

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。

人間が基本的な図形を描くためには、ペンや定規、コンパスなどがあると便利です。実はプログラミングにも似たものがあり、それらは関数と呼ばれております。

ただ、図形を表示するためには、表示場所を作る必要があるため、まずは表示領域の作成方法を知りましょう。

表示領域は右のように決めます。

```
size(横方向の長さ, 縦方向の長さ);
```

では、基本的な図形の描き方を下で学びましょう。

基本的な図形として挙がるものは以下のものです

① 点と直線 ② 円・楕円 ③ 四角形 ④ 三角形 ⑤ おうぎ形

これらはそれぞれ以下のように表示していきます。

① 点 `point(x方向の位置, y方向の位置);` 直線 `line(x1の値, y1の値, x2の値, y2の値);`

② `ellipse(中心のx座標, 中心のy座標, width, height);`

③ 正・長方形 `rect(左上x座標, 左上y座標, width, height);`

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

④ `triangle(x1, y1, x2, y2, x3, y3);`

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

基本的な図形を描きたいときは、以上の関数を使用します。

<メモ>

これらを組み合わせて描く際には、ノート に描いてみて、位置決めをする方が良い。

その後、実際に実行し、どのくらいの位置かを決めて、微調整を加えていく。

私たち人間には 1 ページのような図形以外にもいろいろな図形が存在している。

例えば右のようなものが挙げられる。

複雑な図形の代表例

これらを描くためには先ほどとは違う形で、
図形を描く必要がある。

- ・ ☆(星、スター)
- ・ 四以上の多角形
- ・ 六芒星
- ・ 傘(応用で相合傘も)

⇒なぜなら、開発者(Casey Reas と Ben Fry)
からすると利用者がどのような図形を描くかを

考える事が難しいから。あとは実装するメリットが ほぼない ことが理由。

<書き方>

右のように書きます。右の意味は以下で見ましょう。

最初の `beginShape()`; は図形の 作成 の宣言。

「俺はここから書き始めるぞ!」という意思の表れです。

次から 5 行ある `vertex(x, y)`; の内容は一つ一つの
点のプロット位置(座標)です。

```
beginShape();  
vertex(x1, y1);  
vertex(x2, y2);  
vertex(x3, y3);  
vertex(x4, y4);  
vertex(x5, y5);  
endShape(CLOSE);
```

プログラムの下に実行した図形で見てみると、

書きこんだ状態と同じようになります。

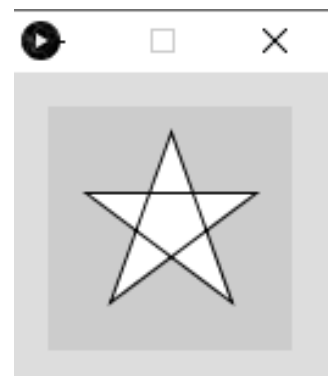
最後の `endShape(CLOSE)`; は 1 行目とセットです。

ここまでで図形の作成終了という意味になります。

「俺は図形作成をやり切ったぞ!」という達成感の表れです。

また、`endShape()`; でも実行することができます。

このように()の中に入れる値は関数によって違います。この()内に入れる数値、文字などのデータを 引数と言います。この引数を間違えてしまうと、エラーが出るので気を付けましょう。



<次回までの課題>

国旗を形だけ作成しておく。