

# JavaScript基礎-第2回資料

## ◇算術演算子

### 例1

```
1 print ( 1 + 2 + '¥n' ); //3
2 print ( 1 - 2 + '¥n' ); //-1
3 print ( 1 * 2 + '¥n' ); //2
4 print ( 1 / 2 + '¥n' ); //0.5
5 print ( 1 + 2 * 3 + '¥n' ); //7
6 print ( ( 1 + 2 ) * 3 + '¥n' ); //9
7 print ( 1 % 2 ); //1
```

算術演算子とは数学的計算を行うための命令であり、<例1>は上から順に足す、引く、かける、割る、割った余りとなっている

かけるが\*（アスタリスク）になっているので注意

計算順序は5行目のように、普通の数学と同じで+-より\*/のほうが先である

また、6行目のように、()を使って順序を変更することもできる

なお、第1回で登場した=も左辺に右辺を代入する算術演算子である

### 例2

```
1 var i = 0;
2 print ( ++ i + '¥n' ); //1
3 print ( i + '¥n' ); //1
4 print ( i ++ + '¥n' ); //1
5 print ( i + '¥n' ); //2
6 print ( -- i + '¥n' ); //1
7 print ( i + '¥n' ); //1
8 print ( i -- + '¥n' ); //1
9 print ( i ); //0
```

### 例3

```
1 var i = 0;
2 function A () { // ++ i
3     i = i + 1;
4     return i;
5 }
6 function B () { // i ++
7     var tmp = i;
8     i = i + 1;
9     return tmp;
10 }
11 print ( A () + '¥n' ); //1
12 print ( i + '¥n' ); //1
13 print ( B () + '¥n' ); //1
14 print ( i ); //2
```

++, --はそれぞれインクリメント、デクリメントと呼ばれ、+1、-1の意味がある<例2>

先にインクリメントがあれば+1してから返し、後にインクリメントがあれば値を返してから+1される

デクリメントはインクリメントにおける+1を-1と読み替えるだけである  
インクリメントを例にして、"++対象"と"対象++"をそれぞれ関数として再現してみたのが<例3>である

## ◆switch

### 例4

```
1  var sw = 3 ;
2  switch ( sw ){
3      case 0 :
4          //sw==0の時の処理
5          break ;
6      case 1 :
7          //sw==1の時の処理
8          break ;
9      case 2 :
10         //sw==2の時の処理
11         break ;
12     case 3 :
13         //sw==3の時の処理
14         break ;
15     default :
16         //どれにも一致しなかったときの処理
17 }
```

構文は"switch(比較される変数){比較ブロック}"となり、比較される変数が比較ブロックのcaseに続く定数と等しければ、そのcase以降が実行される<例4>

breakは現在のブロックの属する制御構文を抜け出す命令であり、これがないと、あるcaseに比較される変数が等しいとき、それ以下のcaseの内容が実行されてしまう

なお、defaultはどのcaseにも一致しなかったときに呼び出される

忘れがちだが、比較ブロックの一番下にあるセクション（この場合はdefault）でのbreakは無意味なので普通書かない

### 例5

```
1  var sw = "abc" ;
2  switch ( sw ){
3      case "abc" :
4          print ( "abc¥n" );
5          break ;
6      case "def" :
7          print ( "def¥n" );
8          break ;
9      case "ghi" :
10         print ( "ghi¥n" );
11         break ;
12     case "jkl" :
13     case "mno" :
14         print ( "jkl or mno¥n" );
15         break ;
16     default :
17         print ( "N/A¥n" );
18 }
```

JavaScriptでは<例5>のようにswitchで複数文字を比較対照にできる

また、発展的な書き方として12-13行目のようにしてAまたはBを条件にすることも可能