

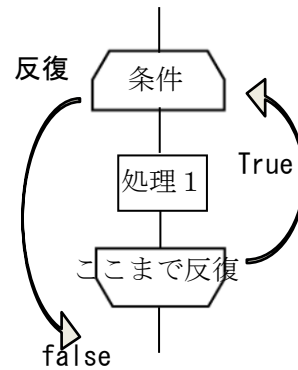
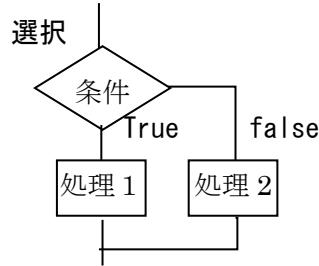
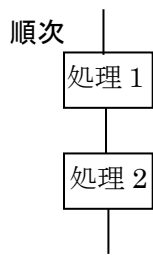
アルゴリズムとプログラミング JavaScript 作法 2

本ドキュメント内リンク INDEX

Jump

- [実践 入力ダイアログ表示 prompt](#)
- [実践 確認ダイアログ表示 confirm](#)
- [実践 アラート表示 alert](#)
- [問題 三角形の面積表示](#)
- [復習 数値型の場合、演算可能](#)
- [復習 文字型の場合、演算不可](#)
- [実践 問題発見 prompt により得られるデータは文字列](#)
- [実践 問題解決 parseInt 関数を使い、文字列を数値に変換](#)
- [実践 問題解決 入力値を 0 が入力されるまで合計するプログラム](#)
- [実践 問題解決 入力値 x, y の大小を表示するプログラムその 1](#)
- [実践 問題解決 入力値 x, y の大小を表示するプログラムその 2](#)
- [復習 Javascript-1 から反復再掲 “false” を 800 回表記](#)
- [実践 prompt で数値を読み、合計値を表示するプログラムを while 文でコーディング](#)
- [実践 prompt で数値を読み、最大値を表示するプログラムを while 文でコーディング](#)
- [実践 prompt で数値を読み、合計値を表示するプログラムを for 文でコーディング](#)
- [配列 並びの値を扱うデータ構造、変数が複数並ぶ](#)
- [実践 配列内値の呼び出し](#)
- [実践 配列内の値を辞書順にソートする \(SORT メソッドを使うシンプルな並べ替え\)](#)
- [実践 配列内の値を昇順にソートするコード](#)
- [実践 配列内の値を降順にソートするコード](#)
- [実践 配列内の値を昇順にソートする別コード](#)
- [実践 配列内の値を降順にソートする別コード](#)
- [実践 配列内の文字を昇順にソートするコード](#)
- [実践 配列内ソート \(直接選択法を使う並べ替え\)](#)
- [実践 配列内サーチ \(indexOf を使うシンプルな検索\)](#)
- [実践 選択処理その他](#)
- [実践 getElementById](#)
- [実践 getElementById 曜日取得と表示](#)
- [実践 getElementById を利用したモンテカルロ法 \(試行回数を入力させて実行\)](#)
- [実践 SWITCH-CASE](#)
- [実践 クイズアプリ](#)
- [実践 マウスに追従する画像](#)

復習



「問題を解く手順」が有限個の処理からなり結果を出して終了するとき、それを**アルゴリズム**という。アルゴリズムをコンピュータに実行させるときに**プログラミング言語**が必要になる。

JavaScript 作法の追加説明

`var x = 5;` 変数 `x` を定義し、値が 5 である
`x == 5;` `x` と 5 と等しい、イコール
`x != 5;` `x` は 5 と等しくない、ノットイコール
`x >= 5;` `x` は 5 より大きいか等しい
`x > 5;` `x` は 5 より大きい
`x <= 5;` `x` は 5 より小さいか等しい
`x < 5;` `x` は 5 より小さい

実践 入力ダイアログ表示 prompt

例 氏名入力 OK ⇒ 氏名、キャンセル ⇒ null

```
<script>
  var name = prompt("enter your name");
  document.write('Hi! ' + name + ' san');
</script>
```



実践 確認ダイアログ表示 confirm

例 入力確認 OK ⇒ true、キャンセル ⇒ false

```
<script>
  var answer = confirm('Is there a mistake?');
  document.write(answer);
</script>
```



実践 アラート表示 alert

```
<script>
  alert('Take care of your health');
</script>
```



問題 prompt で変数 teihen、変数 takasa を入力させて、三角形の面積を alert で表示させよ。

```
<script>
  var x = Number(prompt("input bottom value"));
  var y = Number(prompt("input height value"));
  var s = x*y/2;
  alert("area is " +s);
</script>
```



復習 数値型の場合、演算可能

```
<script>
  var x ;
  var y = 50;
  x = 30 + y;
  document.write(x);の
</script>
```



復習 文字型の場合、演算不可

```
<script>
  var x ;
  var y = "50";
  x = 30 + y;
  document.write(x);
</script>
```



実践 問題発見 promptにより得られるデータは文字列

promptにより得られるデータは文字列であり、演算を実行するときには parseInt 関数を使い、数値に変換する必要がある。

```
<script>

var x;

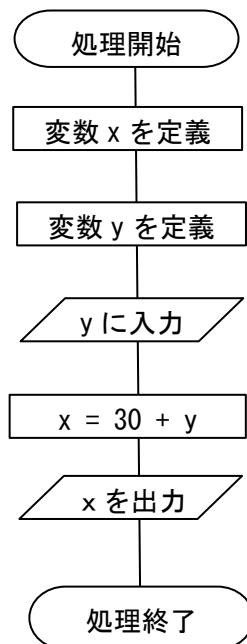
var y;

y = prompt("input number"+"");

x = 30 + y;

document.write(x);

</script>
```



実践 問題解決 parseInt 関数を使い、文字列を数値に変換

```
<script>

var x;

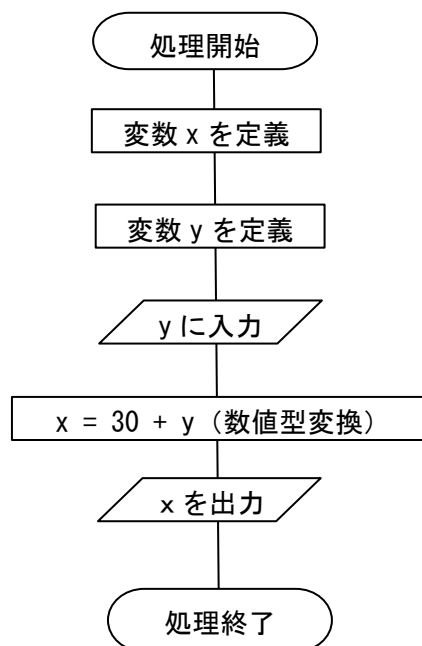
var y;

y = prompt("input number"+"");

x = 30 + parseInt(y);

document.write(x);

</script>
```



実践 問題解決 入力値を 0 が入力されるまで合計するプログラム

<script>

```
var x = 0;
```

```
var y;
```

```
y = Number(prompt("input number" + "0 まで"));
```

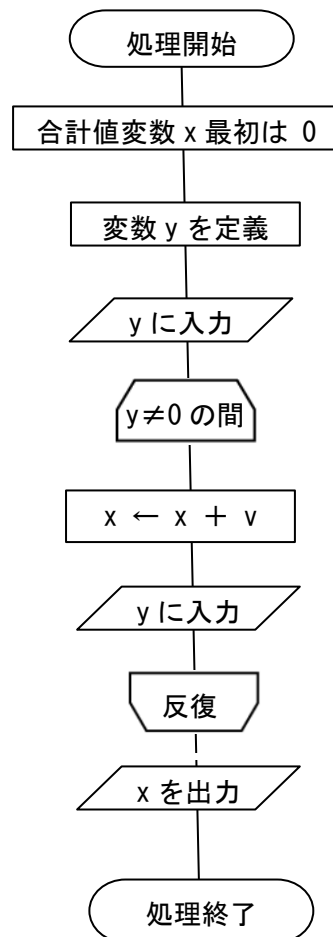
```
while(y !=0) {
```

```
    x = x + parseInt(y);
```

```
    y = Number(prompt("input number" + "0 まで"));
```

```
document.write(x);
```

</script>



実践 問題解決 入力値 x, y の大小を表示するプログラム

<script>

```
var x = Number(prompt('input number'));
```

```
var y = Number(prompt('input number'));
```

```
if( x > y ){
```

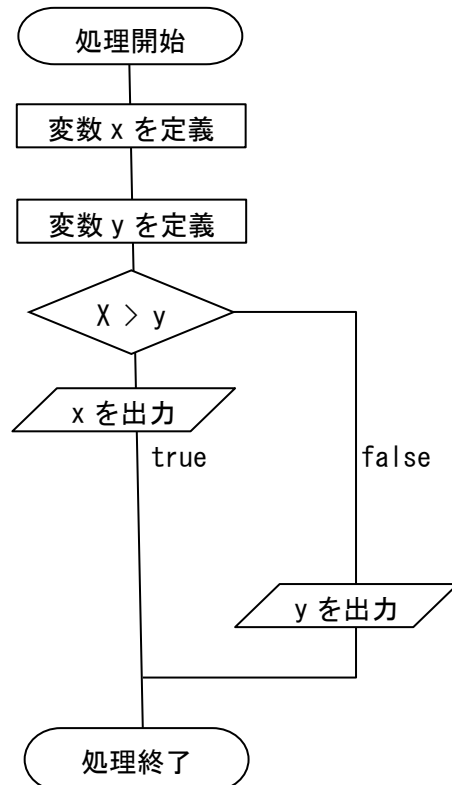
```
    document.write( x + ' is more large than ' + y );
```

```
}else{
```

```
    document.write( y + ' is more large than ' + x );
```

```
}
```

</script>



実践 問題解決 入力値 x, y の大小を表示するプログラム

<script>

```
var x = Number(prompt('input number'));
```

```
var y = Number(prompt('input number'));
```

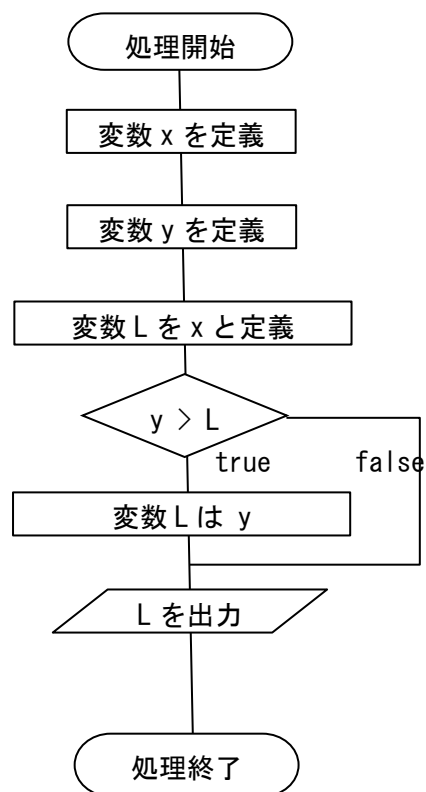
```
var L = x;
```

```
if( y > L ){ L = y };
```

```
document.write( L + ' is more large ' );
```

```
}
```

</script>



JavaScript 作法の説明再掲・追加

問題 Javascript-1 から反復再掲 “false” を 800 回表記するプログラムを for 文でコーディング
「嘘八百」ですな！

```
<script>

    for(var i = 0; i < 800; i++){

        document.write(i + " false ");

    }
</script>
```



実践 問題解決 prompt で数値を読み、合計値を表示するプログラムを while 文でコーディング

```
<script>

var x =0;

var y = Number(prompt("input number until 0"));

while (y !=0) {

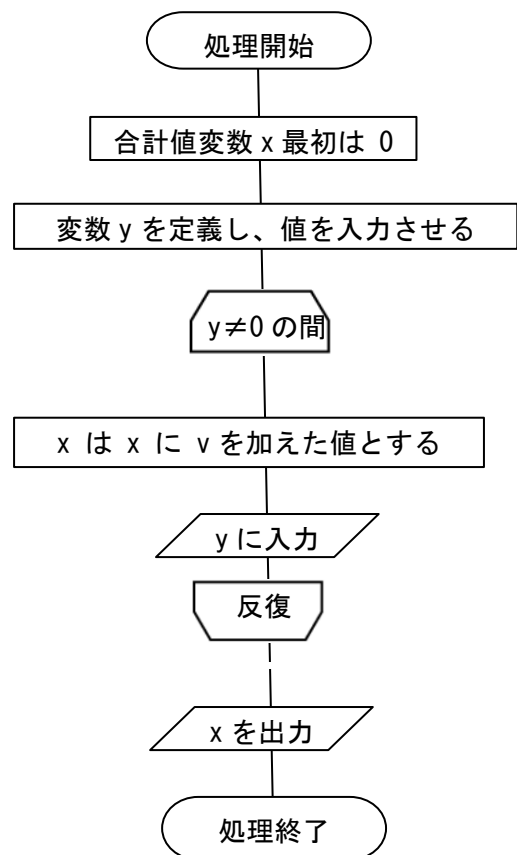
    x = x + y;

    y = Number(prompt("input number until 0"));

}

document.write(x);

</script>
```



実践 問題解決 prompt で数値を読み、最大値を表示するプログラムを while 文でコーディング

<script>

```
var a = Number(prompt("enter value until 0"));
```

```
var max=0;
```

```
while(a !=0) {
```

```
    if(a > max) {
```

```
        max = a;}

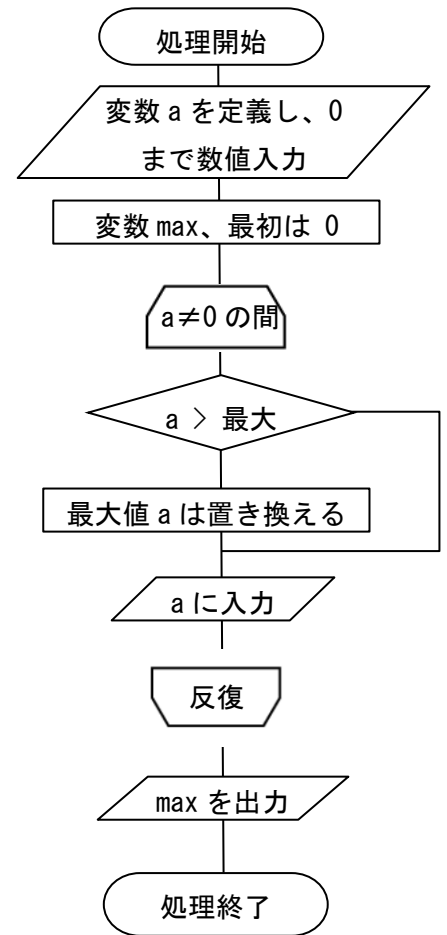
```

```
    a = Number(prompt("enter value until 0"));
```

```
}
```

```
document.write(max);
```

</script>



実践 問題解決 prompt で数値を読み、合計値を表示するプログラムを for 文でコーディング

<script>

```
var a = prompt("enter values by commas").split(",").map(Number);

var sum = 0;

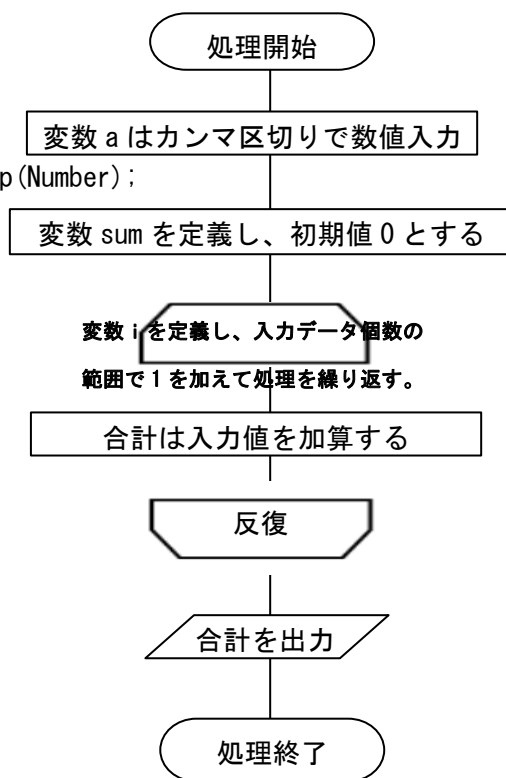
for(var i = 0; i < a.length; i=i+1){

    sum = sum + a[i];

}

document.write(sum);
```

</script>



配列 並びの値を扱うデータ構造、変数が複数並ぶ

変数 x(i)

x(0)	x(1)	x(2)	x(3)	x(4)	x(5)	x(6)	x(7)	x(8)	x(9)
5	1	4	8	0	3	7	6	2	9

← x.length = 10 データ個数 10 →

配列内ソート (SORT メソッドを使うシンプルな並べ替え)

実践 配列内の値を呼び出す

```
<script>
    var array1= [6, 8, 5, 4];
    document.write(array1[1]);
</script><br>
```



実践 配列内の値を辞書順にソートする

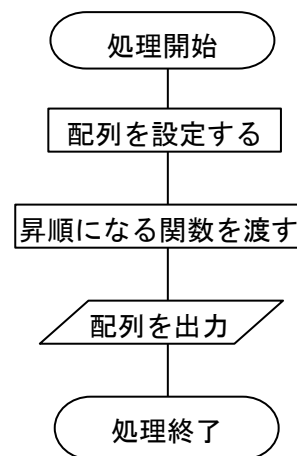
```
<script>  
  let array2 = [7, 1, 8, 12];  
  document.write(array2.sort());  
</script><br>
```

辞書的な文字と解すると 12 の 1 文字目が 1 で 4 の前



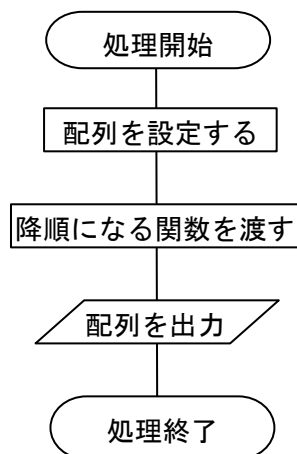
実践 配列内の値を昇順にソートするコード例

```
<script>  
  
  let array3 = [7, 1, 8, 12];  
  
  array3.sort( (a,b) => a-b );  
  
  document.write(array3);  
  
</script><br>
```



実践 配列内の値を降順にソートするコード例

```
<script>  
  
  let array4 = [7, 1, 8, 12];  
  
  array4.sort( (a,b) => b-a );  
  
  document.write(array4);  
  
</script><br>
```



実践 配列内の値を昇順にソートする別コード例

```
<script>
```

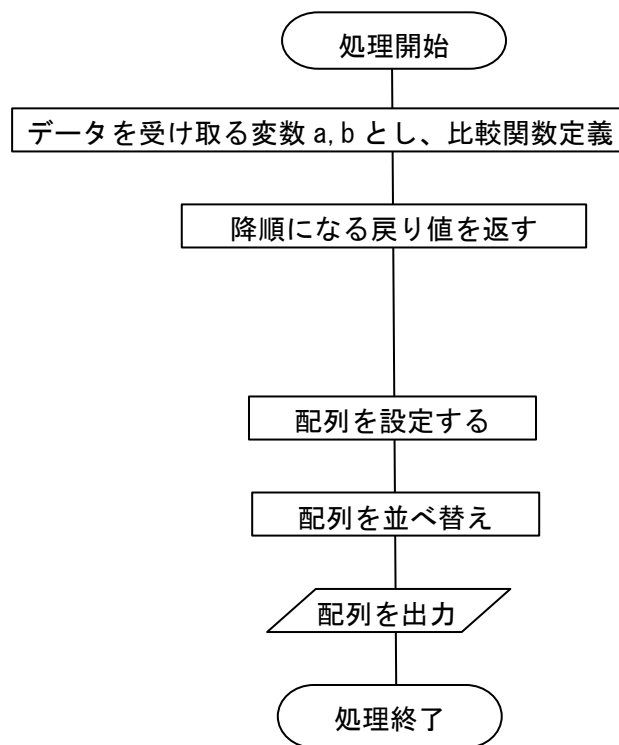
```
function compareFunc(a, b) {  
  return a - b;  
}
```

```
var array5 = [35, 24, 6, 92, 75];
```

```
array5.sort(compareFunc);
```

```
document.write(array5);
```

```
</script><br>
```



実践 配列内の値を降順にソートする別コード例

```
<script>
```

```
function compareFunc(a, b) {  
  return b - a;  
}
```

```
var array5 = [35, 24, 6, 92, 75];
```

```
array5.sort(compareFunc);
```

```
document.write(array5);
```

```
</script><br>
```



実践 配列内の文字を昇順にソートするコード例

```
<script>
```

```
    var str = ['Janet', 'Kenny', 'Ian', 'Cindy'];
```

```
    str.sort();
```

```
    document.write(str);
```

```
</script><br>
```



実践 配列内ソート（直接選択法を使う並べ替え）

x(0)	x(1)	x(2)	x(3)
7	1	8	5
1	7	8	5
1	5	8	7
1	5	7	8

x(0)~x(3)の最小値 x(1) を x(0) と入れ替える

x(1)~x(3)の最小値 x(3) を x(1) と入れ替える

x(2)~x(3)の最小値 x(3) を x(2) と入れ替える

<script>

```
var a= [6, 8, 5, 4];
```

```
var i, j, p, q, min;
```

```
for (i=0; i<a. length-1; i++) {
```

```
    min=a[i];
```

```
    p=i;
```

```
    for (j=i+1; j<a. length; j++) {
```

```
        if (a[j]<min) {
```

```
            min=a[j];
```

```
            p=j;
```

```
        }
```

```
    }
```

```
    q=a[i] ;
```

```
    a[i]=a[p];
```

```
    a[p]=q;
```

```
}
```

```
for (i=0; i<a. length; i++) {
```

```
    document. write(a[i]+",");
```

```
}
```

```
</script><br>
```



実践 配列内サーチ (indexOf を使うシンプルな検索)

```
<script>
```

```
    var array1= [6, 8, 5, 4];  
    var result = array1.indexOf(6);  
    document.write(result);
```

```
</script><br>
```

```
<script>
```

```
    let array2 = ["drink", "road", "icon", "new", "keep"];  
    var result = array2.indexOf("new");  
    document.write(result);
```

```
</script><br>
```

```
<script>
```

```
    let array3 = [7, 1, 8, 12];  
    var result = array3.some( function( value ) {  
        return value === 8;  
    });  
    document.write(result);
```

```
</script><br>
```

```
<script>
```

```
    let array4 = [7, 1, 8, 12];  
    var result = array4.some( function( value ) {  
        return value === 5;  
    });  
    document.write(result);
```

```
</script><br>
```

```
<script>
```

```
    let array5 = [7, 1, 8, 12];  
    var result = array5.some( function( value ) {  
        return value > 6;}  
    );  
    document.write(result);
```

```
</script><br>
```

```
<script>
```

```
    let array6 = [7, 1, 8, 12];  
    var result = array6.every( function( value ) {  
        return value > 6;}  
    );
```

```
document.write(result);  
</script><br>
```



その他の活用 以下は、氷山の一角である。何が可能であるか興味を持つ方はトライしてください。実は…あとは授業で…

実践 選択処理その他

```
<script>  
var choice = prompt("1 or 2 or 3");  
    if(choice==1) {  
document.write("Japanese")  
    }  
    else if(choice==2) {  
document.write("English")  
    }  
    else if(choice==3) {  
document.write("Hypertext markup language ")  
    }  
</script>
```



実践 getElementById

`getElementById` とは HTML タグで指定した ID (HTML タグに設定できる属性・・・HTML の項目で紹介した `bgcolor` や `href` など) にマッチするドキュメント要素を取得するメソッド

引数として ID である String (文字) オブジェクトを要し、戻り値は取得した要素。

実践 getElementById による曜日取得と表示

```
<body onload="X();">
```

```
    Now, It is<br>
```

```
<div id="now"></div>
```

```
<script>
```

```
    var now = new Date();
```

```
    function X() {
```

```
        var target = document.getElementById("now");
```

```
        var Y = now.getFullYear();
```

```
        var M = now.getMonth()+1;
```

```
        var D = now.getDate();
```

```
        var H = now.getHours();
```

```
        var Min = now.getMinutes();
```

```
        var S = now.getSeconds();
```

```
        target.innerHTML = Y + "," + M + "," + D + "," + H + ":" + Min + ":" + S;
```

```
    }
```

```
</script>
```

```
<div>
```

```
    <input id="Button1" type="button" value="repeat" onclick="Button_Click();" />
```

```
</div>
```

```
</body>
```



実践 getElementById を利用したモンテカルロ法（試行回数を入力させて実行）

```
<Script>
    var m=Number(prompt("enter value for trial"));

    p=0;
    for (i=1;i<=m;i++){
        x=Math.random();
        y=Math.random();
        if(x*x+y*y<1 ){
            p++;
        }
    }
    pi=p*4/m;
    document.write("π=",pi);
</Script>
```



実践 SWITCH-CASE

Switch-Case は複数の選択を可能とするが、大小は判定できない。ちなみに HTML 中の span タグは要素のグループ化を行う。

JavaScript **DOM** を使う ⇒ 内容を取得・書き換え HTML

Document Object Model=HTML にプログラムからアクセスするためのオブジェクト

下の実践例では `` 「day」で要素を括る。id をつけた要素は、HTML 全体を保持する document オブジェクトにある指定 id の要素を取得する `getElementById()` で参照を取得。

- ・ `window.onload = function () {` 関数を読む
- ・ `var str = "";` 文字を変数「str」とする
- ・ `var day = new Date().getDay();` 曜日の配列を「day」とする
- ・ `⇒0 は日曜日、1 は月曜日、2 は火曜日、3 は水曜日、4 は木曜日、5 は金曜日、6 は土曜日`
- ・ `switch(day) {` 「day」で振り分ける
- ・ `switch` 文は式を評価し、`case` に関連付けられた文を実行する
- ・ `document.getElementById("day").textContent = str;` 「day」で取得した文字を「str」に

以下実践

```
Today, It is <span id="day"></span>
<script>
window.onload = function () {
var str = "";
var day = new Date().getDay();
switch (day) {
case 0:
str = "Sunday"; break;
case 1:
str = "Monday"; break;
case 2:
str = "Tuesday"; break;
case 3:
str = "Wednesday"; break;
case 4:
str = "Thursday"; break;
case 5:
str = "Friday"; break;
case 6:
str = "Saturday"; break;
}
document.getElementById("day").textContent = str;
}
</script>
```



実践 クイズアプリ

- ・ var question = [] 問題配列
- ・ var answer =[] 正解配列
- ・ var input; 回答定義
- ・ var score = 0 ; 得点初期値を 0 と定義
- ・ var i=0 ; i<question.length ; i++ 出題順を 0 からとし、問題個数まで順に反復
- ・ input = confirm(question[i]); i 番のユーザ回答 (true または false) の確認
- ・ if(score==20 || score==30) {document.write("good");} score が 20 または 30 ならば
- ・ if(X || Y) X または Y ならば
- ・ if(X && Y) X かつ Y ならば

<script>

```
var question = [
  "本校の海拔は 9m である？",
  "本校の創立は 1932 年である？",
  "本校のプールからスカイツリーと東京タワー両方が見える？",
  "橋の木は本校のキャンパス通りにある？",
  "神奈川県政令指定都市の数は 2 つである？",
  "日本の最南端は東京都である？",
  "千葉県は、標高が最も高い場所でも 500m 以下である？",
  "埼玉県に隣接している都県は 6 つである？",
  "C の和音に 7 度 9 度 11 度 13 度の音を付加すると D の和音になる？",
  "我が国最南端の氷河地形カールは北岳にある？"
];
var answer = [ true , false , true , false ,false , true , true , false ,
true,false];
var input;
var score = 0 ;
for( var i=0 ; i<question.length ; i++ ) {
input = confirm( question[i] );
alert( "your answer is " + input );

if( input == answer[i] ) {
alert( "Well, That's correct!" );
score += 10;
}else{
alert( "Uuum, Sorry, You've missed" );
}
}
document.write ( score + " point" );
```

```
if(score==0 || score==10) {document.write("poor");}
if(score==20 || score==30) {document.write("good");}
if(score==40 || score==50) {document.write("very good");}
if(score==60 || score==70) {document.write("excellent");}
if(score==80 || score==90 ) {document.write("Wow!");}
if(score==100 ) {document.write("Fabulous!! You've got it! You're Genius!");}
</script>
</script>
```



SAMPLE

実践 マウス操作に追従する画像

簡単な GIF 画像などを html と同じ階層にファイル名 x.gif として保存しておくこと

```
<body onMouseMove="move(event)">
  <script>
    function move(event) {
      var obj=document.getElementById("img1");
      obj.style.left=event.clientX;
      obj.style.top=event.clientY;
    }
  </script>
  
</body>
```



SAMPLE