

◇Stringの各種メソッド

リスト1

```
1 String.charAt ( X );
2 //X文字目の取得
3 String.charCodeAt ( X );
4 //X文字目の文字コード (Unicode値) の取得
5 String.concat ( A , B );
6 //文字列AとBを連結した文字列を返す
7 String.indexOf ( Str [, index ] );
8 /*
9 文字列Strが文字列中の最初にある位置を返す
10 なお、文字列中のindex (省略時0) 文字目から検索し、
11 Strが見つからない場合は-1を返す
12 */
13 String.lastIndexOf ( Str [, index ] );
14 /*
15 文字列Strが文字列中の最後にある位置を返す
16 なお、文字列中のindex (省略時0) 文字目から検索し、
17 Strが見つからない場合は-1を返す
18 */
19 String.match ( Reg );
20 /*
21 正規表現Regにマッチする値を返す
22 gフラグがあればマッチした部分の配列が返される
23 */
24 String.replace ( Reg , Text );
25 /*
26 文字列中の正規表現Regにマッチする部分を
27 文字列Textのフォーマットにしたがって置換する
28 なお、JS1.3よりTextの代わりに関数を使って柔軟な処理ができるようになった
29 くわしくは
30 http://developer.mozilla.org/ja/docs/Core\_JavaScript\_1.5\_Reference:Global\_Objects:String:replace
31 の"4.2 パラメータとしての関数の指定"を参照のこと
32 */
33 String.search ( Reg );
34 //正規表現Regにマッチする部分が文字列のどの位置にあるか値を返す
35 String.slice ( start [, end ] );
36 //第6回資料のArray.sliceの文字列版で挙動は同じ
37 String.split ( Sep [, lim ] );
38 /*
39 文字列をセパレータSep (文字列or正規表現) で分割した配列を返す
40 limが指定されていれば結果の配列はlim個以下の長さとなる
41 */
42 String.substring ( start [, end ] );
43 /*
44 String.sliceと似ているがstartやendに負数は使えない
45 start>endの場合、startとendは交換される
46 引数がNaNまたは負の値なら0となる
47 引数が文字の文字長を超える場合は引数=文字長となる
48 */
49 /*
50 String.toLowerCase ();
51 //小文字に変換した文字列を返す
52 String.toUpperCase ();
53 //大文字に変換した文字列を返す
```

<リスト1>はStringオブジェクトのメソッドである

リストのString部分に文字（列）を指定し、それぞれにあった引数を与えることで各種機能を使うことができる

（注：あまり使わない一部メソッドと廃止されたString.substrは省略）

リスト2

```
1
2 String.fromCharCode ( char1 [, char2 [, char3 [...]]]);
3 /*
4 引数を文字コード値として文字列を構築し、返す
5 例)
6 print(String.fromCharCode(74,97,118,97,83,99,114,105,112,116));//JavaScript
7 */
8
```

<リスト2>はStringコンストラクタのプロパティである

ここでのStringは文字列ではなく、Stringコンストラクタをさすのでそのまま表記する

つまり、例のように書くので注意