2002年度数理情報実験資料
JavaScriptの学習
(JavaScriptによるプログラミング)

担当：河野敏行

平成14年10月9日

目次

1	DHTML	1
2	DHTMLの基礎	2
2.1	スタイルシート	2
2.2	スクリプト	2
3	DHTMLの書式	4
3.1	IEの特徴	4
3.2	Netscapeの特徴	5
3.3	DHTMLの記述方法	5
4	DHTMLの作成手順	6
5	JavaScript	8
5.1	文字の表示 document.write()	10
5.2	変数 var	10
5.3	配列と繰り返し処理 for(<i>i</i> = 0; <i>i</i> < 5; <i>i</i> ++){ }	10
5.4	条件処理 if(条件文){ }else{ }	12
5.5	ループ while 文	12
5.6	関数 function	14
5.7	< form > タグの利用	16
5.8	イベント処理	18
5.9	オブジェクトの位置と配列	20
6	オブジェクトについて	22
7	JavaScript リファレンス	28
7.1	演算コマンド (Math)	28
7.2	変換/演算	28
7.3	ウインドウ	29
8	その他	30
8.1	時間と配列を使った例 (sample1.htm)	30
8.2	日付, 曜日の表示例 (sample2.htm)	31
8.3	マーカーを利用したサンプル (sample3.htm)	31
8.4	フィルタの例 (sample4.htm)	34
8.5	クイズの例 (sample5.htm)	36

1 DHTML

DHTML,つまりダイナミック HTMLは「HTMLをダイナミック」にしようという技術のことである。静的だったホームページを Dynamic (動的)にするものである。たとえば,アクセスしてその後,文章の中の1部が変更するようなホームページを作成する場合,その一部のためにページ全体を丸ごと読み直したり,Java やプラグインコンテンツという大掛かりな技術を用いることは大変面倒である。そこで,この DHTML を使用するとページを読み直すことなく可能になる。それは従来の HTML に1行のみの変更で出来る。このように,従来の HTML に簡単な手間で,動的な表現手段を実現できるものが DHTML の特徴である。

具体的には,スタイルシートと JavaScript などによって実現する。

2 DHTMLの基礎

先に示したように、DHTMLではスタイルシートとJavaScriptを用いる。まず、その説明をする。

スタイルシート	HTMLの表現を記述
JavaScript	HTMLの動作を記述

動作を記述するためにMicrosoftのIEではVBScriptも使用可能であるが、本実験では、JavaScriptを用いる。

2.1 スタイルシート

「スタイルが良いね」というと「格好が良いね」ということを表すように、ここではHTMLの格好を設定するためのシートを意味している。

2.2 スクリプト

「HTMLタグ」や「スタイルシートの要素」をコントロールするためのものである。またHTMLタグの中での「イベントハンドラ」によって情報を取得したりスクリプトを起動したり出来る。記述方法は次の2通りある。

- 1 <script> タグの中
- 2 HTML タグの中

作業の仕方

1. 画面上のアイコンからテキストエディタを立ち上げる。
2. 例にある通りに入力する。
3. テキストエディタの[ファイル]-[名前を付けて保存]を選択する。
4. 保存する場所はPドライブの適当なフォルダの中にファイル名を付け保存する。
5. Zドライブの保存したファイルをIEで確認する。

例 2.1 スタイルシートの書き方

rei31.htm

```
<html>
<head>
<title> スタイルシート </title>
<style type="text/css">
<!--
    h1{color:blue}
    p{font-size:16pt; font-style:italic}
//-->
</style>
</head>
<body>
<h1>スタイルシートで指定した色？</h1>
<p>スタイルシートでは，テキストの書体や大きさ，色，位置などの
細かい設定ができる．
</p>
</body>
</html>
```

例 2.2 インラインスタイルシート

rei32.htm

```
<html>
<head>
<title> スタイルシート </title>
</head>
<body>
<h1 style="color:blue">スタイルシートで指定した色？</h1>
<p style="font-size:16pt; font-style:italic">スタイルシートでは，
テキストの書体や大きさ，色，位置などの細かい設定ができる．
</p>
</body>
</html>
```

3 DHTMLの書式

皆さんが用いるブラウザは大きく分けて、2種類ある。

Microsoft Internet Explorer

Netscape Communicator

この2つのブラウザにおいて、DHTMLは使えるが、その仕様が違う。本実験ではIEを
そして可能であればその両方の書式について説明を行う。まず、各ブラウザの特徴を示す。

3.1 IEの特徴

基本的にスタイルシートとしては Cascading Style Sheet、スクリプト言語として VB-
Script や JavaScript を用いる。以下の4つの特徴がある。

- 1 Dynamic Style and Positioning
- 2 Dynamic Contents
- 3 Data Binding
- 4 Multimedia Control

Dynamic Style and Positioning

従来のHTMLでは、1度ページを読み込んでしまった後に、ページの概観を変更することは困難で、どうしても変更をしたい場合は、リロードするかスクリプトなどによる処理が必要であったが、サーバーに接続しているかどうかに関わらず、スタイルシートの要素をスクリプト上で設定したりスタイルシートで設定されている要素をスクリプトで変更することでテキストの色や装飾、配置などの表現をダイナミックに変えることが出来る。たとえば、マウスがあるテキストの上に来たときに、そのテキストのフォントの種類やサイズ、色を変える事が出来る。

CSSを用いて「X座標」「Y座標」「Z座標」という形で、位置決めをすることが出来るようになり、具体的な数値での3次元的な位置指定が可能となった！「Z平面」の指定は要素の重ねあわせを可能にする。

Dynamic Contents

IEではほとんどのタグを VBScript または、JavaScript によって呼び出し、任意の命令を与えられるようになっている。これを「オブジェクトモデル」と呼ぶ。この「オブジェクトモデル」により、ほとんどのタグでコンピュータ上のブラウザに対してどのような操作を行ったかというイベントの検知が出来るようになり、スクリプトを使ってコンテンツ側でユーザーが行うマウス操作やキーボード操作にきめこまやかに対応できる。

Data Binding

Web ページを切り替えることなく、表示しているオブジェクトに対してデータを更新したり、並び替えや条件によるデータのピックアップが出来る。

Multimedia Control 表示上の特殊な効果を与えるものでスタイルシートだけでは、表現しきれない視覚的な機能を付け加える「パス」、「Visual filter」、「ドラジション」など。

3.2 Netscape の特徴

Netscape においても同様の特徴がある。

- 1 Cascading Style Sheet (CSS)
- 2 Document Object Model (DOM)
- 2 Dynamic Layer
- 3 Dynamic Font

ここでは、簡単に説明をする「ダイナミックスタイルシート」はスタイルシートでレイアウトとテキストの装飾を行いブラウザに表現される部分を限定する「レイヤー」はセル画のように何枚も重ねられるような透明な用紙のようなものである。他のアプリケーションにおいてもレイヤーという概念が用いられ、大変便利である。「ダウンロードダブルフォント」(ダイナミックフォント)はフォントの情報を「PFR」(Portable Font Resource)という形式のファイルに格納しておきページが表示される際に HTML ファイルと同時に PFR をダウンロードし、コンピュータに無いフォントを表示できるものである。

3.3 DHTML の記述方法

DHTML を記述する方法としては以下の 2 通りがある。

- ・ スクリプトとして記述
- ・ タグのオプションとして記述

具体例を示すが、先にも述べたように IE と NN とで使用が違いため、それぞれの例を示す。まず、簡単なタグのオプションで記述する方を示す。

例 3.1 タグで記述した場合

タグ for NN

```
<layer bgcolor="red">これは赤色</layer>
```

タグ for IE

```
<div style="position:absolute;background-color:red">これは赤色</div>
```

4 DHTMLの作成手順

次にスクリプトに関数を定義し、タグでイベントの発生条件を付加した簡単な例題を通して、実際の作成手順を学習する。作成手順はほとんどが以下のように行う。

DHTMLの作成手順

1. ダイナミック HTML としてコントロールしたいオブジェクトをタグで囲む。
2. ID やスタイル設定などのオプションを決める。
3. イベントの発生条件を決める。
4. コントロールするためのスクリプト (プログラム) を書く。

簡単な例題を示す。

例 4.1 次のような動作をするものを作成せよ。

マウスが文字に重なったとき、バックの色を変える。

動きに関係なく、バックに色がついていた例は例 3.1 に示した通りである。

演習 4.1 次のような動作をするものを作成せよ (*ex51.htm*)

マウスが文字に重なったときに、文字の色を変え、クリックすると大きくなる。

例 5.1 解 マウスで背景色を変える for NN(rei51n.htm)

```
<html>
<head>
<title>マウスでテキスト背景色を変える for NN</title>
<script Language="JavaScript">
function backiro(CC){
    document.layers["test2"].bgColor = CC;
}
</script>
</head>
<body>
<LAYER ID="test2" LEFT=10 top=10 bgcolor="black"
onMouseOver="backiro('red')" onMouseOut="backiro('yellow')">
変わった?
</LAYER>
</body>
</html>
```

例 5.1 解 マウスで背景色を変える for IE(rei51i.htm)

```
<html>
<head>
<title>マウスでテキスト背景色を変える for IE</title>
<script Language="JavaScript">
<!--
function backiro(CC){document.all["test2"].style.backgroundColor=CC}
//-->
</script>

</head><body>
<div id="test2" STYLE="position:absolute;
background-color:black;top:10;left:10"
onMouseOver="backiro('red')" onMouseOut="backiro('yellow')">
変わった?
</div>
</body>
</html>
```

5 JavaScript

先の例題でも ,Language="JavaScript" という書式を見たように ,使う言語は JavaScript と宣言して ,プログラムを行っています .まず ,JavaScript で何が出来るか考えてみましょう .理解のために HTML で出来ることを示します .

HTML で出来ること

1. 文字の設定
2. レイアウトとイメージ表示の設定
3. リンクの設定

さらに ,入力フォームやボタンを貼り付けることは出来るが ,入力された内容やボタンを押す動作によって応答を変えるには ,CGI や Java という HTML 以外の方法をとらなければならない .またそれらを用いる場合 ,難しい ,すぐ試せない ,サーバー側の設定が必要など面倒なことがあるが ,JavaScript では次のような特徴がある .

JavaScript の特徴

1. プログラムが簡単
2. 実行が簡単
3. サーバーなど別のファイルでの設定は不要

JavaScript の記述の基本的な形は次のようなものです .

JavaScript の書式

```
<script Language="JavaScript">
<!--
    JavaScript プログラム , 命令
-->
</script>
```

そしてその命令は次のような形式でかかれます .

JavaScript の命令

「どこに」!「どうする(~)」

以下は各ユーザーのトップページのここまでに最低必要な内容である．各自工夫をすること．

index.html

```
<html>
<head><title>学番</title>
</head>
<body>
<h1>実験課題の作成 (学番)</h1>
<a href="./rei31.htm">スタイルシートの書き方</a><br>
<a href="./rei32.htm">インラインスタイルシートの書き方</a><br>
<a href="./rei51n.htm">マウスで背景色を変える ( forNN )</a><br>
<a href="./rei51i.htm">マウスで背景色を変える ( forIE )</a><br>
<hr>
<a href="./ex51.htm">演習：マウスで変化 </a><br>
</body>
</html>
```

5.1 文字の表示 document.write()

書式

```
document.write("表示する文章");
```

付け加えて、画像の場合は、命令を次のように変えれば良い。

```
document.write("<img src=test.gif>");
```

5.2 変数 var

変数の書式

```
var 変数名;  
var 変数名=値;  
var 変数名="文字列";
```

変数の命名規則

- 変数名の最初はアルファベット
- 255文字を超えてはいけない
- ピリオドを変数名に使ってはいけない

5.3 配列と繰り返し処理 `for(i = 0; i < 5; i++) {}`

先ほどの変数を配列として定義し、繰り返し処理によって全て足すことを考えましょう。

演習 5.1 「私は…である」という文章の…の部分いくつか用意し、それらを 5.1, 5.2, 5.3を参考に表示する `html` を作成せよ。(ex61.htm)

文字の表示 (test-J1.htm)

```
<html>
<head>
<title>test_J1</title>
</head>
<body>
<script Language="JavaScript">
<!--
document.write("私は誰だ！");
//-->
</script>
</body>
</html>
```

変数の例 (test-J2.htm)

```
var hako = 5;
var tana;
tana = 10;
document.write("棚の中に箱が"+tana+"個あり，各箱の中に人形が"+hako+"体あります．<br>");
document.write("全部で人形は"+tana*hako+"体あります．");
```

繰り返し処理の例 (test-J3.htm)

```
var hako = new Array(5);
hako[0]=2; hako[1]=10; hako[2]=3; hako[3]=10; hako[4]=3;
var tana =0 ;
for(i=0;i<5;i++){
    tana = tana + hako[i]
}
document.write("全部で人形は"+tana+"体あります．");
```

5.4 条件処理 `if(条件文){ }else{ }`

条件式は次のように C でならったとおりである .

```
if(a > b)    aが bより大ならば
if(a < b)    aが bより小ならば
if(a >= b)   aが b以上ならば
if(a <= b)   aが b以下ならば
if(a == b)   aと bが等しいならば
if(a != b)   aと bがちがうならば
```

詳しくは以下の表のようになる .

表 6.1 演算子について

算術演算子		比較演算子		論理演算子	
+	加算	==	等しい	!	反転 (not)
-	減算	!=	等しくない	&	論理積 (and)
*	乗算	<	より小		論理和 (or)
/	除算	>	より大	^	排他的論理和 (xor)
%	余剰	<=	以下		
++	インクリメント	>=	以上		
--	デクリメント	&&	どちらも満足		
	補数		一方を満足		

5.5 ループ while 文

while の書式

```
while(条件式){
    条件を満たしている間の処理
}
```

演習 5.2 あなたの現在の所持金は 0 円ですが , 1 日目に 1 円 , 2 日目には 2 円 , 3 日目 4 円と昨日の倍のお金をもらっていき , 何日で , 所持金が 1 万円を超えるかを 1 日 1 日何円もらったか表示し , 示すような *html* を作れ . (*ex62.htm*)

条件処理 (test-J4.htm)

```
var doll = 30;
document.write("全部で人形は"+doll+"体あります。");
if(doll < 10){
    document.write("普通くらいかな。");
}
else{
    document.write("ちょっとマニアなんじゃない!?");
}
```

while ループ (test-J5.htm)

```
var doll = 3;
var i=0;
document.write("現在あなたは人形を"+doll+"体持っています。<br>");
if(doll >= 10){
    document.write("あなたはマニアです!");
}
else{
    document.write("1日1体ずつ増えていくと後");
    while(doll < 10){
        i++;
        doll=doll+1;
    }
    document.write(i+"日後には,あなたはマニアです!");
}
```

5.6 関数 function

よく使う処理などを1つの「関数」として定義しておく．その基本的な書式は次の通りである．

関数の書式

```
function 関数名(引数){  
    実行するプログラムなど  
  
    return 戻り値  
}
```

```
<html>
<head>
<title>test_J6</title>
<script Language="JavaScript">
<!--
function kake(a,b){
    return a*b;
}
//-->
</script>
</head>
<body>
<script Language="JavaScript">
<!--
    var kosu = 5;
    var niti = 12;
    document.write("人形を 1 日に"+kosu+"体買うと",niti,"日間で",kake(kosu,niti),"体集まります");
//-->
</script>
</body>
</html>
```

5.7 < form > タグの利用

ブラウザ上から入力されたデータを JavaScript で処理できるようにする．まず form の種類を例で示す．

資料：HTML のタグ

A	ADDRESS	APPLET	AREA	B
BASE	BASEFONT	BGSOUND	BIG	BLINK
BLOCKQUOTE	BODY	BR	BUTTON	CAPTION
CENTER	CITE	CODE	COL	COLGROUP
<-- -->	DD	DFN	DIR	DIV
DL	DOCTYPE	DT	EM	EMBED
FONT	FORM	FRAME	FRAMESET	H1 - 6
HEAD	HR	HTML	I	IFRAME
ILAYER	IMG	INPUT	ISINDEX	KBD
LABEL	LAYER	LH	LI	LINK
MAP	MARQUEE	META	MULTICOL	NOBR
NOEMBED	NOFRAME	NOLAYERS	NOSCRIPT	OBJECT
OL	OPTION	P	PARAM	PRE
S	SAMP	SCRIPT	SELECT	SMALL
SPACER	SPAN	STRIKE	STRONG	STYLE
SUB	SUP	TABLE	TBODY	TD
TEXTAREA	TFOOT	TH	THEAD	TITLE
TR	TT	U	UL	VAR
WBR	XMP			

演習 5.3 選択問題を作成し，答えが選択されたとき，正解か不正解かを示すような *html* を作成せよ．(*ex63.htm*)

```
<html>
<head>
<title>
入力 form の種類
</title>
</head>
<body>
<form>
text:<input type="text"><br>
password:<input type="password"><br>
button:<input type="button"><br>
submit:<input type="submit"><br>
reset:<input type="reset"><br>
hidden:<input type="hidden"><br>
checkbox:<input type="checkbox"><br>
radio:<input type="radio"><br>
select:<select name="soujou">
<option value="suuri">数理情報
<option value="buturi">シミュレーション物理
<option value="seiti">生物地球システム
<option value="shajou">社会情報
</select><br>
textarea:<textarea rows=8 cols=40>
</textarea>
</form>
</body>
</html>
```

5.8 イベント処理

ボタンをクリック、リンクテキストに対してカーソルを合わせる、フォームの入力文字を変えるなどの「イベント」に対してプログラムを起動させる方法を示す。

イベント処理

<タグ名 イベントハンドラ = 実行プログラム>

イベントハンドラの例は以下のとおりです。

onChange フォームの内容を変化させたとき

onClick ボタンやアンカーをクリックしたとき

onSubmit フォームの内容を送信 (Submit) したとき

詳しくは最後に付ける。

イベント処理 (test-J8.htm)

```
<html>
<head>
<title>イベント処理</title>
</head>
<body>
<form>
<input type="button" value="?" onClick="alert(' なんて押したの !? ')">
</form>
</body>
</html>
```

演習 5.4 *test-J2, J3, J4, J5, J6* の最初の変数を入力できるように改良しなさい (ヒント: *form* タグと *prompt* などを利用)

```

<html>
<head>
<title>イベント処理</title>
<script Language="JavaScript">
function httpc(nakami){
  if(-1 == nakami.indexOf("http://")){
    alert('変ですよ')}} }
</script>
</head>
<body>
<form>
<input type="text" value="http://" onChange="httpc(this.value)">
</form>
</body>
</html>

```

イベント処理 (test-J10)

```

<html>
<head>
<title>
イベント処理
</title>
<script Language="JavaScript">
function indata(koko){
  if(koko.dolls.value==""){
    document.write("まじめにやって"); }
  else{
    kosu=koko.dolls.value;
    alert("あなたはゲーム機を "+kosu+"台持っているんだ");
  }
}
</script>
</head>
<body>
<form name="ireru">
あなたはゲーム機を何台持っていますか<br>
<input type="text" name="dolls">
<input type="button" name="okuru" value="入力"
onClick="indata(this.form)">
</form>
</body>
</html>

```

5.9 オブジェクトの位置と配列

ここまでの例題のように、特定のオブジェクトから値を取り出す、書き込むなどを行う場合、その位置を指定しなければならない。指定の仕方は、日本の住所のように日本 岡山県 岡山市 理大町 1丁目 1番 岡山理科大学 1号館というふうに大きいエリアから小さいエリアに的を絞る方法です。

forms 配列で form の位置を指定

document.forms[n]

```
<form name="a">
</form>
<form name="b">
</form>
```

```
document.forms[0]
document.forms[1]
```

form の名前で位置を指定

document. フォームの名前

```
<form name="a">
</form>
<form name="b">
</form>
```

```
document.a
document.b
```

this.form で位置を指定する例

onClick="alert(this.form)"

elements 配列で位置を指定する例

document. フォームの名前.elements[n]

element の名前で位置を指定する例

document. フォームの名前. エレメントの名前

this.form で位置を指定する例

this.form. エレメントの名前

セレクト, オプションの位置を指定する例

document.forms[n].elements[n].options[n]

リンク, アンカーの位置を指定

A タグの中で「HREF=" "」に対して links[n] 配列
A タグの中で「NAME=" "」に対して anchor[n] 配列

6 オブジェクトについて

Document Object Model を使いこなすことが出来れば，DHTML を使いこなすのも楽勝です．DHTML ではホームページ上にある全てのオブジェクトをすきりぷなどをを用いて変化させられることが出来る．そのとき，それぞれのオブジェクトには，メソッドやプロパティがあり，利用することによってオブジェクトを変化させられる．このように DHTML は従来の HTML と異なり，ここで変化させたプロパティの値がすぐに画面上に反映されるため，動的な効果が生まれる．

オブジェクトの種類はページで表示できる全てのものを指定でき，大量にある．オブジェクトに応じて，プロパティ，メソッド，イベント，コレクションがある．

表 7.1 オブジェクト

A	ACRONYM	ADDRESS	all	APPLET
applets	anchors	AREA	areas	B
BASE	BASEFONT	BGSOUND	BIG	BLOCKQUOTE
BODY	BR	BUTTON	CAPTION	cells
CENTER	children	CITE	CODE	coL
COLGROUP	COMMENT	DD	DEL	DFN
DIR	DIV	DL	document	DT
elements	EM	EMBED	embeds	event
external	FIELDSET	filters	FONT	FORM
forms	FRAME	frames	FRAMESET	H1
H2	H3	H4	H5	H6
HEAD	history	HR	HTML	I
IFRAME	images	IMG	INPUT	INS
KBD	LABEL	LEGEND	u	LINK
links	LISTING	location	MAP	MARQUEE
MENU	META	navigator	NEXTID	OBJECT
oL	OPTION	options	P	PLAINTEXT
plugins	PRE	Q	rows	rules
S	SAMP	screen	SCRIPT	scripts
SELECT	selection	SMALL	SPAN	STRIKE
sTRONG	STYLE	style	styleSheet	styleSheets
SUB	SUP	TABLE	TBODY	TD
TEXTAREA	TextRange	TFOOT	TH	THEAD
TITLE	TR	TT	U	UL
userprofile	VAR	window	XMP	

表 7.2 input type の種類

button	image	submit	checkbox	password
text	file	radio	hidden	reset

表 7.3 プロパティの種類

accessKey	action	activeElement
align	aLink	alinkColor
alt	altHTML	altKey
appCodeName	appMinorVersion	appName
appVersion	availHeight	availWidth
background	backgroundAttachment	backgroundColor
backgroundImage	backgroundPosition	backgroundPositionX
backgroundPositionY	backgroundRepeat	balance
behavior	bgColor	bgProperties
border	borderBottom	borderBottomColor
borderBottomStyle	borderBottomWidth	borderColor
borderColorDark	borderColorLight	borderLeft
borderLeftColor	borderLeftStyle	borderLeftWidth
borderRight	borderRightColor	borderRightStyle
borderRightWidth	borderStyle	borderTop
borderTopColor	borderTopStyle	borderTopWidth
borderWidth	bottomMargin	boundingHeight
boundingLeft	boundingTop	boundingWidth
browserLanguage	bufferDepth	button
cancelBubble	caption	cellIndex
cellPadding	CellSpacing	Charset
checked	classid	className
clear	clientHeight	clientLeft
clientInformation	clientTop	clientWidth
clientX	ClientY	clip
closed	code	codeBase
codeType	color	colorDepth
cols	ColSpan	compact
complete	content	cookie
cookieEnabled	coords	cpuClass
cssText	ctrlKey	cursor
data	dataFld	dataFormatAs
dataPageSize	dataSrc	defaultCharset
defaultChecked	defaultSelected	defaultStatus
defaultValue	defer	dialogArguments
dialogHeight	dialogLeft	dialogTop
dialogWidth	direction	disabled
display	domain	dynsrc

encoding	event	expand
face	fgColor	fileCreatedDate
fileModifiedDate	fileSize	fileUpdatedDate
filter	font	fontFamily
fontSize	fontSmoothingEnabled	fontStyle
fontVariant	fontWeight	form
frame	frameBorder	frameSpacing
fromElement	hash	height
hidden	host	hostname
href	hspace	htmlFor
htmlText	httpEquiv	I
id	lineHeight	indeterminate
index	innerHTML	innerText
isMap	isTextEdit	keyCode
lane	language	lastModified
left	leftMargin	length
letterSpacing	link	linkColor
listStyle	listStyleImage	listStylePosition
listStyleType	location	loop
lowsrc	map	marginTop
margin	marginBottom	marginHeight
marginLeft	marginRight	marginWidth
maxLength	media	method
Methods	mimeTypes	multiple
name	noHref	noResize
noShade	noWrap	object
offsetLeft	offsetLeft	offsetHeight
offsetParent	offsetTop	offsetWidth
offsetX	offsetY	online
opener	outerHTML	outerText
overflow	owningElement	paddingBottom
padding	paddingLeft	paddingRight
paddingTop	pageBreakAfter	pageBreakBefore
palette	parent	parentElement
parentStyleSheet	parentTextEdit	parentWindow
pathname	pixelHeight	pixelLeft
pixelTop	pixelWidth	platform
plugins	pluginspage	port
posHeight	position	posLeft

posTop	posWidth	protocol
readOnly	readyState	reason
recordNumber	recordset	referrer
rep	returnValue	rev
rightMargin	rowIndex	rows
rowSpan	rules	screenX
screenY	scroll	scrollAmount
scrollDelay	scrollHeight	scrolling
scrollLeft	scrollTop	scrollWidth
search	sectionRowIndex	selected
selectedIndex	self	shape
shiftKey	size	sourceIndex
span	src	srcElement
srcFilter	status	style
styleFloat	systemLanguage	tabIndex
tagName	target	text
textAlign	textDecoration	textDecorationBlink
textDecorationLineThrough	textDecorationNone	textDecorationOverline
textDecorationUnderline	textIndent	textTransform
tfoot	thead	toElement
top	topMargin	trueSpeed
type	type	units
updateInterval	url	urn
useMap	userAgent	userLanguage
vAlign	value	verticalAlign
visibility	vLink	vlinkColor
volume	vspace	vwidth
wrap	x	y
zIndex		

表 7.4 メソッドの種類

add	addChannel	addImport
addReadRequest	addRule	alert
assign	back	blur
clear	clearInterval	clearRequest
clearTimeout	click	close
collapse	compareEndpoints	confirm
contains	createCaption	createElement
createRange	createStyleSheet	createTextRange
createTHead	createTFoot	deleteCaption
deleteCell	deleteRow	deleteFoot

deleteThead	doReadRequest	duplicate
elementFromPoint	empty	execCommand
execScript	expand	findText
focus	forward	getAttribute
getBookmark	go	inRange
insertAdjacentHTML	insertAdjacentText	insertCell
insertRow	isEqual	isSubscribed
item	javaEnabled	move
moveBy	moveEnd	moveStart
moveTo	moveToBookmark	moveToElementText
moveToPoint	navigate	nextPage
open	ParentElement	pasteHTML
previousPage	prompt	queryCommandEnabled
queryCommandIndeterm	queryCommandState	queryCommandSupported
queryCommandValue	refresh	reload
remove	removeAttribute	replace
reset	resizeBy	resizeTo
scroll	scrollBy	scrollTo
scrollIntoView	select	setAttribute
setEndPoint	setInterval	setTimeout
showHelp	showModalDialog	start
Stop	Submit	tags
taintEnabled	write	writeln
zOrder		

表 7.5 イベントの種類

マウス	キーボード	ロード
onmouseover	onkeypress	onload
onmouseout	onkeydown	onunload
onmousedown	onkeyup	
onmouseup		
onmousemove		
onclick		
ondblclick		

表 7.6 その他のイベントの種類

onabort	onafterupdate	onbeforeunload	onbeforeupdate
onblur	onbounce	onchange	ondataavailable
ondatasetchanged	ondatasetcomplete	ondragstart	onerror
onerrorupdate	onfilterchange	onfinish	onfocus
onhelp	onreset	onresize	onrovventer
onrowexit	onscroll	onselect	onselectstart
onstart	onsubmit		

7 JavaScript リファレンス

7.1 演算コマンド (Math)

abs	…	絶対値
acos	…	アークコサイン
asin	…	アークサイン
atan	…	アークタンジェント
atan2	…	X,Y 座標から角度を求める
ceil	…	もっとも大きくて近い整数を返します
cos	…	コサイン
E	…	自然対数の底 e
exp	…	E を基数とする指数を求める
floor	…	最も小さい整数値にする
LN10	…	10 の自然対数の値
LN2	…	2 の自然対数の値
log	…	常用対数
LOG10E	…	E の常用対数
LOG2E	…	2 を底とする E の常用対数
max	…	2 つの数値を比較し最大をとる
min	…	2 つの数値を比較し最小をとる
PI	…	円周率
pow	…	べき乗
random	…	乱数
round	…	値を四捨五入
sin	…	サイン
sqrt	…	平方根
SQRT1_2	…	2 の平方根の半分
SQRT2	…	2 の平方根
tan	…	タンジェント

7.2 変換 / 演算

escape	…	文字列のアスキーコード (ISO Latin 1) を返します
eval	…	文字列を数式として演算します
isNaN	…	数値かどうか調べます (false / true)
parseFloat	…	小数を含む文字列を数値にします
parseInt	…	文字列を整数値に変換します
toString	…	n 進数に変換します
unescape	…	ISO Latin 1 文字列を変換します
void	…	何も返さない値を返します

7.3 ウィンドウ

メッセージウィンドウ

```
alert("メッセージ");
```

ウィンドウを開いて「メッセージ」を表示する。

確認ウィンドウ

```
if(confirm("確認メッセージ")){  
    OKのときの処理  
}else{  
    キャンセルのときの処理  
}
```

ウィンドウを開いて確認メッセージを表示し、OKのときとキャンセルのときとで別の処理をする。

入力ウィンドウ

```
変数 = prompt("メッセージ", "初期値");
```

ウィンドウを開いてメッセージを表示し、入力ボックスに初期値を表示し、OKを押すと入力ボックスの値が変数に代入される。

ブラウザの新しいウィンドウを開く

```
open("URL", "ウィンドウ名", "width=w,height=h,menuber=yes,...");
```

高さhドット、幅wドットのウィンドウを開き、URLを表示する。

8 その他

8.1 時間と配列を使った例 (sample1.htm)

```
<html>
<head>
<title>時間によるメッセージ</title>
<script Language="JavaScript">
<!--
key= new Array(24);
key[0]="午前 0 時です . 真夜中です . ";
key[1]="午前 1 時です . ";
key[2]="午前 2 時です . 丑三つ時です";
key[3]="午前 3 時です . ";
key[4]="午前 4 時です . もうすぐ夜明け";
key[5]="午前 5 時です . 新聞はもう来ているかな";
key[6]="午前 6 時です . ";
key[7]="午前 7 時です . ";
key[8]="午前 8 時です . ";
key[9]="午前 9 時です . お仕事開始";
key[10]="午前 10 時です . ";
key[11]="午前 11 時です . もうそろそろランチに";
key[12]="午後 0 時です . お昼休みを満喫";
key[13]="午後 1 時です . ";
key[14]="午後 2 時です . ";
key[15]="午後 3 時です . おやつを食べると太るぞ";
key[16]="午後 4 時です . ";
key[17]="午後 5 時です . ";
key[18]="午後 6 時です . ";
key[19]="午後 7 時です . もうそろそろ仕事を切り上げて";
key[20]="午後 8 時です . ";
key[21]="午後 9 時です . まだ , 仕事か";
key[22]="午後 10 時です . もう帰って TV でも見て";
key[23]="午後 11 時です . もう寝るべ";

function printkey(){
    d=new Date();
    h=d.getHours();
    document.write(key[h]);
}
//-->
</script>
</head>
```

```
<body>
<hr>
<script language="JavaScript">
<!--
printkey();
//-->
</script>
<hr>
</body>
</html>
```

8.2 日付, 曜日の表示例 (sample2.htm)

```
<html>
<head>
<title>アクセス日付を表示</title>
<script Language="JavaScript">
<!--
function printDate(){
    dt=new Date();
    youbi ="日月火水木金土".substring(dt.getDay(),dt.getDay()+1);
    document.write("平成",dt.getYear()-1988,"年",dt.getMonth() +1,
        "月",dt.getDate(),"日(",youbi,")");
}
//-->
</script>
</head>
<body>
<script langage="JavaScript">
<!--
    printDate();
//-->
</script>
</body>
</html>
```

8.3 マーキーを利用したサンプル (sample3.htm)

```
<HTML>
<HEAD>
```

```
<TITLE>MARQUEE</TITLE>
<STYLE TYPE="text/css">
BODY {
background-color: #000000;
color: #FFFFFF;
font-size: 12pt;
font-family: "MS Pゴシック";
margin-top: 10px;
margin-left: 5px;
margin-right: 5px;
}
.marquee {
background-color: #FFFFFF;
color: #000000;
font-size: 16px;
width: 200px;
height: 60px;
}
.on {
background-color: #94ED94;
color: #8A2BE2;
font-size: 12pt;
font-family: "MS Pゴシック";
font-weight: bold;
}
.off {
background-color: #000000;
color: #FFFFFF;
font-size: 12pt;
font-family: "MS Pゴシック";
font-weight: normal;
}
</STYLE>
</HEAD>

<SCRIPT>
function ugoki(n)
{
if (n==0)
{
sample.direction = "left";
string1.className = "on";
string2.className = "off";
```

```

string3.className = "off";
string4.className = "off";

}
if (n==1){
sample.direction = "right";
string1.className = "off";
string2.className = "on";
string3.className = "off";
string4.className = "off";
}
if (n==2)
{
sample.direction = "up";
string1.className = "off";
string2.className = "off";
string3.className = "on";
string4.className = "off";
}
if(n==3){
sample.direction = "down";
string1.className = "off";
string2.className = "off";
string3.className = "off";
string4.className = "on";
}
}
</SCRIPT>

<BODY>

<TABLE ALIGN="center" BORDER="0" WIDTH="95%" CELLPADDING="10"
        CELLSPACING="0">
<TR>
<TD><SPAN ID="string1" CLASS="on" ONMOUSEOVER="ugoki(0)">左向き
        </SPAN></TD>
<TD><SPAN id="string2" CLASS="off" ONMOUSEOVER="ugoki(1)">右向き
        </SPAN></TD>
<TD><SPAN ID="string3" CLASS="off" ONMOUSEOVER="ugoki(2)">上向き
        </SPAN></TD>
<TD><SPAN ID="string4" CLASS="off" ONMOUSEOVER="ugoki(3)">下向き
        </SPAN></TD>
</TR>

```

```

</TABLE>

<DIV STYLE="margin-left:50">
<TABLE BORDER="8" CELLSPACING="8" BORDERCOLORLIGHT="#COCOCO"
      BORDERCOLORDARK="#696969">
<TR><TD>
<MARQUEE ID="sample" CLASS="marquee" DIRECTION="left" SCROLLDELAY="1"
      SCROLLAMOUNT="1" BORDER="0">掲示板 </MARQUEE>
</TD></TR>
</TABLE>
</DIV>
</body>
</html>

```

8.4 フィルタの例 (sample4.htm)

```

<html>
<head>
<title></title>
<script language="JavaScript">
<!--
function alphaTest1(){
  Test.style.filter="alpha(opacity=100, finishOpacity=0, style=1,
  startX=0, startY=0, finishX=70, finishY=80)";
}
function alphaTest2(){
  Test.style.filter="alpha(opacity=100, finishOpacity=0, style=2,
  startX=0, startY=0, finishX=70, finishY=80)";
}
function blurTest(){
  Test.style.filter="blur(add=1,direction=150,strength=30)";
}
function kageTest(){
  Test.style.filter="dropShadow(color=#aaaaaa,offX=5,offY=5,positive=1)";
}
function fliphTest(){
  Test.style.filter="flipH()";
}
function flipvTest(){
  Test.style.filter="flipV()";
}
}

```

```

function glowTest(){
    Test.style.filter="glow(color=#9999FF, strength=10)";
}
function kaeru(iro){
    Test.style.color=iro;
}
//-->
</script>
</head>
<body>
<DIV ID="Test" style="position:absolute;width:100%;font-size:12pt;
font-family:Times">
<center>
<H1>2000 年度 数理情報実験 </h1><br>
<h2>DynamicHTML によるマルチメディアコンテンツの作成</h2><BR>
<h3>担当：河野敏行</h3>
</center>
</DIV><P>
<div id="sousal" style="position:absolute;top:200;left:150">
<p onmouseover="alphaTest1()">alpha Test1</p>
<p onmouseover="alphaTest2()">alpha Test2</p>
<p onmouseover="blurTest()">blur Test</p>
<p onmouseover="kageTest()">drop shadow Test</p>
<p onmouseover="fliphTest()">左右反転 Test</p>
<p onmouseover="flipvTest()">上下反転 Test</p>
<p onmouseover="glowTest()">発光 Test</p>
</div>
<div id="sousal" style="position:absolute;top:200;left:400">
<p onmouseover="Test.style.filter='gray()'">グレイ Test</p>
<p onmouseover="Test.style.filter='invert()'">カラー反転 Test</p>
<p onmouseover="Test.style.filter='shadow(color=gray,direction=150)'">
影 Test</p>
<p onmouseover="Test.style.filter='wave(add=0,freq=3,lightstrength=2,
phase=10,strength=1)'">波 Test</p>
<p onmouseover="Test.style.filter='Xray()'">モノクロ Test</p>
</div>
<div id="sousal" style="position:absolute;top:500;left:120">
<table width="100%">
<tr>
<td onmouseover="kaeru('black')">黒</td>
<td onmouseover="kaeru('red')">赤</td>
<td onmouseover="kaeru('yellow')">黄</td>
<td onmouseover="kaeru('blue')">青</td>

```

```
</tr>
</table>
</div>
</body>
</html>
```

8.5 クイズの例 (sample5.htm)

```
<html>
<head>
<title>クイズ</title>
</head>
<body>
<form onsubmit="if(this.elements[0].checked==true){alert('当然です
ね')}else{alert('そんなわけ無いでしょ')}">
問題 <b>信号で，赤は何を意味しているか？</b>
<p>
<input type="radio" name="q">止まれ<br>
<input type="radio" name="q">気をつけて行け<br>
<input type="radio" name="q">気にせず行け<br>
<input type="submit" value="答え">
</form>
</body>
</html>
```

参考文献

- [1] 株式会社アंक，Dynamic HTML がやってきた！，翔泳社 (1998) .
- [2] 古籟一浩 監修，松尾忠則，半場方人，すぎうらしろう，Dynamic HTML でつくるホームページ，株式会社技術評論者 (1998)
- [3] 小田政明，やさしい数値計算法，日刊工業新聞 (1994)