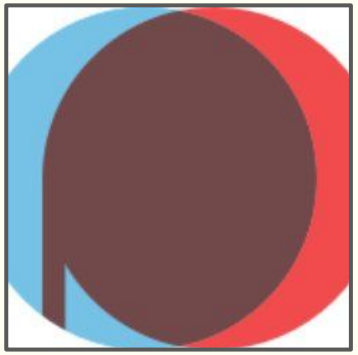
OpenProcessing体験

OpenProcessingって何？



OpenProcessingは、

ブラウザ上で簡単に

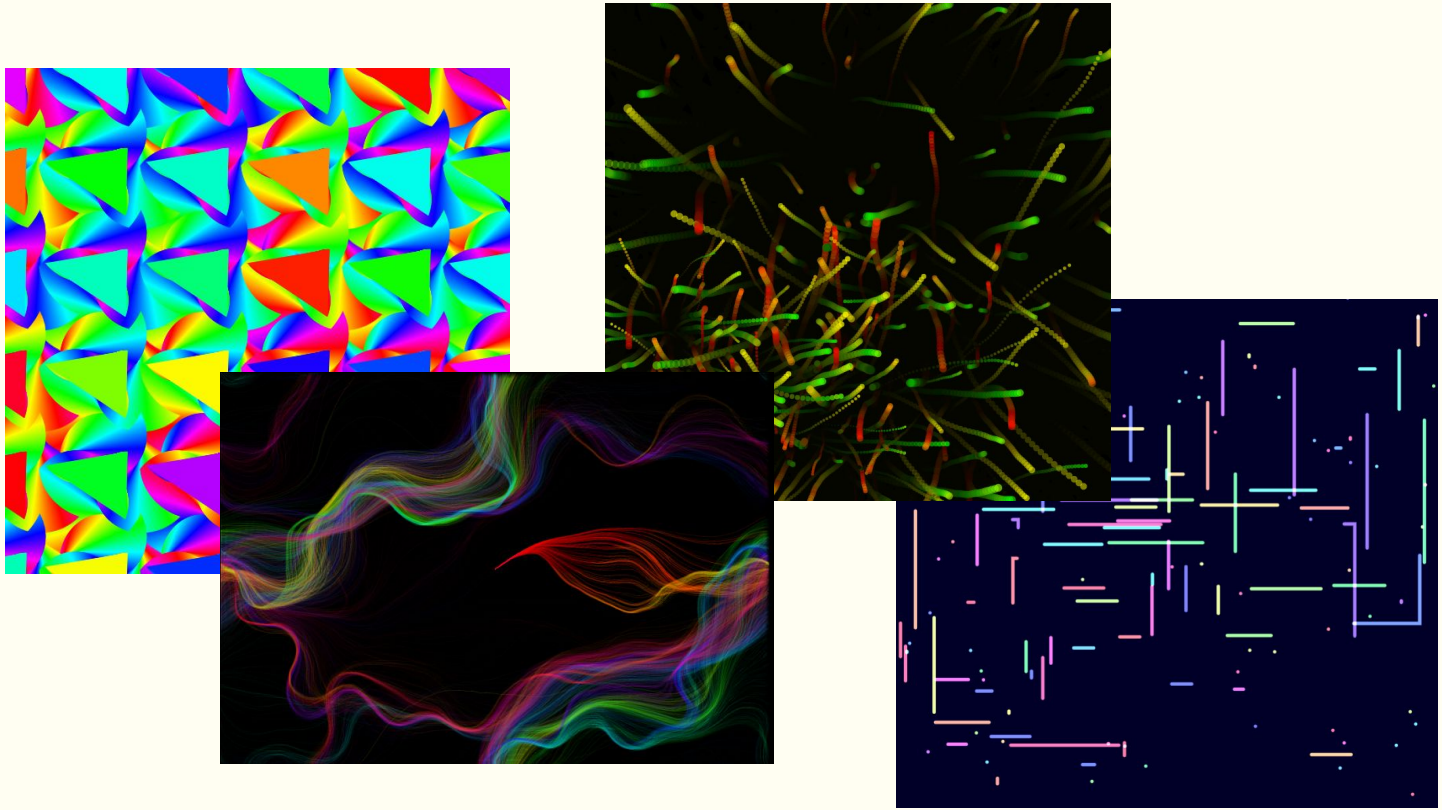
作る

遊ぶ

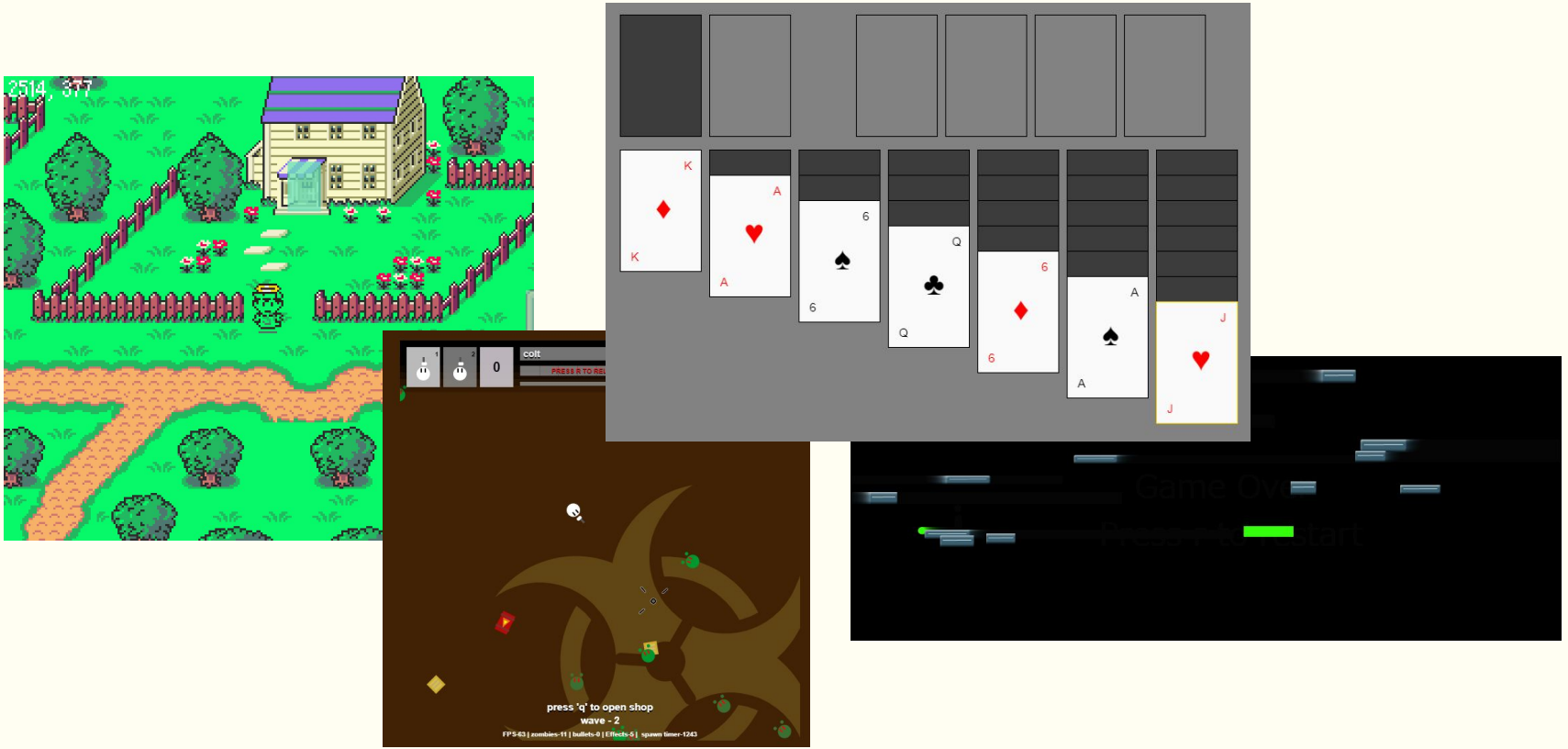
見せる

ことができるプログラミング環境です

こんな感じのデジタルアートや



こんな感じのゲームが作れちゃいます



なんと環境構築

必要なし!!

※メアドでアカウントを作る手間はあります

携帯でも、

作品を遊ぶ

作品の共有

コーディング ※

ができちゃいます！

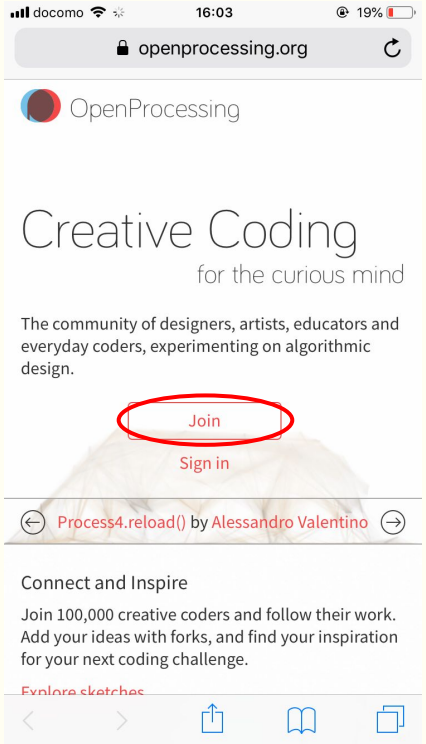
※携帯のキーボードはコーディングに向いてない

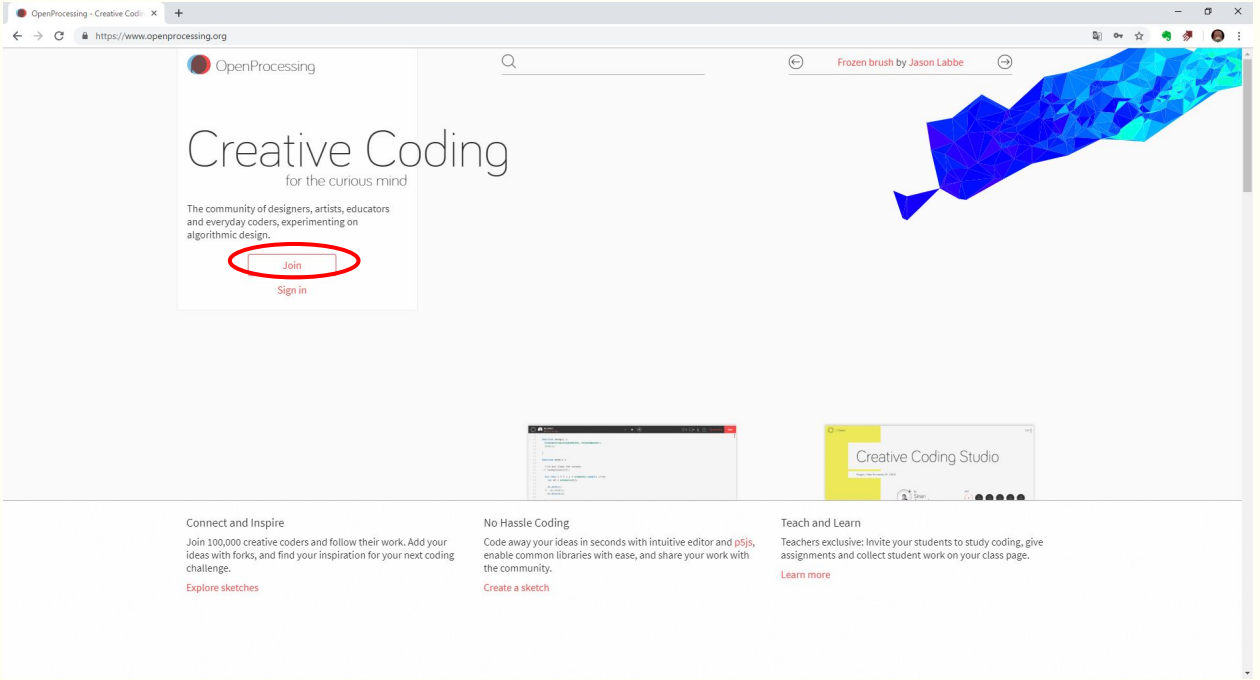
OpenProcessingの始め方

まずは以下のURLにアクセス

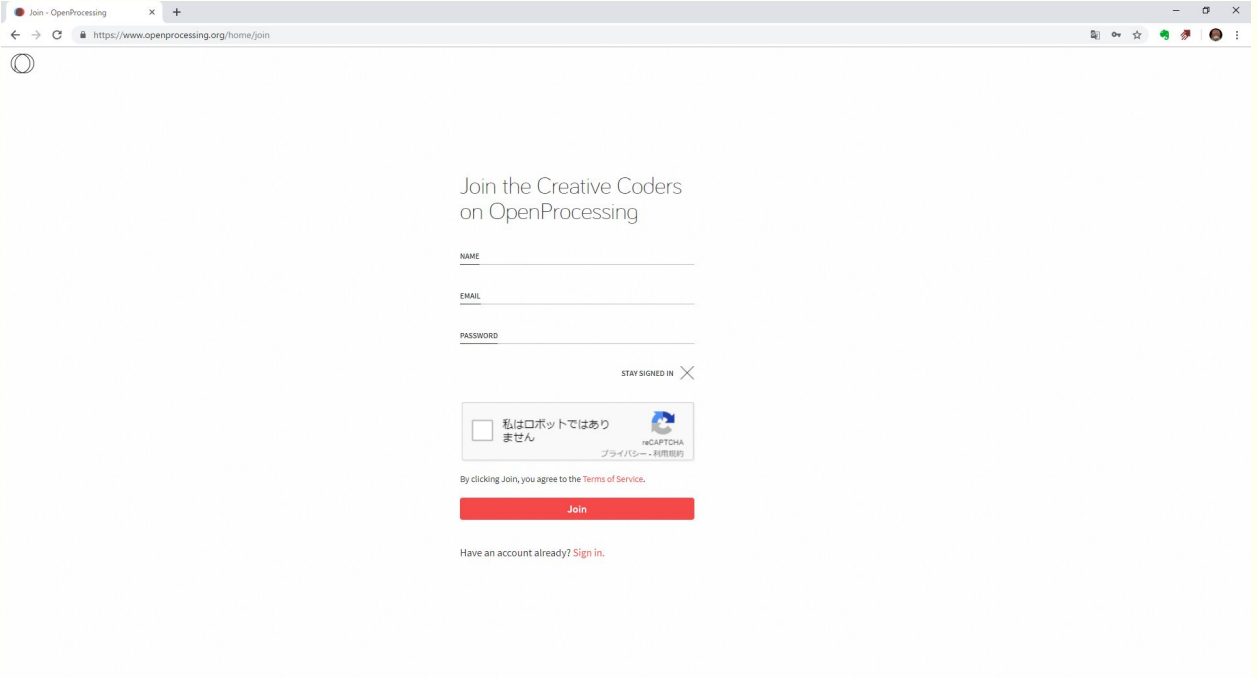
https://www.openprocessing.org

Joinを押して登録開始

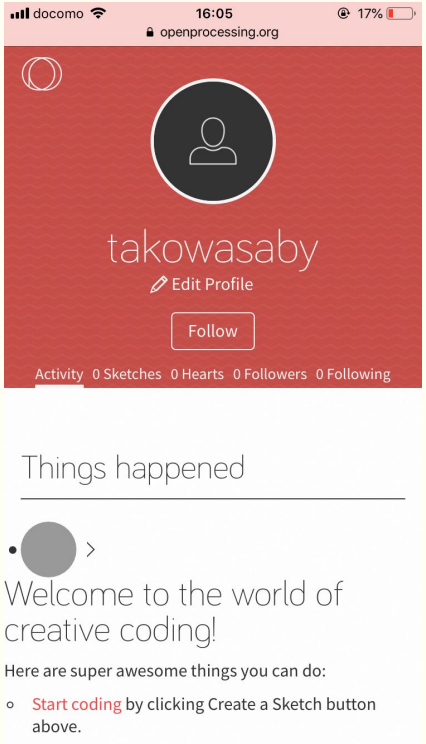


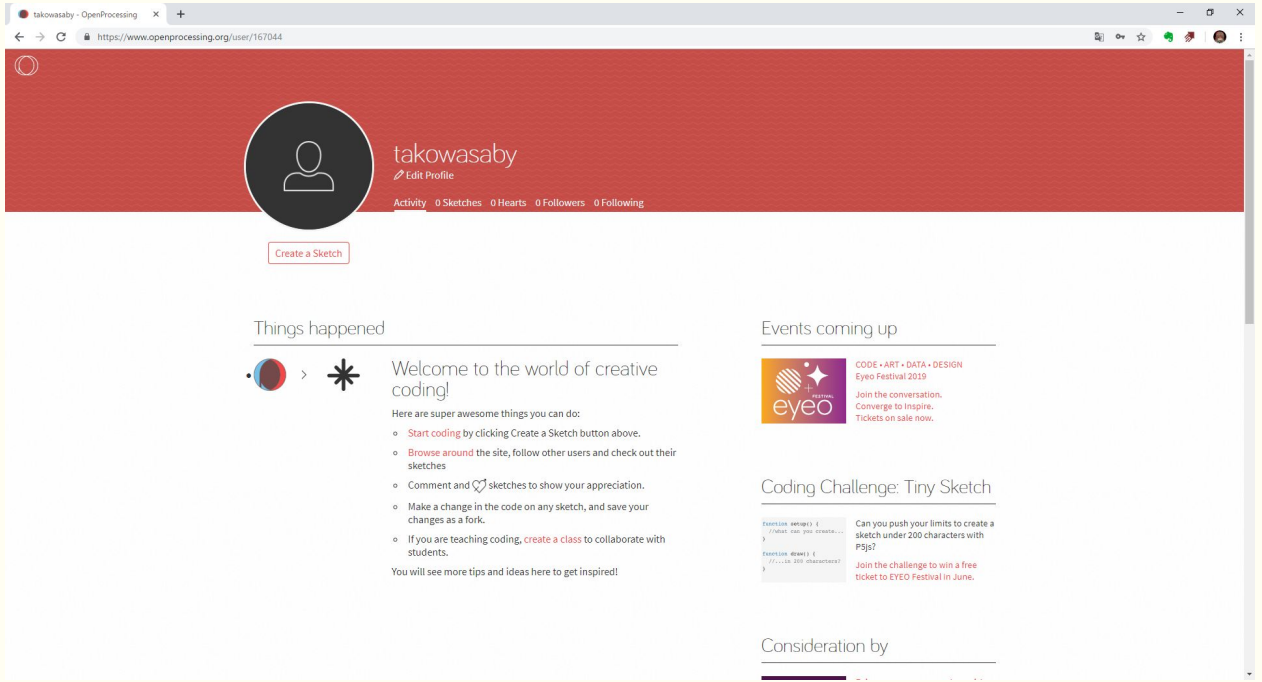


必要事項を入力して登録



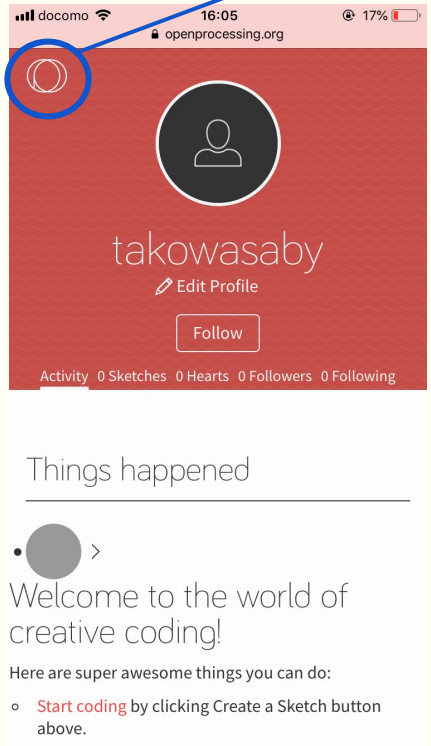
登録したらマイページに到着

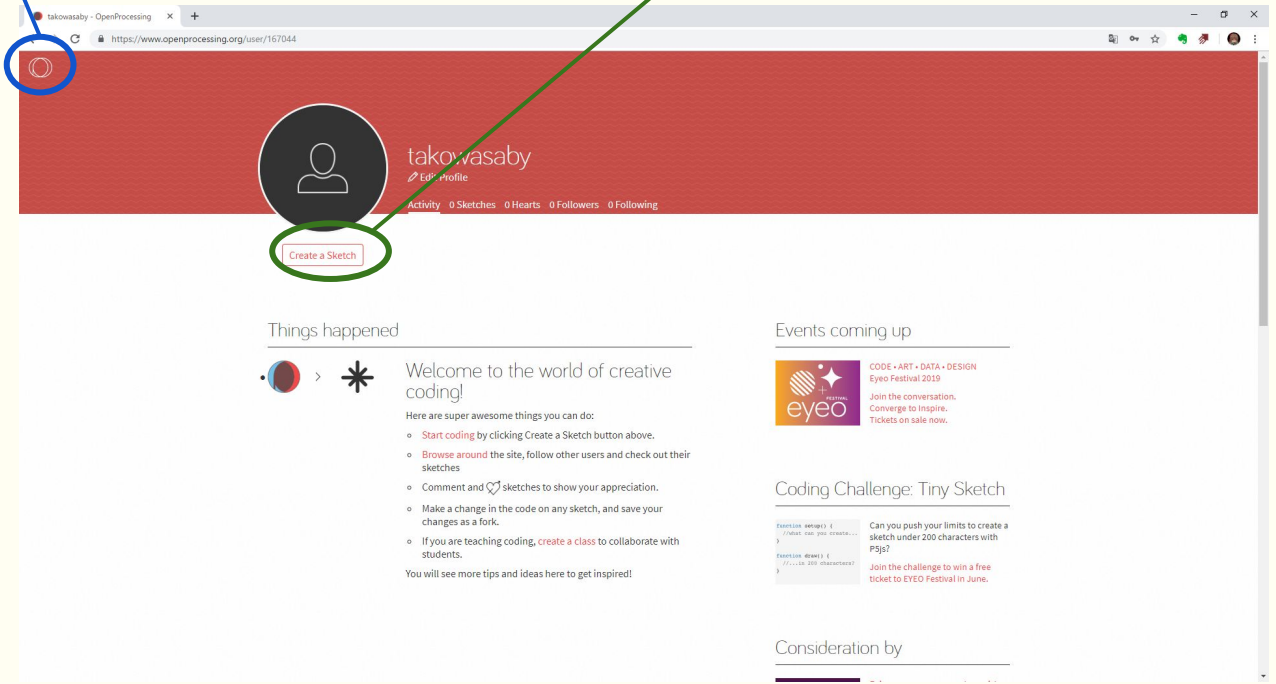




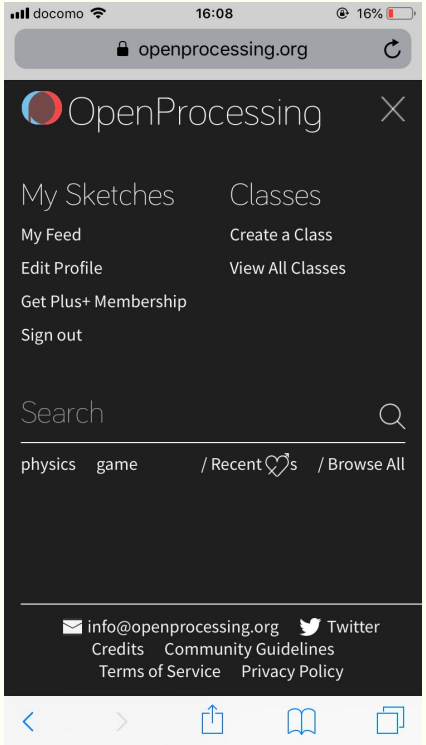
左上アイコンでメニューへ

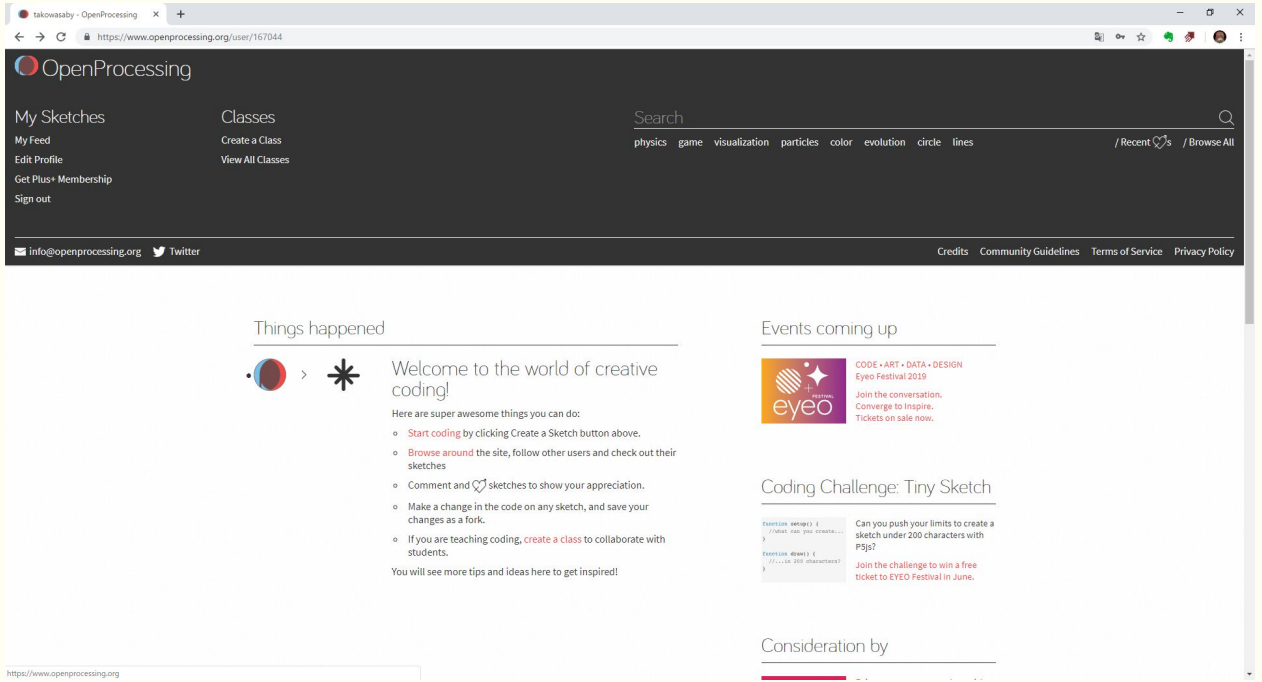
Create a Sketchで作品を作る



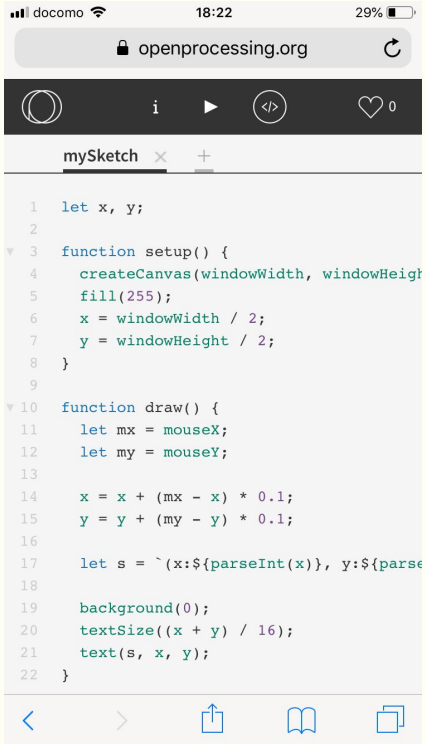


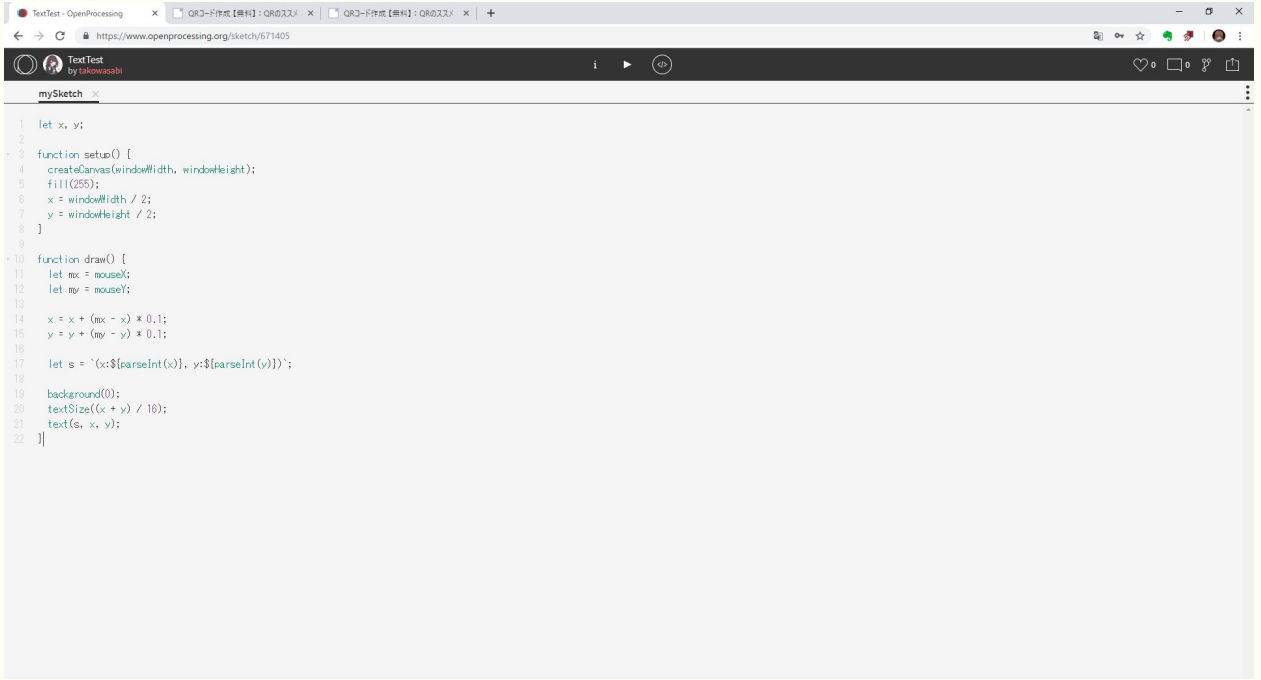
メニューからは作品の検索ができます



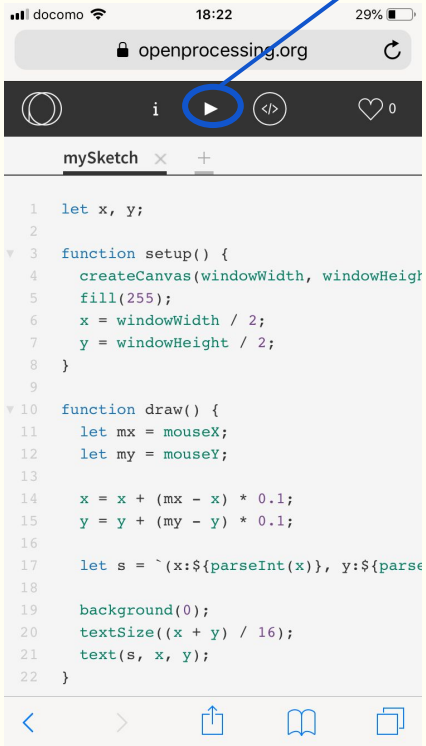


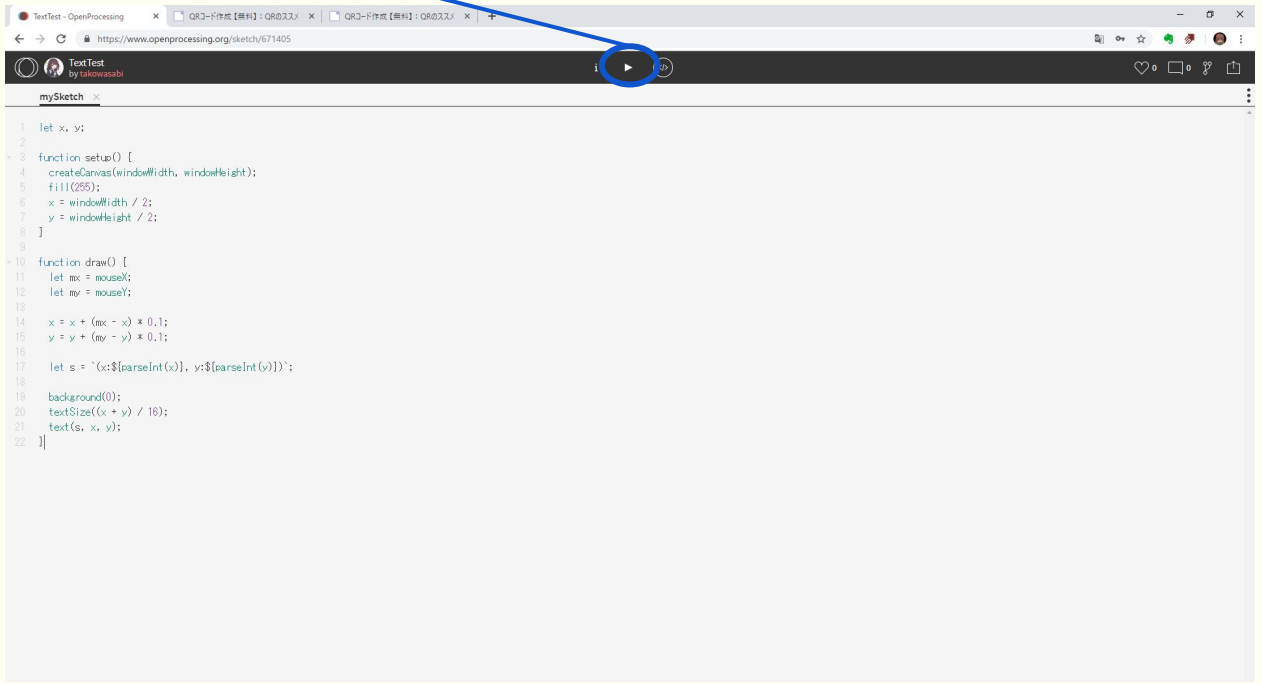
作品の編集画面はこんな感じ





再生ボタンで作品を見れる



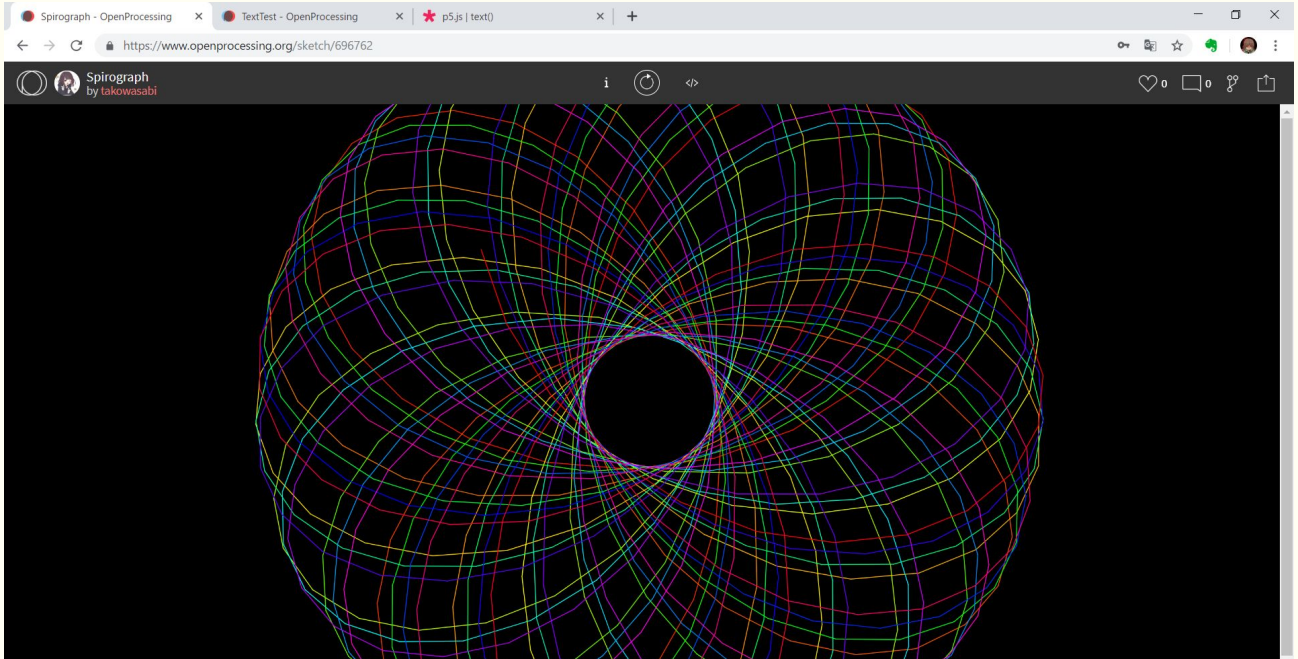


作品制作開始

本日の作品

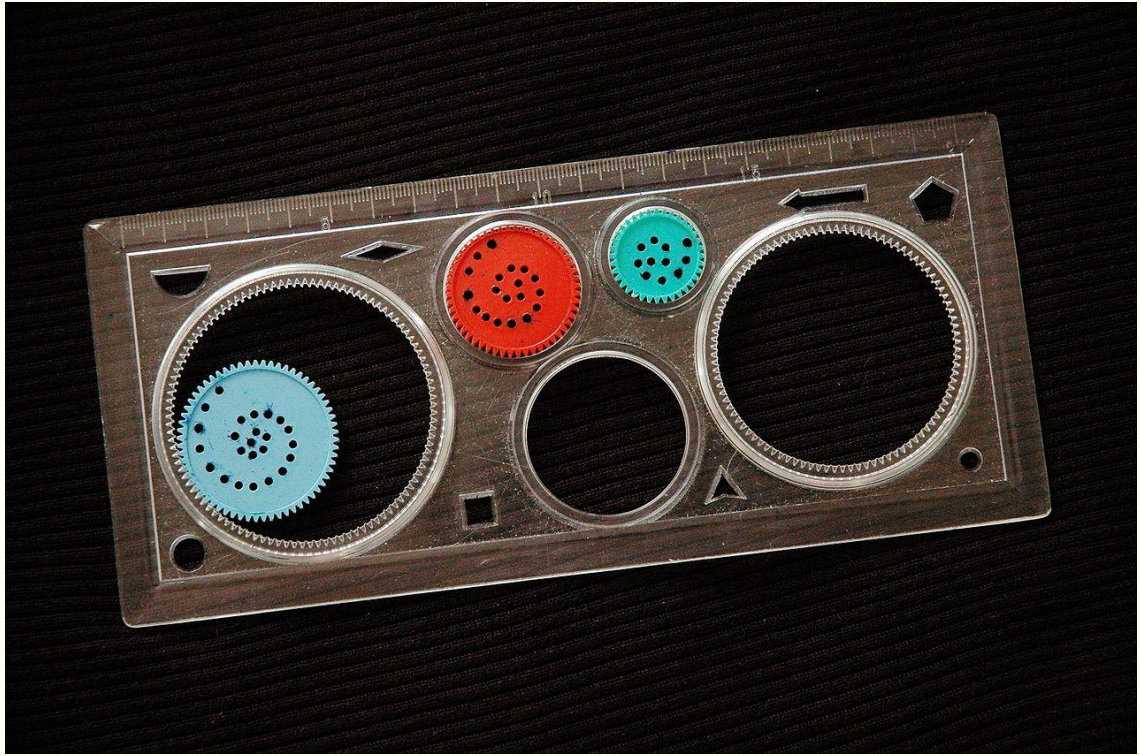
本日の作品

スピログラフ（Spirograph）



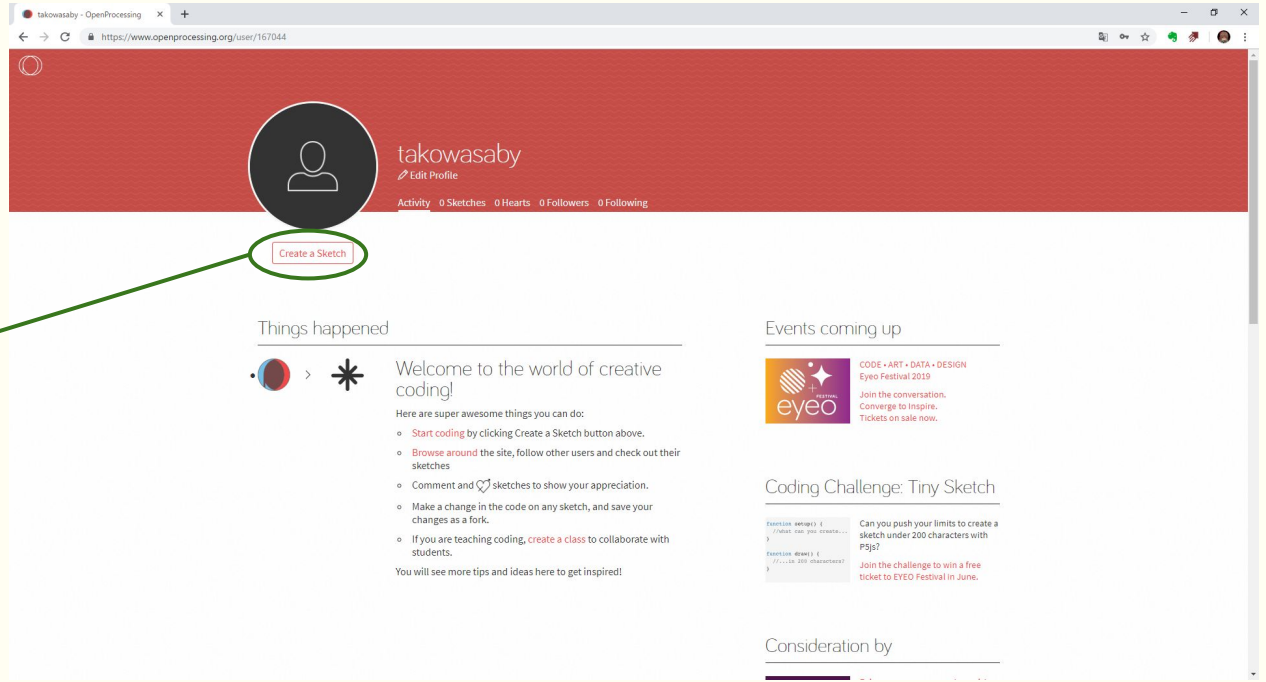
スピログラフって何？

スピログラフって何？



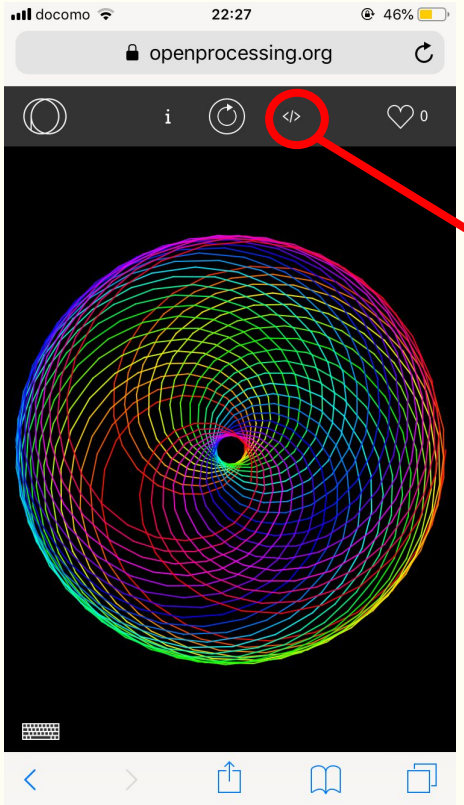
スケッチブックを作ろう

まずは新規の
スケッチブック（作品）
を作成



携帯の方はコードを追っていきましょう

https://www.openprocessing.org/sketch/696762



ここから
コードが見れます

どうやって再現するの？

プログラムで再現したい

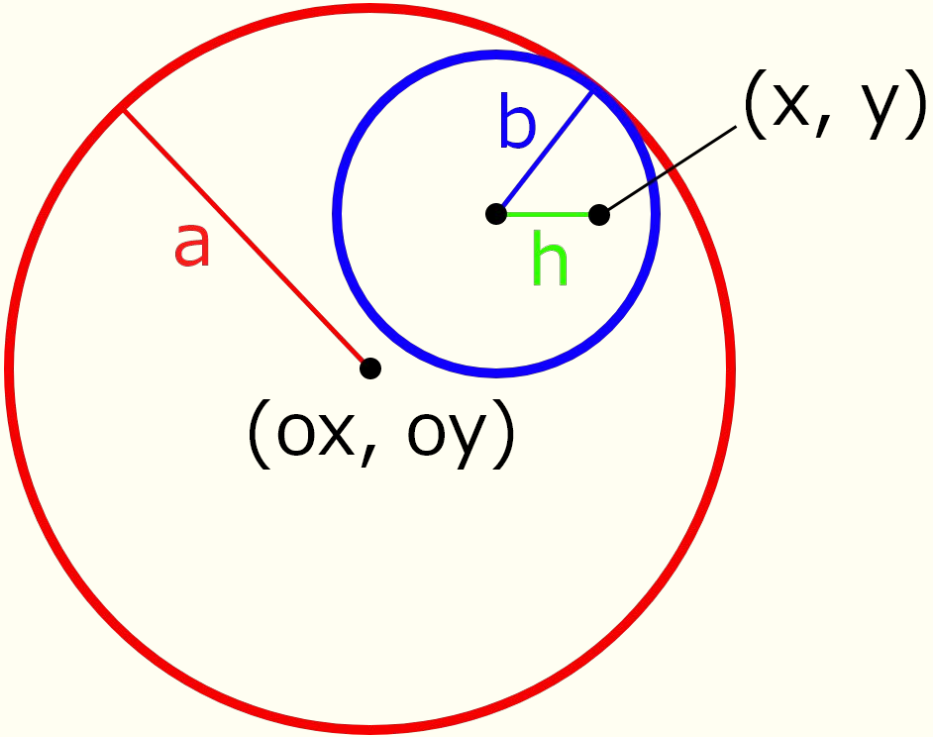
↓

数式があるとプログラムが書きやすい

↓

数式を立てます

まずは変数定義



数式はこんな感じ

$$\begin{cases} x = (a-b)cos\theta - h\cos\left(\dfrac{a+b}{b}\theta\right) \\ y = (a-b)sin\theta - h\sin\left(\dfrac{a+b}{b}\theta\right) \end{cases}$$

プログラム書きます

この状態がデフォルト

```
function setup() {
    createCanvas(windowWidth, windowHeight);
    background(100);
}

function draw() {
    ellipse(mouseX, mouseY, 20, 20);
}
```

プログラム書きます

この状態がデフォルト

```
function setup() {
    createCanvas(windowWidth, windowHeight);
    background(100);
}
```

始めにやる処理

```
function draw() {
    ellipse(mouseX, mouseY, 20, 20);
}
```

描画の処理

要らない部分は消しましょう

```
function setup() {
    createCanvas(windowWidth, windowHeight);
    background(100);
}

function draw() {
    ellipse(mouseX, mouseY, 20, 20);
}
```

こうなればOK

```
function setup() {
    createCanvas(windowWidth, windowHeight);
    background(100);
}

function draw() {
}
```

プログラムでも変数を作ります

```
let a = 0.0;

function setup() {
    createCanvas(windowWidth, windowHeight);
    background(100);
}

function draw() {
    a = mouseX;
}
```

変数の定義（JavaScript）

```
let a = 0.0;
```

変数　a　は　0.0　です

変数への値の変更（JavaScript）

```
a = mouseX;
```

a　は　mouseX(変数)と同じ値 です

mouseX:マウスのX座標を表す

他の変数も同じ場所で定義

```
let a = 0.0;
let b = 0.0;
let h = 0.0;
let x1 = 0.0;
let y1 = 0.0;
let x2 = 0.0;
let y2 = 0.0;
let theta1 = 0.0;
let theta2 = 0.0;
…
```

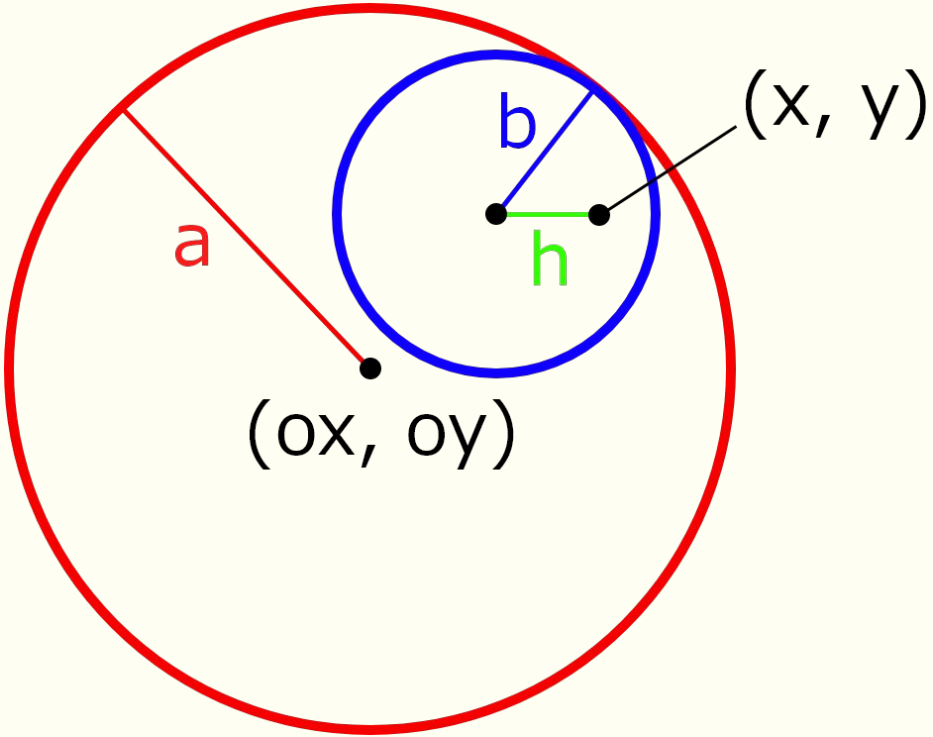
a, b, hの値を設定

```
…
function draw() {
    a = mouseX;
    b = windowWidth / 2;
    h = mouseY;
}
```

mouseY:マウスのY座標を表す

windowWidth:画面の横幅

変数の意味を再確認



曲線の描画

プログラムで曲線を描くのは大変

↓

たくさん線分を書いて曲線に近づける

↓

少しずつθをずらして線分を引いていく

たくさん線分を描くために

プログラムでは、たくさんの処理を

繰り返し行うときに for文を使う

```
for (let i = 0; i < 1024; i++){}
```

部分ごとの意味

`i`：繰り返した回数を数えるカウンタ

```
for (let i = 0; i < 1024; i++)
```

カウンタを用意　カウンタが1024以下なら繰り返す

正の方向にカウントしていく

たくさん線分を描くためには

```
…
function draw() {
    a = mouseX;
    b = ox;
    h = mouseY;
    for (let i = 0; i < 1024; i++)
    {
        ここに、少しずつ線分を描く処理を書く
    }
}
```

i を用いてθ1を決定し、ずらしたものをθ2とする

```
theta1 = (i / 32) * 3 * PI;
theta2 = ((i + 1) / 32) * 3 * PI;
```

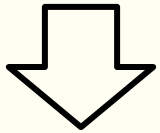
ＰＩ:いわゆるπを表す

i を用いてθ1を決定し、ずらしたものをθ2とする

```
…
    for (let i = 0; i < 1024; i++)
    {
        theta1 = (i / 32) * 3 * PI;
        theta2 = ((i + 1) / 32) * 3 * PI;
    }
…
```

θをもとにx, yを求める

$$\begin{cases} x = (a-b)cos\theta - h\cos\left(\dfrac{a+b}{b}\theta\right) \\ y = (a-b)sin\theta - h\sin\left(\dfrac{a+b}{b}\theta\right) \end{cases}$$



```
x = (a - b) * cos(theta1) + h * cos(((a - b) / b) * theta);
y = (a - b) * sin(theta1) + h * sin(((a - b) / b) * theta);
```

θ1, θ2でそれぞれ(x1, y1), (x2, y2)を求める

```
…
    for (let i = 0; i < 1024; i++)
    {
        theta1 = (i / 32) * 3 * PI;
        theta2 = ((i + 1) / 32) * 3 * PI;

        x1 = (a - b) * cos(theta1) + h * cos(((a - b) / b) * theta1);
        x2 = (a - b) * cos(theta2) + h * cos(((a - b) / b) * theta2);

        y1 = (a - b) * sin(theta1) + h * sin(((a - b) / b) * theta1);
        y2 = (a - b) * sin(theta2) + h * sin(((a - b) / b) * theta2);
    }
…
```

線分を描くためには

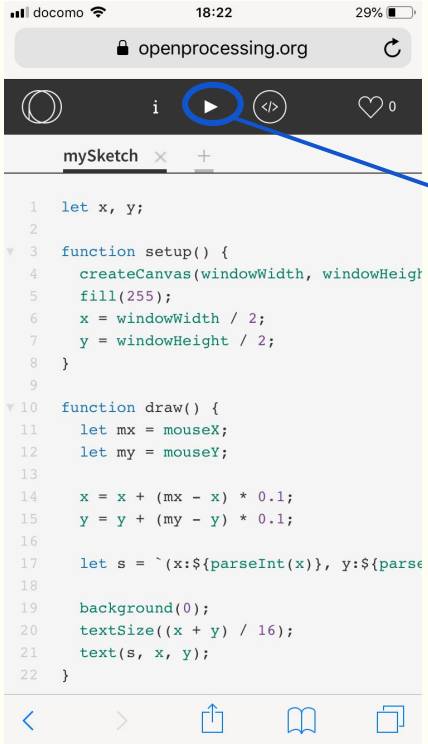
```
line(x1, y1, x2, y2);
```

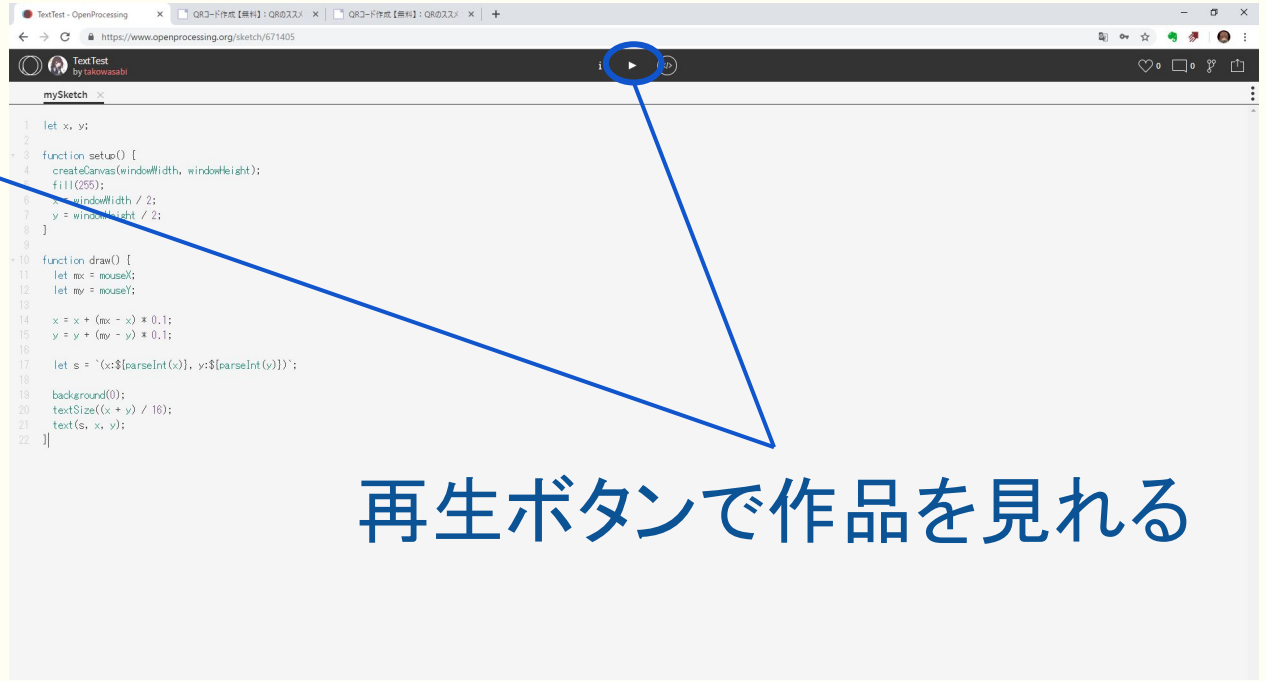
座標(x1, y1)の点と座標(x2, y2)の点の間に線を引く

線分を描く部分を追加

```
…
    for (let i = 0; i < 1024; i++)
    {
        theta1 = (i / 32) * 3 * PI;
        theta2 = ((i + 1) / 32) * 3 * PI;

        x1 = (a - b) * cos(theta1) + h * cos(((a - b) / b) * theta1);
        x2 = (a - b) * cos(theta2) + h * cos(((a - b) / b) * theta2);

        y1 = (a - b) * sin(theta1) + h * sin(((a - b) / b) * theta1);
        y2 = (a - b) * sin(theta2) + h * sin(((a - b) / b) * theta2);

        line(x1, y1, x2, y2);
    }
…
```

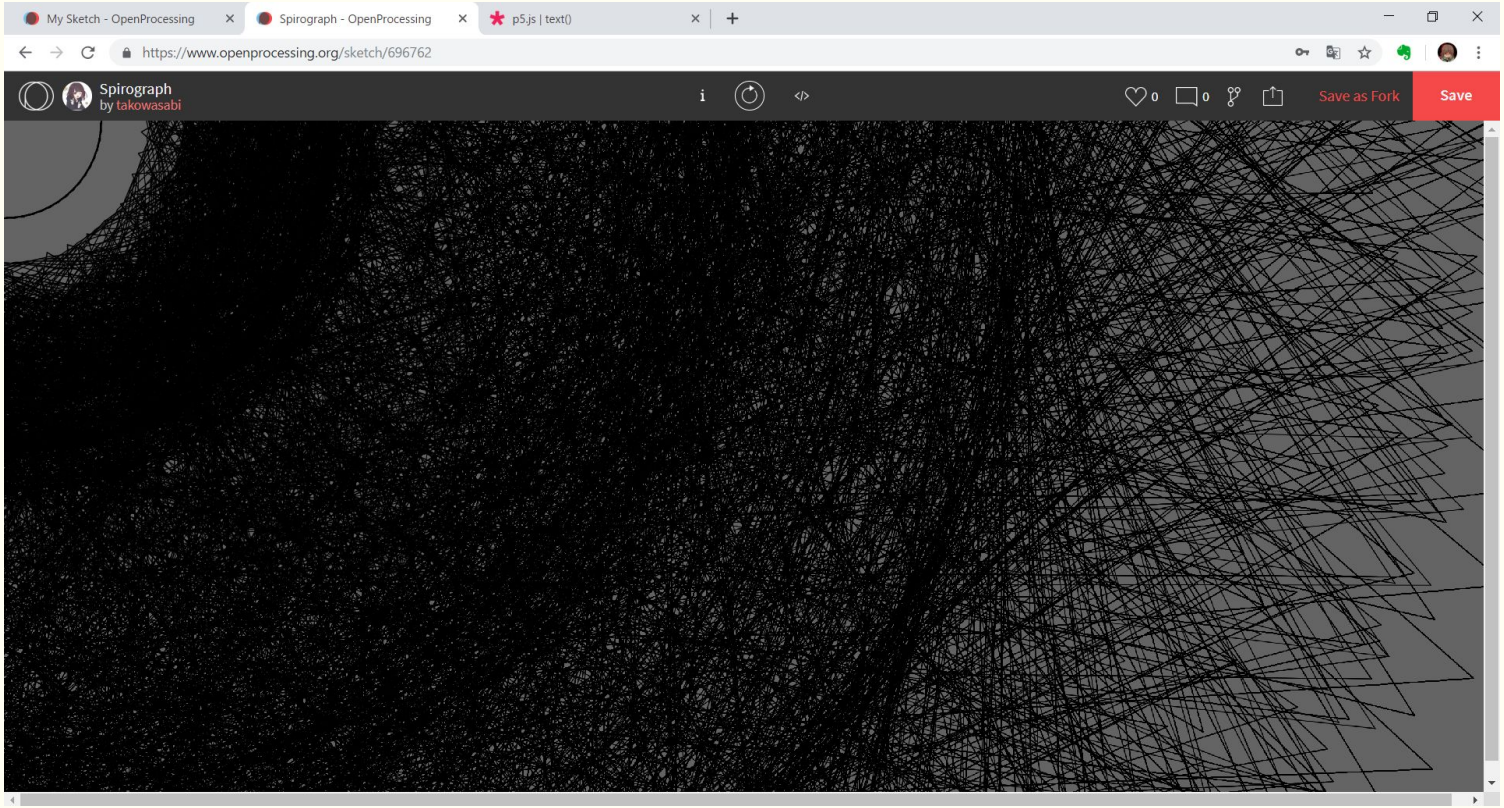
ちょっと動かしてみよう





再生ボタンで作品を見れる

こんな感じになるはず


My Sketch - OpenProcessing
Spirograph - OpenProcessing
p5.js | text()
https://www.openprocessing.org/sketch/696762
Spirograph
by takowasabi
Save as Fork
Save

完成版と何が違うのか？

- 左上が中心になってしまっている
- 書いた線が消えない
- 色がついていない

完成版と何が違うのか？

- 左上が中心になってしまっている
- 書いた線が消えない
- 色がついていない

processingでの座標

実はprocessingでは原点が左上にある
このため、中心に動かす場合は描画するときにに原点を移動する必要がある

画面の真ん中を原点とする

```
…
let theta2 = 0.0;
let ox;
let oy;

function setup() {
…


…
function draw() {
     ox = windowWidth / 2;
     oy = windowHeight / 2;
…
```

描画の時に原点の分ずらす

```
line(x1, y1, x2, y2);
```

↓

```
line(ox + x1, oy + y1, ox + x2, oy + y2);
```

完成版と何が違うのか？

- 左上が中心になってしまっている
- 書いた線が消えない
- 色がついていない

何故線がどんどん増えていくのか？

マウスの座標が変わるたびに

キャンバスの上に線を何度も描いている感じ

↓

線を描くたびにキャンバスを綺麗にする

background(0)で背景を黒く塗りつぶす

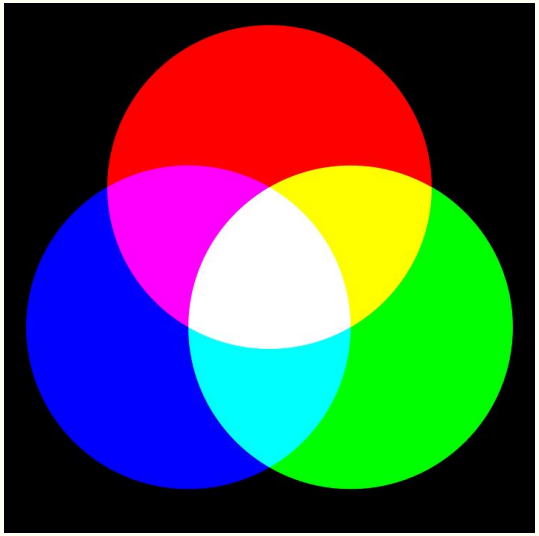
```
…
function draw() {
   background(0);

   a = mouseX;
…
```

完成版と何が違うのか？

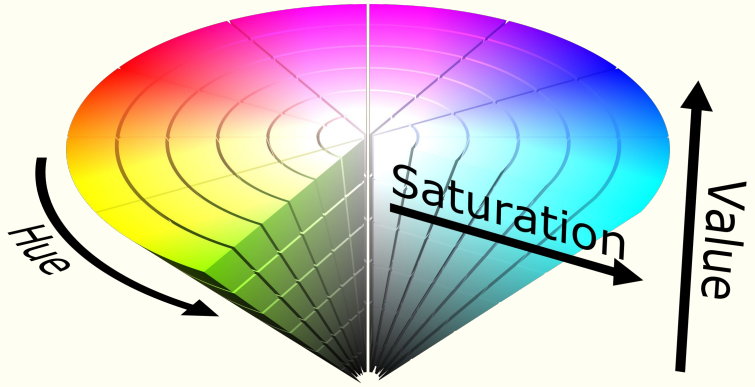
- 左上が中心になってしまっている
- 書いた線が消えない
- 色がついていない

色の表し方

RGB



HSB

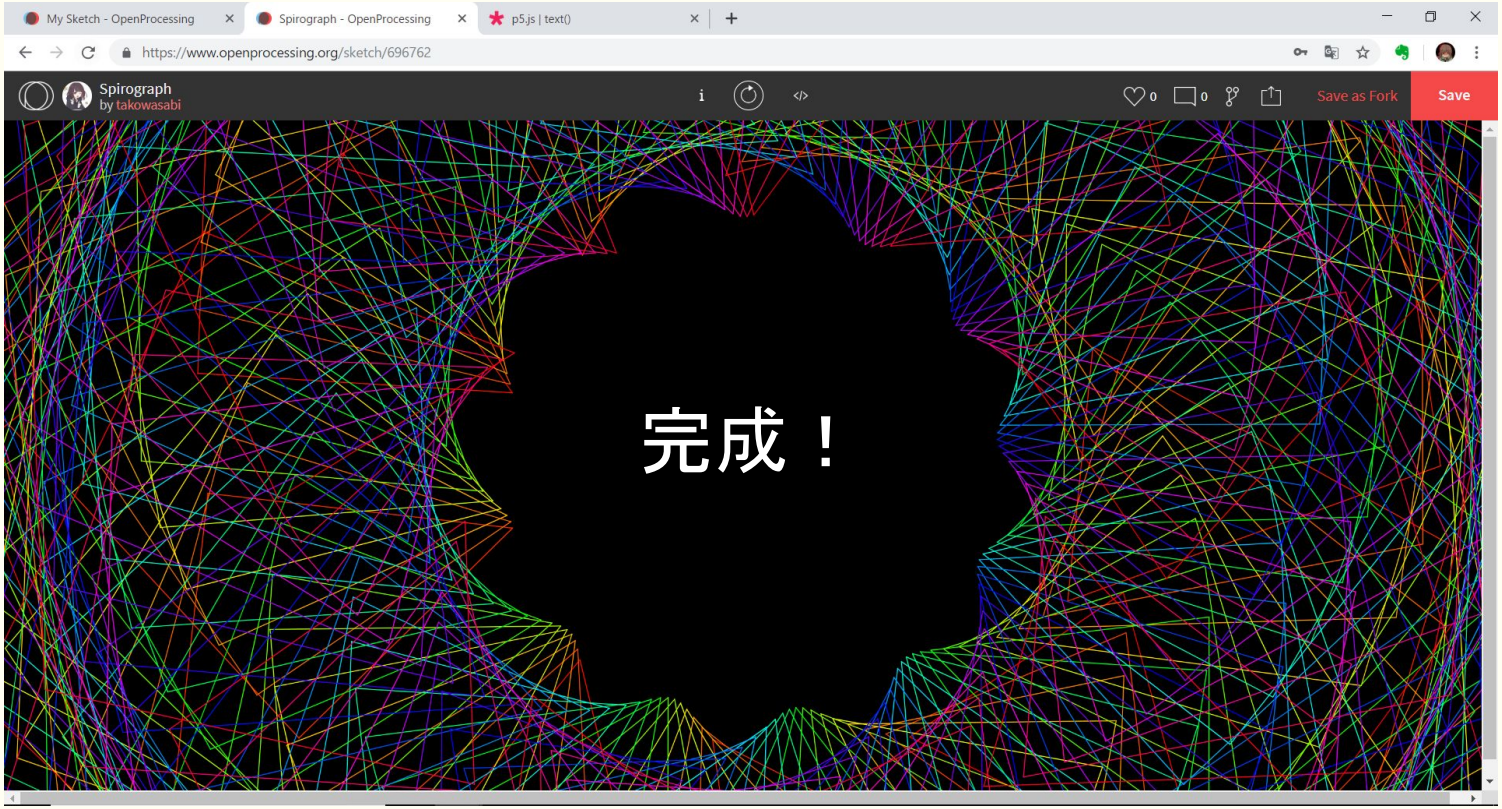


色の指定の形式を変更

```
…
function setup() {
   createCanvas(windowWidth, windowHeight);
   colorMode(HSB, 1023);
}
…
```

描画の前にStrokeで色を指定

```
…
    stroke(i, 1023, 1023);
    line(ox + x1, oy + y1, ox + x2, oy + y2);
…
```


My Sketch - OpenProcessing
Spirograph - OpenProcessing
p5.js | text()
https://www.openprocessing.org/sketch/696762
Spirograph
by takowasabi
0
0
Save as Fork
Save
完成！

おまけ

さらに詳しく知りたい方は

以下のリファレンスを参照してください

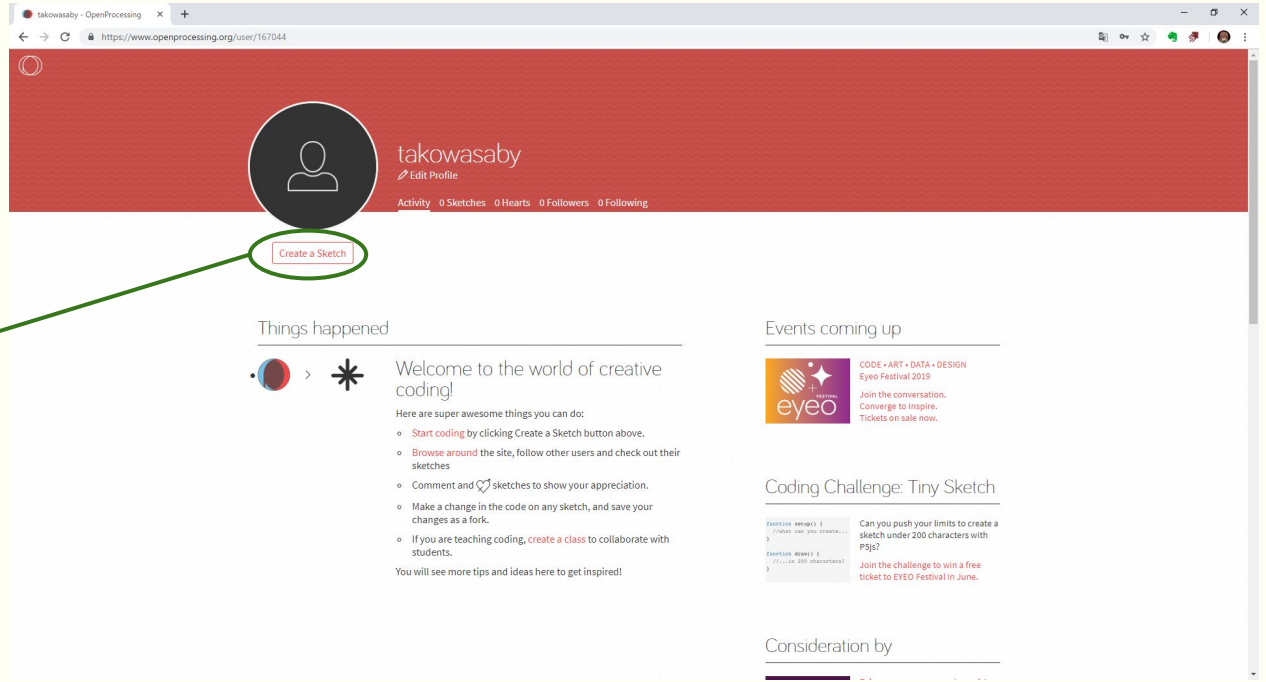
https://p5js.org/reference/

※リファレンスにあるサンプルコードはなんと

書き換え可能です、リファレンス見てるだけで楽しい！

実際に作品をアップロードしてみよう in PC

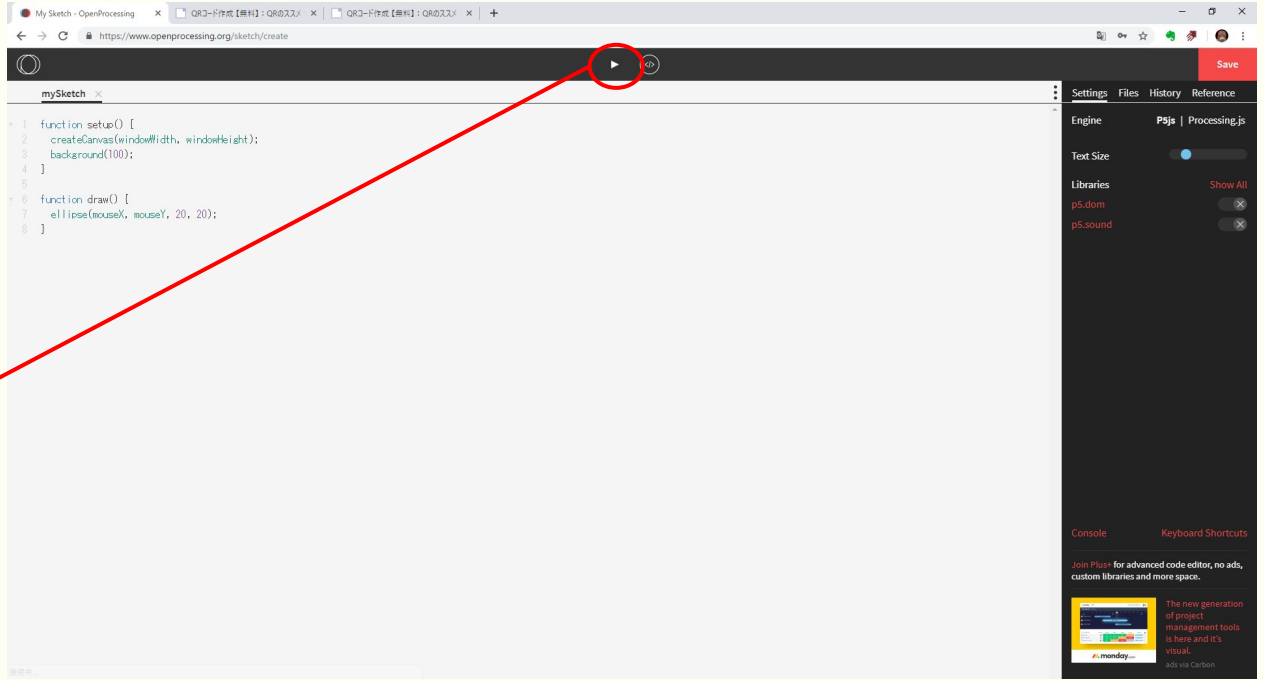
まずは新規の
スケッチブック（作品）
を作成



実際に作品をアップロードしてみよう in PC

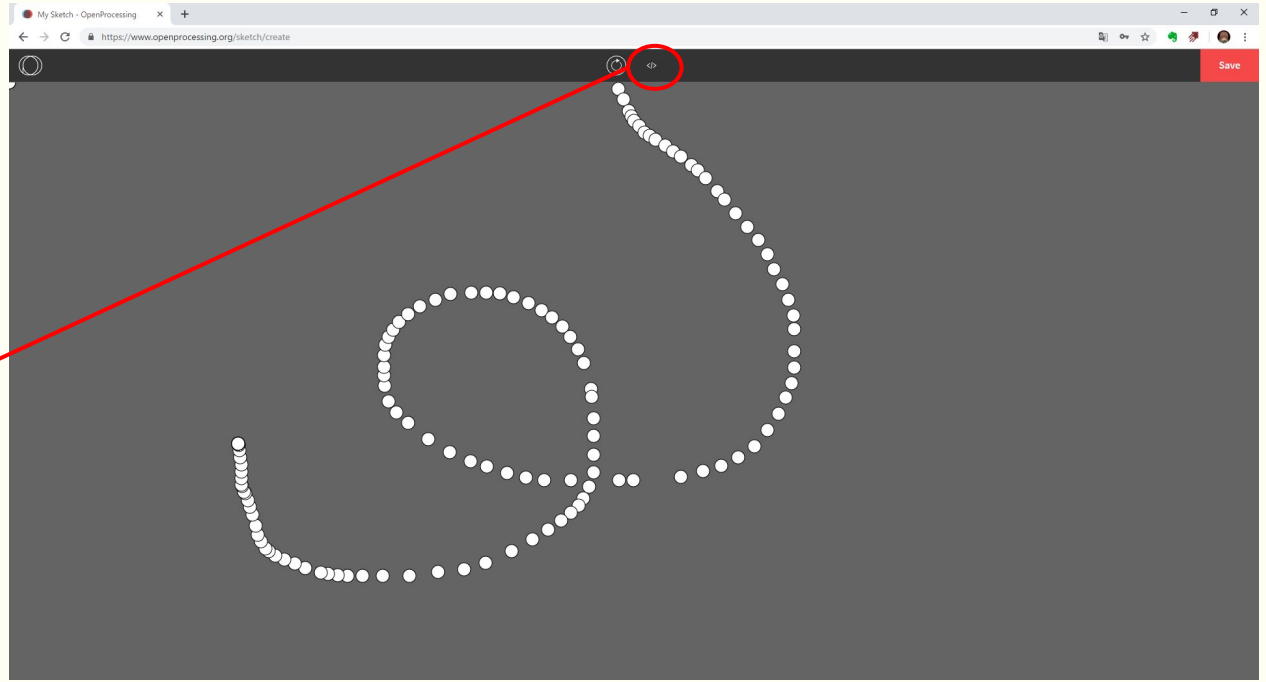
スケッチブックを
作成すると、
すでにデフォルトの
コードが書かれています

再生マークで
挙動を確認しよう



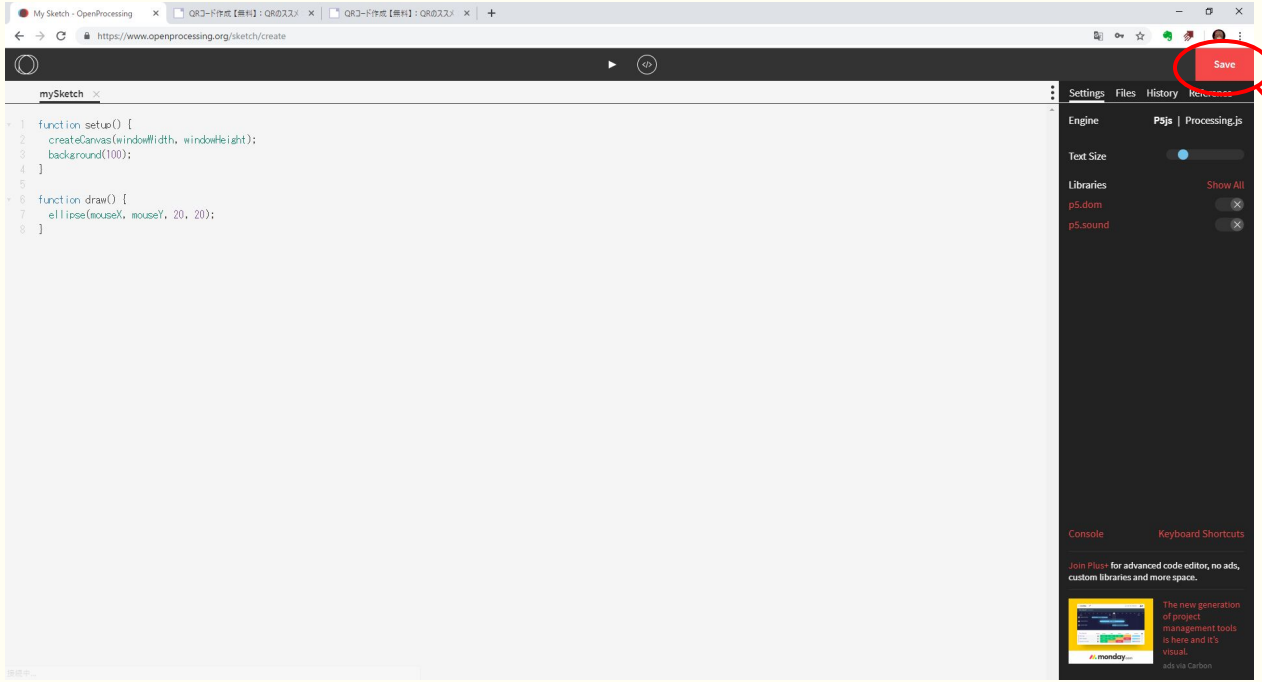
実際に作品をアップロードしてみよう in PC

マウスの動きに合わせて
白丸が出来ます



確認したら
</>マークで
編集画面に戻ろう

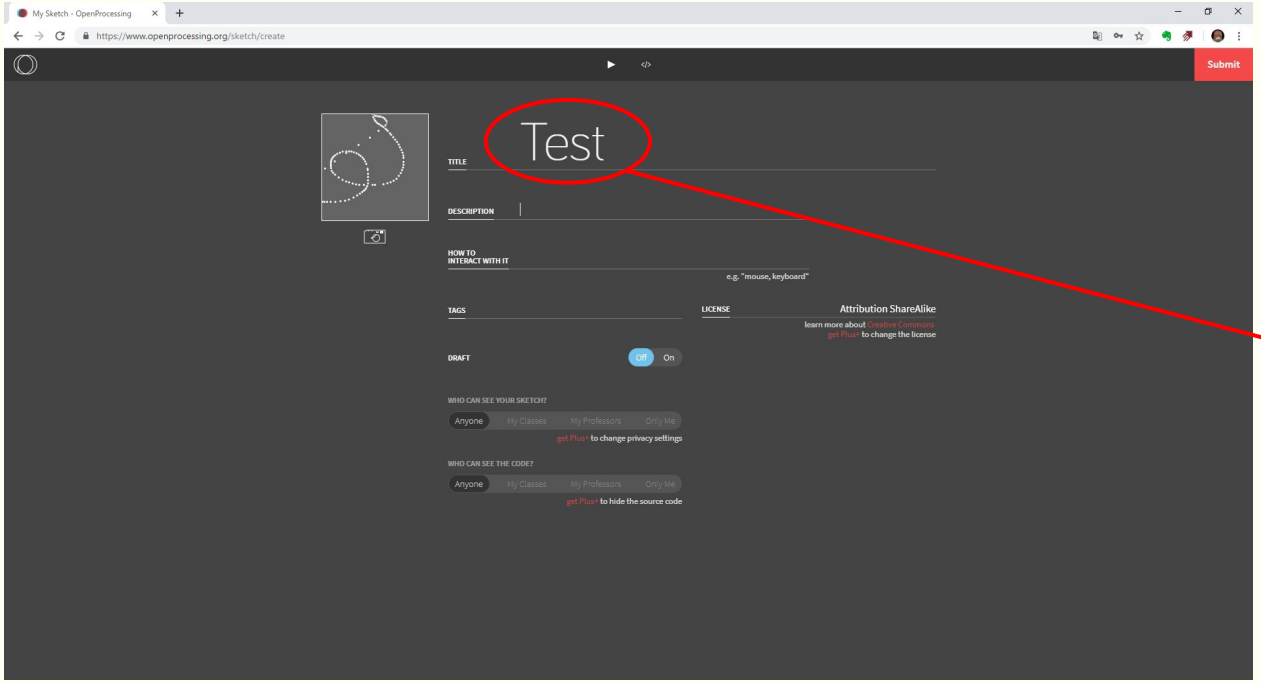
実際に作品をアップロードしてみよう in PC



では、作品をアップロードしていきましょう

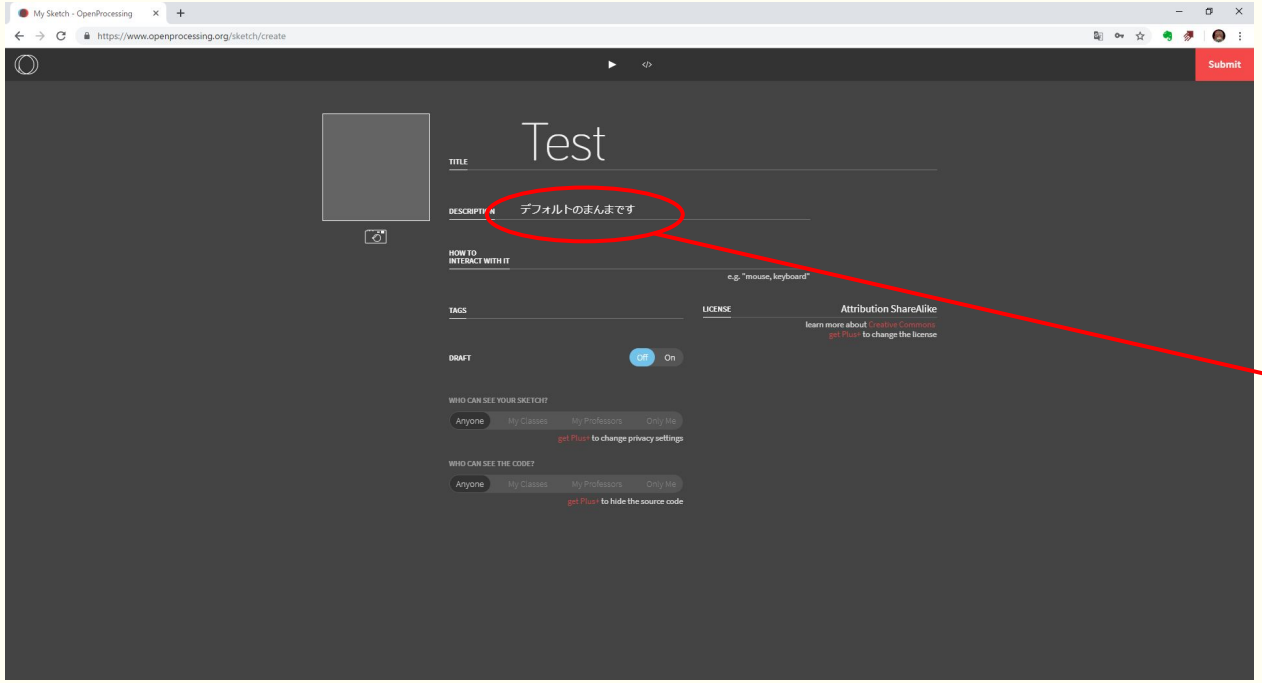
「Save」を押してアップロード画面へ

実際に作品をアップロードしてみよう in PC



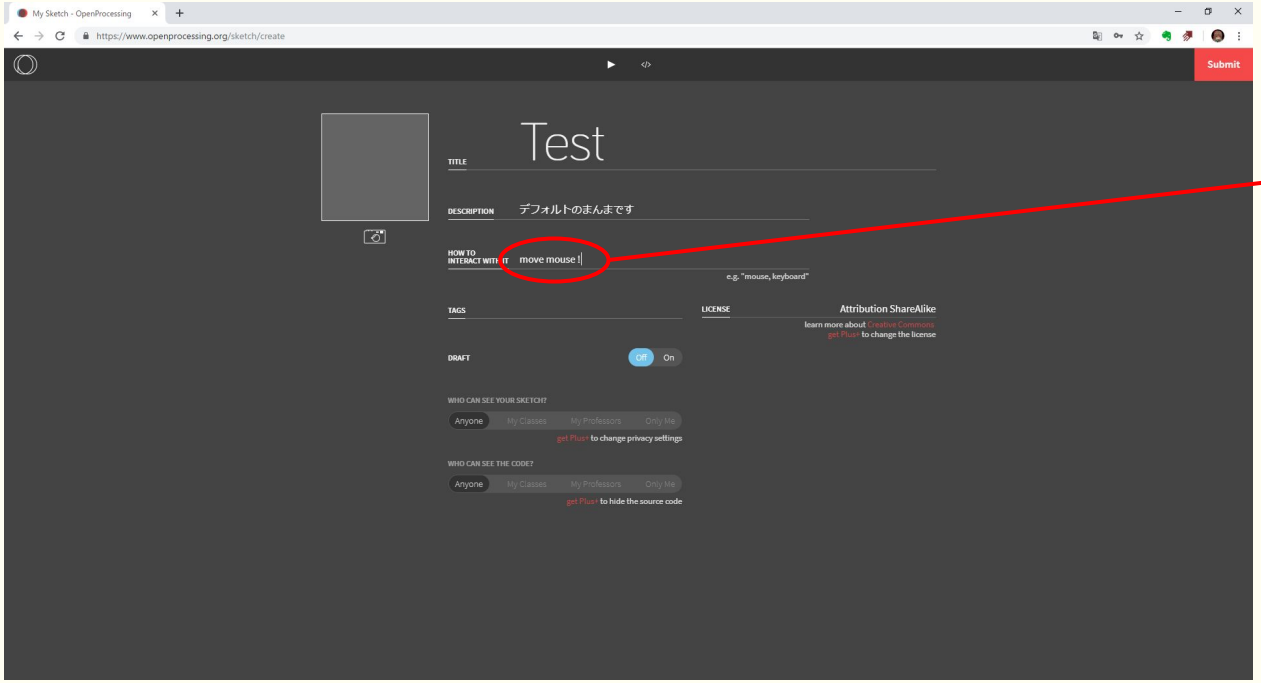
ここに作品の
タイトルを入力します

実際に作品をアップロードしてみよう in PC



ここに作品の
説明を書きます

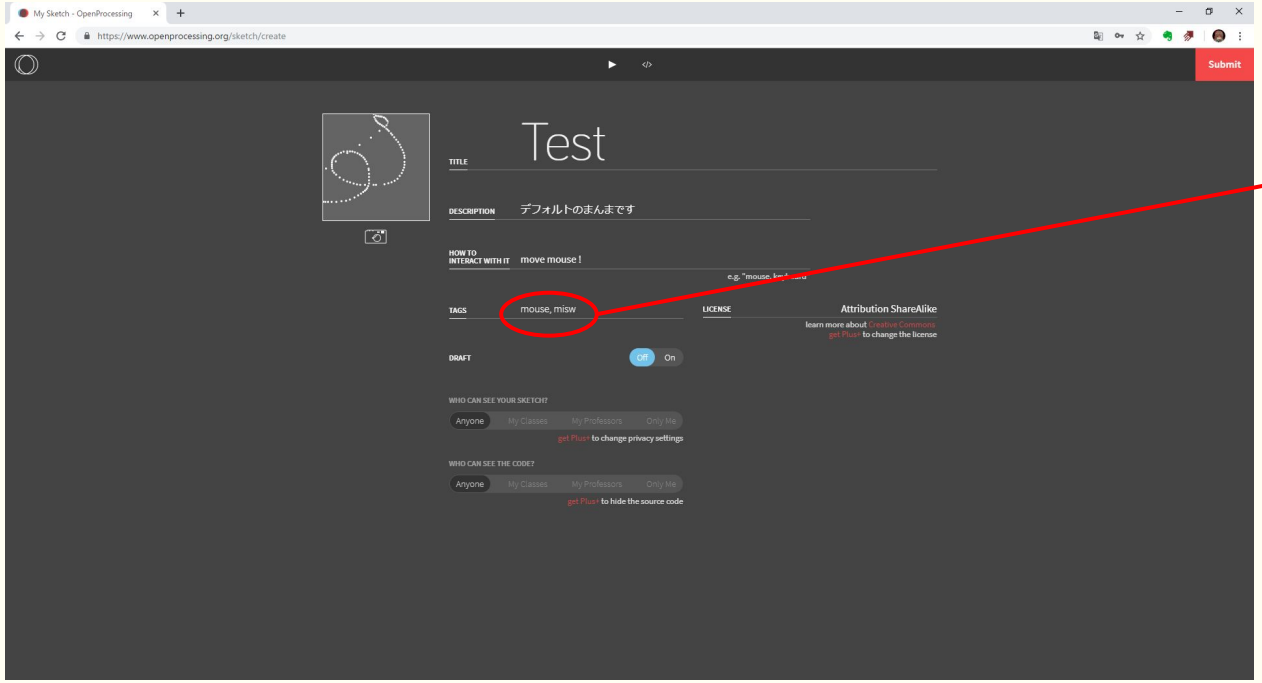
実際に作品をアップロードしてみよう in PC



ここに作品の
操作説明を書きます

この説明文は作品を見る
ときに表示されるように
なっています

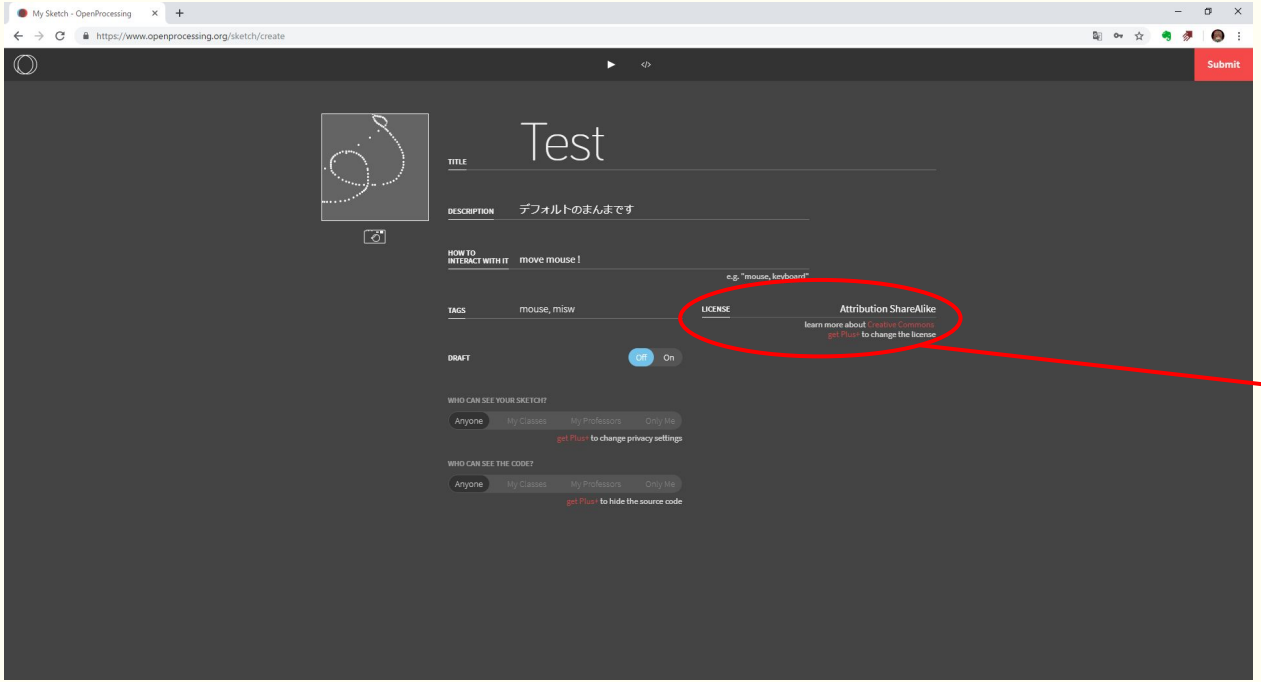
実際に作品をアップロードしてみよう in PC



ここに作品に
つけたいタグを
書きます

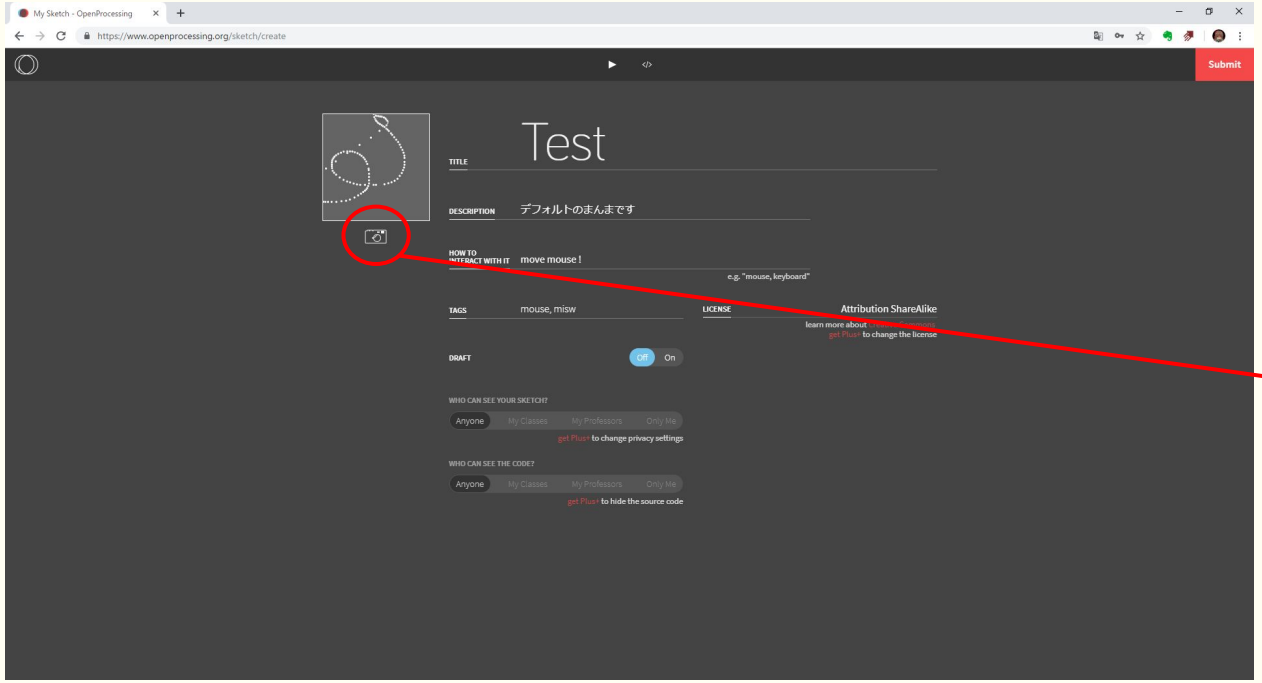
「misw」タグを
お忘れなく

実際に作品をアップロードしてみよう in PC



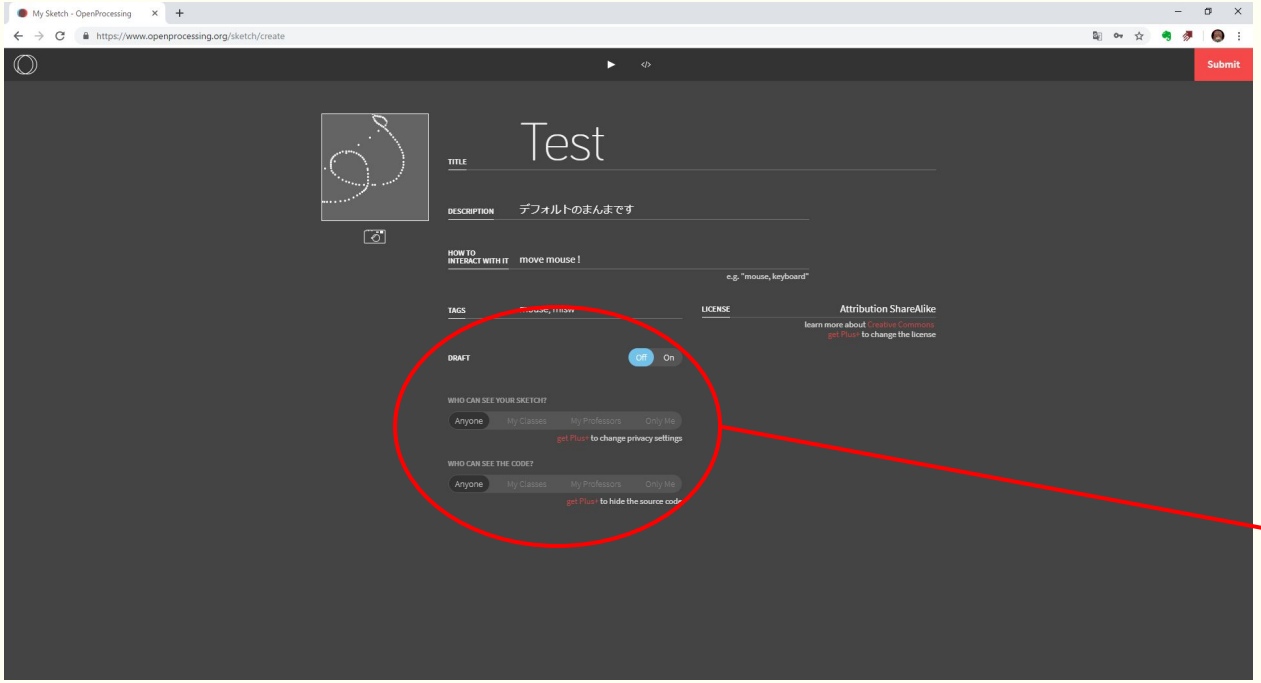
LICENSEは
とりあえず気にしなくて大丈夫です

実際に作品をアップロードしてみよう in PC



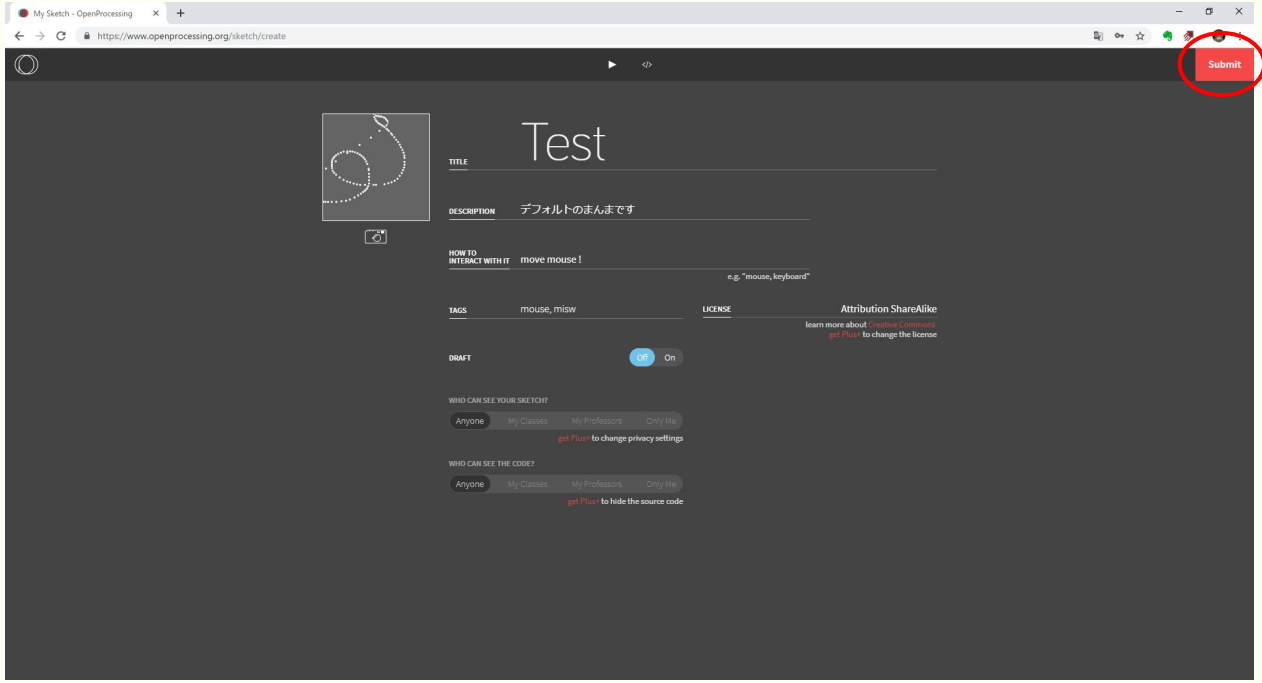
Cameraボタンで
スクリーンショットを
撮り直せます

実際に作品をアップロードしてみよう in PC



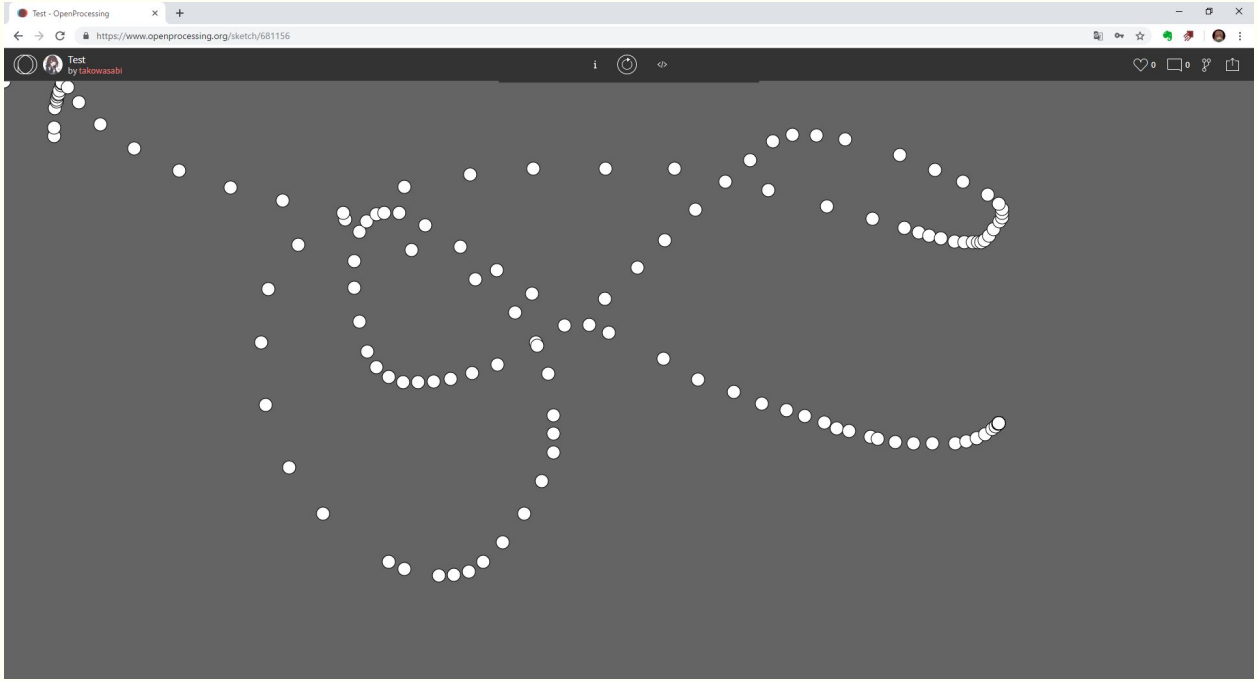
ここら辺で
アクセス範囲を
指定できます

実際に作品をアップロードしてみよう in PC



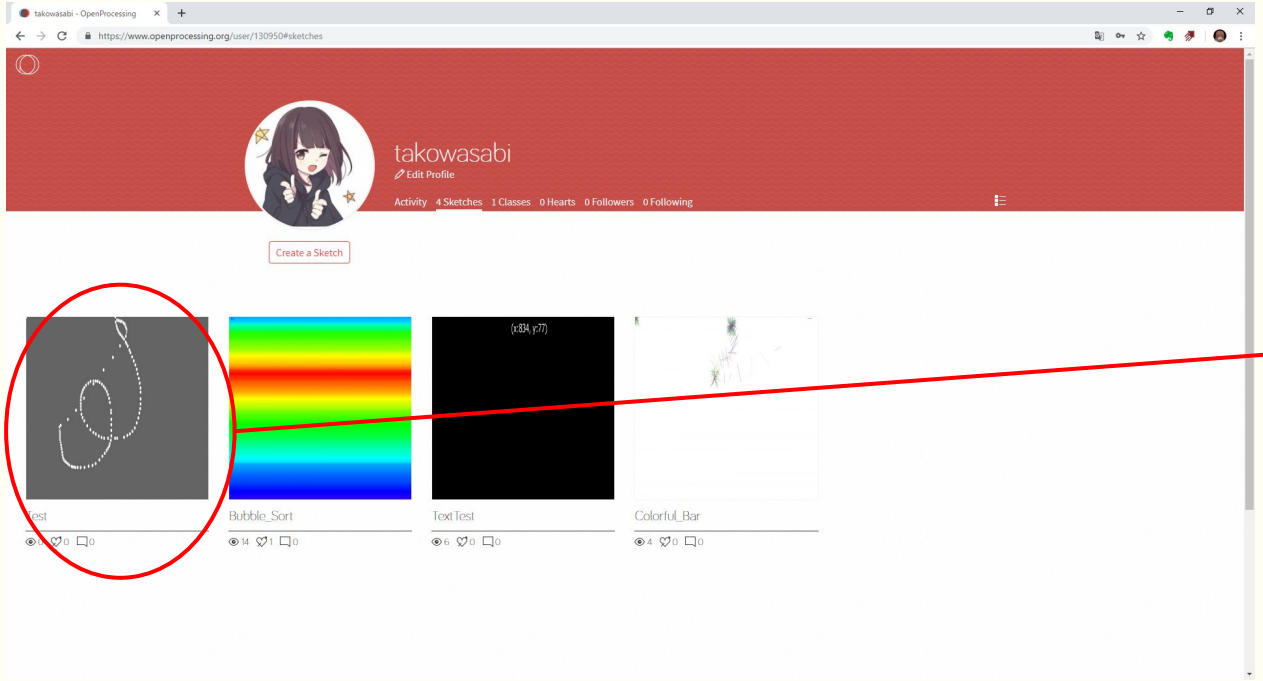
入力が終わったら
「Submit」で公開！

実際に作品をアップロードしてみよう in PC



作品の視聴画面に行けたら公開完了です！！！

実際に作品をアップロードしてみよう in PC



公開した作品はマイページのSketchesから確認できます