

Creative Coding with p5.js

Jiwon Shin

jiwonshin.com

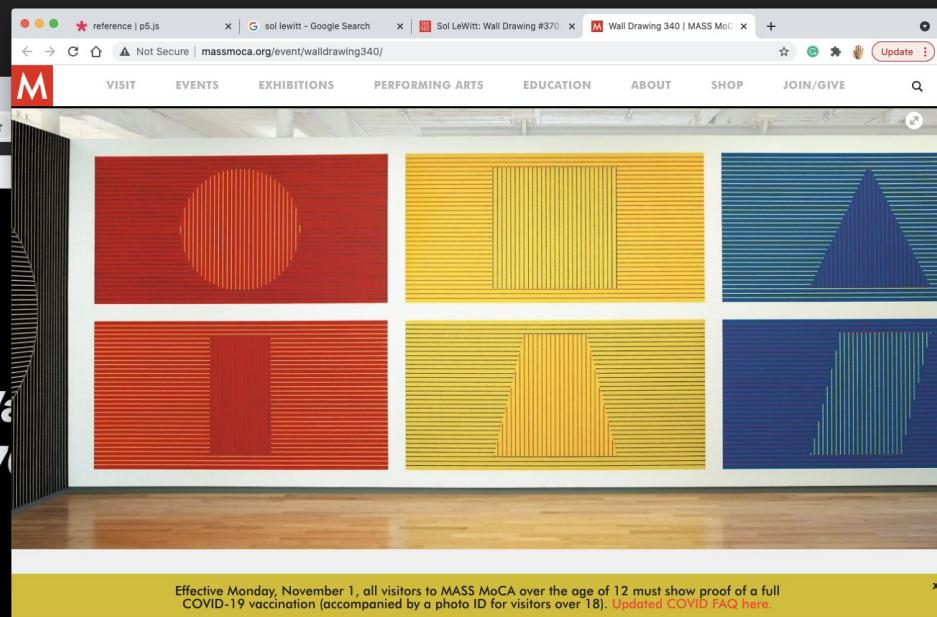
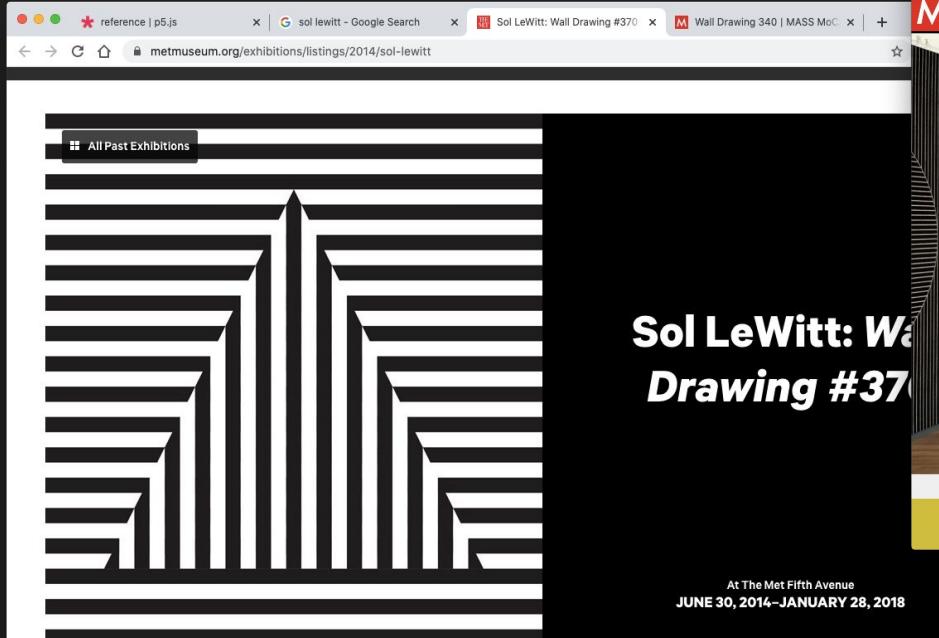
What is Creative Coding?

How can you apply creativity to programming?
What is the goal of creative coding?

Creative Coding:

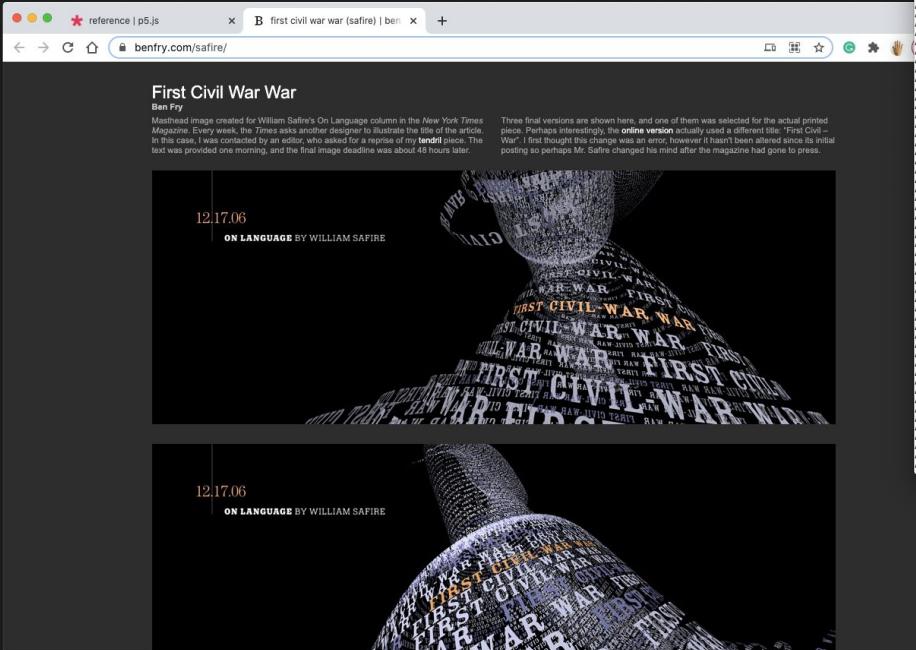
Using coding to express
an interest, a passion, or a message.

Algorithmic Art (Sol LeWitt)



Effective Monday, November 1, all visitors to MASS MoCA over the age of 12 must show proof of a full COVID-19 vaccination (accompanied by a photo ID for visitors over 18). Updated COVID FAQ [here](#).

Generative Art



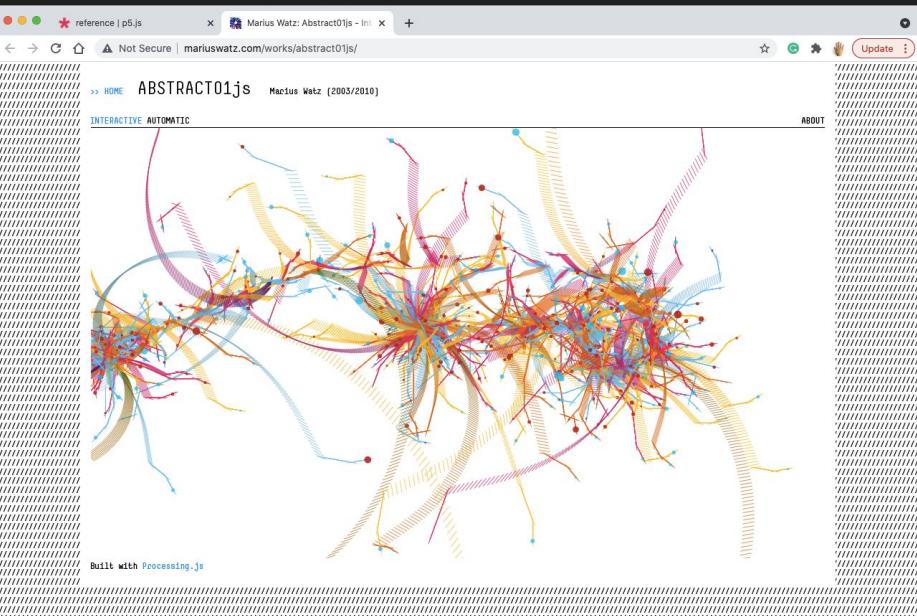
First Civil War War

Ben Fry

Masthead image created for William Safire's On Language column in the New York Times Magazine. Every week, the Times asks another designer to illustrate the theme of the article. In this case, I was contacted by an editor, who asked for a reprise of my [lizard](#) piece. The text was provided one morning, and the final image deadline was about 48 hours later.

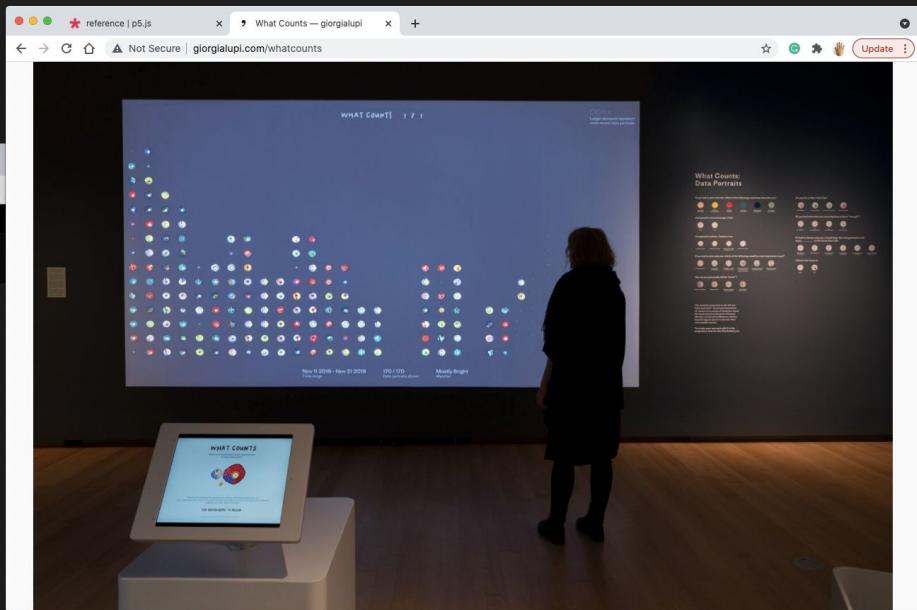
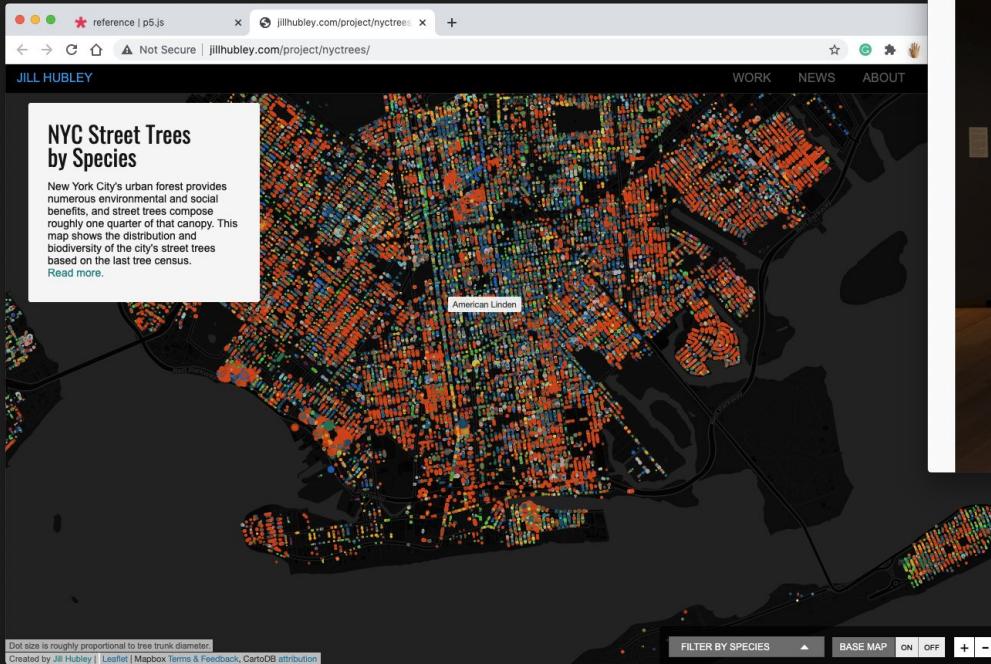
12.17.06

ON LANGUAGE BY WILLIAM SAFIRE

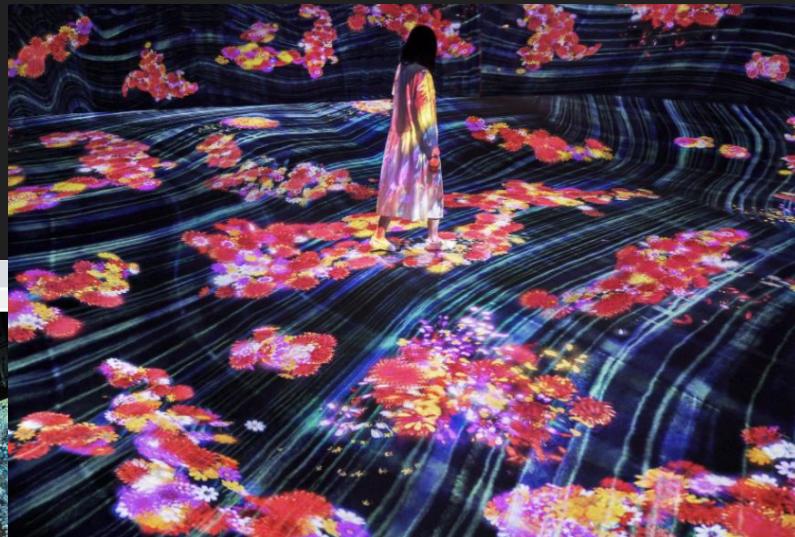
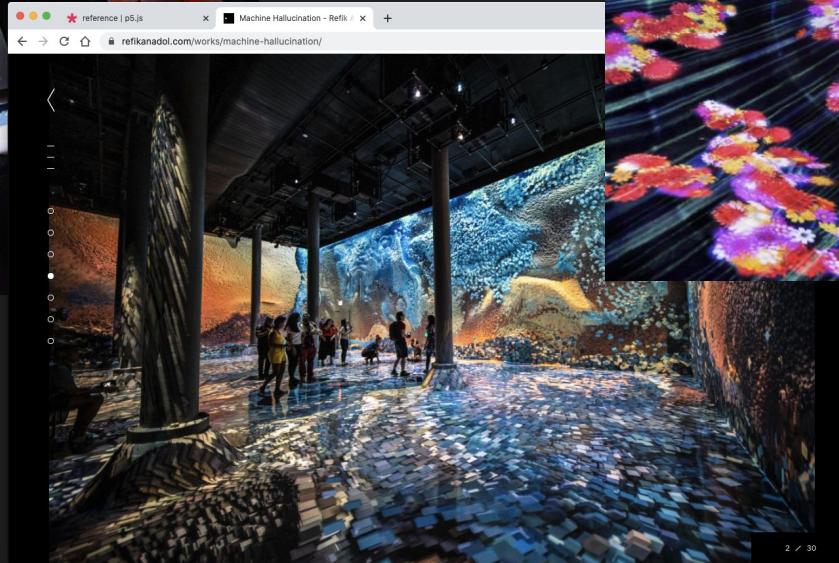
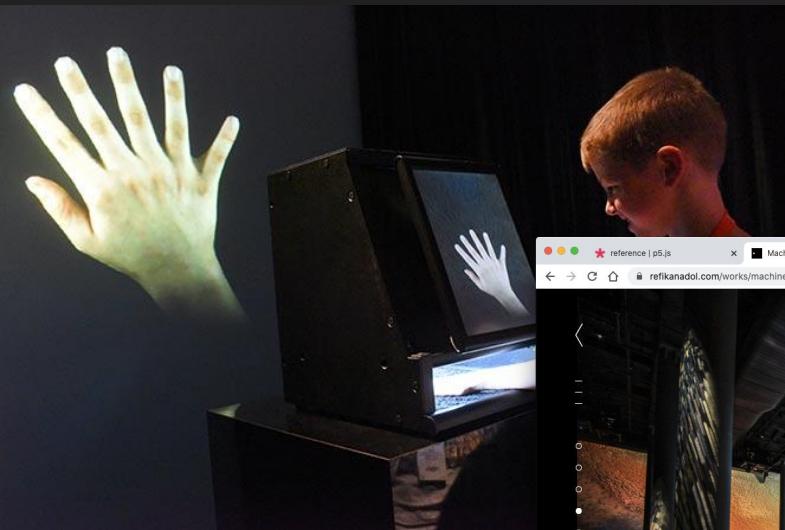


Built with Processing.js

Data Visualization



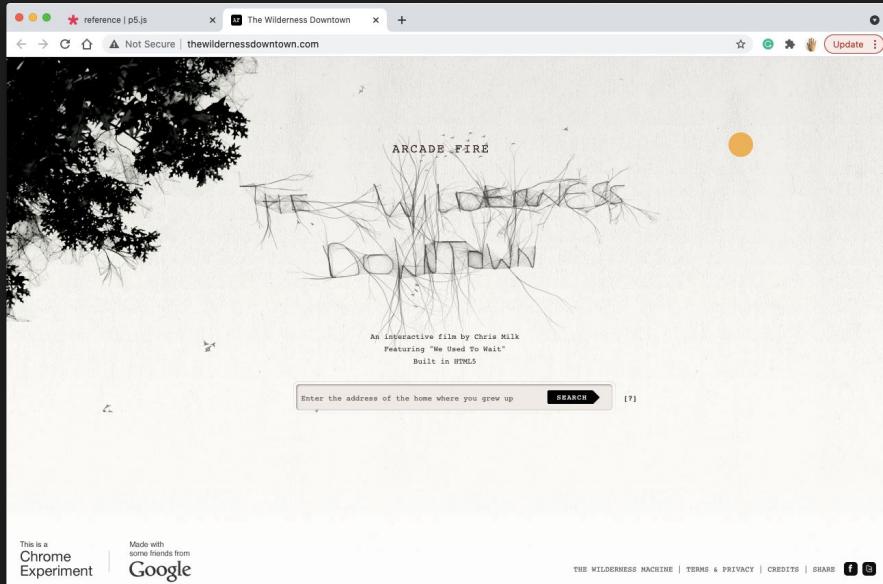
Immersive Experiences



Interactive Installations



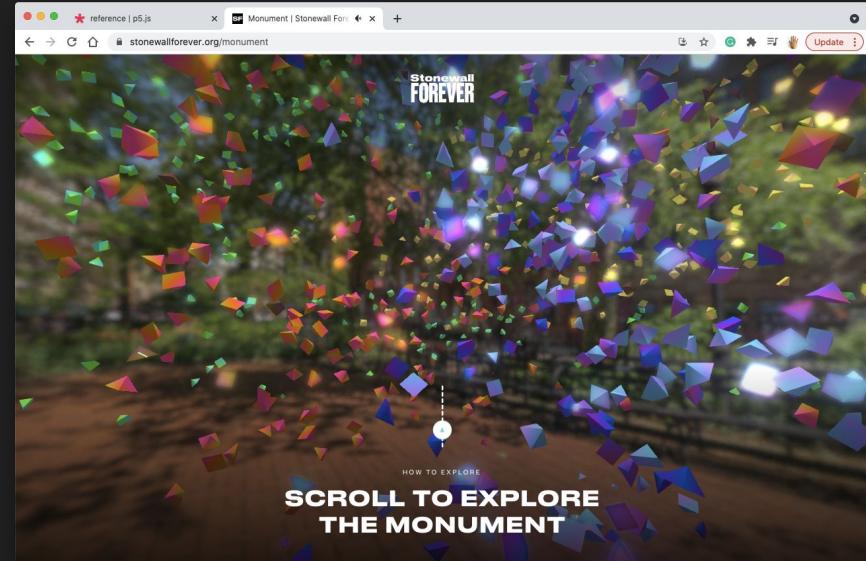
Interactive Films



This is a
Chrome
Experiment

Made with
some friends from
Google

THE WILDERNESS MACHINE | TERMS & PRIVACY | CREDITS | SHARE [f](#) [g](#)



What is p5.js?

Open-source JavaScript library for creative coding
that provides us with tools to simplify the process of creating
interactive visuals with code in the web browser.

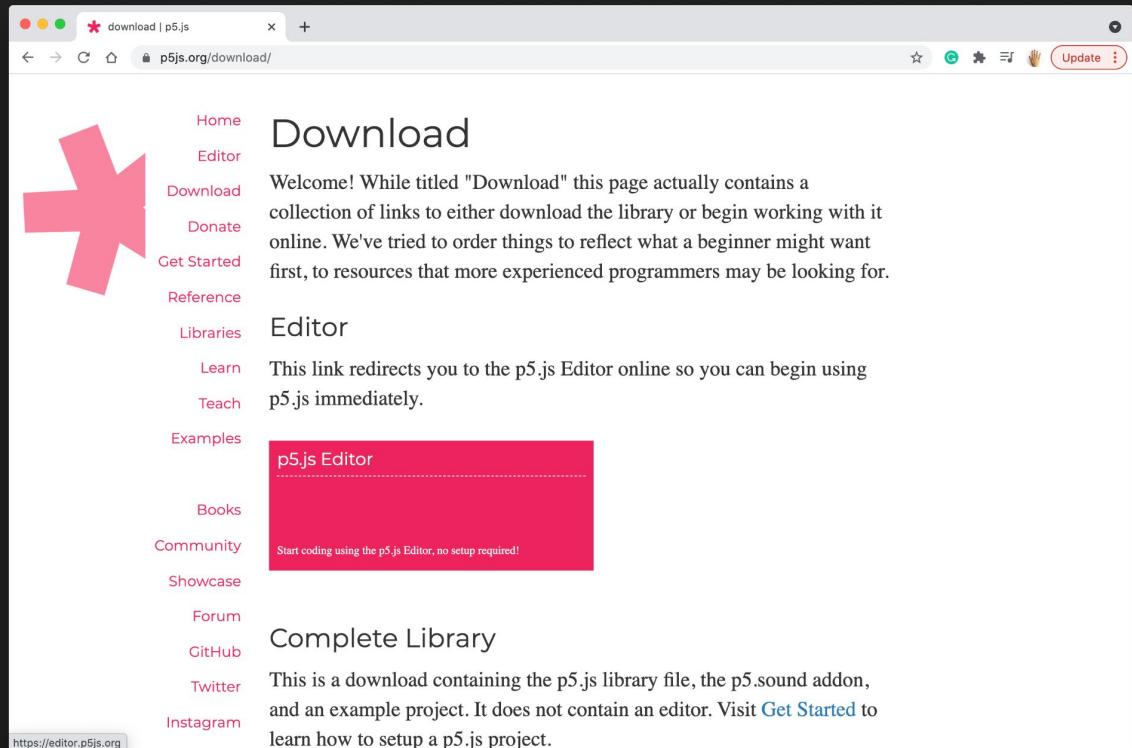
Using p5.js library: p5.js Editor

Navigate to p5js.org/download/.

Click on the p5.js Editor button.

Create an account.

Projects in p5.js Editor is already set up with the p5.js library.



The screenshot shows a web browser window with the URL p5js.org/download/. The page has a white background with a large pink p5.js logo on the left. On the right, there is a sidebar with various navigation links: Home, Editor (which is bolded), Download, Donate, Get Started, Reference, Libraries, Learn, Teach, Examples, Books, Community, Showcase, Forum, GitHub, Twitter, and Instagram. Below the sidebar, the main content area has a title "Download" and a paragraph of text. A large red rectangular button labeled "p5.js Editor" is centered in the middle of the page. At the bottom, there is a small text box with the message "Start coding using the p5.js Editor, no setup required!". At the very bottom of the page, there is a link "https://editor.p5js.org".

Download

Welcome! While titled "Download" this page actually contains a collection of links to either download the library or begin working with it online. We've tried to order things to reflect what a beginner might want first, to resources that more experienced programmers may be looking for.

Editor

This link redirects you to the p5.js Editor online so you can begin using p5.js immediately.

p5.js Editor

Start coding using the p5.js Editor, no setup required!

<https://editor.p5js.org>

Using p5.js library: <script> tag

Scroll down to the Single Files section on the download page.

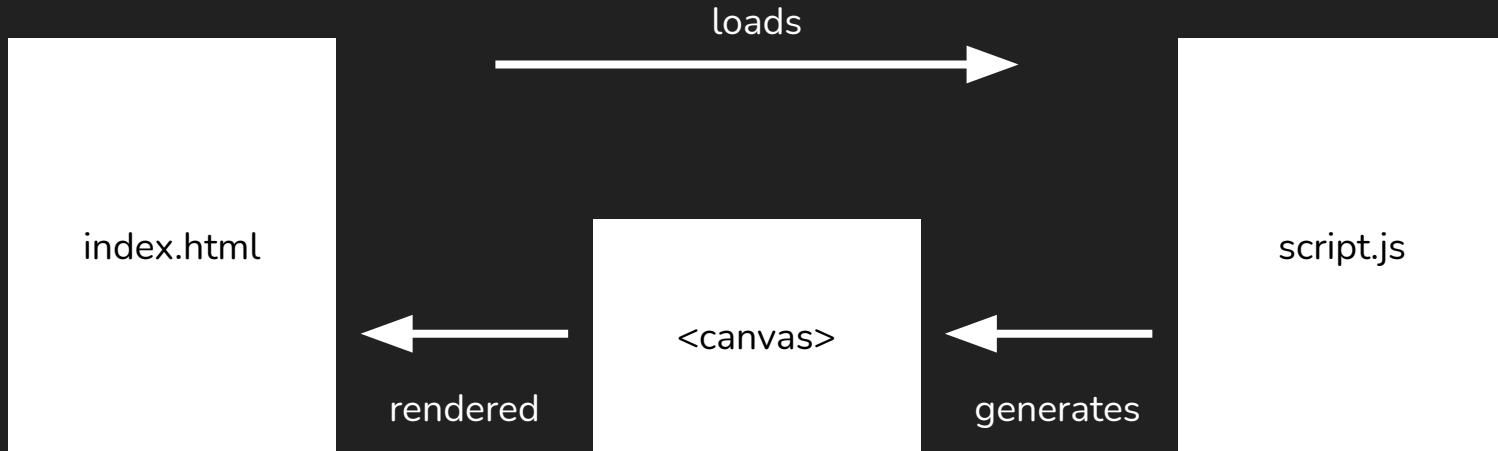
Click on the CDN button.

Copy-paste the <script> tag for the p5.min.js library.

Insert <script> tag in <head> section of your HTML document.

The screenshot shows a web browser window with the URL p5js.org/download/. At the top, there's a large pink plus sign logo. Below it, the heading "Single Files" is displayed. A descriptive text follows: "These are downloads or links to the p5.js library file. No additional contents are included." Three buttons are shown below: "p5.js" (with "Single file: Full uncompressed version"), "p5.min.js" (with "Single file: Compressed version"), and "CDN" (with "Link: Statically hosted file"). Further down, under "Github Resources", there's a bulleted list: "Previous versions (older releases and changelog)", "Code repository (GitHub)", "Report issues, bugs, and errors", and "Supported browsers". At the bottom, a footer note states: "p5.js is currently led by [Qianqian Ye](#) and [evelyn masso](#) and was created by [Lauren Lee McCarthy](#). p5.js is developed by a community of collaborators, with support from the [Processing Foundation](#) and [NYU ITP](#). Identity and graphic design by [Jerel Johnson](#). © [Info](#)". A small link at the very bottom left points to <https://cdnjs.com/libraries/p5.js>.

Anatomy of p5.js Projects



setup()

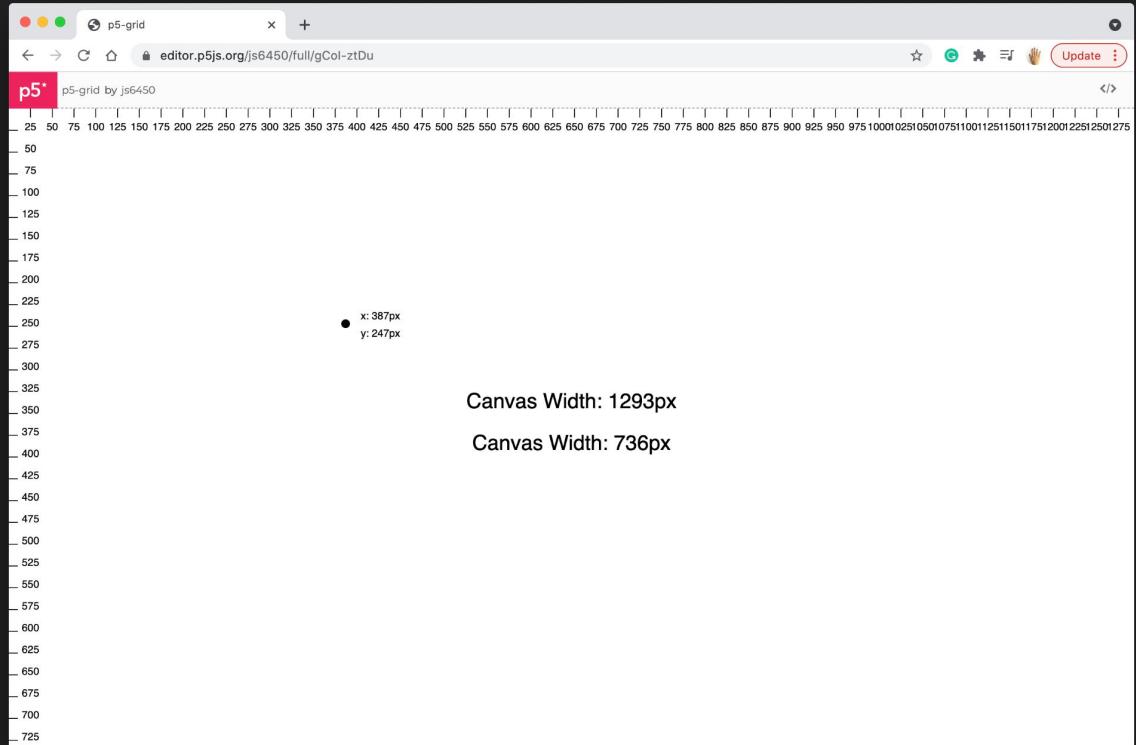
```
function setup(){  
    // a built-in p5.js function  
    // used to set the sketch's initial state when the sketch begins  
    // is not explicitly called  
    // runs only once  
    // commonly used functions: createCanvas\(\), background\(\)  
}
```

draw()

```
function draw(){  
    // a built-in p5.js function  
    // runs after setup() function  
    // is not explicitly called  
    // repeatedly executes in an infinite draw loop  
}
```

Coordinate System

The canvas uses a two-dimensional grid with ordered pairs.



Shapes

`point(x, y)`

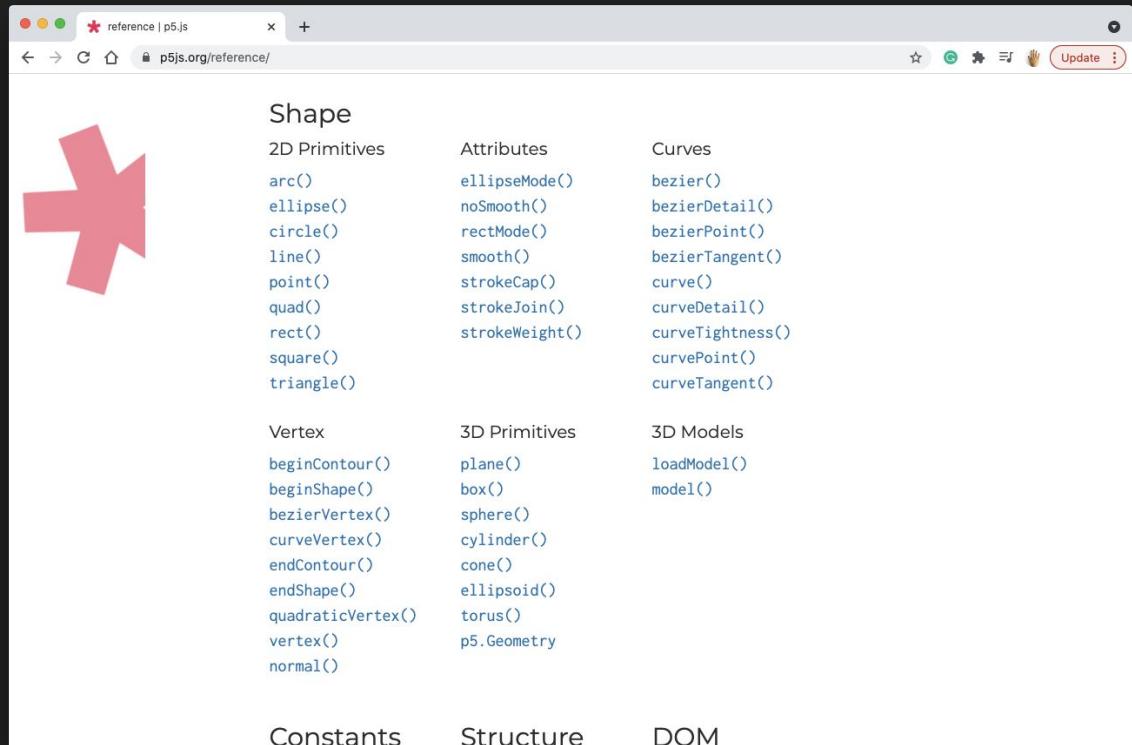
`line(x, y)`

`square(x, y, size), rect(x, y, w, h)`

`circle(x, y, size), ellipse(x, y, w, h)`

`triangle(x1, y1, x2, y2, x3, y3)`

`quad(x1, y1, x2, y2, x3, y3, x4, y4)`



The screenshot shows the p5.js reference page with the title "reference | p5.js". The main content area is titled "Shape" and contains three columns of methods:

2D Primitives	Attributes	Curves
<code>arc()</code>	<code>ellipseMode()</code>	<code>bezier()</code>
<code>ellipse()</code>	<code>noSmooth()</code>	<code>bezierDetail()</code>
<code>circle()</code>	<code>rectMode()</code>	<code>bezierPoint()</code>
<code>line()</code>	<code>smooth()</code>	<code>bezierTangent()</code>
<code>point()</code>	<code>strokeCap()</code>	<code>curve()</code>
<code>quad()</code>	<code>strokeJoin()</code>	<code>curveDetail()</code>
<code>rect()</code>	<code>strokeWeight()</code>	<code>curveTightness()</code>
<code>square()</code>		<code>curvePoint()</code>
<code>triangle()</code>		<code>curveTangent()</code>

Vertex	3D Primitives	3D Models
<code>beginContour()</code>	<code>plane()</code>	<code>loadModel()</code>
<code>beginShape()</code>	<code>box()</code>	<code>model()</code>
<code>bezierVertex()</code>	<code>sphere()</code>	
<code>curveVertex()</code>	<code>cylinder()</code>	
<code>endContour()</code>	<code>cone()</code>	
<code>endShape()</code>	<code>ellipsoid()</code>	
<code>quadraticVertex()</code>	<code>torus()</code>	
<code>vertex()</code>	<code>p5.Geometry</code>	
<code>normal()</code>		

Below the tables are navigation links: "Constants", "Structure", and "DOM".

Styling

Colors

background(color)

Grayscale:

background(51);

fill(color), noFill()

RGB:

background(255, 204, 0);

stroke(color), noStroke()

Named CSS color string:

background('red');

Hexadecimal:

background('#ffaaee');

reference | p5.js x p5.js Web Editor | shape-pattern x +

editor.p5js.org/js6450/sketches/zxtK6fOaV

p5* File ▾ Edit ▾ Sketch ▾ Help ▾

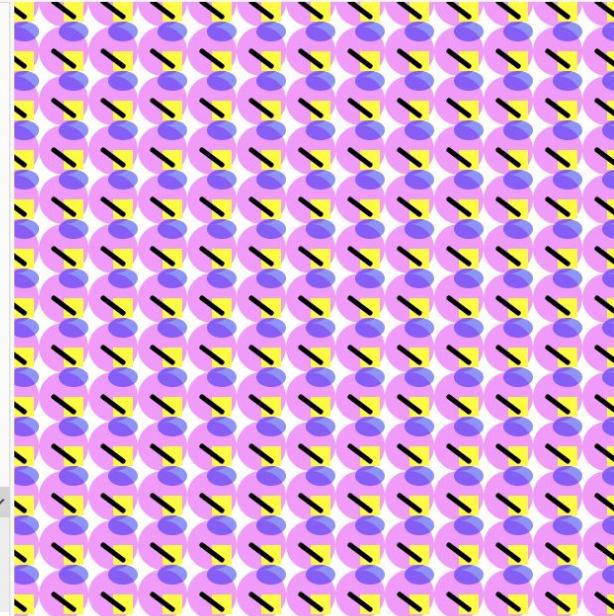
Auto-refresh shape-pattern by js6450

sketch.js

Saved: just now Preview

```
1 function setup() {
2   createCanvas(windowWidth, windowHeight);
3 }
4
5 function draw() {
6   background(255);
7
8   for(let y = 0; y < height; y += 50){
9     for(let x = 0; x < width; x += 50){
10       noStroke();
11       fill(255, 0, 255, 100);
12       circle(x, y, 50);
13
14       fill(0, 0, 255, 100);
15       ellipse(x + 10, y - 20, 30, 20);
16
17       fill(255, 255, 0)
18       square(x, y, 20);
19
20       stroke(0);
21       strokeWeight(5);
22       line(x - 10, y, x + 10, y + 15);
23     }
24   }
25 }
```

Console Clear >



Grid of Shapes

Making things move!

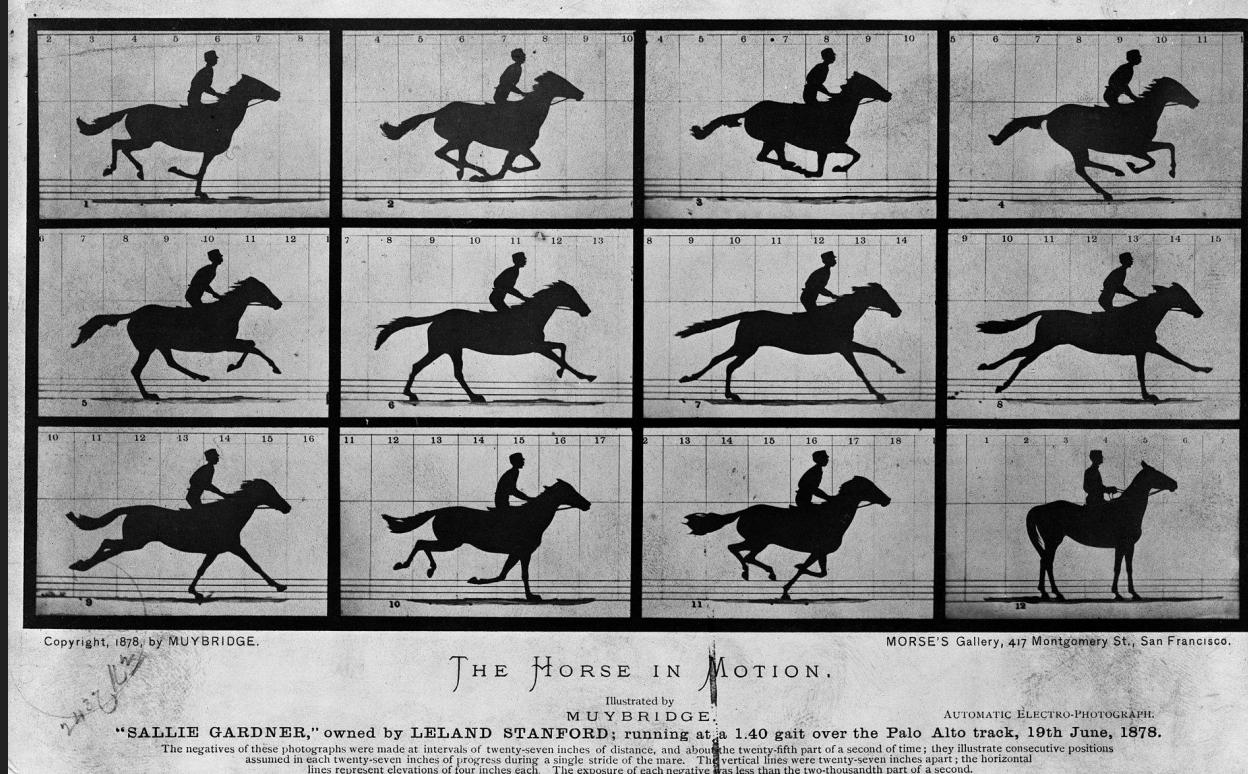
Frames

Individual pictures in a sequence of images.

When multiple frames are shown at a fast enough rate, our eyes translate and blend them into a single moving image.

In p5.js, we use the draw() loop

→ one draw() loop = 1 frame



p5.js Web Editor | bouncing-ball

File ▾ Edit ▾ Sketch ▾ Help ▾

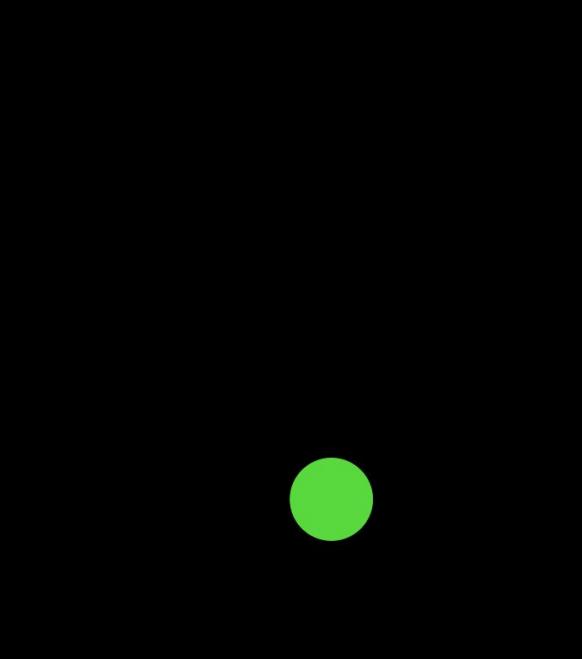
Auto-refresh bouncing-ball by js6450

English ▾ Hello,js6450! ▾

sketch.js

Saved: 9 months ago

Preview



```
/*
EXTRA CHALLENGES:
1) Can you randomize the xSpeed and ySpeed values every time the ellipse hits the canvas edges?
2) Can you tweak the if statements to factor in the radius of the ellipse to make the ellipse bounce off its edge rather than its center?
3) Can you randomize the color of the ellipse as well? Can you randomize it every time it hits the canvas edges?
*/
// Define global variables for x and y positions and x and y speeds
let xPos;
let yPos;
let xSpeed;
let ySpeed;
// Define global variables for red, green, and blue color values to be used as fill color of the bouncing ball
let redValue;
let greenValue;
let blueValue;
function setup() {
  // Create a canvas using browser (or preview window) width and browser (or preview window) height
  createCanvas(windowWidth, windowHeight);
  // Set initial value of x and y positions to be the middle of the canvas
  xPos = width / 2;
  yPos = height / 2;
  // Generate a random number between -3 and 3 for the x and y speeds
  xSpeed = random(-3, 3);
  ySpeed = random(-3, 3);
}
```

Console

Clear ▾

Bouncing Ball



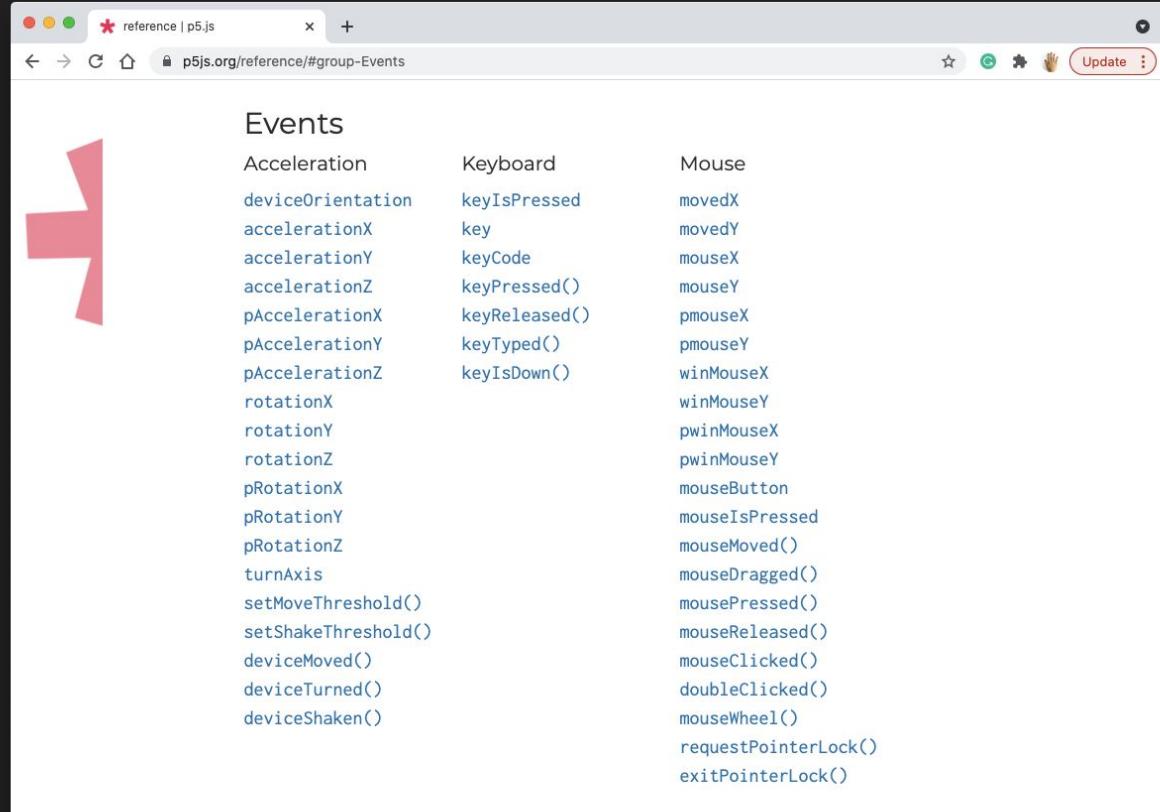
[Random](#) vs [Noise](#)

Adding interactivity!

Events

p5.js built-in variables and functions for detecting user events

- Mouse
- Keyboard
- (mobile) device orientation
- (mobile) acceleration
- (mobile) rotation



The screenshot shows a web browser window displaying the p5.js reference documentation. The title bar reads "reference | p5.js". The address bar shows the URL "p5js.org/reference/#group-Events". The main content area is titled "Events" and lists various built-in variables and functions categorized into three groups: Acceleration, Keyboard, and Mouse.

Acceleration	Keyboard	Mouse
deviceOrientation	keyIsPressed	movedX
accelerationX	key	movedY
accelerationY	keyCode	mouseX
accelerationZ	keyPressed()	mouseY
pAccelerationX	keyReleased()	pmouseX
pAccelerationY	keyTyped()	pmouseY
pAccelerationZ	keyIsDown()	winMouseX
rotationX		winMouseY
rotationY		pwinMouseX
rotationZ		pwinMouseY
pRotationX		mouseButton
pRotationY		mouseIsPressed
pRotationZ		mouseMoved()
turnAxis		mouseDragged()
setMoveThreshold()		mousePressed()
setShakeThreshold()		mouseReleased()
deviceMoved()		mouseClicked()
deviceTurned()		doubleClicked()
deviceShaken()		mouseWheel()
		requestPointerLock()
		exitPointerLock()

p5.js Web Editor

File ▾ Edit ▾ Sketch ▾ Help ▾

Auto-refresh Tricky opal

English ▾ Hello,js6450! ▾

sketchjs*

```
1 let redVal = 255;
2 let greenVal = 255;
3 let blueVal = 255;
4
5 function setup() {
6   createCanvas(windowWidth, windowHeight);
7   background(0);
8   noStroke();
9 }
10
11 function draw() {
12   // background(0, 5);
13
14   fill(redVal, greenVal, blueVal, 50);
15   ellipse(mouseX, mouseY, 100, 100);
16 }
17
18 function mousePressed(){
19   redVal = random(255);
20   greenVal = random(255);
21   blueVal = random(255);
22 }
```

Preview

Console

Clear ▾

[Simple Drawing Tool](#)

Resources

[GitHub repo](#) with code from workshop

[p5.js reference page](#)

Codecademy's [Learn p5.js](#) Course

Daniel Shiffman's [Coding Train](#)

[Make: Getting Started with p5.js](#)

[Generative Design](#)