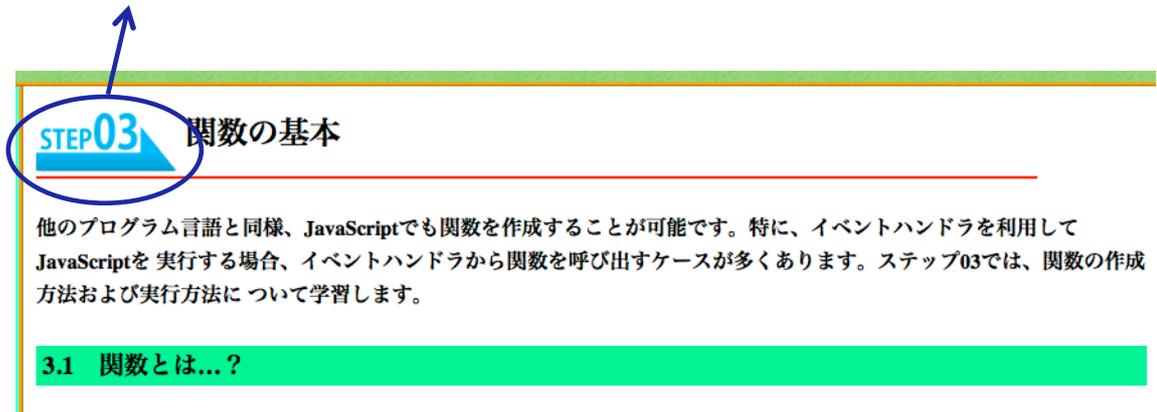
実行

(補足説明)

変数「kai」の初期値を指定する「var kai=0;」は、必ず関数内に記述しなければいけません。これを関数の外に記述すると最終的な計算結果が「kai」に保持されるため、ボタンを2回以上クリックしたときに正しくない計算結果が表示されます。ボタンを何回クリックしても正しい計算結果を表示するには、関数内で最初に「kai」の値を0にリセットする必要があります。

図7 「STEP～」の画像

この部分を自分で作成した



STEP 03 関数の基本

他のプログラム言語と同様、JavaScriptでも関数を作成することが可能です。特に、イベントハンドラを利用してJavaScriptを実行する場合、イベントハンドラから関数を呼び出すケースが多くあります。ステップ03では、関数の作成方法および実行方法について学習します。

3.1 関数とは...?

図8 JavaScriptの記述説明に関する画像

▼Sample1_1.html (抜粋)

```
<!--ここからJavaScript-->

<script type="text/JavaScript">
<!--
var total = 4800;
var a = new Array(3);
a[0] = 2000;
a[1] = 400;
a[2] = 500;
function goukei(i){
    if(document.f1.elements[i].checked == true){
        total = total + a[i];
    } else{
        total = total - a[i];
    }
    document.f1.elements[3].value = total;
}
//-->
</script>

<!--ここまでがJavaScript-->

</head>
<body>
<h1>体育館の利用料金</h1>
<div class="st1">基本使用料(2面、4時間) 4,800円</div>
<form name="f1">
<input type="checkbox" onclick="goukei(0)">夜間照明 (2,000円) <br />
<input type="checkbox" onclick="goukei(1)">バレーボール用ネット×2面 (400円)
<br />
<input type="checkbox" onclick="goukei(2)">バレーボール×10個(500円)<br />
<br />
合計金額 <input type="text" size="6" value="4800">円
</form>
</body>
</html>
```

onClick イベント

関数「goukei()」

チェックボックスをクリックすると、関数「goukei()」が実行される

図 9-1 変数の説明画像

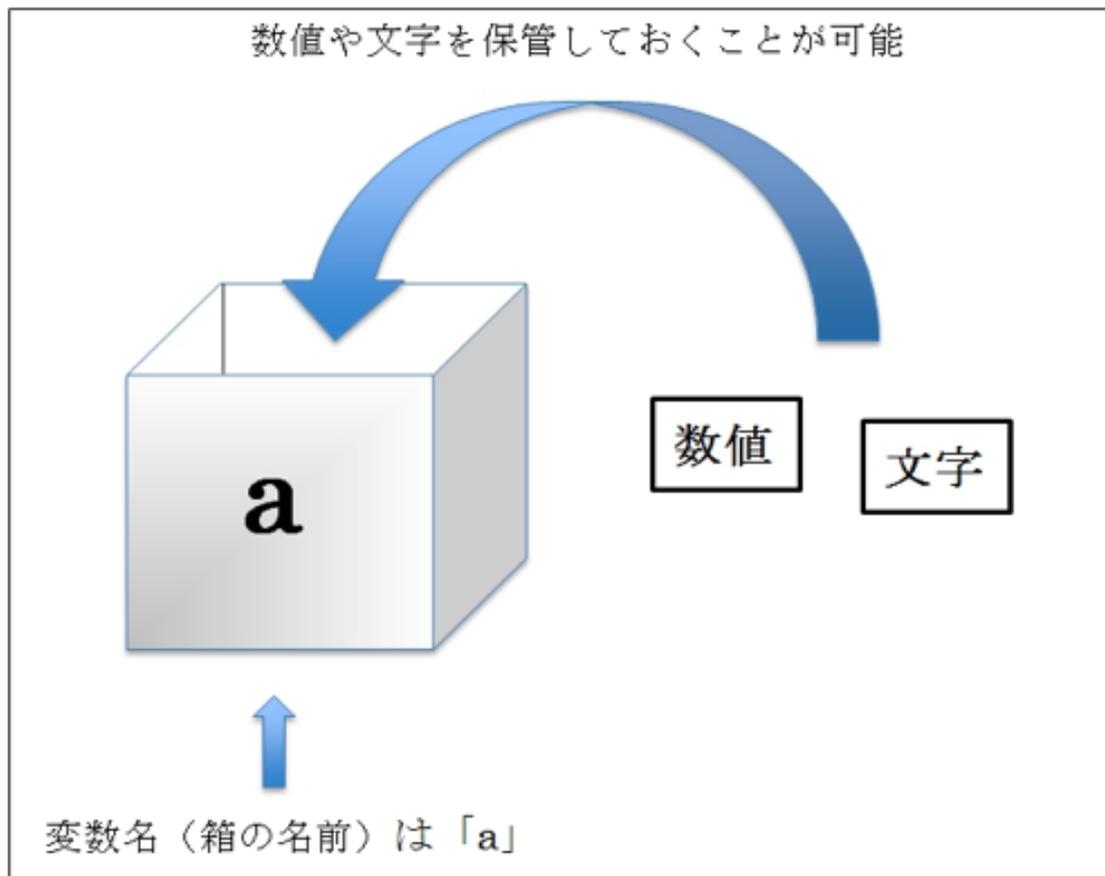


図 9-2 配列の説明画像

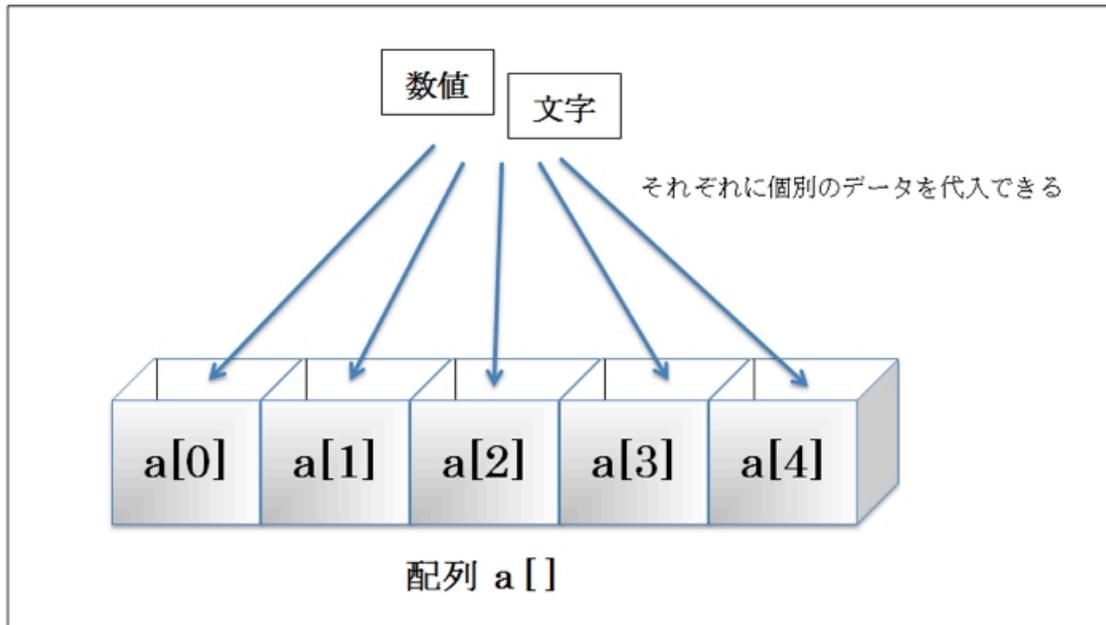


図 9-3 繰り返し処理の説明画像

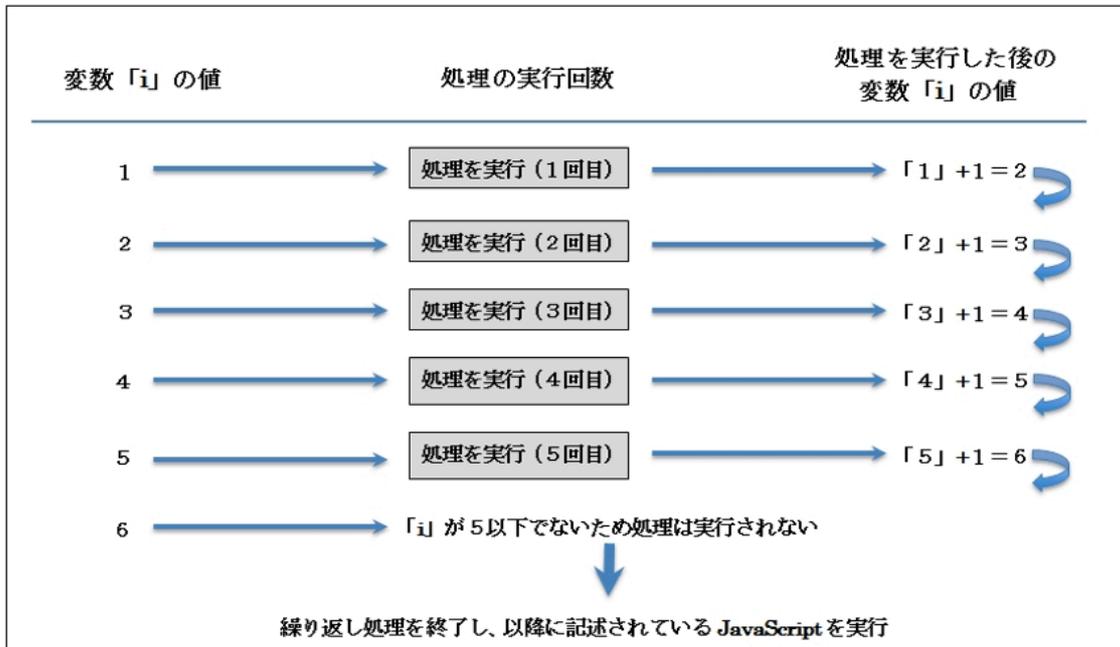


図 9-4 条件分岐の説明画像

■変数「i」と実行される処理の関係

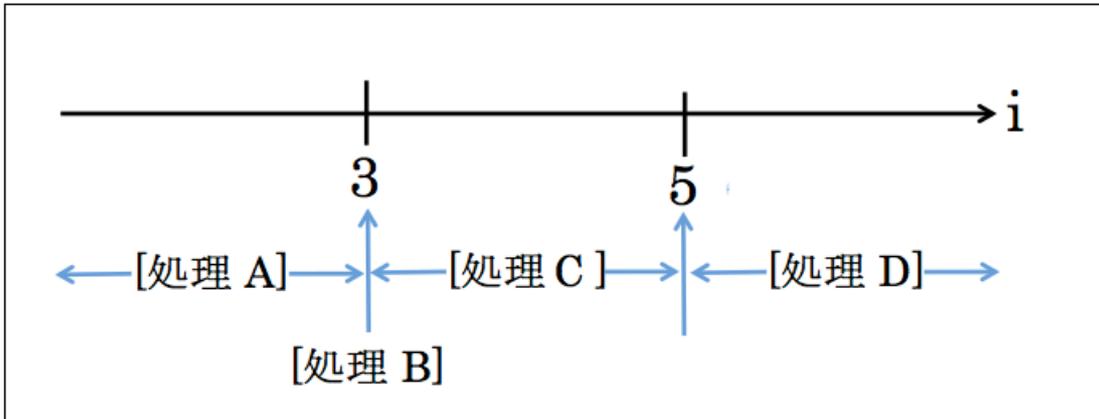


図10 サブウィンドウ

※ 'kakezan0()' 'kakezan1()' 'kakezan2()' という3つの関数を作成し、ボタンをクリックした際に対応する関数実行されるようにJavaScriptを記述します。

※各関数では、変数 a, b を用いて計算を行います。

※配列「a」「b」および「kai」を定義し、計算結果を出力します。

※表の各セルの横軸に添字を指定し、計算結果を出力します。

※なお、このJavaScriptが好ましいです。

※ここに直接入力して実行してください。

```
function kakezan0(a, b) {
    kai = a + b;
    alert("答えは" + kai);
}
function kakezan1(a, b) {
    kai = a * b;
    alert("答えは" + kai);
}
function kakezan2(a, b) {
    kai = a - b;
    alert("答えは" + kai);
}
// -->
</script>
</head>
<body>
<h4>配列の演算</h4>
<br />
<table border="1" style="width: 100%; text-align: center;">
<tr><th>添字</th><th>a</th><th>b</th><th>a×bを計算</th></tr>
<tr><th>0</th><td>5</td><td>33</td><td><input type="button" value="計算結果" onClick="kakezan0()" /></td></tr>
<tr><th>1</th><td>12</td><td>14</td><td><input type="button" value="計算結果" onClick="kakezan1()" /></td></tr>
<tr><th>2</th><td>18</td><td>65</td><td><input type="button" value="計算結果" onClick="kakezan2()" /></td></tr>
</table>
</body>
</html>
```

添字	a	b	a×bを計算
0	5	33	<input type="button" value="計算結果"/>
1	12	14	<input type="button" value="計算結果"/>
2	18	65	<input type="button" value="計算結果"/>

実行