

この時間のねらい: ① 基本的な図形の描き方。  
② オリジナル図形の作成方法。

皆さんが直線を引きたいときには、あるツールを用意すると思います。それは\_\_\_\_\_ですね。

これをプログラミング内で利用する際には\_\_\_\_\_を利用して実現させます。

今回のねらいでもある、図形の描画を行う関数を学びましょう。

〈図形を描画する関数〉

①点 `point(_____の位置, _____の位置);` 直線 `line(x1, y1, x2, y2);`

②円・楕円 `ellipse(_____のx座標, _____のy座標, width, height);`

③正・長方形 `rect(_____x座標, _____y座標, width, height);`

他四角形 `quad(x1, y1, x2, y2, x3, y3, x4, y4);`

④三角形 `triangle(x1, y1, x2, y2, x3, y3);`

⑤\_\_\_\_\_ `arc(中心x座標, 中心y座標, width, height, _____の始まり, _____の終わり);`

基本的な図形を描く際には以上の関数が使用されます。よく使われる図形を簡単に作成できるように用意されています。

まあ、察しの良い人はその前に\_\_\_\_\_を開く必要があるのでは?と思うでしょう。

それでは、大きさの指定し、ウィンドウを表示する関数を学びましょう。

〈ウィンドウを表示する関数〉

`size(横方向の長さ, 縦方向の長さ);`

ちなみに、Processingに限らずプログラミングでの大きさの指定方法は\_\_\_\_\_→\_\_\_\_\_の大きさの順が多いです。

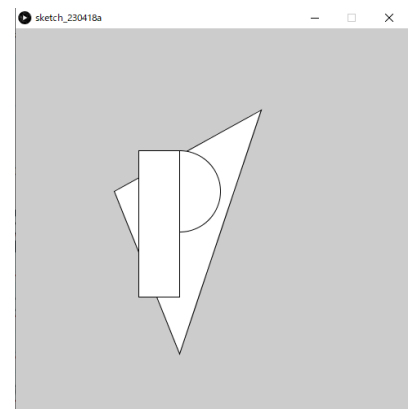
また、数学と同じように横方向を\_\_\_\_\_, 縦方向を\_\_\_\_\_, というように表します。

それでは、関数を組み合わせて実行してみましょう。私は無難な感じで下のようにしてみました。

左はプログラムで右はそれを実行した結果になります。この結果から、あることが分かりますね。どんなことでしょうか?

ヒント: 図形の出ている順番はどんな順番でしょうか?

```
sketch_230418a
1 size(480, 480);
2 triangle(300, 100, 120, 200, 200, 400);
3 ellipse(200, 200, 100, 100);
4 rect(150, 150, 50, 180);
```



それでは、図形の表示の順番が\_\_\_\_\_→\_\_\_\_\_→\_\_\_\_\_という順番になっていることに注目しましょう。

このことから、プログラムで記述したものは上から順番に実行していくということが分かりますね。(圧)

これはプログラムの処理の種類の一つとして、\_\_\_\_\_又は\_\_\_\_\_と言います。

続いて、私たちの世界に存在する形というのは先ほどの5種類以外に何かあるでしょうか？私の考える例としては  
(例)・\_\_\_\_\_・\_\_\_\_\_・\_\_\_\_\_・\_\_\_\_\_

まあ、もちろんこの図形も実装できるなら実装したほうがよいと考える人もいるかもしれませんがね。でも、されていない理由は開発者(Casey Reas と Ben Fry)の開発理由は、「あくまでもJavaを簡単に使えるように」ということであるため、Javaの開発環境にて使用できる図形のみが実装されているということです。

というより、もし、実装が可能であったとしても、利用するタイミング、人を選ぶような図形を入れておくのは余りにメリットが\_\_\_\_\_ように感じますよね。たとえできたとしてもやられないでしょう。

さて、それでは例の中の図形を作成してみましょう。また、例を見ただけで自分でも図形を作成してください。  
例)

```
sketch_230424a
1 beginShape();
2 vertex(10, 40); //左側上の点
3 vertex(90, 40); //右側上の点
4 vertex(20, 90); //左側下の点
5 vertex(50, 10); //一番上の点
6 vertex(80, 90); //右側下の点
7 endShape();
```

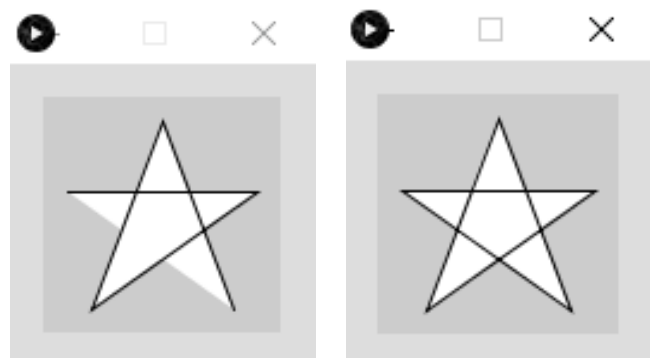


図1:

図2:

左のように打ち込みを行うと右上の図1, 2のどちらかになります。実際に実行し、該当の図の下に「左を実行」と書きましょう。

さて、それでは該当しない図はどのように表示するか。それは7行目にあるendShape(); に秘密があります。

必ずと言って良いほどに、関数(青字になるやつ)には()がありますね。これの中に値が入ることがあります。この値を\_\_\_\_\_とい  
います。それでは以下のように打ち込んでください。

```
8 endShape(CLOSE);
```

このendShapeの引数は大文字で"CLOSE"と書くだけです。それではこれで表示される図の下に「引数ありバージョン」と記入し  
ましょう。引数については以上です。

それでは最後にこれを作成しているプログラムを細かく見ていきましょう。

1行目:beginShape(); はプログラミングでよくいう\_\_\_\_\_です。必ず記述するものです。忘れないように。

2~6行目:vertex(x座標, y座標); は点をプロット(取る)ための命令です。点を取った後、次の点を取る前に2点を直線で繋ぐ  
いう処理を続けていきます。そのため、最後と最初をつなげる場合は閉じることを引数で書く必要があります。

7行目:endShape(); はここで終了という宣言です。パソコンというのは言われたことしかできません。「ここまでだよ」と伝え  
てあげないと永遠に処理し続けます。そこに意志はあるのか?・・・ありません。

さあ、以上のものを使い国旗をがんばって作成してみましょう。まず手始めに日本の国旗、その後にほかの国旗を作成です。

ちなみに、色を付ける際はfill関数を使います。次回詳しくやりますが、気になる人は調べてみてください。