



Girls Who Code At Home

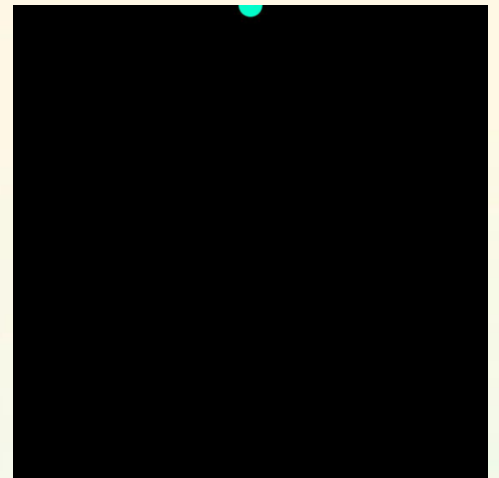
流星キャッチャーゲーム: Part 2

流星をつかまえる

学習概要

お帰りなさい。先週は、p5.jsの環境を調査し、流星キャッチャーゲームのパーツを作る計画を立てましたね。今回は、座標系を使用してゲームの最初のコンポーネントである流星を描画します。また、p5.jsでの色の使い方を学び、流星とスケッチの背景の色を設定します。

このアクティビティに取り掛かる前に、「**流星キャッチャーゲームシリーズ**」の[パート1](#)を終わらせてください。



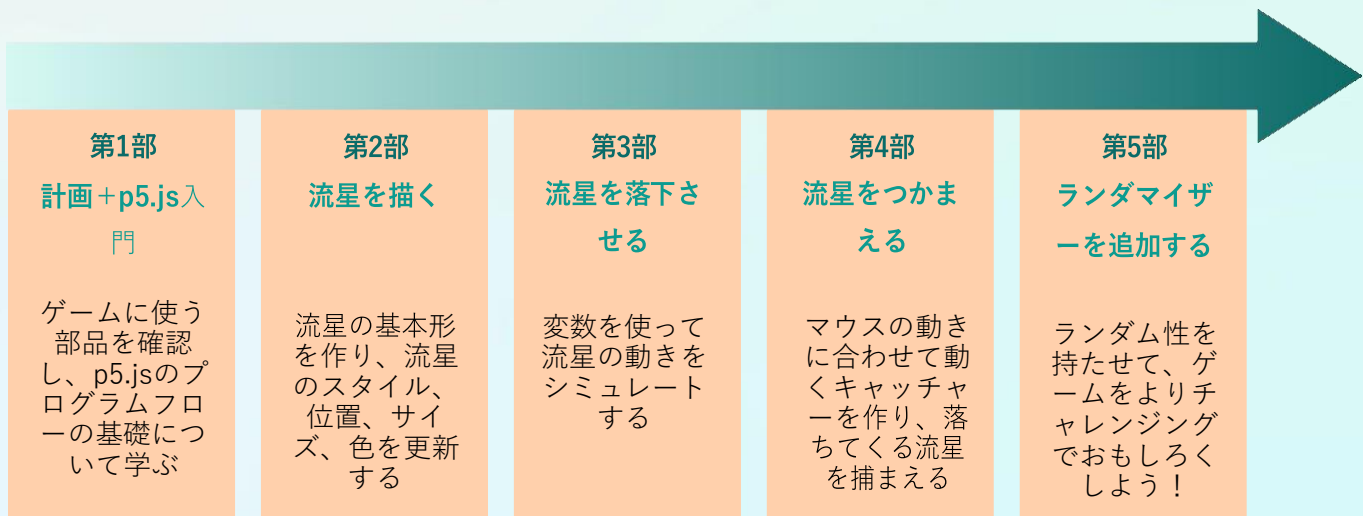
学習目標

このアクティビティが終わるころには、こんなことができるようになっていくことでしょう。

- p5.jsの座標系と、画面上のピクセルとの関係について説明することができる。
- 内蔵されている関数とコマンドを使って、座標平面上に基本的な図形を描くことができる。

マテリアル

- [p5.js オンラインエディタ](#)
- [流星キャッチャーゲーム サンプルプロジェクト](#)
- [流星キャッチャーゲームPart2 参考資料](#)



流星キャッチャーゲームのパート2はビデオチュートリアル**でもご覧いただけます！**

Women in Tech スポットライト：Phoenix Perry



画像の出典はこち

[Hackaday.io](https://hackaday.io)

Phoenixは幼い頃、両親に[アタリ社](#)の家庭用ゲーム機を買ってもらいました。新しいテクノロジーの世界に触れ、Phoenixはテクノロジーに興味を持ち、[BASIC](#)でプログラミングを学び始めました。そして大学卒業後、[Evite](#)のウェブデザイナーとしてキャリアをスタートさせました。ここで彼女は、会社では当たり前の徹夜に何度も耐え、休養不足で手首に負担がかかり、鎮痛剤を飲む生活を送っていました。そして手の骨や靱帯を酷使するプログラマーにはよく見られる神経症候群である[手根管症候群](#)を発症してしまいました。

Phoenixは、体調不良を理由にテクノロジー業界から何年も離れ、アートディレクターとして働いていました。その後、[NYU TandonのIntegrated Digital Media](#)プログラムに出会い、非常勤教授兼研究員となり、ゲーム開発のデザイン、遊び、体現について教えるようになりました。この経験を経て、[Code Liberation](#)の共同設立者となりました。Code Liberationは、女性、ノンバイナリー、フェムなど、自分を女性と認識する人々がSTEAMの分野で仕事を獲得できるよう支援することを目的としています。無料のクラス、ワークショップ、ゲーム大会、ハッカソン、ソーシャルゲームナイトなどを、あらゆる年齢やキャリアのさまざまな段階にある女性たちに提供しています。Phoenixは自身の経験を活かして、次世代の女性がテクノロジー産業で活躍するのに必要なスキルを身につけるための支援を行っています。現在は、[ロンドン芸術大学](#)と提携し、英国にCode Liberationの支部を開設しています。

この[ビデオ](#)では、Phoenixの詳細と、なぜテクノロジーにおいて多様性が重要であるかを紹介しています。Phoenixのことをもっと知りたいですか？この[記事](#)も、Phoenixが今日に至るまでの学問の道のりや苦難について詳しく紹介しています。

考えてみましょう

コンピュータサイエンティストであることは、単にコーディングが得意というだけではありません。Phoenixや彼女の仕事が、偉大なコンピュータサイエンティストは、勇気、レジリエンス（困難や脅威に直面している状況に対して、「うまく適応できる能力」）、創造性や目標にフォーカスする強さを持っていることと、同関連しているのか、考えてみましょう。もっているのです。



勇気

手根管症候群を発症したPhoenixは、仕事もできず、動くこともままならない状態に陥り、テクノロジー業界への復帰に苦労しました。

PhoenixがCode Liberationで他の女性を支援しながら、自身も業界に復帰することで、いかに勇気を体現したかについて議論してください。

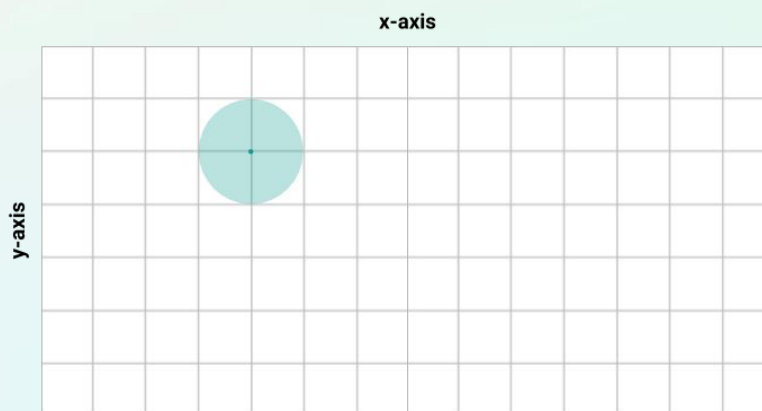
あなたの意見を家族や友人と共有しましょう。他の人にもPhoenixについてもっと読むように勧めて、議論に参加してもらいましょう！

ステップ1：円の場所を決める（2～4分）

例えば、友人から「紙に丸を描いて」と指示を受けたとします。あなたは、ただ円を描くこともできますし、もっと詳しい情報を求めることもできます。もし、もっと情報を求めるのであれば、「紙のどこに描くのですか？大きさは？何色？真円なのか楕円なのか？」というになるでしょう。どこに描くのかという質問に対して友人から「左側から3分の1のところ、下に向かって4分の3のところに描いて」と答えが返ってきました。この指示は、あなたが人間と話しているのであれば有効です。また、具体的である必要はありません。しかし、コンピュータにはそれでは通用しません。座標系を使って、プログラム中に表示したい要素の位置を指定しなければいけません。

座標系とは、空間内の点の位置を特定するために、1つ以上の数値を使用するシステムのことです。座標系は2次元平面上にある場合と3次元空間上にある場合があります。座標平面（つまり2次元）は、水平に走るX軸と垂直に走るY軸を持ち、格子を形成しています。座標系では、点を表すのに（x位置、y位置）という順序のペアを使用します。

p5.jsはz軸を持つ3次元空間でも動作します。p5のドキュメントではいくつかの関数でz軸パラメータとして使用されている例がありますが、使用するべきではありません。



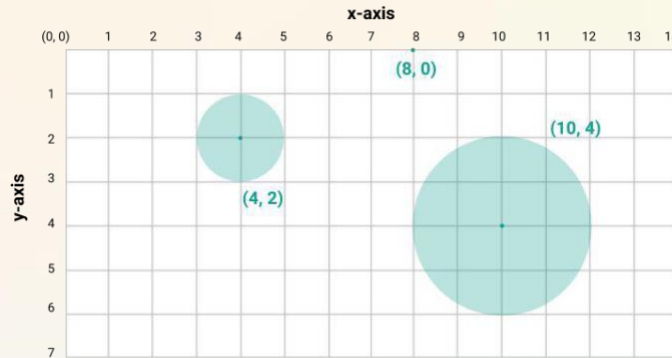
上の座標平面を見て下さい。その場所に円を描くようにコンピュータに指示することを考えてみましょう。



リファレンスガイドの2ページ目で自分のアイデアを確認してください。

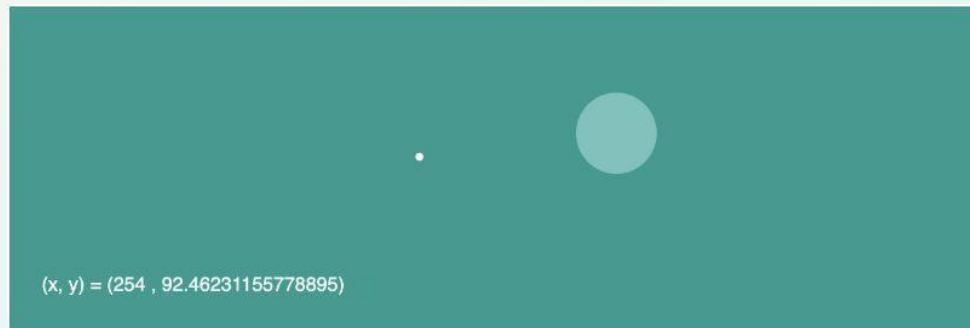
ステップ2：座標系を使う練習（5～8分）

画面上の各ピクセルは、座標系で固有のアドレスを持っています。ピクセルを画面に描画するには、プログラムにピクセルのx 座標（つまりx軸上の位置）とy 座標（つまりy軸上の位置）を与える必要があります。



画面座標系の原点（0, 0）は、左上の隅に位置しています。画面上で右に移動するにつれて、X座標の値は大きくなります。画面上で下に行くほど、Y座標の値は大きくなります。数学の授業で見た、中央や左下に原点があるような座標系レイアウトとは少し違うように見えるかもしれません。

このスケッチを見ながら、マウスを使って円の中心、幅、高さを推定してみてください。



リファレンスガイドの2ページ目で自分のアイデアを確認してください。

ステップ3: プロジェクトスケッチの作成 (5-10分)

残りのセクションで、ゲームのコードを書き始めます。まず、メインプロジェクトのスケッチを作成する必要があります。

- **p5.jsオンラインエディタにログインします。** エディタは、自動的に初めのコードを含む空白のスケッチを開きます。また、[ファイル] > [新規]で新しいスケッチを作成することもできます。



- **鉛筆のアイコンをクリックして、スケッチに名前を付けます。** 「Meteor Catcher Game v1」ような分かりやすい名前を付けましょう。

注：これはツールバーの下に"sketch information area"にあります。

- **[ファイル]で[保存]をクリックします。** キーボードショートカットのCommand S (Mac) または Control S (Windows) を使用して保存することもできます。これらのショートカットを使用する前に、必ずエディタ画面内に移動してください。
 - **スケッチの一番上に、以下の情報を記載した複数行のコメントを作成します。**
 - **プログラムのタイトル：**これはスケッチ名と同じにしましょう。
 - **プログラムのバージョン：**これは最初のバージョンですか、それとも2番目のバージョンですか？大きな変更を加えた場合は、新しいバージョンを作成するようにしましょう。
 - **著者：**By (名前)
 - **説明：**何をするものなのか、1~2文。

スケッチの先頭には、スケッチに関する基本的な情報を提供するコメントを入れます。**コードコメント**は、何がどのように機能するか、決定の理由、またはフォローアップのタスクなどを思い出すために使用します。

→ 一行コメントは、ダブルスラッシュ、`//`を使用します。

→ 複数行に及ぶコメントは、開くときはフォワードスラッシュとアスタリスク、`/*`閉じるときはアスタリスクとフォワードスラッシュ、`*/`を使用します。

```
// これは1行のコメントです
```

```
/*
これは複数行のコメントです
*/
```

ステップ4：流星を描く（3分～5分）

プロジェクトのスケッチが保存されたので、最初のゲームコンポーネントである流星を作りましょう！
まず、特定の位置に図形を描く方法を学び、次に色を付けて見た目を変えてみましょう。

形状・位置

前回、座標系について学んだことを思い出してください。今までの知識からすると、もし図形を描きたいなら、コンピュータに各ピクセルを個別に描くように指示しなければなりません。これは非常に面倒なことなので、p5は私たちが簡単に図形を描けるよう関数があらかじめ用意されています。必要なのは、図形を描きたい場所を指定し、幅と高さを指定することだけです。以下、円を描くための入力法を見てみましょう。

JavaScript	説明
<code>ellipse(x, y, width, height);</code>	<ul style="list-style-type: none">→ ellipse: 楕円を描く関数名。楕円は英語でEllipseまたはOval→ p5.jsのリファレンス→ (): 括弧を使用することで、プログラムに関数を呼び出す時に、関数のパラメータや入力を括弧の中に入れることもある→ x: 楕円の中心の x 座標→ y: 楕円の中心での y 座標→ width: 楕円の幅をピクセル単位で設定する→ height: 楕円の高さをピクセル単位で設定する→ ,: 関数内の異なるパラメータや入力を区切るためにカンマを使用する→ ;: p5.jsのコード行はすべてセミコロンで終える

`ellipse()`関数で流星を画面に描画します

- `draw()` 関数の中に、`ellipse()` 関数をスケッチに追加する
- キャンバスの中央（X軸の位置）、一番上（Y軸の位置）に配置する再確認が必要な場合は上記のグラフを参照する
- 円または楕円の幅を20ピクセル高さ20ピクセルに設定する
- これが流星だと分かるようコメントをつける
- 再生ボタンを押してコードを実行し、テストする



リファレンスガイドの3ページ目でコードを確認することを忘れないでください。

ステップ5：流星を色で塗りつぶす（5～8分）

図形に色をつけるには、**fill()**関数を使います。これは、コマンドの後に描かれた図形を、指定された色で塗りつぶすものです。下の例では、**fill()**関数の後に描かれたので、2つの円は青色になっています。

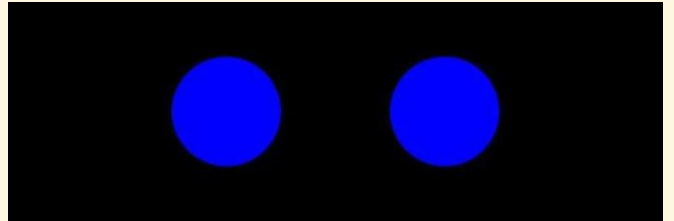
JavaScript

```
function setup()
{
  createCanvas(600, 200);
}

function draw()
{
  background(0);

  // 両方の円を青にする
  fill(0, 0, 255);
  ellipse(width / 3, height / 2, 100, 100);
  ellipse(width / 3*2, height / 2, 100, 100);
}
```

結果



図形に色をつけるのにも、2番目の図形を青ではなく青緑にしたい場合はどうすればよいでしょうか。もうひとつ **fill()** 関数の形を指定する前に追加する必要があります。

JavaScript

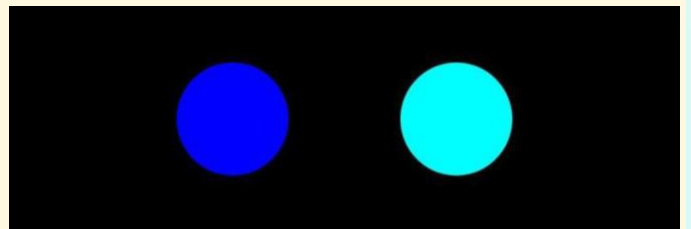
```
function setup()
{
  createCanvas(600, 200);
}

function draw()
{
  background(0);

  // 青い円
  fill(0, 0, 255);
  ellipse(width / 3, height / 2, 100, 100);
  // 青緑色の円
  fill(0, 255, 255);
  ellipse(width / 3*2, height / 2, 100, 100);
}
```

結果

この変更により、青と水色の丸が1個ずつ表示されるはず。



fill()メソッドは、赤、緑、青の光の組み合わせで色のスペクトルを作成するRGBカラーモードを使用します。

- ➔ (0,0,255) を指定すると青色で表示されます。
- ➔ (0,255,255) を指定すると、青緑色になります。

ステップ5：流星を色で塗りつぶす（続き）

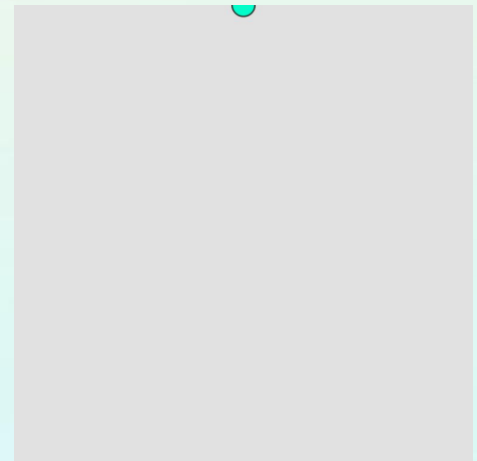
RGBカラーモードを使用した**fill()** 関数の入力を以下で確認します。RGBカラーモードは、赤、緑、青の光の組み合わせを使用して、さまざまなデジタルカラーを作成します。各色には0から255までの値を割り当てることができます。例えば、（255, 0, 0）は赤、（0, 0, 0）は黒、（255, 136, 0）はオレンジとなります。

[Color Pickers](#)や[Coolors](#)などのパレットツールを使って、さまざまな値や色で遊ぶことができます。

JavaScript	説明
fill (redValue, greenValue, blueValue);	<ul style="list-style-type: none">→ fill: 関数名→ (): 括弧を使用して、プログラムに関数を呼び出す必要がある時に関数のパラメータや入力を括弧の中に入れることもある→ redValue: 0～255の緑の値→ blueValue: 0～255の緑の値→ greenValue: 0～255の緑の値→ :: p5.jsのコード行はすべてセミコロンで終える

fill() 関数を使って、流星に色をつけます。

- ☐ 流星の色を選び、RGB値をメモしておく
- ☐ 流星のRGB値を**fill()** 関数で呼び出す
（位置が重要であることを忘れずに！）
- ☐ 終わったら再生ボタンを押して、プログラムを実行する。流星の色が指定した色に変わるはずですよ。



リファレンスガイドの3ページ目でコードを確認することを忘れないでください。

ステップ6：アウトラインの変更（3～5分）

流星の周りに黒い輪郭があることにお気づきでしょうか。この輪郭は `noStroke()` 関数で消すことができます。この関数を呼び出すと、スケッチ内のどの形もアウトラインを持たなくなります。もし、アウトラインを有効にしたい場合は、そのシェイプの上で `stroke()` 関数を呼び出す必要があります。

`stroke()` や `strokeWeight()` などの他の関数を使って、図形の境界線の色や太さを指定することができます。

`noStroke()` 関数でアウトラインを削除します。

- ❑ `background()` 関数の下に `noStroke()` 関数を追加します。どの図形にもアウトラインをかけたくないので、一番上に配置します。
- ❑ プログラムを実行します。

流星に輪郭が表示されなくなります。以下は、スケッチの上部に流星がどのように表示されるかのクローズアップしたスクリーン・ショットです。



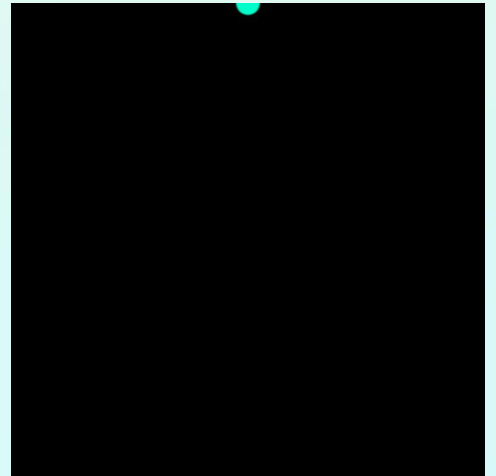
リファレンスガイドの4ページ目でコードを確認することを忘れないでください。

ステップ7：背景色の追加（3～5分）

あなたの色の知識を使って、`background()` 関数でキャンバスを塗りつぶしましょう。今は1つの値しか持っていませんが、`fill()` 関数と同じように、RGB値などを使って色を付けることができます。

ゲームの背景を変更しましょう。

- ❑ 背景色を選び、そのRGB値をメモしておく。（ここで、先ほどの `Coolors` のようなカラーピッカーやパレットツールを使用）
- ❑ `draw()` 関数内の `background()` 関数を新しい色値で更新する

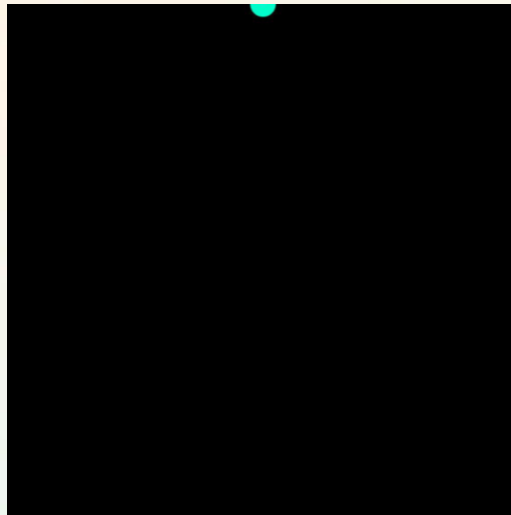


リファレンスガイド4ページ目でコードを確認することを忘れないでください。

ステップ8: コードのテスト (3-5分)

ここまでに書いたものをテストして、プログラムが思い通りに動くことを確認しましょう。再生ボタンをクリックして、スケッチを実行してください。画面上には以下のものが見えているはずです：

- 背景に色が付いた400×400のキャンバス
- キャンバスの中央上部に、新しい色で縁取りのない円形の流星



思い通りに動きませんか？もし、コードのコンパイルや実行を妨げるエラーが発生した場合、p5.jsはエラーメッセージがコンソールに表示されます。正しく動作していない場合は、それをもとに問題を解決してください。

それ以外の場合は、以下の**デバッグ方法**をお試してください。

- あなたのコードは正しい波括弧{}の中にありますか？
- コードの各行の末尾にセミコロンはついていますか？
- 変数名や関数名のスペルは正しく書けていますか？JavaScriptは大文字と小文字が区別されることを忘れないでください。
- 関数は正しい位置と順序で配置されていますか？プログラムの流れは順序が重要であることを忘れないでください！
- パラメータの値は、その関数に対して正しい範囲にありますか？例えば、キャンバスの幅が400ピクセルなのに、xの値が500になっていませんか？RGBの値は0から255の間ですか？

デバッグの最善な方法について再確認したい場合は、p5.jsコミュニティからの[素晴らしい投稿](#)をご覧ください。

ステップ9：理解度の確認

その形状の位置、大きさ、色を下のコードに記述してください。

```
function setup() { createCanvas(100, 100);  
}  
  
function draw() {  
  fill(0, 0, 255);  
  ellipse(50, 50, 5, 5);  
}
```



リファレンスガイド5ページ目で自分のアイデアを確認することを忘れないでください。

ステップ10: Girls Who Code at Home プロジェクトを共有しましょう(5分)

私たちはみなさんの作品を見ることを楽しみにしています。また、他の方も同じように思っているでしょう。

あなたの擬似コードを私たちとシェアしてください。@girlsswhocode #codefromhomeのタグをお忘れなく！
私たちのアカウントであなたを紹介するかもしれませんよ！

Girls Who Code at Homeのプロジェクトをこれからもよろしくお願いします！

