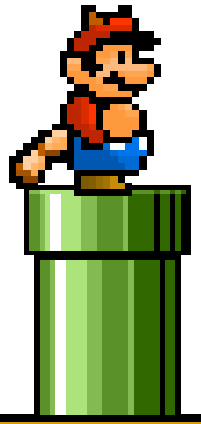
JavaScript
プログラミング入門

2-5　配列

2-5-2 配列を生成するArrayコンストラクタ
2-5-3 ユーザ定義コンストラクタによる配列オブジェクトの作成
2-5-4 配列の並び替え
2-5-5 Arrayオブジェクトのメソッド



2006年10月20日（金）　南　慶典

2-5 配列

配列を生成するArrayコンストラクタ

```
var dayOfTheWeek = new Array();
```

```
dayOfTheWeek[0] = "Sunday";
dayOfTheWeek[1] = "Monday";
dayOfTheWeek[2] = "Tuesday";
dayOfTheWeek[3] = "Wednesday";
dayOfTheWeek[4] = "Thursday";
dayOfTheWeek[5] = "Friday";
dayOfTheWeek[6] = "Saturday";
```

なお、Arrayコンストラクタには、次のようにして配列に格納する値を直接渡すこともできる

```
var name = new Array("山田","田中","江藤");
```

要素数は可変

```
var Name = new Array(2);
Name[0] = "山田";
Name[1] = "田中";
Name[2] = "江藤";
```

- これもOK。自動的に要素数が増やされる

```
var Name = new Array();
Name[0] = "山田";
Name[1] = "田中";
Name[2] = "江藤";
```

- 要素数を指定しなくてもよい。
  - 必要になった時に自動的に増やされる。

中身もこだわらない

文字列と数値を別々に格納することもできる

```
var customer = new Array(2);
customer[0] = "山田";
customer[1] = 1001;
```

配列リテラル

var name = new Array("山田","田中","江藤");

これを配列リテラルで記述すると、

```
var name = new Array["山田","田中","江藤"];
```

配列の長さを示すプロパティ

Arrayオブジェクトには、その時点の要素数を示すlengthプロパティが用意されている。

```
// 配列のすべての要素を出力する
for(var i=0;i<dayOfTheWeek.length;i++){
    document.write(dayOfTheWeek[i],"<br>");
}
for(var i=0;i<dayOfTheWeek.length;i++){
    document.write(dayOfTheWeek[i],"<br>");
}
```

このとき、配列の添え字は「0」から始まり、最後の要素は「length-1」であることに注意!!

ユーザ定義コンストラクタによる配列オブジェクトの作成

Arrayコンストラクタの代わりにユーザが独自にコンストラクタを定義して、配列オブジェクトを作成することもできる。これは、あらかじめ配列の要素に値を入れておきたいといった場合に便利。

たとえば、すべての要素の初期値が「100」の配列を生成するためのコンストラクタは、functionキーワードを使用して次のように記述する。

```
Function MakeArray(len){
    this.length=len;
    for(i=0;i<this.length;i++){
        this[i]=100;
    }
}
```

var myArray = new MakeArray(5);

- これで要素数が「5」の、値が「100」の配列が生成される

配列の並び替え

配列をソートする（並び替える）にはsort()メソッドを使用する。

メソッド：sort( )
構文： Arrayオブジェクト名.sort(compareFunction)
引数： compareFunction・・・比較関数への参照

たとえば、

```
var numArray = new Array("5", 50, 9, 10, 5);
numArray.sort();
```

結果は、

10, 5, 5, 50, 9

比較関数を使用した並び替え

- 配列の中の2つの要素を比較するために使用される。
- 比較関数は2つの引数を比べて「小さい」か「等しい」か「大きい」かによって、それぞれ負、0、正の値を返すものでなければならない。

```
// 比較関数 compareの定義（昇順）
function compare(a,b){
    return a-b;
}
```

- 使用例

```
var numArray = new Array("5", 50, 9, 10, 5);
numArray.sort(compare);
```

処理結果＞＞＞
5, 5, 9, 10, 50

ちなみに降順にするには「return b-a;」にすればよい。

Arrayオブジェクトのメソッド（1）

- Arrayオブジェクトに用意されている、その他のメソッド

メソッド名	説明
toString()	配列の要素をカンマ「，」でつなげた文字列で戻す
join()	配列の要素をつなげた文字列を戻す
reverse()	配列の要素を逆順にする
concat()	配列に要素を追加して新たな配列を生成する
slice()	配列の要素の一部を取り出した新たな配列を生成する
push()	配列の最後に要素を追加する
pop()	配列の最後の要素を取り出す
shift()	配列の最初に要素を追加する
unshift	配列の最初の要素を取り出す

Arrayオブジェクトのメソッド（2）

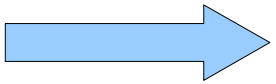
- toString()メソッド

```
var SeasonStr;
var season = new Array("春","夏","秋","冬");
SeasonStr=season.toString();
```

「春、夏、秋、冬」

- join()メソッド

```
var SeasonStr;
var season = new Array("春","夏","秋","冬");
SeasonStr=season.join();
```



「春夏秋冬」

Arrayオブジェクトのメソッド（3）

- reverse()メソッド

```
var season = new Array("春","夏","秋","冬");
season.reverse();
for(var i=0;i<season.length;i++){
    document.write(season[i],"<BR>");
}
```

冬
秋
夏
春

宿　題

問題1
年月日を表示しなさい。ただし、曜日と月は配列を使うこと。
年数と日は関数を使うこと。