

3. 制御命令を駆使して

効率的なプログラミングをしちゃおう

「さあ、今日はついに「これぞ！プログラミング」という内容に入っていきますよ。制御命令といってコンピュータの最も特徴的な2つの処理、繰り返しと条件分岐です。」

「繰り返しというのは、1から10までを足せとか、パスワードが入力されるまで入力ボックスを表示せよ、などという場合に使うんですね。」

「また、条件分岐は「もし～だったら～、そうでないならば～」という命令ですから、もし、18才未満だったらこちら、それ以上ならばこちらへ、とか(^); もしブラウザがネスケの4だったらこのページへ、IE4ならばこちらへ、それ以外だったらこっちへ、というふうに使えばいいんですね。な～んとなくゲームとかも作れそうな気になってきたでしょー、でしょー。(^^)

「では、早速、繰り返しから書いていくことにしましょう。」

1. for 命令

```
for ([初期値:] [条件:] [増減式]) {
  【命令】
}
```

初期設定値から繰り返しを実行し、条件が満たされる場合のみ、【命令】実行される。増減は for の先頭に戻るとき。

2. while 命令

```
while (条件) {
  【命令】
}
```

初期設定値から繰り返しを実行するが、もし条件が満たされていない場合は【命令】は実行されずにループ外の命令に処理が移る。条件判断はループ開始時とリピート時。

「さて、それでは【問題1】を見て下さい。」

「文字列を繰り返して表示させていますねー。これは前回までならば同じような行をコピー & ペーストして書いていたところでしょう。そうそう、昔はコピー & ペーストしていけばいいんですけど、ん～、しかも、今はHDのサイズやCPUのスペックも良くなってきたので、たしかに、効率的に書かなくても…、それでも…いいんですけどお。しかし、それではジャバスクリプトを開発した研究者に申し訳が立たないので！（笑）覚えていくことにしましょう。」

「繰り返しには2つの命令をよく使います。」

「for は数値を条件とする場合、while は特定の条件、たとえばパスワードが入力されるまで、など、そういう使い分けをします。ですから、たいていは繰り返しというと、数値を伴いますので、for が一般的ということになるわけです。(^^)」

「では、どういうふうにするのか、見ていきましょう。【問題1】を見て下さい。」

【問題1】「ようこそジャバスク道場へ」と5回表示するプログラムを書きなさい。



```
for(i=1;i<=5;i++){
document.write("ようこそジャバスク道場へ<BR>");
}
```

「この場合は5回繰り返せ、ということですから、解答では一、変数*i*をとりあえず for のためだけに使ってえ、「1から5以下の間は繰り返す、1つつ増やしながらか」という意味のループになるわけですねえ。」

「もちろん、for(i=0;i<5;i++)でもいいんですよ。それでも5回繰り返しますからね。」

「今、以上と未満の記号を私は何気なく使い分けてしまいましたが、これは「演算子」としてプログラムでは非常に重要な記号なんですなえ。それでは他にどういふものがあるかプリントを見て下さい。」

1. 算術演算子

- + 加算
- 減算
- * 乗算
- / 除算
- % 剰余
- ++ 1つ増やす(インクリメント)
- 1つ減らす(デクリメント)

2. 論理演算子

- && 左辺値と右辺値の両方が true ならば true を返します
- || 左辺値と右辺値のどちらかが true ならば true を返します
- ! 値が false なら true を、true なら false を返します

3. 代入演算子

- = 右辺値を左辺へ代入します
- += (例... x += y; は x = x + y;と同じ)
- = (例... x -= y; は x = x - y;と同じ)
- *= (例... x *= y; は x = x * y;と同じ)
- /= (例... x /= y; は x = x / y;と同じ)
- %= (例... x %= y; は x = x % y;と同じ)
- &= (例... x &= y; は x = x & y;と同じ)
- |= (例... x |= y; は x = x | y;と同じ)
- ^= (例... x ^= y; は x = x ^ y;と同じ)

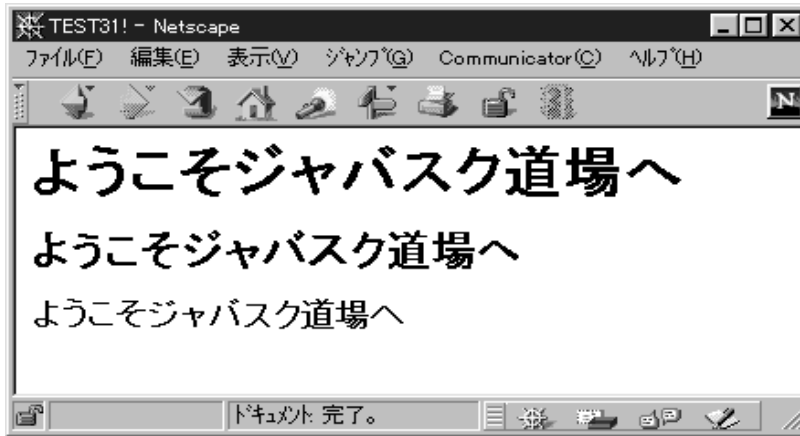
4. 比較演算子

- == 値が等しいかどうかを調べ等しければ true を返す
- != 値が等しくないかどうかを調べ等しければ true を返す
- > 左辺値が右辺値より大きければ true を返します
- >= 左辺値が右辺値以上ならば true を返します
- < 左辺値が右辺値より小さければ true を返します
- <= 左辺値が右辺値以下ならば true を返します

【演算子一覧】

「では、要領がわかったところで【問題2】をやってみてください。」

【問題2】以下のように段々と見出しが小さくなるようなプログラムを作成しなさい。



```
for(i=1;i<4;i++){
document.write("<H"+i+">ようこそジャバスク道場へ</H"+i+">");
}
```

「はい。これは見出しのタグHタグが数値を使って文字のサイズを調節するところを思い出せばOKでしたね。」
 「今日のテーマは繰り返しや条件分岐を使ってプログラムを効率的に組む、ということですから、さあ、もっと for を使っていきますよ。」

【問題3】をやってみて下さい。これも応用でできます。」

【問題3】矢印画像素材の「arrow.gif」を使って、以下のように画像を5コ、少しずつ大きくするようなプログラムを組みなさい。



```
for(i=0;i<5;i++){
var w=20+i*10;
document.write('<IMG SRC="arrow.gif" WIDTH='+w+'>');
}
```

「さて、ついに画像を使う例題になりましたねー。そうなんです、画像もこうやって変数と組み合わせて使うことがで

きるんです。ほーら、なんだかゲーム作りたくなってきただしよー(^_^)

「もちろん、画像ファイルの名前にも変数は使えますので、image1～image9 までを表示するなーんてのも、なんだか出来そうな気になるでしょー。」

「さあ、それでは先に進みましょう。」

【問題4】演算オブジェクトを組み合わせて、1～10までの平方根を以下のように表示させるプログラムを作成しなさい。ただし、while を使うこと。



```
var i=1;
while(i<=10){
document.write(i+"の平方根は"+Math.sqrt(i)+"<BR>");
i++;
}
```

「はい、【問題4】Math オブジェクトに気づくことが重要ですねえ。前回のテーマはリファレンスに慣れる、でしたねー。」

「そして、この問題では while をつかうということでしたので、while は初期値を設定する部分がありませんから、それはプログラマーが自分でそういう行を書かなくてはなりません。」

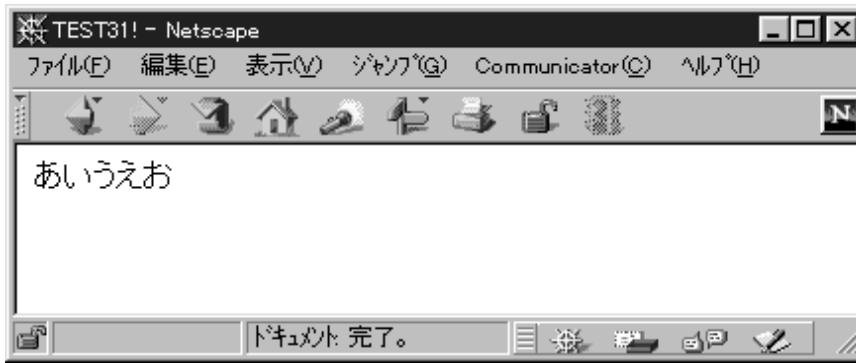
「1行目で初期値を設定して、2行目では、iが10以下のとき、次の命令を繰り返せ、と書いているわけです。」

「さあ、それでは、また少し難しくしていきますよー。(^^)p」

「次の問題です。」

「今度は配列変数というものがでできます。この問題は私が書いてみましょう。」

【問題5】1文字ずつ配列変数に文字を格納し、以下のように表示するプログラムを作成しなさい。



```
m=new Array("あ","い","う","え","お");
for(i=0;i<5;i++){
document.write(m[i]);
}
```

「この問題では配列変数というものが出てきます。これは変数に背番号のようなものをつけて、関連のある内容を代入したいときに使います。変数の名前には変数を使えないのでこのようにして[]を使ってmの何番、という処理をするんですね。」

「JavaScript では配列もひとつのオブジェクトとして扱われます。」

「では、書いてみましょう。」

3. 配列オブジェクト

変数 = new Array(〇〇);

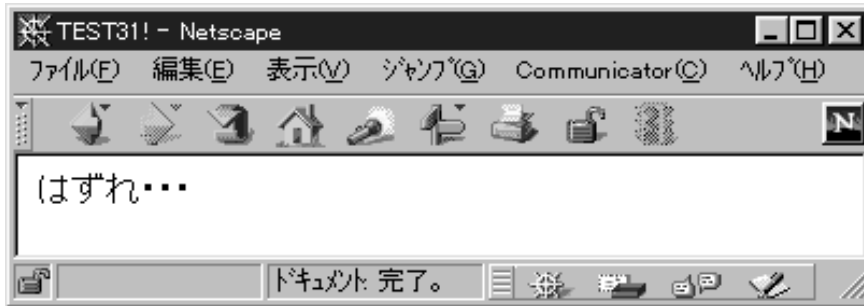
〇〇個の配列を宣言する。このとき直接代入することもできる。

「このようにして新しい配列を宣言するときは new という命令を使うんですね。」

「しかし、JavaScript ではファイルの保存などが弱いためあまり配列を使うことはありません。なぜなら、配列は何かデータがあって、それを読み込んで加工するときに都合が良いからです。JavaScript では読み込むデータ自体不自由なのです。(TT)」

「さて、それでは、いよいよ条件分岐にいきましょう。」

【問題6】乱数を使って、1ならあたり、0ならはずれを表示するプログラムを作成しなさい。



```
var r=Math.round(Math.random());
if (r==1){
document.write("あたり！");
}
else{
document.write("はずれ...");
}
```

「いよいよ、if 文が出てきましたねえ。早速書いてみましょう。」

4. if 命令

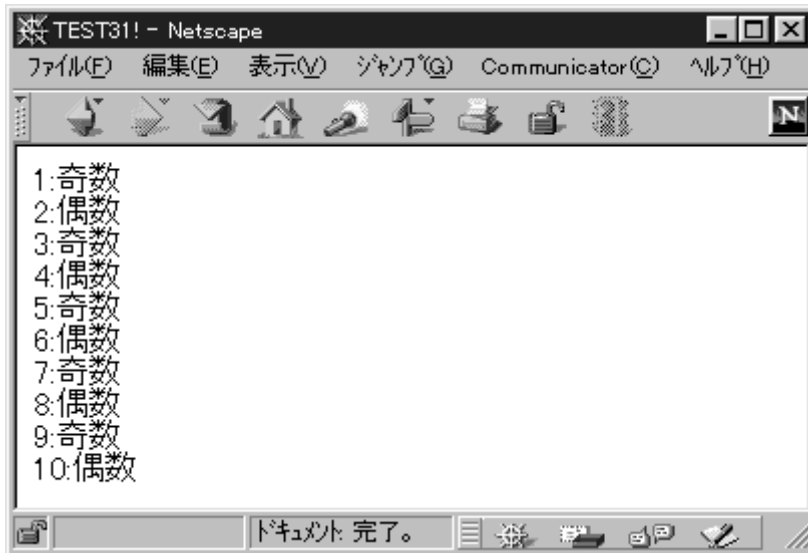
```
if (条件){
    【命令1】
}
else {
    【命令2】
}
```

条件が満たされていれば【命令1】をなかった場合、else 文があれば、【命令2】を実行する。複数の命令を記述するときは{~}で囲み、if 文の中にさらに if 文を記述することもできる。

「ここで見られるのはやはり条件式です。これは while のときと同じ比較演算子を使います。」

「まずは論より実行、早速次の問題をやってみましょう。」

【問題7】1～10までの数字を表示し、以下のように偶数なら偶数、奇数なら奇数と表示するプログラムを書きなさい。



```
for(i=1;i<11;i++){  
    document.write(i+":");  
    if (i%2){  
        document.write("奇数"+"<BR>");  
    }  
    else{  
        document.write("偶数"+"<BR>");  
    }  
}
```

「この問題ではまず何をもって偶数か、奇数か、ということから考えなくてはなりません。偶数か奇数かというのは2で割って余りが0ならば偶数ということですから、演算子のところを見てみますと、余りを求める演算子がありましたね。そう、%ですね。(^^)v」

「数値を表示する部分はどちらも同じですから、if文の前に書き、それ以降を条件分岐させるわけですねー。」

「さあ、どんどんと行きましょう。わからなかった人もなれるまではファイト！ファイト！」

【問題8】1～10までの数字を表示させ、奇数か偶数か、それとも3の倍数かを表示させるプログラムを作成しなさい。



```
for(i=1;i<11;i++){
    document.write(i+":");
    if (i%2==0){
        document.write(" 偶数"+"<BR>");
    }
    else{
        if (i%3==0){
            document.write(" 3の倍数<BR>");
        }
        else{
            document.write(" 奇数"+"<BR>");
        }
    }
}
```

「さて、今度は3の倍数を含んできましたねー。これはネ스팅と言って、if の後にさらに if を入れる、同じような処理を繰り返して行っているんですね。」

「3行目では2で割って余りがないものを、6行目では3で割って余りがないものが、それぞれひっかかるように組まれています。このようにして if や for は重ねて使うことが可能なんですね。」

「では次の問題は、どういう処理がされるか、考えてみて下さい。」

【問題9】次のプログラムはどういう処理を実現するものか。簡単に記述しなさい。

【実行結果】ブラウザがネットスケープ ver4 なら forNN4.html へ、IE4 なら forIE4.html へ、それ以外なら、for3.html へ自動的にジャンプさせるプログラム。



```

if(navigator.appVersion.charAt(0)>=4){
  if(navigator.appName.charAt(0)=="N"){
    location.href="forNN4.html"; }
  else{
    location.href="forIE4.html"; }
}
else{
  location.href="for3.html"; }

```

「リファレンスのナビゲーターオブジェクトを見て下さい。」

「そこにそれぞれ書いてありますねえ。ブラウザの名前、バージョン、と。そして、文字を抜き出す文字列オブジェクト charAt と組み合わせてますから、こういうことなんですねー。」

「このようにしてナビゲーターのプロパティを書き換えると、自動的にジャンプさせることもできるんですねえ。ですから、いずれメニューによるジャンプも学習していきます。」

「それでは、次回はフォームについて勉強していくことにしましょう。これを覚えると、ちょっとインタラクティブなページができるんですよー。」

「今日の授業はこれで終わります。(ˆvˆ)」