

この時間のねらい:

1. 同じ処理を何度も行う方法を学ぼう。
2. ループ回数を自動で決めるプログラムを組めるようになろう。

メモ欄

そもそも、プログラムはパソコンに(効率よく)仕事をを行わせるための命令文です。

それを踏まえて、演習問題「ウィンドウ内に100個の円を描きなさい。」という問題を出された場合、
皆さんはどうします？

あなたの考え:

それでは効率よく、スマートにこの問題を解決させましょう。(ループ関数)を利用します。

```
for(int i = 0; i < 100; i++) {  
    //処理の内容  
}
```

上のプログラムでは i の値によって、ループさせるかどうかを決めています。

左側(int i = 0)は初期の値、真ん中(i < 100)は条件文、右側(i++)は次に進む処理
となっています。利用してみて、わかっていくと思います。

ここから、ループ関数と相性の良い関数を紹介していきます。

- ① random(100); ... (ランダムな値を決める)
- ② random(0, 100); ... (左から右の間で、ランダムな値を決める)
- ③ randomSeed(1); ... (あらかじめ決まっているランダム値を呼び出す)

※これらの関数のほかに、変数宣言、図形、色など多くのものがあります。

実は、ループさせる関数にもう一種類あります。それは(while)関数です。

以下のように作成します。

```
int i = 0;  
while(i < 100) {  
    //処理  
    i++;  
}
```

さっきのfor関数と比べれば、こちらのほうがもとのプログラムと同じような見た目になっていま
す。(画像を見て流れがつかめた人は、そこそこにプログラムの考え方が身についてきています。)
今の2つの関数のうち、どちらか一方を利用できれば、効率よく、プログラムを組むことができるよう
になります。

ここから先は、ループ関数の応用的な使い方をしましょう。これができるとほぼ自動化をさせられる

メモ欄

ため、ぜひできるといいなという内容です。

右の画像は細かくてわかりづらいですが、48ピクセルの円が画面

いっぱい敷き詰められています。これこそ、ループ関数の為せる業ですね。

それでは、これをウィンドウサイズを変更しても、すべて敷き詰めるというのを
実現できるようにしましょう。

とりあえず、考えるウィンドウサイズを(480, 480)に決めましょう。

このとき、縦・横48ピクセルの円は(10)個並びます。(右の画像)

ということは、ループ回数は(10)回となればよいですね。

ただし、円の基準があるため、円の座標を下のどちらかで解消させましょう。

解決策1:座標に円の半径($d/2$)を足す。

解決策2:(`ellipseMode(CORNER);`)を利用して、座標はそのままにする。

ここまで実行できれば、敷詰めは完了ですね。ただ、これでは、ウィンドウサイズの変更に
対応していません。ウィンドウサイズへの対応に関しては、ループ回数の部分が関わります。

今回の直径(48)ピクセルの円を縦横(480)に敷き詰めるので、(10)個並ぶ。

計算方法は($480 \div 48 = 10$)となります。

ということで、右下の画像のようにプログラムを作成すると、ウィンドウサイズにも対応
できます。

さすがに、皆さんの中で、これらを理解できるという人は少ないと思います。ただ、プログラムの作成
をこれから先、どんどんやっていく場合、これくらいのことを考えるのは必要になってきます。

```
int d = 48;
```

```
size(480, 480);
```

```
ellipseMode(CORNER);
```

```
for (int i = 0; i < width/d; i++) {  
  for (int j = 0; j < height/d; j++) {  
    ellipse(d*i, d*j, d, d);  
  }  
}
```

☆課題

前半で学んだ中に、`while()` 関数があります。

右の画像は、この関数を利用しても、プログラム作成が可能です。

作成してみてください。

自分の作成したプログラムをメモしてください。

担当教員の例をメモしてください。

