



# (ヒント)円を動かしたい

画面の左端から右端へと移動する場合は、`drawCharacter1`を利用して下記のようにしたらOK

```
void setup(){
  size( 800, 600 );
}
void draw(){
  background( 255, 255, 255 );
  println( frameCount );
  drawCharacter1( frameCount, 300 );
}
```

右端に来たら左端に移動するようにするには、座標をどう移動するかを考えるだけ

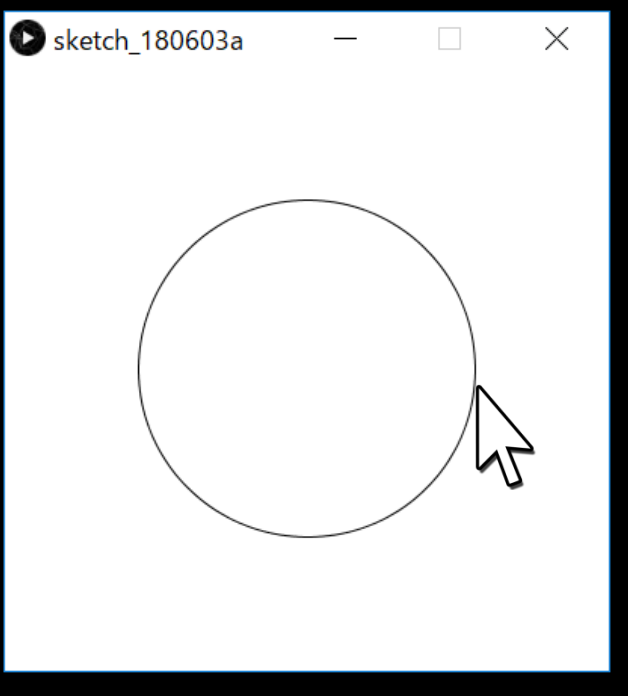
- `frameCount`が 801のときは？( `0`, 300 )
- `frameCount`が 830のときは？( `0`, 300 )
- `frameCount`が1700のときは？( `0`, 300 )

# プログラミング演習I (第3回) 課題

## ● 基本課題② スケッチ名: circle

- 400x400のウィンドウの中心からマウスカーソルが円周上にあるような円を描くプログラムを作成せよ
- また、その円の半径と面積, 円周の長さを標準出力せよ

```
半径の長さ111.8034
面積39269.91
円周の長さ702.4815
半径の長さ111.8034
面積39269.91
円周の長さ702.4815
半径の長さ111.8034
面積39269.91
円周の長さ702.4815
半径の長さ111.8034
面積39269.91
円周の長さ702.4815
半径の長さ111.8034
面積39269.91
円周の長さ702.4815
半径の長さ111.8034
面積39269.91
円周の長さ702.4815
半径の長さ111.8034
面積39269.91
円周の長さ702.4815
```

A screenshot of a software window titled "sketch\_180603a". The window contains a white canvas with a black circle drawn on it. A mouse cursor is positioned on the right side of the circle's circumference. The window has standard OS window controls (minimize, maximize, close) in the top right corner.

# プログラミング演習I (第3回) 課題

## • 基本課題③ アルキメデスの螺旋: graph

–  $x$  と  $y$  の座標が  $\theta$  (theta) の値【度】によって変化する下記の数式の計算結果の座標( $x, y$ )をpointを利用してアルキメデスの螺旋を描け. ただし, theta は draw() 毎に1ずつ増加するようにせよ. またウィンドウサイズは800x800とせよ. thetaをcos, sinの中で使う場合はradiansで変換すること

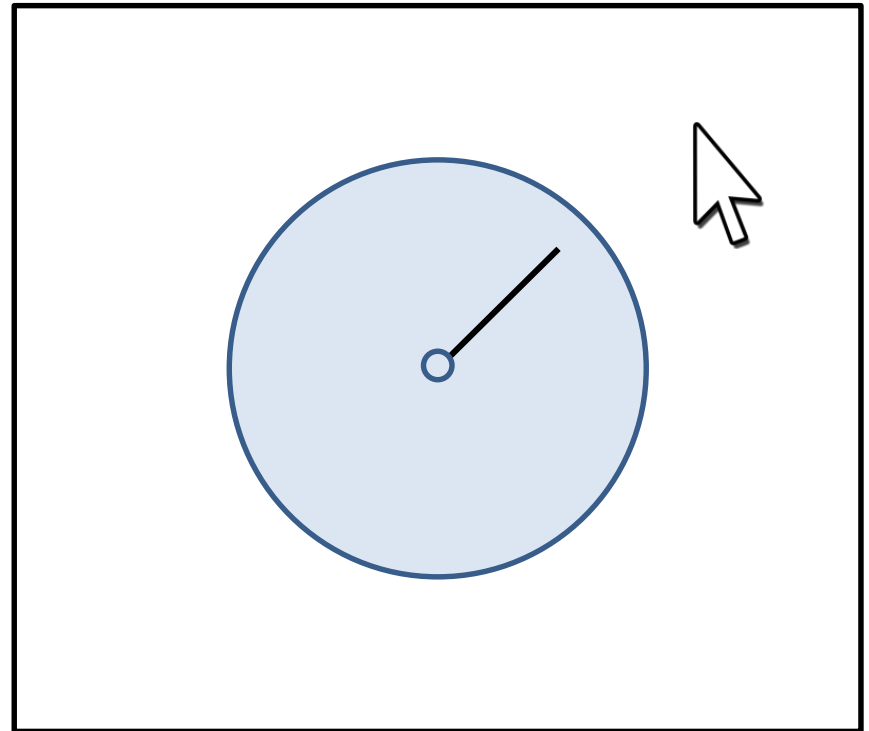
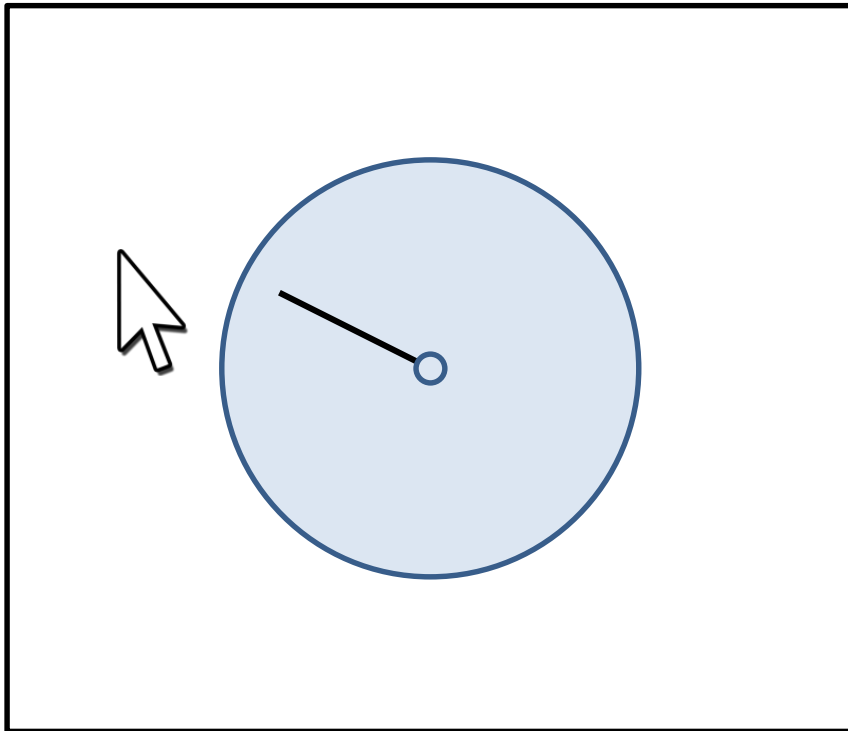
- $x = a\theta\cos\theta + b$
- $y = a\theta\sin\theta + c$
- $a = 0.1, b = 400, c = 400$  とする

```
int theta = 0;
void draw(){
    x,yを定義
    x,yの計算
    point( x, y );
    theta = theta + 1;
}
```

# プログラミング演習I (第3回) 課題

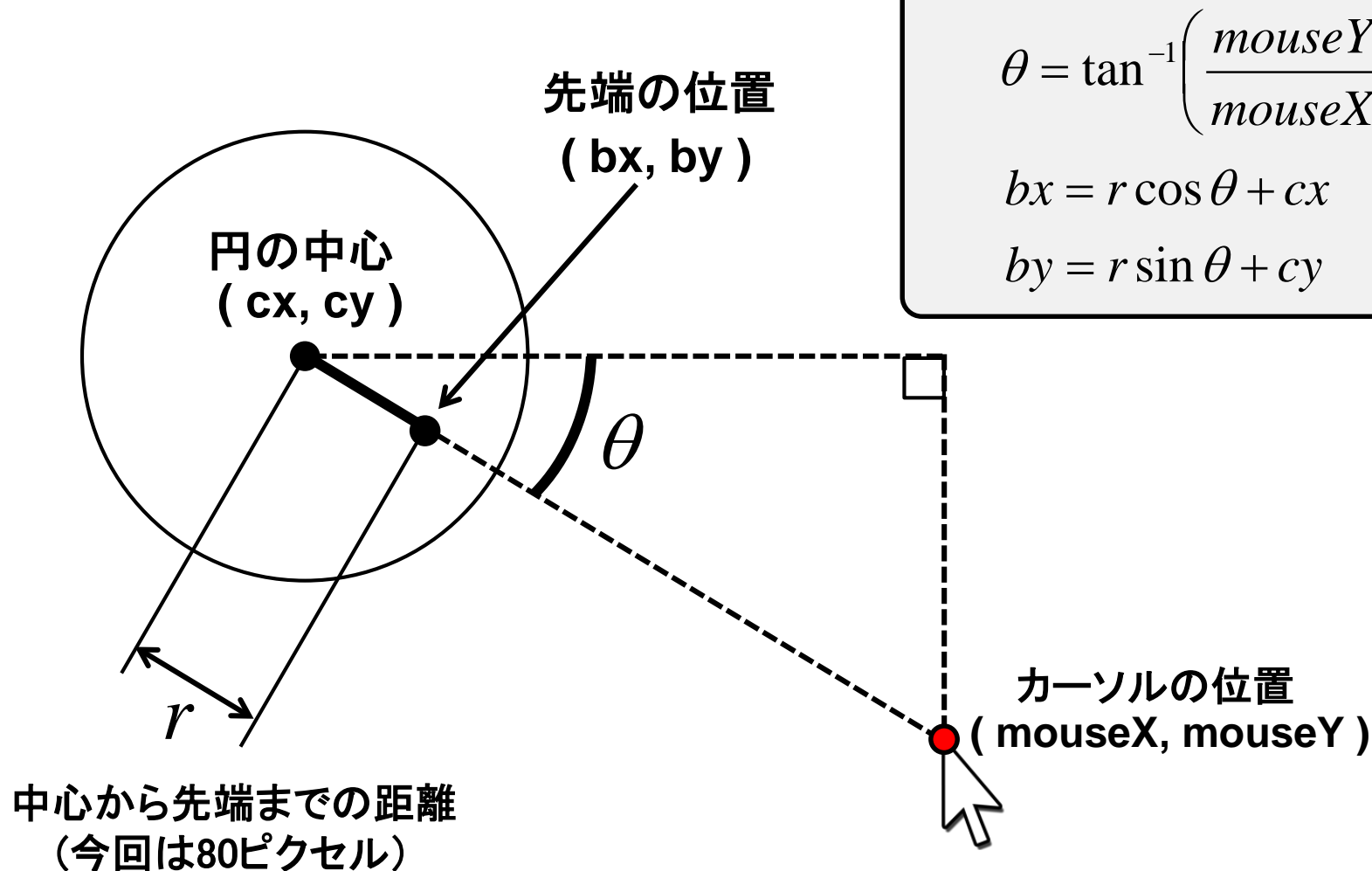
## • 発展課題① スケッチ名: **direction**

- 下記のように, 400x300のウインドウの中に直径200の大きさの円を描き, その円の中心からマウスカーソルの方向まで半径80の直線を描き, そちらを示すようにせよ



# プログラミング演習I (第3回) 課題

## 発展課題①のヒント



黒目の位置を求めるための数式

$$\theta = \tan^{-1} \left( \frac{mouseY - cy}{mouseX - cx} \right)$$

$$bx = r \cos \theta + cx$$

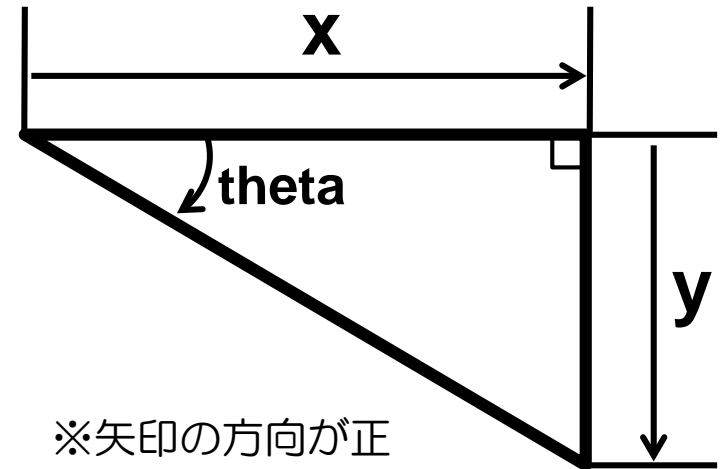
$$by = r \sin \theta + cy$$

# プログラミング演習I (第3回) 課題

- $\tan^{-1}$  の求め方は2つある

アークタンジェントの計算には `atan()` と `atan2()` があり、それぞれ値域が異なります。なお、いずれも計算結果は実数値(float)です。

今回の課題では `atan2()` を使うとよいでしょう (`atan` の場合は問題が発生するが理由はわかるかな?)



**`theta = atan( y/x );`**

値域は  $-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$

**`theta = atan2( y, x );`**

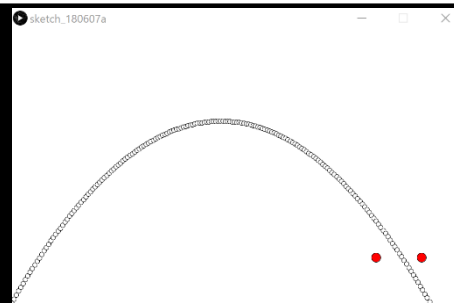
値域は  $-\pi \leq \theta \leq \pi$

# プログラミング演習I (第3回) 課題

## ● 発展課題② スケッチ名: ball

- 地上で斜め上方向に白色のボールを投げたときの様子をシミュレーションするプログラムを作成してください。
- ただし、射出角度は60度とし、800m先と900m先に100mの高さの赤色のターゲットの間を通すように投射速度 ( $v_0$ ) を調整せよ。なお下記の setup を利用せよ

```
void setup()
{
  size( 1000, 600 );
  background( 255 );
  fill( 255, 0, 0 );
  ellipse( 800, height-100, 20, 20 );
  ellipse( 900, height-100, 20, 20 );
  fill( 255 );
}
```



ヒント: 斜方投射の式

$$x = v_0 t \cos \theta$$

$$y = v_0 t \sin \theta - \frac{1}{2} g t^2$$

$v_0$  は初速 (投射速度)  
 $g$  は重力加速度 (9.8)  
 $t$  は経過時間となる

※1フレームの経過時間は0.1秒とせよ。また、1ピクセル=1mと考えてください。