メディアアート・プログラミング I
第3回: 変数と定数、オブジェクト

2020年5月29日

東京藝術大学芸術情報センター（AMC）
田所淳

今日の内容

- 変数と定数
  - システム変数 - マウスの位置を取得してみる
  - 変数とは?
  - 変数を宣言して使用する
  - 変数と定数

- オブジェクト
  - オブジェクトとは
  - オブジェクトを使ってみる!

復習 - p5.jsの構造

復習 - p5.jsの構造

- function setup() と function draw() は何なのか？
- → アニメーションを実現する仕組み

- すこしづつ変化する画像を一定間隔で入れ替えている
- パラパラ漫画のイメージ

復習 - p5.jsの構造

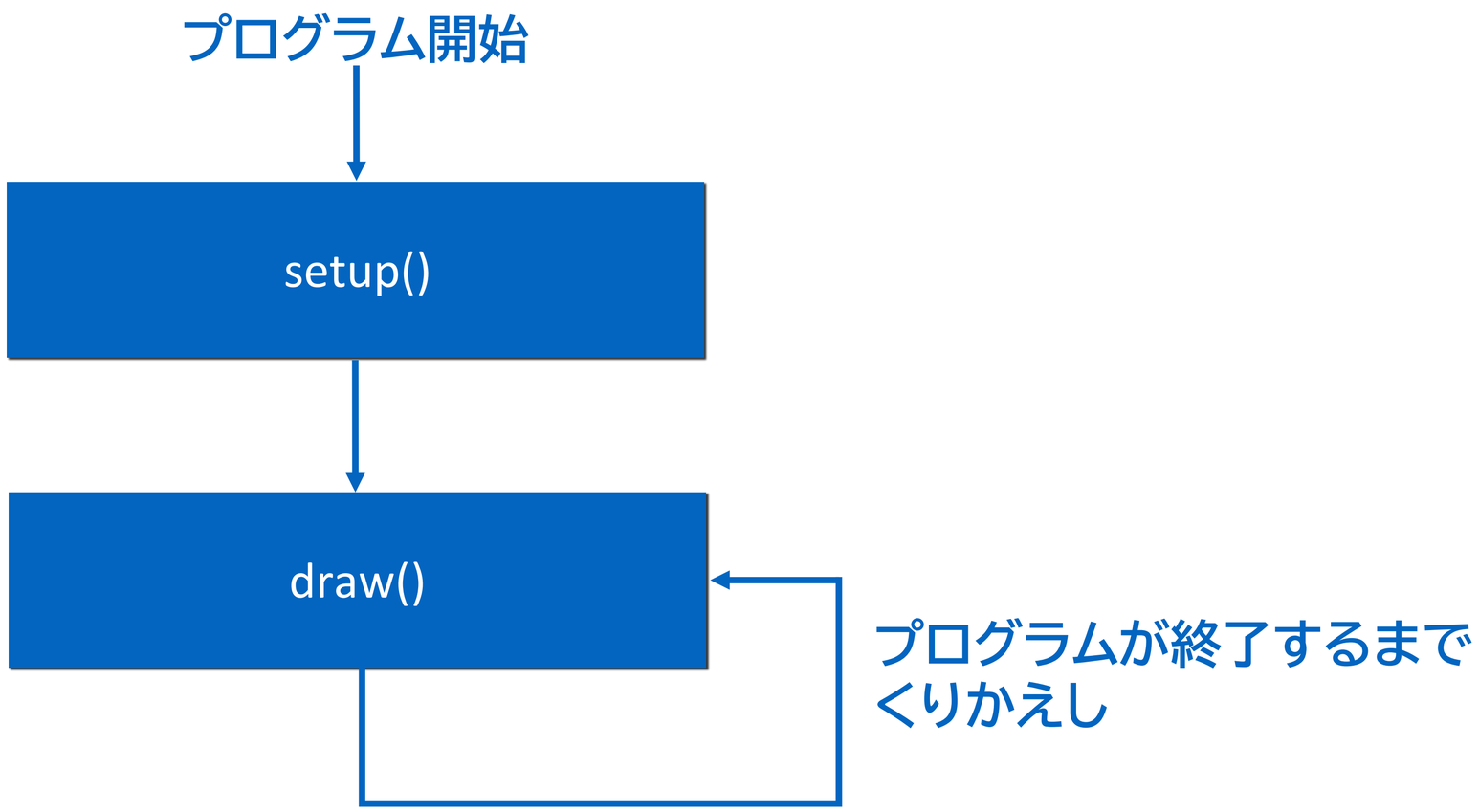
- setup()とupdate()という二つのパートに構造化してアニメーションを実現

- setup() - 初期設定:
- プログラムの起動時に一度だけ実行
- 画面の基本設定やフレームレートなどを設定します。

- draw() - 描画:
- 設定した速さ(フレームレート)でプログラムが終了するまでくりかえし実行されます。

復習 - p5.jsの構造

- setup()とdraw() のイメージ



復習 - p5.jsの構造

- setup：初期化関数
  - プログラムの最初に、1回だけ実行される処理を記述
  - アニメーションの前準備

- setupの中で行われることの多い処理
  - createCanvas：画面のサイズを設定
  - frameRate：画面の書き換え速度を設定
  - colorMode：カラーモードを設定

復習 - p5.jsの構造

- draw：メインループ関数
  - プログラムが終了するまでくりかえし
  - ループの中で図形の場所や色、形を操作してアニメーションにする
  - 画面の書き換え頻度はframeRate()関数で設定する

マウスの位置を取得する - システム変数

マウスの位置を取得する - システム変数

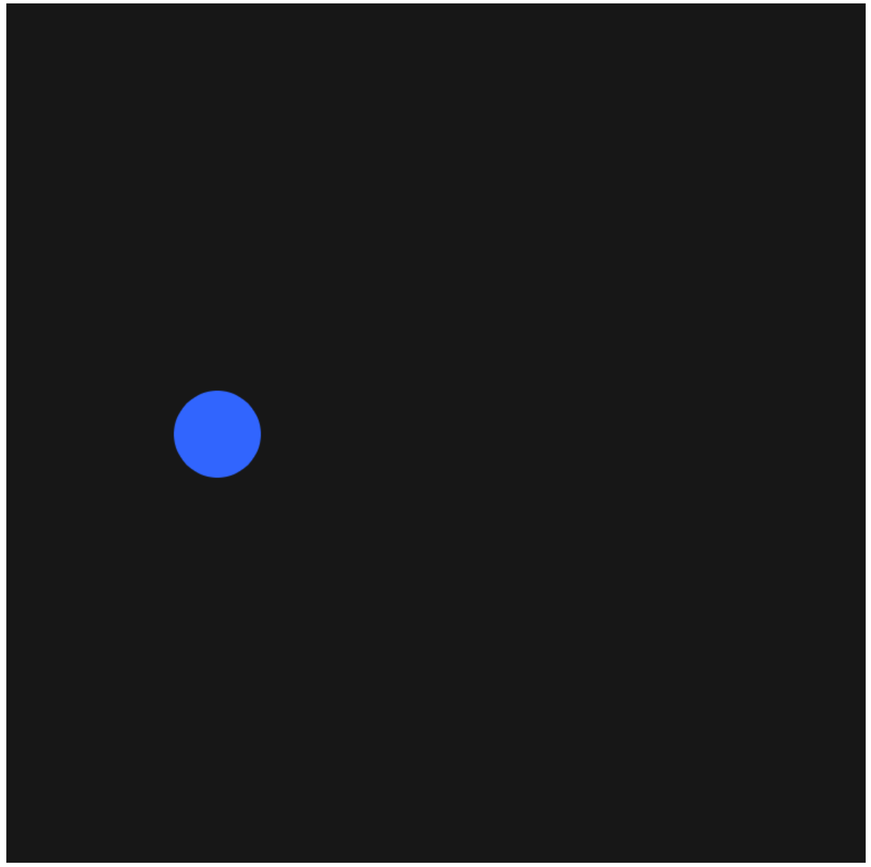
- まず初めに、今まで通りのやり方で円を描く

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

function draw() {
  background(31);
  noStroke();
  fill(63, 127, 255);
  circle(100, 200, 20);
}
```

マウスの位置を取得する - システム変数

- (100, 200)の位置に円が描かれる



マウスの位置を取得する - システム変数

- 円の中心座標（位置）をマウスで変更できるようにしてみたい!

- p5.jsでは、マウスの現在の位置を取得する機能が搭載されている
  - mouseX - マウスのx座標の位置
  - mouseY - マウスのy座標の位置

- システム変数（System Variables）
  - システムによって決められている変化する値を格納する場所
  - mouseX, mouseYもシステム変数

マウスの位置を取得する - システム変数

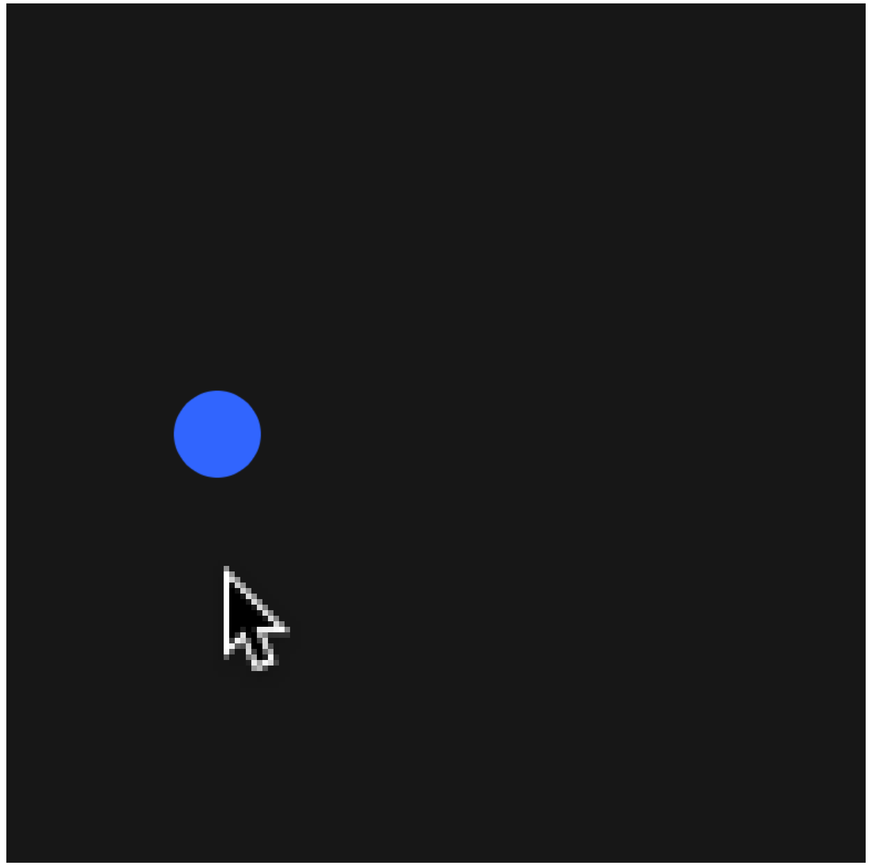
- circleのx座標をmouseXへ

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

function draw(){
  background(31);
  noStroke();
  fill(63, 127, 255);
  circle(mouseX, 200, 20);
}
```

マウスの位置を取得する - システム変数

- 円の横方向の位置（x座標）がマウスで操作可能に!



マウスの位置を取得する - システム変数

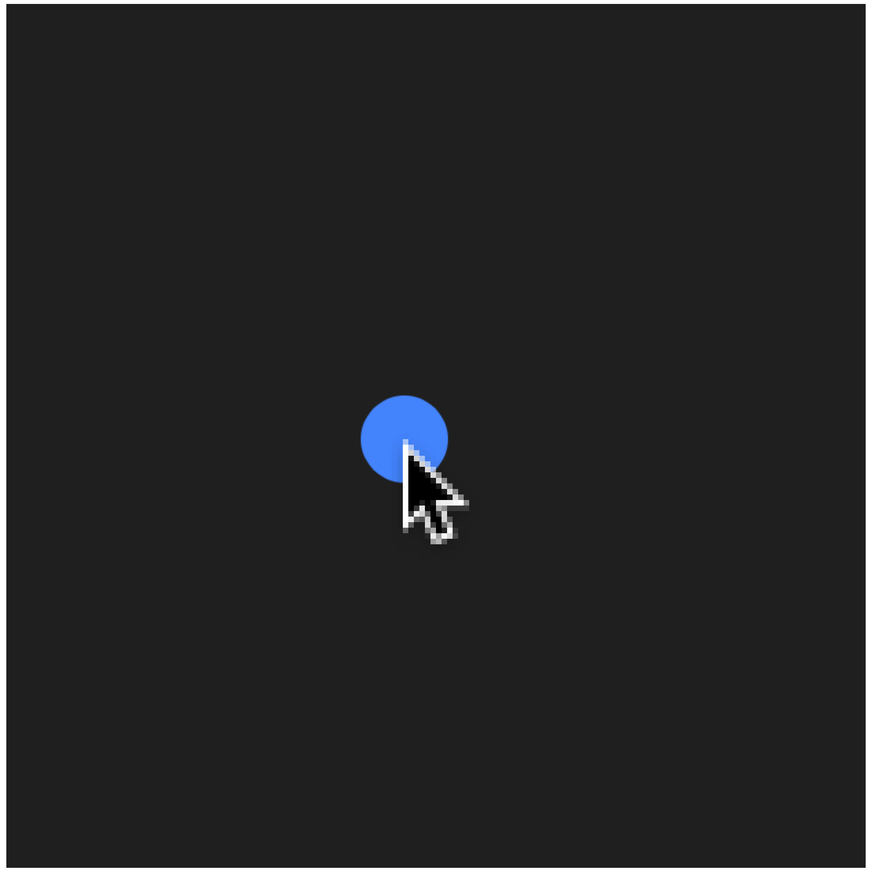
- y座標もmouseYに

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

function draw(){
  background(31);
  noStroke();
  fill(63, 127, 255);
  circle(mouseX, mouseY, 20);
}
```

マウスの位置を取得する - システム変数

- 円の中心座標がマウスの位置と一致した!



マウスの位置を取得する - システム変数

- backgroundの位置をsetup()に移動するとどうなるか？

```
function setup() {
  createCanvas(400, 400);
  background(31);
}

function draw() {
  noStroke();
  fill(63, 127, 255);
  circle(mouseX, mouseY, 20);
}
```

マウスの位置を取得する - システム変数

- お絵描きアプリに変身!!

マウスの位置を取得する - システム変数

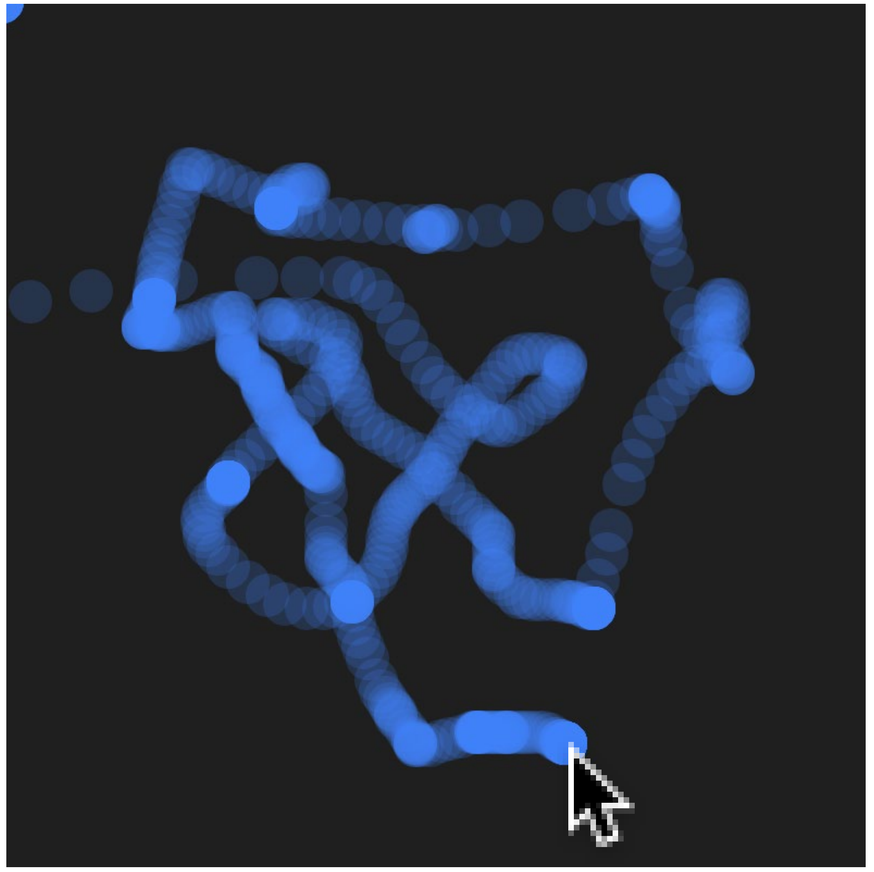
- 円のサイズと透明度を調整

```
function setup(){
  createCanvas(400, 400);
  background(31);
}

function draw(){
  noStroke();
  fill(63, 127, 255, 50);
  circle(mouseX, mouseY, 10);
}
```

マウスの位置を取得する - システム変数

- 濃淡がついた!



システムで定められた値 - システム変数

システムで定められた値 - システム変数

- p5.jsにはシステム変数の他に、変化しない値（= システム定数）が存在する

- 代表的なものとして…
  - width：画面（キャンバス）の幅
  - height：画面（キャンバス）の高さ
  - windowWidth：現在開いているウィンドウの幅
  - windowHeight：現在開いているウィンドウの高さ

マウスの位置を取得する - システム変数

- windowWidth, windowHeightを利用するとフルスクリーン表示に!

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

function draw(){
  background(31);
  noStroke();
  fill(63, 127, 255);
  circle(mouseX, mouseY, 20);
}
```

マウスの位置を取得する - システム変数

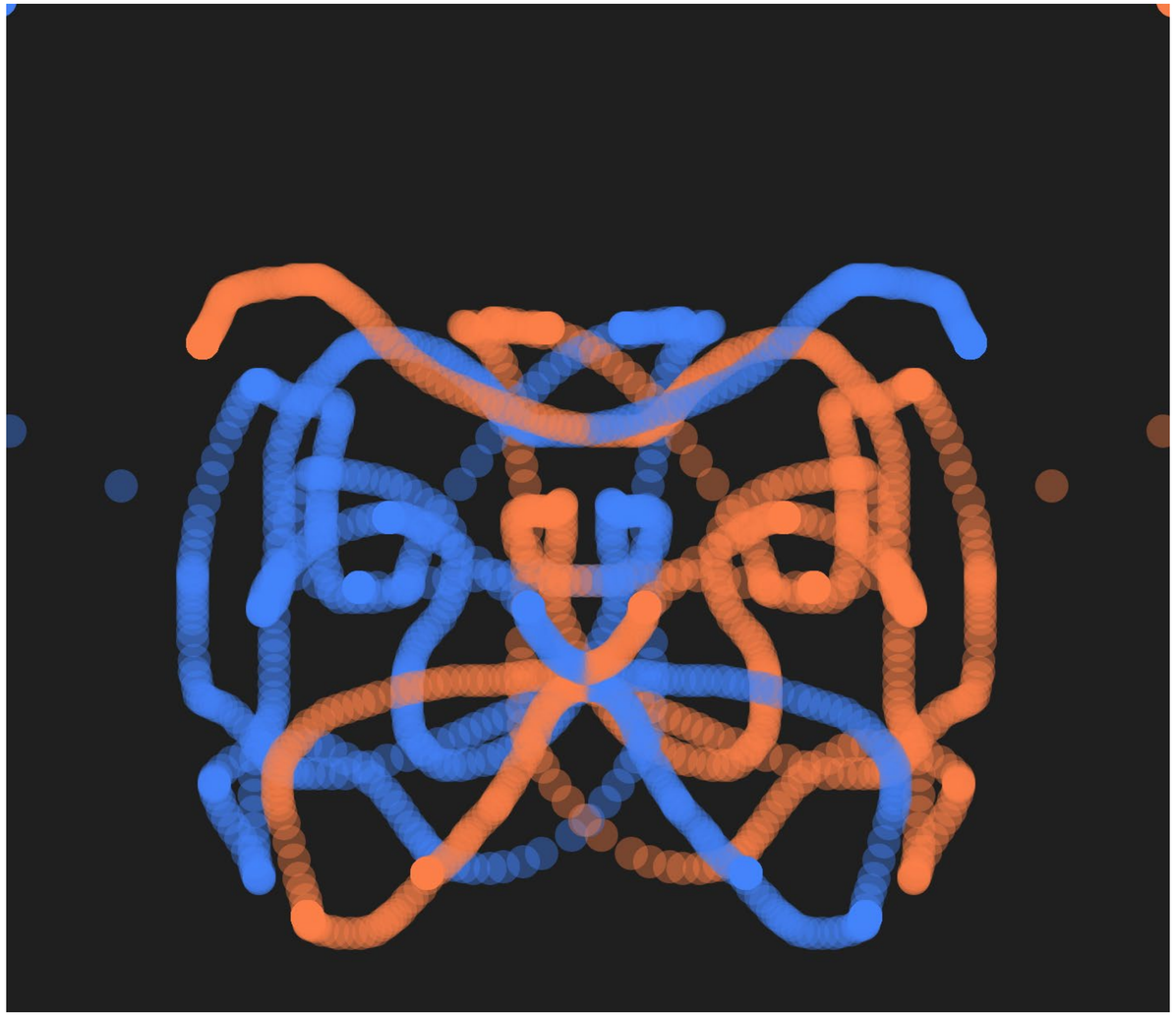
- widthを活用して別の円を描いてみる

```
function setup(){
  createCanvas(windowWidth, windowHeight);
  background(31);
}

function draw(){
  noStroke();
  fill(63, 127, 255, 100);
  circle(mouseX, mouseY, 10);
  fill(255, 127, 63, 100);
  circle(width - mouseX, mouseY, 10);
}
```

システムで定められた値 - システム変数

- 左右反転した形が描かれる!



マウスの位置を取得する - システム変数

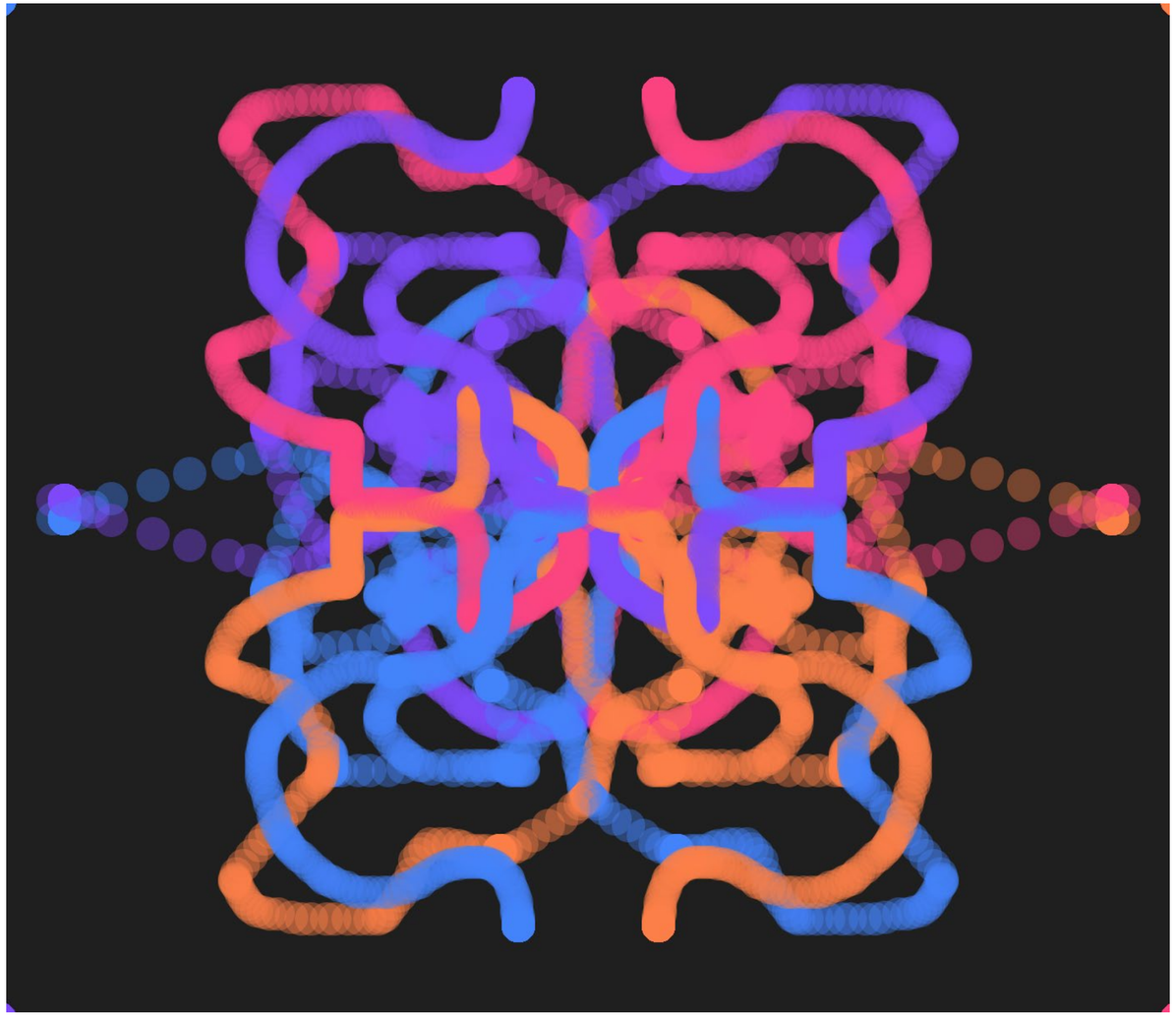
- さらにheightを使用して縦方向にも

```
function setup(){
  createCanvas(windowWidth, windowHeight);
  background(31);
}

function draw(){
  noStroke();
  fill(63, 127, 255, 100);
  circle(mouseX, mouseY, 10);
  fill(255, 127, 63, 100);
  circle(width - mouseX, mouseY, 10);
  fill(127, 63, 255, 100);
  circle(mouseX, height - mouseY, 10);
  fill(255, 63, 127, 100);
  circle(width - mouseX, height - mouseY, 10);
}
```

システムで定められた値 - システム変数

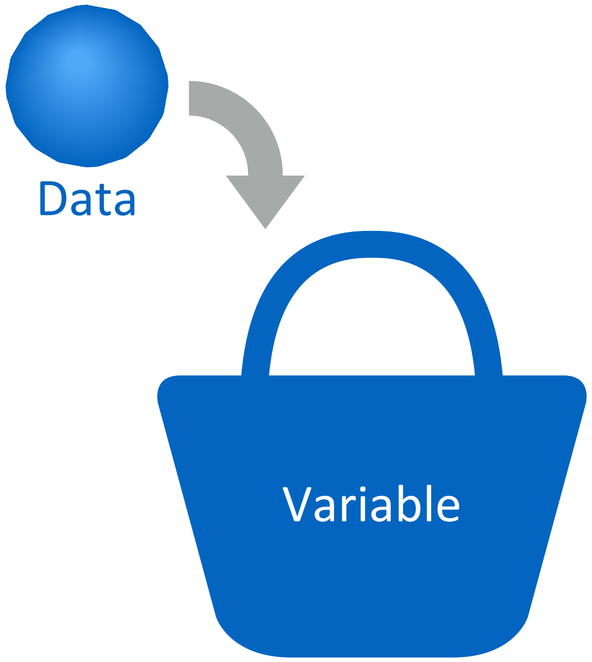
- 万華鏡のような形を描くことができる!



変数と定数

変数と定数

- 変数（variable）:
  - プログラミング言語の仕組み
  - 値を格納して必要なときの取り出すことのできる仕組み
  - 値を何度も入れ直すこと（代入）が可能
  - ただし、一度に入れられるのは1つだけ!



変数と定数

- 定数（constant）：
  - 変数と同様に値を格納して参照する仕組み
  - ただし、一度格納した値は変更することができない
  - 定まった数：定数

変数と定数

- p5.jsはJavaScriptのライブラリ
- 変数や定数の使用方法は、JavaScriptと同じ!

- 変数：let
- 定数：const

変数と定数

- 例えばJavaだと…
  - int, float, char, stringなどがあった

- JavaScript（p5.js）の場合
  - 変数・定数は「型（data type）」を指定しなくて良い!
  - 便利!

変数と定数

- 変数の使用手順
  1. 宣言（Declare）
  2. 初期化（Initialize）
  3. 参照（Reference）

- 変数を活用して円を動かしてみよう!

変数と定数

- まずはベースとなるプログラム

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

function draw(){
  background(63);
  noStroke();
  fill(63, 127, 255);
  circle(100, 100, 40);
}
```

変数と定数

- 宣言と初期化

```
let circleX; ← 宣言

function setup(){
  createCanvas(windowWidth, windowHeight);
  circleX = 100; ← 初期化
}

function draw(){
  background(63);
  noStroke();
  fill(63, 127, 255);
  circle(100, 100, 40);
}
```

変数と定数

- 変数を参照して円のx座標に

```
let circleX;

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

function draw() {
  background(63);
  noStroke();
  fill(63, 127, 255);
  circle(circleX, 100, 40);← 参照
}
```

変数と定数

- 円の座標を少しずつ変化させるとどうなるか？
- drawで代入をくりかえしてみる
- やってみよう!

変数と定数

- 変数を参照して円のx座標に

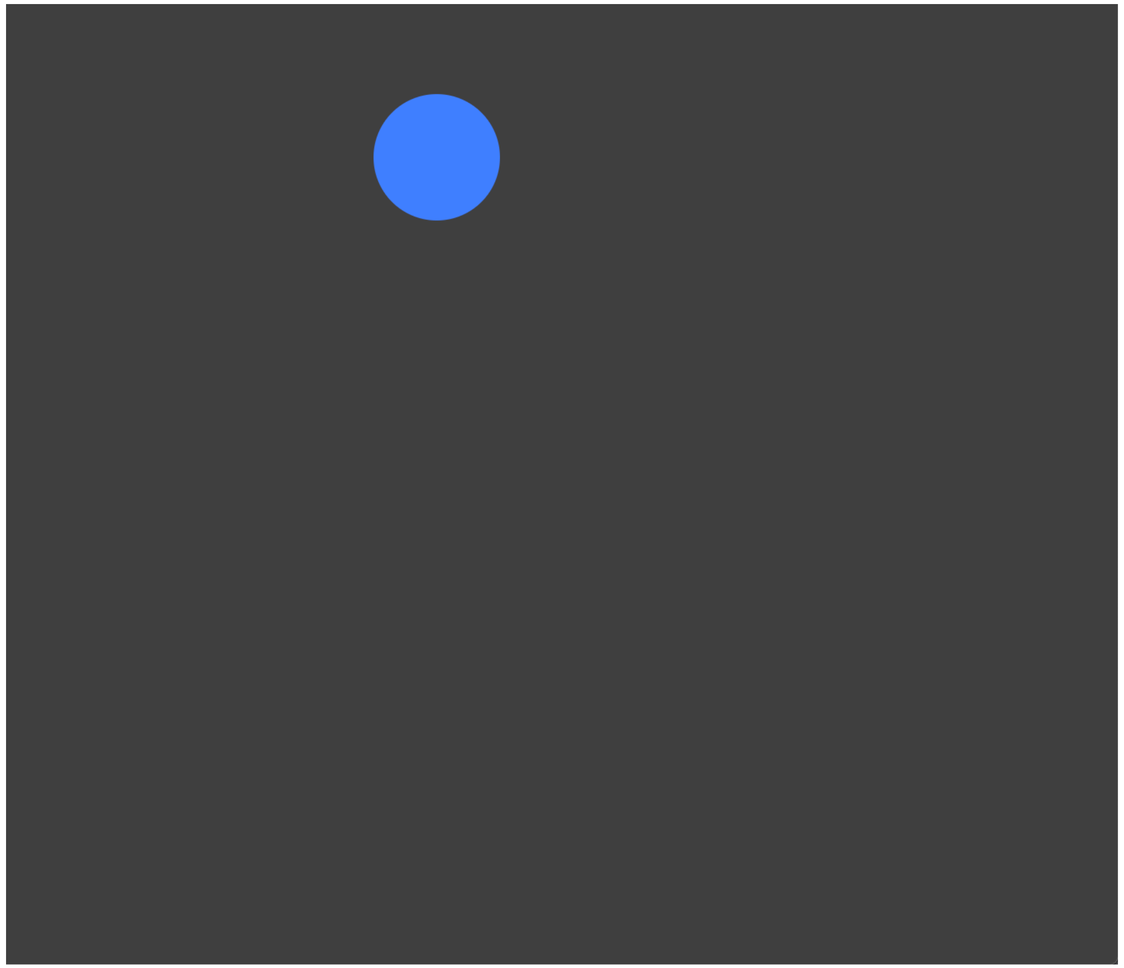
```
let circleX;

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

function draw(){
  background(63);
  noStroke();
  fill(63, 127, 255);
  circle(circleX, 100, 40);
  circleX = circleX + 1; ← 格納されている値に1を足して再度代入
}
```

変数と定数

- 円が動いた!



変数と定数

- 円に関するいろいろな値を変数にしてみる
  - x座標
  - y座標
  - 半径
  - 色（red, green, blue）

変数と定数

- 円に関する変数をいろいろ定義

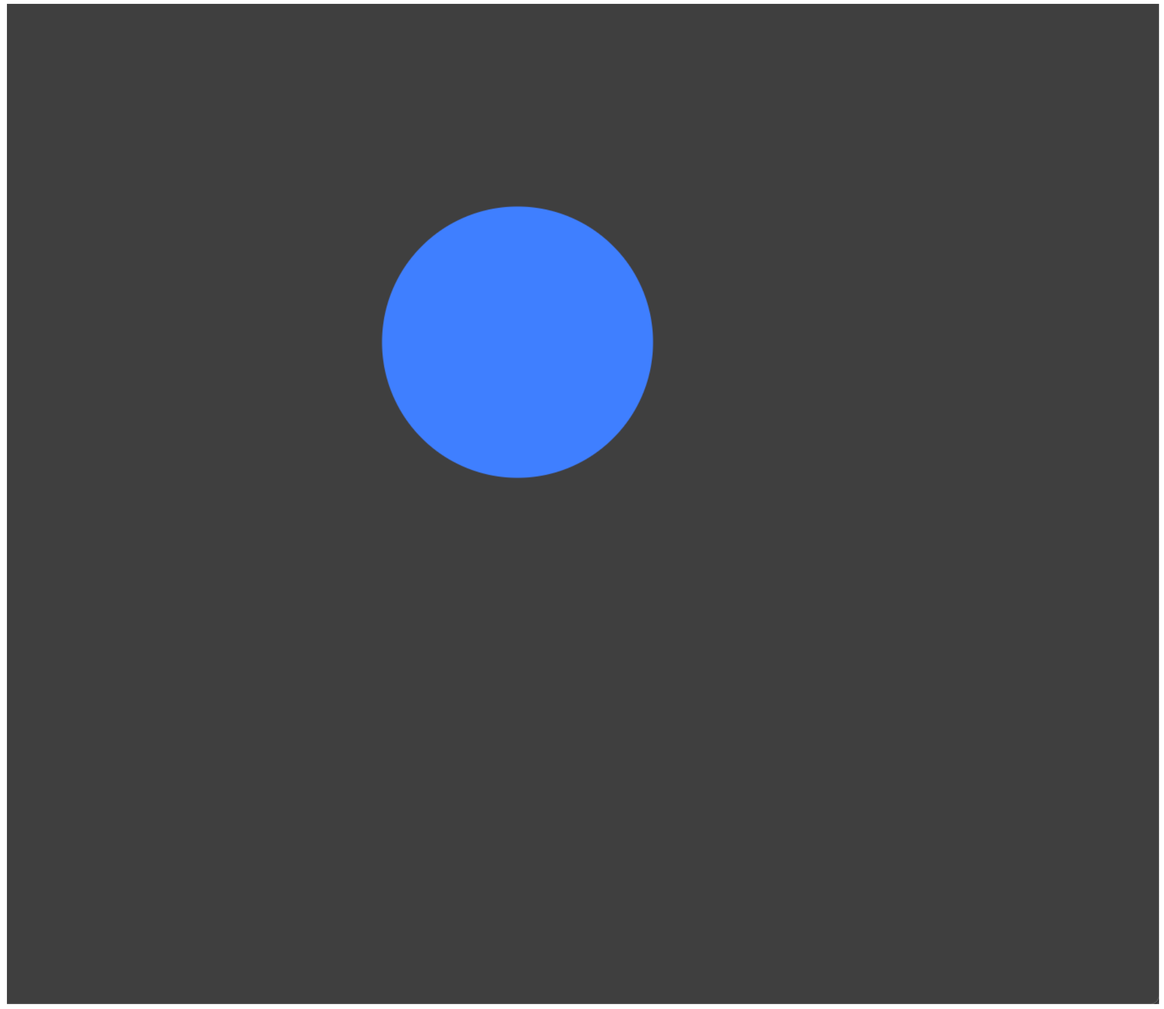
```
let circleX = 100;
let circleY = 100;
let radius = 40;
let red = 63;
let green = 127;
let blue = 255;

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

function draw(){
  background(63);
  noStroke();
  fill(red, green, blue);
  circle(circleX, circleY, radius);
  circleX = circleX + 1;
  circleY = circleY + 0.5;
  radius = radius + 0.2;
}
```

変数と定数

- 膨張しながら斜め下に移動



オブジェクト

オブジェクト

- 円に関する変数がだいぶ増えてしまった…
  - もう少しデータの格納方法を工夫したい
  - 円に関するデータをまとめたい

- → オブジェクト（Object）を利用すると便利!

オブジェクト

- JavaScriptのオブジェクト

```
let objectName = {
  property1: value1,
  property2: value2,
  property3: value3,
  ...
};
```

オブジェクト

- オブジェクトの値の参照のやりかた

```
let objectName = {
  property1: value1,
  property2: value2,
  property3: value3,
  ...
};

objectName.property1; ← value1
objectName.property2; ← value2
objectName.property3; ← value3
…
```

オブジェクト

- 円に関する値をオブジェクトballにまとめてみる!

```
let ball = {
  x: 100,
  y: 100,
  radius: 40,
  r: 63,
  g: 127,
  b: 255
};
```

オブジェクト

- プログラムに適用してみる!

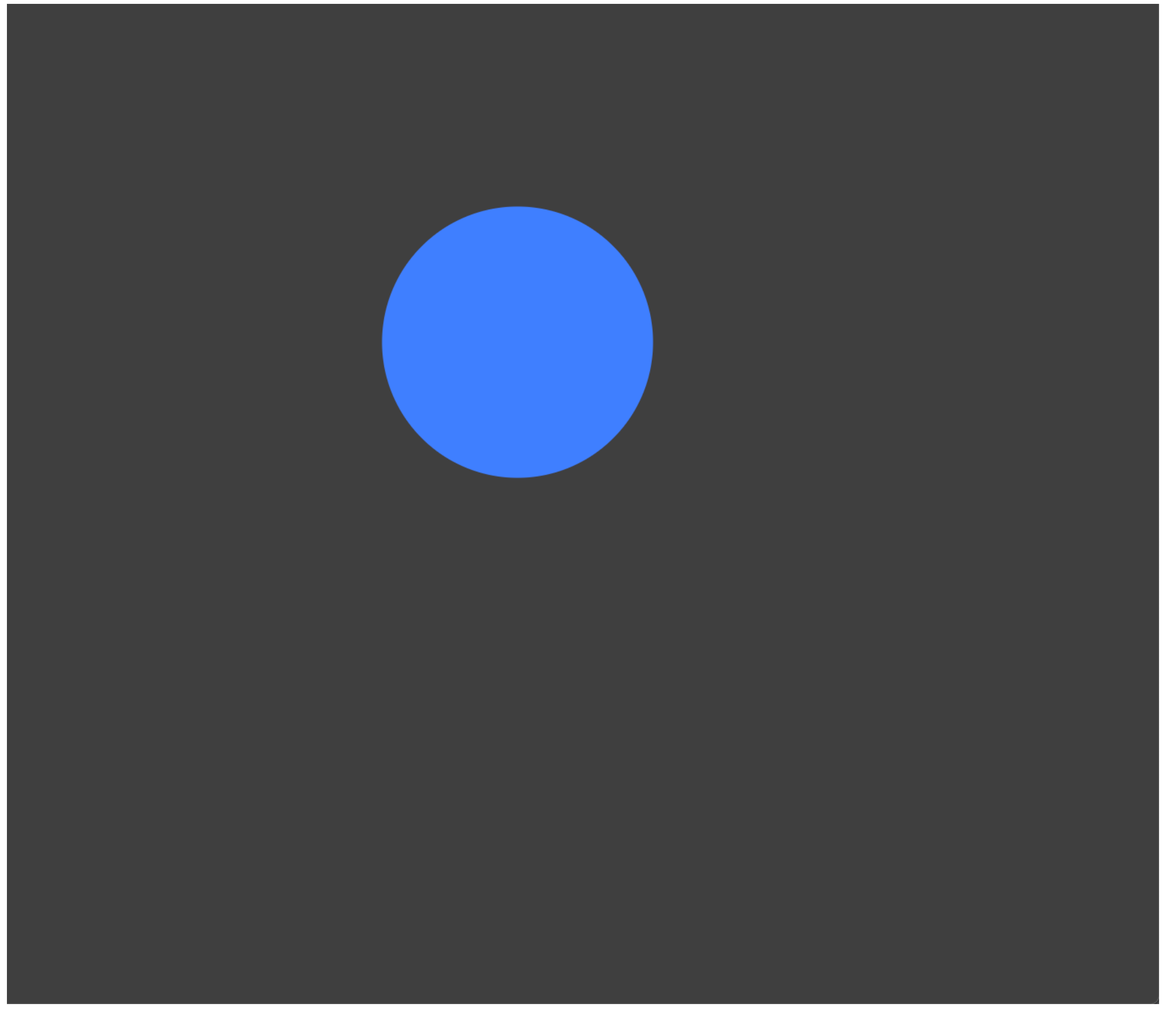
```
let ball = {
  x: 100, y: 100, radius: 40,
  r: 63, g: 127, b: 255
};

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

function draw() {
  background(63);
  noStroke();
  fill(ball.r, ball.g, ball.b);
  circle(ball.x, ball.y, ball.radius);
  ball.x = ball.x + 1;
  ball.y = ball.y + 0.5;
  ball.radius = ball.radius + 0.2;
}
```

オブジェクト

- 同じ動きが再現された!



実習!：マウスの位置によるインタラクション

実習!：マウスの位置によるインタラクション

- 後半は実習!

- テーマ：マウスの位置によるインタラクション
  - マウスの位置を取得するシステム変数（mouseX, mouseY）を使用
  - 面白いインタラクションを考えてみる
  - 位置、色、大きさなどを変化させてみる
  - 今回はフルスクリーンで
    - createCanvas(windowWidth, windowHeight);