

## 第15章 JavaScriptを活用しよう

本章では JavaScript の簡単な例を試してみます。JavaScript そのものはかなり複雑なオブジェクト指向の言語です。数多くのオブジェクト<sup>1</sup>とそれに付随したメソッドを持っており、役に立ちそうな所だけ抜粋してもかなりの分量になります。ここでは小さな例題を示し、その雰囲気を掴み取って貰えば十分です。

### 15.1 one liner な JavaScript

最近良く見かけるのは、Web ページ内のボタンをクリックすると、ツールバーもステータスバーもないのっぺら坊のウィンドウが開き、そこに別の Web ページが表示されるというものです。

この機能を抜粋した例を図 15.1 に示します。この HTML ファイルを作成してブラウザで表示すると、フォーム内に「ウィンドウを開く」というボタンが出現します。これをクリックすると図 15.3 の open\_window2.html が別ウィンドウに表示されます(図 15.3)。

```
<html>
<title>別 Window を開く</title>
<body>
<form>
<input type="button" value="ウィンドウを開く" onclick="window.open('open_win
dow2.html', 'win2', 'scrollbars=no,toolbar=no,width=200,height=50');">
</form>
</body>
</html>
```

図 15.1: 別ウィンドウを表示する

この open\_window.html は図 15.4 のように書くこともできます。このように

```
<script language="javascript">
...
</script>
```

に囲まれた部分や、”onclick=”以降に記述された部分で JavaScript を記述しています。

<sup>1</sup>変数に関数(メソッド)が融合したようなもの。

```
<html>
<title>別 Window</title>
<body>
別 Window 画面です。
</body>
</html>
```

図 15.2: open\_window2.html

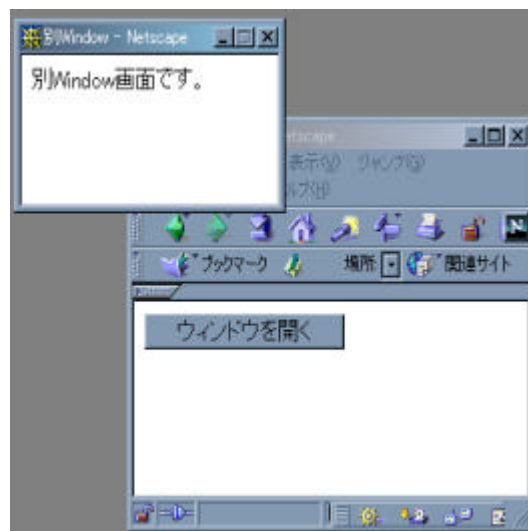


図 15.3: 別 Window を表示

```
<html>
<title>別 Window を開く</title>
<script language="javascript">
function openwin(uri)
{
window.open(uri, "win2", "scrollbars=no,toolbar=no,width=200,height=50");
}
</script>
<body>
<form>
<input type="button" value="ウィンドウを開く" onclick="openwin('open_window2.html')">
</form>
</body>
</html>
```

図 15.4: 別 Window を表示する (簡易版)

## 15.2 JavaScript を書いてみよう

もうすこし、長いプログラム (スクリプト) らしいものを書いてみましょう。まずは時刻を表示する JavaScript です (図 15.5)。

これは HTML ファイルがブラウザに読み込まれる時に一度だけ `document.write(gettime())` (現在時刻を取得し、Web ページ内に表示する) を実行してして終了する JavaScript です。時刻を変化させるためには再読み込みを行う必要があります (図 15.6)。

そこで、自動的に時刻を更新させるためにタイマ (Timer) を使います。

図 15.5 中の

```
document.write(gettime());
```

という部分を

```
window.setInterval("window.status = gettime()", 1000);
```

と書き換えて下さい。こうすると、この場合は 1 秒ごとに指定の関数 (`gettime()`) を繰り返し実行します。すると、図 15.7 のように、ステータスバー (ウィンドウ下部) に時刻が表示され、刻々と変化していくのが確認できます。

今度は現在時刻を Web ページ内に表示させてみましょう。そのためにフォームを利用します (図 15.8)。どこを変更したか分かりますか？

実行すると図 15.9 のように、テキストボックスに時刻が表示されます。

```
<html>
<title>時計</title>
<script language="javascript">
function gettime()
{
    time = new Date();
    display_time = "";
    display_time += String(time.getFullYear()) + "年";
    display_time += time.getHours() + "時";
    display_time += time.getMinutes() + "分";
    display_time += time.getSeconds() + "秒";
    return display_time;
}

document.write(gettime());

</script>
<body>
</body>
</html>
```

図 15.5: 日付を表示する



図 15.6: リロードすると時刻が変わる

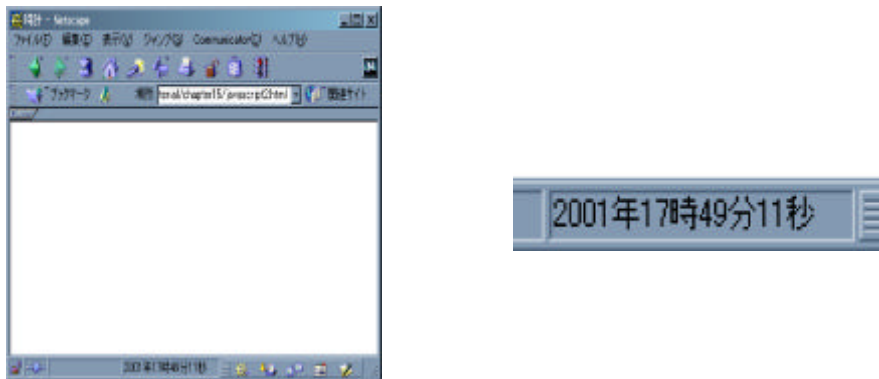


図 15.7: ステータスバーに時計を表示

```

<html>
<title>時計</title>
<script language="javascript">
function getTime()
{
time = new Date();
display_time = "";
display_time += String(time.getFullYear()) + "年";
display_time += time.getHours() + "時";
display_time += time.getMinutes() + "分";
display_time += time.getSeconds() + "秒";
return display_time;
}

window.setInterval("document.forms[0].time.value = getTime()", 1000);

</script>
<body>
<form>
<input type="text" name="time" maxlength=18>
</form>
</body>
</html>

```

図 15.8: 日付を表示する

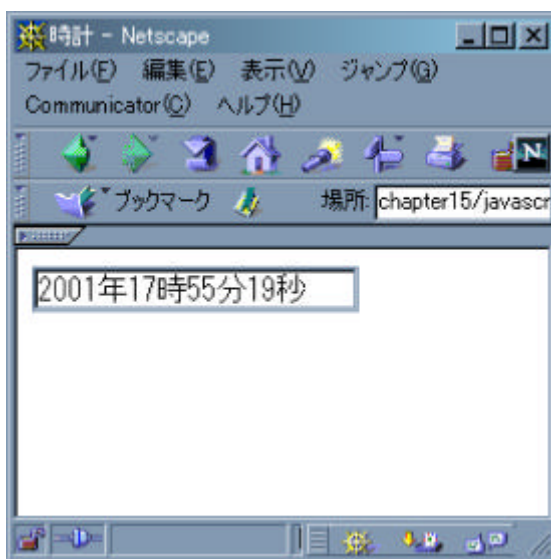


図 15.9: フォームに時計を表示

### 15.3 フォーカスを使ってボタンを変える

最近ではマウスカーソルをアンカー (リンクが張られた場所) に移動すると、その画像や文字列が変化するような Web ページがあります。そこでも JavaScript が活躍しています。

その原理を図 15.10 で確認してみてください。これは図 15.11 のように動作します。

<A> タグ内の "onmouseover=..." では、マウスカーソルがその位置に移動してきたときに実行される関数を指定しています。ここでその時々に合わせて表示される図を変化させれば良いわけです。このように、何か起こったときにその事 (イベントと呼びます) の対応した動作を行うことをイベントドリブン (event driven) と呼び、Web に限らず、GUI の基本となっています。

図 15.10 の例では、最初は "green.gif" (緑の正方形の画像) を表示し、マウスカーソルが来ると、"red.gif" (赤の正方形の画像) を表示します。マウスカーソルが外れると元の "green.gif" を表示しています。

#### 練習問題

1. ボタンを押すと別 Window に時計を表示させる JavaScript を書け。実行画面は図 15.12 のようになる。

```
<html>
<script language="javascript">
var col = 1; // 1..green, 0..red
function change_color(imange_name)
{
    if(col == 1)
    {
        col = 0;
        document.image1.src = "red.gif";
    }
    else
    {
        col = 1;
        document.image1.src= "green.gif";
    }
}
</script>
<body>
<a href="http://www.kanto-gakuin.ac.jp/" onmouseover="change_color()" onmou
seout="change_color()">
</a>
</body>
</html>
```

図 15.10: フォーカスに応じて変化する画像

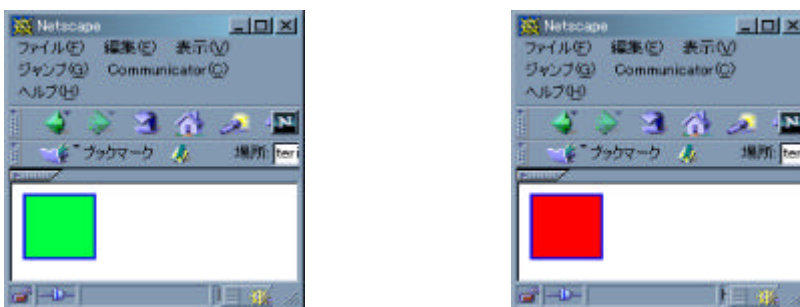


図 15.11: (左) フォーカスがない時 (右) フォーカスが移ってきた時

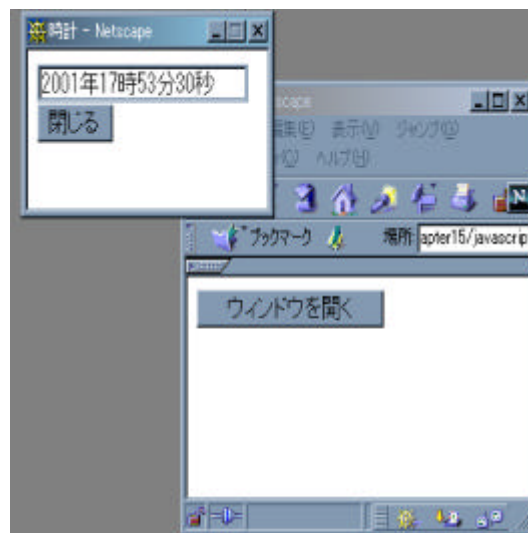


図 15.12: 別 Window に時計を表示