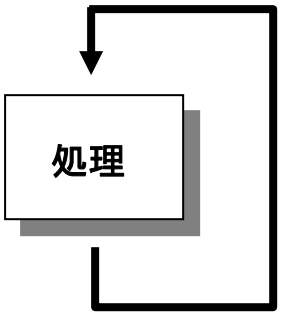
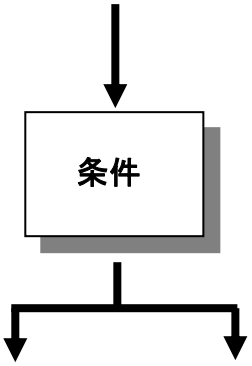
プログラミングで「繰り返し」「条件分岐」「ジャンプ」などの流れをコントロールする塊を制御文と呼びます。

Control Statement　制御文

制御文は、決められた回数**繰り返す**ことや、条件によって**分岐**するなど、直線的な処理でないプログラムを作るのに絶対に必要です。



繰り返し



分岐

繰り返し　for

for 文は、決められた回数、「処理」を繰り返すときに使われます。

```
文法：
    for(i=初期値; 条件式; 増分)処理文；
            または、

    for(i=初期値; 条件式; 増分){
        処理文１；*
        処理文２；
    }
```

「何回まで」と数えるので、こうした働きのことを「カウンター」と呼ばれることもあります。
for 文の中の変数には、一般的に「ｉ」が使われます。
さらに、変数が必要な場合は、「ｊ」「ｋ」と、順番に使われます。ただ、i と j は見間違えやすいので j を使わないで k が使われる場合もあります

* for などの（ ）や｛ ｝でくくられた文章は、見やすくするために、文頭にスペースを入れる。タブで字下げをしてもかまわないが、ほかのエディターで読みこんだとき、文字が置き換えられるときがあるので、注意。

>> for の例文：

```
for(i=1; i<=20; i++)
```

「1」から「１」づつ増やしながら、「20」になるまで処理をする。

「ｉ＋＋」は、
ｉ＝ｉ＋１と同じことをあらわしています。
「ｉ」を２度書く手間と、「ｉと１」の見間違いをなくすために使われます。

```
for(i=0; i<=360; i=i+30)
```

「0」からはじめて、「360」まで「30」ずつ加算して、処理をする、と書かれています。

■関係演算子

条件式には、次のような関係演算子が使われます。

関係演算子	
> 大きい	論理演算子
>= 大きいか、等しい	! NOT（否定）
< 小さい	&& AND（論理積）
<= 小さいか、等しい	|| OR（論理和）
== 等しい	
!= 等しくない	

上のサンプルの条件式

for(i=1;i<=20;i++)

「1 から 20 まで、1 ずつ増やす」は、プログラム的には本当は次のように書きます。

```
for(i=0; i<20; i++)
```

「0 から 20 まで（未満）、1 ずつ増やす」と書きます。
これまでは、人間に分かりやすく書いてきましたが、この方がコンピュータには分かりやすいのです。

プログラム 1：カウンターを使って、フォントサイズを順番に変える。

```
<SCRIPT>
    for(i=1; i<=7; i++){
      document.write("<FONT SIZE=”+i+”>");
      document.write("JavaScript</FONT><BR>");
    }
</SCRIPT>
```

（解説）

フォントサイズを変えるのは、ここでは＜font size=””＞タグを使います。
タグは以前学習したように document.write の中でダブルクォーテーションで囲うと実行されます。

「＋」は文字の連結で、"ABC"+"123"のように書けます。この出力は、ABC123 です。
また、文字だけではなく、「文字列」と「数値」の混在した記述もできます。

```
例: for(i=0; i<3; i++) {
```

この出力は、文字＋数字になります。

```
ABC0ABC1ABC2
```

>> write 内での文字の連結表現

write メソッドは、文字列の連結のために「＋」もしくは「,」が使えます。

```
document.write("ABC"+"abc");
document.write("ABC","abc");
```

上の 2 つの文は、同じ結果が得られる。
初めのうちは分かりやすい「＋」を使えば良いのですが、これもプログラミングの世界では「,」が使われます。書籍などのサンプルも「,」で書かれていますが、非常に見づらいです。

2. 条件分岐　if

次に条件分岐を学びます。

if 文は、条件を判断し、条件を満たした場合、それに続く処理を実行します。

私たちは日常さまざまな「条件分岐」をしています。

たとえば、「昨日、バイトの給料が入ったので、友達と食事に行ける」「5 分のバスが行ってしまったので、慌てる必要はない」「あの授業、2 回しか休んでないので、今日は行くのやめよう」などなど。

条件の判断で行動することが多くあります。

これをプログラムの中でするのが条件分岐です。

ほとんどのプログラムでは「if」を使います。

```
文法：
    if(条件式) {
     条件成立時の実行文
    }
```

例で書けば、

```
if（これまで授業を休んだ回数は 5 回以下か）{
    授業をサボる；
}
```

条件が成立しない場合は、}の次の行に進みます。

```
文法：
    if(条件式) {
     条件成立時の実行文
    }
. . . . .
```

条件不成立

また、不成立の条件が、次々にでてくる場合の書き方。

この場合には「elseif」を使います。

```
文法：
    if(条件式 1){
        処理文；
    }
    elseif（条件式 2）{
        条件 2 成立の時の処理文；
    }
    elseif（条件式 3）{
        条件 3 成立の時の処理文；
    }
    else{
        それ以外の場合の処理文；
    }
```

たとえば、
「小学生ならランドセル、中学生ならカバン、高校生ならスポーツバック、それ以外は手ぶら」
のように、条件を次々と出し、条件に合えば、その処理分を実行する。実行後は一番下にです。この場合は、else があるので、どんな場合も必ずなんらかの処理がなされる。

■「＝」記号 代入

数学で、等しいことを表す「＝」は、コンピュータでは「**代入**」を示す場合に使われます。

A=A+B

上の式は、A と B を加えたものを、左辺の「A」に代入することを示しています。
したがって、等しいことを示す場合には、別の記号が必要になる。ここで使われるのが「＝＝」

例:　if(A==3)
「もし、A が３だっら（３と等しいのならば)...」

ちなみに、「＝＝＝」というのもある。この場合は完全一致。

コラム「日本語プログラミング言語」
インターネットが登場する直前、電子手帳ブームがあった。その時世界中で大ヒットしたのが HP200LX。その中で「日本語プログラミング言語」があった。ただ実際にやってみると日本語のコマンドがめんどくさい。日本語にはいろいろな表現があるので間違えると当然動かない。
やっぱり「if」とだけ打った方が楽。

プログラム 2：文字"A"をフォントサイズを 1 から 7 まで変えながら、50 回書く。
7 文字ごとに改行する。

```
<SCRIPT>
  for(i=0; i<=48; i++){
  document.write("<FONT SIZE="+(i%7+1)+">");
  document.write("A");
    if (i%7==6){
        document.write("<BR>");
    }
  }
</SCRIPT>
```

（解説）
i%7==0 は、i を 7 で割ったときの余りを求めているので、i が 7 になったとき余りが 0 になる。それを 6 とで比較（==6）している。この割り切れた場合が、条件に合う場合で、改行
の指示が実行される。

3-3.多重ループ

JavaScript では、多重ループもできます。
「多重ループ」とは、ひとつのループの中に、別のループが入っていることを言います。このことを「ネスト（入れ子）」ともよびます。
ループは、外側のループから始まり、内側のループが実行される。内側のループが終了すると、実行は再び、外側のループに戻る。

```
for(i=0; i<=3; i++){
 外側のループ１
  for(j=1; j<=2; j++){
    内側のループ
  }
 外側のループ２
}
```

プログラム：「2 行 3 列のテーブル(表)を作成し、その中に行と列の位置を示す文字を表示する。」

```
<SCRIPT>
   document.write("<TABLE BORDER>");
   for (i=0; i<2; i++){   //外側のループ －＞ 行の要素を決めている
      document.write("<TR>");
      for (j=0; j<3; j++){
      //内側のループ －＞列の要素を決め、ひとつひとつのセルに位置を示す番号を表示
         document.write("<TD>cell(",i,",",j,")</TD>");
      }
      document.write("</TR>");
   }
   document.write("</TABLE>");
</SCRIPT>
```

応用： 上の方法を応用すれば、「順番に番号の付けられたイメージを表示する」ことができる。（あまり実用的ではないけど...）

```
<SCRIPT>
   document.write("<TABLE BORDER>");
   for (i=0; i<2; i++){
      document.write("<TR>");
      for (j=0; i<3; j++){
         document.write("<TD><IMG
SRC=img",i*3+j+1,".gif></TD>");
      }
      document.write("</TR>");
   }
   document.write("</TABLE>");
</SCRIPT>
```

（参考）テーブルの作成

テーブル(表)の作成のためには、大枠に「タグ」

```
<TABLE BORDER>
```

```
</TABLE>
```

が、必要である。

各「行(row,line)」は、<TR></TR>でつくる。

「列(column)」は、行のブロックの中に書く。

```
<TR>
<TH>***</TH><TH>###</TH>
</TR>
```

データ要素の場合の列は、<TD></TD>である。

```
<TR>
<TH>A</TH><TD>123</TD>
</TR>
```

TH（head）

TR

***	###
A	123
B	456

TD（data）