

Generative Art

P5.js / *lab-1b*

Standup!

*Show your
concept!*

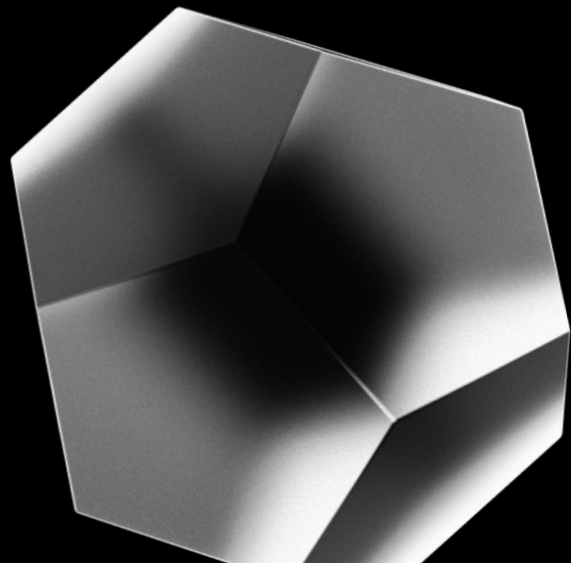
today

~~I. Standup~~

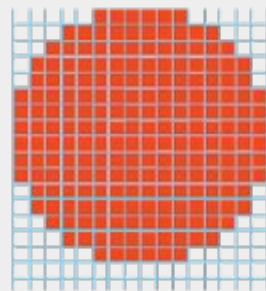
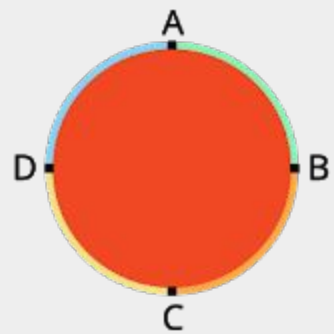
II. Vectors

III. Programming Basics

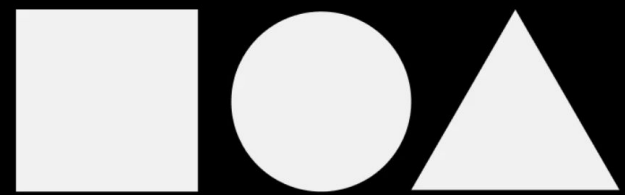
IV. P5.js Basics

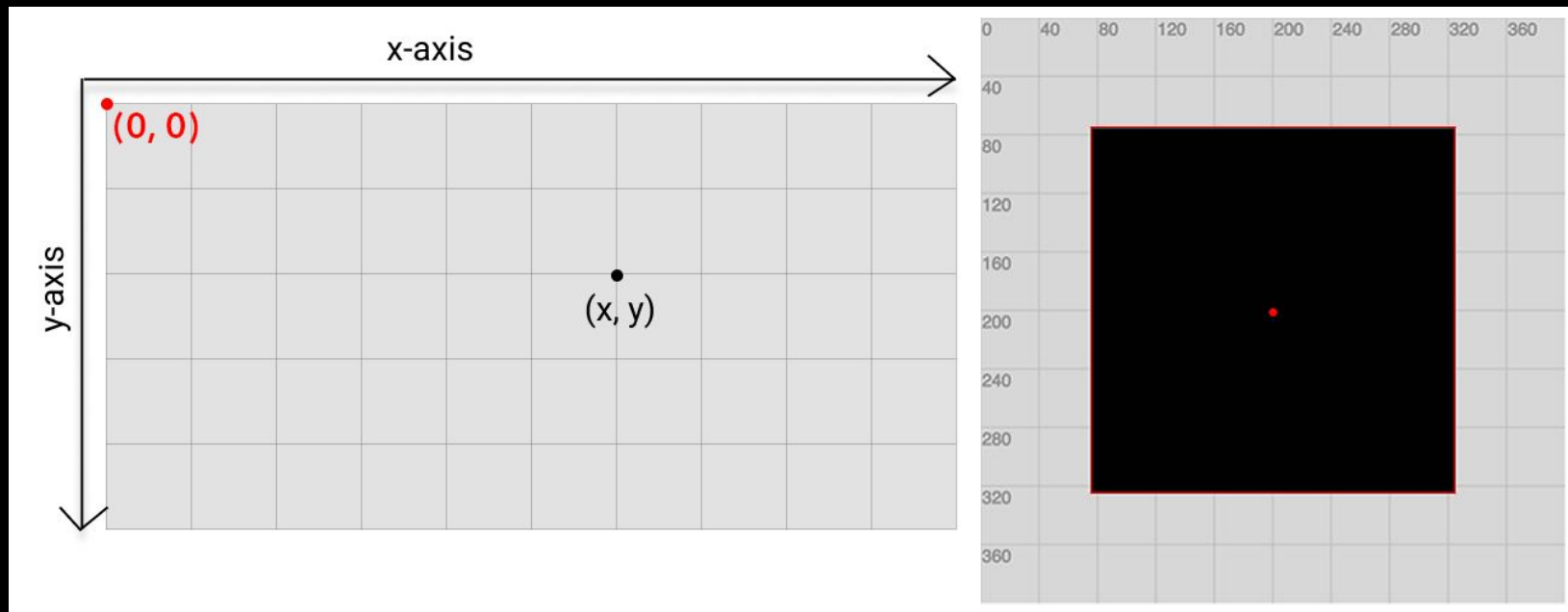


Vectors



[] () { }





Cardinal

Rectmode

Vectors

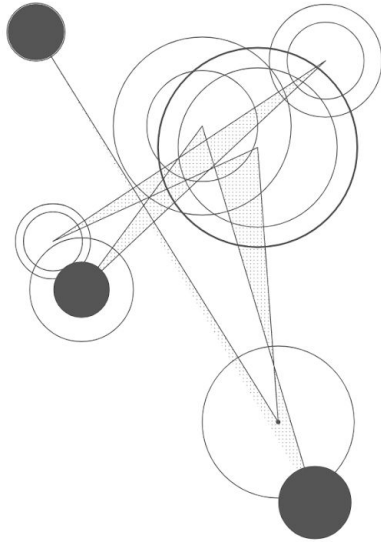
Math

The great thing about vectors (and points) is that you can do **mathematical calculations** on them.

```
speed = speed + 100;  
width = 0.3 * 10
```


Programming Basics

**Focus of
this course**



Screen (browser)

HTML

CSS

JS

~~HTML~~

~~CSS~~

JS

JavaScript

generative-art-20-21/example: X +
https://github.com/cmda-minor-vid/generative-art-20-21/tree/main/examples/basic_website 150%

cmda-minor-vid / generative-art-20-21 ✓

Watch 0 Star 0 Fork

Code Issues 2 Pull requests Actions Projects 1 Wiki Settings

main generative-art-20-21 / examples / basic_website /

dandevri Add readme to basic_website ... ✓ 27 days ago History

index.html	Add basic website .zip	2 months ago
readme.md	Add readme to basic_website	27 days ago
sketch.js	Add basic website .zip	2 months ago
style.css	Add basic website .zip	2 months ago

Procesverslag

Over

Programming

Instructions

JavaScript is the programming language for the browser.

You are providing instructions on what to appear on the screen. Instruct the computer a series of steps.

Variables

Functions

Operators

Conditionals

Loops

Data Types

**Methods
(arguments &&
Parameters)**

P5.js structure

Processing

?

Processing is a free graphical library and integrated development environment (IDE) built for the electronic arts.

In processing you create '*sketches*' by *writing functions*.

Processing

Processing

p5.js

p5.js is an open-source JavaScript library for creative coding. A **collection of pre-written code**, it provides us with tools that simplify the process of creating interactive visuals **with code in the web browser.**

Processing

Processing

p5.js

p5.js makes coding inclusive and accessible

What's cool about p5.js is if you *know very little*, you can get something on the screen pretty quickly.

Processing

Processing

p5.js

p5.js has a full set of *drawing functionality*. But not limited to drawing. You can think of your whole browser page as your sketch [...]

including HTML5 objects for text, input, video, webcam, and sound.

p5js.org

reference | p5.js

https://p5js.org/reference/

English Español 简体中文 한국어



p5.js

Home Reference

Editor

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

Donate You can also download an offline version of the reference.

Get Started

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

Learn

Examples

Books

Community

Showcase

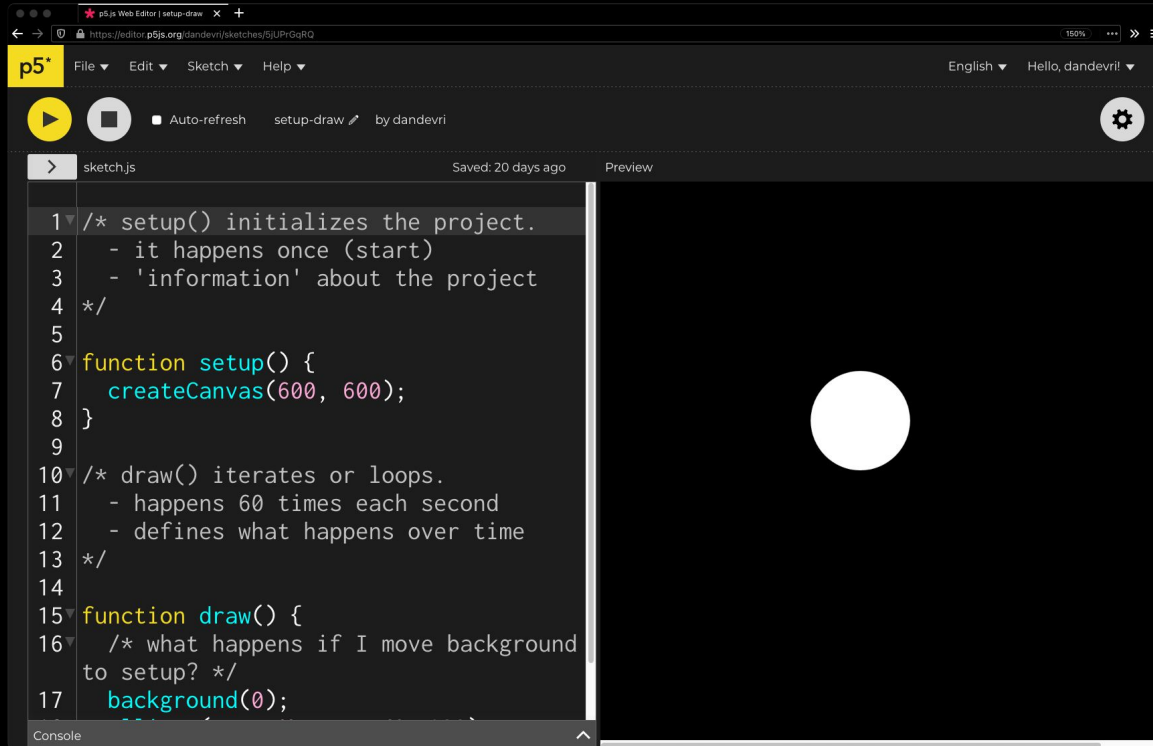
Forum

GitHub

Twitter

Environment Color

describe()	Creating & Reading	Setting
describeElement()	alpha()	background()
textOutput()	blue()	clear()
gridOutput()	brightness()	colorMode()
print()	color()	fill()
frameCount	green()	noFill()
deltaTime	hue()	noStroke()
focused	lerpColor()	stroke()
cursor()	lightness()	erase()
frameRate()	red()	noErase()
noCursor()	saturation()	
displayWidth	p5.Color	
displayHeight		
windowWidth		



Set-up / Draw

p5.js Web Editor | order-matters

https://editor.p5js.org/dandevri/sketches/DyzCekYJe

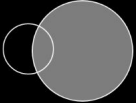
150%

File Edit Sketch Help English Hello, dandevri

Auto-refresh order-matters by dandevri

sketch.js Saved: 20 days ago Preview

```
1 /* the order in which you call
2 functions matters. each line of code
3 is read from top > bottom */
4
5 function setup() {
6   createCanvas(600, 600);
7 }
8
9 function draw() {
10  background(0)
11
12  /* what happens if I re-arrange
13 these values? */
14  fill(125)
15  ellipse(300, 300, 100)
16  noFill()
17  stroke(255)
18  ellipse(mouseX, mouseY, 50)
```



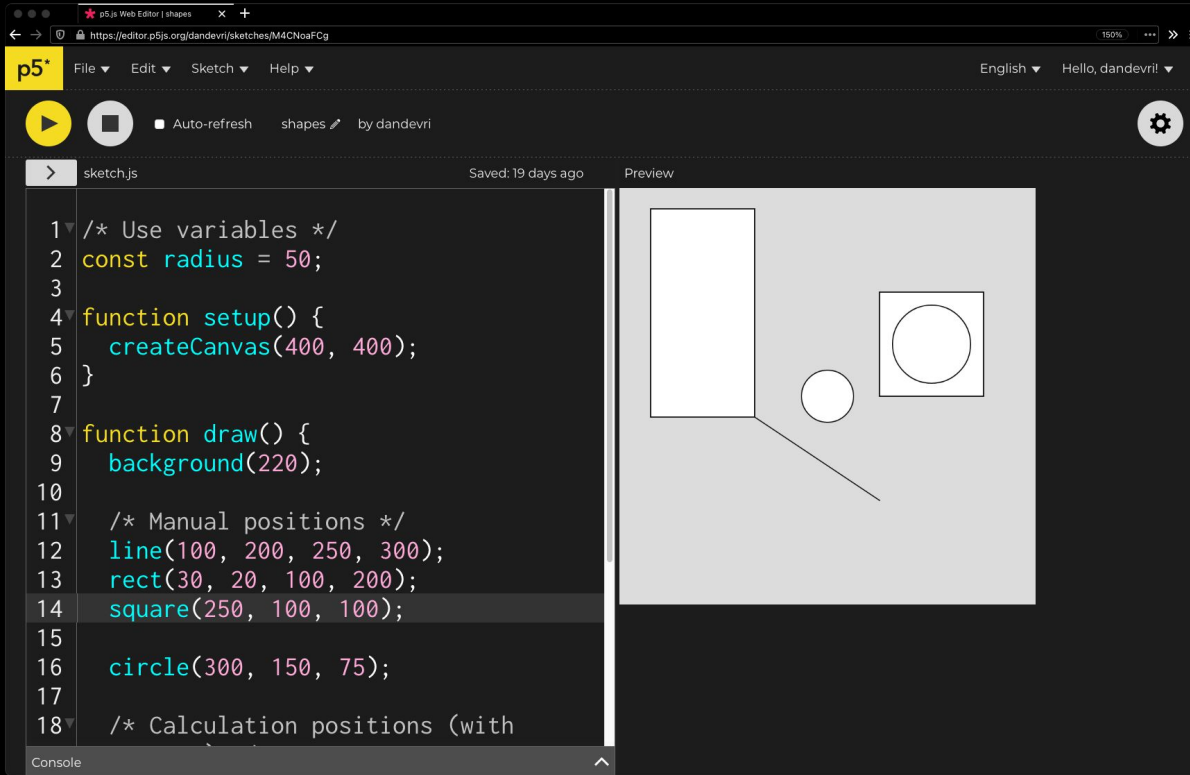
Console

Order Matters

Browser window showing a p5.js sketch editor. The URL is <https://editor.p5js.org/dandevri/sketches/M4CNaaFCg>. The sketch is named "sketch.js" and was saved 19 days ago. The code in the editor is as follows:

```
1 /* Use variables */
2 const radius = 50;
3
4 function setup() {
5   createCanvas(400, 400);
6 }
7
8 function draw() {
9   background(220);
10
11   /* Manual positions */
12   line(100, 200, 250, 300);
13   rect(30, 20, 100, 200);
14   square(250, 100, 100);
15
16   circle(300, 150, 75);
17
18   /* Calculation positions (with
```

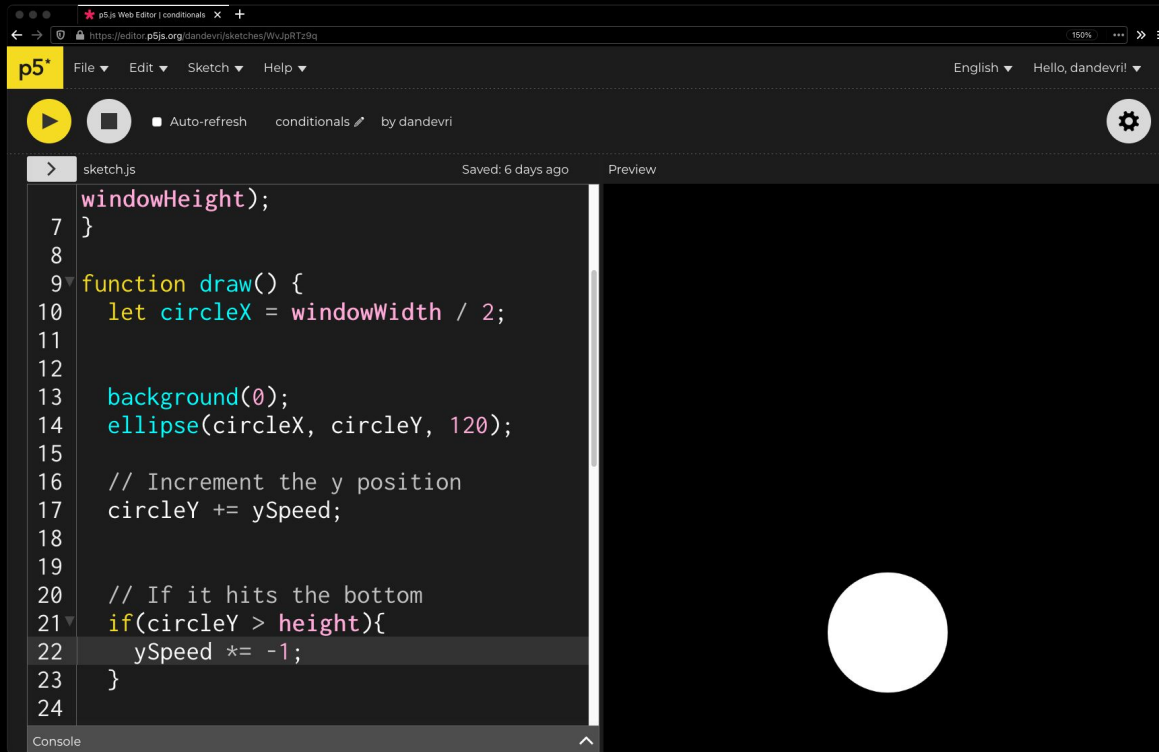
The preview window shows the rendered sketch on a light gray background. It contains a vertical rectangle on the left, a horizontal line extending from its right side to a small circle, a square on the right, and a larger circle centered within the square.



Shapes


```
1 function setup() {  
2   createCanvas(400, 400);  
3 }  
4  
5 function draw() {  
6   background(220);  
7  
8   /* Change the 4/75 values to change the  
9   pattern */  
10  for(let lineX = 0; lineX < 4; lineX++){  
11    line(75 + lineX * 75, 0, 75 + lineX *  
12    75, height);  
13  
14  /* Nesting for loops are usually used for  
   creating grids */  
   /* Change the 10/50 values to change the  
   pattern */
```

Shapes (loops)



```
7 }
8
9 function draw() {
10   let circleX = windowWidth / 2;
11
12
13   background(0);
14   ellipse(circleX, circleY, 120);
15
16   // Increment the y position
17   circleY += ySpeed;
18
19
20   // If it hits the bottom
21   if(circleY > height){
22     ySpeed *= -1;
23   }
24
```

Console

Conditionals

The screenshot shows the p5.js Web Editor interface. The top bar includes the p5.js logo, menu items (File, Edit, Sketch, Help), language settings (English), and user information (Hello, dandevri). Below the top bar, there are playback controls (play, stop) and a status bar showing 'Auto-refresh', 'animation-fps', and 'by dandevri'. The main workspace is split into two panes: a code editor on the left and a preview window on the right. The code editor shows the following JavaScript code:

```
1
2 function setup() {
3   createCanvas(windowWidth, windowHeight);
4   /* Change the framerate */
5   frameRate(10);
6 }
7
8 function draw() {
9   background(0);
10  fill(255);
11  ellipse(frameCount, height / 2,
12          300, 300);
13  fill(120);
14  textSize(72);
15  textAlign(CENTER);
```

The preview window displays a white semi-circle on a black background, with the number '26' centered to its right. The code editor has a 'Console' pane at the bottom, which is currently empty.

Animation (incrementing)

The screenshot shows a p5.js web editor interface. The top navigation bar includes the p5.js logo, menu items (File, Edit, Sketch, Help), language settings (English), and user information (Hello, dandevri). Below the navigation bar, there are control buttons for play, stop, and auto-refresh, along with the file name 'animation-fps' and the author 'by dandevri'. The main workspace is split into two panes: a code editor on the left and a preview window on the right. The code editor shows the following code:

```
1  
2 function setup() {  
3   createCanvas(windowWidth, windowHeight);  
4   /* Change the framerate */  
5   frameRate(10);  
6 }  
7  
8 function draw() {  
9   background(0);  
10  fill(255);  
11  ellipse(frameCount, height / 2,  
12         300, 300);  
13  fill(120);  
14  textSize(72);  
15  textAlign(CENTER);
```

The preview window displays a white semi-circle on a black background, with the number '26' centered to its right. The code editor has a line number column on the left and a scrollbar on the right. The preview window has a scrollbar on the right. At the bottom of the code editor, there is a 'Console' tab.

Transforms

What to do?

**Experiment with P5.js and start
creating your concept!**

exit;

see you in *lab-2a*