ステップ30

JavaScript
ワークブック

もくじ

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。これらをまとめると、HTMLファイルは以下の2つの言語から構成されると考えられます。

- **HTML** …… ホームページに掲載する文章や画像を指定したり、ホームページ全体のレイアウトを指定したりする
- **CSS** …… ホームページ内の各要素について、色やサイズ、枠線などの書式を指定する

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 ”にしかなりません。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。

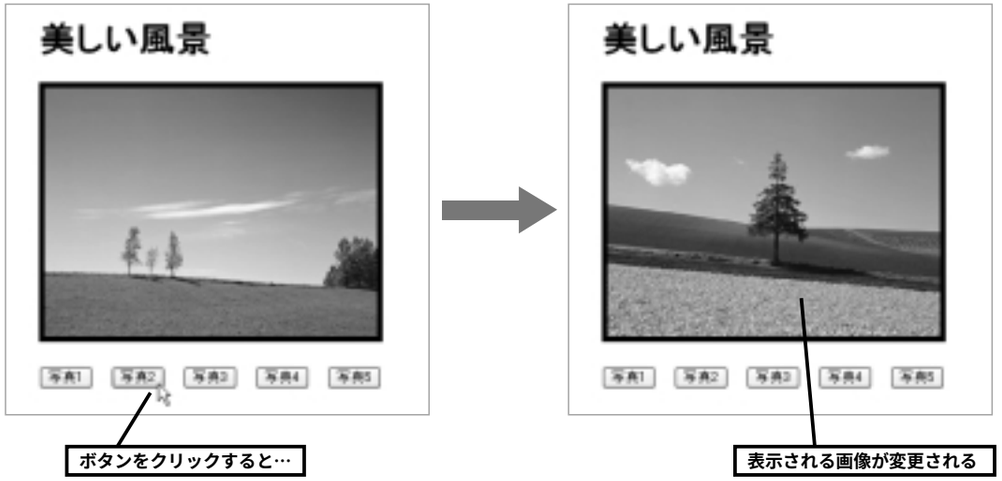


図1-1 JavaScriptを利用したホームページの演出

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で実現することも可能です。

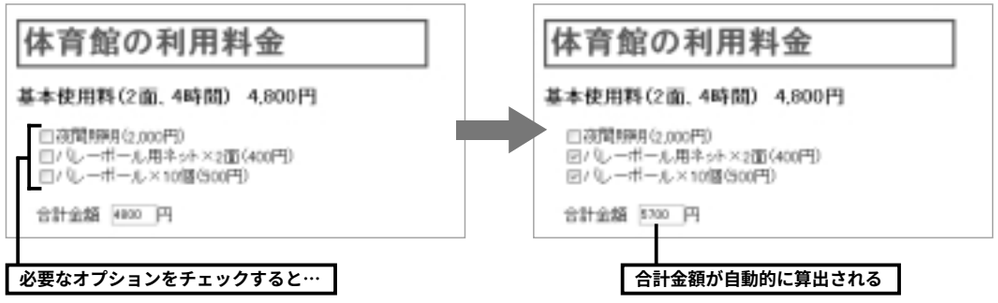


図1-2 JavaScriptを利用した実用的なホームページ

このように、閲覧者の操作に応じて、ホームページにさまざまな処理をプログラミングできるのがJavaScriptです。

1.2 JavaScriptを記述する場所

続いては、JavaScriptの記述方法について解説します。通常、JavaScriptを利用したホームページは、HTMLファイル内にJavaScriptを記述するのが一般的です。たとえば、図1-2の場合、HTMLファイルの内容は以下のようになります。

※現時点では、JavaScriptの記述内容を理解できなくても構いません。JavaScriptの記述内容については、後のステップで詳しく解説していきます。

▼Sample1_1.html

```
01 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
02  "http://www.w3.org/TR/html4/loose.dtd">
03
04 <HTML>
05
06 <HEAD>
07 <TITLE>体育館の利用料金</TITLE>
08 <META http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
09
10 <STYLE type="text/css">
11 <!--
12 H1{
13   border:solid 3px #993300;
14   padding:3px;
15   color:#993300;
```

CSS

（次ページへ続く）

```
16    width:400px;
17  }
18  DIV.st1{
19    font-size:14pt;
20    font-weight:bold;
21  }
22  FORM{margin-left:20px;}
23  -->
24  </STYLE>
25
26  <SCRIPT language="JavaScript">
27  <!--
28  var total = 4800;
29  var a = new Array(3);
30  a[0] = 2000;
31  a[1] = 400;
32  a[2] = 500;
33  function goukei(i){
34    if(document.f1.elements[i].checked == true){
35      total = total + a[i];
36    } else {
37      total = total - a[i];
38    }
39    document.f1.elements[3].value = total;
40  }
41  //-->
42  </SCRIPT>
43
44  </HEAD>
45
46  <BODY>
47  <H1>体育館の利用料金</H1>
48  <DIV class="st1">基本使用料（2面、4時間）　4,800円</DIV>
49  <FORM name="f1">
50  <INPUT type="checkbox" onClick="goukei(0)">夜間照明（2,000円）<BR>
51  <INPUT type="checkbox" onClick="goukei(1)">バレーボール用ネット×2面（400円）
    <BR>
52  <INPUT type="checkbox" onClick="goukei(2)">バレーボール×10個（500円）<BR>
53  <BR>
54  合計金額　<INPUT type="text" size="6" value="4800">円
55  </FORM>
56  </BODY>
57
58  </HTML>
```

CSS

JavaScript

このように、HTMLファイルには以下の3つの言語を含むことが可能です。

（HTMLファイルに必須となる言語）

- **HTML** …… ホームページに掲載する文章や画像を指定したり、ホームページ全体のレイアウトを指定したりする

（必要に応じてHTMLに追記する言語）

- **CSS** …… ホームページ内の各要素について書式を指定する
- **JavaScript** … 閲覧者の操作に応じて、ホームページの処理をプログラミングする

1.3 JavaScriptを記述する際のルール

これで、JavaScriptを記述する場所は理解できたと思います。続いては、JavaScriptを記述する際のルールについて解説します。命令文や各種指定などを半角英数字で記述する点はHTMLやCSSと同じですが、JavaScriptでは**大文字／小文字が区別される**ことに注意してください。そのほか、JavaScriptを記述する際のルールは以下のとおりです。

- **<SCRIPT>～</SCRIPT>の間にJavaScriptを記述する**

  HTMLファイル内にJavaScriptを記述する場合は、<SCRIPT>～</SCRIPT>内にJavaScriptを記述しなければいけません。また、**language属性**に**"JavaScript"**を指定し、タグ内の記述がJavaScriptであることを示しておく必要があります。

- **コメントを示す<!--　～　//-->を記述する**

  SCRIPTタグに対応していないWebブラウザでは、JavaScriptの記述がそのまま画面に表示されてしまう場合があります。これを防ぐには、JavaScriptの記述を**<!--**と**//-->**で挟み、JavaScriptの記述をコメント文として処理する必要があります。

```
<SCRIPT language="JavaScript">
<!--
  :
  :
（ここにJavaScriptを記述）
  :
  :
//-->
</SCRIPT>
```

- **JavaScriptは半角文字で記述する**

  JavaScriptの命令文や各種指定は**半角文字**で記述します。これらの記述に全角文字を使用すると、正しくJavaScriptが動作しません。

・大文字と小文字は区別される

JavaScriptでは、**大文字と小文字が別の文字として認識されます**。たとえば「ABC」「abc」「Abc」は、いずれも別の文字となります。HTMLやCSSのように、大文字／小文字を無視して記述できない点に十分注意してください。

・命令文の最後にセミコロン

JavaScriptでは、各文の最後に「**;**」（セミコロン）を記述し、文の区切りを明確に示す必要があります。CSSの場合と同様、改行は無視される点に注意してください。

○正しい記述

```
var a = 1;
var b = 2;
var c;
c= a + b;
```

×間違った記述

```
var a = 1
var b = 2
var c
c= a + b
```

・半角スペース、タブ文字の扱い

JavaScriptでは、**連続した半角スペースおよびタブ文字**は無視されます。このため、行頭に半角スペースやタブ文字でインデント（余白）を設け、JavaScriptを見やすく記述することも可能です。

```
function abc(){
  c= a + b;
  d= c * 1.05;
  alert(a);
}
```

ワンポイント

JavaScriptに対応していないWebブラウザへの対応

JavaScriptに対応していないWebブラウザに対して、ページにJavaScriptが含まれていることを示す場合は、**<NOSCRIPT>～</NOSCRIPT>**を利用します。このタグ内に記述された内容は、JavaScriptに対応していないWebブラウザにだけ表示されます。

```
<SCRIPT language="JavaScript">
<!--
  :
（JavaScriptの記述）
  :
//-->
</SCRIPT>
<NOSCRIPT>
このページにはJavaScriptが使用されています。<BR>
JavaScriptを有効にしてから閲覧してください。
</NOSCRIPT>
```


演習

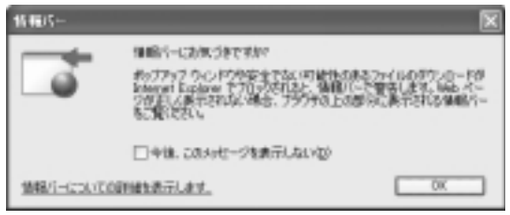
（1）P7の「Sample1_1.html」に示したHTMLファイルを作成してみましょう。

（2）演習（1）で作成したHTMLファイルをWebブラウザに表示し、チェックボックスの操作に応じて合計金額が変化することを確認してみましょう。

ワンポイント

Internet ExplorerでJavaScriptの実行を許可する

ハードディスクに保存されているHTMファイルをInternet Explorerで閲覧した場合、以下のようなメッセージが表示される場合もあります。



このメッセージが表示された場合は、Internet Explorerのセキュリティ保護によりJavaScriptが無効になっています。よって、記述したJavaScriptも実行されません。JavaScriptを実行させるには、以下のように操作してセキュリティ保護を解除する必要があります。

情報バーをクリックし、「ブロックされているコンテンツを許可」を選択します。

このようなウィンドウが表示されるので、「はい」ボタンをクリックします。すると、セキュリティ保護が解除され、JavaScriptが実行可能となります。

Step 02

イベントハンドラの利用

マウス操作などに応じて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に利用するのが**イベントハンドラ**です。イベントハンドラには、「マウスでクリックしたとき」「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際にJavaScriptを実行させる場合、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。

```
<IMG src="photo01.jpg" onClick="alert('北海道の風景写真です！')">
```

上の例では、「onClick」の部分がイベントハンドラ名、「alert('北海道の風景写真です！')」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「'」（シングルクォーテーション）で挟んで記述します。

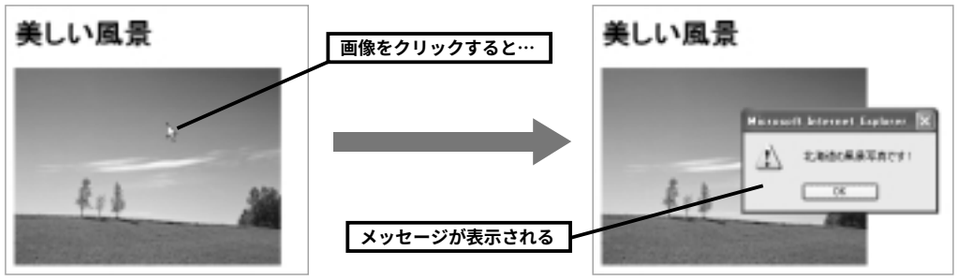


図2-1 onClickイベント

2.3 マウスの移動でJavaScriptを実行 onMouseOver、onMouseOut

要素の上にマウスポインタが来た際にJavaScriptを実行させるには、**onMouseOverイベント**を利用します。以下の例の場合、マウスポインタを画像の上に移動させると、「北海道の風景写真です！」というメッセージウィンドウが表示されます。

```
<IMG src="photo01.jpg" onMouseOver="alert('北海道の風景写真です！')">
```

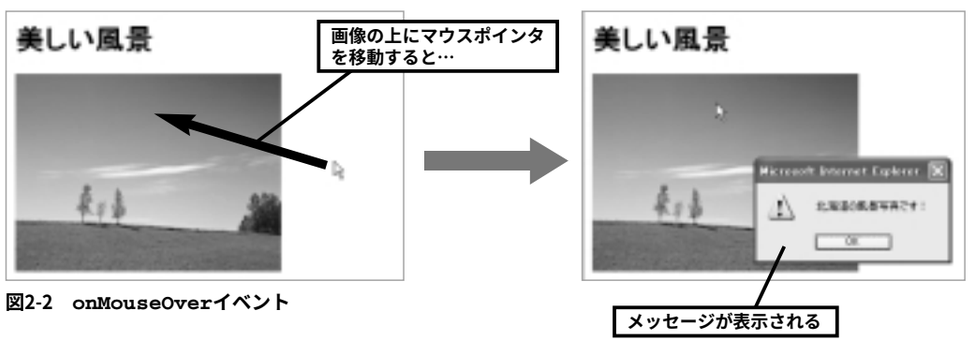


図2-2 onMouseOverイベント

これとは逆に、要素の外へマウスポインタを移動した際にJavaScriptを実行させるには、**onMouseOutイベント**を使用します。

```
<IMG src="photo01.jpg" onMouseOut="alert('北海道の風景写真です！')">
```

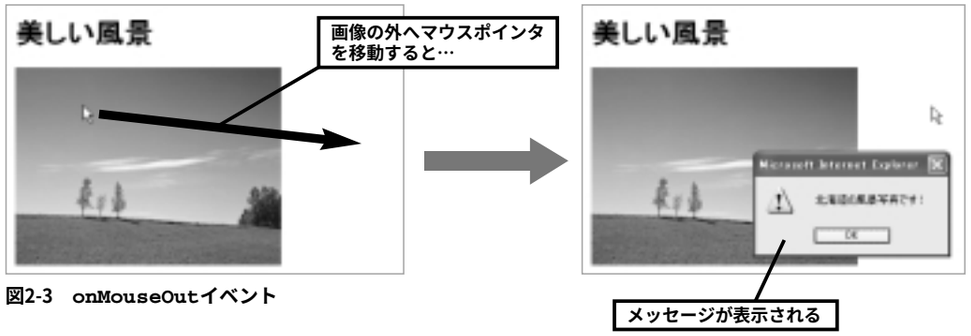


図2-3 onMouseOutイベント

2.4 ページが表示された直後にJavaScriptを実行する onLoad

ページが表示された直後にJavaScriptを実行する場合は、**onLoadイベント**を使用します。onLoadイベントはBODYタグに記述するのが普通であり、この場合、特に操作を行わなくても自動的にJavaScriptが実行されます。たとえば、次ページの例では、ホームページ全体が表示された直後に「ようこそホームページへ」というメッセージウィンドウが表示されます。

```
<BODY onLoad="alert('ようこそホームページへ')">
```

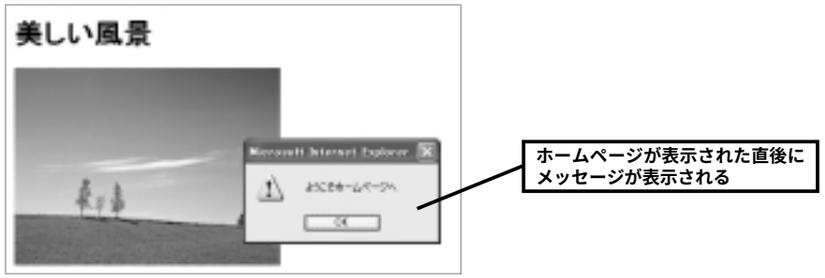


図2-4　onLoadイベント

2.5　その他のイベントハンドラ

これらのほかにも、JavaScriptを実行するタイミングに応じて以下のようなイベントハンドラが用意されています。

- **onAbort**
  Webブラウザの［停止］ボタンをクリックした場合などで、画像の読み込みが中断された場合にJavaScriptを実行します。

- **onBlur**
  その要素からフォーカスが離れた際にJavaScriptを実行します。
  ※フォーカスは、［Tab］キーやマウスのクリックで移動できます。

- **onChange**
  フォームの内容を変更した際にJavaScriptを実行します。

- **onError**
  ページや画像の読み込み時にエラーが発生した場合にJavaScriptを実行します。

- **onFocus**
  その要素にフォーカスが移動した際にJavaScriptを実行します。

- **onReset**
  フォームがリセットされた場合にJavaScriptを実行します。

- **onSelect**
  テキストボックスやテキストエリア内をクリックし、フォームへの入力が可能になった際にJavaScriptを実行します。

- **onSubmit**
  フォームのsubmitボタンをクリックした際にJavaScriptを実行します。

- **onUnload**
  別のページへ移動する際にJavaScriptを実行します。

演習

（1）以下のようなHTMLファイルを作成し、画像をクリックすると、メッセージウィンドウが表示されることを確認してみましょう。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
 "http://www.w3.org/TR/html4/loose.dtd">

<HTML>

<HEAD>
<TITLE>イベントハンドラの練習</TITLE>
<META http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
</HEAD>

<BODY>
<IMG src="tahiti01.jpg" onClick="alert('タヒチの写真です！')">
</BODY>

</HTML>
```

※この演習で使用する画像「tahiti01.jpg」は、以下のURLからダウンロードできます。
http://www.cutt.co.jp/book/html_818.html

（2）onClickイベントをonMouseOverイベントに変更し、メッセージウィンドウが表示されるタイミングがどのように変化するかを確認してみましょう。

（3）onMouseOverイベントをonMouseOutイベントに変更し、メッセージウィンドウが表示されるタイミングがどのように変化するかを確認してみましょう。

（4）ページが表示された直後にメッセージウィンドウが表示されるようにイベントハンドラを書き換えてみましょう。