



社会と情報 プログラミング超入門①

プログラミング

- 小学校からプログラミング教育が必修化された
- 「プログラミングは大事！」という風潮がある

プログラミングを学ぶ目的

- プログラミングのスキルを身に付けるため
- 「プログラミング的思考」を身に付けるため
 - 問題を解決するためには何をどういう順番で組み合わせれば実現できるか？ を論理的に考えられる力

アルゴリズムの3構造

順次(順接)

- ・ A をしたら次にBをする

分岐

- ・ もし〇〇ならAをする(そうでなければBをする)

反復(繰り返し、ループ)

- ・ 〇〇である限りずっとAを繰り返す

すべてこの3つで
表現できる

プログラミングとは

- アルゴリズムをコンピュータが実行できる形にすること



```
if (a-c!=0) {  
    return (-b+d)/(a-c);  
}  
else if (-b+d==0) {  
    return "Indeterminate";  
}  
else {  
    return "Unsolvable";  
}
```

```
01100011 01101111 01101110 01110011 01110100 00100000 01100111 01100011  
01100100 00111101 00101000 01100001 00101100 01100010 00101001 00111101  
00111110 01111011 00001010 00100000 00100000 01101100 01100101 01110100  
00100000 01110010 00111101 01100001 00100101 01100010 00111011 00001010  
00100000 00100000 01101001 01100110 00101000 01110010 00111101 00111101  
00110000 00101001 00100000 01110010 01100101 01110100 01110101 01110010  
01101110 00100000 01100010 00111011 00001010 00100000 00100000 01100101  
01101100 01110011 01100101 00100000 01110010 01100101 01110100 01110101  
01110010 01101110 00100000 01100111 01100011 01100100 00101000 01100010  
00101100 01110010 00101001 00111011 00001010 01111101 00111011 00001010
```

行わせたい処理をプロ
グラミング言語で書く

機械語に変換する

実行する

授業全体の目的

- アニメーションの作成を通し、**実際にコードを書くこと**による**プログラミング**を体験する
- **プログラミング的思考の基礎**を改めて身に付けていく

授業の流れ（全5回）

1

・ 導入（図形を表示させる）

2

・ 変数（図形を動かす）

3

・ 分岐（図形を跳ね返らせる）

4

・ 作品制作①（アニメーションを再現する）

5

・ 作品制作②（アニメーションをアレンジする）

授業で使うシミュレータの紹介

- <https://mathinfolec.github.io/canvas/>
 - Canvasシミュレータ
 - 本来図形を描く前後に必要な処理を省略できるツール

授業内容の記録について

- 皆さんがシミュレータを使っている状況を記録し、大学での研究にデータを利用する予定です
- 個人が特定されることはありません
- データの利用を拒否しても成績には影響しません
- もしデータを使われたくない場合は教えてください

シミュレータの仕組み

変数定義エリア

必要な変数（データを入れる箱）
を定義する

キャンバスのリセット

（勝手にリセットされる）

描画エリア

1フレーム
(1/30秒)

1フレームごとの画面を描く
⇒コマ送りアニメのようになる

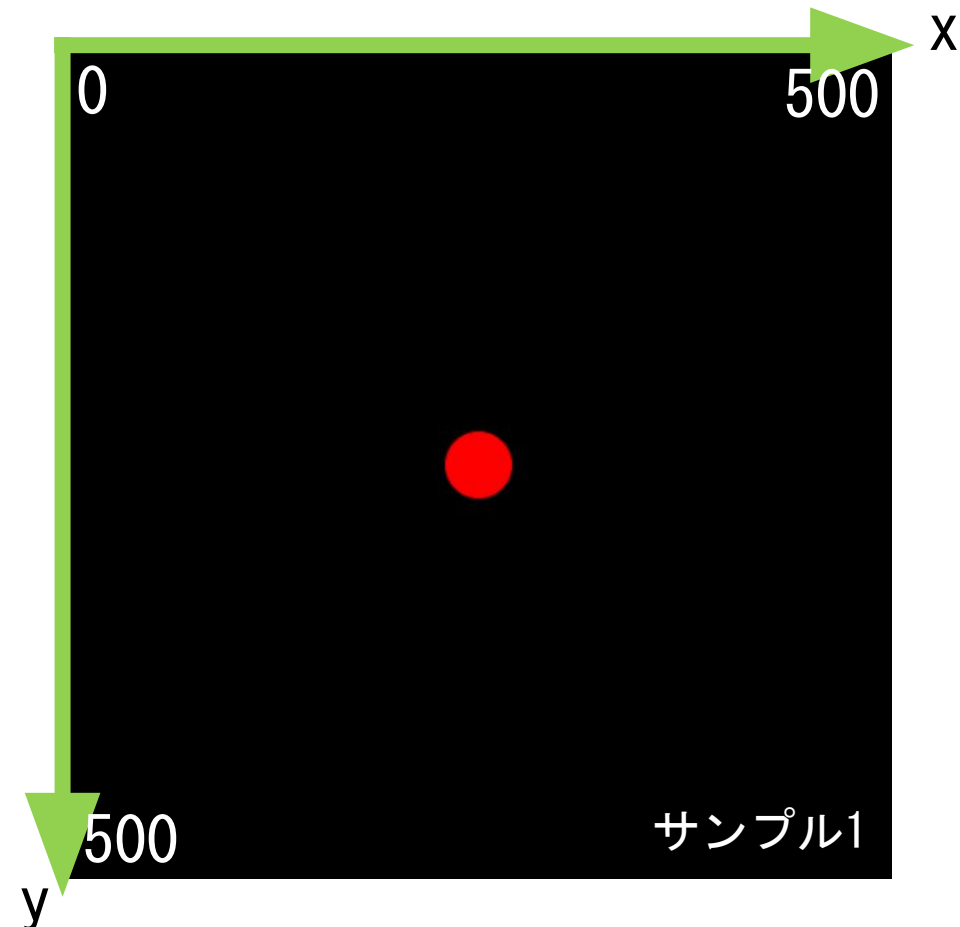
円の表示

これ以降に描く図形の色を赤色に設定したのち、(250, 250)を中心に半径20の円を描く
(この処理を毎フレーム行う)

描画エリア

```
setColor("red");  
drawCircle(250, 250, 20);
```

すべて半角で記入する
大文字・小文字は区別する
セミコロンを末尾に必ず付ける



変数を使った表示

変数定義エリア

```
let x=250;  
let y=250;  
let r=20;
```

描画エリア

```
setColor("red");  
drawCircle(x, y, r);
```

- 変数は「データを入れる箱」のようなもの
- これを使うと後々の計算が便利になる（詳細は次回以降）

演習

変数定義エリア

```
let x=250;  
let y=250;  
let r=20;
```

描画エリア

```
setColor("red");  
drawCircle(x, y, r);
```

1. 左のコードを写し、変数の使い方と描画方法を理解する
2. 円をもう1つ描いてみる
3. 【応用】赤い円のちょうど左隣に同じ大きさの黄色い円が表示されるようにする(黄色い円の中心のx座標は $x-2*r$ になる)