初めての JavaScript(基本)

http://developer.mozilla.org/en/docs/Core_JavaScript_1.5_Reference

データ型と変数

■スコープ

・var をつけると、関数内で有効。つけないとグローバルとなる。
・var をつけても、ブロックレベルの変数にはならない。

■単純なデータ型

・JavaScript には、以下の 3 つの単純なデータ型しかない

型	内容
string	文字列
numeric	数値
boolean	真偽値

・一方で、String、Number、Boolean という組み込みオブジェクトも存在する。

■ null と undefined

・null は定義がされたが、値が設定されていない状態
・undefined は宣言されたが、初期化されていない状態

■定数

・キーワード const を利用すると、定数になる
・IE7.0 は対応していない？

演算子と文

■演算子

演算子	意味
+	加算
-	減算
*	乗算
/	除算
%	剰余
++	インクリメント
--	デクリメント

■ビット演算子

演算子	意味
&	ビット論理積 (AND)
|	ビット論理和 (OR)
^	ビット排他的論理和 (XOR)
~	ビット否定 (NOT)
<<	左シフト
>>	符号付右シフト
>>>	符号なし右シフト

```
newValue = oldValue << 1; ' 左に 1 ビットシフト
```

■条件文

if、if else

```
if ( 式 ) {
  ・・・
}
```

```
if ( 式 ) {
  ・・・
} else {
  ・・・
}
```

switch

```
switch ( 式 ) {
case ラベル 1:
  ・・・
  break;
case ラベル n:
  ・・・
  break;
default:
  ・・・
}
```

■等値演算子

・「==」等値演算子
・「!=」 不等値演算子

・データ型を変換する ->2 つの変数が数値と文字列の場合、数値を文字列に置き換えてから比較する。

```
alert(("1.0" == 1)); // 結果は true
```

・「===」同値演算子 または 厳密等価演算子
・「!==」厳密不等価演算子

・値とデータ型の両方が等しいときだけ結果が真
・JavaScript1.3 から

```
alert(("1.0" === 1)); // 結果は false
alert((1.0 === 1.0)); // 結果は true
alert(((new Number(1.0)) === (new Number(1.0)))); // 結果は false
var num1 = new Number(1.0);
var num2 = num1;
alert((num1 === num2)); // 結果は true
```

■比較演算子

演算子	内容
>	左が右より大きい
>=	左が右以上
<	左が右より小さい
<=	左が右以下

■三項演算子

・条件 ? 真のときの値 : 偽の時の値 ;

```
var isAdult = (age >= 20)? true : false;
```

■論理演算子

演算子	内容
&&	論理積演算子 両辺が真の時のみ真 左辺が偽の時右辺は評価されない
||	論理和演算子 両辺の何れかが真の場合真 左辺が真の場合右辺は評価されない

■繰り返し

while

・条件が真の間繰り返し

```
while ( 条件 ) {
  ・・・
}
```

do while

・条件が真の間繰り返し
・条件を満たさなくても、1 回処理される

```
do {
  ・・・
} while ( 条件 )
```

for

```
for ( 初期値設定 ; 条件 ; 更新 ) {
  ・・・
}
```

for in

配列の要素を取り出す

```
for ( 変数名 in 配列 ) {
  ・・・
}
```

※ for in を利用してもビルトインのプロパティ (String.indexOf、Object.toString) は取得できないが、ユーザが定義したプロパティは取得される。

→ 配列の要素取得に利用すべきではない！

```
// 例
function test() {
    var s = ['a', 'b'];
    s.foo = "foo";
    s.bar = function() {
        alert("bar");
    };
    var i=0;
    for (p in s) {
        document.write(p + " : " + s[p] + "<br>");
    }
}
// 結果
0 : a
1 : b
foo : foo
bar : function() { alert("bar"); }
```

document のプロパティを列挙

```
var win = window.open("","_blank");
for (elm in document) {
  win.document.write(elm + "<br>");
}
win.document.close();
```

条件判定にも利用できる (linkColor プロパティが document オブジェクトに存在するか)

```
alert(("linkColor" in document)); // true
```
