# Q&A

[LIVE-CODE & Q&A](https://docs.google.com/document/d/19QN-UgmMW2K5IA-WzPfqMvhYpCkSgShCsCWO244uZ0Q/edit?tab=t.n7vk1t7dkdey)

# CSS

Here is the final template (including CSS and JS): [CCLab\_F24](https://drive.google.com/drive/folders/1vl2u7cMdHkfD_ih8HqiNf0SGwY9bl1ki?usp=drive_link)

Here is what we did in the class

### HTML file:

| **<html lang="en">**  **<head>**  **<meta charset="UTF-8">**  **<meta name="viewport" content="width=device-width, initial-scale=1.0">**  **<title>Marcelaurus</title>**  **<link rel="stylesheet" href="style.css">**  **</head>**  **<body>**  **<div id="title-wrapper">**  **<h1 class="headline yellow">Marcelaurus</h1>**  **</div>**  **<div id="main-wrapper">**  **<div id="p5-canvas-container">**  **<!-- we will add an image -->**  **<img src="assets/canvas\_placeholder.png" width="800px" alt="this is a photo of my creature">**  **</div>**  **<div class="two-columns">**  **<div class="left">**  **<h3 class="red">Scientific Name:</h3>**  **<p> the name of my <br> creature is <strong><i>Marcelaurus</i> </strong></p>**  **<h3 class="green">Discovered by/at:</h3>**  **<p> marcele on Tuesday Oct 15</p>**  **<h3>Family:</h3>**  **</div>**  **<div class="right">**  **<h3 class="red">Habitat:</h3>**  **<h3 class="green">Appearance:</h3>**  **<h3>Key Characteristics:</h3>**  **<ul>**  **<li>Flies in the dark</li>**  **<li>eats metal</li>**  **<li>it is vegan</li>**  **</ul>**  **</div>**  **</div>**  **<div class="one-column">**  **<p> <span class="yellow">what we know so far about this creature:</span> blah blah blah**  **blah blah blahblah blah blahblah blah blahblah blah blahblah blah blahblah blah blahblah blah blahblah blah blahblah blah blahblah blah blahblah blah blahblah blah blahblah blah blahblah blah blah**  **</p>**  **</div>**  **</div>**  **<div id="footer">**  **<p>A CCLab project made by marcele in Fall 24</p>**  **</div>**  **<div id="home"><a href="https://cclab.work/atlas">Go Back to the Atlas</a></div>**  **</body>**  **</html>** |  |
| --- | --- |

### CSS File:

| **body {**  **background-color: black;**  **color: aliceblue;**  **font-family: 'Courier New', Courier, monospace;**  **font-size: 1em;**  **}**  **#title-wrapper {**  **text-align: center;**  **}**  **.headline {**  **font-size: 4em;**  **font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode', Geneva, Verdana, sans-serif;**  **}**  **#main-wrapper {**  **width: 800px;**  **position: relative;**  **margin: auto;**  **}**  **.two-columns {**  **display: flex;**  **}**  **.left {**  **width: 50%;**  **}**  **.right {**  **width: 50%;**  **}**  **h3{**    **}**  **.red{**  **color: coral;**  **}**  **.green{**  **color: aquamarine;**  **}**  **.yellow{**  **color: gold;**  **}**  **#home a{**  **position: fixed;**  **right: 50px;**  **top: 50px;**  **color: white;**  **text-decoration: none;**  **border: solid white 0.5px;**  **padding: 20px;**  **}**  **#home a:hover{**  **color: gold;**  **border: solid gold 0.5px;**  **}**  **#footer{**  **font-size: 0.8em;**  **text-align: center;**  **}** |  |
| --- | --- |

# HTML

| **<html lang="en">**  **<head>**  **<meta charset="UTF-8">**  **<meta name="viewport" content="width=device-width, initial-scale=1.0">**  **<title>Marcelaurus</title>**  **</head>**  **<body>**  **<h1>Marcelaurus</h1>**  **<!-- we will add an image -->**  **<img src="assets/canvas\_placeholder.png" width="800px" alt="this is a photo of my creature">**  **<h3>Scientific Name:</h3>**  **<p> the name of my <br> creature is <strong><i>Marcelaurus</i> </strong></p>**  **<h3>Discovered by/at:</h3>**  **<p> marcele on Tuesday Oct 15</p>**  **<h3>Family:</h3>**  **<h3>Habitat:</h3>**  **<h3>Appearance:</h3>**  **<h3>Key Characteristics:</h3>**  **<ul>**  **<li>Flies in the dark</li>**  **<li>eats metal</li>**  **<li>it is vegan</li>**  **</ul>**  **<p>A CCLab project made by marcele in Fall 24</p>**  **<a href="https://cclab.work/atlas">Go Back to the Atlas</a>**  **</body>**  **</html>** |  |
| --- | --- |

# Arrays

Let’s recap what we did last class. We drew a face and then we created a function to draw it, so we were able to create faces in different positions of the screen and with different sizes.

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **drawFace(width / 2, height / 2, 100);**  **}**  **function drawFace(x, y, s) {**  **push();**  **translate(x, y);**  **//rotate(frameCount \* 0.1);**  **//fill(255);**  **circle(0, 0, s);**  **fill(0);**  **circle(-s\*0.3, 0, s\*0.05);**  **circle(s\*0.3, 0, s\*0.05);**  **arc(0, 0, s\*0.3, s\*0.3, 0, PI);**  **pop();**  **}** |  |
| --- | --- |

How can we move it with linear motion?

| **let x = 0 ;**  **let y = 0 ;**  **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **drawFace(x, y, 100);**  **}**  **function drawFace(x, y, s) {**  **push();**  **translate(x, y);**  **rotate(frameCount \* 0.1);**  **fill(255);**  **circle(0, 0, s);**  **fill(0);**  **circle(-s\*0.3, 0, s\*0.05);**  **circle(s\*0.3, 0, s\*0.05);**  **arc(0, 0, s\*0.3, s\*0.3, 0, PI);**  **pop();**  **}**  **function move(){**  **x = x + 1;**  **y = y + 0.5;**  **}** |  |
| --- | --- |

## **Noise**

We can create another function to update the position

| **let x, y;**  **function setup() {**  **createCanvas(400, 400);**  **x = random(width);**  **y = random(height);**  **}**  **function draw() {**  **background(220);**  **drawFace(x, y, 100);**  **move();**  **}**  **function drawFace(x, y, s) {**  **push();**  **translate(x, y);**  **//rotate(frameCount \* 0.1);**  **//fill(255);**  **circle(0, 0, s);**  **fill(0);**  **circle(-s\*0.3, 0, s\*0.05);**  **circle(s\*0.3, 0, s\*0.05);**  **arc(0, 0, s\*0.3, s\*0.3, 0, PI);**  **pop();**  **}**  **function move(){**  **x=width\*noise(frameCount\*0.01);**  **y=height\*noise(frameCount\*0.005);**  **}** |  |
| --- | --- |

Let’s create another face

| **let x, y, x2, y2;**  **function setup() {**  **createCanvas(400, 400);**  **x = random(width);**  **y = random(height);**  **x2 = random(width);**  **y2 = random(height);**  **}**  **function draw() {**  **background(220);**  **drawFace(x, y, 100);**  **drawFace(x2, y2, 50);**  **move();**  **}**  **function drawFace(x, y, s) {**  **push();**  **translate(x, y);**  **//rotate(frameCount \* 0.1);**  **//fill(255);**  **circle(0, 0, s);**  **fill(0); -**  **circle(-s\*0.3, 0, s\*0.05);**  **circle(s\*0.3, 0, s\*0.05);**  **arc(0, 0, s\*0.3, s\*0.3, 0, PI);**  **pop();**  **}**  **function move(){**  **x=width\*noise(frameCount\*0.01);**  **y=height\*noise(frameCount\*0.005);**  **x2=width\*noise(frameCount\*0.002);**  **y2=height\*noise(frameCount\*0.01);**  **}** |  |
| --- | --- |

But what if we want 100 of them?

In this case we can use arrays

| **let x = [];**  **let y = [];**  **function setup() {**  **createCanvas(400, 400);**  **x[0] = random(width);**  **y[0] = random(height);**  **x[1] = random(width);**  **y[1] = random(height);**  **}**  **function draw() {**  **background(220);**  **drawFace(x[0], y[0], 100);**  **drawFace(x[1], y[1], 50);**  **move();**  **}**  **function drawFace(x, y, s) {**  **push();**  **translate(x, y);**  **//rotate(frameCount \* 0.1);**  **//fill(255);**  **circle(0, 0, s);**  **fill(0);**  **circle(-s\*0.3, 0, s\*0.05);**  **circle(s\*0.3, 0, s\*0.05);**  **arc(0, 0, s\*0.3, s\*0.3, 0, PI);**  **pop();**  **}**  **function move(){**  **x[0]=width\*noise(frameCount\*0.01);**  **y[0]=height\*noise(frameCount\*0.005);**  **x[1]=width\*noise(frameCount\*0.005);**  **y[1]=height\*noise(frameCount\*0.01);**  **}** |  |
| --- | --- |

Array will have elements with an index number that is growing linearly:  
X[0], x[1], x[2], x[3]

And we can add as many elements as we want. Then why don’t we use for loops?

| **let x = [];**  **let y = [];**  **let n = 2;**  **function setup() {**  **createCanvas(400, 400);**  **for (let i = 0; i < n; i++) {**  **x[i] = random(width);**  **y[i] = random(height);**  **}**  **}**  **function draw() {**  **background(220);**  **for (let i = 0; i < n; i++) {**  **drawFace(x[i], y[i], 100);**  **}**  **move();**  **}**  **function drawFace(x, y, s) {**  **push();**  **translate(x, y);**  **//rotate(frameCount \* 0.1);**  **//fill(255);**  **circle(0, 0, s);**  **fill(0);**  **circle(-s \* 0.3, 0, s \* 0.05);**  **circle(s \* 0.3, 0, s \* 0.05);**  **arc(0, 0, s \* 0.3, s \* 0.3, 0, PI);**  **pop();**  **}**  **function move() {**  **for (let i = 0; i < n; i++) {**  **x[i] = width \* noise(frameCount \* 0.01);**  **y[i] = height \* noise(frameCount \* 0.005);**  **}**  **}** |  |
| --- | --- |

Now the issue is that both are in the same position and moving with the same speeds. We can create also arrays for those variables

| **let x = [];**  **let y = [];**  **let s = [];**  **let speedX = [];**  **let speedY = [];**  **let n = 2;**  **function setup() {**  **createCanvas(400, 400);**  **for (let i = 0; i < n; i++) {**  **x[i] = random(width);**  **y[i] = random(height);**  **s[i] = random(30, 100);**  **speedX[i] = random(0.001, 0.01);**  **speedY[i] = random(0.001, 0.01);**  **}**  **}**  **function draw() {**  **background(220);**  **for (let i = 0; i < n; i++) {**  **drawFace(x[i], y[i], s[i]);**  **}**  **move();**  **}**  **function drawFace(x, y, s) {**  **push();**  **translate(x, y);**  **//rotate(frameCount \* 0.1);**  **//fill(255);**  **circle(0, 0, s);**  **fill(0);**  **circle(-s \* 0.3, 0, s \* 0.05);**  **circle(s \* 0.3, 0, s \* 0.05);**  **arc(0, 0, s \* 0.3, s \* 0.3, 0, PI);**  **pop();**  **}**  **function move() {**  **for (let i = 0; i < n; i++) {**  **x[i] = width \* noise(frameCount \* speedX[i]);**  **y[i] = height \* noise(frameCount \* speedY[i]);**  **}**  **}** |  |
| --- | --- |

Now we can easily create 100

| **let x = [];**  **let y = [];**  **let s = [];**  **let speedX = [];**  **let speedY = [];**  **let n = 100;**  **function setup() {**  **createCanvas(400, 400);**  **for (let i = 0; i < n; i++) {**  **x[i] = random(width);**  **y[i] = random(height);**  **s[i] = random(30, 100);**  **speedX[i] = random(0.001, 0.01);**  **speedY[i] = random(0.001, 0.01);**  **}**  **}**  **function draw() {**  **background(220);**  **for (let i = 0; i < n; i++) {**  **drawFace(x[i], y[i], s[i]);**  **}**  **move();**  **}**  **function drawFace(x, y, s) {**  **push();**  **translate(x, y);**  **//rotate(frameCount \* 0.1);**  **//fill(255);**  **circle(0, 0, s);**  **fill(0);**  **circle(-s \* 0.3, 0, s \* 0.05);**  **circle(s \* 0.3, 0, s \* 0.05);**  **arc(0, 0, s \* 0.3, s \* 0.3, 0, PI);**  **pop();**  **}**  **function move() {**  **for (let i = 0; i < n; i++) {**  **x[i] = width \* noise(frameCount \* speedX[i]);**  **y[i] = height \* noise(frameCount \* speedY[i]);**  **}**  **}** |  |
| --- | --- |

## **Bouncing**

Let’s try moving the face and bouncing:

| **let x;**  **let y;**  **let speedX;**  **let speedY;**  **let s;**  **function setup() {**  **createCanvas(400, 400);**  **x = width/2;**  **y = height/2;**  **speedX = 3;**  **speedY = 2;**  **s = 100;**  **}**  **function draw() {**  **background(220);**  **drawFace(x, y, s);**    **x+=speedX;**  **y+=speedY;**    **if (x < s/2 || x > width-s/2){**  **speedX = -speedX;**  **}**  **if (y < s/2 || y > height-s/2){**  **speedY = -speedY;**  **}**    **}**  **function drawFace(x, y, s) {**  **push();**  **translate(x, y);**  **//rotate(frameCount \* 0.1);**  **//fill(255);**  **circle(0, 0, s);**  **fill(0);**  **circle(-s\*0.3, 0, s\*0.05);**  **circle(s\*0.3, 0, s\*0.05);**  **arc(0, 0, s\*0.3, s\*0.3, 0, PI);**  **pop();**  **}** |  |
| --- | --- |

Can you make another face move independently with different speeds?

| **let x1;**  **let y1;**  **let speedX1;**  **let speedY1;**  **let s1;**  **let x2;**  **let y2;**  **let speedX2;**  **let speedY2;**  **let s2;**  **function setup() {**  **createCanvas(400, 400);**  **x1 = width/2;**  **y1 = height/2;**  **speedX1 = 3;**  **speedY1 = 2;**  **s1 = 100;**    **x2 = width/2;**  **y2 = height/2;**  **speedX2 = 2;**  **speedY2 = 4;**  **s2 = 150;**  **}**  **function draw() {**  **background(220);**  **drawFace(x1, y1, s1);**  **drawFace(x2, y2, s2);**    **x1+=speedX1;**  **y1+=speedY1;**    **x2+=speedX2;**  **y2+=speedY2;**    **if (x1 < s1/2 || x1 > width-s1/2){**  **speedX1 = -speedX1;**  **}**  **if (y1 < s1/2 || y1 > height-s1/2){**  **speedY1 = -speedY1;**  **}**  **if (x2 < s2/2 || x2 > width-s2/2){**  **speedX2 = -speedX2;**  **}**  **if (y2 < s2/2 || y2 > height-s2/2){**  **speedY2 = -speedY2;**  **}**    **}**  **function drawFace(x, y, s) {**  **push();**  **translate(x, y);**  **//rotate(frameCount \* 0.1);**  **//fill(255);**  **circle(0, 0, s);**  **fill(0);**  **circle(-s\*0.3, 0, s\*0.05);**  **circle(s\*0.3, 0, s\*0.05);**  **arc(0, 0, s\*0.3, s\*0.3, 0, PI);**  **pop();**  **}** |  |
| --- | --- |

But what if we want 100 of them?

In this case we can use arrays

| **let x = [];**  **let y = [];**  **let speedX = [];**  **let speedY = [];**  **let s = [];**  **function setup() {**  **createCanvas(400, 400);**  **x = [width/2, width/2];**  **y = [height/2, height/2];**  **//you can also do this:**  **// x[0] = width/2;**  **// x[1] = width/2;**  **speedX = [3, 2];**  **speedY = [2, 4];**  **s = [100, 150];**  **}**  **function draw() {**  **background(220);**  **drawFace(x[0], y[0], s[0]);**  **drawFace(x[1], y[1], s[1]);**    **x[0]+=speedX[0];**  **y[0]+=speedY[0];**    **x[1]+=speedX[1];**  **y[1]+=speedY[1];**    **if (x[0] < s[0]/2 || x[0] > width-s[0]/2){**  **speedX[0] = -speedX[0];**  **}**  **if (y[0] < s[0]/2 || y[0] > height-s[0]/2){**  **speedY[0] = -speedY[0];**  **}**    **if (x[1] < s[1]/2 || x[1] > width-s[1]/2){**  **speedX[1] = -speedX[1];**  **}**  **if (y[1] < s[1]/2 || y[1] > height-s[1]/2){**  **speedY[1] = -speedY[1];**  **}**    **}**  **function drawFace(x, y, s) {**  **push();**  **translate(x, y);**  **//rotate(frameCount \* 0.1);**  **//fill(255);**  **circle(0, 0, s);**  **fill(0);**  **circle(-s\*0.3, 0, s\*0.05);**  **circle(s\*0.3, 0, s\*0.05);**  **arc(0, 0, s\*0.3, s\*0.3, 0, PI);**  **pop();**  **}** |  |
| --- | --- |

So far it’s still a bunch of lines in our code, but what about we create 100 faces?

Do you see something familiar in the code?

| **let number = 100;**  **let x = [];**  **let y = [];**  **let speedX = [];**  **let speedY = [];**  **let s = [];**  **function setup() {**  **createCanvas(400, 400);**  **for (let i = 0; i < number; i++) {**  **x[i] = width/2;**  **y[i] = height/2;**  **speedX[i] = random(-5, 5);**  **speedY[i] = random(-5, 5);**  **s[i] = random(50, 200);**  **// x[i] = random(s[i], width);**  **// y[i] = random(s[i], height);**  **}**  **}**  **function draw() {**  **background(220);**  **for (let i = 0; i < number; i++) {**  **drawFace(x[i], y[i], s[i]);**  **x[i] += speedX[i];**  **y[i] += speedY[i];**  **if (x[i] < s[i] / 2 || x[i] > width - s[i] / 2) {**  **speedX[i] = -speedX[i];**  **}**  **if (y[i] < s[i] / 2 || y[i] > height - s[i] / 2) {**  **speedY[i] = -speedY[i];**  **}**  **}**  **}**  **function drawFace(x, y, s) {**  **push();**  **translate(x, y);**  **//rotate(frameCount \* 0.1);**  **//fill(255);**  **circle(0, 0, s);**  **fill(0);**  **circle(-s\*0.3, 0, s\*0.05);**  **circle(s\*0.3, 0, s\*0.05);**  **arc(0, 0, s\*0.3, s\*0.3, 0, PI);**  **pop();**  **}** |  |
| --- | --- |

## **Growing**

Now, play with the code and make some variations that will produce a completely different result.

Create a ball that increases and decreases the size from 0 to 100 pixels.

| **let s;**  **function setup() {**  **createCanvas(400, 400);**  **s = 100;**  **}**  **function draw() {**  **background(255);**  **fill(0);**  **circle(width/2, height/2, s);**  **s = sin(frameCount\*0.05);**  **s = map(s, -1, 1, 0, 100);**  **}** |  |
| --- | --- |

Now create 30 of them in random positions, increasing and decreasing their sizes at different speeds.

| **let number = 30;**  **let s = [];**  **let x = [];**  **let y = [];**  **let speedS = [];**  **function setup() {**  **createCanvas(400, 400);**  **for (let i = 0; i < number; i++) {**  **s[i] = 100;**  **x[i] = random(width);**  **y[i] = random(height);**  **speedS[i] = random(0.05, 0.1);**  **}**    **}**  **function draw() {**  **background(255);**  **fill(0);**  **for (let i = 0; i < number; i++) {**  **circle(x[i], y[i], s[i]);**  **s[i] = sin(frameCount\*speedS[i]);**  **s[i] = map(s[i], -1, 1, 0, 100);**  **}**  **}** |  |
| --- | --- |

You can also place them in a grid:

| **let s = [];**  **let x = [];**  **let y = [];**  **let speedS = [];**  **let counter = 0; //to count how many circles in the grid**  **let sep = 100; //grid separation also the size of the circle**  **function setup() {**  **createCanvas(400, 400);**  **for (let i = sep/2; i < width; i+=sep) {**  **for(let j=sep/2; j < height; j+=sep){**  **s[counter] = sep;**  **speedS[counter] = random(0.05, 0.1);**  **x[counter] = i;**  **y[counter] = j;**  **counter++;**  **}**  **}**  **}**  **function draw() {**  **background(255);**  **console.log(counter);**  **fill(0);**  **for (let i = 0; i < counter; i++) {**  **circle(x[i], y[i], s[i]);**  **s[i] = sin(frameCount \* speedS[i]);**  **s[i] = map(s[i], -1, 1, 0, sep);**  **}**  **}** |  |
| --- | --- |

## **Push and Splice methods**

Something interesting to use with arrays is the method Push. This method adds an item to the end of the array.

| **//let number = 16;**  **let s = [];**  **let x = [];**  **let y = [];**  **let speedS = [];**  **function setup() {**  **createCanvas(400, 400);**  **// for (let i = 0; i < number; i++) {**  **// s[i] = 100;**  **// x[i] = s[i]/2 + (i % 4) \* s[i] ;**  **// y[i] = s[i]/2 + floor(i / 4) \* s[i] ;**  **// speedS[i] = random(0.05, 0.1);**  **// console.log(x[i] + ","+ y[i]);**  **// }**  **}**  **function draw() {**  **background(255);**  **fill(0);**  **for (let i = 0; i < x.length; i++) {**  **circle(x[i], y[i], s[i]);**  **s[i] = sin(frameCount \* speedS[i]);**  **s[i] = map(s[i], -1, 1, 0, 100);**  **}**  **}**  **function mousePressed() {**  **s.push(100);**  **x.push(mouseX);**  **y.push(mouseY);**  **speedS.push(random(0.05, 0.1));**  **}**  **function keyPressed(){**  **for (let i = 0; i < x.length; i++) {**  **x.splice(i, 1);**  **y.splice(i, 1);**  **s.splice(i, 1);**  **freq.splice(i, 1);**  **}**  **}** |  |
| --- | --- |

Going back to the faces code:

| **let n = 20;**  **let x = [];**  **let y = [];**  **let s = [];**  **let speedX = [];**  **let speedY = [];**  **function setup() {**  **createCanvas(400, 400);**  **// for (let i = 0; i < n; i++) {**  **// x[i] = random(width);**  **// y[i] = random(height);**  **// speedX[i] = random(0.001, 0.01);**  **// speedY[i] = random(0.001, 0.01);**  **// s[i] = random(30, 100);**  **// }**  **}**  **function draw() {**  **background(220);**  **move();**  **for (let i = 0; i < x.length; i++) {**  **drawFace(x[i], y[i], s[i]);**  **}**  **}**  **function drawFace(x, y, s) {**  **push();**  **translate(x, y);**  **rotate(0.5 \* sin(frameCount \* 0.1));**  **fill(255);**  **circle(0, 0, s);**  **fill(0);**  **circle(-s \* 0.3, 0, s \* 0.05);**  **circle(s \* 0.3, 0, s \* 0.05);**  **arc(0, 0, s \* 0.3, s \* 0.3, 0, PI);**  **pop();**  **}**  **function move() {**  **for (let i = 0; i < x.length; i++) {**  **x[i] = width\* noise(frameCount\*speedX[i]), 0.1;**  **y[i] = height\* noise(frameCount\*speedY[i]), 0.1;**  **}**  **}**  **function mousePressed(){**  **x.push(random(width));**  **y.push(random(height));**  **s.push(random(30, 100));**  **speedX.push(random(0.001, 0.01));**  **speedY.push(random(0.001, 0.01));**  **}** |  |
| --- | --- |

Because we are using noise the face only stays one frame in the mouse position and moves quickly to the random position defined by the noise. We can fix this using LERP  
This function will make a linear interpolation between two positions. Here there is a useful video explaining this function: [Processing / p5.js Tutorial: What is lerp? (Linear Interpolation)](https://www.youtube.com/watch?v=8uLVnM36XUc&ab_channel=TheCodingTrain)  
  
Here a very simple example:

| **let x;**  **let y;**  **function setup() {**  **createCanvas(400, 400);**  **x = random(width);**  **y = random(height);**  **}**  **function draw() {**  **background(220);**  **circle(x, y, 50);**  **x = lerp(x, mouseX, 0.05);**  **y = lerp(y, mouseY, 0.05);**  **}** |  |
| --- | --- |

Now with the face:

| **let n = 20;**  **let x = [];**  **let y = [];**  **let s = [];**  **let speedX = [];**  **let speedY = [];**  **function setup() {**  **createCanvas(400, 400);**  **// for (let i = 0; i < n; i++) {**  **// x[i] = random(width);**  **// y[i] = random(height);**  **// speedX[i] = random(0.001, 0.01);**  **// speedY[i] = random(0.001, 0.01);**  **// s[i] = random(30, 100);**  **// }**  **}**  **function draw() {**  **background(220);**  **move();**  **for (let i = 0; i < x.length; i++) {**  **drawFace(x[i], y[i], s[i]);**  **}**  **}**  **function drawFace(x, y, s) {**  **push();**  **translate(x, y);**  **rotate(0.5 \* sin(frameCount \* 0.1));**  **fill(255);**  **circle(0, 0, s);**  **fill(0);**  **circle(-s \* 0.3, 0, s \* 0.05);**  **circle(s \* 0.3, 0, s \* 0.05);**  **arc(0, 0, s \* 0.3, s \* 0.3, 0, PI);**  **pop();**  **}**  **function move() {**  **for (let i = 0; i < x.length; i++) {**  **x[i] = lerp(x[i], width\* noise(frameCount\*speedX[i]), 0.1);**  **y[i] = lerp(y[i], height\* noise(frameCount\*speedY[i]), 0.1);**  **}**  **}**  **function mousePressed(){**  **x.push(random(width));**  **y.push(random(height));**  **s.push(random(30, 100));**  **speedX.push(random(0.001, 0.01));**  **speedY.push(random(0.001, 0.01));**  **}** |  |
| --- | --- |

# FUNCTIONS

Let’s draw a circle in the middle of the screen

Now let’s make a happy face 🙂

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **stroke(0);**  **fill(255);**  **circle(width/2, height/2, 100);**  **fill(0);**  **noStroke();**  **circle(width/2-30, height/2, 5);**  **circle(width/2+30, height/2, 5);**  **arc(width/2, height/2, 30, 30, 0, PI);**  **}** |  |
| --- | --- |

Can you rotate it like this?

|  |  |
| --- | --- |

| **function draw() {**  **background(220);**  **push();**  **translate(width/2, height/2);**  **rotate(frameCount\*0.1);**  **fill(255);**  **circle(width/2, height/2, 100);**  **fill(0);**  **circle(width/2-30, height/2, 5);**  **circle(width/2+30, height/2, 5);**  **arc(width/2, height/2, 30, 30, 0, PI);**  **pop();**  **}** |  |
| --- | --- |

Remember to change the position of the face!

| **function draw() {**  **background(220);**  **push();**  **translate(width/2, height/2);**  **rotate(frameCount\*0.1);**  **fill(255);**  **circle(0, 0, 100);**  **fill(0);**  **circle(-30, 0, 5);**  **circle(30, 0, 5);**  **arc(0, 0, 30, 30, 0, PI);**  **pop();**  **}** |  |
| --- | --- |

So, how can we draw a second face on the screen? In a different position and different size?

| **function draw() {**  **background(220);**  **push();**  **translate(width/2, height/2);**  **rotate(frameCount\*0.1);**  **fill(255);**  **circle(0, 0, 100);**  **fill(0);**  **circle(-30, 0, 5);**  **circle(30, 0, 5);**  **arc(0, 0, 30, 30, 0, PI);**  **pop();**    **push();**  **translate(width/4, height/4);**  **rotate(frameCount\*0.1);**  **fill(255);**  **circle(0, 0, 150);**  **fill(0);**  **circle(-30, 0, 5);**  **circle(30, 0, 5);**  **arc(0, 0, 30, 30, 0, PI);**  **pop();**  **}** |  |
| --- | --- |

We are repeating the same instructions

Let’s create a function instead. But what’s a function? ([slide 6](https://docs.google.com/presentation/d/1ZV3LzDKOZqd6o5S8bHzYu1d7mMsvOfYzSUXwMqdusoY/edit#slide=id.g9b40322adc_0_72))

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **//remember to call the function!**  **drawFace();**  **}**  **function drawFace(){**  **push();**  **translate(width/2, height/2);**  **rotate(frameCount\*0.1);**  **fill(255);**  **circle(0, 0, 100);**  **fill(0);**  **circle(-30, 0, 5);**  **circle(30, 0, 5);**  **arc(0, 0, 30, 30, 0, PI);**  **pop();**  **}** |  |
| --- | --- |

There are functions that can return true or false values, for example:

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **console.log(checkMouse());**  **if (checkMouse()) {**  **fill(255, 0, 255);**  **} else {**  **fill(255);**  **}**  **drawFace(width / 2, height / 2, 100);**  **drawFace(width / 4, height / 4, 150);**  **}**  **function drawFace(x, y, s) {**  **push();**  **translate(x, y);**  **rotate(frameCount \* 0.1);**  **//fill(255);**  **circle(0, 0, s);**  **fill(0);**  **circle(-30, 0, 5);**  **circle(30, 0, 5);**  **arc(0, 0, 30, 30, 0, PI);**  **pop();**  **}**  **function checkMouse() {**  **let d = dist(mouseX, mouseY,width/2, height/2);**  **if (d < 50) { //the radius of the face**  **return true;**  **} else {**  **return false;**  **}**  **}** |  |
| --- | --- |

They can also return a result of a calculation:

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **console.log(checkMouse());**  **if (checkMouse()) {**  **fill(255, 0, 255);**  **} else {**  **fill(255);**  **}**  **drawFace(width / 2, height / 2, 100);**  **drawFace(width / 4, height / 4, 150);**  **}**  **function drawFace(x, y, s) {**  **push();**  **translate(x, y);**  **rotate(frameCount \* 0.1);**  **//fill(255);**  **circle(0, 0, s);**  **fill(0);**  **circle(-30, 0, 5);**  **circle(30, 0, 5);**  **arc(0, 0, 30, 30, 0, PI);**  **pop();**  **}**  **function checkMouse() {**  **let value;**  **let d = dist(mouseX, mouseY,200, 200);**  **if (d < 100) {**  **value = true;**  **} else {**  **value = false;**  **}**  **return d;**  **}** |  |
| --- | --- |

If you go to the extra slides, you will find that there is a **mousePressed function** that will return a true or false value. Also the distance function returns a calculation. Remember that **mouseIsPressed was a variable**. Check the difference when you use one or the other one below:

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **//background(220);**  **if(mouseIsPressed){**  **background(random(255),random(255),random(255));**  **}**  **drawFace();**  **angle+=0.05;**  **if(checkMouse()){**  **fill(255,0,255);**  **}else{**  **fill(255);**  **}**  **}**  **function drawFace(){**  **push();**  **translate(width/2, height/2);**  **rotate(frameCount \* 0.1);**  **//fill(255);**  **circle(0, 0, 100);**  **fill(0);**  **circle(0-30, 0, 5);**  **circle(0+30, 0, 5);**  **arc(0, 0, 30, 30, 0, PI);**  **pop();**  **}**  **function checkMouse(){**  **let d = dist(mouseX, mouseY, width/2, height/2);**  **if(d < 100){**  **return true;**  **}else{**  **return false;**  **}**  **}**  **// function mousePressed(){**  **// background(random(255),random(255),random(255));**  **// }** |  |
| --- | --- |

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **//background(220);**  **// if(mouseIsPressed){**  **// background(random(255),random(255),random(255));**  **// }**  **drawFace();**  **angle+=0.05;**  **if(checkMouse()){**  **fill(255,0,255);**  **}else{**  **fill(255);**  **}**  **}**  **function drawFace(){**  **push();**  **translate(width/2, height/2);**  **rotate(frameCount \* 0.1);**  **//fill(255);**  **circle(0, 0, 100);**  **fill(0);**  **circle(0-30, 0, 5);**  **circle(0+30, 0, 5);**  **arc(0, 0, 30, 30, 0, PI);**  **pop();**  **}**  **function checkMouse(){**  **let d = dist(mouseX, mouseY, width/2, height/2);**  **if(d < 100){**  **return true;**  **}else{**  **return false;**  **}**  **}**  **function mousePressed(){**  **background(random(255),random(255),random(255));**  **}** |  |
| --- | --- |

But how do we make two faces? We need parameters. What did we change to make two before? Position and size

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **drawFace(width/2, height/2, 100);**  **drawFace(width/4, height/4, 150);**  **}**  **function drawFace(x, y, s){**  **push();**  **translate(x, y);**  **rotate(frameCount\*0.1);**  **fill(255);**  **circle(0, 0, s);**  **fill(0);**  **circle(-30, 0, 5);**  **circle(30, 0, 5);**  **arc(0, 0, 30, 30, 0, PI);**  **pop();**  **}** |  |
| --- | --- |

Can you change the speed? Or the color?

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **console.log(checkMouse());**  **if (checkMouse()) {**  **fill(255, 0, 255);**  **} else {**  **fill(255);**  **}**  **drawFace(width / 2, height / 2, 100, 0.1);**  **fill(0, 255, 200);**  **drawFace(width / 4, height / 4, 150, 0.03);**  **}**  **function drawFace(x, y, s, speed) {**  **push();**  **translate(x, y);**  **rotate(frameCount \* speed);**  **//fill(255);**  **circle(0, 0, s);**  **fill(0);**  **circle(-30, 0, 5);**  **circle(30, 0, 5);**  **arc(0, 0, 30, 30, 0, PI);**  **pop();**  **}**  **function checkMouse() {**  **let value;**  **let d = dist(mouseX, mouseY,200, 200);**  **if (d < 100) {**  **value = true;**  **} else {**  **value = false;**  **}**  **return value;**  **}** |  |
| --- | --- |

Right now, if I change the size the eyes and the mouth are not growing proportionally. How we can fix this?

| **drawFace(width / 2, height / 2, 300, 0.1);** |  |
| --- | --- |

| **function drawFace(x, y, s, speed) {**  **push();**  **translate(x, y);**  **rotate(frameCount \* speed);**  **//fill(255);**  **circle(0, 0, s);**  **fill(0);**  **circle(-s\*0.3, 0, s\*0.05);**  **circle(s\*0.3, 0, s\*0.05);**  **arc(0, 0, s\*0.3, s\*0.3, 0, PI);**  **pop();**  **}** |  |
| --- | --- |

Let’s scale the rectangles according to the mouse distance to each rectangle

| **let gridSize = 100;**  **function setup() {**  **createCanvas(400, 400);**  **colorMode(HSB,100);**  **}**  **function draw() {**  **background(220);**  **console.log(checkMouse());**  **if (checkMouse()) {**  **fill(255, 0, 255);**  **} else {**  **fill(255);**  **}**  **for (let i = gridSize/2; i < width; i += gridSize) {**  **for (let j = gridSize/2; j < height; j += 100) {**  **let h = map(i, 0, width, 30, 60);**  **fill(h, 80, 100);**  **drawFace(i, j, gridSize\*0.8, 0.1);**  **}**  **}**  **}**  **function drawFace(x, y, s, speed) {**  **push();**  **translate(x, y);**  **rotate(frameCount \* speed);**  **//fill(255);**  **circle(0, 0, s);**  **fill(0);**  **circle(-s \* 0.3, 0, s \* 0.05);**  **circle(s \* 0.3, 0, s \* 0.05);**  **arc(0, 0, s \* 0.3, s \* 0.3, 0, PI);**  **pop();**  **}**  **function checkMouse() {**  **let value;**  **let d = dist(mouseX, mouseY, 200, 200);**  **if (d < 100) {**  **value = true;**  **} else {**  **value = false;**  **}**  **return value;**  **}** |  |
| --- | --- |

Maybe placing them in a spiral?

| **function draw() {**  **background(220);**  **console.log(checkMouse());**  **for (let angle = 0; angle < 10\*2\*PI; angle += PI/16){**  **let size = map(angle, 0, 10\*2\*PI, 10, 100);**  **let r = map(angle, 0, 10\*2\*PI, 0, width);**  **let x = r\*cos(angle) + width/2;**  **let y = r\*sin(angle) + height/2;**  **let h = map(angle, 0, 10\*2\*PI, 0, 100);**  **fill(h, 50, 100);**  **drawFace(x, y, size, 0.05);**    **}**  **}** |  |
| --- | --- |

In a circle increasing the size and with a back and forth movement?

| **function draw() {**  **background(220);**  **for (let angle = 0; angle < 1.5\*PI; angle += PI/30){**  **let size = map(angle, 0, 1.5\*PI, 10, 100);**  **let x = 50\*cos(angle) + width/2;**  **let y = 50\*sin(angle) + height/2;**  **let h = map(angle, 0, 1.5\*PI, 0, 100);**  **fill(h, 50, 100);**  **drawFace(x, y, size, 0.1);**    **}**  **}**  **function drawFace(x, y, s, speed) {**  **push();**  **translate(x, y);**  **let speed = sin(frameCount\*speed);**  **rotate(speed);**  **//fill(255);**  **circle(0, 0, s);**  **fill(0);**  **circle(-s \* 0.3, 0, s \* 0.05);**  **circle(s \* 0.3, 0, s \* 0.05);**  **arc(0, 0, s \* 0.3, s \* 0.3, 0, PI);**  **pop();**  **}** |  |
| --- | --- |

Now we can move the whole thing around by applying transformations again

| **function draw() {**  **background(220);**  **for (let angle = 0; angle < 1.5\*TWO\_PI; angle += PI/30){**  **push();**  **translate(width/2,height/2);**  **rotate(frameCount\*0.1);**  **let size = map(angle, 0, 1.5\*TWO\_PI, 10, 100);**  **let R = map(angle, 0, 1.5\*TWO\_PI, 0, 100);**  **let x = R\*cos(angle); // + width/2;**  **let y = R\*sin(angle); // + height/2;**  **let h = map(angle, 0, 1.5\*TWO\_PI, 0, 100);**  **fill(h, 50, 100);**  **drawFace(x, y, size);**  **pop();**  **}**  **}**  **function drawFace(x, y, s) {**  **push();**  **translate(x, y);**  **let speed = sin(frameCount\*0.1)\*0.5;**  **rotate(speed);**  **//fill(255);**  **circle(0, 0, s);**  **fill(0);**  **circle(-s \* 0.3, 0, s \* 0.05);**  **circle(s \* 0.3, 0, s \* 0.05);**  **arc(0, 0, s \* 0.3, s \* 0.3, 0, PI);**  **pop();**  **}** |  |
| --- | --- |

Adding some noise:

| **function draw() {**  **background(220);**  **for (let angle = 0; angle < 1.5 \* PI; angle += PI / 30) {**  **push();**  **translate(width/2, height/2);**  **rotate((frameCount\*0.1));**  **let r = map(angle, 0, 10\*PI, 0, width);**  **let x = 150\*noise(frameCount\*0.01) \* cos(angle);**  **let y = 150\*noise(frameCount\*0.01) \* sin(angle);**  **let faceSize = map(angle, 0, 1.5\*PI, 10, 100);**  **let h = map(angle, 0, 1.5\*TWO\_PI, 0, 100);**  **fill(h, 50, 100);**  **drawFace(x, y, faceSize, 0.1);**  **pop();**  **}**  **}**  **function drawFace(x, y, s, speed) {**  **push();**  **noStroke();**  **translate(x, y);**  **rotate(sin(frameCount \* speed));**  **//fill(255);**  **circle(0, 0, s);**  **fill(0);**  **circle(-s \* 0.3, 0, s \* 0.05);**  **circle(s \* 0.3, 0, s \* 0.05);**  **arc(0, 0, s \* 0.3, s \* 0.3, 0, PI);**  **pop();**  **}** |  |
| --- | --- |

How can you make something like this?

|  |  |
| --- | --- |

Can you create your own function with a new shape composed of several different shapes?

| **let gridSize = 100;**  **function setup() {**  **createCanvas(400, 400);**  **colorMode(HSB, 100);**  **}**  **function draw() {**  **background(220);**  **for (let i = gridSize / 2; i < width; i += gridSize) {**  **for (let j = gridSize / 2; j < height; j += 100) {**  **let h = map(i, 0, width, 30);**  **stroke(h, 80, 100);**  **drawSquares(i, j, gridSize \* 0.85, 0.5);**  **}**  **}**  **}**  **function drawSquares(x, y, s) {**  **for (let i = 0; i < 10; i++) {**  **noFill();**  **push();**  **translate(x, y);**  **let a = map(i, 0, 5, -PI/8,PI/8);**  **let f = map(i, 0, 5, 0.3, 0.7);**  **rotate(a);**  **rectMode(CENTER);**  **strokeWeight(map(i, 0, 10, 0.5, 1));**  **rect(0, 0, s\*f, s\*f);**  **pop();**  **}**  **}** |  |
| --- | --- |

The shape is quite simple! I didn’t have this result in mind. I just played with the parameters!

| **let gridSize = 100;**  **function setup() {**  **createCanvas(400, 400);**  **colorMode(HSB, 100);**  **}**  **function draw() {**  **background(220);**  **drawSquares(width/2, height/2, gridSize \* 0.85, 0.5);**  **}**  **function drawSquares(x, y, s) {**  **for (let i = 0; i < 10; i++) {**  **noFill();**  **push();**  **translate(x, y);**  **let a = map(i, 0, 5, -PI/8,PI/8);**  **let f = map(i, 0, 5, 0.3, 0.7);**  **rotate(a);**  **rectMode(CENTER);**  **strokeWeight(map(i, 0, 10, 0.5, 1));**  **rect(0, 0, s\*f, s\*f);**  **pop();**  **}**  **}** |  |
| --- | --- |

# TRANSFORMATIONS

Our goal will be rotating the rectangle in the middle of the screen

|  |  |
| --- | --- |

Let’s start by drawing a rectangle in the middle

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(255);**  **rectMode(CENTER);**  **noFill();**  **rect(width/2, height/2, 150, 150);**  **}** |  |
| --- | --- |

Rotate PI/4

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **rotate(PI/4);**  **rectMode(CENTER);**  **noFill();**  **rect(width/2, height/2, 150, 150);**  **}** |  |
| --- | --- |

| **let angle = 0;**  **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(255);**  **rectMode(CENTER);**  **rotate(angle);**  **rect(width/2, height/2, 100,100);**  **angle = angle+PI\*0.005;**  **}** |  |
| --- | --- |

we need to move the (0,0) to the center of the rectangle

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **translate(width/2, height/2);**  **rectMode(CENTER);**  **noFill();**  **rect(width/2, height/2, 150, 150);**  **}** |  |
| --- | --- |

| **let angle = 0;**  **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(255);**  **rectMode(CENTER);**  **translate(width/2, height/2);**  **rotate(angle);**  **rect(width/2, height/2, 100,100);**  **angle = angle+PI\*0.005;**  **}** |  |
| --- | --- |

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **translate(width/2, height/2);**  **rectMode(CENTER);**  **noFill();**  **rect(0, 0, 150, 150);**  **}** |  |
| --- | --- |

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **translate(width/2, height/2);**  **rotate(PI/4);**  **rectMode(CENTER);**  **noFill();**  **rect(0, 0, 150, 150);**  **}** |  |
| --- | --- |

Let’s try to move it!

| **let angle = 0;**  **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**    **background(255);**  **translate(width/2, height/2);**  **rotate(angle);**  **rectMode(CENTER);**  **noFill();**  **rect(0, 0, 150, 150);**  **angle+=0.05;**  **}** |  |
| --- | --- |

Let’s draw another square and rotate it differently (you can’t or maybe it’s just too hard, we need to use push() and pop())

| **let angle = 0;**  **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(255);**  **push();**  **translate(width/2, height/2);**  **rotate(angle);**  **rectMode(CENTER);**  **noFill();**  **rect(0, 0, 150, 150);**  **pop();**    **push();**  **translate(width/4, height/4);**  **rotate(-angle);**  **rectMode(CENTER);**  **noFill();**  **rect(0, 0, 150, 150);**  **pop();**  **angle+=0.05;**  **}** |  |
| --- | --- |

Now let’s do it for a lot of rectangles in a grid:

|  |  |
| --- | --- |

Let’s start by drawing now a grid of rectangles

| **let s = 30;**  **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **noFill();**  **for (let i = 0; i < width; i+=s) {**  **for (let j = 0; j < width; j+=s) {**  **rect(i,j,s,s);**  **}**  **}**  **}** |  |
| --- | --- |

Let’s rotate them

| **let angle = 0;**  **let s = 30;**  **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **noFill();**  **for (let i = 0; i < width; i+=s) {**  **for (let j = 0; j < width; j+=s) {**  **translate(i,j);**  **rotate(angle);**  **rectMode(CENTER);**  **rect(0,0,s,s);**  **}**  **}**  **angle+=0.05;**  **}** |  |
| --- | --- |

| **let angle = 0;**  **let s = 30;**  **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(255);**  **noFill();**  **for (let i = 0; i < width; i+=s) {**  **for (let j = 0; j < width; j+=s) {**  **push();**  **translate(i,j);**  **rotate(angle);**  **rectMode(CENTER);**  **rect(0,0,s,s);**  **pop();**  **}**  **}**  **angle+=0.05;**  **}** |  |
| --- | --- |

Let’s scale the rectangles according to the mouse distance to each rectangle

| **let angle = 0;**  **let s = 30;**  **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(255);**  **//fill(0);**  **noFill();**  **for (let i = 0; i < width; i+=s) {**  **for (let j = 0; j < width; j+=s) {**  **let d = dist(mouseX, mouseY, i, j);**  **let f = map(d, 0, sqrt(width\*width+height\*height),0.1,3);**  **push();**  **translate(i,j);**  **//scale(f)**  **rotate(angle);**  **rectMode(CENTER);**  **rect(0,0,s\*f,s\*f);**  **pop();**  **}**  **}**  **angle+=0.05;**  **}** |  |
| --- | --- |

What’s the difference between using scale and multiplying the size?

Change colors according to distance and angle?

Change the rotation angle according to the distance?

| **let angle = 0;**  **let s = 30;**  **function setup() {**  **createCanvas(400, 400);**  **colorMode(HSB,100);**  **}**  **function draw() {**  **background(0);**  **//fill(0);**  **noFill();**  **for (let i = 0; i < width; i+=s) {**  **for (let j = 0; j < width; j+=s) {**  **let d = dist(mouseX, mouseY, i, j);**  **let f = map(d, 0, sqrt(width\*width+height\*height),0.1,3);**  **let h = map(d, 0, width,0,100);**  **let sat = map(angle%2\*PI, 0, 2\*PI,0,100);**  **angle= map(d, 0, sqrt(width\*width+height\*height),0,2\*PI);**  **stroke(h,sat,100);**  **push();**  **translate(i,j);**  **//scale(f)**  **rotate(angle);**  **rectMode(CENTER);**  **rect(0,0,s\*f,s\*f);**  **pop();**  **}**  **}**  **//angle+=0.05;**  **}** |  |
| --- | --- |

It also doesn't need to be rectangles, what if we switch to lines?

| **let angle = 0;**  **let s = 30;**  **function setup() {**  **createCanvas(400, 400);**  **colorMode(HSB,100);**  **}**  **function draw() {**  **background(0);**  **//fill(0);**  **noFill();**  **for (let i = 0; i < width; i+=s) {**  **for (let j = 0; j < width; j+=s) {**  **let d = dist(mouseX, mouseY, i, j);**  **let f = map(d, 0, sqrt(width\*width+height\*height),0.1,3);**  **let h = map(d, 0, width,0,100);**  **let sat = map(angle%2\*PI, 0, 2\*PI,0,100);**  **angle= map(d, 0, sqrt(width\*width+height\*height),0,2\*PI);**  **stroke(h,sat,100);**  **push();**  **translate(i,j);**  **//scale(f)**  **rotate(angle);**  **rectMode(CENTER);**  **line(0,0,s\*f,s\*f);**  **pop();**  **}**  **}**  **//angle+=0.05;**  **}** |  |
| --- | --- |

How would you change the shape between circles and rectangles in a grid?

| **let s = 30;**  **let angle =0;**  **function setup() {**  **createCanvas(400, 400);**  **colorMode(HSB);**  **background(0);**  **rectMode(CENTER);**    **noFill();**  **for(let x =0; x < width; x+=s){**  **for(let y =0; y < height; y+=s){**  **let d = dist(mouseX, mouseY, x, y);**  **//let s2 = map(d, 0, width\*1.5, 5, 50);**  **//angle = map(d, 0, width\*1.5, 0, PI);**  **let h = map(d, 0, width\*1.5, 0, 360);**  **push();**  **translate(x,y);**  **stroke(h, 100, 100);**  **//rotate(angle);**  **let num = random(100);**  **if(num < 50){**  **rect(0, 0, random(s), random(s));**  **}else{**  **circle(0,0,random(s));**  **}**  **pop();**  **}**  **}**  **}**  **function draw() {**  **//angle = angle+PI/100;**  **}** |  |
| --- | --- |

# 

# Loops

## **Review**

Linear growing:

**x = x + 1;**

#### **Shorthands**

**x = x + 1; ⇒ x++;**

**X = x + 5; ⇒ x+=5;**

| **let x = 0;**  **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **//x = x + 1;**  **x++;**  **console.log(x);**  **}** |
| --- |

**if(mouseIsPressed == true) ⇒ if(mouseIsPressed)**

## **Loops**

Circles with the same center, coding with what we know so far…

| **let x,y;**  **function setup() {**  **createCanvas(400, 400);**  **x = width/2;**  **y = height/2;**  **}**  **function draw() {**  **background(255);**  **noFill();**  **circle(x,y,50); //50\*1**  **circle(x,y,100); //50\*2**  **circle(x,y,150); //50\*3**  **circle(x,y,200); //50\*4**  **circle(x,y,250); //50\*5**  **circle(x,y,300); //50\*6**  **}** |  |
| --- | --- |

| **function setup() {**  **createCanvas(400, 400);**  **for (let i = 0; i < 10; i++) {**  **console.log(i);**  **}**  **}**  **function draw() {**  **background(220);**  **}** |  |
| --- | --- |

| **let x,y;**  **function setup() {**  **createCanvas(400, 400);**  **x = width/2;**  **y = height/2;**  **}**  **function draw() {**  **background(255);**  **noFill();**  **for(let i=0; i<6; i++){**  **circle(x,y,i\*50+50)**  **}**  **}** //wow! It’s the same |  |
| --- | --- |

| **let x,y;**  **let s = 50;**  **function setup() {**  **createCanvas(400, 400);**  **x = width/2;**  **y = height/2;**  **}**  **function draw() {**  **background(255);**  **noFill();**  **for(let i=0; i<6\*s; i+=s){**  **circle(x,y,i+s)**  **}**  **}** //also the same but now you can change the size easily |  |
| --- | --- |

#### **Lines:**

| **let rectSize = 10;**  **function setup() {**  **createCanvas(400, 400);**  **background(0);**  **noStroke();**  **for (let x = 0; x < width; x += rectSize) {**  **fill(random(255), random(255), random(255));**  **rect(x, 0, rectSize, height);**  **}**  **}**  **function draw() {**  **}** |  |
| --- | --- |

Can you change the color according to the position in X?

| **let rectSize = 1;**  **function setup() {**  **createCanvas(400, 400);**  **background(0);**  **noStroke();**  **for (let x = 0; x < width; x += rectSize) {**  **let r = map(x, 0, width, 0, 255);**  **fill(r, 0, 100);**  **rect(x, 0, rectSize, height);**  **}**  **}**  **function draw() {**  **}** |  |
| --- | --- |

#### 

#### **X axis repetition: Vera Molnar**

|  |  |
| --- | --- |

#### 

| **function setup() {**  **createCanvas(400, 400);**  **for (let i = 0; i < 10; i++) {**  **console.log(i);**  **}**  **}**  **function draw() {**  **background(0);**  **Circle (0,height/2, 100); // x = 0**  **circle(100,height/2, 100); // x = 1\*s**  **circle(200,height/2, 100);// x = 2\*s**  **circle(300,height/2, 100); //x = 3\*s**  **}** |  |
| --- | --- |

We can do that with a for loop

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(0);**  // circle(50,height/2, 100);  // circle(150,height/2, 100);  // circle(250,height/2, 100);  // circle(350,height/2, 100);  **for (let i = 0; i < 4; i++) {**  **circle(i\*100,height/2, 100);**  **}**  **}** |
| --- |

Changing the size for a variable

| **let s = 30;**  **function setup() {**  **createCanvas(400, 400);**  **for (let i = 0; i < 3; i++) {**  **circle(i,height/2, 100);**  **}**  **}**  **function draw() {**  **background(0);**  // circle(50,height/2, 100);  // circle(150,height/2, 100);  // circle(250,height/2, 100);  // circle(350,height/2, 100);  **for (let i = 0; i < width; i++) {**  **circle(i\*s,height/2, s);**  **}**  **}** |  |
| --- | --- |

You can also do

| **for (let i = 0; i < width; i+=s) {**  **circle(i,height/2, 100);**  **}** |
| --- |

How to do the loop in the Y axis?

| **for (let j = 0; j < height; j+=s) {**  **circle(0,j, s);**  **}** |
| --- |

Grid of circles

| **for (let i = s/2; i < width; i += s) {**  **for (let j = s/2; j < height; j += s) {**  **circle(i, j, s);**  **}**  **}** |  |
| --- | --- |

Change colors randomly in the draw lop

| **let s = 10;**  **function setup() {**  **createCanvas(400, 400);**  **background(0);**  **}**  **function draw() {**  **for (let i = s/2; i < width; i += s) {**  **fill( random(255), random(255), random(255) );**  **for (let j = s/2; j < height; j += s) {**  **circle(i, j, s);**  **}**  **}**  **}** |  |
| --- | --- |

Inside the other loop

| **let s = 10;**  **function setup() {**  **createCanvas(400, 400);**  **background(0);**  **}**  **function draw() {**  **for (let i = s/2; i < width; i += s) {**  **for (let j = s/2; j < height; j += s) {**  **fill( random(255), random(255), random(255) );**  **circle(i, j, s);**  **}**  **}**  **}** |  |
| --- | --- |

Now in the setup function

Other loops…

| **for (let i = 0; i < 40; i++) {**  **let f = map(mouseX, 0, width, 1, 30);**  **let t = sin(i\*f);**  **t = map(t, -1, 1, 1, 10);**  **stroke(255);**  **let x = i\*20;**  **rect(x,0, t, height);**  **}** |  |
| --- | --- |

Circles in circles:

| **for (let i = 0; i < 3; i++) {**  **let R = i \* 30 + 50;**  **for (let angle = 0; angle < 2 \* PI; angle+=PI/5) {**  **let x = width / 2 + R \* cos(angle);**  **let y = height / 2 + R \* sin(angle);**  **circle(x, y, 10);**  **}**  **}** |  |
| --- | --- |

Changing size with sin function:

| for (let i = 0; i < 3; i++) {  let R = i \* 50 + 50;  for (let angle = 0; angle < 2 \* PI; angle+=PI/6) {  let x = width / 2 + R \* cos(angle);  let y = height / 2 + R \* sin(angle);  **let s = map(angle, 0, 2\*PI, 5, 20);**  circle(x, y, s);  }  } |  |
| --- | --- |

Can you change color?

| for (let i = 0; i < 3; i++) {  let R = i \* 50 + 50;  for (let angle = 0; angle < 2 \* PI; angle+=PI/6) {  let x = width / 2 + R \* cos(angle);  let y = height / 2 + R \* sin(angle);  let s = map(angle, 0, 2\*PI, 5, 20);  **let h = map(angle, 0, 2\*PI, 0, 360);**  **let sat = map(i, 0, 3, 0, 100);**  **fill(h,sat,100);**  circle(x, y, s);  }  } |  |
| --- | --- |

## **Review**

Linear growing:

**x = x + 1;**

#### **Shorthands**

**x = x + 1; ⇒ x++;**

**X = x + 5; ⇒ x+=5;**

| **let x = 0;**  **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **//x = x + 1;**  **x++;**  **console.log(x);**  **}** |
| --- |

**if(mouseIsPressed == true) ⇒ if(mouseIsPressed)**

## **Loops**

Circles with the same center, coding with what we know so far…

| **let x,y;**  **function setup() {**  **createCanvas(400, 400);**  **x = width/2;**  **y = height/2;**  **}**  **function draw() {**  **background(255);**  **noFill();**  **circle(x,y,50); //50\*1**  **circle(x,y,100); //50\*2**  **circle(x,y,150); //50\*3**  **circle(x,y,200); //50\*4**  **circle(x,y,250); //50\*5**  **circle(x,y,300); //50\*6**  **}** |  |
| --- | --- |

| **function setup() {**  **createCanvas(400, 400);**  **for (let i = 0; i < 10; i++) {**  **console.log(i);**  **}**  **}**  **function draw() {**  **background(220);**  **}** |  |
| --- | --- |

| **let x,y;**  **function setup() {**  **createCanvas(400, 400);**  **x = width/2;**  **y = height/2;**  **}**  **function draw() {**  **background(255);**  **noFill();**  **for(let i=0; i<6; i++){**  **circle(x,y,i\*50+50)**  **}**  **}** //wow! It’s the same |  |
| --- | --- |

| **let x,y;**  **let s = 50;**  **function setup() {**  **createCanvas(400, 400);**  **x = width/2;**  **y = height/2;**  **}**  **function draw() {**  **background(255);**  **noFill();**  **for(let i=0; i<6\*s; i+=s){**  **circle(x,y,i+s)**  **}**  **}** //also the same but now you can change the size easily |  |
| --- | --- |

#### **Lines:**

| **let rectSize = 10;**  **function setup() {**  **createCanvas(400, 400);**  **background(0);**  **noStroke();**  **for (let x = 0; x < width; x += rectSize) {**  **fill(random(255), random(255), random(255));**  **rect(x, 0, rectSize, height);**  **}**  **}**  **function draw() {**  **}** |  |
| --- | --- |

Can you change the color according to the position in X?

| **let rectSize = 1;**  **function setup() {**  **createCanvas(400, 400);**  **background(0);**  **noStroke();**  **for (let x = 0; x < width; x += rectSize) {**  **let r = map(x, 0, width, 0, 255);**  **fill(r, 0, 100);**  **rect(x, 0, rectSize, height);**  **}**  **}**  **function draw() {**  **}** |  |
| --- | --- |

### **NESTED LOOPS**

X axis repetition: Vera Molnar

|  |  |
| --- | --- |

#### 

| **function setup() {**  **createCanvas(400, 400);**  **for (let i = 0; i < 10; i++) {**  **console.log(i);**  **}**  **}**  **function draw() {**  **background(0);**  **Circle (0,height/2, 100); // x = 0**  **circle(100,height/2, 100); // x = 1\*s**  **circle(200,height/2, 100);// x = 2\*s**  **circle(300,height/2, 100); //x = 3\*s**  **}** |  |
| --- | --- |

We can do that with a for loop

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(0);**  // circle(50,height/2, 100);  // circle(150,height/2, 100);  // circle(250,height/2, 100);  // circle(350,height/2, 100);  **for (let i = 0; i < 4; i++) {**  **circle(i\*100,height/2, 100);**  **}**  **}** |
| --- |

Changing the size for a variable

| **let s = 30;**  **function setup() {**  **createCanvas(400, 400);**  **for (let i = 0; i < 3; i++) {**  **circle(i,height/2, 100);**  **}**  **}**  **function draw() {**  **background(0);**  // circle(50,height/2, 100);  // circle(150,height/2, 100);  // circle(250,height/2, 100);  // circle(350,height/2, 100);  **for (let i = 0; i < width; i++) {**  **circle(i\*s,height/2, s);**  **}**  **}** |  |
| --- | --- |

You can also do

| **for (let i = 0; i < width; i+=s) {**  **circle(i,height/2, 100);**  **}** |
| --- |

How to do the loop in the Y axis?

| **for (let j = 0; j < height; j+=s) {**  **circle(0,j, s);**  **}** |
| --- |

### Grid of circles

| **for (let i = s/2; i < width; i += s) {**  **for (let j = s/2; j < height; j += s) {**  **circle(i, j, s);**  **}**  **}** |  |
| --- | --- |

Change colors randomly in the draw lop

| **let s = 10;**  **function setup() {**  **createCanvas(400, 400);**  **background(0);**  **}**  **function draw() {**  **for (let i = s/2; i < width; i += s) {**  **fill( random(255), random(255), random(255) );**  **for (let j = s/2; j < height; j += s) {**  **circle(i, j, s);**  **}**  **}**  **}** |  |
| --- | --- |

Inside the other loop

| **let s = 10;**  **function setup() {**  **createCanvas(400, 400);**  **background(0);**  **}**  **function draw() {**  **for (let i = s/2; i < width; i += s) {**  **for (let j = s/2; j < height; j += s) {**  **fill( random(255), random(255), random(255) );**  **circle(i, j, s);**  **}**  **}**  **}** |  |
| --- | --- |

Now in the setup function

Other loops…

| **for (let i = 0; i < 40; i++) {**  **let f = map(mouseX, 0, width, 1, 30);**  **let t = sin(i\*f);**  **t = map(t, -1, 1, 1, 10);**  **stroke(255);**  **let x = i\*20;**  **rect(x,0, t, height);**  **}** |  |
| --- | --- |

Circles in circles:

| **for (let i = 0; i < 3; i++) {**  **let R = i \* 30 + 50;**  **for (let angle = 0; angle < 2 \* PI; angle+=PI/5) {**  **let x = width / 2 + R \* cos(angle);**  **let y = height / 2 + R \* sin(angle);**  **circle(x, y, 10);**  **}**  **}** |  |
| --- | --- |

Changing size with sin function:

| for (let i = 0; i < 3; i++) {  let R = i \* 50 + 50;  for (let angle = 0; angle < 2 \* PI; angle+=PI/6) {  let x = width / 2 + R \* cos(angle);  let y = height / 2 + R \* sin(angle);  **let s = map(angle, 0, 2\*PI, 5, 20);**  circle(x, y, s);  }  } |  |
| --- | --- |

Can you change color?

| for (let i = 0; i < 3; i++) {  let R = i \* 50 + 50;  for (let angle = 0; angle < 2 \* PI; angle+=PI/6) {  let x = width / 2 + R \* cos(angle);  let y = height / 2 + R \* sin(angle);  let s = map(angle, 0, 2\*PI, 5, 20);  **let h = map(angle, 0, 2\*PI, 0, 360);**  **let sat = map(i, 0, 3, 0, 100);**  **fill(h,sat,100);**  circle(x, y, s);  }  } |  |
| --- | --- |

# Motion

## **Review**

#### **Linear motion:**

Position = position + speed // speed is a constant value

| **let x;**  **let speedX = 5;**  **function setup() {**  **createCanvas(400, 400);**  **x = 0;**  **}**  **function draw() {**  **background(255);**  **x = x + speedX;**  **circle(x, height / 2, 50);**  **}** |  |
| --- | --- |

But it doesn’t need to be!

| **let x;**  **let speedX = 5;**  **function setup() {**  **createCanvas(400, 400);**  **x = 0;**  **}**  **function draw() {**  **background(255);**  **circle(x, height / 2, 50);**  **console.log(frameCount);**  **}** |  |
| --- | --- |

We can use this variable to move things:

| **let x;**  **let speedX = 5;**  **function setup() {**  **createCanvas(400, 400);**  **x = 0;**  **}**  **function draw() {**  **background(255);**  **x = frameCount;**  **circle(x, height / 2, 50);**  **console.log(frameCount);**  **}** |  |
| --- | --- |

But we can also use this variable to accelerate things (you can create your own acceleration variable too)

| **let x;**  **let speedX = 5;**  **function setup() {**  **createCanvas(400, 400);**  **x = 0;**  **}**  **function draw() {**  **background(255);**  **x = x + frameCount;**  **circle(x, height / 2, 50);**  **console.log(frameCount);**  **}** |  |
| --- | --- |

It’s super fast, so let’s reduce it:

| **let x;**  **let speedX = 5;**  **function setup() {**  **createCanvas(400, 400);**  **x = 0;**  **}**  **function draw() {**  **background(255);**  **x = x + frameCount\*0.1;**  **circle(x, height / 2, 50);**  **console.log(frameCount);**  **}** |  |
| --- | --- |

## Oscillation and Angular Motion

Inspiration

Memo Akten

Alexander Chen

## **Angles:**

Radians and Degrees

| **function setup() {**  **createCanvas(500,500);**  **}**  **function draw() {**  **//convert degrees to radians**  **console.log(sin(PI/2));**  **//console.log(sin(radians(90)));**    **// angleMode(DEGREES);**  **// console.log(sin(90));**  **}** |  |
| --- | --- |

## **Oscillation:**

**Example1:** Position = position + speed //speed is not constant, it is a sin value

Let’s start by converting frameCount to radians

| **let x;**  **let speedX = 5;**  **function setup() {**  **createCanvas(400, 400);**  **x = 0;**  **}**  **function draw() {**  **background(255);**  **x = x + sin(radians(frameCount));**  **circle(x, height / 2, 50);**  **}** |  |
| --- | --- |

What if we don’t convert it?

| **let x;**  **let speedX = 5;**  **function setup() {**  **createCanvas(400, 400);**  **x = 0;**  **}**  **function draw() {**  **background(255);**  **x = x + sin(frameCount);**  **circle(x, height / 2, 50);**  **}** |  |
| --- | --- |

Let’s increase the amplitude

| **let x;**  **let speedX = 5;**  **function setup() {**  **createCanvas(400, 400);**  **x = 0;**  **}**  **function draw() {**  **background(255);**  **x = x + 10\*sin(frameCount);**  **circle(x, height / 2, 50);**  **}** |  |
| --- | --- |

Let’s decrease the angle

| **let x;**  **let speedX = 5;**  **function setup() {**  **createCanvas(400, 400);**  **x = 0;**  **}**  **function draw() {**  **background(255);**  **x = x + 10\*sin(frameCount\*0.1);**  **circle(x, height / 2, 50);**  **}** |  |
| --- | --- |

Now with what you know so far, try to move it this way:

| **let x, y;**  **function setup() {**  **createCanvas(400,400);**  **x = width/2;**  **y = -25;**  **}**  **function draw() {**  **background(0);**  **x = x + 10\*cos(frameCount\*0.1);**  **circle(x, y, 50);**  **y = y+3;**  **// Put it back!**  **if(y > height+25){**  **y = -25;**  **}**  **}** |  |
| --- | --- |

#### 

**Example 2:** position = startingPos (constant) + speed //speed is not constant, it is a sin value

| **let x, y;**  **function setup() {**  **createCanvas(400,400);**  **x = width/2;**  **y = 0;**  **}**  **function draw() {**  **background(0);**  **x = width/2 + 10\*sin(frameCount\*0.1);**  **circle(x, y, 50);**  **y = y+3;**  **if(y > height+25){**  **y = -25;**  **}**  **}** |  |
| --- | --- |

Kind of the same result but we can now control better the amplitude

| **let x, y;**  **function setup() {**  **createCanvas(400,400);**  **x = width/2;**  **y = 0;**  **}**  **function draw() {**  **background(0);**  **x = width/2 + 100\*sin(frameCount\*0.1);**  **circle(x, y, 50);**  **y = y+3;**  **if(y > height+25){**  **y = -25;**  **}**  **}** |  |
| --- | --- |

And the frequency by multiplying the variation of the angle (frameCount)

| **let x, y;**  **function setup() {**  **createCanvas(400,400);**  **x = width/2;**  **y = 0;**  **}**  **function draw() {**  **background(0);**  **x = width/2 + 100\*sin(frameCount\*0.05);**  **circle(x, y, 50);**  **y = y+3;**  **if(y > height+25){**  **y = -25;**  **}**  **}** |  |
| --- | --- |

100 is the amplitude and the angle is the frequency of the movement

The function SIN always moves in a range between -1 and 1, so if the Amplitude is 100 the ball will move from -100 to 100 from the starting position. But what if we want to move it in a different range?  
  
We can use the function **map** to make it easier (slide23)

| **let x, y;**  **function setup() {**  **createCanvas(400,400);**  **x = width/2;**  **y = 0;**  **}**  **function draw() {**  **background(0);**  **x = width/2 + 100\*sin(frameCount\*0.05);**  **fill(255);**  **circle(x, y, 50);**  **y = y+3;**  **if(y > height+25){**  **y = -25;**  **}**  **let x2 = map(sin(frameCount\*0.1),-1,1,0,400);**  **fill(255,0,255);**  **circle(x2, y, 50);**  **}** |  |
| --- | --- |

When is this useful too? When we need to translate a range of values into another one.  
For example we want to change the background color according to the mouse position  
Mouse position range of values: 0 to width

Color range of values: 0 to 255

| **let x, y;**  **function setup() {**  **createCanvas(400,400);**  **x = width/2;**  **y = 0;**  **}**  **function draw() {**  **background(0);**  **x = width/2 + 100\*sin(frameCount\*0.05);**  **fill(255);**  **circle(x, y, 50);**  **y = y+3;**  **if(y > height+25){**  **y = -25;**  **}**  **let x2 = map(sin(frameCount\*0.1),-1,1,0,400);**  **let c = map(sin(frameCount\*0.1), -1, 1, 0, 255);**  **fill(c, 0, 255);**  **circle(x2, y, 50);**  **}** |  |
| --- | --- |

A bit more fun with HSB colors ( hue(like the tone), saturation, brightness) <https://p5js.org/reference/#/p5/HSB>

CHECK THE EXTRA SLIDES ABOUT COLORS!

| **let x, y;**  **function setup() {**  **colorMode(HSB, 100);**  **createCanvas(400,400);**  **x = width/2;**  **y = 0;**  **}**  **function draw() {**  **background(0);**  **x = width/2 + 100\*sin(frameCount\*0.05);**  **fill(100);**  **circle(x, y, 50);**  **y = y+3;**  **if(y > height+25){**  **y = -25;**  **}**  **let x2 = map(sin(frameCount\*0.1),-1,1,0,400);**  **let c = map(sin(frameCount\*0.1), -1, 1, 0, 100);**  **fill(c,100,100);**  **circle(x2, y, 50);**  **}** |  |
| --- | --- |

**Example 3:** position = startingPos (constant) + speed //speed is not constant, it is a sin value

We can use another variable to change the angle, we don’t need to use frameCount, but we need a variable that is updating, for example y

| **let x, y;**  **function setup() {**  **colorMode(HSB, 100);**  **createCanvas(400,400);**  **x = width/2;**  **y = 0;**  **}**  **function draw() {**  **background(0);**  **x = width/2 + 100\*sin(y\*0.05);**  **fill(100);**  **circle(x, y, 50);**  **y = y+3;**  **if(y > height+25){**  **y = -25;**  **}**  **}** |  |
| --- | --- |

Other things you can change with sin speed:

Try to change the size!

| **let x, y;**  **function setup() {**  **colorMode(HSB, 100);**  **createCanvas(400,400);**  **x = width/2;**  **y = 0;**  **}**  **function draw() {**  **background(0);**  **x = width/2 + 100\*sin(y\*0.05);**  **fill(100);**  **let s = map(sin(y\*0.05), -1,1, 10,100);**  **circle(x, y, s);**  **y = y+3;**  **if(y > height+25){**  **y = -25;**  **}**  **}** |  |
| --- | --- |

## Noise:

Adding noise will make it move less predictable

| **let x, y;**  **function setup() {**  **colorMode(HSB, 100);**  **createCanvas(400,400);**  **x = width/2;**  **y = 0;**  **}**  **function draw() {**  **background(0);**  **let noiseVal = noise(frameCount\*0.01);**  **console.log(noiseVal);**  **x = width\*sin(noiseVal);**  **fill(100);**  **let s = map(sin(y\*0.05), -1,1, 10,100);**  **circle(x, y, 100);**  **y = y+1;**  **if(y > height+25){**  **y = -25;**  **}**  **}** |  |
| --- | --- |

We can change the movement direction to make the effect of bubbles. Changing the size could create the effect of a 3D space, like moving far and close

| **let x, y;**  **function setup() {**  **colorMode(HSB, 100);**  **createCanvas(400,400);**  **x = width/2;**  **y = 0;**  **}**  **function draw() {**  **background(0);**  **// x = width/2 + 100\*sin(y\*0.05);**  **// fill(100);**  **// let s = map(sin(y\*0.05), -1,1, 10,100);**  **// circle(x, y, s);**  **// y = y+3;**  **// if(y > height+25){**  **// y = -25;**  **// }**  **let noiseVal = noise(frameCount\*0.01);**  **console.log(noiseVal);**  **x = width/2 + 100\*sin(noiseVal);**  **fill(100);**  **let s = map(sin(y\*0.05), -1,1, 10,30);**  **circle(x, y, s);**  **y = y-1;**  **if(y < -25){**  **y = height-25;**  **}**  **}** |  |
| --- | --- |

#### **Circular motion**

Angle = angle + speed (Position = position + speed)

x = width/2 + cos(angle) \* radDist;

y = height/2 + sin(angle) \* radDist;

| **let x, y;**  **function setup() {**  **createCanvas(500, 500);**  **background(255);**  **}**  **function draw() {**  **background(0);**  **x = width / 2 + cos(frameCount\*0.1) \* 100;**  **y = height / 2 + sin(frameCount\*0.1) \* 100;**  **circle(x, y, 50);**  **}** |  |
| --- | --- |

Adding noise to the circular movement:

| **let x, y;**  **function setup() {**  **createCanvas(500, 500);**  **background(255);**  **}**  **function draw() {**  **background(0);**  **let noiseVal = map(noise(frameCount\*0.01), 0, 1, 0, 2\*PI);**  **x = width / 2 + cos(noiseVal) \* 100;**  **y = height / 2 + sin(noiseVal) \* 100;**  **circle(x, y, 50);**  **}** |  |
| --- | --- |

What about adding the noise to the Radius instead?

| **let x, y;**  **function setup() {**  **createCanvas(500, 500);**  **background(255);**  **}**  **function draw() {**  **background(0);**  **let noiseVal = map(noise(frameCount\*0.1), 0, 1, 100, 150);**  **x = width / 2 + cos(frameCount\*0.05) \* noiseVal;**  **y = height / 2 + sin(frameCount\*0.05) \* noiseVal;**  **circle(x, y, 50);**  **}** |  |
| --- | --- |

What happens if the angle is different for the cos and sin functions? Try it!  
Experiment changing the variables to find interesting results.

Spiral motion is the same as circular motion but the amplitude is variable, can you do that?

| **let x, y;**  **let a = 0;**  **function setup() {**  **createCanvas(400, 400);**  **background(255);**  **}**  **function draw() {**  **//background(0);**  **x = width / 2 + cos(frameCount\*0.1) \* a;**  **y = height / 2 + sin(frameCount\*0.1) \* a;**  **circle(x, y, 20);**  **a = a + 0.3;**  **}** |  |
| --- | --- |

Can you change the color? Can you draw another shape instead? Experiment!

| **let x, y;**  **let a = 0;**  **function setup() {**  **colorMode(HSB,100);**  **createCanvas(400, 400);**  **background(0);**  **}**  **function draw() {**  **//background(0);**  **x = width / 2 + cos(frameCount\*0.1) \* a;**  **y = height / 2 + sin(frameCount\*0.1) \* a;**  **let c = map(cos(frameCount\*0.1), -1,1,0,100);**  **stroke(c,50,100);**  **//noStroke();**  **line(width/2,height/2, x, y);**  **a = a + 0.3;**  **}** |  |
| --- | --- |

## Linear motion:

Position = position + speed // speed is a constant value

| **let x;**  **let speedX = 5;**  **function setup() {**  **createCanvas(400, 400);**  **x = 0;**  **}**  **function draw() {**  **background(255);**  **x = x + speedX;**  **circle(x, height / 2, 50);**  **}** |  |
| --- | --- |

How do you make it bounce?

| **let x;**  **let speedX = 5;**  **function setup() {**  **createCanvas(400, 400);**  **x = 0;**  **}**  **function draw() {**  **background(255);**  **x = x + speedX;**  **circle(x, height / 2, 50);**  **//conditionals!**  **if(x > width || x < 0){**  **speedX = -speedX;**  **}**  **}** |  |
| --- | --- |

How do you make it return as in the gif?

| **let x;**  **let speedX = 5;**  **function setup() {**  **createCanvas(400, 400);**  **x = 0;**  **}**  **function draw() {**  **background(255);**  **x = x + speedX;**  **circle(x, height / 2, 50);**  **//conditionals!**  **if(x > width){**  **x = 0;**  **}**  **}** |  |
| --- | --- |

We can also use a variable to change the speed interactively:

| **let x;**  **let y;**  **function setup() {**  **createCanvas(400, 400);**  **x = 0;**  **}**  **function draw() {**  **background(220);**  **circle(x,height/2, 50);**  **x = x + mouseY/20;**  **if(x > width){**  **x = 0;**  **}**  **}** |  |
| --- | --- |

In these examples the speed is constant, but it doesn’t need to be!

| **let x;**  **let speedX = 5;**  **function setup() {**  **createCanvas(400, 400);**  **x = 0;**  **}**  **function draw() {**  **background(255);**  **circle(x, height / 2, 50);**  **console.log(frameCount);**  **}** |  |
| --- | --- |

We can use this variable to move things:

| **let x;**  **//let speedX = 5;**  **function setup() {**  **createCanvas(400, 400);**  **x = 0;**  **}**  **function draw() {**  **background(255);**  **x = frameCount;**  **circle(x, height / 2, 50);**  **console.log(frameCount);**  **}** |  |
| --- | --- |

But we can also use this variable to accelerate things (you can create your own acceleration variable too)

| **let x;**  **//let speedX = 5;**  **function setup() {**  **createCanvas(400, 400);**  **x = 0;**  **}**  **function draw() {**  **background(255);**  **x = x + frameCount;**  **circle(x, height / 2, 50);**  **console.log(frameCount);**  **if(x > width){**  **x = 0;**  **}**  **}** |  |
| --- | --- |

It’s super fast, so let’s reduce it:

| **let x;**  **//let speedX = 5;**  **function setup() {**  **createCanvas(400, 400);**  **x = 0;**  **}**  **function draw() {**  **background(255);**  **x = x + frameCount\*0.1;**  **circle(x, height / 2, 50);**  **console.log(frameCount);**  **if(x > width){**  **x = 0;**  **}**  **}** |  |
| --- | --- |

What type of variable have we been using so far?  
Do you remember another type of variable?  
String, Integer, Float, Boolean

## GENERATIVE ART and RANDOMNESS

**How computational generative art aligns or contrasts with the artworks you've created for Mini Projects 1 and 2.**

**What is your understanding of Generative Art?**

**What does the function random(); do?**

What if we let the computer decide where the ball is going to start moving?

| **let x;**  **//let speedX = 5;**  **function setup() {**  **createCanvas(400, 400);**  **x = random(0, width);**  **console.log(x);**  **}**  **function draw() {**  **background(255);**  **x = x + frameCount\*0.1;**  **circle(x, height / 2, 50);**  **}** |  |
| --- | --- |

**Let’s try to do it with the y axis:**

| **let x;**  **//let speedX = 5;**  **function setup() {**  **createCanvas(400, 400);**  **x = 0;**  **}**  **function draw() {**  **background(255);**  **x = x + frameCount\*0.1;**  **circle(x, random(0, height), 50);**  **}** |  |
| --- | --- |

**We only the computer choose one at the beginning**

| **let x, y;**  **function setup() {**  **createCanvas(400, 400);**  **x = 0;**  **y = random(0, height);**  **console.log(y);**  **}**  **function draw() {**  **background(255);**  **x = x + frameCount\*0.1;**  **circle(x, y, 50);**  **}** |  |
| --- | --- |

What if the speed is random?

| **let x, y;**  **function setup() {**  **createCanvas(400, 400);**  **x = width/2;**  **y = random(0, height);**  **console.log(y);**  **}**  **function draw() {**  **background(255);**  **x = x + random(-frameCount\*0.01,frameCount\*0.01);**  **circle(x, y, 50);**  **}** |  |
| --- | --- |

Add movement in the Y axis too.

| **let x, y;**  **function setup() {**  **createCanvas(400, 400);**  **x = width/2;**  **y = random(0, height);**  **console.log(y);**  **}**  **function draw() {**  **background(255,5);//transparency**  **x = x + random(-frameCount\*0.01,frameCount\*0.01);**  **y = y + random(-frameCount\*0.01,frameCount\*0.01);**  **fill(0);**  **circle(x, y, 10);**  **}** |  |
| --- | --- |

Can you change other variables too? Size, color?

| **let x, y;**  **function setup() {**  **createCanvas(400, 400);**  **x = width/2;**  **y = random(0, height);**  **console.log(y);**  **colorMode(HSB,100);**  **background(0);**  **}**  **function draw() {**  **//background(255);**  **x = x + random(-frameCount\*0.01,frameCount\*0.01);**  **y = y + random(-frameCount\*0.01,frameCount\*0.01);**  **stroke(random(100), 50,100);**  **noFill();**  **circle(x, y, random(1,40));**  **}** |  |
| --- | --- |

**What’s the difference between this movement and movements in nature?**

How would you keep it inside the canvas?

Check the video in the slides: [Coding Challenge #52: Random Walker](https://www.youtube.com/watch?v=l__fEY1xanY)

# VARIABLES, CONDITIONALS & INTERACTION

## **Review**

How can we place a square of size 100 in the middle of the screen?

Let’s draw a rectangle in the middle of the screen

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **rectMode(CENTER);**  **rect(width/2,height/2, 100, 100);**  **}** |  |
| --- | --- |

How can we update the position in the X axis of the rectangle?

| **//declare!**  **let x;**  **function setup() {**  **createCanvas(400, 400);**  **//assign value**  **x = width/2;**  **}**  **function draw() {**  **background(220);**  **rectMode(CENTER);**  **//use!**  **rect(x, height/2 , 100, 100);**  **//update**  **x = x + 1;**  **}** |  |
| --- | --- |

How can we make it move in the Y axis too?

| **//declare!**  **let x;**  **let y;**  **function setup() {**  **createCanvas(400, 400);**  **//assign value**  **x = 50;**  **y = 50;**  **}**  **function draw() {**  **background(220);**  **rectMode(CENTER);**  **//use!**  **rect(x, y, 100, 100);**  **//update**  **x = x+2;**  **y = y+3;**  **}** |  |
| --- | --- |

#### 

What type of variable have we been using so far?  
Do you remember another type of variable?

What type of motion is this?  
  
Linear motion:

Position = position + speed // speed is a constant value

We update the position, but there are other ways to update it. We just need something that changes over the time:

## FrameCount

Slide 9

| **//declare!**  **let x;**  **let y;**  **function setup() {**  **createCanvas(400, 400);**  **//assign value**  **x = 50;**  **y = 50;**  **}**  **function draw() {**  **background(220);**  **rectMode(CENTER);**  **//use!**  **rect(x, y, 100, 100);**  **//update**  **x = x+2;**  **y = y+3;**  **console.log(frameCount);**  **}** |  |
| --- | --- |

Then we can use this to update position for example

| **//declare!**  **let x;**  **let y;**  **function setup() {**  **createCanvas(400, 400);**  **//assign value**  **x = 50;**  **y = 50;**  **}**  **function draw() {**  **background(220);**  **rectMode(CENTER);**  **//use!**  **rect(x, y, 100, 100);**  **//update**  **x = frameCount;**  **y = y+3;**  **console.log(frameCount);**  **}** |  |
| --- | --- |

Maybe to change something else

| **//declare!**  **let x;**  **let y;**  **function setup() {**  **createCanvas(400, 400);**  **//assign value**  **x = 50;**  **y = 50;**  **}**  **function draw() {**  **background(220);**  **rectMode(CENTER);**  **//use!**  **rect(x, y, frameCount, 100);**  **//update**  **x = x+2;**  **y = y+3;**  **console.log(frameCount);**  **}** |  |
| --- | --- |

String, Integer, Float, Boolean

## **Boolean Variables & Conditionals**

Boolean Variables: Slides 12

Conditionals: Slide 17

A conditional statement is based on a concept of if / then / else.

#### **Operators**

Boolean variable **==** true

Boolean variable **>=**   
Boolean variable==false **&&**

So, let’s try to keep the square inside the window:

| **//declare!**  **let x;**  **let y;**  **let sx = 2;**  **let sy = 3;**  **function setup() {**  **createCanvas(400, 400);**  **//assign value**  **x = 50;**  **y = 50;**  **}**  **function draw() {**  **background(220);**  **rectMode(CENTER);**  **//use!**  **rect(x, y, 100, 100);**  **//update**  **x = x + sx;**  **y = y + sy;**    **if(x > width-50 || x<50){**  **sx = -sx;**  **}**  **if(y > height-50 || y<50){**  **sy = -sy;**  **}**  **}** |  |
| --- | --- |

#### Can you make something fall or go up repeatedly?

| **//declare!**  **let x;**  **let y ;**  **function setup() {**  **createCanvas(400, 400);**  **//assign value**  **x = width/2;**  **y = height;**  **}**  **function draw() {**  **background(220);**  **fill(200,0,255);**  **noStroke();**  **circle(x-20, y, 50);**  **circle(x+20, y, 50);**  **triangle(x-43, y+12, x+43, y+12, x, y+50);**  **y = y -3;**    **if(y < 0){**  **y = height+50;**  **}**  **}** |  |
| --- | --- |

## INTERACTIONS

#### **Mouse**

mouseX

mouseY

pmouseX

pmouseY

| **function setup() {**  **createCanvas(400, 400);**  **background(220);**  **}**  **function draw() {**  **line(pmouseX, pmouseY, mouseX, mouseY);**  **}** |  |
| --- | --- |

#### **mouseIsPressed**

**mouseIsPressed is an environment variable that stores the current state of the mouse button as a boolean value, true for pressed and false for not pressed.**

| **function setup() {**  **createCanvas(400, 400);**  **background(220);**  **}**  **function draw() {**  **if (mouseIsPressed == true) {**  **ellipse(mouseX, mouseY, 10, 10);**  **}**  **}** |  |
| --- | --- |

#### **keyIsPressed**

| **function setup() {**  **createCanvas(400, 400);**  **background(220);**  **}**  **function draw() {**  **if (mouseIsPressed == true) {**  **ellipse(mouseX, mouseY, 10, 10);**  **}**  **if (keyIsPressed == true) {**  **fill(255,0,0);**  **console.log(key);**  **}**  **}** |  |
| --- | --- |

**How to stop it?**

| **function setup() {**  **createCanvas(400, 400);**  **background(220);**  **}**  **function draw() {**  **if (mouseIsPressed == true) {**  **ellipse(mouseX, mouseY, 10, 10);**  **}**  **if (keyIsPressed == true) {**  **fill(255,0,0);**  **console.log(key);**  **}**  **else{**  **noFill();**  **}**  **}** |  |
| --- | --- |

**Changing color according to the key pressed**

| **function setup() {**  **createCanvas(400, 400);**  **background(220);**  **}**  **function draw() {**  **if (mouseIsPressed) {**  **ellipse(mouseX, mouseY, 10, 10);**  **}**  **if (keyIsPressed == true) {**  **console.log(key);**  **if (key == "r") {**  **fill(255, 0, 0);**  **}**  **if (key == "g") {**  **fill(0, 255, 0);**  **}**  **} else {**  **noFill();**  **}**  **}** |  |
| --- | --- |

**Specific keys but changing the shape**

| **function setup() {**  **createCanvas(400, 400);**  **background(220);**  **}**  **function draw() {**  **if (keyIsPressed == true) {**  **if (key == "e" || key == "E") {**  **ellipse(mouseX, mouseY, 50, 50);**  **} else if (key == "r" || key == "R") {**  **rect(mouseX, mouseY, 50, 50);**  **}**  **}**  **}** |  |
| --- | --- |

**keyCode:**

**UP\_ARROW, DOWN\_ARROW, LEFT\_ARROW, RIGHT\_ARROW, BACKSPACE, DELETE, ENTER, RETURN, TAB, ESCAPE, SHIFT, CONTROL, OPTION, ALT**

| **function setup() {**  **createCanvas(400, 400);**  **background(220);**  **}**  **function draw() {**  **if (keyIsPressed) {**  **if (keyCode == BACKSPACE) {**  **background(220);**  **}**  **}**  **ellipse(mouseX, mouseY, 50, 50);**  **}** |  |
| --- | --- |

#### **Exercise**

**Draw a circle in the middle of the screen**

**Now move the shape with the arrows**

**UP\_ARROW, DOWN\_ARROW, LEFT\_ARROW, RIGHT\_ARROW (slide 93)**

| **let x ;**  **let y ;**  **function setup() {**  **createCanvas(400, 400);**  **background(220);**  **x = width/2;**  **y = height/2;**  **}**  **function draw() {**  **if (keyIsPressed) {**  **if (keyCode == UP\_ARROW) {**  **y = y - 1;**  **} else if (keyCode == DOWN\_ARROW) {**  **y = y + 1;**  **} else if (keyCode == LEFT\_ARROW) {**  **x = x - 1;**  **} else if (keyCode == RIGHT\_ARROW) {**  **x = x + 1;**  **}**  **}**  **ellipse(x, y, 50, 50);**  **}** |  |
| --- | --- |

# GENERATIVE ART & VARIABLES

## **Recitation**

Problems you faced in the last exercise?

What do you think about drawing with code so far?

## **Review**

Order of execution (slides)

## **How to know what’s going on?**

#### **Errors and debugging**

Slides 52

#### **console.log();**

For example, difference between setup and draw

| **function setup() {**  **console.log("hello");**  **}**  **function draw() {**  **}** |  |
| --- | --- |

| **function setup() {**    **}**  **function draw() {**  **console.log("hello");**  **}** |  |
| --- | --- |

Now with a variable

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **console.log(width);**  **}** |  |
| --- | --- |

**width** and **height** are **Environment Variables**. They store the size of the canvas  
But we can create our own variables:

| **let x = 0; //declare+initialize**  **function setup() {**  **console.log(x);**  **}**  **function draw() {**  **x = x + 1; //update**  **}** |  |
| --- | --- |

| **let x = 0; //declare+initialize**  **function setup() {**    **}**  **function draw() {**  **console.log(x);**  **x = x + 1; //update**  **}** |  |
| --- | --- |

## **Variables:**

Slide 59 - 65

Variables are containers for storing values.

#### **Environment Variables**

What if I want my shape in the middle?

| **function setup() {**  **createCanvas(500,500);**  **background(255);**  **}**  **function draw() {**  **//face big circle**  **circle(250,250, 200);**  **//eyes**  **fill(0);**  **circle(250-40, 250, 10);**  **circle(250+40, 250, 10);**  **//mouth**  **noFill();**  **arc(250, 250+10, 20,20, 0, PI);**  **}** |  |
| --- | --- |

But what happens if I change the canvas size?

| **function setup() {**  **createCanvas(800,800);**  **background(255);**  **}**  **function draw() {**  **//face big circle**  **circle(250,250, 200);**  **//eyes**  **fill(0);**  **circle(250-40, 250, 10);**  **circle(250+40, 250, 10);**  **//mouth**  **noFill();**  **arc(250, 250+10, 20,20, 0, PI);**  **}** |  |
| --- | --- |

We can use instead width and height as variables

| **function setup() {**  **createCanvas(500,500);**  **background(255);**  **}**  **function draw() {**  **circle(250,250, 200);**  **// circle(width/2, height/2, 200);**  //I can continue drawing in function of these variables  **fill(0);**  **circle(width/2-40, height/2, 10);**  **circle(width/2+40, height/2, 10);**  **noFill();**  **arc(width/2, height/2+10, 20,20, 0, PI);**  **}** |  |
| --- | --- |

#### **JavaScript Variables**

| **let x = 200;**  //declare+initialize  **function setup() {**  **createCanvas(500,500);**  **background(255);**  **}**  **function draw() {**  **circle(width/2, height/2, x);**  //use it!  **fill(0);**  **circle(width/2-40, height/2, 10);**  **circle(width/2+40, height/2, 10);**  **noFill();**  **arc(width/2, height/2+10, 20,20, 0, PI);**  **x = x + 1;**  //update  **console.log(x);**  **}** |  |
| --- | --- |

**How can I increase the speed?**

**x = x + 5;**  //update

| **let x = 200;**  //declare+initialize  **function setup() {**  **createCanvas(500,500);**  **background(255);**  **}**  **function draw() {**  **circle(width/2, height/2, x);**  //use it!  **fill(0);**  **circle(width/2-40, height/2, 10);**  **circle(width/2+40, height/2, 10);**  **noFill();**  **arc(width/2, height/2+10, 20,20, 0, PI);**  **x = x + 5;**  //update  **console.log(x);**  **}** |  |
| --- | --- |

**Why am I seeing the circle repeated?**

#### **JavaScript Variables Types**

Slide 71

#### **Naming variables**

Slide 77

#### **JavaScript Variables Global and Local**

| **function setup() {**  **createCanvas(500, 500);**  **background(255);**  **let x = 200; //declare+initialize**  **}**  **function draw() {**    **circle(width / 2, height / 2, x); //use it!**  **fill(0);**  **circle(width / 2 - 40, height / 2, 10);**  **circle(width / 2 + 40, height / 2, 10);**  **noFill();**  **arc(width / 2, height / 2 + 10, 20, 20, 0, PI);**  **x = x + 1; //update**  **console.log(x);**  **}** |  |
| --- | --- |

| **function setup() {**  **createCanvas(500, 500);**  **background(255);**  **}**  **function draw() {**  **let x = 200; //declare+initialize**  **circle(width / 2, height / 2, x); //use it!**  **fill(0);**  **circle(width / 2 - 40, height / 2, 10);**  **circle(width / 2 + 40, height / 2, 10);**  **noFill();**  **arc(width / 2, height / 2 + 10, 20, 20, 0, PI);**  **x = x + 1; //update**  **console.log(x);**  **}** |  |
| --- | --- |

**Why isn’t the value X increasing?**

# DRAWING WITH CODE

Let’s draw a circle (ellipse also works) in the middle of the screen:

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **circle(200,200,10);**  **}** |  |
| --- | --- |

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **circle(100,150,10);**  **}** |  |
| --- | --- |

Another one

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **circle(100,150,10);**  **circle(300,150,10);**  **}** |  |
| --- | --- |

What about a rectangle?

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **circle(100,150,10);**  **circle(300,150,10);**  **rect(100,200,200,20);**  **}** |  |
| --- | --- |

We can use comments to organize our code. These are like mental notes, so we will use our language and not the computer language. But we need to hide that from the computer by using **//**

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **//these are the eyes**  **circle(100,150,10);**  **circle(300,150,10);**  **//this is the mouth**  **rect(100,200,200,20);**  **}** |  |
| --- | --- |

Tip: place the cursor in any line or select many lines and use **COMMAND KEY+ /** to comment them faster

What if we want to make teeth? We can draw several rectangles and move them in the X axis:

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **// these are the eyes**  **circle(100,150,10);**  **circle(300,150,10);**  **//this is the mouth**  **rect(100,200,20,20);**  **rect(120,200,20,20);**  **rect(140,200,20,20);**  **rect(160,200,20,20);**  **rect(180,200,20,20);**  **rect(200,200,20,20);**  **rect(220,200,20,20);**  **rect(240,200,20,20);**  **rect(260,200,20,20);**  **rect(280,200,20,20);**    **}** |  |
| --- | --- |

Let’s use the arc shape instead of a rectangle:

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **// these are the eyes**  **circle(100,150,10);**  **circle(300,150,10);**    **//this is the mouth**  **// rect(100,200,20,20);**  **// rect(120,200,20,20);**  **// rect(140,200,20,20);**  **// rect(160,200,20,20);**  **// rect(180,200,20,20);**  **// rect(200,200,20,20);**  **// rect(220,200,20,20);**  **// rect(240,200,20,20);**  **// rect(260,200,20,20);**  **// rect(280,200,20,20);**  **arc(200,200, 50,50,0, PI);**    **}** |  |
| --- | --- |

Visit the references and find the way to close the arc, so there is an horizontal black line closing the arc.  
<https://p5js.org/reference/#/p5/arc>

Can we get rid of the outline?  
Can we get rid of the inside color?

How can we change colors?

Removing the outline: **noStroke();**

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **// these are the eyes**  **circle(100,150,10);**  **circle(300,150,10);**  **noStroke();**  **//this is the mouth**  **// rect(100,200,20,20);**  **// rect(120,200,20,20);**  **// rect(140,200,20,20);**  **// rect(160,200,20,20);**  **// rect(180,200,20,20);**  **// rect(200,200,20,20);**  **// rect(220,200,20,20);**  **// rect(240,200,20,20);**  **// rect(260,200,20,20);**  **// rect(280,200,20,20);**  **arc(200,200, 50,50,0, PI);**    **}** |  |
| --- | --- |

Removing the inside color: noFill();

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **// these are the eyes**  **circle(100,150,10);**  **circle(300,150,10);**  **noFill();**  **//this is the mouth**  **// rect(100,200,20,20);**  **// rect(120,200,20,20);**  **// rect(140,200,20,20);**  **// rect(160,200,20,20);**  **// rect(180,200,20,20);**  **// rect(200,200,20,20);**  **// rect(220,200,20,20);**  **// rect(240,200,20,20);**  **// rect(260,200,20,20);**  **// rect(280,200,20,20);**  **arc(200,200, 50,50,0, PI);**    **}** |  |
| --- | --- |

But, how can we change colors? [Check the extra slides about color!](https://docs.google.com/presentation/d/1WupcUMCAJhNmw9282I51Dsp7eHVGR2ptnP7hsmmcKzY/edit?usp=sharing)

The function stroke() will change color

The function strokeWeight() will change the thickness of the line

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **// these are the eyes**  **circle(100,150,10);**  **circle(300,150,10);**  **noFill();**  **stroke(255,0,255);**  **strokeWeight(4);**  **//this is the mouth**  **// rect(100,200,20,20);**  **// rect(120,200,20,20);**  **// rect(140,200,20,20);**  **// rect(160,200,20,20);**  **// rect(180,200,20,20);**  **// rect(200,200,20,20);**  **// rect(220,200,20,20);**  **// rect(240,200,20,20);**  **// rect(260,200,20,20);**  **// rect(280,200,20,20);**  **arc(200,200, 50,50,0, PI);**    **}** |  |
| --- | --- |

Now if you want to change the inside color you can use fill()

And if you want to get rid of the stroke you can use noStroke()

| **function setup() {**  **createCanvas(400, 400);**  **}**  **function draw() {**  **background(220);**  **// these are the eyes**  **circle(100,150,10);**  **circle(300,150,10);**  **noStroke();**  **fill(255,0,255);**  **//this is the mouth**  **// rect(100,200,20,20);**  **// rect(120,200,20,20);**  **// rect(140,200,20,20);**  **// rect(160,200,20,20);**  **// rect(180,200,20,20);**  **// rect(200,200,20,20);**  **// rect(220,200,20,20);**  **// rect(240,200,20,20);**  **// rect(260,200,20,20);**  **// rect(280,200,20,20);**  **arc(200,200, 50,50,0, PI);**    **}** |  |
| --- | --- |

Can you make this?

|  |  |
| --- | --- |

Find colors easily using the color picker: <https://g.co/kgs/CgiibVL>

Check the extra slides about color: [1.2extra Colors | F24](https://docs.google.com/presentation/d/1WupcUMCAJhNmw9282I51Dsp7eHVGR2ptnP7hsmmcKzY/edit?usp=sharing)