# 応用情報処理 No. 6

## JavaScript

Netscape社が自社のWebブラウザのNetscape Navigator用に開発したプログラミング言語LiveScriptは、Webページに対するプログラミング言語で、ページの内容を操作することで、ダイナミック（動的）でインタラクティブ（相互作用的）なWebページにすることができた。Sun Microsystem社が開発したプログラミング言語Javaのブームにのり、Netscape社はJavaと似た書き方を採用できる権利を買い取り、JavaScriptという名前に変えた。JavaScriptとJavaは書き方が似ているという以外は、全く別のプログラミング言語である。

　その後、他社のWebブラウザでもJavaScriptをサポートするようになったが、ブラウザごとに動作したりしなかったりしたので、標準化が求められ、ECMA(ヨーロッパ電子計算機工業会：European Computer Manufacturer Association)で標準化された（ECMAScript）。現在は、ほとんどのWebブラウザは、JavaScriptを理解して動作させることができるようになっている。JavaScriptはWebブラウザというソフトウェアの中で動作するものであることに注意しよう。従って、JavaScriptを動作させるには、Webブラウザは必要だが、WWWサーバは必要ない。ネットに接続されていないためにWWWサーバにアクセスできなくても、ローカルに保存したHTMLファイルでJavaScriptを動作させることができるのはこの理由による。

JavaScriptでは以下のようなことができる。

1. ページの中の要素（HTML要素、つまり文字や画像など）の内容（属性）を変更する。
2. ページの中の要素のスタイルを変更する。
3. ページの中の要素を動かす。
4. ページの中の要素を非表示/表示する。
5. フォーム送信データのチェック。
6. クッキーの読み書き。（クッキーは、自動的にブラウザからサーバへ送られる情報である）
7. クリックなどのユーザのアクションに対応できる。
8. サーバと通信できる。

これらを実現するための基本事項として、変数と演算、制御文、関数、JavaScriptで用意されているオブジェクトなどについて学んでいこう。

## 2. JavaScriptの実行方法

JavaScriptで書かれたプログラムは、HTMLファイルに埋め込むか、別のファイルとして用意してHTMLファイルへ読み込むのが普通である。

埋め込む場合の例：

[hello.html]

<html>

<head>

<title>Hello, world!</title>

<script type="text/javascript">

<!--

document.write(‘Hello, ’);

//-->

</script>

</head>

<body>

World!

</body>

</html>

document.write(‘Hello, ’);という部分がJavaScriptのソースコードである。他の部分はHTMLのソースコードである。HTMLの<script type=”text/javascript”>タグ要素で囲まれた部分にJavaScriptのソースコードのソースコードを記述することができる。

「<!--」と「//-->」は、JavaScriptを理解しないブラウザで、JavaScriptのソースコードが画面に表示されないようにするための“おまじない”である。

読み込む場合の例：

[hello\_src.html]

<html>

<head>

<title>Hello, world!</title>

<script type="text/javascript" src=”hello.js”></script>

</head>

<body>

World!

</body>

</html>

[hello.js]

document.write(‘Hello, ’);

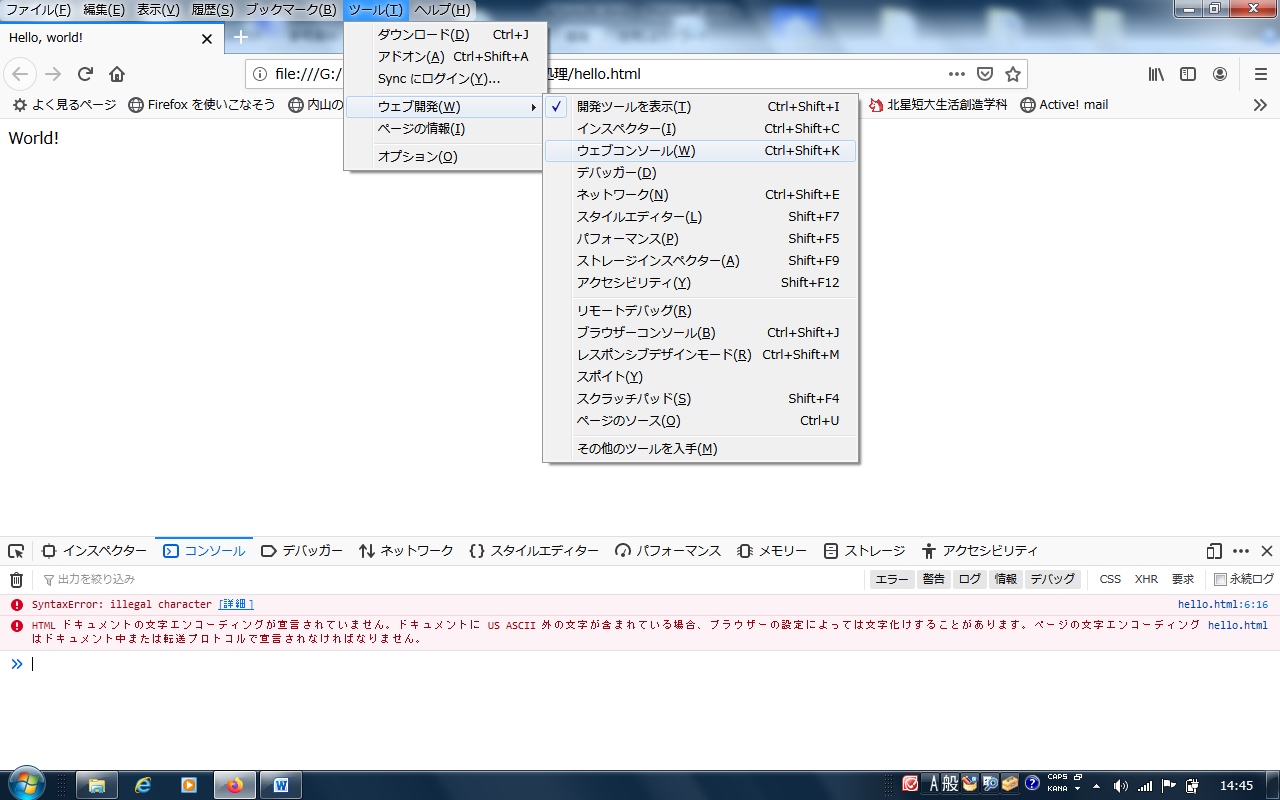
かつてはFirefoxやInternet Explorerなどのブラウザでは、ロケーションバーにURLの代わりに、「javascript:JavaScriptのソースコード」と入力すると、そのJavaScriptのプログラムを表示されているページに対して実行することができたが、セキュリテイ上の理由で、最近のブラウザではこれができなくなっている。

## 3. Webコンソール（Firefox）

JavaScriptはWebブラウザで動作するので、Firefoxなどの一部のWebブラウザでは、JavaScriptのソースコードのエラーを表示する機能を持っているものがある。Firefoxの場合は、Webコンソールを開くと、エラーが表示される。

Webコンソールを開く方法

① メニューバー《ツール》→《Web開発》→《Webコンソール》を選択。



Webコンソール

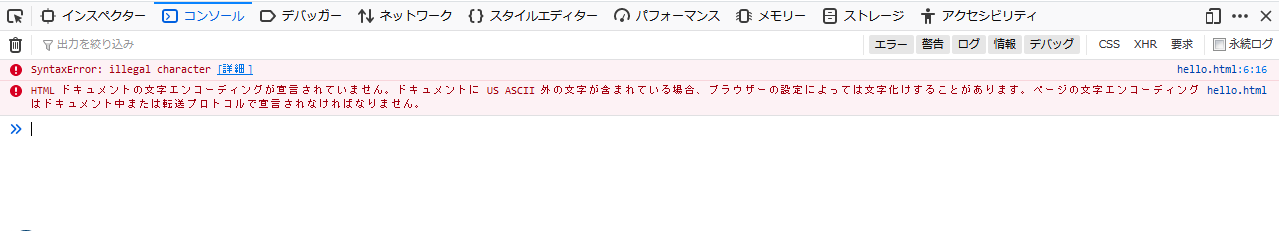
コマンドライン

メッセージ表示ペイン

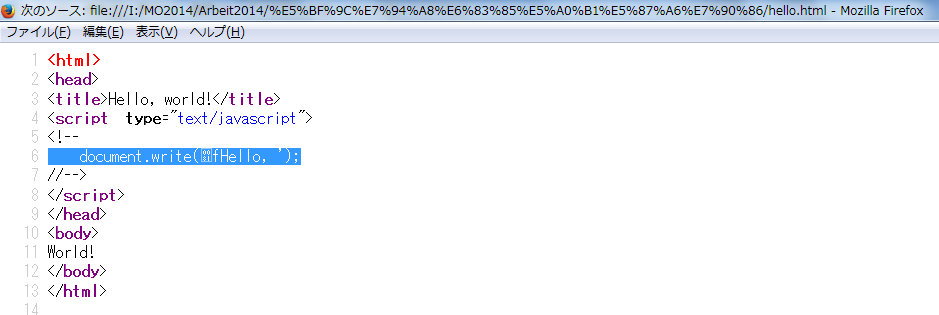
Javascriptのエラー/警告 表示選択ボタン

### Webコンソールの使い方

1. [消去]ボタンをクリックして、エラー（、警告）メッセージを消去して初期化する。
2. Webブラウザで、チェックしたいHTMLファイルの再読み込みをする。
3. エラーがあると、エラーコンソールに表示される。（下の例では、”SyntaxError: illegal character”と表示されている。）



1. 上の例の場合、右端の「hello.html: 6」をクリックすると、hello.htmlのソースコードが表示されて、エラーの行が反転表示される。



【重要】JavaScriptはそのコードが記述されているHTMLファイルがWebブラウザに読み込まれたとき、Webブラウザがコードを解読して実行される。従って、コードを書き換えたときは、HTMLファイルを上書き保存し、Webブラウザで再読み込みをして実行し、動作を確かめるという手順になる。

## 3. JavaScriptを使ったアドベントカレンダーの例

クリスマスまでの約4週間をアドベント（待降節）という。この時期に、日付を開くと小さな絵やお菓子などが入っているアドベントカレンダーというものをつくって、毎日をクリスマスの準備としていくことを行うのはキリスト教文化の一つである。

アドベントカレンダーをまねて、日付をクリックするとクリスマスらしい絵が表示されてその絵をダブルクリックすると絵が閉じるというカレンダーをJavaScriptで作成してみた。このようにJavaScriptでwebページをクリックしたりダブルクリックしたりすることで表示が変わるインタラクティブ（相互作用的）なwebページを作成することができる。このインタレクティブなwebページの動作を確認してみよう。

1. Firefoxを起動して、 http://www.ipc.hokusei.ac.jp/~z00328/cgi01/adv.htmlのページを開く。
2. 数字をクリックしてみる。閉じるときは、ダブルクリックする。

## 4. 実験：グルグルJavaScript

上のアドベントカレンダーのページは、たくさんの画像が組み合わさってできている。これらの画像の表示位置をJavaScriptを使って一定時間ごとに変更することで、動かしてみよう。

1. Webコンソールのコマンドラインに以下のJavaScriptのコードを入力し、[Enter]を押す。

R=0;x1=.1;y1=.05;x2=.25;y2=.24;x3=1.6;y3=.24;x4=300;y4=200;x5=300;y5=200;DI=document.images;DIL=DI.length;function A(){for(i=0;i<DIL;i++){DIS=DI[i].style;DIS.position='absolute';DIS.left=Math.sin(R\*x1+i\*x2+x3)\*x4+x5;DIS.top=Math.cos(R\*y1+i\*y2+y3)\*y4+y5;}R++;}setInterval('A()',5);void(0);

* + このコードを手で入力するのは大変なので、http://www.ipc.hokusei.ac.jp/~z00328/adjyou/index.htmlのページのコードをコピー＆ペーストして良い。初めてコマンドラインにペーストするときは警告が表示されるので、コマンドラインに「allow pasting」(貼り付けを許可)と入力をして[Enter]を押してから、もう一度貼り付けて実行すること。

このプログラムのソースコードの部分をセミコロン（;）で改行して整理すると以下のようである。理解の助けとなるようにコメント（//以下の部分）を追加しておいた。

コメントは、本来はメモをつけたいときに使うものだが、Webブラウザは//から行末までをコメントとして無視するので、そこに何を書いてもJavaScriptのエラーにはならない。より重要なコメントの活用法は、ソースコードの一つの命令文を一時的に実行しないようにしたいときに、その命令文を削除しないで、先頭に//をつけてコメントにするという使い方である。その命令文を復活したいときは、先頭の//を削除するだけでよいからである。

JavaScriptのソースコードを書く場合の一般的な約束ごととして、代入などの一つの命令文の最後にセミコロン（;）をつける必要がある。JavaScriptのこのセミコロンは、日本語の文章の終わりをしめす句点（。）に相当するものである。Webブラウザは、ソースコードが改行されているか否かではなく、このセミコンがあるところで命令文の終わりと判断するので、上のようにセミコロンさえ打ってあれば、改行せずに続けてソースコードを書いても、エラーにはならない。しかし、人間には読みづらいので、改行や空白、タブを入れて、エラーにならない範囲で読みやすくソースコードを書くことが求められる。

R = 0; //関数Aが呼び出されるたびに1ずつ増加して、画像の表示位置を変えるための変数

x1 = .1; //画像の軌跡を決定するパラメーターの値をセット

y1 = .05;

x2 = .25;

y2 = .24;

x3 = 1.6;

y3 = .24;

x4 = 300;

y4 = 200;

x5 = 300;

y5 = 200;

DI = document.images; //Webページに含まれる画像オブジェクトの配列

DIL = DI.length; //画像オブジェクトの個数

function A()

{

for( i = 0; i < DIL; i++ ){ //個々の画像オブジェクトの表示位置を指定

DIS = DI[ i ].style;

DIS.position = 'absolute';

DIS.left = Math.sin( R \* x1 + i \* x2 + x3 ) \* x4 + x5;

DIS.top = Math.cos( R \* y1 + i \* y2 + y3 ) \* y4 + y5;

}

R++;

}

setInterval('A()',5 ); //5ミリ秒間隔で関数A()を実行

void(0);

## 5. JavaScriptの変数

変数は、数値や文字列を代入することができる箱のようなものである。JavaScriptでは、最初に変数を使用するときに、その変数名を宣言する。宣言の仕方は、以下のようである。

var x;

var x, y;

var x = 120;

文字列のデータ（文字リテラル）は二重引用符（“）または引用符（‘）で囲む必要がある。

=は代入演算子で、変数xにabcという文字列のデータ（文字リテラル）を代入したいときは、この=を使って、

x = “abc”;

と書く。

**Do!** variable.htmlという名前で以下のHTMLファイルを作成せよ。

<html>

<head>

<title>変数と演算</title>

</head>

<body>

<h1>JavaScriptの変数</h1>

<script type="text/javascript">

// 文字列

var str;

str = '今日は雨です。'; //文字列は、'で囲みます

document.write( str );

</script>

</body>

</html>

JavaScriptでは、文字列のデータを表すために、’で囲んでも、”で囲んでも良い。文字列データの始まりと終わりを示す対という意味で全く同じで、どちらを使っても良い。

document.write( "<br>");

+演算子は、文字列のデータに作用させると、それらを文字列として結合する。[[1]](#footnote-1)

str = str + '明日は晴れです。';

document.write( str );

document.write( "<br>");

数値データは、先頭に0xを付けると16進数を表す。

var a = 0x15;

document.write( a );

document.write( '<br>');

10進数表記と16進数表記の関係：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10進数 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 16進数 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10進数 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 16進数 | 9 | A | B | C | D | E | F | 10 | 11 |

## 6. JavaScriptの数値演算

数値に関する演算には以下のものがある。

|  |  |
| --- | --- |
| 演算 | 意味 |
| x + y | xとyを足す |
| x - y | xからyを引く |
| x \* y | xとyを掛ける |
| x / y | xをyで割る |
| x % y | xをyで割った余り |
| ++x | xの値を1増やす |
| --x | xの値を1減らす |
| x++ | xの値を1増やす（代入後） |
| x-- | xの値を1減らす（代入後） |

**Do!** variable.htmlに以下のコードを追加せよ。

<h1>JavaScriptの演算</h1>

<script type="text/javascript">

var b, c; //二つ以上の変数を宣言するときは、コンマで区切ります

b = 3;

c = b / a;

document.write( '割り算： ' + b + ' / ' + a + ' = ' + c );

document.write( '<br>');

</script>

**Do!** variable.htmlに以下のコードを追加せよ。

乗算は、\*を使う。

c = b \* a;

document.write( '掛け算： ' + b + ' \* ' + a + ' = ' + c );

document.write( '<br>');

加算は、＋を使う。

c = b + a;

document.write( '足し算： ' + b + ' + ' + a + ' = ' + c );

document.write( '<br>');

減算は、－を使う。

c = b - a;

document.write( '引き算： ' + b + ' - ' + a + ' = ' + c );

document.write( '<br>');

剰余演算子は、%を使う。

a = 5; b = 17;

c = b % a;

document.write( b + ' / ' + a + ' の余りは ' + c );

document.write( '<br>');

＋＋は１だけ増加させる。

document.write('<h2>増分演算子++</h2>');

c = b % ++a;

document.write( b + ' / ' + a + ' の余りは ' + c );

document.write( '<br>');

代入の右辺で、＋＋を変数の後に書くと、代入した後に増分する。

document.write('<h2>増分演算子++</h2>');

c = b % a++;

document.write( b + ' / ' + a + ' の余りは ' + c + '?');

document.write( '<br>');

## 7. JavaScriptのダイアログボックス

JavaScriptには、警告ダイアログ、入力ダイアログ、確認ダイアログの三種類のダイアログボックスを表示する関数[[2]](#footnote-2)がある。

**Do!** この節の学習のために以下の準備作業をしよう。

1. variable.htmlのコピーを作成し、JSdialog.htmlという名前に変更する。
2. JSdialog.htmlを秀丸エディタで開く。
3. タイトル(<title>…</title>)を、「JavaScriptのダイアログボックス」に書き換える。
4. 本分の見出し１（<h1>…</h1>）の一つを「JavaScriptのダイアログボックス」に書き換える。
5. 書き換えた見出し１と、一組の<script type="text/javascript"></script>だけ残して、本文中の他のコードはすべて削除する。

alert(メッセージ)

alert()関数は、メッセージを表示する警告ダイアログボックスを表示する。

ダイアログボックスの表示中は、JavaScriptの実行が中断されるので、実行の途中で変数の値がどうなっているか調べるときに使うと便利である。

**Do!** JSdialog.htmlのscript要素（<script type="text/javascript">～</script>）に以下のコードを追加せよ。

alert(‘文字を入力してください。’);

prompt(メッセージ, 初期値)

prompt()関数は、メッセージと初期値の書かれたテキストボックスがある入力ダイアログボックスを表示する。テキストボックスに書かれた値が返り値になる。

**Do!** JSdialog.htmlのscript要素（<script type="text/javascript">～</script>）に以下のコードを追加せよ。

var str = “?”;

str = prompt( str + ‘を削除して、何か文字列を入力してください。キャンセルはしないでください。’, str );

confirm(メッセージ)

confirm()関数は、[OK]と[キャンセル]の二つのボタンをもった確認ダイアログボックスを表示する。返り値は、[OK]をクリックした場合はtrue、[キャンセル]をクリックした場合はfalseである。

**Do!** JSdialog.htmlのscript要素（<script type="text/javascript">～</script>）に以下のコードを追加せよ。

var ret = confirm( ‘「’ + str + ‘」を表示して良いですか？キャンセルしても構いません。’);

if( ret ){

document.write(‘<p>’ + str + ‘</p>’);

}else{

alert(‘表示しませんでした。’);

}

ここで、if(条件){ 処理1 }else{ 処理2 }という条件分岐の制御文の意味は、条件が真(true)のときは処理1を実行し、条件が真ではない、すなわち偽(false)のときは処理2を実行せよということである。

　今の場合、confirm()関数の返り値がretに格納されるので、条件はretを使用している。retが真の場合はdocument.write(‘<p>’ + str + ‘</p>’);で入力された文字列をページに表示し、retが偽の場合はalert(‘表示しませんでした。’);で警告ダイアログボックスを表示するようになっている。

## 8. 練習問題：消費税の計算

**Do!** この練習問題をやるために以下の準備作業をしよう。

1. variable.htmlのコピーを作成し、tax8.htmlという名前に変更する。
2. tax8.htmlを秀丸エディタで開く。
3. タイトル(<title>…</title>)を、「8%消費税価格」に書き換える。
4. 本分の見出し１（<h1>…</h1>）の一つを「消費税の計算」に書き換える。
5. 書き換えた見出し１と、一組の<script type="text/javascript"></script>だけ残して、本文中の他のコードはすべて削除する。

**Do!** 5%の消費税を含んだ価格が1260円の場合の計算をするために、最初に以下のコードを書いておこう。

var price5 = 1260; // 5%消費税込価格

document.write('<p>5%消費税込の価格は' + price5 + '円です。</p>');

### 税抜価格を求めて、表示するようにしなさい。

Hint: 空欄に税抜価格を計算する式を入れて、以下のコードを追加せよ。

var price\_ex\_tax = 　　空　欄　　; // 消費税抜き価格

document.write('<p>税抜価格は' + price\_ex\_tax + '円です。</p>');

### 5%消費税を求めて、表示するようにしなさい。

Hint: 空欄に5%消費税を計算する式を入れて、以下のコードを追加せよ。

var tax5 = 　　空　欄　　; // 5%消費税

document.write('<p>5%消費税は' + tax5 + '円です。</p>');

### 8%消費税込価格を求めて、表示するようにしなさい。

Hint: 空欄に8%消費税込価格を計算する式を入れて、以下のコードを追加せよ。

var price8 = 　　空　欄　　; // 8%消費税込価格

document.write('<p>8%消費税込の価格は' + price8 + '円です。</p>');

## 9. 課題06

練習問題で完成したtax8.htmlを、1260円以外の8%消費税価格が簡単に計算できるように、prompt関数を使用して、5%消費税込価格を入力できるように改造しなさい。更に、ページの表示デザインに関して、独自の工夫もしてみること。

（Hint: prompt関数の返り値を、price5に代入するようにすれば良い。）

## 10. Check List

* JavaScriptはWWWサーバではなく、Webブラウザで動作することを理解したか？
* HTMLファイルでJavaScriptを実行する場合に、HTMLファイル内にJavaScriptのソースコードを埋め込むにはどうすればよいか？
* HTMLファイルでJavaScriptを実行する場合に、HTMLファイルに外部のJavaScriptファイルを読み込むにはどうすればよいか？
* JavaScriptで//はどのような働きをする記号か？
* JavaScriptのセミコロン（;）の意味は？
* 変数とは何か？
* 文字列を変数名として使用することを宣言する方法は？
* 文字列のデータ（文字リテラル）の表現方法は？
* 文字列のデータを結合する演算子は何か？
* FirefoxのWebコンソールの使い方がわかったか？
* 数値演算子にはどのようなものがあるか？
* document.write()は何をする関数か（その働きは何か）？
* alert()は何をする関数か（その働きは何か）？
* prompt()は何をする関数か（その働きは何か）？
* confirm()は何をする関数か（その働きは何か）？

1. +演算子は、数値のデータに作用させると、足し算になる。このように、JavaScriptでは、数値演算子と文字列演算子が同じ記号で表されるものがある。 [↑](#footnote-ref-1)
2. 正確にはwindowオブジェクトのメソッドである。 [↑](#footnote-ref-2)