

JavaScript の基礎

伊藤泰子

NDEC Web Magazine 2

2023.1.10

京都ノートルダム女子大学教育センター

<https://www.notredame.ac.jp/ndec/index.html>

e-mail to: vsadmin(a)ml.notredame.ac.jp

はじめに

プログラム言語 JavaScript を利用した動的な Web コンテンツの作成について、プログラムの基本制御構造、演算子、オブジェクト指向、イベント、関数 (Function) の定義の 5 項目にわたって解説しています。Web の情報を単に受け取るだけの利用者ではなく、インタラクティブな情報発信の技術を養い、プログラミングの技術、Web サーバとのやりとりなど、Web における一歩進んだ知識・技術を身につけることができるようになるための手引きとして構成しました。

JavaScript

JavaScript は Netscape 社が開発したスクリプト言語です。HTML ドキュメント (Web ページ) に埋め込まれ、ブラウザがそのページを表示する際にスクリプトの記述を解釈し、実行されます。

スクリプト言語: 機械語への変換を簡単に実行できるようにした簡易プログラム言語。

※「Java」とは別の言語です。

実習問題 1

サンプルプログラム「exe1.html」を作成しましょう。

(テキストエディタで以下のプログラムを入力)

※大文字・小文字の区別、スペル間違いに注意しましょう。

```
<!DOCTYPE HTML>
<html lang="ja">

<head>
<meta charset="UTF-8">
<title>演習問題 1</title>

<script type="text/javascript">

document.write( 'Hello<br>' );

</script>

</head>
<body>
</body>
</html>
```

<Point>

- ・ 処理内容

document.write(): 文字列を HTML 文書に出力するメソッド

実習問題 2

実習問題 1 (exe1.html) をもとにして、以下の 3 行を表示するようにしましょう。

Hello
こんにちは
Bonjour

ファイル名 : exe2.html

1. プログラムの基本制御構造

1. 順次 (連続) 上から順に処理をすすめる。
2. 反復 (繰返し) 条件や回数によって処理を繰り返す。
3. 選択 (分岐) 条件によって処理を分岐する。

1-2-1. 反復 (繰返し) for 文による繰返し

書式)

```
for( 初期化式 ; 繰返し条件式 ; ループの更新式 ) {  
    繰り返す処理  
}
```

例)

```
for( i=0 ; i<5 ; i++ ) { 繰り返す処理 }
```

1-2-2. 反復 (繰返し) while 文による繰返し

書式)

```
while( 繰返し条件式 ) {  
    繰り返す処理  
}
```

例) while(i<5) { 繰り返す処理 }

1-3-1. 選択 (分岐) if 文による分岐処理

書式 1)

```
if( 条件式 ) {  
    処理  
}
```

書式 2) if(条件式) {

```
    処理  
}else if( 条件式 ) {  
    処理  
}else if( 条件式 ) {  
    処理  
}else {  
    処理  
}
```

例 1) if(a>3) { 処理 }

例 2) if(a>=60 && b>=60) { 処理 }
else if(a==0 || b==0) { 処理 }
else { 処理 }

1-3-2. 選択 (分岐) switch case 文による多分岐

```
書式) switch( 条件式または値 ){
    case 値 1 :
        処理
        break;
    case 値 2 :
        処理
        break;
    default:
        処理
        break;
}
```

```
例) switch( a%3 ){
    case 0 :
        処理
        break;
    case 1 :
        処理
        break;
    default:
        処理
        break;
}
```

2. 演算子

演算子 . . . 基本的な演算を実行する命令

算術演算子 . . . 加減乗除などの基本的な計算を実行する演算子

演算子	例	説明
+	$a + b$	加算。a と b を足す。
-	$a - b$	減算。a から b を引く。
*	$a * b$	乗算。a と b を掛ける。
/	a / b	除算。a を b で割った商。
%	$a \% b$	剰余。a を b で割った余り。
++	$a ++$	インクリメント。a の値を 1 増やす。
--	$a --$	デクリメント。a の値を 1 減らす。

代入演算子 . . . 変数に値を代入する演算子

演算子	例	説明
=	$a = 3$	変数 a に 3 を代入する。
	$a = b + c$	変数 a に $b + c$ したものを代入する。

関係演算子 . . . 2 つの値を比較し、結果を true または false を返す演算子

演算子	例	説明
>	$a > 3$	a が 3 より大きいとき
>=	$a >= 3$	a が 3 以上のとき
<	$a < 3$	a が 3 より小さいとき
<=	$a <= 3$	a が 3 以下のとき
==	$a == 3$	a が 3 と等しいとき 等価演算子
!=	$a != 3$	a が 3 以外するとき 非等価演算子
===	$a === 3$	a が 3 と等しいとき 厳密等価演算子
!==	$a !== 3$	a が 3 以外するとき 厳密非等価演算子

条件演算子 . . . 複数の条件の関係を表す演算子

演算子	例	説明
&&	$a == 0 \ \&\& \ b == 0$	a が 0 かつ b が 0 のとき (両方)
	$a == 0 \ \ b == 0$	a が 0 もしくは b が 0 のとき (どちらか)

実習問題 3

for 文を使って、「Hello」を 10 行繰り返すプログラムを作成しましょう。
ファイル名: exe3.html

<アレンジ>

1) 行番号を表示してみましょう。

```
document.write( i + ' Hello<br>' );
```

2) 100 行表示するようにしてみましょう。

行番号は、0 から 99 までになります。

実習問題 4

while 文を使って、「Hello」を 10 行繰り返すプログラムを作成しましょう。
ファイル名: exe4.html

実習問題 5

実習問題 3 (exe3.html) を元にして、if 文を使って行番号 4, 5 の間に区切り線を入れてみましょう。

ファイル名: exe5.html

実習問題 6

if 文を使って、偶数 (2 の倍数) を表示するプログラムを作成しましょう。
ファイル名: exe6.html

ヒント)

```
for( i=0 ; i<100 ; i++){
    if( i を 2 で割った余り が 0 と等しければ ){
        document.write( i + ' Hello<br>' );
    }
}
```

実習問題 7

実習問題 6 (exe6.html) を元にして、 i を 3 で割った余りが 0 の場合「Hello」、余りが 1 の場合「こんにちは」、それ以外の場合「Bonjour」と表示するプログラムを作成しましょう。

ファイル名 : exe7.html

ヒント)

```
for( i=0 ; i<100 ; i++) {
    if( i を 3 で割った余り が 0 と等しければ ){
        document.write( i + ' Hello<br>' );
    }else if ( 条件文 ) {
        処理
    }else{
        処理
    }
}
```

実習問題 8

- 1) 実習問題 5 (exe5.html) を元にして、10 行ごとに区切り線が表示するようにしましょう。
- 2) 区切り線を 2 パターン交互 (10 行、5 行に区切り線) に表示するようにしてみましょう。

ファイル名 : exe8.html

実習問題 9

if 文を使って、ランダムに「グー」「チョキ」「パー」のいずれかを表示するプログラムを作成しましょう。

ファイル名 : exe9.html

<Point>

- ・ `Math.random()` 0.0 以上、1.0 未満の擬似乱数的に返すメソッド
- ・ `Math.floor()` 小数点以下の数値を切り捨てるメソッド

例

<code>Math.random()</code> の値	<code>Math.random()*3</code>	<code>Math.floor(Math.random()*3);</code>
0.0	0.0	0
0.35	1.05	1
0.7	2.1	2
0.99	2.97	2

<説明>

「`i = Math.floor(Math.random()*3);`」とすると、変数 `i` には、メソッド実行ごとにランダムに、0, 1, 2 が入ります。`i` が 0 の場合には「グー」、1 の場合には「チョキ」、2 の場合には「パー」と画面に表示します。

実習問題 10

実習問題 9 を `switch case` 文を使ったプログラムに変更しましょう。

ファイル名 : exe10.html

3. オブジェクト指向

オブジェクト指向とは

現実世界の物と同様にプログラムを作成できるようにした考え方

オブジェクト指向の利点

1. カプセル化
2. 部品の独立性
3. 部品の再利用性

[キーワード]

オブジェクト	物。状態（プロパティ）とふるまい（メソッド）をもつ。
プロパティ	オブジェクトの属性。
メソッド	オブジェクトに対する動作。
インスタンス	生成されたオブジェクト。

[記述方法]

オブジェクト名.メソッド () ;
オブジェクト名.プロパティ=指定する値 ;

例)

```
document.write( 'Hello<br>' );  
history.back();
```

ドキュメントに Hello と表示する
ひとつ前のページに戻る

```
document.backgroundColor = '#ffff00';
```

ドキュメントの背景色を黄色にする

```
document.write( navigator.appName );
```

ドキュメントにブラウザの種類を表示する

参考 :

<http://www.tohoho-web.com/js/basic.htm#object>
<http://www.scollabo.com/banban/java/object.html>
http://www.scollabo.com/banban/java/ref_13.html

HTML ドキュメントをロードすると、JavaScript で利用可能なオブジェクトを自動的に生成します。

オブジェクトは、階層構造をもっており、それぞれのオブジェクトのプロパティを指定するためには、最上位からの親の名前を記述します。

但し、window オブジェクトの部分だけは省略することができます。

主なオブジェクト

<ブラウザに関するオブジェクト>

- ・ **window オブジェクト**

トップレベルのオブジェクト。

location、history、document、Frame の 4 つの下位オブジェクトをもつ。

- ・ **location オブジェクト**

現在表示しているアドレス (URL) に関する情報をもつオブジェクト。

- ・ **history オブジェクト**

ひとつ前に見ていたページ、ふたつ前に見ていたページなどの、ウィンドウのヒストリ情報を保持、制御するオブジェクト。

- ・ **document オブジェクト**

表示している文書自体を示すドキュメントオブジェクト。

<その他>

- ・ **ビルトインオブジェクト**

JavaScript が独自にブラウザに組み込んでいるオブジェクト。

Array、Boolean、Date、Event、Function、Math、Number、Object、RegExp、screen、String

実習問題 1 1

以下のプログラムを作成しましょう。

ファイル名: exe11.html

下線部 (1~3) は記述しないでください。

```
<!DOCTYPE html>
<html lang="ja">

<head>
<meta charset="utf-8">
<title>実習問題 1 1</title>

<script type="text/javascript">

document.write( 'ブラウザ名:' + navigator.appName + '<br>' );
アレンジ 1) を記述

window.alert( 'alert メソッドの実行' );

i = 10;
アレンジ 2) を記述

document.bgColor = '#ffff00';
アレンジ 3) を記述

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

<アレンジ>

- 1) ドキュメントの最終更新時刻と、ドキュメントのタイトルを表示しましょう。
- 2) alertメソッドを使って「i」の値を表示しましょう。
- 3) ドキュメントの文字色を変更しましょう。

<メソッド>

ウィンドウのメソッド

window.open(url, name [, style]) ウィンドウの表示

新しいウィンドウを開く。引数はすべて省略可能。

url 表示する URL を指定

name ウィンドウの名前

style ステータスバーの表示/非表示などウィンドウに関するいろいろなオプション

参考 http://www.tagindex.com/javascript/window/open_option.html

リターン値: ウィンドウ名

ダイアログボックスのメソッド

window.alert(message) 警告ダイアログ

指定した message と [OK] ボタンのついたダイアログを表示。

※メッセージ内の改行: 「¥n」

リターン値: なし

window.confirm(message) 確認ダイアログ

指定した message と [OK]、[キャンセル] ボタンのついたダイアログを表示。

リターン値: true, false

window.prompt(message [, default]) 文字列入力ダイアログ

指定した message と [OK]、[キャンセル] ボタンのついた入力用のダイアログを表示。

default (省略可) は、文字列の初期値を指定。

リターン値: [OK] の場合は入力された文字列、[キャンセル] の場合は null

<プロパティ>

window.navigator.appName ブラウザ名を表す文字列

document.backgroundColor ドキュメントの背景色

document.fgColor ドキュメントの前景色 (文字色)

document.lastModified ドキュメントの最終更新時刻

document.title ドキュメントのタイトル

実習問題 1 2

以下のプログラムを作成しましょう。
ファイル名: exe12.html

※script 部分のみを記述しています。適宜 HTML の記述を書き加えてください。

```
<script type="text/javascript">

ret = window.confirm( '背景色を変更しますか?' );
document.write( ret );

</script>
```

<アレンジ>

if 文を利用して、確認ダイアログ (confirm) で [OK] ボタンをクリックした場合のみ、背景色を変更するプログラムに書き換えましょう。

ヒント)

if 文 (p. 3)、関係演算子 (p. 5)、ドキュメントの背景色を示すプロパティ (p. 12) を利用します。

※背景色は好きな色にしてください。

実習問題 1 3

confirm メソッドを利用し、[OK] ボタンをクリックした場合、新しいウィンドウを開くプログラムを作成しましょう。

- ・新しくウィンドウ
「url」「name」「style」には何も指定しない
- ・確認メッセージ
「新しいウィンドウを開きますか？」

ファイル名 : exe13.html

<アレンジ>

url: <http://www.yahoo.co.jp>

name: subwin

style: 任意 (いろいろな指定を試してみましょう)

http://www.tagindex.com/javascript/window/open_option.html 参照

実習問題 1 4

以下のプログラムを作成しましょう。

ファイル名 : exe14.html

※script 部分のみを記述しています。適宜 HTML の記述を書き加えてください。

```
<script type="text/javascript">

document.write( ' <h1>背景色・文字色を指定できます。 </h1>' );
ret = window.prompt( '背景色を入力してください。' , '' );
document.write( ret );

</script>
```

※色番号は、「#ff0000」というように、#記号と数字6桁でしてします。

<アレンジ>

- 1) prompt メソッドのリターン値が「null」以外の場合は、背景色を変更するようにしてみましょう。
- 2) 文字色を変更する命令も書き加えてみましょう。

実習問題 15

prompt メソッドを利用し、URL を指定し「OK」ボタンをクリックした場合、その URL を表示するウィンドウを新しく開くプログラムを作成しましょう。

ファイル名: exe15.html

※ window.open メソッドの「name」は「subwin」、「style」はおまかせします。

実習問題 16

prompt メソッドを利用し、新しく開くウィンドウの数を指定し「OK」ボタンをクリックした場合、指定数のウィンドウを開くプログラムを作成しましょう。

ファイル名: exe16.html

※ for 文 (p. 3) を利用します。

※ window.open メソッドの「url」「style」はおまかせします。「name」は「subwin」にします。

→ 指定した数のウィンドウが表示されましたか？

ウィンドウの数をいくつに指定しても1つしかウィンドウが開かないですね。

<アレンジ>

window.open メソッドの「name」を「subwin + i」などとし、異なるウィンドウ名を指定するようにしましょう。

実習問題 17

以下のプログラムを作成しましょう。

ファイル名: exe17.html

※script 部分のみを記述しています。適宜 HTML の記述を書き加えてください。

```
<script type="text/javascript">

document.write( ' <h1>現在時間の表示</h1>' );

mytime = new Date();
y = mytime.getFullYear();
m = mytime.getMonth()+1;
d = mytime.getDate();
h = mytime.getHours();
min = mytime.getMinutes();
sec = mytime.getSeconds();

day = mytime.getDay();

document.write( y + '年' + m + '月' + d + '日 ' + h + '時' + min +
'分' + sec + '秒<br>' );

document.write( day + '曜日<br>' );

</script>
```

<point>

- ・ new 演算子 オブジェクトの新しく生成します。

<アレンジ>

- 1) 変数 day には、0 (日曜日) ~6 (土曜日) までの値が代入されています。
日曜日、月曜日など単語で表示するように変更しましょう。
- 2) 現在時刻によって異なるメッセージを表示しましょう。

8 時以降 12 時未満 (20 秒以降 35 秒未満)	「おはよう。」
12 時以降 18 時未満 (35 秒以降 50 秒未満)	「こんにちは。」
18 時以降 2 時未満 (50 秒以降 5 秒未満)	「こんばんは。」
2 時以降 8 時未満 (5 秒以降 20 秒未満)	「就寝中。」

※関係演算子、条件演算子 (p.5) 参照

4. イベント

イベントとは

クリックやキー入力などのユーザ操作や、メッセージの受信など、あるオブジェクトに対して発生したアクションをイベントといいます。このイベントの発生を検知し、何らかの処理を実行させる機能をイベントハンドラといいます。

イベントハンドラ一覧

イベントハンドラ	イベントが起こるタイミング
onBlur	ページやフォーム要素がフォーカスを失った時
onFocus	ページやフォーム要素がフォーカスされた時
onChange	フォーム要素の選択、入力内容が変更された時
onSelect	テキストが選択された時
onSubmit	フォーム内容を送信しようとした時
onReset	フォーム内容がリセットされた時
onAbort	画像の読み込みを中断した時
onError	画像の読み込み中にエラーが発生した時
onLoad	ページや画像の読み込みが完了した時
onUnload	ウィンドウを閉じた時、他のページに切り替えた時、ページをリロード（更新）した時
onClick	要素やリンクをクリックした時
onDblick	要素をダブルクリックした時
onKeyUp	押していたキーをあげた時
onKeyDown	キーを押した時
onKeyPress	キーを押している時
onMouseOut	マウスを離れた時
onMouseOver	マウスを重ねた時
onMouseUp	クリックしたマウスを上げた時
onMouseDown	マウスでクリックした時
onMouseMove	マウスを動かしている時

5. 関数 (function) の定義

ひとまとまりの処理に名前をつけて関数として定義することができます。
この関数を呼び出すことで、必要な時に実行したい処理をすることができます。

書き方)

```
function 関数名 ( [引数 1、引数 2...] ){  
    処理  
}
```

※1 引数 (パラメータ) は省略可能。

※2 1つページ内で、同じ関数名は使用できません。

例1)

```
function changeBGColor () {  
    document.bgColor = '#ff0000';  
}
```

例2)

```
function changeBGColor( col ){  
    document.bgColor = col;  
}
```

実習問題 18

背景色を変更するプログラムを作成しましょう。

ファイル名: exe18.html

```
<!DOCTYPE html>
<html lang="ja">

<head>
<meta charset="utf-8">
<title>実習問題18</title>

<script type="text/javascript">

function changeBGColor() {
    document.bgColor = '#ff0000';
}

</script>
</head>
<body>
<h1>背景色・文字色の変更</h1>
<hr>
<form>
    <input type="button" value="背景色を赤" onclick="changeBGColor();"><br>
    <input type="button" value="文字色を緑" onclick=""></form>
</body>
</html>
```

<アレンジ>

- 1) 文字色を緑にする関数「changeFGColor」を作成し、「文字色を緑」ボタンクリックで呼び出すように変更しましょう。
少し暗い緑: #009900
- 2) 背景色を白 (#ffffff) にする関数「changeBGColorW」、文字色を黒 (#000000) にする関数「changeFGColorB」を作成し、それぞれ実行できるようにしましょう。
- 3) 「changeBGColor」「changeFGColor」を引数のある関数に書き換えましょう。

- ・関数の記述

```
function changeBGColor( col ) {  
    document.bgColor = col;  
}
```

- ・呼び出し時の記述

```
<input type= "button" value="背景色を赤"  
    onclick="changeBGColor( ' #ff0000' );">
```

「changeFGColor」も同様に変更しましょう。

※ダブルクォーテーション内で文字列を表す場合は、シングルクォーテーションを使用します。

- 4) 背景色、文字色の変更ができるボタンをいくつか作成しましょう。

参考)

- ・紹介したプログラムでの色設定

	赤系	緑系	青系
背景色	#ff0000	#00ff00	#0000ff
文字色	#990000	#009900	#000099

- ・Web Safe Color 一覧表/カラーチャート

<http://www.scollabo.com/banban/lectur/websafe.html>

実習問題 19

いろいろなフォーム部品（セレクトボックス、テキストフィールド）を利用するプログラムを作成しましょう。

ファイル名 : exe19.html

※body 部分のみを記述しています。適宜 HTML の記述を書き加えてください。

```
<body>
<h1>背景色・文字色の変更</h1>
<hr>
<form>
<label>背景色選択</label>
<select id="bgCol">
  <option value="#ffffff" selected="selected">白</option>
  <option value="#ff0000">赤</option>
  <option value="#00ff00">緑</option>
  <option value="#0000ff">青</option>
  <option value="#000000">黒</option>
  <option value="#ffff00">黄</option>
</select>

<br>

<hr>

<input type="text" value="#" size="10" id="colText">
<input type="button" value="背景色を変更" onclick="">

</form>
</body>
</html>
```


<アレンジ>

1) セレクトボックスで値を変更すると背景色を指定した色に変更するようにしましょう。

- ① セレクトボックスの値を変更したときに関数を呼び出します。

「onchange="changeBGColor();"」

- ② 背景色を変更する関数を作成します。

関数名 : changeBGColor

処理内容

```
document.bgColor = document.getElementById('bgCol').value;
```

2) 同様に文字色も変更できるようにしましょう。

文字色のセレクトボックスの id : fgCol

関数名 : changeFGColor

3) 「背景色を変更」ボタンをクリックすると背景色をテキストフィールドで指定した色に変更するようにしましょう。

関数名 : changeColor

処理内容

ドキュメントの背景色に id:colText の値 (value) を代入します。

4) 次のステップへの準備です。

- ・関数「changeColor」の変更
引数付きの関数にします。

引数名 : mode

処理内容

mode の値が「0」ならば、背景色を変更する。

- ・呼び出し方の変更

<input type="button" value="背景色を変更" onclick="changeColor();"> を
<input type="button" value="背景色を変更" onclick="changeColor(0);">
と変更します。

5) テキストフィールドで指定した値で、文字色も変更できるようにしましょう。

- ・ボタンを追加

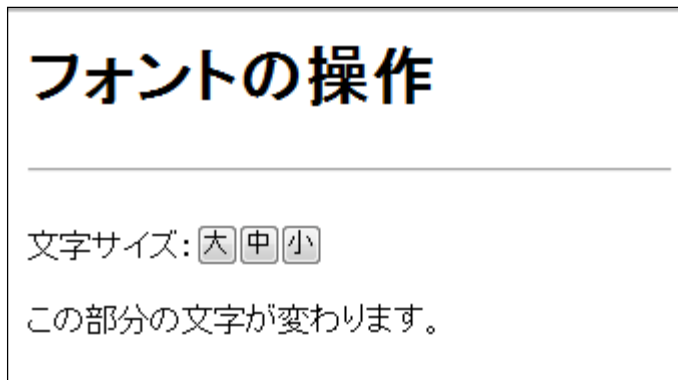
クリック時の処理 onclick="changeColor(1)";

- ・関数「changeColor」に、mode が「1」の場合の処理を追加

実習問題 20

文字サイズを変更するプログラムを作成しましょう。
ファイル名: exe20.html

完成図



ヒント)

- ・「この部分の文字が変わります。」は、以下のようにマークアップしましょう。

```
<div id="contents">  
  この部分の文字が変わります。  
</div>
```
- ・関数は1つだけ作成します。
引数によって、大中小の文字サイズに変更するようにしましょう。
- ・大中小のボタンをクリックすることで、id:「contents」部分の文字サイズを変更します。

指定 ID の文字サイズ変更

```
document.getElementById(ID).style.fontSize = 値;  
値: xx-small, x-small, small, medium, large, x-large, xx-large
```

例)

```
document.getElementById('contents').style.fontSize = 'xx-large';
```

どの文字サイズを選択するかはお任せします。

<アレンジ>

1) id:「contents」部分の文字色を変更しましょう。

指定 ID の文字色変更

```
document.getElementById(ID).style.color = 値;  
値: 色番号 (#ffffff , #000000)、色名 (white, black, red, blue)
```

例)

```
document.getElementById('contents').style.color = '#00ffff';  
document.getElementById('contents').style.color = 'blue';
```

2) id:「contents」部分の表示内容を変更しましょう。

指定 ID の表示内容変更

```
document.getElementById(ID).innerHTML = 文字列;
```

例)

```
document.getElementById('contents').innerHTML = 'Hello';  
document.getElementById('contents').innerHTML = '<h1>Bye</h1>';
```

実習問題 2 1

背景画像を変更するプログラムを作成しましょう。

ファイル名: exe21.html

※body 部分のみを記述しています。適宜 HTML の記述を書き加えてください。

```
<body id="mybody">
<h1>背景画像の変更</h1>
<form>
  <input type="button" style="background:#fff url(img/r2.gif); width:
60px; height:60px;" onclick="changeBGImage('url(img/r2.gif)');">
  <input type="button" style="background:#fff url(img/yg3.gif); width:
60px; height:60px;" onclick="changeBGImage('url(img/yg3.gif)');">
  <input type="button" style="background:#fff url(img/p2.gif); width:
60px; height:60px;" onclick="changeBGImage('url(img/p2.gif)');">
  <input type="button" style="background:#fff url(); width:60px; height:60px;"
onclick="changeBGImage('url()');">
</form>
</body>
</html>
```

・ style 属性 タグに対して直接スタイルを指定します。

上記の例では

background:#fff url(img/r2.gif); 背景指定 (色、画像)

※背景画像は、url(ファイル名)と表記します。

width:60px; 幅

height:60px; 高さ

ヒント)

関数名 : changeBGImage

引数 : value

処理内容

document.getElementById('mybody').style.backgroundImage = value;

<アレンジ>

・セレクトボックスでも背景色を変更できるようにしてみましょう。

実習問題 2 2

画像のロールオーバープログラムを作成しましょう。

ファイル名: exe22.html

配布教材: 画像ファイル (img フォルダ)

※script、body 部分のみを記述しています。適宜 HTML の記述を書き加えてください。

```
<script type="text/javascript">
function changeImg( filename ){
    document.getElementById( 'img1' ).src = filename;
}
</script>
</head>
<body>
<h1>マウスオーバーによる画像の変化</h1>
<hr>
<div>
    
</div>
</body>
</html>
```

<アレンジ>

現在 1 つ画像のロールオーバーができます。

3 つの画像をロールオーバーできるようにしてみましょう。

ヒント)

- ・関数「changeImg」に src を変更したい画像の id を渡します。
以下のように、引数を 2 つにするわけです。

```
function changeImg( id , filename ){ . . . }
```

※使用する画像は、各自用意しましょう。

実習問題 2 3

ロールオーバーした画像を拡大表示するプログラムを作成しましょう。

ファイル名: exe23.html

※body 部分のみを記述しています。適宜 HTML の記述を書き加えてください。

```
<body>
<h1>マウスオーバーによる画像の変化</h1>
<hr>
<div>
  
  
  
  
</div>
<hr>
<div>
  
</div>
</body>
```

ヒント)

- 関数「zoomImg」を作成しましょう。

```
function zoomImg( filename ){
    処理: id「zoom」の「src」を「filename」にする。
}
```
- 各画像にマウスオーバーした時に、関数「zoomImg」を呼び出すようにしましょう。

実習問題 2 4

四則演算をするプログラムを作成しましょう。

ファイル名: exe24.html

完成図

※script 部分の記述です。

```
function calcFunc() {  
    document.getElementById( 'ans' ).innerHTML = '計算結果';  
}
```

※body 部分の記述です。

```
<h1>四則演算</h1>  
<hr>  
<form>  
  
<label>=</label>  
<span id="ans"></span>  
<br>  
<input type="button" value="計算" onclick="calcFunc();">  
</form>
```

<アレンジ>

- 1) 値を入力するテキストフィールド2つ、演算子を選択するセレクトボックス1つを追加しましょう。

作成条件)

各テキストフィールドの id	「x」 「y」
セレクトボックスの id	「op」
セレクトボックスの各要素の value	「+」 「-」 「*」 「/」

- 2) 「計算」 ボタンをクリックすると計算実行するように関数「calcFunc」内の記述を変更しましょう。

→できましたか? できないでしょ。解答例は後ほど。

実習問題 2 5

BMI を計算するプログラムを作成しましょう。

ファイル名 : exe25.html

1) まずは、フォーム部品を記述しましょう。(計算ボタンを押しても何も実行されません。)

※script、body 部分のみを記述しています。適宜 HTML の記述を書き加えてください。

```
<script type="text/javascript">
function bmiFunc() {
}
</script>
</head>
<body>
<h1>標準体重と肥満度の計算</h1>
<form>
<label>身長 (cm) :</label> <input type="text" size="10" id="height"><br>
<label>体重 (Kg) :</label> <input type="text" size="10" id="weight"><br>
<br>
<input type="button" value="計算" onclick="bmiFunc()"><br><br>
<hr>
<h2>結果</h2>
<label>標準体重 :</label> <span id="std"></span><br>
<label>肥満度 :</label> <span id="bmi"></span><br><br>
<span id="bmi2"></span>
</form>
</body>
```

・ BMI 計算式

標準体重 (kg) = 身長 (m) × 身長 (m) × 22

肥満度 = 体重 (kg) ÷ (身長 (m) × 身長 (m))

- 2) 標準体重を計算し、その結果を表示するようにしましょう。
id「height」の値を利用して計算し、その結果を id「std」に表示します。
- 3) 肥満度を計算し、その結果を表示するようにしましょう。
id「height」、id「weight」の値を利用して計算し、その結果を id「bmi」に表示します。
- 4) 標準体重・肥満度を、小数点第三位を四捨五入した形式で表示しましょう。

※「Math.round()」小数点以下を四捨五入するメソッドを利用します。

使用例)

```
a = Math.round( 4.567 );           a には 5 が代入されます。  
b = Math.round( 4.567 * 10 );      b には 46 が代入されます。  
c = Math.round( 4.567 * 10 ) / 10 ; c には 4.6 が代入されます。
```

- 5) BMI 判定結果を表示しましょう。
下記の BMI 判定をもとに、if 命令を使って肥満度を判定し、その指標を id「bmi2」に表示しましょう。

・ BMI 判定

状態	指標
低体重	18.5 未満
普通体重	18.5 以上、25 未満
肥満 (1 度)	25 以上、30 未満
肥満 (2 度)	30 以上、35 未満
肥満 (3 度)	35 以上、40 未満
肥満 (4 度)	40 以上

- 6) BMI 判定結果を異なる文字色で表示しましょう。

実習問題 26

google map を表示するプログラムを作成しましょう。

ファイル名 : gmap.html

1)

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
<style type="text/css">
    html { height: 100% }
    body { height: 100%; margin: 1em; padding: 0px }
    #map_canvas { height: 100% }
</style>
<script type="text/javascript" src="http://maps.google.com/maps/api/js?sensor=false">
</script>
<script type="text/javascript">
var map;
var geocoder;
function initialize() {
    var latlng = new google.maps.LatLng(35.050421, 135.772473);
    var myOptions = {
        zoom: 13,
        center: latlng,
        disableDefaultUI: true,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    };
    var map = new google.maps.Map(document.getElementById("map_canvas"), myOptions);
}
</script>
<title>google map</title>
</head>
<body onload="initialize()">
<h1>google map</h1>
<div id="map_canvas" style="width: 500px; height: 500px;"></div>
</body>
</html>
```

2) いろいろ変更してみましょう。

①地図のサイズ

②表示している位置

自分の家を表示するように変更しましょう。

経緯、緯度を調べる: <http://www.geocoding.jp/>

③myOptions の設定値の変更

I zoom レベル

II disableDefaultUI: true, を
disableDefaultUI: false , としてみると?

III mapTypeId: google.maps.MapTypeId.ROADMAP の「ROADMAP」を
「HYBRID」, 「SATELLITE」, 「TERRAIN」のいずれかに変更してみると?

参考:

<https://developers.google.com/maps/documentation/javascript/reference?hl=ja>

3) テキストボックスに任意の地名を入れると、その場所を表示するようにしてみましょう。

「Geocoder」クラスの利用

```
<!DOCTYPE html>
<html><head><meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
<style type="text/css">
  html { height: 100% }
  body { height: 100%; margin: 1em; padding: 0px }
  #map_canvas { height: 100% }
</style>
<script type="text/javascript" src="http://maps.google.com/maps/api/js?sensor=false"></script>
<script type="text/javascript">
var map;
var geocoder;
function initialize() {
  var latlng = new google.maps.LatLng(35.050421, 135.772473);
  var myOptions = {
    zoom: 13,
    center: latlng,
    disableDefaultUI: false,
    mapTypeId: google.maps.MapTypeId.ROADMAP
  };
  map = new google.maps.Map(document.getElementById("map_canvas"), myOptions);
  geocoder = new google.maps.Geocoder();
}
function showAddress() {
  var address = document.getElementById('address').value;
  geocoder.geocode( { 'address': address }, function(results, status) {
    if (status == google.maps.GeocoderStatus.OK) {
      map.setCenter(results[0].geometry.location);
      var marker = new google.maps.Marker({
        map: map,
        position: results[0].geometry.location
      });
    } else {
      alert('Geocode was not successful for the following reason: ' + status);
    }
  });
}
</script>
<title>google map</title>
</head>
<body onload="initialize()">
<h1>google map</h1>
  <form>
    <input id="address" type="text" size="60" value="京都府京都市">
    <input type="button" value="表示" onclick="showAddress()">
  </form>
  <hr>
  <div id="map_canvas" style="width: 500px; height: 500px;"></div>
</body></html>
```

4) マーカーをクリックするとウィンドウ表示するようにしてみましょう。
「infolowindow.open」メソッドの利用

- ・ script 部分の変更 二重枠線の部分を書き加えましょう。

```
if (status == google.maps.GeocoderStatus.OK) {
  map.setCenter(results[0].geometry.location);
  var marker = new google.maps.Marker({
    map: map,
    position: results[0].geometry.location
  });
  var infowindow = new google.maps.InfoWindow({
    content: address
  });
  google.maps.event.addListener(marker, 'click', function() {
    infowindow.open(map, marker);
  });
} else {
```

5) セレクトボックスの利用

- ① Script 部分の変更 二重枠線の部分を変更しましょう。

```
function showAddress(no) {
  var address = document.getElementById("location").options[no].value;
```

- ②html 部分の変更 二重枠線の部分を変更しましょう。

```
<form>
  <select id="location" onchange="showAddress(this.selectedIndex)">
    <option value="京都府京都市">場所を選択してください</option>
    <option value="京都市">京都市</option>
    <option value="札幌市">札幌市</option>
    <option value="福岡市">福岡市</option>
    <option value="那覇市">那覇市</option>
  </select>
</form>
```

6) マーカーのメッセージをカスタマイズ

①表示するメッセージの配列を作成します。

Script 部分の変更 二重枠線の部分を書き加えましょう。

```
var map;
var geocoder;
var infoString = new Array(
    "<h1>京都市</h1><div><a href='http://www.city.kyoto.lg.jp/' >
click!</a></div>",
    "<h1>祇園祭</h1><div><a href='http://www.gionmatsuri.jp/' >京
都祇園祭</a></div>",
    "<h1>雪まつり</h1><div><a href='http://www.snowfes.com/' >さ
っぽろ雪まつり公式サイト</a></div>",
    "博多どんたく",
    "ゴーヤチャンプルー"
);
```

②作成したメッセージの配列を infoWindow にセットします。

Script 部分の変更 二重枠線の部分を変更しましょう。

```
var infowindow = new google.maps.InfoWindow({
    content: infoString[no]
});
```

7) セレクトボックスで表示する場所、表示メッセージなどを変更してみましょう。