

この時間のねらい：① 繰り返し処理を行えるようになる。

前回まで座標自体を操作するための関数を学んでいました。今回は、繰り返し処理を行えるようになることが目的です。

繰り返し処理の記述方法は 2 種類あり、それぞれ以下のような記述方法になります。

```
1 for(int i = 0; i < /*繰り返し回数*/; /*iの増加方法*/) {
2   /*処理内容(iを使って記述)*/
3 }
```

```
1 int i = 9; //iを宣言し値を代入
2
3 while(i > 8/*iの条件を記入*/) {
4   /*処理内容(iを使って記述)*/
5   /*iの増加方法を記入*/
6 }
```

それぞれ_____と呼びます。どちらもやっていることはほぼ同じになります。_____繰り返すか、繰り返す際、_____はどれほどかを記述します。

その記述方法が違うというだけであるので、繰り返しであれば、どちらを使っても大丈夫です。(というより、どちらかを使えば、開発は可能になります。) また、i の増加方法の記述については、以下のような宣言方法があります。

i++;	後からインクリメントを行う演算子	1 つ目から 4 つ目の演算子はそれぞれインクリメント演算子、デクリメント演算子と呼びます。やっていることは簡単で変数 i に 1 を足す、1 を引くという処理です。
++i;	先にインクリメントを行う演算子	
i--;	後からデクリメントを行う演算子	
--i;	先にデクリメントを行う演算子	
i += 整数 or 変数;	i に整数又は変数を足す。	

ただし、書く場所によって処理の順番が変わります。

演習問題 1

インクリメント(デクリメント)演算子を重ねることで、どのような値の変化が起こっているか、コンソールへ表示して確認しなさい。また、各場所によっての値の変化も同じように確認をしなさい。

for 文、while 文にてこれらの演算子を使用する場合はどちらを使用しても問題ないので、好きな方を利用してください。

繰り返し文を理解するために演習問題に挑戦して、使い方をマスターしましょう。

演習問題 2

```
1 void setup() {  
2   //ウィンドウを開く関数を記述  
3 }  
4  
5 void draw() {  
6   //中心座標を(120, 120)で円を作成  
7 }
```

左のプログラムを実行させ、また、コメントのある部分については関数を書くことで、ウィンドウの表示と円を表示させなさい。

演習問題 3

条件文を利用することで、横に円を 4 つ並べるプログラムを作成しなさい。また、それらをさらに、縦に 4 つずつ並べて、合計 16 個の円を表示しなさい。

<メモ欄>

演習問題 4

新規ファイルを作成しなさい。(左上の「ファイル」→「新規」です。)

この問題では、九九の表の作成を行う。以下のようにコンソールに表示するプログラムを作成してください。

```
1 2 3 4 5 6 7 8 9  
2 4 6 8 10 12 14 16 18  
3 6 9 12 15 18 21 24 27  
4 8 12 16 20 24 28 32 36  
5 10 15 20 25 30 35 40 45  
6 12 18 24 30 36 42 48 54  
7 14 21 28 35 42 49 56 63  
8 16 24 32 40 48 56 64 72  
9 18 27 36 45 54 63 72 81
```

縦があっていないという状態になっていますが、コンソールにて表示をする場合、そろえる方法が面倒なため、とりあえずは左のような形で表示することを考えてみてください。

最後に繰り返し文に関するサンプルは Ex_04_06 から Ex_04_13 にあるので困った場合はそれを見て確認するようにしてください。