



ゼミC

第2回

2014年8月5日

エレクトロニクス研究部

JavaScript

JAVASCRIPTとは...

- 1995年ごろから登場した言語。オブジェクト指向のスクリプト言語。
- 実行環境は主にウェブブラウザ。開発当初はLiveScriptと呼ばれていたが、開発当初Javaが大きな注目を浴びていたためJavaScriptと名付けられた。よってJavaとは**関係は無い**。
- Internet Explorerなどに実装されるようになってから急速に普及。そこから多くのウェブブラウザに実装されるようになった。



- ウェブ上のテキストを記述するにはHTML、HyperText Markup Language (ハイパーテキストマークアップ ランゲージ) という言語を使用している。文字を書くだけならこのHTMLだけでいいがもうちょっと複雑なことをしていこうとするときはこのJavaScriptを使用する。なので形はこのHTMLの中に記していく、あるいは外部で用意したものをHTMLから呼び出す形になる。
- このJavaScriptのほかにスタイルシートやレイヤなどの技術を纏めてDHTML (ダイナミックHTML) と呼ぶ。このJavaScriptはDHTMLの中核にあたる。



JAVASCRIPTのための準備

- 前で説明したとおりウェブブラウザがあれば動作することができる。
必要なのは「テキストエディタ」「ブラウザ」。
- 勿論ブラウザ側、IEやFirefox・Google Chrome・Safari もろもろでJavaScriptの設定をオンにしておく必要がある。大体はデフォでオンになっているはず。



HTMLについて

- HTMLについてちょっとだけ。
HTMLはタグ、`<>`この括弧に囲まれたものを使って記していく。開始のタグと終了のタグが必要。必須のタグは以下の4つである。
- `<html> </html>`
このテキストがHTMLであることを記す。
- `<head> </head>`
この中に文章のタイトルなどのヘッダ情報を記す。
- `<title> </title>`
この中にページのタイトル名を記す。
- `<body> </body>`
この中に実際にブラウザ上に表示する文章などを記す。



○ 例

```
<html>

<head>
  <title>テスト</title>
</head>

<body>
  テストです
</body>

</html>
```

これをtest.htmlで保存し、開くとブラウザ上で「テストです」と一文が表示されるはず。

JavaScriptはhead・body内、タグ内、あるいは外部から呼び出す。



JAVASCRIPT

```
<script>
```

ここにJavaScriptのソースを記入

```
</script>
```

head、bodyタグに記す際は上のように**<script>** で囲まれた中に記入する。



なのでこんな感じ

```
<html>

<head>
  <title>テストページ</title>
</head>

<body>
  <script>
    alert("てすとういんどう");
  </script>
</body>

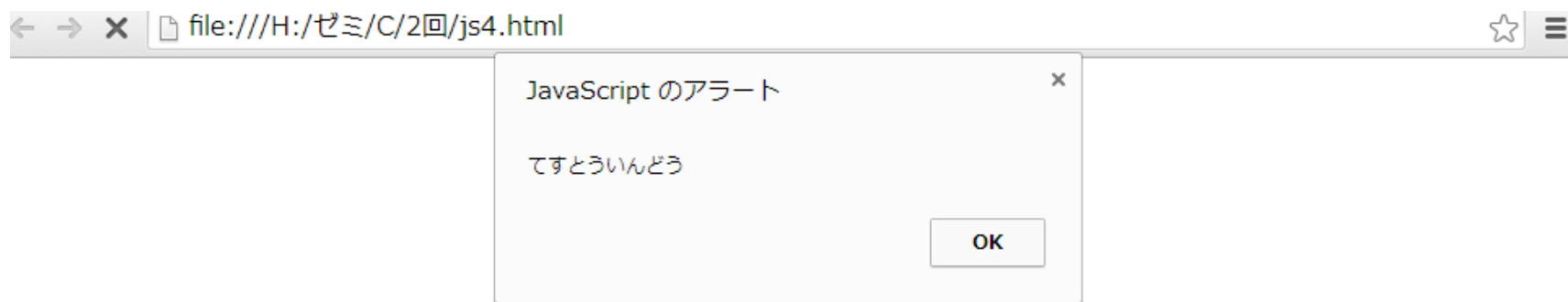
</html>
```

外部から呼び出す方法もありますがとりあえずはこの方法でやっていきます。



アラートの表示

- 前のスライドのプログラムを実行させると以下のように出ます。



表示され方はブラウザによって違います(上のはクローム)

これをアラートといいます。上だと**OK**を押すことでページの内容が表示されるようになります。



```
alert("てすとらいんどう");
```

上のように記すことでアラートを表示させることができます。文字列が入っている部分を差し替えることで他の文字列がアラート時に表示されます。

アラート内で文章を改行したい際はCと同じく¥nを使用する。演算も行うことができる。以下を実行すると10がアラームとして表示される。””をつける文字列として扱われるので注意。

```
alert(3+7);
```

文字列と演算jの結果を一緒に表示させたいときは

```
alert("テスト"+(3+7));
```

このように文字列と演算の結果を+でつなげる。

上を実行すると「テスト10」と表示されたアラートが出る。



変数の宣言

- JavaScriptでは変数を宣言するときは以下のような形になります。

```
var a;
```

こうすることで名前aの変数が宣言されました。
宣言時初期化することもできます。

```
var a = 5;  
var b = "zemiC";
```

さて、見てわかると思いますがJavaScriptでは宣言するときCでやってきたintとかの型の明示をしません。自動的に割り当てられます。
上の例ではaには数字の5、bには文字列「zemiC」が代入されました。



文字列の表示

```
document.write()
```

カッコの中には表示させたい文字列が入ります。”で囲んで表示するようにしましょう。また括弧内に計算式のみ書くことで計算結果を出力することもできます。

例

```
document.write(“10+20の計算結果は”);  
document.write(10+20);
```

出力

```
10+20の計算結果は30
```



ドキュメント内で改行させたいときはC言語と同じ方法では改行できない。

```
<br>
```

上の文字列を出力することで改行することができる。この
はHTML言語であり、そのまま改行を意味している。

```
document.write("<br>");
```

これを実行すると改行される。



また変数、数字と一緒に文字列を表示させたい場合はアラームと同じく+を使うことに注意。

```
document.write(n+“<br>”+(10+5));
```

上を実行するとnに代入された値(数字でも文字列でも可)を出力した後、改行。そして10+5の計算結果15を出力する。

```
document.write(n“<br>”(10+5));
```

これだとエラーが起きる。

```
document.write(“n+<br>+(10+5)”);
```

これだと「n+
+(10+5)」と表示されてしまう。



演算

- Cと同じく整数同士の足し算、引き算などを行うことができます。

```
var a = 10;  
var b = 2;  
document.write(a+b);
```

これを実行すると足した値が表示されます。

またプログラム途中で変数内の値を増やすこともできます。

例) `a=a+10`, `a++` など

足した変数同士が文字列だった場合はその文字列がつながった状態で表示されます。

例) `var a="hello"; var b=" world";`の場合
`a+b`の出力: 「hello world」



- ちなみに

HTMLにおいての文字列の表示はそのまま<body>内にそのまま書くことで表示されます。

<script>の外で文字列を表示しないといけない際はそうするようにしてください。

改行などがHTMLでは
。そのあたりがこんがらがらないようにしてください。

<body>

これでふつうに表示されます

</body>



定数

変数の中の値を変えることができないものを定数という。
似たようなものでCではdefineがあった。

その定数を宣言するにはconstを使う。

```
const a = 100;
```

このように宣言することによりaの中の値は100で固定される。この後にa=200などをしてしてもaの中の値は100で変わらない。



関数

- Cと同じように関数を作成することができる。

```
function 関数名(引数){ 文 };
```

関数を作成する際は**function**を使用する。値を返す**return**を使う際でも使わない際でも**function**を使用するようにする。

また関数は今までのプログラムは**body**の中に書いてきたが**head**の中に書くようにする。

```
function hello(a){ return a*a; };
```

上の関数は引数として渡された**a**を二乗して返す関数**hello**である。引数として渡すときの型の指定も必要はない。



IF文、SWITCH文、FOR文、WHILE文

- Cで使用してきた条件式での演算子、`<`や`<=`、`&&`はそのまま使うことができる。
- ループ文に使用する`continue`文、`break`文も使用可能。
- `if`文、`else`文、`switch`文、`for`文、`while`文、`do while`文もそのまま使用可能。
`for`文は`(var i=0:~` となる。



配列

- 沢山の変数を一気に宣言する際に使ってきた配列、これを宣言する方法はCとJavaScriptでは大きく違う。

```
var 配列名 = new Array(配列の大きさ);
```

```
var a = new Array(10);
```

上の文では大きさ10の配列a、a[0]~a[9]が宣言された。代入する際はa[0]=10、と今までの変数と同じように行う。

```
var b = new Array(3,33,333,"三");
```

このように宣言することで配列大きさを省略することもできる。上の文は大きさ4の配列aが宣言された。それぞれa[0]には3,a[1]には33,a[2]には333,a[3]には文字列「三」が代入されている。



練習問題1

- ブラウザを開くと100回アラートが表示されるようにせよ。
またアラートのメッセージは今が何回目のクリックか回数
を表示するようにせよ。
(つまり100回アラームのOKを押すようにする)

- エラーが起きれば
ブラウザの開発者ツールにて確認



解答:練習問題1

```
<html>
<head>
  <title>PUSH</title>
</head>
<body>
  <script>
    var i = 0;
    for(i=0;i<100;i++){
      alert((i+1)+"回目のクリック");
    }
  </script>

</body>
</html>
```



外部ファイル

- いままでJavaScriptはbody内に書いてきた。しかし外部ファイルとしてJavaScriptのプログラムを用意し、それをHTMLで読み込みその中の関数をHTML内で実行することができる。

```
<script type="text/javascript" src="JSファイルのパス"></script>
```

これをheadに記入することによりJavaScriptファイルの中の関数をbodyで呼び出して使うことができる。

JSはJavaScriptの略称。JavaScriptファイルは拡張子.jsで保存するようになる。



```
function math(a){
    document.write(a+“に10を足すと”+(a+10)+”<br>2倍する
    と”+(a*2)+”です。”);
}
```

- 上のプログラムをmath.jsという名前でも保存してください。

```
<head>
  <script type="text/javascript" src="math.js"></script>
</head>

<body>
<script>
  math(100);
</script>
</body>
```

- 次にこのプログラムを適当な名前でも保存、math.jsと同じ場所に置き実行してみてください。



フォーム

- ブラウザのページ上にある押したりすることができるボタンや文字などを入力するボックスのことをフォームという。このフォームを使用することによってブラウザ上から情報を入力したりすることができる。

X:	<input type="text" value="50"/>
Y:	<input type="text" value="50"/>
	<input type="button" value="合計"/>

これで言うとXの横の箱や「合計」と書かれたボタンがフォーム



テキストボックス

- まずはテキストボックス、文字が表示されたりまた入力出来たりできる箱の作成を行います。
- フォームを扱う際はフォームタグを使用。

```
<form name="名前"> ~~~~ </form>
```

上のタグのところのname="名前"というところには名前が入ります。このnameというものはHTMLのname属性というものの、要素に名前を付けることができます。名前は半角英数字で何でもいいです。後で使うのでつけるようにしましょう。



- テキストボックス、文字が入力できることのできる場所を作成します。作成するにはインプットタグを使用します。

下に表示しているのはインプットタグでテキストボックスを表示する方法です。

```
<input type="text" name="名前" value="テキスト入力" >
```

type="text"というのはインプットタグで表示することができる物の中からテキストボックスを表示させるというプログラム、name="名前"というのは前と同じこのインプットタグの名前の設定、value="テキスト入力"というのはここではテキストボックスの中に表示される文字列を示します。

上のプログラムをそのまま使うとすれば「名前」という名前が付き、「テキスト入力」という文字が表示されているテキストボックスが表示されます。



- テキストボックスの中の文字列は自由に変えることができます。

←この中の文字列

テキスト入力

- 次はこのボックス内に現在表示されている文字列の取得を行いたいと思います。

```
document.form名.input名.value
```

これで指定したform名の中の指定したinput名の中の値を参照することができます。

```
var a=document.testform.testinput.value
```

上のプログラムは変数aに現在「testform」と名前フォーム名の中の「testinput」と名前のインプット名の中の値を代入しています。この時aの値を出力すると入力されていた文字列を出力できます。



○ 逆に

```
document.testform.testinput.value = a;
```

とおくことで指定したテキストボックス内の値をaに書き換えることができます。



ボタン

- 次にボタンを配置していきたいと思います。これで「ボタンを押したとき～の関数を実行する」ということができます。ボタンの配置も同じくフォームタグ、そしてインプットタグを使用します。

```
<input type="button" value="ボタン" onclick="test()">
```

テキストボックスの配置の時はtypeがtextでしたがボタンの場合はここをbuttonに変えます。

valueはここではボタンに表示されている文章を示しています。

テキスト入力

ぼたん

- ここではテキストボックスのvalueの値は「テキスト入力」、ボタンのvalueの値は「ぼたん」です。



- onclickというのはボタンが押された時何の関数を実行するかを示しています。前のプログラムではtest()という関数を実行する、となっています。



```
<html>
<head>
  <title>ARART</title>
  <script>
    function alert(a){
      alert(a);
    }
  </script>
</head>
<body>
  <form name="textbox">
    <input type="text" name="text" value="なんか入力してね"><br>
    <input type="button" value="アラート表示"
onclick="alert(document.textbox.text.value)"><br>
  </form>
</body>
</html>
```

うえのプログラムはテキストボックスに表示されているものをボタンを押すとアラート表示するプログラムです。



練習問題2

- ボタンを押したときテキストボックスに入力された数字の回数だけアラームを出すようなプログラムを書け。
- これを解くにあたって～
テキストボックスから読み込んだものは全部数字であろうと文字列として読み込まれてしまい、そのままでは計算できません。そのため括弧内の文字列が数字であった場合整数値として返し、そうでない場合はNaNで返す関数

```
Number( );
```

を使うようにしてください。



解答

```
<html>
<head>
  <title>PUSH</title>
  <script>
    function push(){
      var i = 0;
      var num = Number(document.form_s.input_s.value);
      for(i=0;i<num;i++){
        alert((i+1)+"回目のクリック");
      }
    }
  </script>
</head>
<body>
  <form name="form_s">
    <input type="text" name="input_s" value="50">
    <input type="button" value="クリック開始" onclick="push()">
  </form>
</body>
</html>
```



その他インプットタグ

- インプットタグによってテキストボックス、ボタンを配置しましたがこのタグでは勿論ほかのものも表示できます。

```
<p><input type="text" name="c1" value="テキスト入力欄"></p>
```

```
<p><input type="password" name="c2" value="パスワード入力欄"></p>
```

```
<p><input type="radio" name="c3" value="ラジオボタン"></p>
```

```
<p><input type="checkbox" name="c4" value="チェックボックス"></p>
```

```
<p><input type="file" name="c5" value="ファイル選択"></p>
```

```
<p><input type="hidden" name="c6" value="隠しデータ"></p>
```

```
<p><input type="submit" value="送信ボタン"></p>
```

```
<p><input type="reset" value="リセットボタン"></p>
```

```
<p><input type="image" src="button.gif" alt="送信ボタン"></p>
```

```
<p><input type="button" value="汎用ボタン"></p>
```

http://www.tagindex.com/html_tag/elements/input.html

- 今回は詳しくは説明しませんが色々面白いものもあるので時間があればぜひ試してみてください。



DOM

- DOMとはなんぞや?
ずばりDocument Object Modelの略...
ドキュメントオブジェクトを使用する際の決まり事みたいなもの。めっちゃ複雑
- DOMで何ができるの?
ページの再読み込みをすることなくページ、つまりHTMLを書き換えることができます。
ボタンを押すと表示されていた文字列が変化する、ページを開くと自動的に変化が起こっていくなどができる。



HTMLの書き換え

```
document.getElementById("id名").textContent  
    =“書き換えられる文字”  
document.getElementById("id名").innerHTML  
    =“書き換えられる文字”
```

id名のついている文字列を書き換えます。

上の1文textContentはテキストとして””の中をそのまま置き換えるのに対し、下の1文innerHTML

- は””の中にHTMLタグ、つまり
とかが含まれていた場合それを適用します。
次にid名というものについて説明をします。



- id名はid属性を使い、つけられた固有の識別名です。ざっくりいうとname属性でつけるnameとさほど変わりません。

```
<div id="id名"> 文章 </div>
```

上のプログラムで文章にid名をつけることができます。つけるさいに<div>タグを使用しています。divタグはそれ自身はそんなに意味を持ちませんがこんなふうに文字列にid名を付けたるときに使います。囲んだ文字列を一塊の範囲としていろいろやる、とでも考えておいてください。

- これと前のスライドで教えたgetElementById~っのを使うことにより指定した範囲の部分の文(HTML)を書き換えることができるようになりました。



- さて、テキストを書きかえるたびにいちいち `document.getElementById("id名").textContent = "a"` だけの `document.getElementById("id名").textContent = "b"` など書くのはプログラムが長くなり面倒です。そのため同じid名の場所を何度も書きかえるようなときは

```
document.getElementById("id名").textContent = "a"
```

この「`document.getElementById("id名")`」を変数に入れてしまうことがオススメです。

```
var a = document.getElementById("id名");
```

普通にこんな感じで代入ができます。変えたい場合は `a.textContent = "変えたい文字列"` で大丈夫です。



つまり

```
document.getElementById("id名").textContent  
                                =“書き換えられる文字”  
document.getElementById("id名").innerHTML  
                                =“書き換えられる文字”
```

と

```
var A = document.getElementById("id名");  
A.textContent =“書き換えられる文字”;  
A.innerHTML=“書き換えられる文字”;
```

は同じプログラムです。

下のほうのプログラムの書き方になれるようにしましょう。



練習問題3

- テキストボックスを二つ表示、ボタンを押すとそのテキストボックスに入力されている整数の合計をすぐボタンの下に表示するようなプログラムを作成せよ。
- これを解くにあたって～
テキストボックスから読み込んだものは全部数字であろうと文字列として読み込まれてしまい、そのままでは計算できません。そのため括弧内の文字列が数字であった場合整数値として返し、そうでない場合はNaNで返す関数

```
Number( );
```

を使うようにしてください。



解答

```
<html>
<head>
  <title>加算</title>
  <script>
    function add(){
      var i = 0;
      var x = Number(document.form_s.in_x.value);
      var y = Number(document.form_s.in_y.value);
      var z = x + y;
      var r_show = document.getElementById("show_result");
      r_show.innerHTML = x+" "+y+"="+z;
    }
  </script>
</head>
<body>
  <form name="form_s">
    X:<input type="text" name="in_x" value="50"><br>
    Y:<input type="text" name="in_y" value="50"><br>
    <input type="button" value="合計" onclick="add()">
  </form>
  <p id="show_result">ここに結果が出ます。</p>
</body>
</html>
```



ちょっとおまけ ~日付と時間の取得~

- **New Date()**
現在の日付。時刻を取得
例) `var a = New Date();`
変数aに現在の日付&時刻が取得された
- 日付.**getFullYear()**
4桁の西暦年を取得
- 日付.**getMonth()**
月を取得。値は実際よりもひとつ少なくなる
- 日付.**getDay()**
曜日を取得。日曜が0で月曜が1で~という感じ
- 日付.**getDate()**
日にちを取得。



- 時刻.getHours();
時間を取得。
- 時刻.getMinutes();
分を取得。
- 時刻.getSeconds()
秒を取得。

- 例)

```
var a = new Date;  
document.write(  
    a.getHours+"<br>" + a.getMinutes());
```

→時間と分が出力される。

ちなみに取得できる日付時刻などはPCで設定されているもの。



練習問題4

- ページを開くとリアルタイムで現在の日付や時刻が表示されるようにせよ。
- この問題を解くにあたって一定時間で関数を繰り返し実行する
`setInterval(“関数名”, 間隔(ミリ秒));`
を使用してください。
- また、`body`タグ内に
`<body onload=“関数名”>`
とするすことにより、ページを開いた際にこの関数を実行します。



```
<html>
<head>
  <title>現在時刻</title>
  <script>
    function auto_time(){
      setInterval("get_time()",100);
    }
    function get_time(){
      var r_show =
document.getElementById("show_result");
      var now_time = new Date();
      var hour = now_time.getHours();
      var min = now_time.getMinutes();
      var sec = now_time.getSeconds();
      r_show.innerHTML = "現在時刻:"+hour+"時
"+min+"分"+sec+"秒";
    }
  </script>
</head>
<body onload="auto_time()">
  <p id="show_result"></p>
</body>
</html>
```



お疲れ様でした！JavaScriptの基本についてはこれに
終わりです。基本といってもまだまだいろいろあるので興
味がわいたら各自で調べてみてください。

明日はまたこの教室にてVitual basicについてを行います。

時間帯は2時間早まっているので注意してください。

