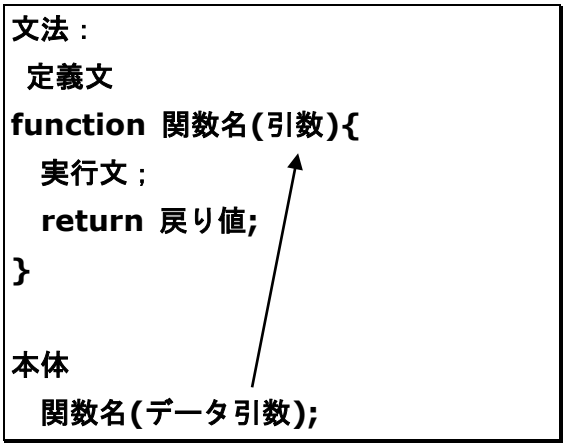
4.Function 関数 【PR32】

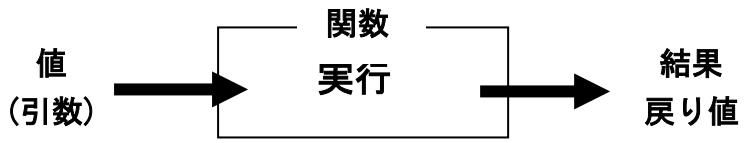
プログラミングでは「ある処理をするひとかたまりのもの」に名前を付けたものを「**関数**」と呼ぶ。本来は、function を訳した「機能」が正しいのだろうが、なぜか、誤訳のまま通ってしまっている。

JavaScript では、関数を自分で定義して使うことができる。関数の中には、数学的なイメージがわくだろうけど、文字列などを処理することもできる。

関数の定義は、<HEAD>の部分で定義され、＜BODY＞の中で実行する。



関数の処理のためプログラム本体からデータを引き渡す場合がある。この引き渡されたデータを引数（ひきすう）と呼ぶ。Function で処理された結果をプログラム本体に返すものもある。この際には return が使われる。返される値を「戻り値」と呼ぶ。何も返さない return もある。【PR17】



JavaScript の関数は、同じ名前で定義してもエラーにならない。あとから定義された名前の方が、有効な関数名になる。

プログラム１：ドルを円に換算する関数。

```
<Script>
 function exchange(doll, rate){   // ドルを円に換算する関数の定義
  return doll*rate;
 }
</SCRIPT>
</HEAD>
<BODY>
<Script>
var jRate=112;   //   レートはここに入れる
var oneDoll=280;   // ドルの値をここに入れる
 document.write(oneDoll,"ドルは、",exchange(oneDoll,jRate),"円です。<BR>");
</SCRIPT>
```

関数の定義では、Python が「def」で表示し、JavaScript は「function」で表示します。
使い方などはどちらも同じですが、一般的な書き方が 2 つの言語では異なります。

＜Python＞
・「タブ」で構造を示す
・行末の記号は必要ない

＜JavaScript＞
・{　}（波括弧）で構造を示す
・各行末には；（セミコロン）が必要

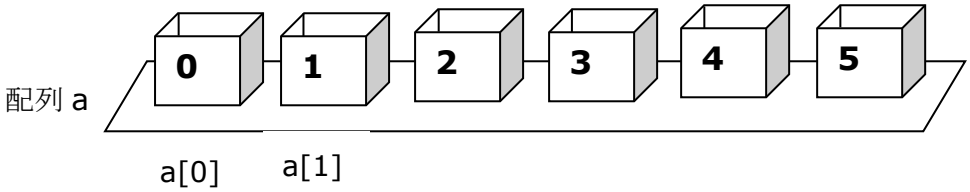
5.Array 配列 【PR86】

配列は、変数をひと塊としてあつかったもの。
たとえば、「今月の売り上げ」、「クラスの成績」など、日常的に配列はよく使われています。

配列は、連続して用意された「番号が書かれたの箱」のようなイメージ。その箱の、番号を指定することで、中に入っている要素を取り出したり、しまったりできる。

文法：
配列名[添字]

例：
a[i]



サンプル：0 から 4 までの５つの箱を用意し、それぞれの箱の中に、同じく 0 から４までの数字が入れられる。

```
for(i=0; i<5; i++)
 abc[i]=i;
```

この場合の配列の名前は「abc」。
ふつう、配列の先頭は「0」から始まる。

他の言語と比較すれば、JavaScript の「配列」の書法は、非常にゆるやかである。

5-1.配列の宣言

配列を使うためには、配列を「宣言」しなくてはならない。

配列の宣言は new でおこなう。*

文法：
配列名＝new Array(配列数)；

たとえば、

var abc=new Array(5); とすれば、

配列名： abc

配列数： 6 個の配列の要素（箱）が作られる。

（※配列は０から始まるので６個になる）

5-2. 配列への直接代入

配列に、要素を直接代入する方法。

alph＝new Array（"A"、"B"、"C"）；

プログラム１：「文字の色を配列に貯えて、表示する」

```
<Script>
   var col=new Array(6);  // 配列 col を作る
   col[1]="0000FF";col[2]="00FF00";col[3]="00FFFF";
                              //配列 col に、要素を代入する。
   col[4]="FF0000";col[5]="FF00FF";col[6]="FFFF00";
   for (i=1; i<=6; i++){
      document.write("<FONT COLOR=#" +col[i]+">");
      document.write("ABCDEFG<BR>");
      document.write("</FONT>");
   }
</SCRIPT>
```

* PR17 参照。new については、日付けオブジェクトで詳しく扱う。

プログラム２：配列を使った「住所録」

```
<HTML>
<HEAD>
<TITLE>配列を使った住所</TITLE>
<Script>
   function Array(n){
      this.length=n;
   }
</SCRIPT>
</HEAD>
<BODY>
<Script>
var cell=new Array(6);
cell[1]="名前";cell[2]="電話";cell[3]="住所";
cell[4]="田中";cell[5]="564-5678";cell[6]="名古屋";
n=1;
document.write("<TABLE BORDER>");
for (i=0; i<2; i++){
   document.write("<TR>");
   for(j=0; j<3; j++){
      document.write("<TD>"+cell[n]+"</TD>");
      n++;
   }
   document.write("</TR>");
}
document.write("</TABLE>");
</SCRIPT>
</BODY>
</HTML>
```

Python の「コレクション」は、一般に「配列」と呼ばれています。
Python にはいくつものコレクション（リスト、辞書、etc）がありますが、JavaScript では「配列」だけです。
配列やコレクションは、メモリ上に連続した番地にデータが格納されます。Python ではそのことを意識する必要があります。

◆リストに対するさまざまな処理

・filter

ある条件を満たしているリストを書き出す

＜Python＞

例：2 以上のリスト（新しいリストが作成される）

```
list(filter(lambda v: v> 2, x))
```

※無名間数は lambda で定義

＜JavaScript＞

```
x.filter(v => v> 2);
```

・map 処理

各要素に特定の処理をして書き出す

＜Python＞

例：リストを 2 倍して書き出す

```
[v * 2 for v in x]
```

・reduce 処理

リストに処理をし、ひとつの結果を出す

例：リストから最大、最初、合計などを選ぶ

・要素の連結

＜Python＞

```
“-“ .join([“a”, “b”, “c”])
```

＜JavaScript＞

```
[“a”, “b”, “c”].join(“-”);
```

・ソート

7. document object ドキュメント・オブジェクト

文字の表示などで、これまでに、馴染みの多い document オブジェクト。
このオブジェクトは、ブラウザの HTML についての管理をするもの。

document のメソッドとプロパティを示すと
メソッド：
clear、close、open、write、writeln など。
プロパティ：
alinkColor、anchors、bgColor、cookie、fgColor、forms、lastModified、linkColor、links、location、referrer、title、vlinkcolor など。

7-1. 文字色・背景色

まず、はじめに示す簡単な例として「**文字色・背景色**」を説明する。
これらのプロパティは、document オブジェクトに所属するので、

```
document.fgColor=”red”;
document.bgColor=”blue”;
```

のように使われる。
これらは、＜BODY＞タグの BGCOLOR や TEXT と同じ結果が得られる。

これらの色の指定には、RGB の 256 色を 16 進法で表したものが、もっとも、確実である。

```
＃RRGGBB
```

１０進数の「２５５」は、１６進数では「ｆｆ」と表される。
それぞれの色成分の「RR」「GG」「BB」には、この１６進数が入る。
この表現のためには、16 進で表現していることを示すために「#」が文字列の先頭に必要になる。

上と同じ色をこの方法で表現すると、

```
document.fgColor=”#ff0000”;
document.bgColor=”#0000ff”;
```

7-2. HTML ドキュメントの管理

document オブジェクトは、HTML ブラウザを管理するものである。
どのためのプロパティもある。

- **title**
  ＜TITLE＞タグに囲まれたもの
- **lastModified**
  最終更新日
- **referrer**
  ドキュメントを呼び出したページの URL
- **URL**
  ドキュメントの URL

プログラム１：「背景が、黒から白へ徐々に変化していく」

```
<HTML>
<HEAD>
<TITLE> </TITLE>
<Script>
   function dispbk(t){
      for (i=0; i<=255; i++){
         for (j=0; j<t; j++)
            document.bgColor=i+i*256+i*256*256;
         }
      }
 </SCRIPT>
</HEAD>
<BODY>
<H1>黒から白へ変化します</H1>
<Script>
   dispbk(5);
</SCRIPT>
</BODY>
</HTML>
```