

# 情報・工学探究

## 第1回 メディアプログラミング授業資料

見て，感じて，描いてみよう

授業担当：情報学部 情報学科 情報メディア専攻 清水 哲也



# 本日の授業内容

- 授業導入・自己紹介, 「メディアとは何か？」
- p5.jsの概要と描画の基本
- 例題：抽象アートを作ろう
- 解説：生成的アートの考え方
- 課題制作：「感情」や「音楽」をテーマに
- ふりかえり・感想

※適当なタイミングで10分間の休憩を入れます

# 「メディア」ってなんだろう？

- メディア＝情報を伝える手段（例：新聞、映像、音楽）
- デジタル時代のメディア：視覚・音・動き・インタラクション
- プログラミングでメディアを自分で作る体験をしよう！

# 本日使用するツール

- iPad（第10世代） + Bluetoothキーボード + マウス（なくてもOK）
- Safari ブラウザ (Google Chrome でも可)
- <https://editor.p5js.org/>

# p5.jsとは？

- JavaScriptベースのビジュアルプログラミングライブラリ
- Webブラウザで動く
- 図形，色，動き，インタラクションを簡単に実装
- 教育，アート，Webで広く利用されている
- p5 = [Processing](#) , js = [JavaScript](#)

# 特徴と利点

- ブラウザ上で使える（アプリ不要）
- アカウントなしでも利用可能（保存は不可）
- p5.jsライブラリに特化した設計
- iPadでも動作（Chrome / Safari）

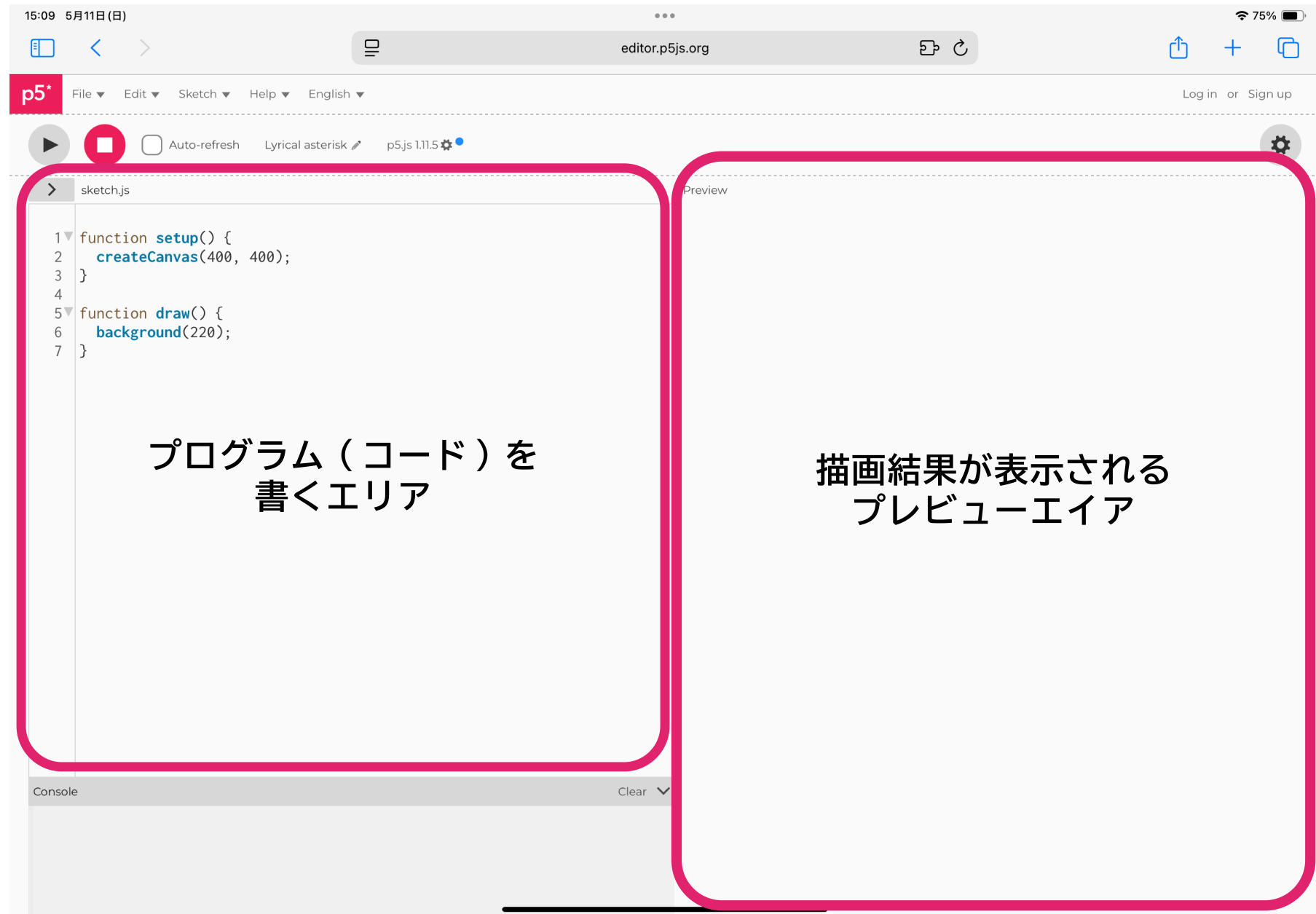
# アクセス方法

- SafariまたはChromeを開く
- アドレスバーに入力：<https://editor.p5js.org/>
- Enterでアクセス



# 画面構成

- 左側：コードを書くエリア
- 右側：描画結果が表示されるプレビューエリア
- 上部：ファイル操作ボタン、実行ボタン、設定



# 実行と停止

- 再生ボタンを押すとプログラムが実行
- 停止ボタンで描画を止めることが可能



# 基本の構成

- `setup()` : 最初に1回実行
- `draw()` : 毎フレーム実行される

```
function setup() {  
  createCanvas(400, 400);  
}  
  
function draw() {  
  background(220);  
  ellipse(200, 200, 100, 100);  
}
```

# よく使う図形

- `rect()` : 長方形（正方形）を描画
  - `rect(左上のX座標, 左上のY座標, 横, 縦)`
- `ellipse()` : 楕円（円）を描画
  - `ellipse(中心のX座標, 中心のY座標, 横, 縦)`
- `line()` : 線を描画
  - `line(始点のX座標, 始点のY座標, 終点のX座標, 終点のY座標)`

```
rect(50, 50, 100, 100);  
ellipse(200, 200, 100, 50);  
line(0, 0, 300, 300);
```

# 色をつける

- `fill()` : 図形などの中を塗りつぶす色
- `stroke()` : 枠線など線の色
- `background()` : 背景色
- 色の指定方法
  - `fill(gray)` : 値が1つの場合グレースケール（値： `0` ~ `255` ）
  - `fill(Red, Green, Blue)` : 値が3つの場合フルカラー（値： `0` ~ `255` ）

```
fill(255, 0, 0);  
stroke(0);  
background(240);
```

# マウス（タッチ）に反応する

- `mouseX`, `mouseY` でマウス（タッチ）の位置を取得

```
function setup() {  
  // 変更なし  
}  
  
function draw() {  
  background(255);  
  ellipse(mouseX, mouseY, 50, 50);  
}
```



# クリック（タッチ）で図形を追加

- マウスを持っているひとは `mousePressed()`
- マウスを持っていないひとは `touchStarted()`

```
function setup() {  
  createCanvas(400, 400);  
  background(255);  
}  
  
function mousePressed() {  
  fill(random(255), random(255), random(255));  
  ellipse(mouseX, mouseY, 30, 30);  
}
```

# ランダムに円を描画

```
function setup() {  
  // 変更なし  
}  
function draw() {  
  fill(random(255), random(255), random(255));  
  ellipse(random(width), random(height), 20, 20);  
}
```

# 例題：抽象アートを作ろう

- `random()` : ランダムな数を生成
  - `random()` : `0` ~ `1` までの乱数（実数）を生成
  - `random(5)` : `0` ~ `5` までの乱数（実数）を生成
  - `random(10, 100)` : `10` ~ `100` までの乱数（実数）を生成
- `touchStarted()` : タッチで追加（マウスのひとは `mousePressed()` を利用）
- `noStroke()` : 枠線を描画しない
- `fill()` , `background()` を工夫

# サンプルコード1：カラフルな円

- ランダムな位置にカラフルな円を描画
- `fill(r, g, b, a)` : 4つ目の値は透明度 `a = 0~255`

```
function setup() {  
  createCanvas(400, 400);  
  background(255);  
  noStroke();  
}  
function draw() {  
  fill(random(255), random(255), random(255), 100);  
  ellipse(random(width), random(height), 30, 30);  
}
```

## サンプルコード2：マウス（タッチ）で描く

- `mouseDragged()`：マウスボタンが押されている間のみ動く関数

```
function setup() {  
  createCanvas(400, 400);  
  background(255);  
}  
  
function mouseDragged() {  
  fill(random(255));  
  ellipse(mouseX, mouseY, 10, 10);  
}
```

## サンプルコード3：画面をタップで反応

- `displayWidth`, `displayHeight` : Preview画面の幅と高さ

```
function setup() {  
  createCanvas(displayWidth, displayHeight);  
  background(255);  
  noStroke();  
}  
  
function touchStarted() {  
  fill(random(255), random(255), random(255));  
  rect(mouseX, mouseY, 50, 50);  
}
```

# 解説：生成的アートとは？


- 「コンピュータの偶然性」を活かしたアート
- 例：毎回違うパターンが生成される
- プログラムがルールを与え，結果は多様

# 有名な生成アートの例

- Generative Design
- Vera Molnár
- Tyler Hobbs (Fidenza)




# Generative Design

- 書籍『Generative Design』シリーズが有名
- JavaScript / p5.js で造形ルールを実装
- 「秩序＋偶然」で構成される視覚表現
-  公式サイト：[generative-gestaltung.de](https://generative-gestaltung.de)

# Generative Design：作例

- グリッド内でのシンプルな円の配置：[Link](#)
- マウス位置による円弧の回転と重ね描き：[Link](#)
- 画像をピクセル単位で分析し円で再描画：[Link](#)
- タイポグラフィの一部を動的に再構成：[Link](#)
- ランダムなラインの生成と重なり：[Link](#)

# Vera Molnár ( ヴェラ・モルナール )

- ハンガリー出身，コンピュータアートの先駆者
- 1960年代から生成的な図形に取り組む
- 「規則」と「わずかなズレ」を重視した作品群
-  公式サイト：<http://www.veramolnar.com/>

作品タイトル	内容	関連リンク
(Des)Ordres (1974)	完全な秩序 (ordre) と、それを少しずつ壊していく (désordre) というテーマ。グリッドの構成要素が徐々にズレていく構成。	<a href="#">DAM.org</a>
Interruptions (1968)	グリッドに配置された線分を「意図的に欠けさせる」ことで、視覚的なリズムや緊張感を作り出す	<a href="#">DAM.org</a>
1% de désordre (1976)	99%の規則と1%のズレというテーマで、ほとんど規則通りの構造に“わずかな逸脱”を持ち込む	<a href="#">The Art of Vera Molnar 1947-1974</a>
Structure de Quadrilatères (1972)	正方形や四角形の回転・重なり・変形により、単純な幾何学構造から複雑な視覚効果を生む	<a href="#">The MFAH Collections</a>
Computer Drawings (1970年代)	MolnárがFORTRANを用いて制作した初期のプロッタードローイングの総称。規則性と偶然性の探究	<a href="#">MoMA</a>

# Tyler Hobbs（タイラー・ホッブス）

- 現代の代表的ジェネラティブアーティスト
- 作品「Fidenza」でNFT界でも注目
- 有機的・抽象的な美しさが特徴
- 🌐 公式サイト：[tylerxhobbs.com](https://tylerxhobbs.com)

# Tyler Hobbs：作例（Fidenza）

## Fidenza（フィデンツァ）

- 概要：2021年に発表された，999点からなるジェネラティブアートのシリーズ
- 特徴：「フローフィールド」アルゴリズムを用いて，有機的で流れるような曲線と色彩のバリエーションを生成
- Link：[Fidenza 公式ページ](#)

# 比較まとめ

作家	特徴	技法例
Generative Design	デザイン寄り，実装例が豊富	グリッド・反復・配色制御
Vera Molnár	美術・理論の重視，初期世代	幾何学・ズレ・規則の操作
Tyler Hobbs	現代的・自然な造形	色・形・構造の融合

# 今日の課題

## 『感情』や『音楽』をテーマに

- 自分だけのデジタルビジュアル作品を制作
- テーマ例：好きな曲，気分，思い出
- 色，形，動きの組合せで気持ちを表現



# 制作のヒント

- `random()` で偶然性を加える
- `mouseDragged()` や `touchStarted()` で操作を加える
- カラーコードや円・線の組合せで自分らしさを
- 背景・透明度で“余白”を表現

# 振り返り

- 今日使った関数を思い出してみよう
- 今日のプログラムは何を表していた？
- 「感情」や「音楽」を表す工夫はできた？

# 参考リンク

- [p5.js公式サイト](#)
- [p5.js日本語公式サイト](#) ※少し情報が古いかも
- [p5.jsリファレンス](#)
- [p5.js 初めの一步 Creative Coding p5.js - HIM.CO ヒム・カンパニー](#)
- [文系大学生のためのp5.js入門](#)

# 【付録】 使用した関数まとめ

# 基本構成

```
function setup() {  
  createCanvas(400, 400);  
}  
function draw() {  
  background(255);  
}
```

# 色・背景・線

```
fill(r, g, b);  
stroke(r, g, b);  
background(gray);  
noStroke();
```

# 図形

```
rect(x, y, w, h);  
ellipse(x, y, w, h);  
line(x1, y1, x2, y2);
```

# マウス操作

```
mouseX, mouseY  
mousePressed()  
mouseDragged()
```



# ランダム

```
random(255);  
random(width);
```

# アルファ（透明度）

```
fill(r, g, b, alpha);
```

# テキスト

```
text("Hello", x, y);  
textSize(24);
```

# おつかれさまでした！

次回はセンサーなどを使ったインタラクティブアートに挑戦！

好きな作品を探してみよう！