Introduction to Processing with p5.js

By Crystal Hess

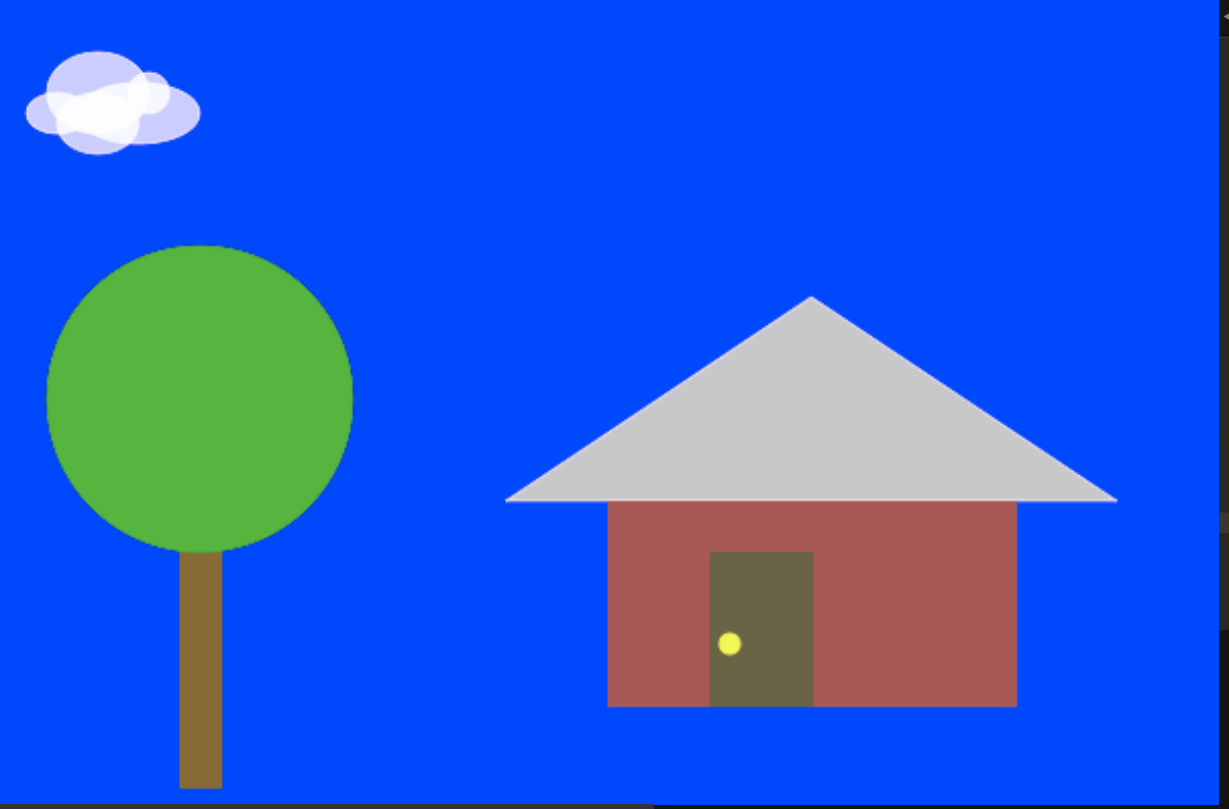
http://csed.fun

Learning Goals

- What is p5.js?
- What is “the documentation”?
- How does the coordinate system work in p5.js?
- What can a for-loop be used for?
- Note similarities between programming languages used to code in p5.js (JavaScript) and Arduino (c/c++)

What is p5.js?

- JavaScript language
- Runs in a web browser
- Spin off from processing.org
- Great for artists, designers, and beginning programmers

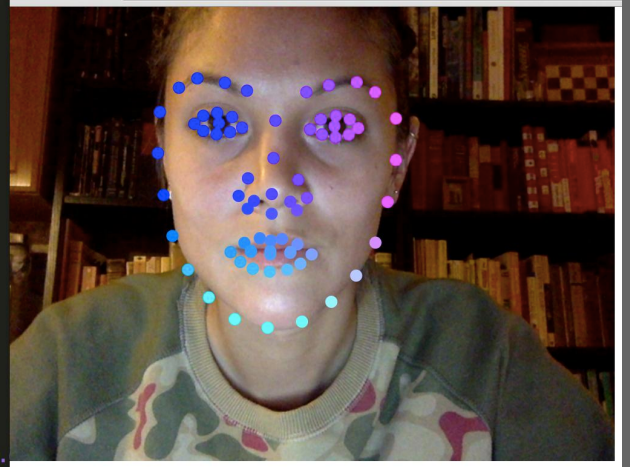


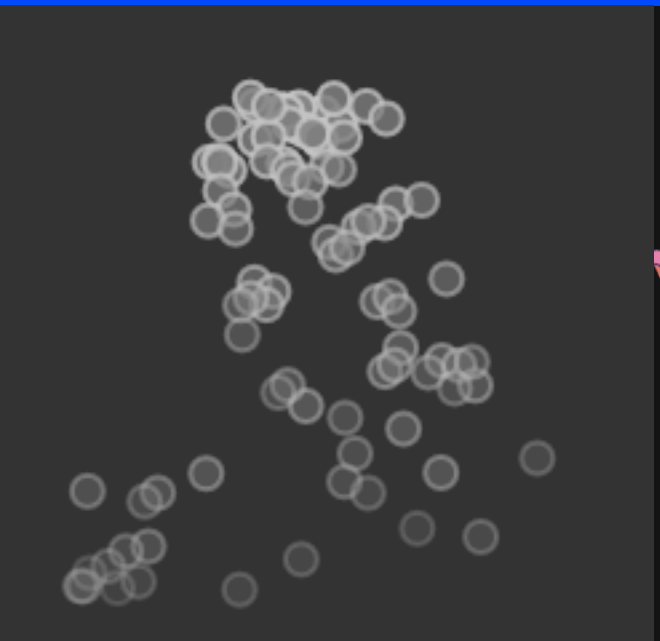
sketch.js | clmtrackr.js | p5.dom.js

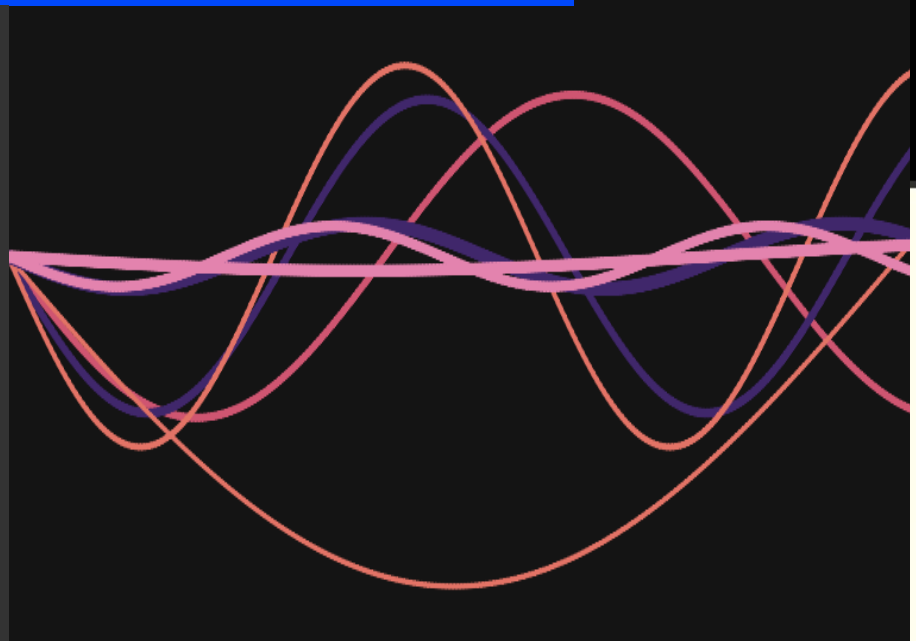
```
var ctracker;

function setup() {

  var videoInput = createCapture();
  videoInput.size(400, 300);
  videoInput.position(0, 0);

  var cnv = createCanvas(400, 300);
  cnv.position(0, 0);

  ctracker = new clm.tracker();
  ctracker.init(pModel);
  ctracker.start(videoInput.elt);

  noStroke();
}

function draw() {
  clear();

  var positions = ctracker.getCurrentPosition();
  for (var i=0; i<positions.length; i++) {
    fill(map(positions[i][0], width*0.33, width*0.
    ellipse(positions[i][0], positions[i][1], 8, 8
  }

}
```

auduno/clmtrackr | localhost:8000

localhost:8000



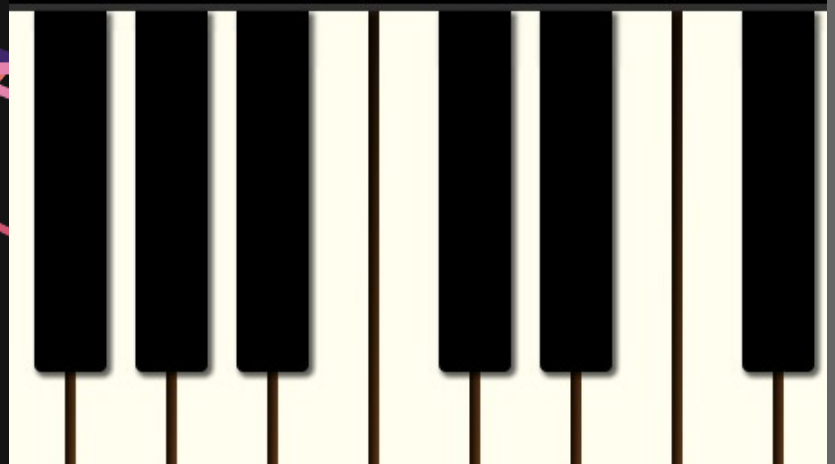




Score: 132

You got 16 out of 23 notes correct.

Note accuracy: 70%
Timing accuracy: 83%



Example: Shapes

- Open the Example: Hello → Shapes
- **Play with the code!**

- How does p5.js look similar to the Arduino language?
- How does it look different?

Reference

Search the API

Can't find what you're looking for? You may want to check out p5.dom or p5.sound.

You can download an offline version of the reference here.

Color	Data	Image	Shape
Constants	Environment	Lights, Camera	Structure
Conversion	Events	Math	Transform
DOM	IO	Rendering	Typography

Color

Creating & Reading	Setting
alpha()	background()
blue()	clear()
brightness()	colorMode()
color()	fill()
green()	noFill()
hue()	noStroke()
lerpColor()	stroke()
lightness()	
red()	
saturation()	

The p5.js API

Read the Documentation!

Example: Shapes

- Using the p5.js documentation, answer the following:
  1. What are the two values that are passed to **`createCanvas`**?
  2. What is the value that is passed to **`background`**? What is the possible range for this value?
  3. What are the values passed to **`fill`** and **`stroke`**?
  4. Figure out how **`rect`**, **`ellipse`**, and **`triangle`** work.
  5. How does the coordinate system work on the canvas?
     a. Where is the origin?
     b. With a canvas that is passed 720,400, where about is the point 700,400?
  6. What does it mean when the documentation puts an argument in brackets? e.g. **`color(v1,v2,v3,[alpha])`**

- Then,
  - Create a square white canvas.
  - Create 4 shapes that are equal in width and height, but each different in color.
  - Create a large semi-transparent "pacman" that looks like it is eating your shapes.

For Loops

```
for (var i = 0; i < 20; i++) {
  rect(50+i*20,25,5,5);
}
```

- Consider the code here
  1. Can you make the line of squares longer?
  2. Can you make the squares appear at the bottom of the canvas instead of the top?
  3. Can you space the squares farther apart?
  4. Can you make the squares larger?
  5. Can you make a border around your canvas of these small squares? (You will need to use one loop per side of the canvas.)

- **Added Fun**
  - Make each square that appears be a random color