

日本大学文理学部 オープンキャンパス

# やさしいプログラミング実習

入門編「図形を描いてみよう」

日本大学 文理学部 情報科学科

# はじめに

- \* 本企画では、ちょっとしたアニメーションを題材に、プログラミングの基本的な仕組みを学びます。
- \* 全くの未経験者を対象としていますので、知識が全くない人でもご安心ください。
- \* ぜひ皆さんも、この動画を一時停止しながら、実際に入力してみても動作を確かめてみてください。

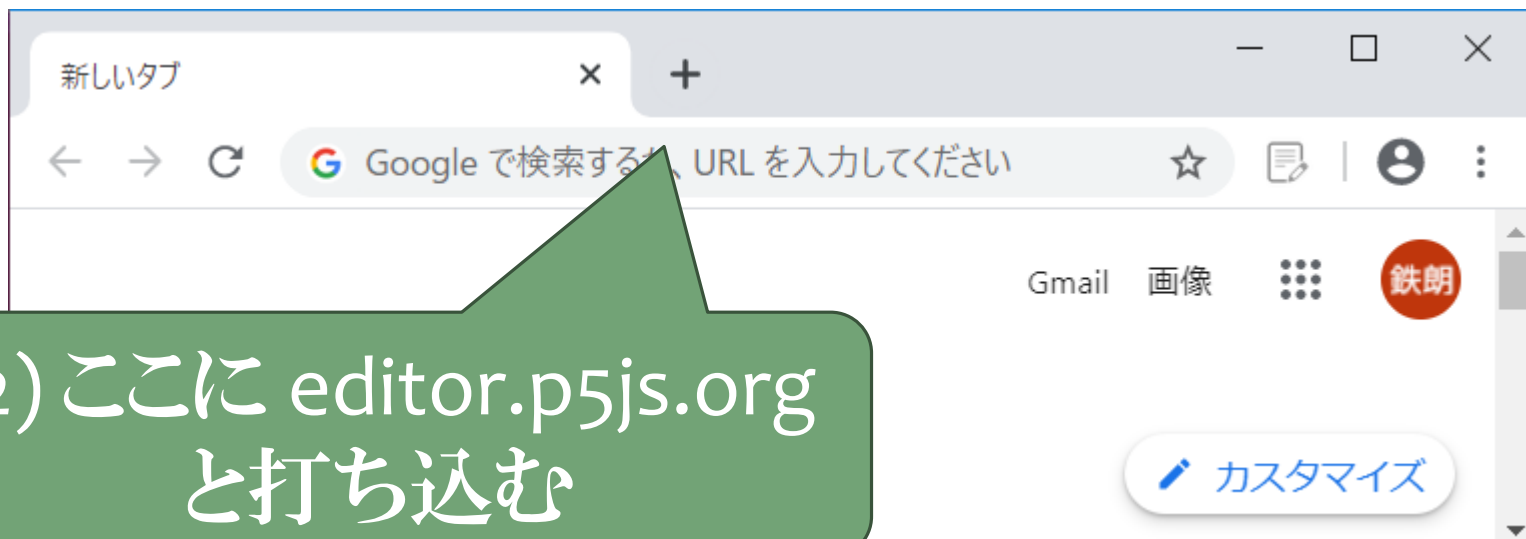
# 0. 準備

Webブラウザ（Google Chromeなど）を開こう。  
アドレスを入力するところに「editor.p5js.org」と打とう。



(1) これをダブルクリック  
(シングルクリックの場合もある)

(2) ここに editor.p5js.org  
と打ち込む



実行開始

実行停止

The image shows a web editor interface with several callouts. At the top, there are two callouts: '実行開始' (Start Execution) pointing to a play button and '実行停止' (Stop Execution) pointing to a red square button. Below these is a menu bar with 'File', 'Edit', 'Sketch', and 'Help'. To the right of the menu bar are 'Log in' and 'Sign up' links. Below the menu bar is a toolbar with a play button, a red square button, an 'Auto-refresh' checkbox, and the text 'Trusting huckleberry'. To the right of the toolbar is a settings gear icon. Below the toolbar is a code editor with a file name 'sketch.js' and a 'Preview' button. The code editor contains the following code:

```
1 function setup() {  
2   createCanvas(400, 400);  
3 }  
4  
5 function draw() {  
6   background(220);  
7 }
```

At the bottom of the code editor is a 'Console' area with a 'Clear' button and a dropdown arrow. A callout 'ここにプログラムを入力' (Enter program here) points to the code editor. Another callout 'ここに実行結果が表示される' (Execution results are displayed here) points to the console area.

ここにプログラムを入力

ここに実行結果が  
表示される

# 1. はじめの一步

すでにプログラムのひな型が入力されている。  
これを実行してみよう。(何も出てこないけど…)

最初に1回だけ実行

```
function setup() {  
  createCanvas(400, 400);  
}
```

画面領域のサイズを400×400にするよ

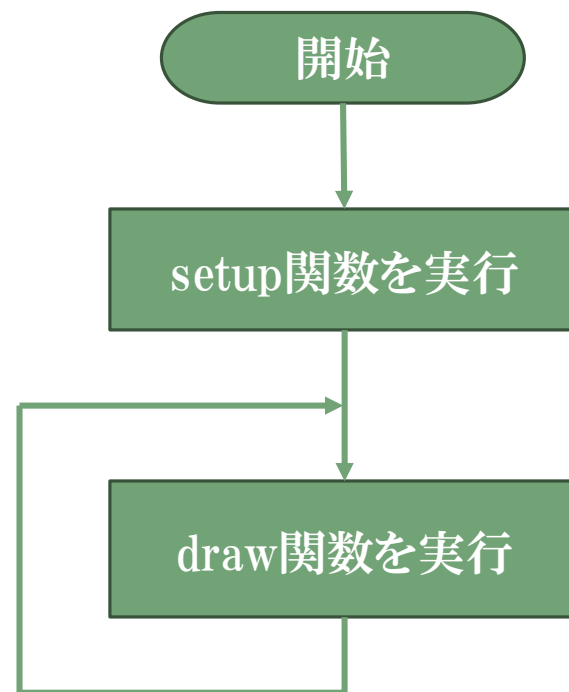
1/60秒ごとに繰り返し実行

```
function draw() {  
  background(220);  
}
```

背景の色を灰色にするよ  
(0:黒、255:白)

# [補足] 実行の流れ

- \* Processingでは、setup関数とdraw関数を作成します。
- \* 関数は、いくつかの命令を1つのカタマリにしたものです。
- \* 最初にsetup関数が1回だけ実行されます。
- \* その後、1/60秒ごとにdraw関数が繰り返し実行されます。



## 2. ボールを描こう

```
function setup() {  
  createCanvas(400, 400);  
}  
  
function draw() {  
  background(220);  
}
```

ここに次の命令を入力しよう。

```
var x = 200, y = 10;  
var r = 10;
```

ここに次の命令を入力しよう。

```
fill(255);  
ellipse(x, y, r*2, r*2);
```

- [注意1] 必ず半角で入力すること。
- [注意2] 必ず小文字で入力すること。
- [注意3] 最後のセミコロンを忘れないこと。

入力したら実行しよう。

ボールが現れます！

The screenshot shows the p5.js Web Editor interface. The browser tab is titled "p5.js Web Editor" and the address bar shows "editor.p5js.org". The editor has a menu bar with "File", "Edit", "Sketch", and "Help". Below the menu bar, there are controls for "Auto-refresh" (unchecked) and "Trusting huckleberry" (checked). The editor is currently editing a file named "sketch.js".

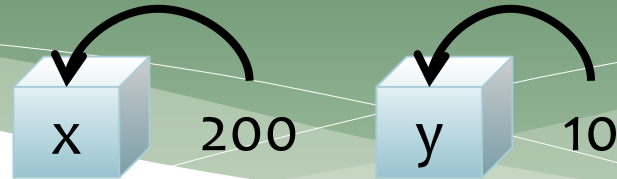
```
1 var x = 200, y = 10;
2 var r = 10;
3
4 function setup() {
5   createCanvas(400, 400);
6 }
7
8 function draw() {
9   background(220);
10  fill(255);
11  ellipse(x, y, r*2, r*2);
12 }
```

The preview window shows a gray canvas with a white circle at the top center. A green callout box with white text "ボールが出現" (Ball appears) points to the circle. Another green callout box with white text "この4行を追加" (Add these 4 lines) points to the code lines 1-4 in the editor.



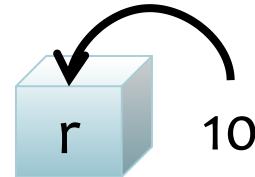
# [補足1] それぞれの命令の意味

```
var x = 200, y = 10;
```



x、yという名前の変数を作って、そこに200、10を代入

```
var r = 10;
```



同様に、rという名前の変数を作って、そこに10を代入

```
fill(255);
```

図形描画の際の色を白に設定

```
ellipse(x, y, r*2, r*2);
```

座標(x, y)を中心とする直径 $r \times 2$ の円を描画

# [補足2] 図形を描く命令

- ellipse: 円や楕円を描く

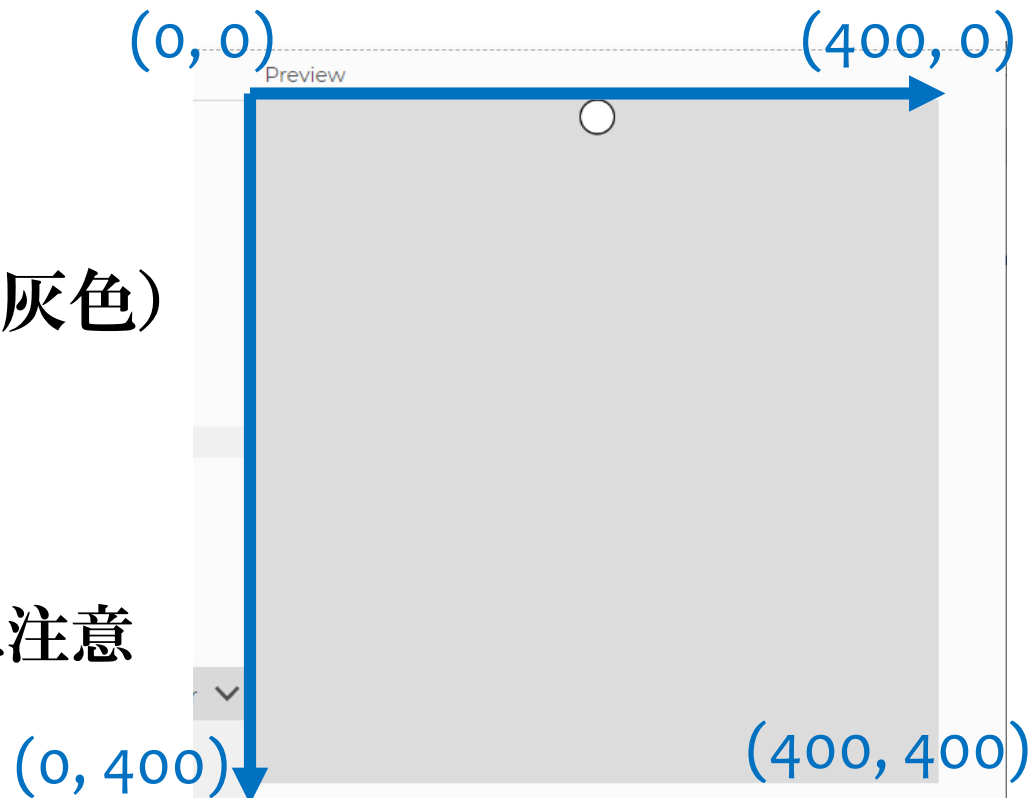
ellipse(中心のx座標, 中心のy座標, 横の長さ, 縦の長さ);

- fill: 色の設定

fill(色の値);

(0:黒、255:白、その間は灰色)

原点が左上で、下にいくほど  
y座標の値が大きくなることに注意



# やってみよう

変数 $x$ ,  $y$ ,  $r$ の値やfill命令に与える値を変えてみて、  
ボールの出現位置、サイズ、色をいろいろと変えてみよう。

# まとめ

入門編では、

- \* Processingのプログラムの基本構成
  - \* 変数
  - \* 色の指定
  - \* 円を描く方法
- を学びました。

実践編では、この円を動かす方法を学びます。

ご覧いただき、ありがとうございました。