

クリエイティブコーディングとRuby学習

12/5 RubyWorld Conference 2024

Miyuki Koshiba



koshiba chobishiba

SmartBank, Inc.

サーバーサイドエンジニア

 @chobishiba

 @ksbmyk



B/43

家計簿初心者におすすめ

シンプルで使いやすい チャージ式カード付き 家計簿プリカ



Google Play
ベストオブ2022
大賞受賞
「生活お役立ち部門」



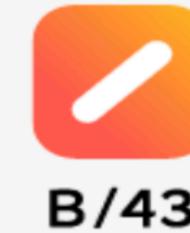
App Store
レビュー数2万件
★4.6
※2024年8月現在



約6万円
改善例も!

2024年度 応募企業様・サービス事例のご紹介

| | | |
|--|--|--|
|  株式会社IVRy |  アウモ株式会社 |  ourly株式会社 |
|  イタンジ株式会社 |  エースチャイルド株式会社 |  株式会社オーディオストック |
|  株式会社SmartHR |  株式会社スマートバンク |  株式会社タイミー |
|  株式会社TwoGate |  株式会社バイタルリード |  バルテス株式会社 |
|  株式会社マイベスト |  株式会社mov |  株式会社ラクーンコマース |
|  株式会社ラクーンフィナンシャル |  株式会社Leaner Technologies |  株式会社Rehab for JAPAN |
|  株式会社リブセンス |  リンカーズ株式会社 |  株式会社ロッカ |



B/43 (ビー ヨンサン)

スマートバンクは、Visaプリペイドカードと家計簿アプリがセットになった家計管理サービス「B/43」を提供しています。B/43は急成長するFintechサービスの基盤システムにRuby on Railsを採用して開発を進めてきました。今後も、多くの人のキャッシュレス生活を支えるようなスケールアップで堅牢なシステムをRuby on Railsで構築できることを証明していきたいと思えます。

 株式会社スマートバンク

SMART BANK シリーズB | 40.8億円調達

資金調達後にやりたい事に対して
「エンジニア」が足りんです!!

LT座談会

2024.12.18 水 オフライン

19:00 - 21:30 @ 虎ノ門



家計管理 × 生成AI



開発ロード
マップ大解説

ぶっちゃけ
トークはお酒と!



クリエイティブコーディングとRuby学習

- Rubyの入門方法としてのクリエイティブコーディング活用方法
 - クリエイティブコーディングとは
 - Rubyの学習にどう活かせるか
 - ワークショップという提案

クリエイティブコーディングとは

- **広義**

- プログラミングはすべからく創造的な行為
- プログラミング全般を指す

- **狭義**

- コードでビジュアルアート作品を作る行為
- 静止画だけでなく、動画やインタラクティブなもの、実行するたびに結果が異なるような作品も作ることができる

クリエイティブコーディングとは

```
<html>
  <head>
    <title>RubyWorld Conference</title>
    <script src="https://cdn.jsdelivr.net/npm/ruby-3_2-wasm-wasi@next/dist/browser.script.iife.js">
  </script>
    <script src="https://cdn.jsdelivr.net/npm/p5@1.5.0/lib/p5.js"></script>
    <script type="text/ruby" src="https://raw.githubusercontent.com/ksbmyk/sketch/main/public/p5.rb">
  </script>
    <script type="text/ruby">
      def setup
        createCanvas(600, 600)
        background(255)

        center_x = width / 2
        center_y = height / 2

        # 外側の六角形
        outer_radius = 200
        draw_hexagon(center_x, center_y, outer_radius, "#9fa0a0", 15)

        # 頂点を結ぶ線
        draw_lines(center_x, center_y, outer_radius, "#cbcbcc", 3)

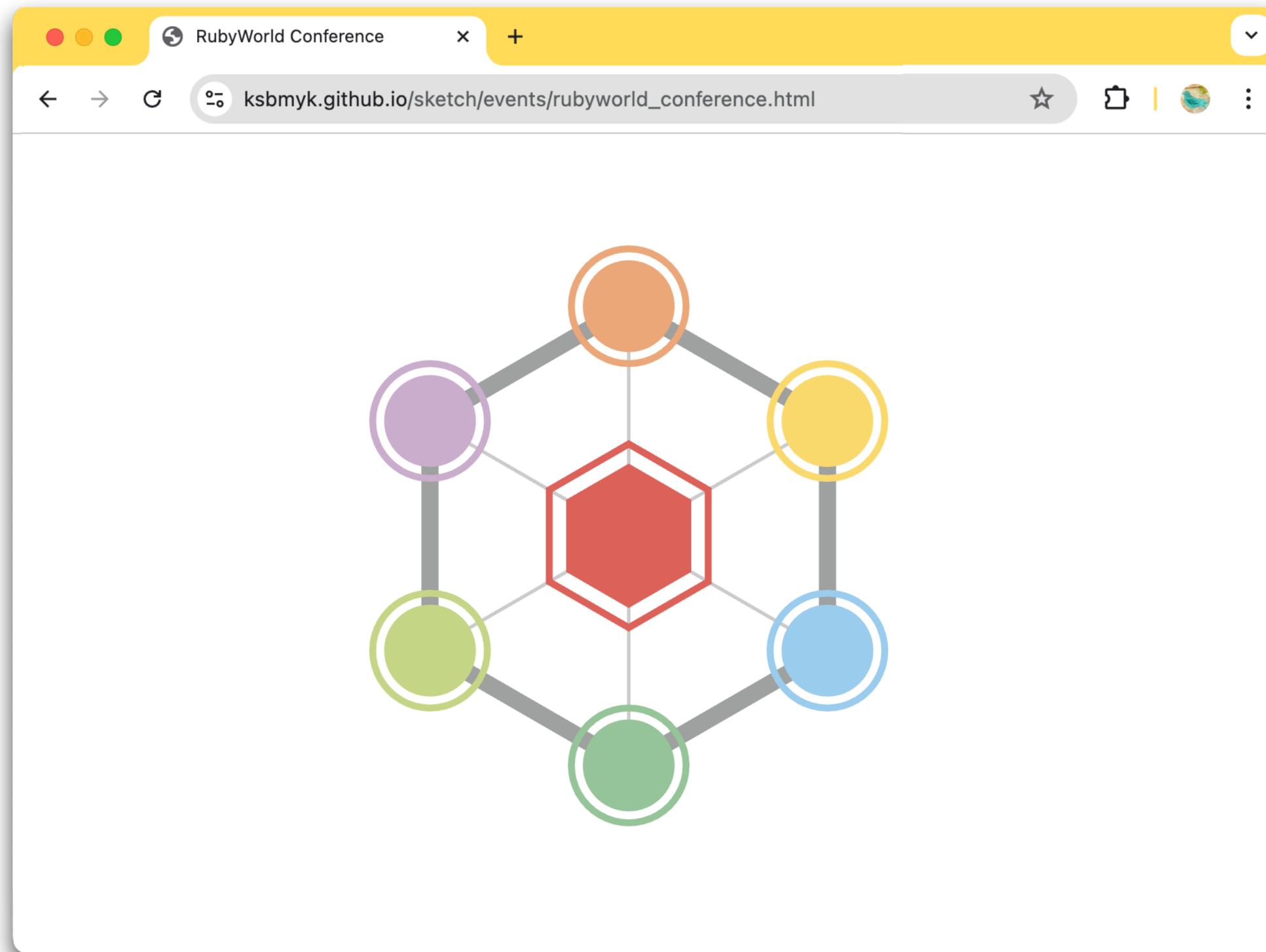
        # 外側の六角形の各頂点に円
        draw_circles_at_vertices(center_x, center_y, outer_radius, 80, 100)

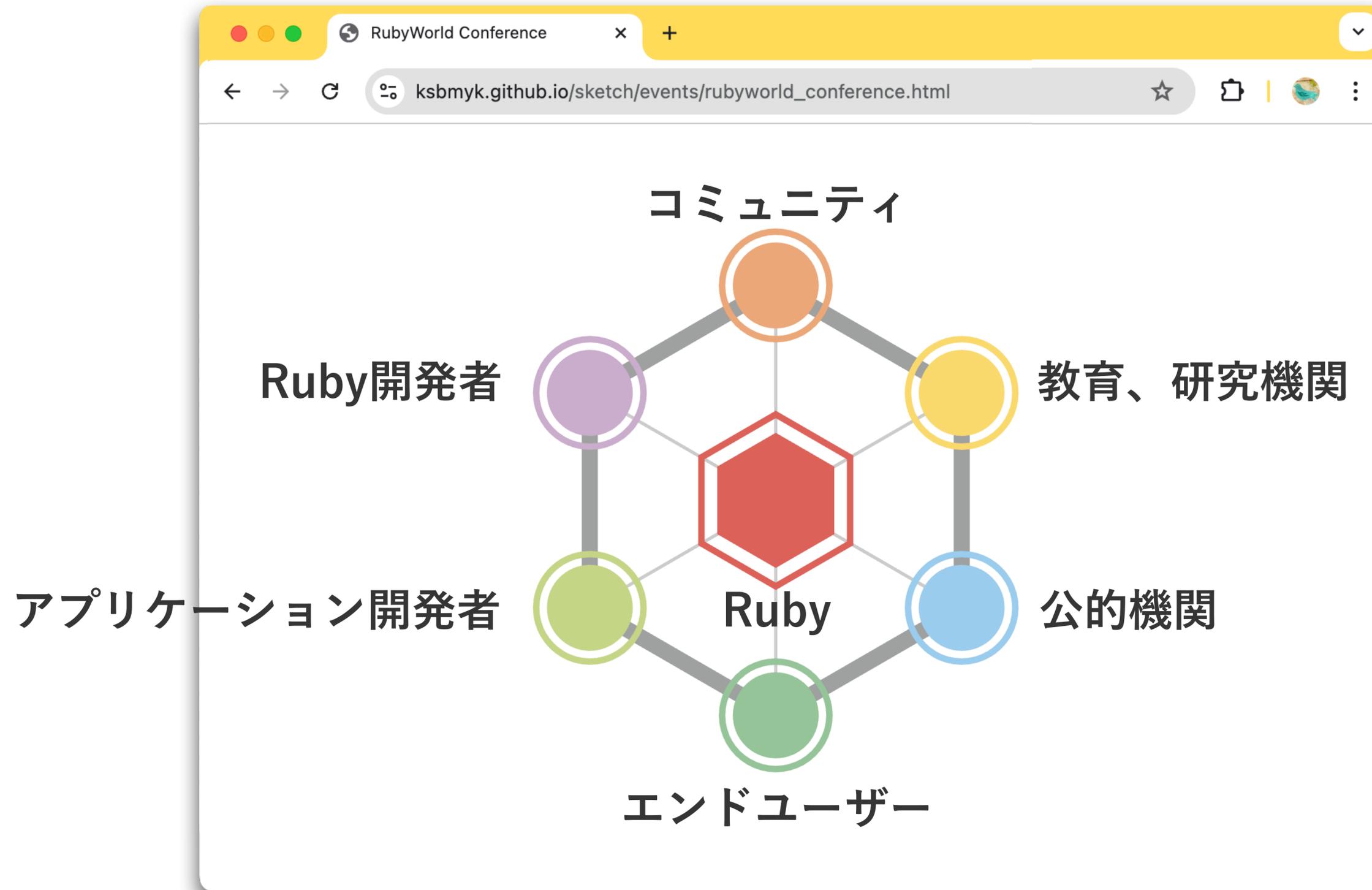
        # 中心の六角形
        middle_radius = 80
        draw_hexagon(center_x, center_y, middle_radius, "#dc6159", 6)

        # 内側の六角形
        inner_radius = 60
        draw_hexagon(center_x, center_y, inner_radius, "#dc6159", 5, true)
      end
    </script>
  </head>
</html>
```

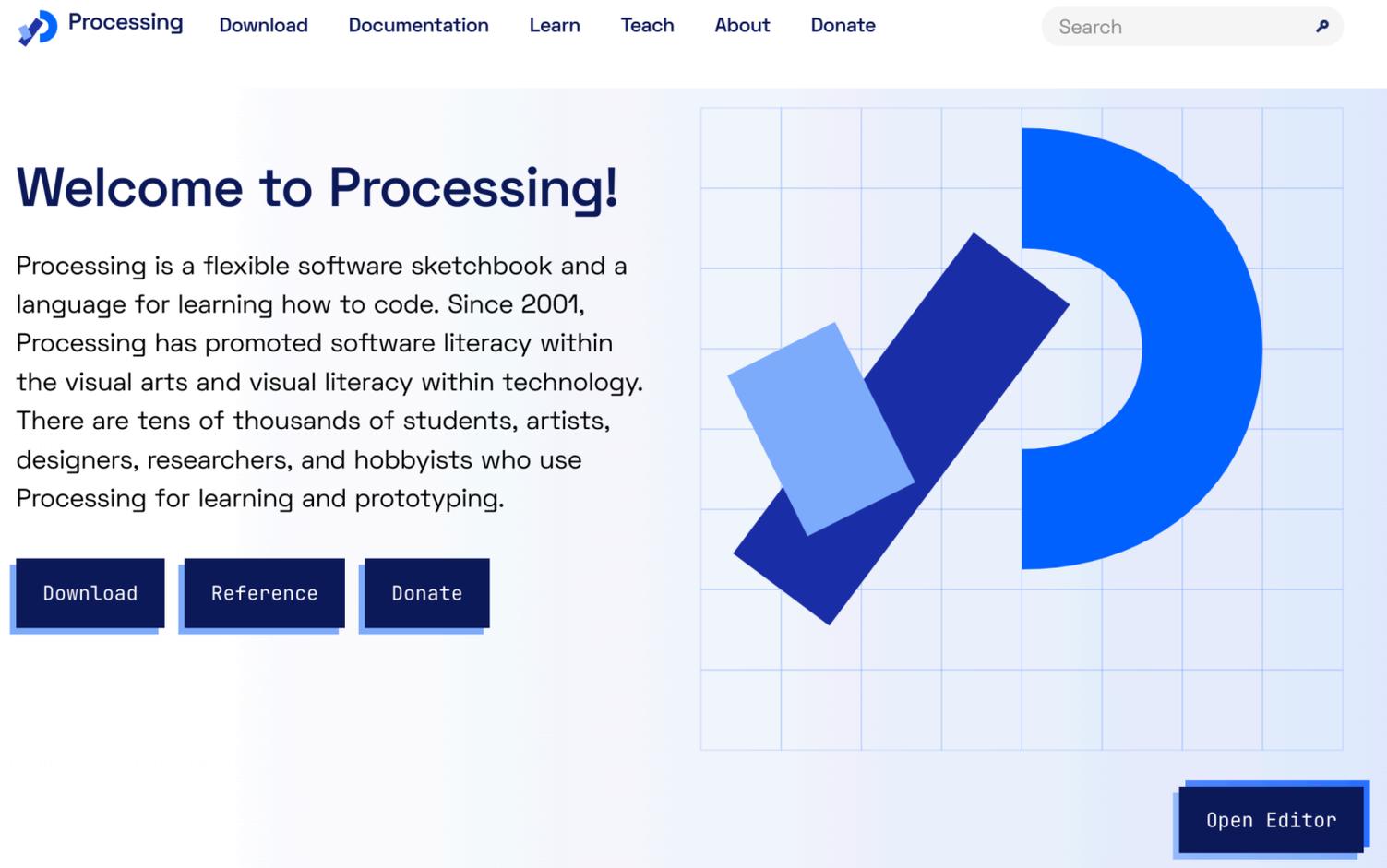
html

Ruby





- Ruby + p5.jsで作成
- p5.js : ProcessingをJavaScriptに移植したもの
- Processing : アートやビジュアルデザイン、インタラクティブな作品を作成するためのプログラミング言語および開発環境



- Processingは、ケイシー・リースとベンジャミン・フライによるオープンソースプロジェクト
- 初心者にとってできるだけシンプルになるように設計
- でも普段使っている言語でやりたい

- 様々な言語でProcessingと組み合わせる方法がつくられてきた
 - JavaScript
 - <https://github.com/processing-js/processing-js>
 - <https://github.com/processing/p5.js>
 - Python
 - <https://github.com/p5py/p5>
 - <https://github.com/py5coding/py5>
 - Go
 - <https://github.com/go-p5/p5>
 - <https://github.com/ryomak/p5go>

ほんの一部

- **2014 ruby-processing** <https://github.com/jashkenas/ruby-processing>
- **2015 processing.rb** <https://github.com/kitao/processing.rb>
- **2019 rp5js** <https://github.com/ksbmyk/rp5js/> 
- **2020 RubySketch** <https://apps.apple.com/jp/app/id1491477639>
- **2022 rbCanvas/p5** <https://rbcanvas.net/p5/>
- **2023 p5.rb Editor** <https://p5rb.ongaeshi.me/editor>

ほんの一部

- 近年安定して動くもの
 - RubySketch
 - rbCanvas/p5
 - p5.rb

ありがとうございます！

RubySketch



RubySketch - Processing 互換の開発環境

Ruby でプログラミングを楽しもう

iPadに対応しています。macOSでは検証されていません。



年齢

4+

歳

カテゴリ



開発ツール

デベロッパ



Noritaka Suzuki

言語

JA

その他1言語

サイズ

87.6

MB

アップデート

- Ruby アソシエーション開発助成金の支援を受け、以下の関数とクラスを追加しました

[さらに表示](#)

[バージョン履歴](#)

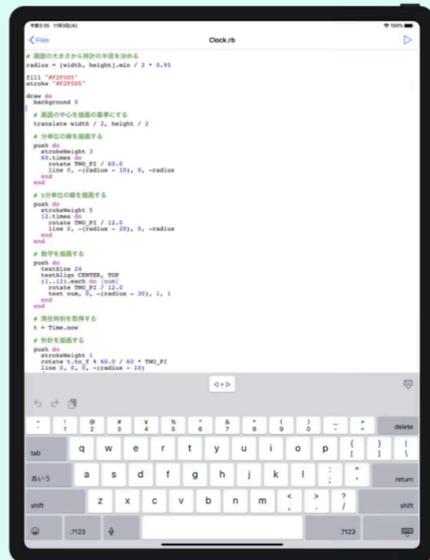
8か月前

バージョン2.10

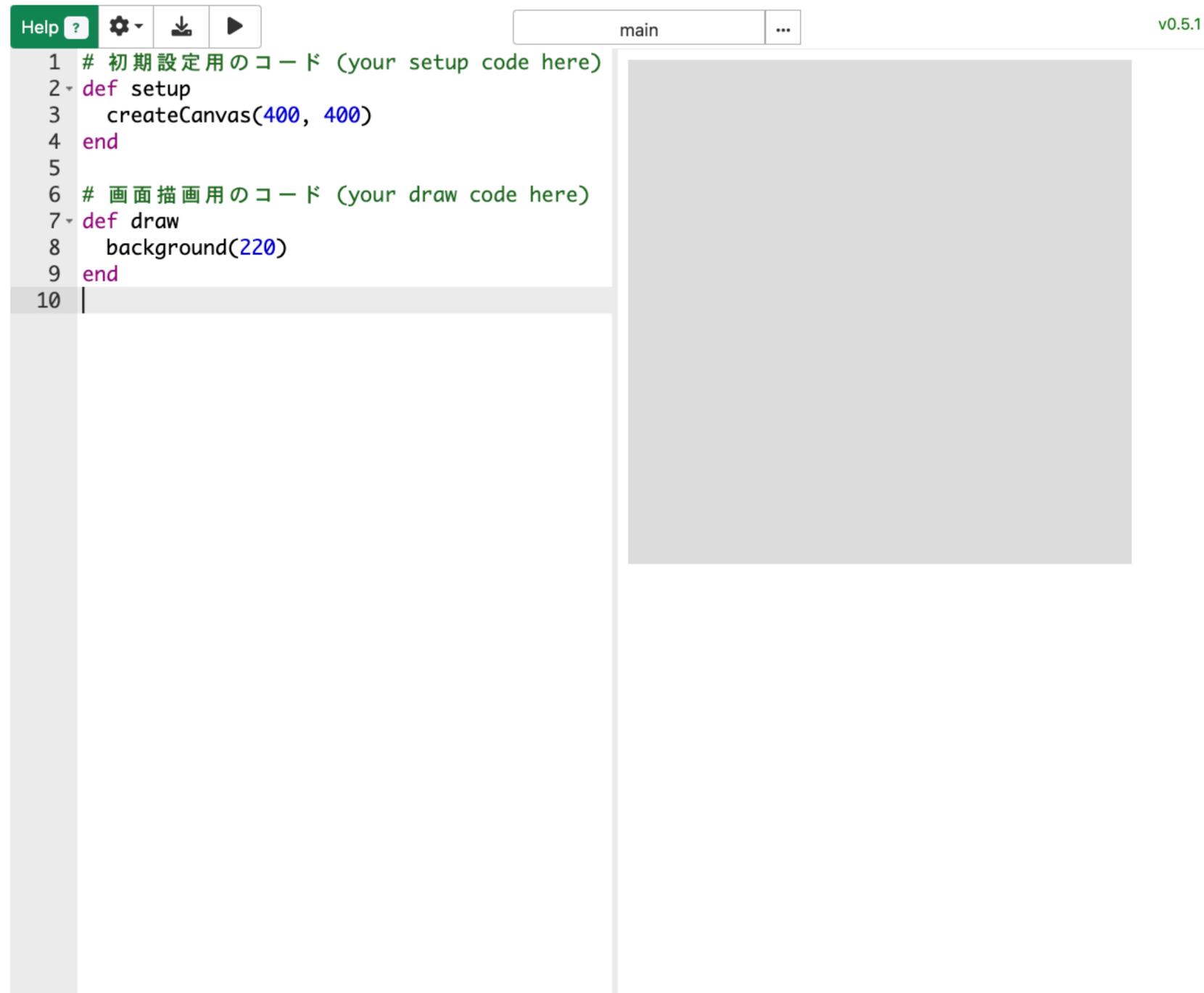
プレビュー

Rubyで
グラフィックス
プログラミングを
楽しもう

Processing 互換 API 搭載

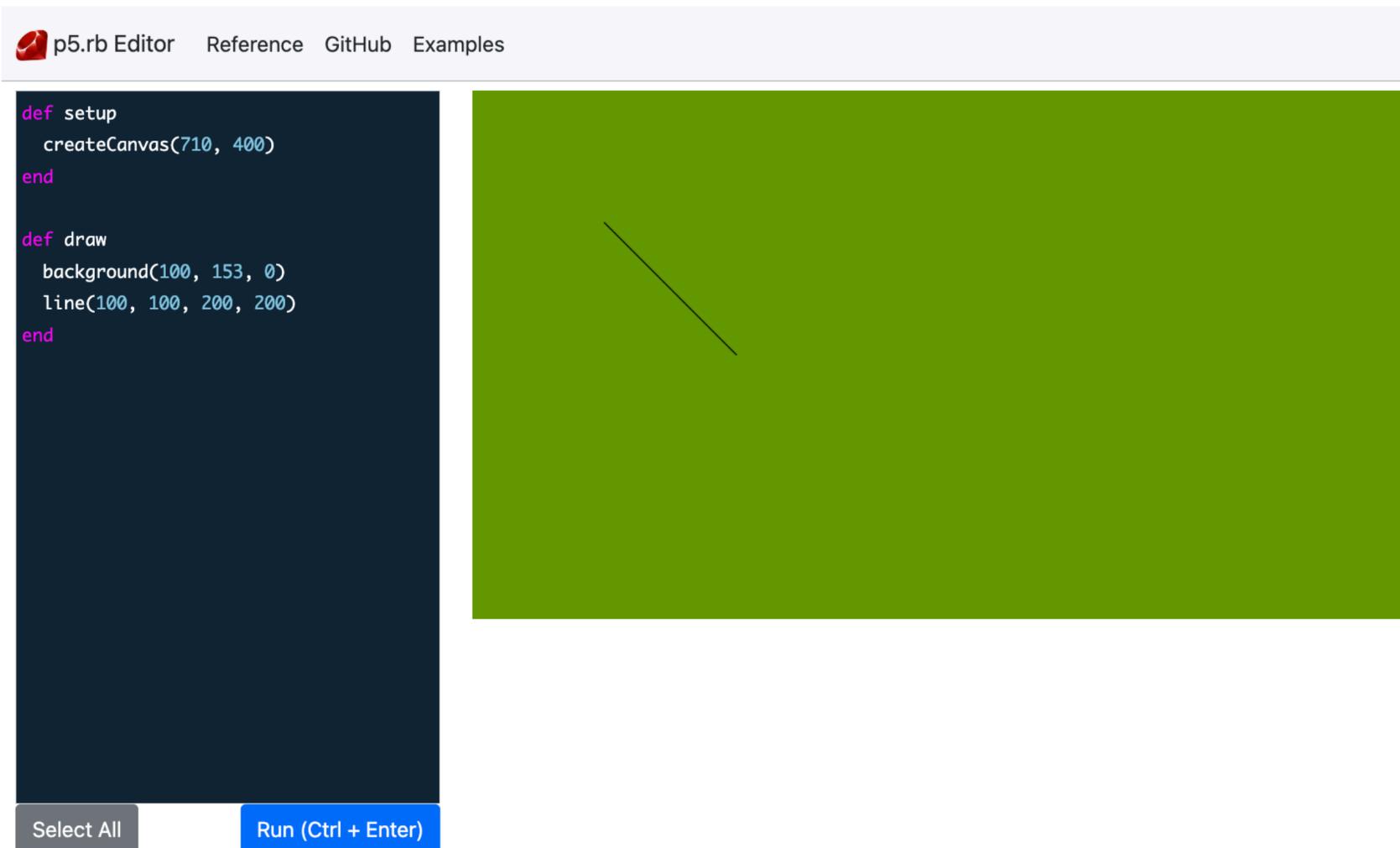


- tokujirosさん作
- iPhone, iPadで利用可能
- ProcessingをRubyに移植したgemが組み込まれた開発環境



```
1 # 初期設定用のコード (your setup code here)
2 def setup
3   createCanvas(400, 400)
4 end
5
6 # 画面描画用のコード (your draw code here)
7 def draw
8   background(220)
9 end
10 |
```

- ブラウザで実行可能
- Opalを用いてRubyからJavaScriptへの変換し、p5.jsと組み合わせて実行
- 通常のRubyと仕様の異なる部分もある
- エラーメッセージがわかりやすい

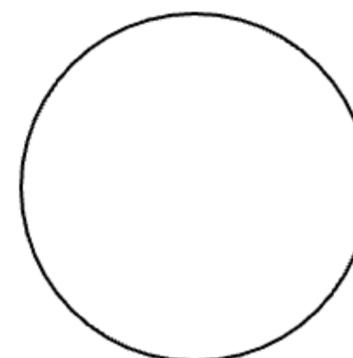


```
def setup
  createCanvas(710, 400)
end

def draw
  background(100, 153, 0)
  line(100, 100, 200, 200)
end
```

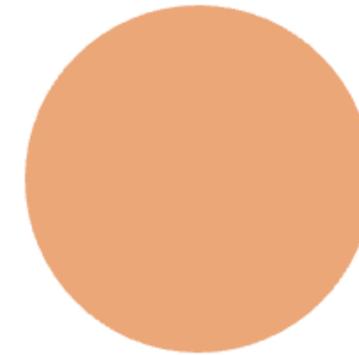
- おんがえしさん作
- ブラウザで実行可能
- ruby.wasm の JS::Object 経由で p5.js のメソッドやプロパティをブリッジしている
- エラーの読み解きが難しい
- いつものRubyで書ける

```
def setup
  # 描画領域のサイズを縦横400に設定
  createCanvas(400, 400)
  # 座標200, 200 に幅120の円を描画
  ellipse(200, 200, 120, 120)
end
```



```
def setup
  # 描画領域のサイズを縦横400に設定
  createCanvas(400, 400)
  # 背景色を設定
  background(255)

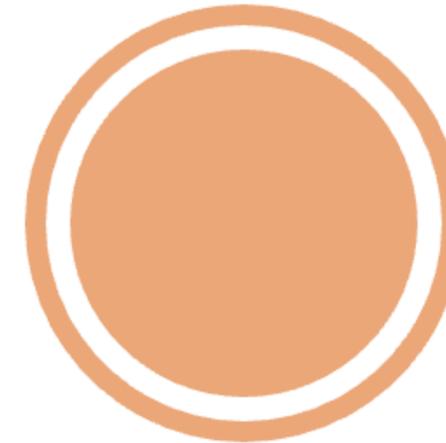
  # 塗りつぶす色を設定
  fill("#eba778")
  # 枠線なしに設定
  noStroke
  # 座標200, 200 に幅100の円を描画
  ellipse(200, 200, 100, 100)
end
```



```
def setup
  # 描画領域のサイズを縦横400に設定
  createCanvas(400, 400)
  # 背景色を設定
  background(255)

  # 塗りつぶす色を設定
  fill("#eba778")
  # 枠線なしに設定
  noStroke
  # 座標200, 200 に幅100の円を描画
  ellipse(200, 200, 100, 100)

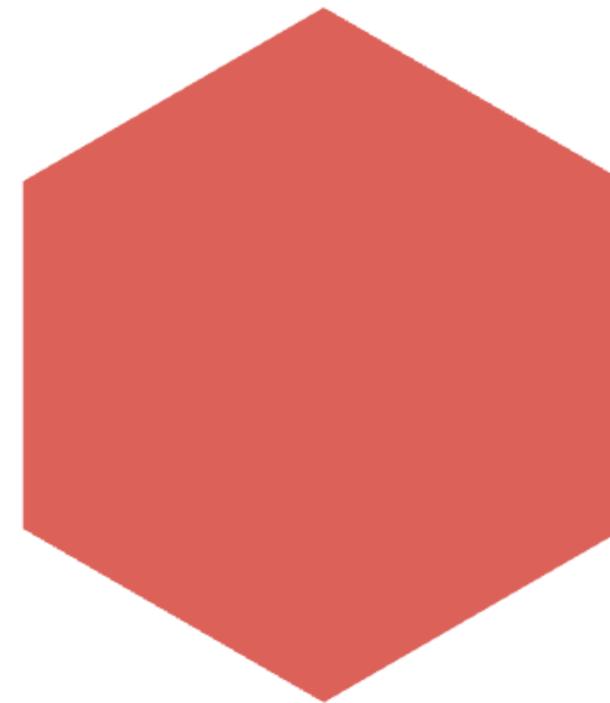
  # 塗りつぶしなしに設定
  noFill
  # 枠線の色を設定
  stroke("#eba778")
  # 枠線の太さを設定
  strokeWeight(6)
  # 座標200, 200 に幅120の円を描画
  ellipse(200, 200, 120, 120)
end
```



```
def setup
  createCanvas(400, 400)
  background(255)

  # 塗りつぶす色を設定
  fill("#dc6159")
  # 枠線なしに設定
  noStroke

  beginShape
  6.times do |i| # 六角形の6つの頂点
    angle = PI / 3 * i - PI / 6 # 30度回転させて、頂点が上にくるように
    x = 200 + cos(angle) * 100 # 頂点のx座標
    y = 200 + sin(angle) * 100 # 頂点のy座標
    vertex(x, y)
  end
  endShape(CLOSE) # 六角形を閉じる
end
```



組み合わせると描ける

```
def setup
  createCanvas(600, 600)
  background(255)

  center_x = width / 2
  center_y = height / 2

  # 外側の六角形
  outer_radius = 200
  draw_hexagon(center_x, center_y, outer_radius, "#9fa0a0", 15)

  # 頂点を結ぶ線
  draw_lines(center_x, center_y, outer_radius, "#cbcbcc", 3)

  # 外側の六角形の各頂点に円
  draw_circles_at_vertices(center_x, center_y, outer_radius, 80, 100)

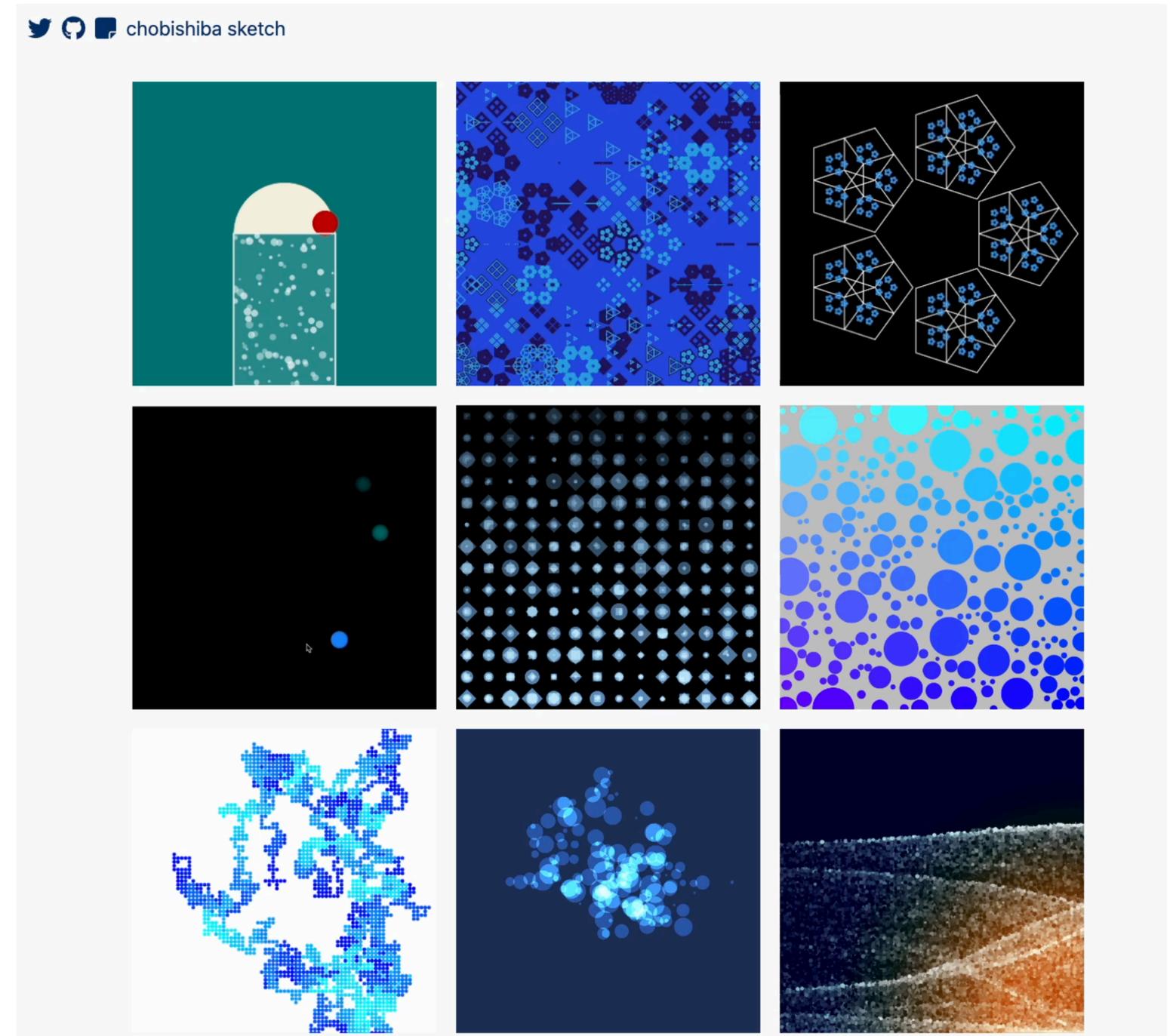
  # 中心の六角形
  middle_radius = 80
  draw_hexagon(center_x, center_y, middle_radius, "#dc6159", 6)

  # 内側の六角形
  inner_radius = 60
  draw_hexagon(center_x, center_y, inner_radius, "#dc6159", 5, true)
end

# 六角形の頂点を計算
def calculate_hexagon_vertices(x, y, radius)
  (0..5).map do |i|
    angle = PI / 3 * i - PI / 6
    { x: x + cos(angle) * radius, y: y + sin(angle) * radius }
  end
end
```

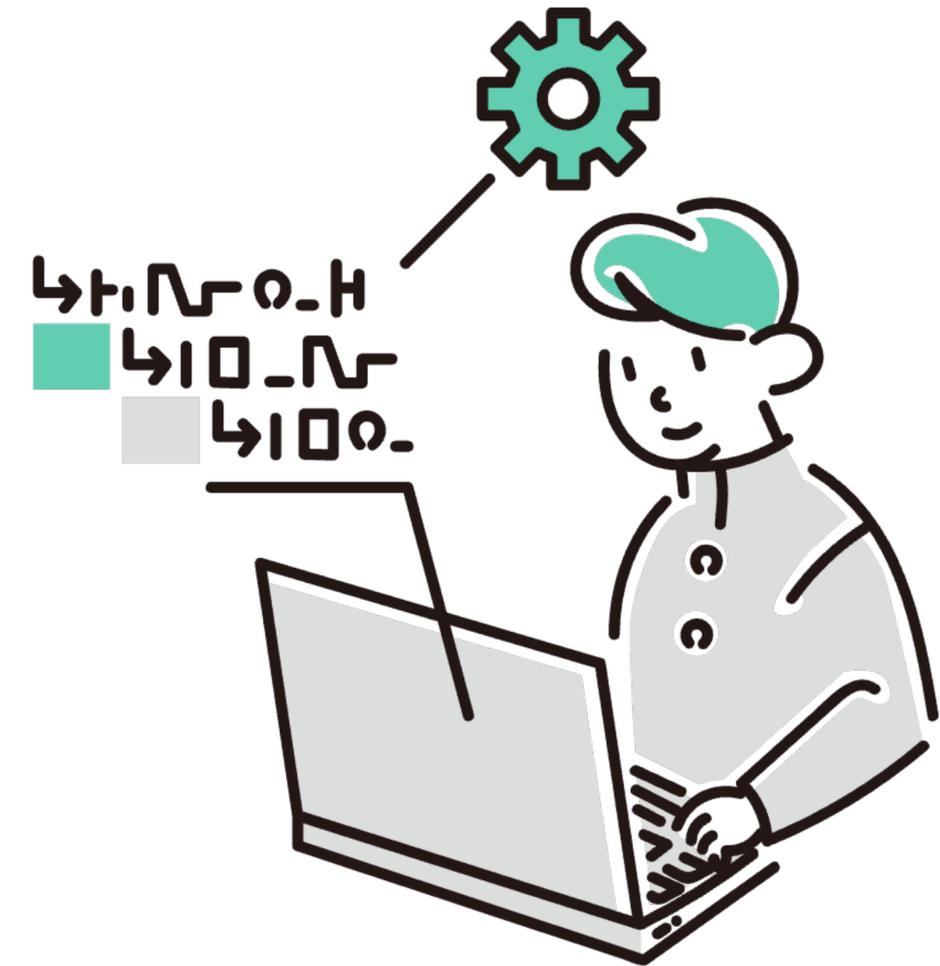


- 静止画、動画、インタラクティブ
さまざまなものが作れる
- 最終的に描画されればよいので重
いことはさほど悪いことではない
- 見ると作るはまた違うので体験し
てもらいたい



Rubyの学習

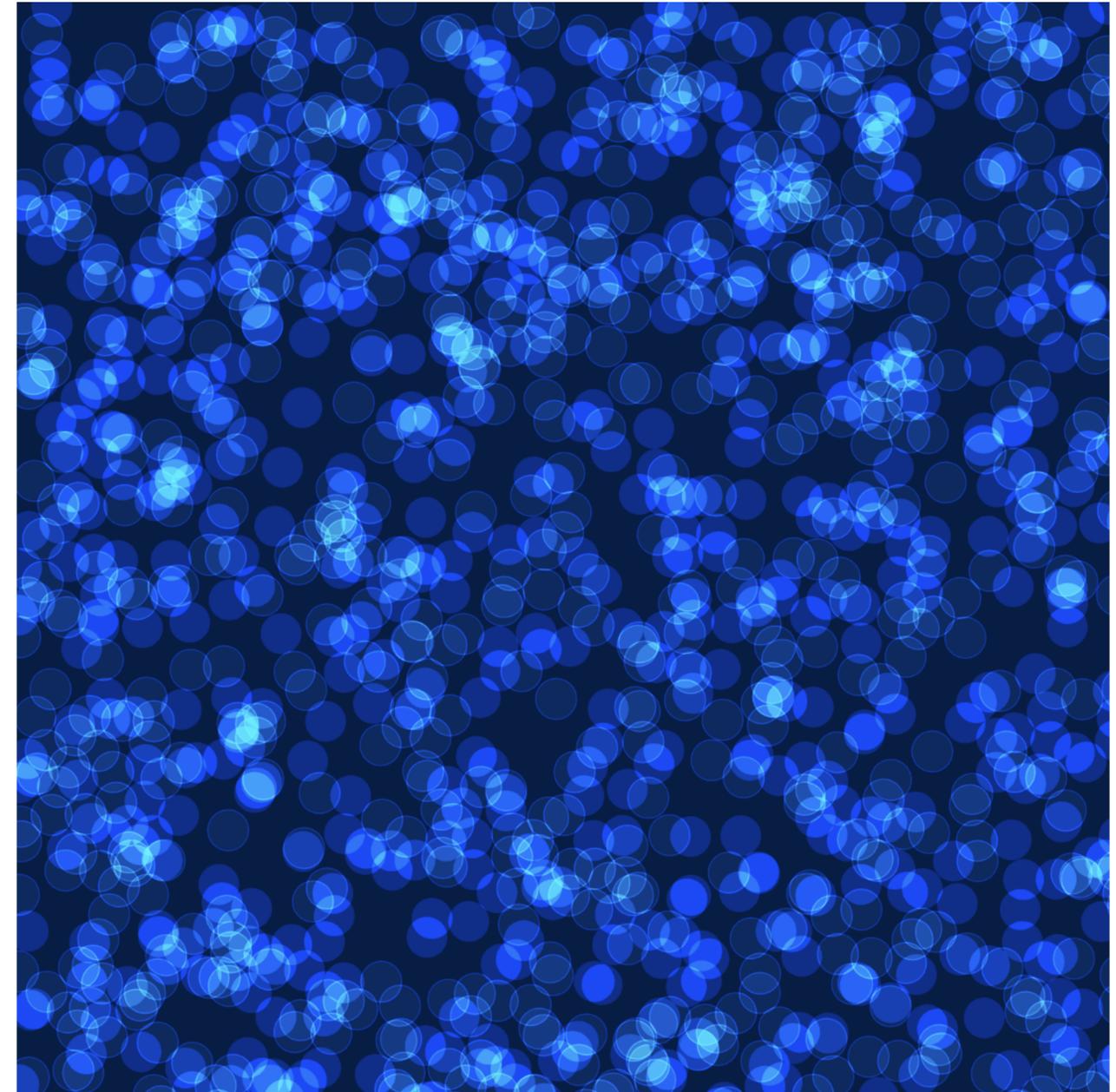
- プログラミングの基礎を体験とともに学んでいきます
 - 変数
 - 条件分岐
 - 繰り返し
 - 関数
 - プログラムの実行される順



```
def setup
  createCanvas(800, 800)
  colorMode(RGB)
  background(0, 30, 70)
  blendMode(OVERLAY)
end

def draw
  noLoop
  1500.times do |i| 1500回繰り返す
    if i.even?
      noStroke
      fill(170, 200, 255)
    else
      stroke(170, 200, 255)
      fill(255, 255, 255, 100)
    end
    ellipse(rand(0..900), rand(0..900), 30)
  end
end
```

分岐



円を1500回ランダムな位置に描いて、偶数回と奇数回で描画効果を変える。たった20行でできる。

Ruby

```
1500.times do |i|  
  # 省略  
  ellipse(rand(0..900), rand(0..900), 30)  
end
```

- 1500回(times) 繰り返す

Processing(Java)

```
for (int i = 0; i < 1500; i++) {  
  // 省略  
  ellipse(random(0, 900), random(0, 900), 30)  
}
```

- intとは整数型で型とは…
- 0からはじまり1500まで1ずつカウントアップして…

```
def setup
  createCanvas(500, 500)
  angleMode(DEGREES)
  background(255)
  noLoop
end

def draw
  bridge = Proc.new do |options|
    stroke(0)
    fill(options[:color])
    arc(width/2, height/2+30, options[:size], options[:size], 180,
36end

  colors = ['#eeb3b2', '#d5e8c5', '#bca7c7']
  radius = 210

  5.times do |i|
    color = colors[i % colors.length]
    options = { color: color, size: radius }
    bridge.call(options)
    radius = radius - (- 5*i + 30)
  end

  options = { color: #ffffff, size: 110 }
  bridge.call(options)
end
```

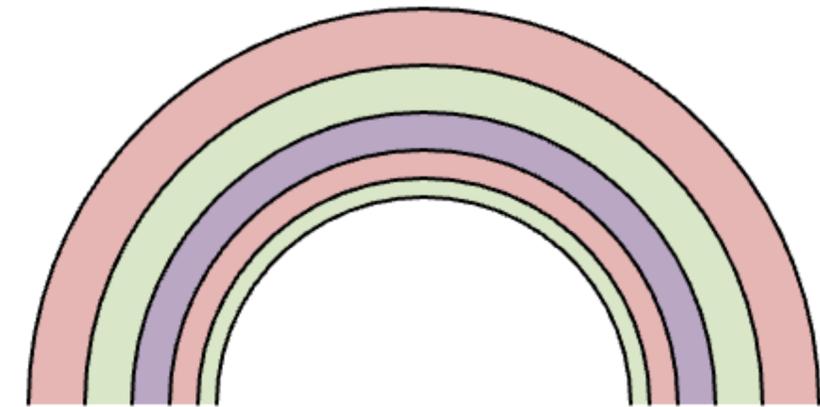
Proc

Array

Hash

Integer

String



各ワークショップ班の名前がRubyのクラス。そのクラスを使って、イベントロゴの一部を描く。

クラスを作る

```
def setup
  createCanvas(720, 720)
  @particles = 50.times.map { Particle.new rand(0..width), rand(0..height) }
end

def draw
  background(0)
  @particles.each do |particle|
    particle.update
    particle.show
  end
end
```

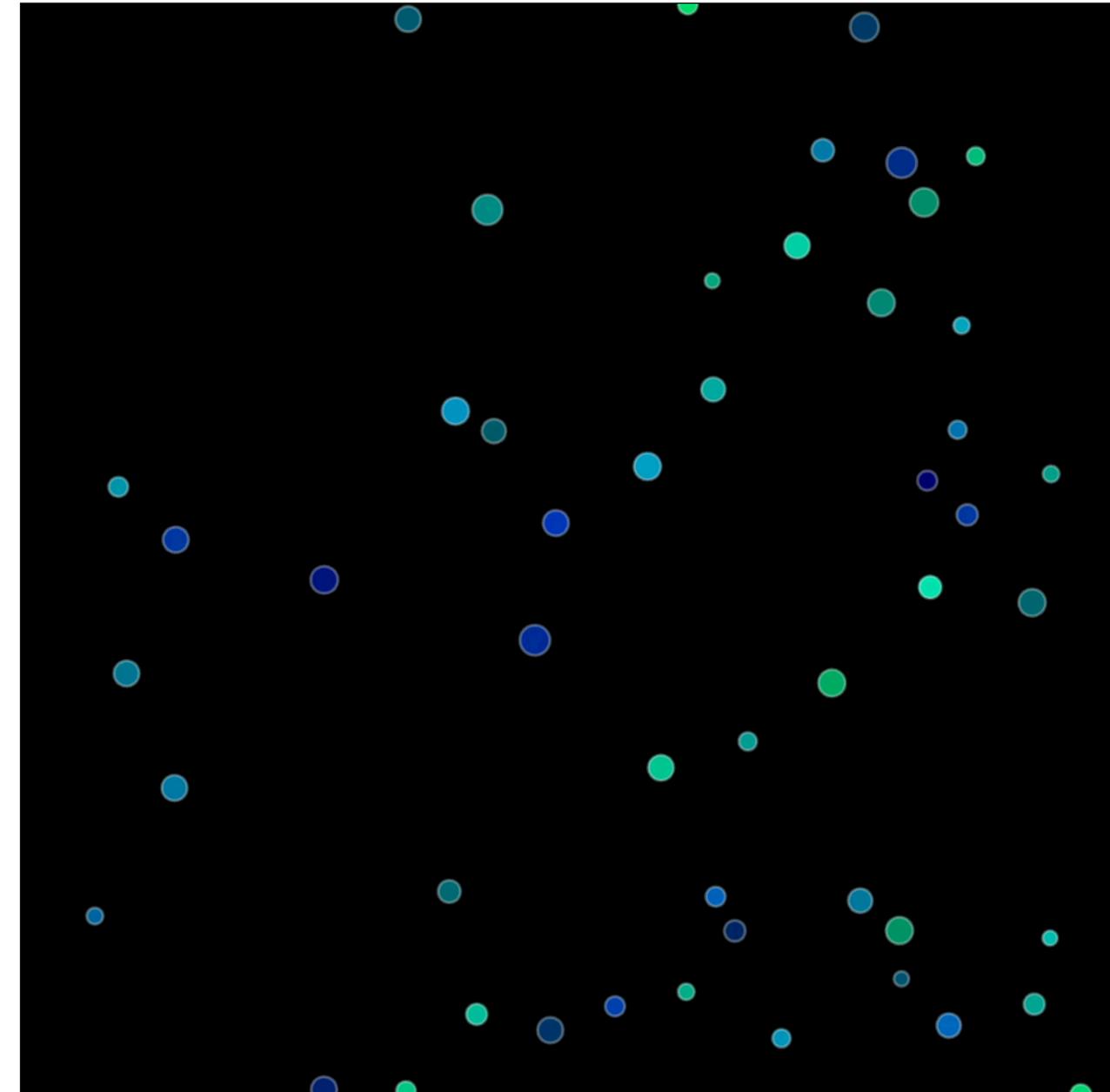
```
class Particle
  attr_accessor :pos, :vel, :acc, :maxspeed, :prev_pos, :size, :color_setting

  def initialize(x, y)
    @pos = createVector(x, y) # 位置
    @vel = createVector(rand(0.1..3.0), rand(-2.0..3.0)) # 速度
    @acc = createVector(0, 0) # 加速度
    @maxspeed = 4
    @prev_pos = pos.copy # 前回の位置を初期化
    @color_setting = color(0, rand(0..255), rand(100..200))
    @size = rand(10..20)
  end

  def update
    vel.add(acc)
    vel.limit(maxspeed)
    pos.add(vel)
    acc.mult(0)

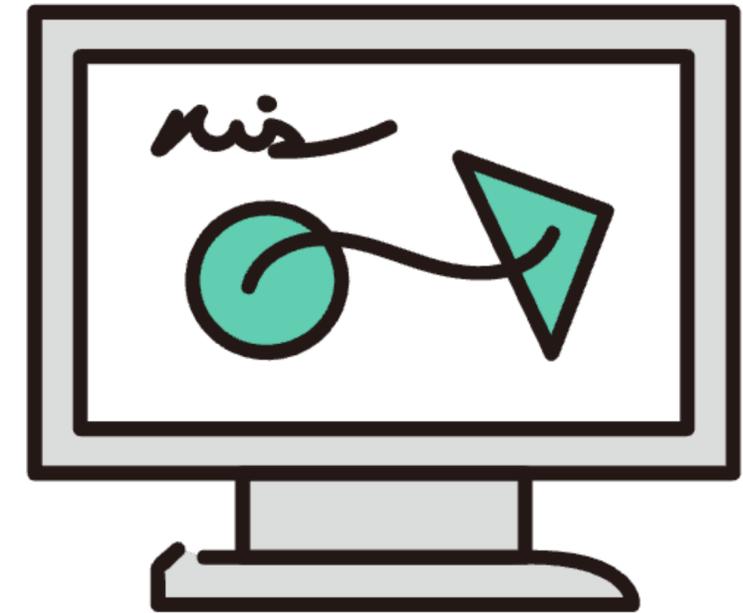
    prev_pos = pos.copy

    if pos.x > width || pos.x < 0 || pos.y > height || pos.y < 0
```



円をオブジェクトとして扱う。位置・速さ・大きさ・色などの属性をもち、属性を変化することで移動しているように見せている

- 画面にビジュアルが描画される
- コマンドプロンプトで実行結果を見るよりもグラフィカルに理解ができる
- 楽しみながら学習できる
- 結果を得るまでのステップが少ない
- Webアプリケーションを作る場合だと画面、バックエンド、DBなどさまざまな準備が必要



- 作れば作るほど次はこうしてみたいという欲求が生まれる
- 目的ができることにより実現するにはどうしたらよいか調べる
- 調べたことを試してみる



- 自分自身が何をやりたいかを考え実行する
- 主体性を育てる機会になる



- プログラミング = 仕事？
- プログラミング = 人の役にたつもの？



それだけではない

- 歌を歌う人が全て歌手ではない
- お菓子を作る人が全てパティシエではない
- 趣味としてのプログラミング
- 日常に彩りを与える行為



ワークショップの提案

- ワークショップ
- **参加者が主体**となった教育であり、その過程や結果を参加者が享受することを目的とするが、その知識や技術の習得や資格の取得などを目的とせず、さらに準備して見守るファシリテータは存在しても、指導して**評価する教師が存在しないもの**

- **ワークショップの効能**

- 一人で始めるにはハードルが高いことでも、サポートのある中であれば安心してチャレンジができる
- やって見たら意外と楽しかったという経験ができればその後の継続に繋がる
- 入門方法に適している

- Waffle Festival 2024
- 女子 & ノンバイナリーの学生向けテック・カンファレンス
- プログラミング入門WSとしてクリエイティブコーディングWSを開催
- p5.jsを使用



主催: 特定非営利活動法人Waffle

- 約2時間
- プログラムの書き方ハンズオン
- 各自好きなものを書いてもらう
- 「プログラミングを体験できて楽しかった」「自分にもできた」

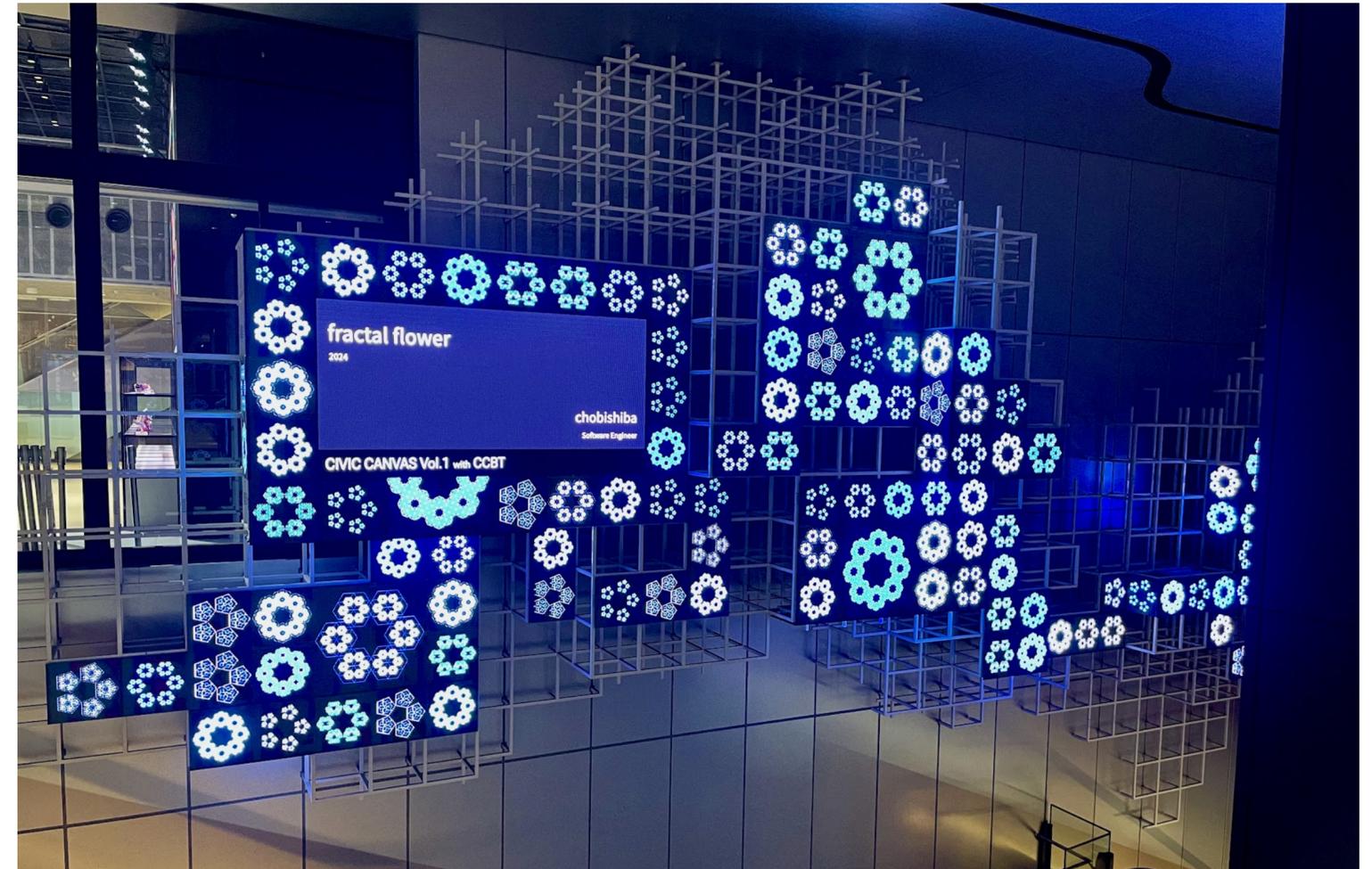


- **CIVIC CANVAS Vol.1**
- 公共の場への展示をアシスト
- プログラミング経験者対象
- 参加する側



主催: 東京都/公益財団法人東京都歴史文化財団/アーツカウンシル東京
/シビック・クリエイティブ・ベース東京 [CCBT]

- 4箇所です8/25～9/7展示
- 「嬉しい」「達成感あった」
- 展示ができるというゴールは強い参加動機になる



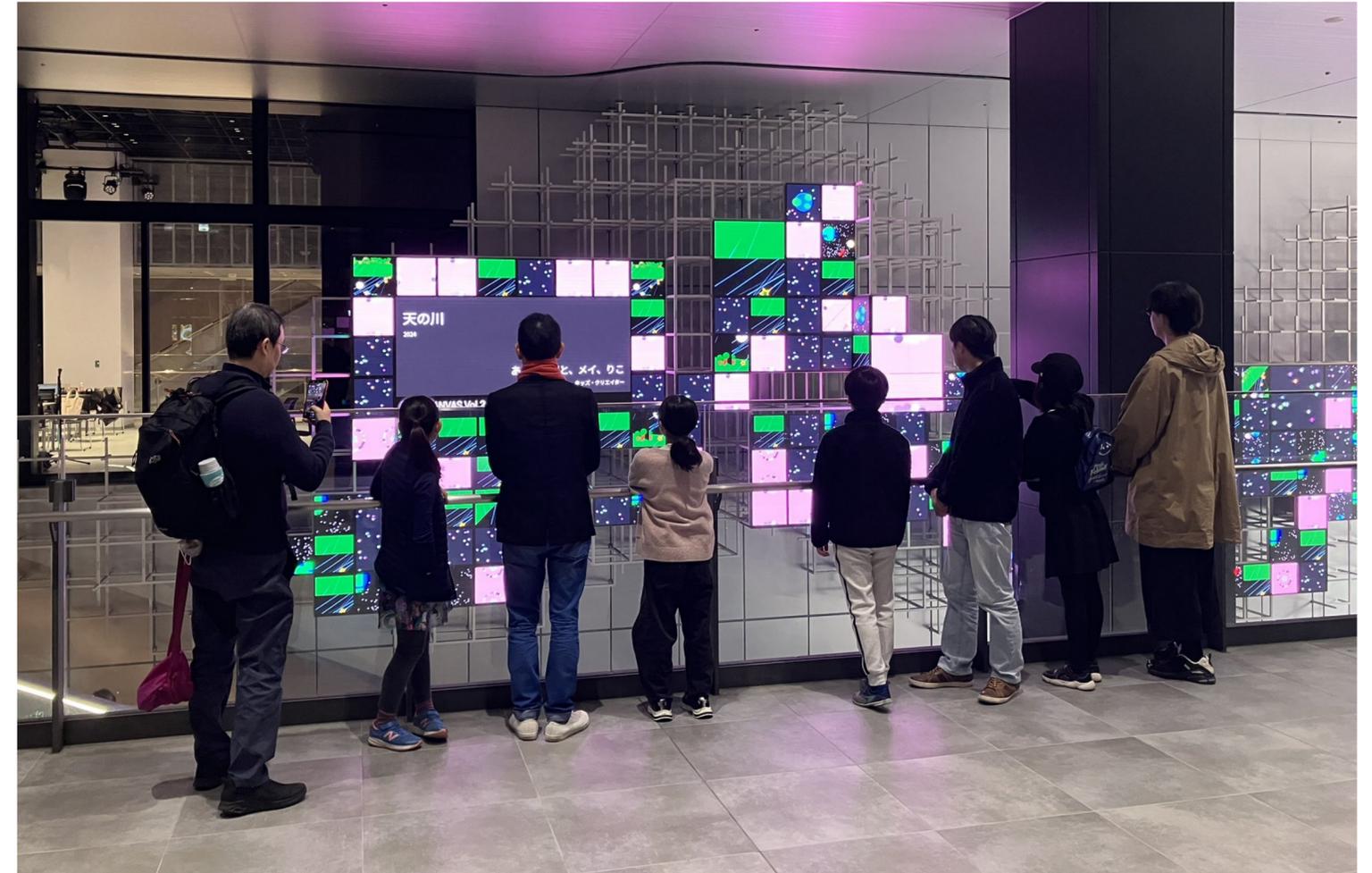
Shibuya Sakura Stage INTER-SQUARE Street vision

- CIVIC CANVAS Vol.2
- 子供向け(小学校4年-6年)
- 生成AIとの関わりを学びながら制作
- ファシリテータとして参加



主催: 東京都/公益財団法人東京都歴史文化財団/アーツカウンシル東京
/シビック・クリエイティブ・ベース東京 [CCBT]

- グループテーマを決めて制作
- 4箇所です12/1～12/15展示
- 「家に帰ってからも続けるには？」 「もっとやりたい」



Shibuya Sakura Stage INTER-SQUARE Street vision

- こういったことをRubyコミュニティでもできないか？

- **ワークショップ開催に必要なもの**
 - 内容
 - ファシリテータ
 - 場所

- 内容作成時に気をつけること
- ゴール設定
 - 終わったあとどんな状態を期待するか
- 説明部分と創意工夫部分の配分
 - 受け身だけではワークショップではない
 - 参加者が主体となる



- 円を使ってかいてみよう
- 円だけでもさまざまなものが描ける
- 変数、関数、分岐、繰り返しを体験できる
- 図形を絞ることでアルゴリズムを作ることにフォーカスできる



- **子供向け、学生向け、プログラミング初心者向け**
 - プログラミングのワークショップ
 - クリエイティブコーディングを題材にしてプログラミングに触れる
- **プログラミング経験者向け**
 - クリエイティブコーディングのワークショップ
 - 趣味としてのRuby
- **Rubyの魅力と出会ってもらう機会**

- ファシリテータ
 - 参加者をサポートする人
 - ワークショップによってはメンターという名称を用いられることもある



- プログラミングをしていく上でエラーは避けて通れない
- しかし最初からエラー続き、解決方法もわからないと嫌になる
- ファシリテータがいれば安心



- どれだけファシリテータを集められるかによって、どれだけ参加者、特に初学者を受け入れられるかが決まる



- **ファシリテータの効能**
 - サポートすることで学べる
 - 自分も体験したことならサポートすることもできる
 - ベテランファシリテータいれば初心者ファシリテータも安心
- 参加者からファシリテータになる

- ワークショップの開催場所
 - 単独で実施
 - カンファレンスなどの中で行う



Rails Girls Tokyo

1st & 2nd March 2024

こんにちは世界！

16回目の Rails Girls Tokyo が開催されます！
Ruby on Rails のすてきな世界を私達と一緒に体験しましょう！

第16回 Rails Girls Tokyo の参加者を募集します。
2日間のワークショップとなります。無料のワークショップ
ですので、お気軽にご参加ください。

参加お申し込みは締め切らせていただきました。
ご応募いただいた皆様ありがとうございました。

[Tweet](#)

- 9:00 - 10:00 **レジストレーション、コーヒー**
金曜日にRuby on Railsのインストールトラブルがあれば、朝のうちに解決しておきましょう。金曜日にすべてうまく行ったひとは参加しなくてもOKです。9:30ごろからのんびりきてください。
- 10:00 - 10:30 **開会**
一日の流れの説明。オーガナイザーから一言。
Friday Hugs!
- 10:30 - 11:00 **自己紹介**
一緒のチームのみんなに自己紹介しましょう！
- 11:00 - 13:00 **ワークショップ - ウェブアプリの構築**
はじめてのウェブアプリにトライしてみよう！
- 13:00 - 13:50 **ランチ & スポンサーLT**
- 13:50 - 15:10 **ワークショップ - ウェブアプリの構築**
- 15:10 - 15:20 **休憩**
- 15:20 - 16:30 **ワークショップ**
自分流のウェブアプリに変えてみよう！

1日かけて実施

RubyKaigi 2019

Schedule Speakers Venue Transport Parties Goodies Sponsors Policies

Schedule

← Back



Masatoshi SEKI
@m_seki

JA

dRuby 20th anniversary hands-on workshop

dRuby (distributed ruby) is a cool library bone in the 20th century.

In this session, we will learn the basics of dRuby. Experience distributed objects on your PC.

Please download (or print) the document. (V0.7) → <http://www.druby.org/fukuoka2019.pdf>

| | | | | |
|---------------|---|---|---|---|
| 14:20 - 15:00 |  <p>The future of the Bundled Bundler with RubyGems Hiroshi SHIBATA @hsbt</p> JA |  <p>Pre-evaluation in Ruby Kevin Deisz @kddeisz</p> EN |  <p>dRuby 20th anniversary hands-on workshop (14:20 - 15:30) Masatoshi SEKI @m_seki</p> JA |  <p>Performance Optimization Techniques of MessagePack-Ruby Sadayuki Furuhashi @frsyuki</p> JA |
| 15:00 - 15:40 | Afternoon Break | | | |

セッションの枠を利用して実施

- 単独で実施（場所を借りて実施）
 - まとまった時間が必要な内容ができる
 - 長時間で疲れてしまうこともある
- カンファレンスの中で行う
 - 短い時間で最初の一歩ができる
 - 興味がなかった、知らなかった人にもリーチできる
 - 短時間だとできることはかぎられてしまう

- Rubyのコミュニティでも増やしていきたい
 - Rubyの魅力と出会う場
 - 楽しんで学ぶ場
 - サポートすることで学ぶ場

- 東京Ruby会議12 前夜祭 第一部
- クリエイティブコーディングワークショップ
- 2025年1月17(金) 16:00～
- 本編のチケット必須
- 参加登録受付中



まとめ

- **クリエイティブコーディングとは**
 - コードでビジュアルアート作品を作ること
- **Rubyの学習にクリエイティブコーディングはどう活かせるか**
 - 楽しみながらRubyの多彩さを体験できる
- **ワークショップという提案**
 - 学びのはじめの一歩として最適

- クリエイティブコーディングやってみませんか？
- クリエイティブコーディングをやってみたい方、興味のある方、ワークショップいいんじゃないかと思う方、この後ぜひお話ししましょう！

