最終更新　2023.2.3

# 基本技

## ボタン

　まずはボタンを作ってみましょう。次のhtmlファイルをエディタで作って、漢字コードをutf-8に設定してセーブしてください。

<!DOCTYPE html>

<html lang="ja">

<head>

<meta charset="utf-8">

<title>ボタン</title>

</head>

<body>

<p>あいうえお</p>

<p><input type="button" value="押す"></p>

</body>

</html>

　htmlファイルは <!DOCTYPE html> で始めます。<html> のタグの中にヘッダとボディがあります。<title> はタブに表示される文字列を指定します。<body> の部分の内容がブラウザに表示されます。

　実行してみると、「あいうえお」と書かれ、その下に "押す" と書かれたボタンが出現しました。value = の後に書いた文字列がボタンの上に表示されます。ボタンを押しても何も起こりません。ボタンを押したら「押した」と表示するようにプログラムを改造しましょう。

<!DOCTYPE html>

<html lang="ja">

<head>

<meta charset="utf-8">

<title>ボタン</title>

<script>

"use strict";

function btn\_click(){

alert("押した");

}

</script>

</head>

<body>

<p>あいうえお</p>

<p><input type="button" value="押す" onclick="btn\_click()"></p>

</body>

</html>

　ボタンを押したときに呼ぶ関数を onclick="関数名()" という形で指定します。

　ここではヘッダの中の <script> と </script> で囲んだ部分にスクリプト（プログラム）を記述しています。<script> はヘッダに配置してもボディに配置してもどちらでも構いません。違いについては後の節で説明します。

　"use strict"; を1行目に書いて、このプログラムは変数宣言を強要することを宣言しています。ただしこのプログラムは変数を使わないので、"use strict"; を省略しても構いません。

　ボタンを押すと、btn\_click() という関数が呼ばれます。関数は以下の形で記述します。

function 関数名(){

}

　alert("文字列"); を実行すると、指定された文字列が書かれたダイアログが開きます。

## オブジェクトの取得

　次に、ボタンを押したら、「あいうえお」の文字列が「かきくけこ」に変化するよう、プログラムを変更しましょう。

<!DOCTYPE html>

<html lang="ja">

<head>

<meta charset="utf-8">

<title>ボタンと段落</title>

<script>

"use strict"; // 変数の宣言を強要する

window.onload = init; // ページが全て読み込まれた後に実行

let p1;

// window.onload = で定義した初期化関数　ページ読み込み後に実行する

function init(){

p1 = document.getElementById("paragraph-1");

}

function btn\_click(){

p1.innerHTML = "かきくけこ";

}

</script>

</head>

<body>

<p id="paragraph-1">あいうえお</p>

<p>

<input type="button" value="押す" onclick="btn\_click()">

</p>

</body>

</html>

　windows.onload = の後にこのページが全て読み込まれた直後に実行する関数名を書きます。ここではinit() としていますが、どんな名前でも構いません。

　関数init() の中にプログラムの最初に実行すべき処理を記述します。ここでは、「あいうえお」と書いてある段落のオブジェクトを取得して、変数p1に代入します。

　変数p1を使うには、宣言が必要です。宣言の形式は以下の通りです。複数の変数を宣言するときは , で区切ります。

let 変数名;

let 変数名1, 変数名2, 変数名3;

　id名と変数名に同じ名前を付けてはいけません。筆者は衝突を避けるため、id名にはハイフンを含む名前を付けることにしています。変数名にハイフンは使えないので（アンダースコアは使える）、変数名に同じ名前をつけることを回避できます。

　段落の中の文字を書き換えたいときは以下のように書きます。

p1.innerHTML = "書き換え後の文字列";

　間違ってinnerHtmlと書くと、何も起こらないので注意してください。innerHtmlというプロパティを追加し、値を設定します。JavaScriptは大変デバッグがしづらい言語ですが、その一つがこの「存在しないプロパティを使ってもエラーにならない」です。

### プログラムの実行とデバッグの方法

　プログラムをエディタで作成したら、拡張子をhtmlとしてファイルに保存します。

　そのhtmlファイルをブラウザにドラッグします。

　プログラムが動かないときはF12を押します。ここからはChromeにおける操作法を述べます。Consoleタブをクリックしてください。エラーが発生してプログラムが実行できない場合、Consoleにエラーメッセージが表示されており、行番号を含む下線付きの文字列をクリックすると、プログラムの該当部分にジャンプします。

　前の例のように、Alertを使うと「ダイアログボックス」の中に記述することができます。

### スクリプトの配置場所

　ここではwindow.onloadを使う方法を提示しました。<script> ～ </script> をbodyの末尾に配置する方法もあります。以下のように書きます。

<!DOCTYPE html>

<html lang="ja">

<head>

<meta charset="utf-8">

<title>ボタンと段落</title>

</head>

<body>

<p id="paragraph-1">あいうえお</p>

<p>

<input type="button" value="押す" onclick="btn\_click()">

</p>

<script>

"use strict"; // 変数の宣言を強要する

let p1;

p1 = document.getElementById("paragraph-1");

function btn\_click(){

p1.innerHTML = "かきくけこ";

}

</script>

</body>

</html>

　スクリプトをbodyの末尾に記述すると、その直前までに記述された要素を利用可能です。<script> のメインプログラムの部分（function以外の場所）に、最初に実行したい処理を書いておくと、window.onloadで初期化用の関数を指定するのと、ほぼ同等の効果が得られます。

　一般的にはbodyの末尾に記述することが推奨されるようです。筆者は、ファイルの末尾にプログラムを置くより、ファイル先頭に置く方がエディタで編集するときに扱いやすいように感じるので、window.onloadを使ってヘッダに初期化用プログラムを書いています。

課題1

　bodyの色を変えるには以下のようにします。

<body id="body-1">

body = document.getElementById("body-1");

body.style.backgroundColor = "#ffc0c0";

　ボタンを2つ作り、1つめのボタンを押したらbodyの色を "#ffc0c0"、もう一つのボタンを押したら "#c0ffc0" にするページを作りなさい。

（発展）

　ボタンにイベントリスナを設定する方法として、以下ようにイベントリスナを登録する方法もあります。以下のようにボタンにもIDを付けます。

<input type="button" value="押す" id="button-1">

　この方法を用いる場合、init() の外側で

let b1;

でボタンオブジェクトを入れる変数を用意し、init() の中に以下のように記述します。

b1 = getElementById("button-1");

b1.addEventListener("click", btn\_click());

## 引数付きの関数を使う

　2つのボタンがあり、片方のボタンを押すと画面が薄黄色、もう片方のボタンを押すと画面が薄ピンクになるプログラムを作ってみます。

　ボタン毎に関数を作ると以下のようになります。

<!DOCTYPE html>

<html lang="ja">

<head>

<meta charset="utf-8">

<title>button and text</title>

<script>

"use strict"; // 変数の宣言を強要する

window.onload = init; // ページが読み込まれたときに実行する関数

let btn1;

// window.onload = で定義した初期化関数　ページ読み込み時に実行する

function init(){

btn1 = document.getElementById("body-1")

}

function change\_color1(){

btn1.bgColor = "#ffff00";

}

function change\_color2(){

btn1.bgColor = "#00ffff";

}

</script>

</head>

<body id="body-1">

<p>

<input type="button" value="押す1" onclick="change\_color1()">

<input type="button" value="押す2" onclick="change\_color2()">

</p>

</body>

</html>

　ここではボタン1個につき1個の関数を用意し、合計2個の関数を書きました。change\_color("#ffc0c0") のように、引数付きで呼び出せたら、関数は1個ですむので便利です。以下のように書くことができます。

function change\_color(color){

btn.bgColor = color;

}

</script>

</head>

<body id="body-1">

<p>

<input type="button" value="押す1" onclick="change\_color('#ffff00')">

<input type="button" value="押す2" onclick="change\_color('#00ffff')">

## コンソールの使い方

　Excel VBAはイミディエイトウィンドウというのがあり、Debug.Print "abc" と書くとイミディエイトウィンドウに abc と書くことができました。同様の機能を持つのがconsole.log("abc") です。F12を押して開くデバッグ用ウィンドウのConsoleタブに表示されます。

　Excel VBAでMsgBox("abc") と書くとダイアログウィンドウが新しく開き、その中に abc と表示されました。これと同様のことをするのは、Alert("abc") です。

## テキストボックスの使い方

　テキストボックスの中の数値を処理するプログラム例を以下に示します。2つのテキストボックスに数値を入力し、「計算」ボタンを押すと、最大公約数を計算して表示します。

<!DOCTYPE html>

<html lang="ja">

<head>

<meta charset="utf-8">

<title>button and text</title>

<script>

"use strict"; // 変数の宣言を強要する

window.onload = init; // ページが読み込まれた後に実行する関数

let t1, t2, p1, a, b, gcm;

// 初期化用関数

function init(){

t1 = document.getElementById("text-1");

t2 = document.getElementById("text-2");

p1 = document.getElementById("paragraph-1");

}

function calc\_gcm(){

let tmp, c;

a = parseInt(t1.value);

b = parseInt(t2.value);

if ( b > a ){ // a > b となるよう入れ替える

tmp = a;

a = b;

b = tmp;

}

while(1){

console.log(String(a) + " " + String(b))

c = a % b;

if ( c == 0 ) break;

a = b;

b = c;

}

p1.innerHTML = "最大公約数は " + String(b)

}

</script>

</head>

<body>

<p>最大公約数の計算

<p>

数値1：<input type="text" id="text-1" value="200">

<p>

数値2：<input type="text" id="text-2" value="124">

<p>

<input type="button" value="計算" onclick="calc\_gcm()">

<p id="paragraph-1" style="background-color:#ffd0d0;">

ここに計算結果を表示します。

</p>

</body>

</html>

　テキストボックスは type="text" です。getElementById でテキストボックスオブジェクトを取得したなら、obj.valueで中に入っている文字列を書き換えたり、取得したりすることができます。テキストボックスの中の文字列は "123" のように数字だけだったとしても、文字列です。計算するためにparseIntあるいはparseFloatで数値に直しています。

　parseIntで数値に直すのを怠っても、自動的に型変換が行われます。また、p1.innerHTML = "最大公約数は " + bとしても、自動的な型変換が行われ、プログラムは正常に動作します。JavaScriptの自動型変換は便利な面もありますが、たちの悪いバグを引き起こすことがあります。私はpythonのように厳格に方を区別する方が好きです。

## キーボードとマウスのイベント

<!DOCTYPE html>

<html lang="ja">

<head>

<meta charset="utf-8">

<title>キーボードイベントとマウスイベント</title>

<script>

"use strict";

window.onload = init;

let p1, p2, p3;

function init(){

p1 = document.getElementById("paragraph-1");

p2 = document.getElementById("paragraph-2");

p3 = document.getElementById("paragraph-3");

window.addEventListener("keydown",key\_manage);

window.addEventListener("click",click\_manage);

window.addEventListener("mousemove",move\_manage);

}

function key\_manage(evt){

let code = evt.keyCode;

p1.innerHTML = "keydown keycode = " + String(code);

}

function click\_manage(evt){

let x, y;

x = evt.screenX;

y = evt.screenY;

p2.innerHTML = "click at (x,y): " + x + " " + y;

}

function move\_manage(evt){

let x, y;

x = evt.screenX;

y = evt.screenY;

p3.innerHTML = "move at (x,y): " + x + " " + y;

}

</script>

</head>

<body>

<p>イベントについて学習します。キーボードを押すかマウスを左クリックして下さい。</p>

<p id="paragraph-1">キーボードに関する情報</p>

<p id="paragraph-2">マウスのクリックに関する情報</p>

<p id="paragraph-3">マウスの位置に関する情報</p>

</body>

</html>

　キーボードのイベントとマウスのイベントを取得して、取得した情報を表示するサンプルです。ゲームなどのリアルタイム処理に生かせそうです。