

# Lesson 3: Music Visualizer Lesson Plan

## Pre-flight instructions:

- [Click this link](#) to download GWN `Fex2016_Projects` folder, unzip and save to desktop.
- Go to the `Lesson_3` folder inside the GWN projects folder on your desktop and extract `p5-win.zip` to your desktop. This zip file contains the files for the p5.js library which we will be using to complete the lesson.
- Go to the `p5-win` folder you just extracted and double-click the `p5.exe` file. It should have a pink icon.
- When the p5.js editor launches, click on the gear icon on the top right, scroll to the 'Show sidebar' option, then click 'On'.

## Intro to p5.js

What does sound actually look like? [Watch the video from npr for a fascinating discovery.](#)

p5.js is a JavaScript library that aims to make coding accessible for artists, designers, educators, and beginners. It uses web technologies like HTML5 Canvas to make everything more interactive.

Let's run through an Introductory Tutorial to how p5.js works. Follow the instructions in the `Intro Tutorial Sketch` section below.

## Intro Tutorial Sketch Program

In this lesson we'll be using the p5.js library to draw basic shapes on the screen. [Here's a preview of what you'll be creating.](#)

Follow the numbered instructions below in order. After you have completed a numbered instruction, save your code, and switch with your partner.

In the p5 editor go to `File > Open`, then navigate to your GWN projects folder. Open `Lesson_3` and navigate to the `IntroTutorial` folder inside (`Fex2016_Projects/Lesson_3/IntroTutorial`). Open the `sketch.js` file from the `IntroTutorial` folder.

Press the play button in the p5.js editor. You should see a small window pop up with nothing in it. This is the drawing canvas where we'll be adding shapes and animations.

## #1: Canvas size

Let's make your drawing canvas bigger. By default, the drawing canvas has a size of 100 pixels by 100 pixels. Don't forget to press play after you changes to see the bigger canvas.

**Change the canvas size to 400 pixels by 400 pixels in the setup function.**

Reference: `setup()`

*The setup function runs once, and is used for setting things up. It's also used for creating a program that does not need a loop running code repeatedly. The setup function is usually created by default for every p5.js sketch.*

Reference: `createCanvas(<canvas width>, <canvas height>)`

*This function creates a canvas of the width and height you specify.*

Example: \* `createCanvas(200, 200)` -> creates a canvas 200px high and wide.

-

## #2: Background Color

Let's add a background color to the canvas.

**Add a gray background to the canvas in the setup function.**

Reference: `background(<number>)`

This function sets the color used for the background of the canvas. It allows you to specify a value between black and white. 0 is black, 255 is white.

Example: \* `background(0)` -> draws a black background on the canvas.

•

## #3: Ellipses

Let's draw an ellipse on the canvas.

**Draw an ellipse using the ellipse function in the setup function.**

Reference: `ellipse(x-coordinate, y-coordinate, width, height)`

This function draws an ellipse (oval) to the screen. The first two parameters set the location, and the last two parameters set the width and height of the ellipse.

Example: `ellipse(20, 40, 80, 50)` -> creates an oval at (20,40) that is 80px wide and 50px high.

## #4: Lines

Let's draw a line on the canvas.

**Draw a line using the line function in the setup function.**

Reference: `line(start x-coordinate, start y-coordinate, end x-coordinate, end y-coordinate)`

This function draws a line between two points on the canvas.

Example: `line(10, 10, 30, 30)` -> creates a line between (10,10) and (30, 30)

## #5: Animation

Let's animate an ellipse moving across the canvas. This instruction will be broken up into multiple sections:

a. Let's create a variable that can store numbers.

**Create a variable called "position" and give it a value of 0. Make sure to place the variable before the 'setup' function.**

Reference example: You can create variables to store numbers you want to use later.  
`var center = 10` -> Create a variable called 'center' and to store the value of 10 in it.

b. **Draw an ellipse on the canvas in the draw function. Use the 'position' variable you created above as the x-coordinate of the ellipse. The y-coordinate should be 50 pixels. The width and height should be 20 pixels.**

Reference: `draw()`

*The draw function runs repeatedly in a loop and is typically used for animation.*

---

*Reference: ellipse(x-coordinate, y-coordinate, width, height)*

*This function draws an ellipse (oval) to the screen. The first two parameters set the location, and the last two parameters set the width and height of the ellipse.*

*Example: ellipse(center, 40, 80, 50) -> creates an oval at (10,40) that is 80px wide and 50px high. The 'center' variable was created above with the value of 10*

c. Let's animate your ellipse by moving it across the canvas.

**Add the number 1 to the 'position' variable. The 'position' variable will get bigger every time the draw function is run and move the ellipse across the screen.**

*Reference example: You can add numbers to variables and store them for later use.  
center = center + 10 -> Adds ten to the 'center' variable every time it's called.*

---

## Music Visualizer Sketch Program