

この時間のねらい：① 移動・伸縮・回転させるための関数を使えるようになる。①

前回までは条件文を効果的に使うための方法を学びました。今回までの内容、これからの内容は、開発において非常に重要なものになるため、しっかりと理解しておきましょう。今回行うのは、移動・伸縮・回転をさせるための関数になります。それぞれ下にまとめておきます。

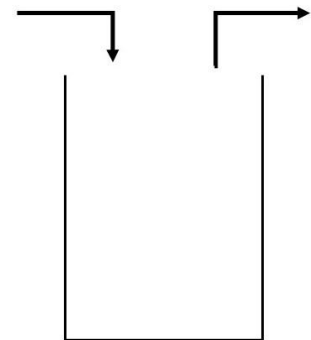
translate(x 座標の値, y 座標の値);	原点の座標を引数の場所に移動する。
scale(大きさの割合);	後に描く図形の大きさを割合で管理する。
rotate(角度);	引数分だけ回転させる。※弧度法で定義する。

今回はこの中でも特に移動のための関数に関して学びます。移動を行うために理解すべきものは、スタックとキューになります。これは IT・パスポート試験でも出題されるものになるため、必要な方は覚えておきましょう。

<スタックとキュー>

・スタックとは

右の図のような容器がスタックになります。この容器にデータを収納する際、出し入れは上からしかできません。この容器にデータを入れることをプッシュ、出すことをポップと言います。また、方式としては FILO 又は LIFO の 2 パターンの表現がありますが、どちらも同じ意味になります。

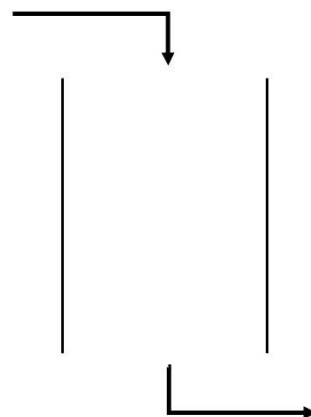


日本語では後入れ先出し方式などと言われます。

・キューとは

右の図のような容器をキューといいます。

スタックとは違い上と下が開いている容器です。正直、容器と言ってもいいのか際どいものになります。この中にデータを入れることをエンキュー、出すことをデキューと言います。また、上から入れて下から出すというように決まっているので間違わないようにしましょう。方式としては、FIFO というように言います。LILO とはあまり聞きません。日本語では待ち行列と言われます。



処理を行っていると、translate()関数を何度も使いたくなるようなことがあります。実は利用しやすい関数であるので、開発するものによっては何度も使う必要が出てきます。その場合、座標を別々に管理するためには、以下の関数を使用する必要があります。

pushMatrix();	座標データを FIFO 方式で格納する。
popMatrix();	FIFO 方式で格納した座標データを取り出す。

以上のものを利用することで、translate()関数の利用方法の幅が広がります。

#### 演習問題 1

ウィンドウサイズ(480, 480)のウィンドウを開き、円を中心に描きなさい。

(void setup(),void draw())を利用して書き始めなさい。)

#### 演習問題 2

先ほど描いた円を translate()関数を利用することで、マウスカーソルの座標によって、動くように追記しなさい。

#### 演習問題 3

円のほかに四角形を作成しなさい。ただし、四角形の表示位置は左上頂点の座標を (80, 60)に固定しなさい。(そのまま定義をすると一緒に動いてしまう。)

translate()関数を使用したサンプルとしては、Sample06Transform のところにあるので、残りの時間はそれを見て確認してください。