

モバイルシステム演習

Processing

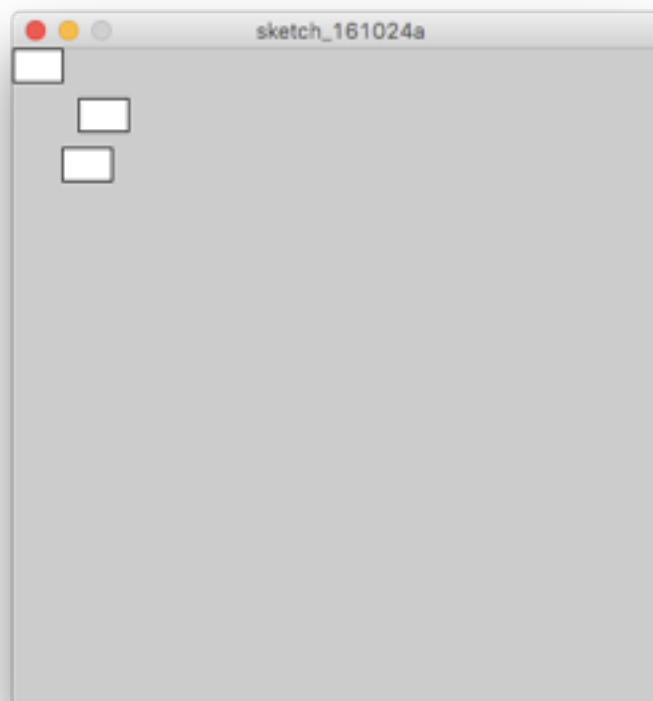
- 図形の移動、回転 -

座標軸を移動

translate(x, y)

現在の座標軸からx, y ピクセル分だけ移動させる

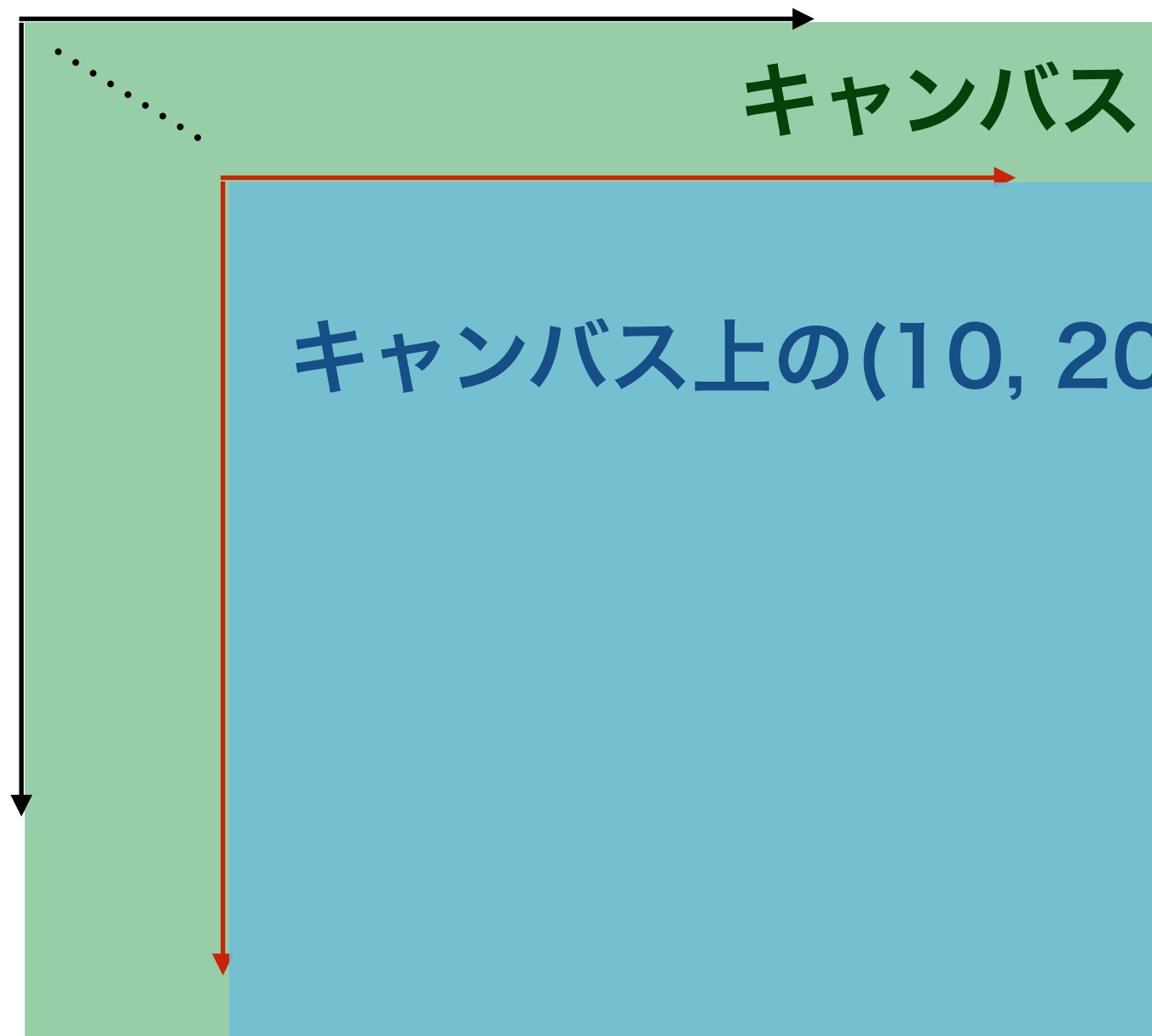
```
rect(0, 0, 30, 20);  
translate(40, 30); // 座標軸を現在の座標軸から右に40px, 下に30px移動  
rect(0, 0, 30, 20);  
translate(-10, 30); // 座標軸を現在の座標軸から左に10px, 下に30px移動  
rect(0, 0, 30, 20);
```



座標軸を移動

(0, 0)が原点

translate(10, 20)

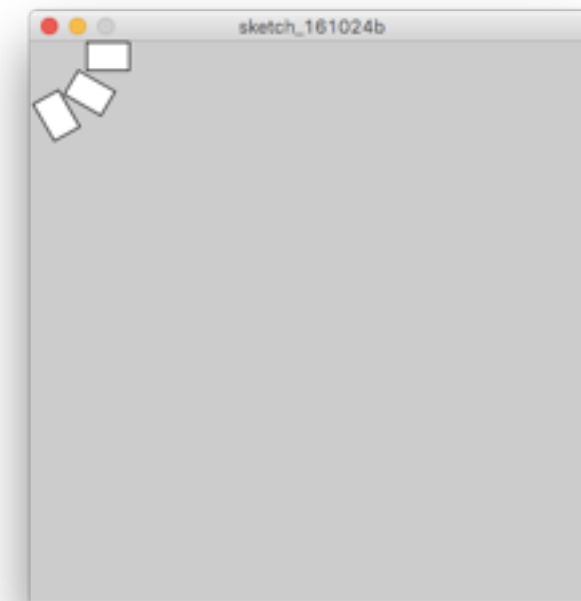


座標軸を回転

rotate(角度)

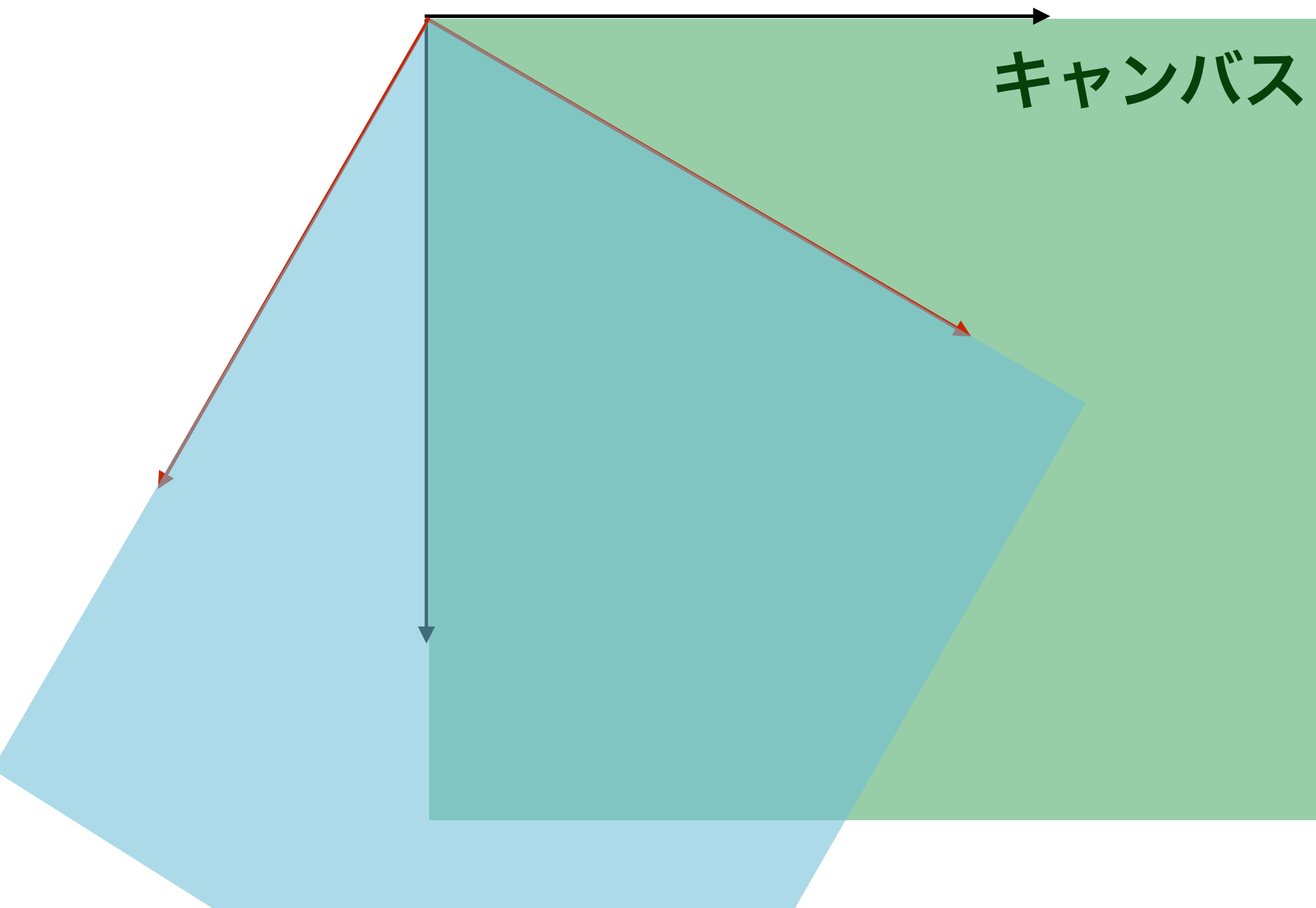
現在の座標軸から角度分だけ回転させる。
角度の単位はラジアン。つまり π で表記する
 $\pi = 180$ 度

```
rect(40, 0, 30, 20);  
rotate(PI/6); // 座標軸を現在の座標軸から 30°回転  
rect(40, 0, 30, 20);  
rotate(PI/6); // 座標軸を現在の座標軸から 30°回転  
rect(40, 0, 30, 20);
```

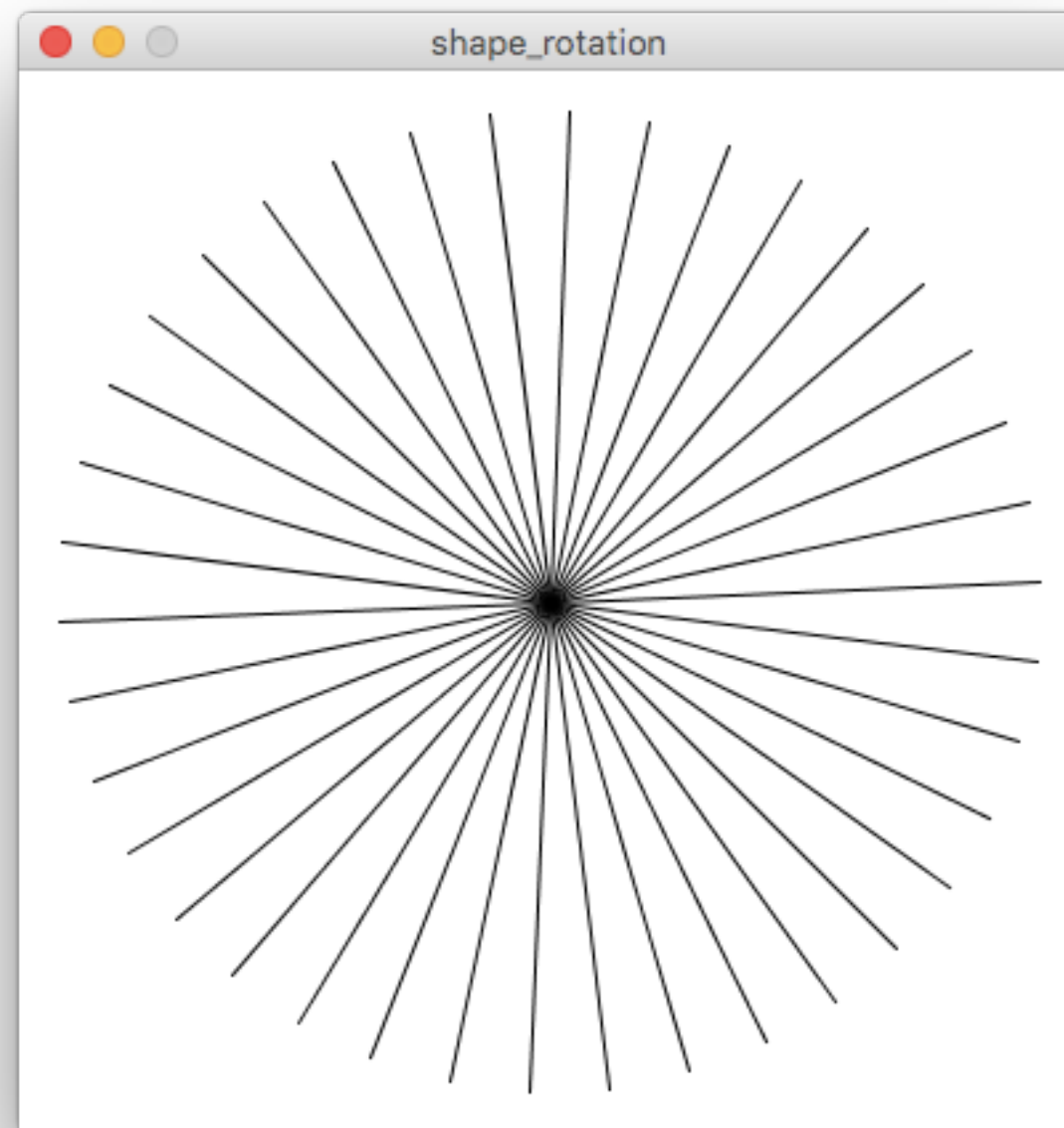


座標軸を移動

translate($\pi / 60$) ... 30度回転

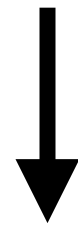


この図形を書いてみよう



座標系の保存

いままでの書き方だと、すべての図形に対して
座標系が変更になってしまう



図形ごとに座標系を決定したい！

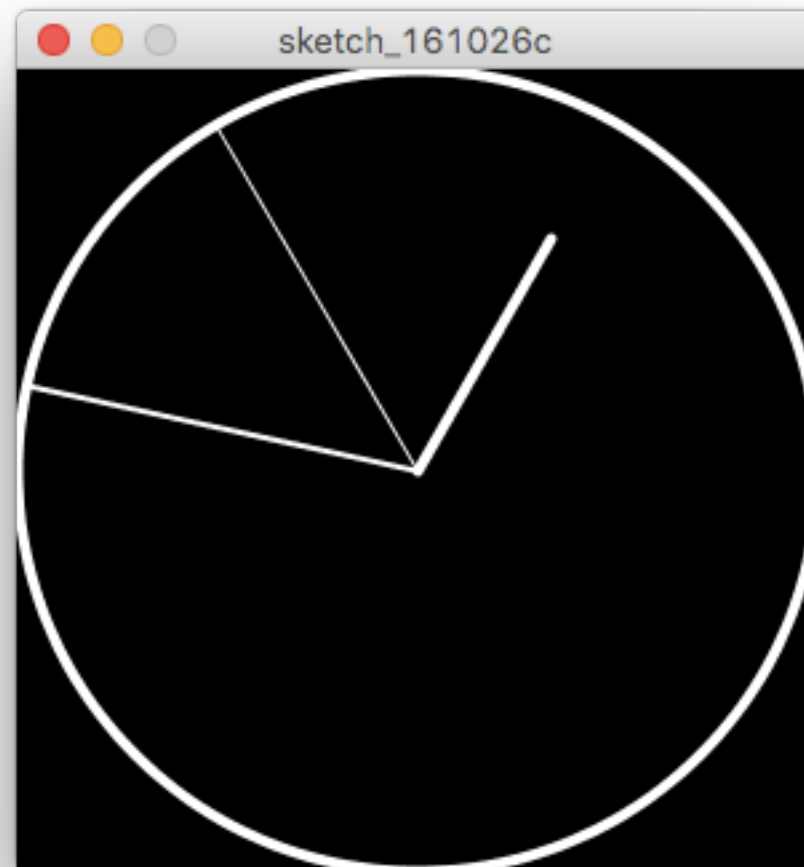
座標系の保存

pushMatrix → 現在の座標を保存する。
popMatrix → 元の座標に戻す。

How to

```
pushMatrix();  
// 座標系を変換するコードを記述  
translate(width / 2, height / 2);  
rect(0, 0, 100, 100);  
popMatrix();
```


今日のゴールは時計を作ること！



まずは現在の時刻を表示してみよう

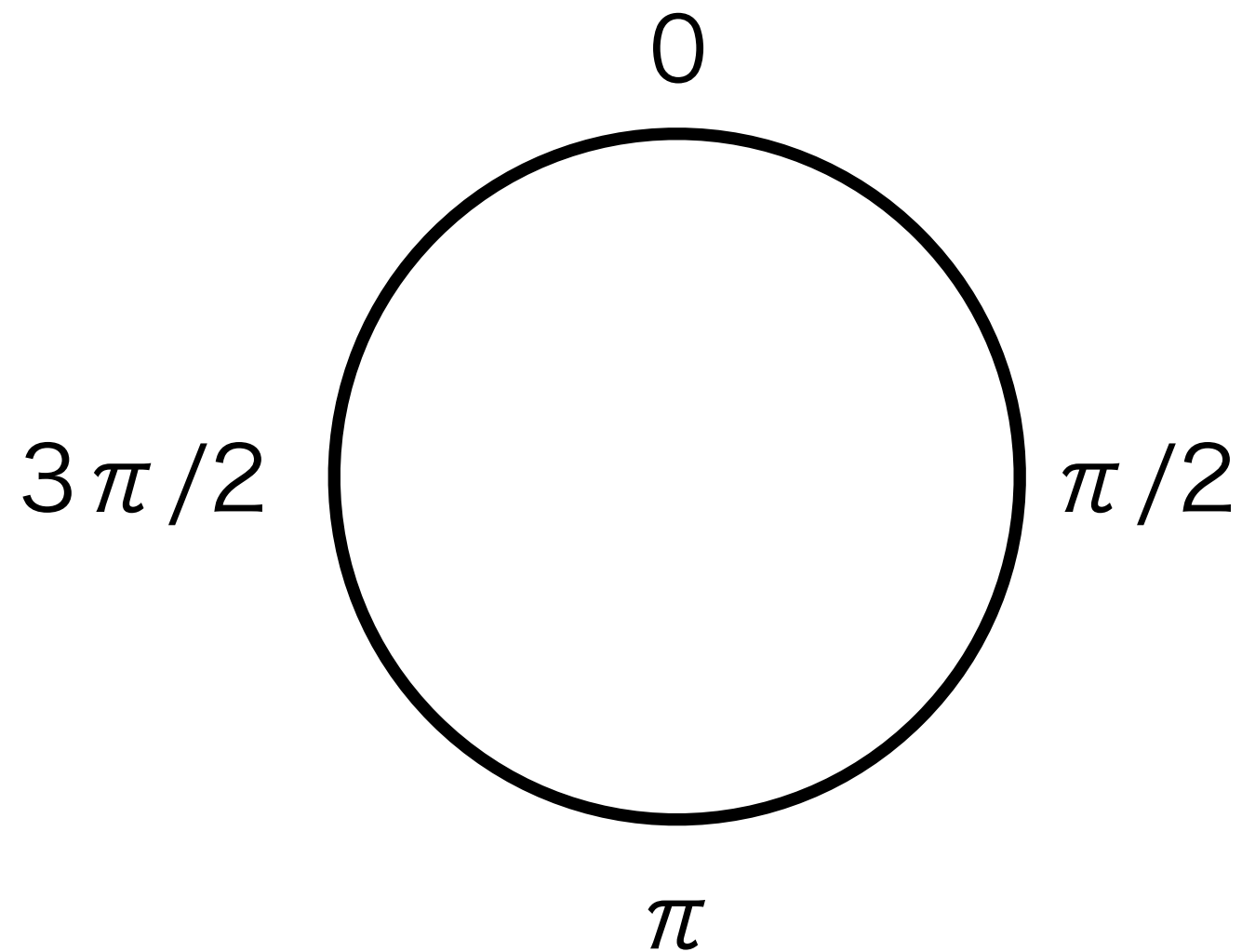
//秒の取得
second();

//分の取得
minute();

//時の取得
hour();

```
void draw() {  
    int h = hour();  
    int m = minute();  
    int s = second();  
    println("現在の時刻は "+h+": "+m+": "+s);  
}
```

長針、短針、秒針がそれぞれ何度ずつ
傾いているかを計算する



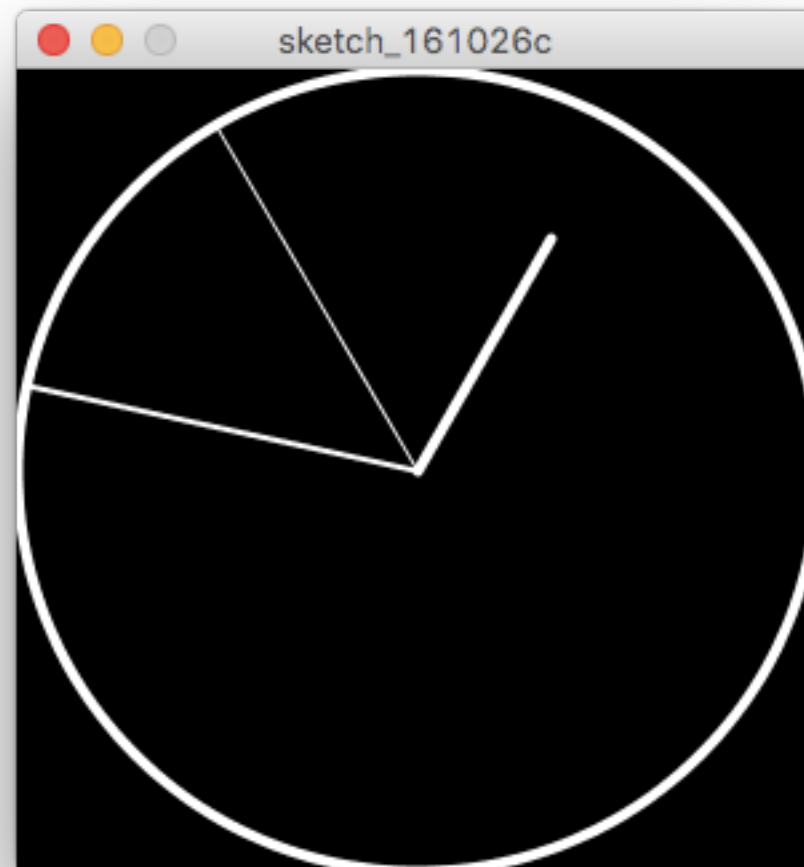
```
//秒針  
pushMatrix();  
rotate(radians(s*(360/60)));  
strokeWeight(1);  
line(0,0,0,-height/2);  
popMatrix();
```

```
//分針  
pushMatrix();  
rotate(radians(m*(360/60)));  
strokeWeight(2);  
line(0,0,0,-height/2);  
popMatrix();
```

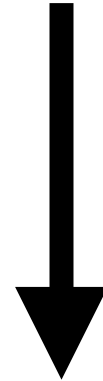
```
//時針  
pushMatrix();  
rotate(radians(h*(360/12)));  
strokeWeight(4);  
line(0,0,0,-height/3);  
popMatrix();
```

それぞれ座標軸を別にして
ラインを描画する

現在時刻の表示が完成



実際には分針、時針はそれぞれゆっくり動いている



実際の分針

秒の影響を受けている

毎秒あたり、 $1/60$ の影響 (1分 = 60秒なので)

$m = m + (s/60);$

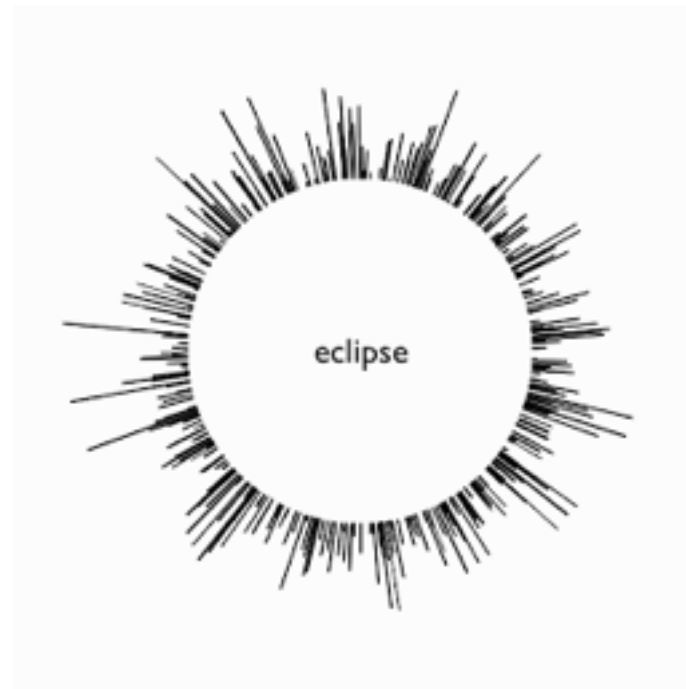
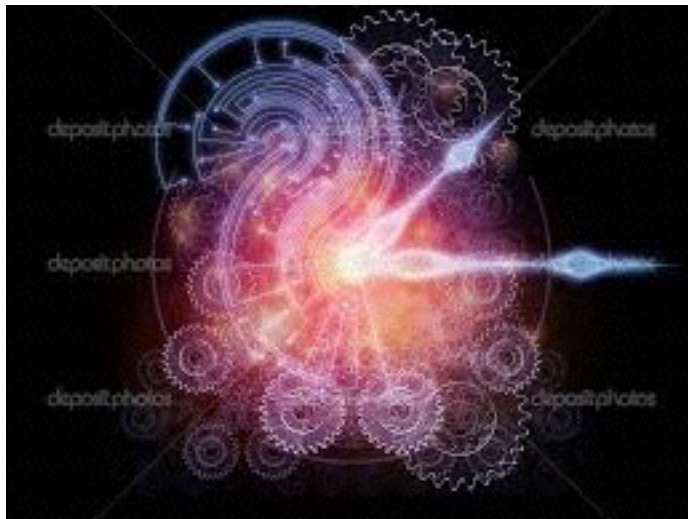
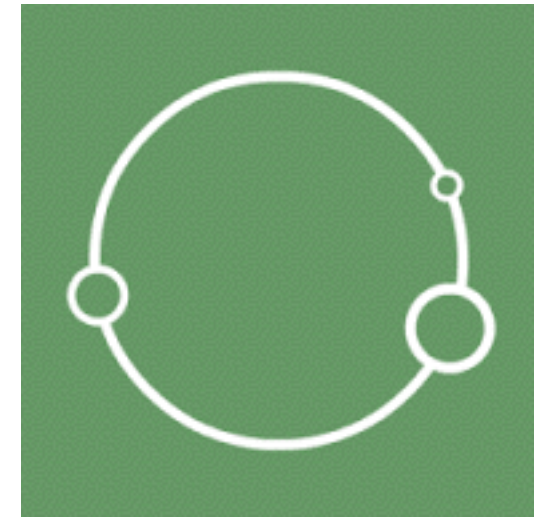
実際の時針

分の影響を受けている

毎分あたり、 $1/60$ の影響 (1時間 = 60分なので)

$h = h + (m/60);$

いろんなデザインの時計を作ろう



来週は
ちょっとした画像処理と
iPad上での表示