CREATIVE CODING

with p5.js

Particle Plotter by Vamoss
https://openprocessing.org/sketch/751983

Loading

preload() // Ensures that all assets are loaded before setup() and draw() are called

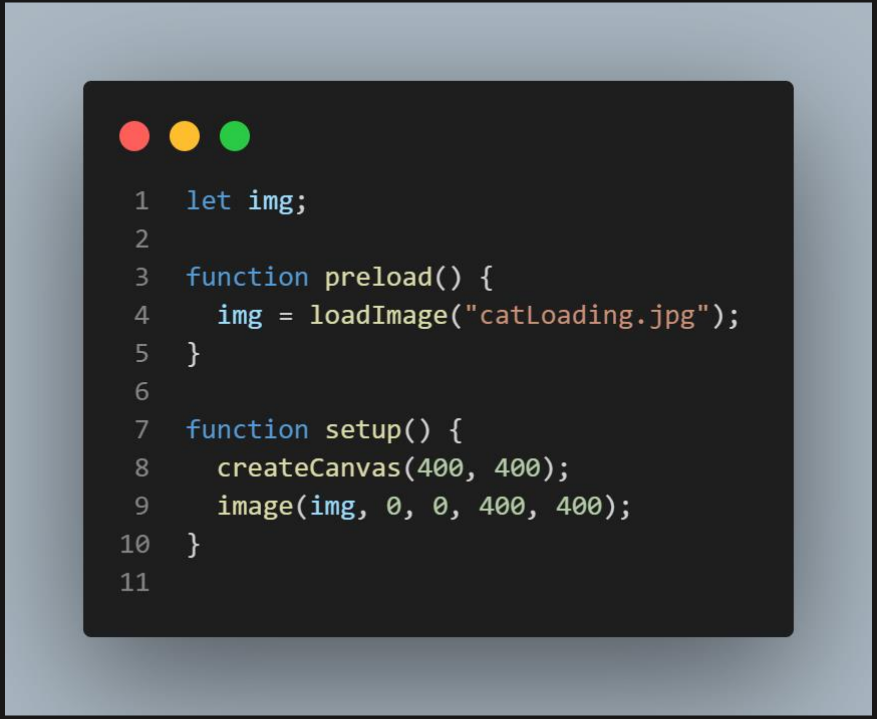
loadImage()

loadJSON()

loadFont()

loadStrings()

Need to run a server to load images -> «Live Server» (extension) in vscode



```
let img;

function preload() {
  img = loadImage("catLoading.jpg");
}

function setup() {
  createCanvas(400, 400);
  image(img, 0, 0, 400, 400);
}
```

Transformations

rotate(angle)

angleMode(DEGREES)

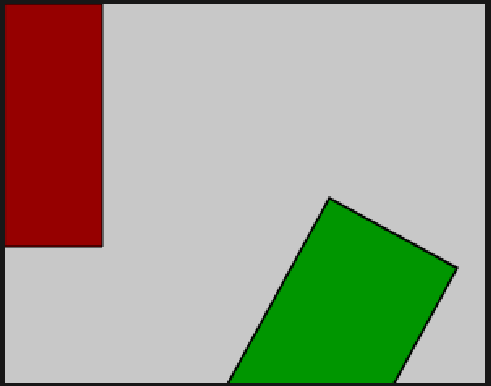
// Default: RADIANS

scale(x, y)

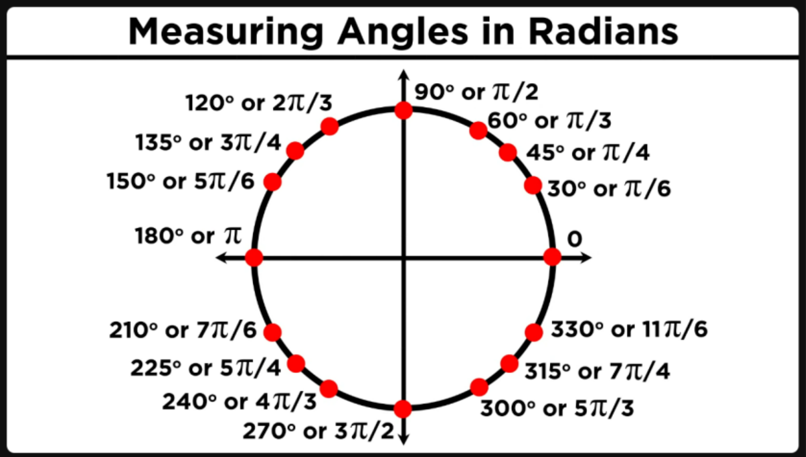
translate(x, y)

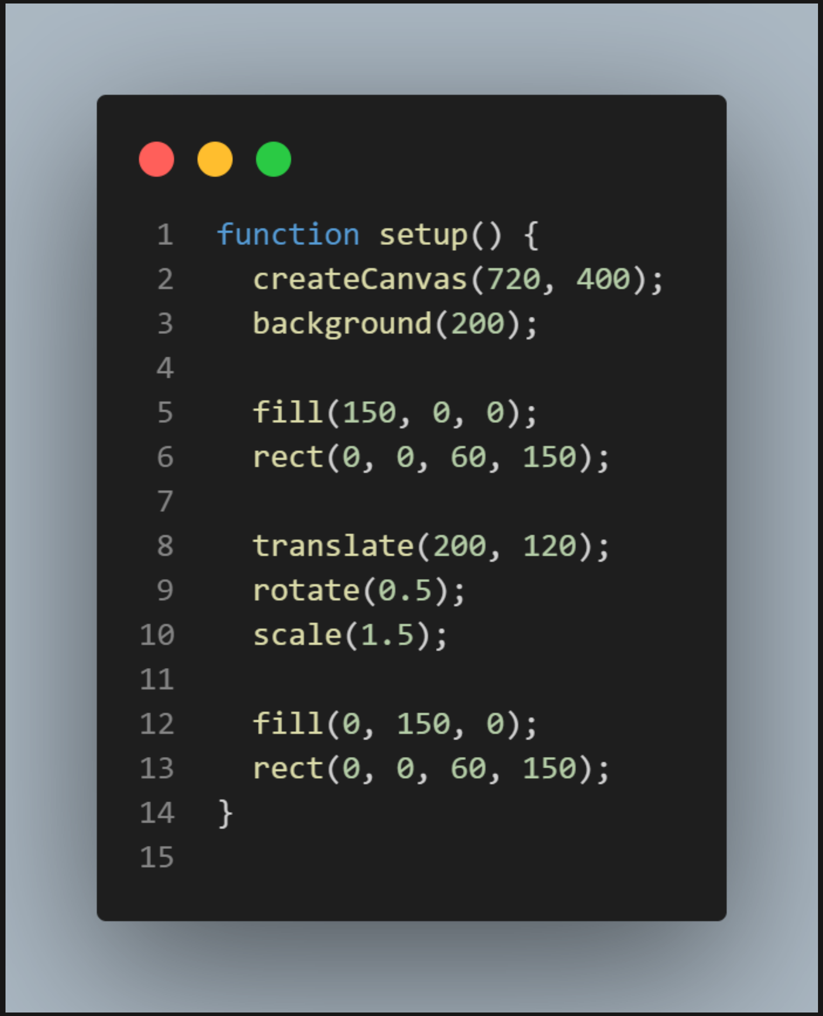
// x = left/right

// y = up/down



https://www.youtube.com/watch?v=0GkmnPdD6jY





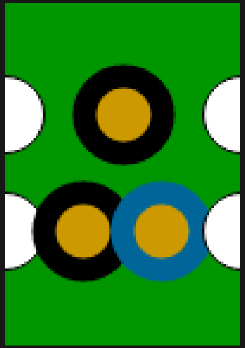
```
function setup() {
  createCanvas(720, 400);
  background(200);

  fill(150, 0, 0);
  rect(0, 0, 60, 150);

  translate(200, 120);
  rotate(0.5);
  scale(1.5);

  fill(0, 150, 0);
  rect(0, 0, 60, 150);
}
```

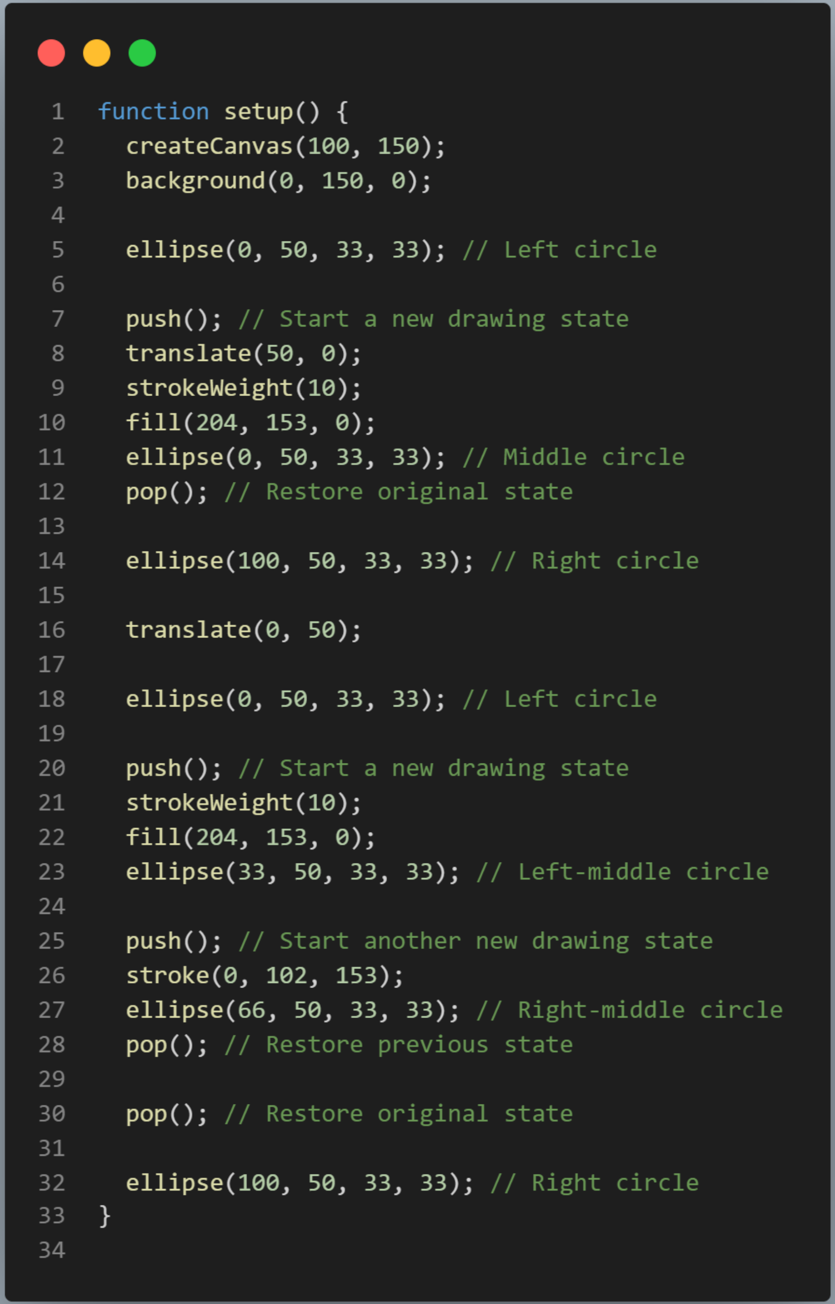
Save style settings

push()

pop()

Can be embedded!





```
function setup() {
  createCanvas(100, 150);
  background(0, 150, 0);

  ellipse(0, 50, 33, 33); // Left circle

  push(); // Start a new drawing state
  translate(50, 0);
  strokeWeight(10);
  fill(204, 153, 0);
  ellipse(0, 50, 33, 33); // Middle circle
  pop(); // Restore original state

  ellipse(100, 50, 33, 33); // Right circle

  translate(0, 50);

  ellipse(0, 50, 33, 33); // Left circle

  push(); // Start a new drawing state
  strokeWeight(10);
  fill(204, 153, 0);
  ellipse(33, 50, 33, 33); // Left-middle circle

  push(); // Start another new drawing state
  stroke(0, 102, 153);
  ellipse(66, 50, 33, 33); // Right-middle circle
  pop(); // Restore previous state

  pop(); // Restore original state

  ellipse(100, 50, 33, 33); // Right circle
}

```

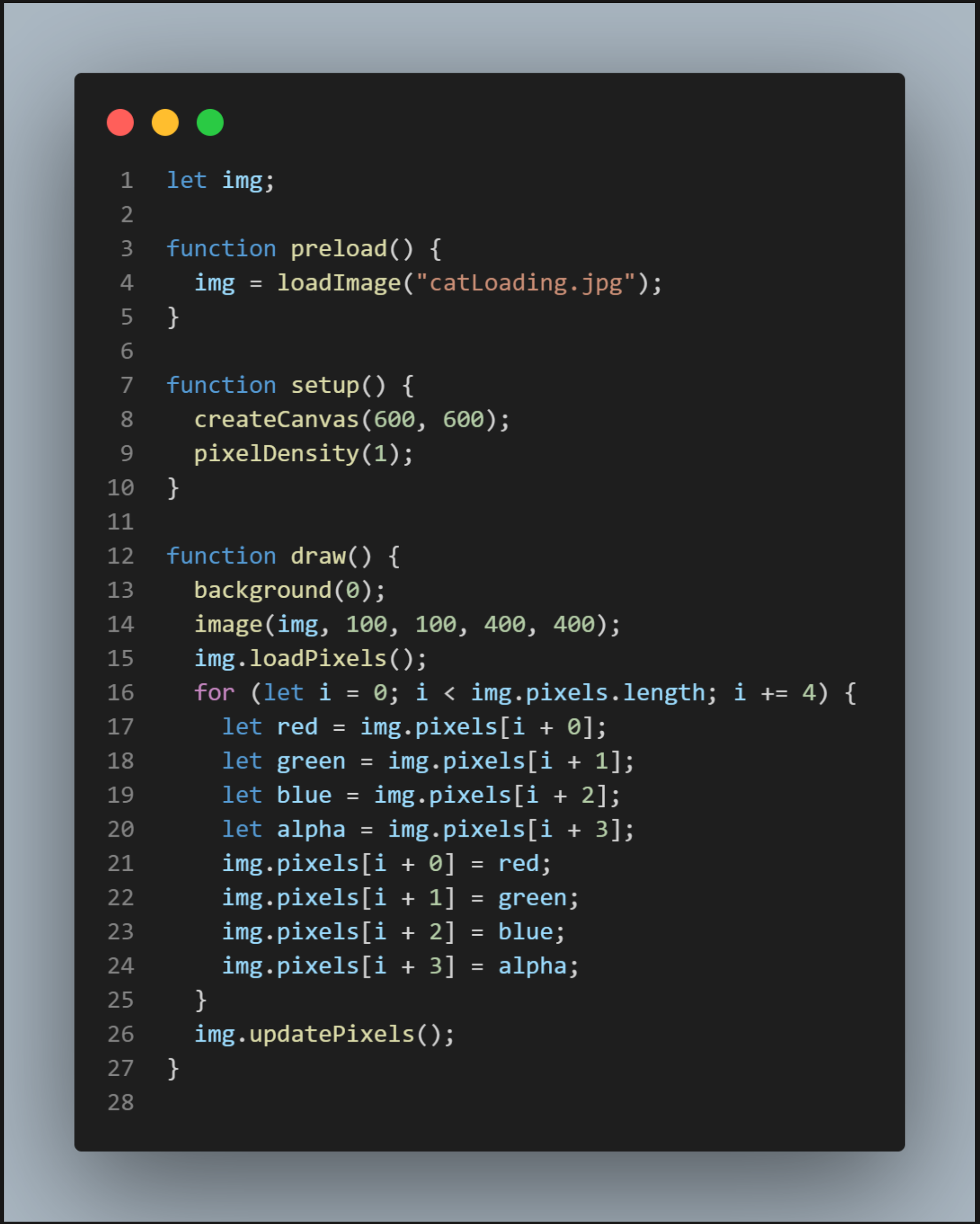
Pixels

pixelDensity(1)

Ensures that each virtual pixel corresponds to exactly one physical pixels, regardless of the device's pixel density

loadPixels()

updatePixels()

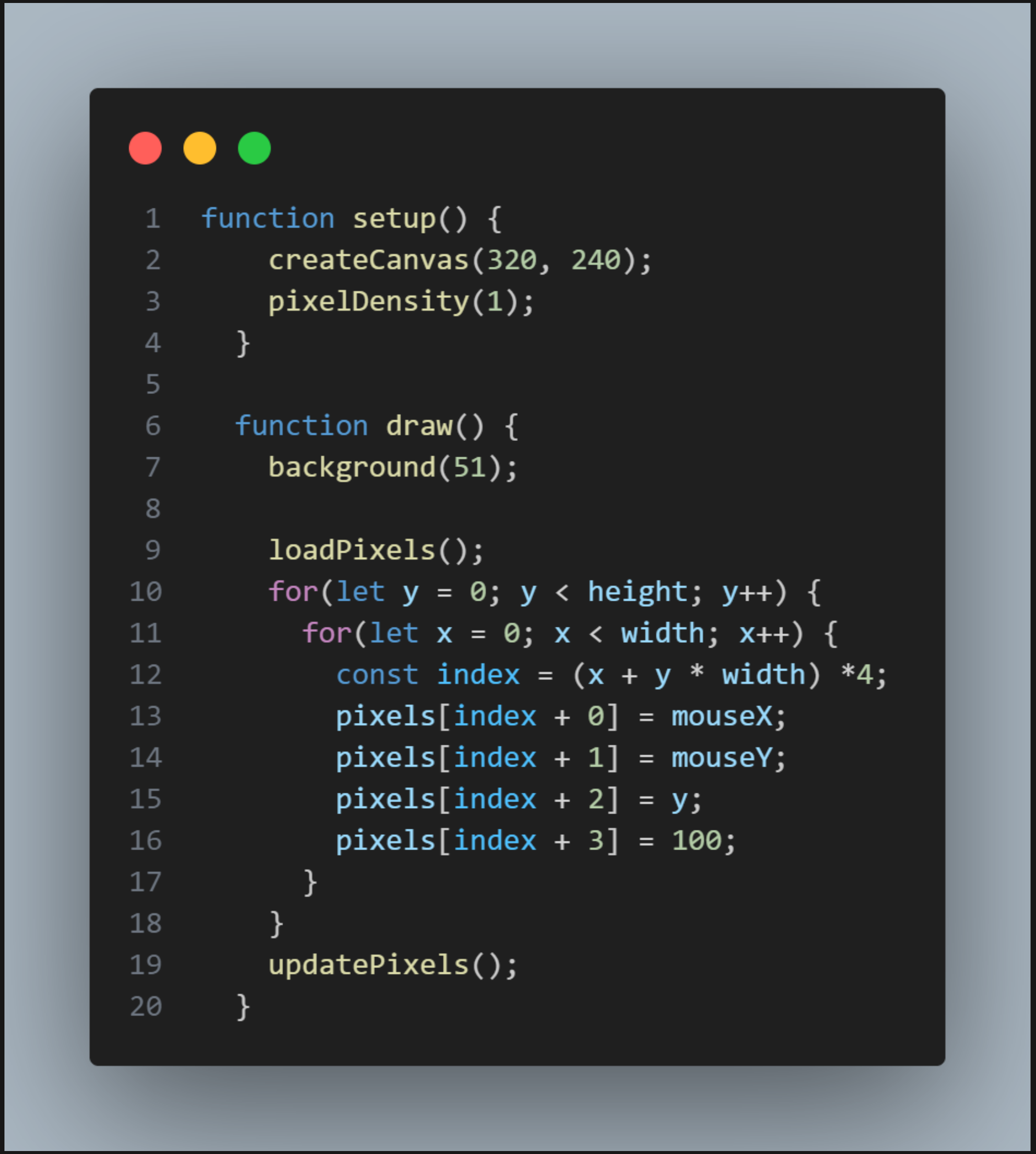


```
let img;

function preload() {
  img = loadImage("catLoading.jpg");
}

function setup() {
  createCanvas(600, 600);
  pixelDensity(1);
}

function draw() {
  background(0);
  image(img, 100, 100, 400, 400);
  img.loadPixels();
  for (let i = 0; i < img.pixels.length; i += 4) {
    let red = img.pixels[i + 0];
    let green = img.pixels[i + 1];
    let blue = img.pixels[i + 2];
    let alpha = img.pixels[i + 3];
    img.pixels[i + 0] = red;
    img.pixels[i + 1] = green;
    img.pixels[i + 2] = blue;
    img.pixels[i + 3] = alpha;
  }
  img.updatePixels();
}

```

Pixels

pixels[] = 1D array!

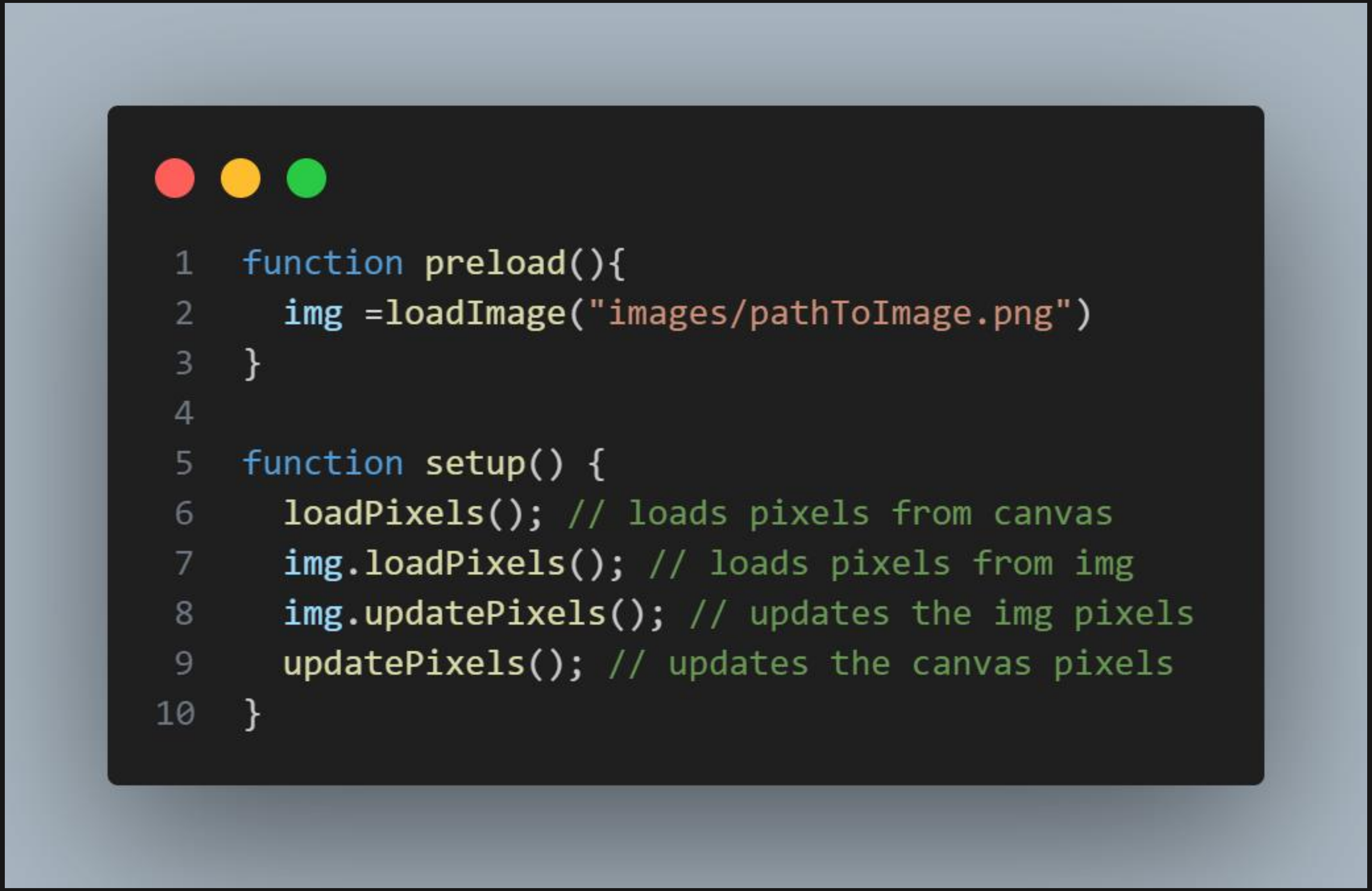
Either increment by +4,
or calculate the index like:
let index = (x + y * width) * 4;



```
function setup() {
    createCanvas(320, 240);
    pixelDensity(1);
  }

  function draw() {
    background(51);

    loadPixels();
    for(let y = 0; y < height; y++) {
      for(let x = 0; x < width; x++) {
        const index = (x + y * width) *4;
        pixels[index + 0] = mouseX;
        pixels[index + 1] = mouseY;
        pixels[index + 2] = y;
        pixels[index + 3] = 100;
      }
    }
    updatePixels();
  }
```

Pixels

Careful:

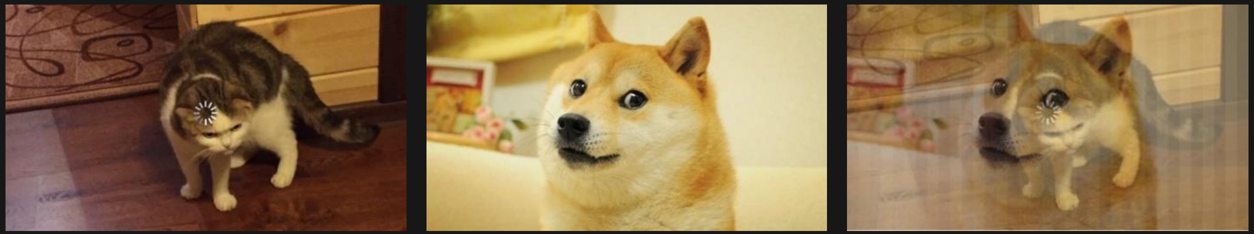


```
function preload(){
  img =loadImage("images/pathToImage.png")
}

function setup() {
  loadPixels(); // loads pixels from canvas
  img.loadPixels(); // loads pixels from img
  img.updatePixels(); // updates the img pixels
  updatePixels(); // updates the canvas pixels
}
```

Tasks 1/2

Image manipulation:

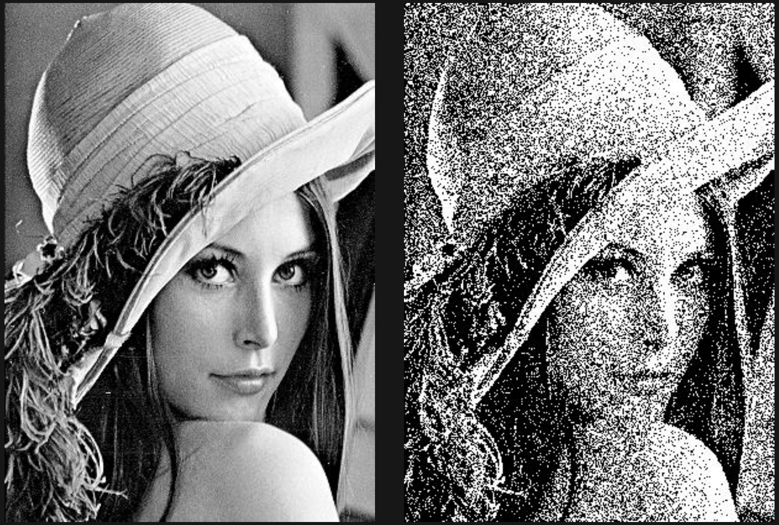
- Create your own image filter
- Sort image pixels by (hue, saturation, lightness, occurance, …)
- Resize / scale an image based on interactivity (mouseX / mouseY /…)
  - Resize with image(…) not with img.resize(…)
    - img.resize(…) uses the PixelDensity(N) => gets blurry.
  - Challenge: Write your own scaling algorithm
- Create an image so, that every odd pixel is from image A, every even pixel from image B



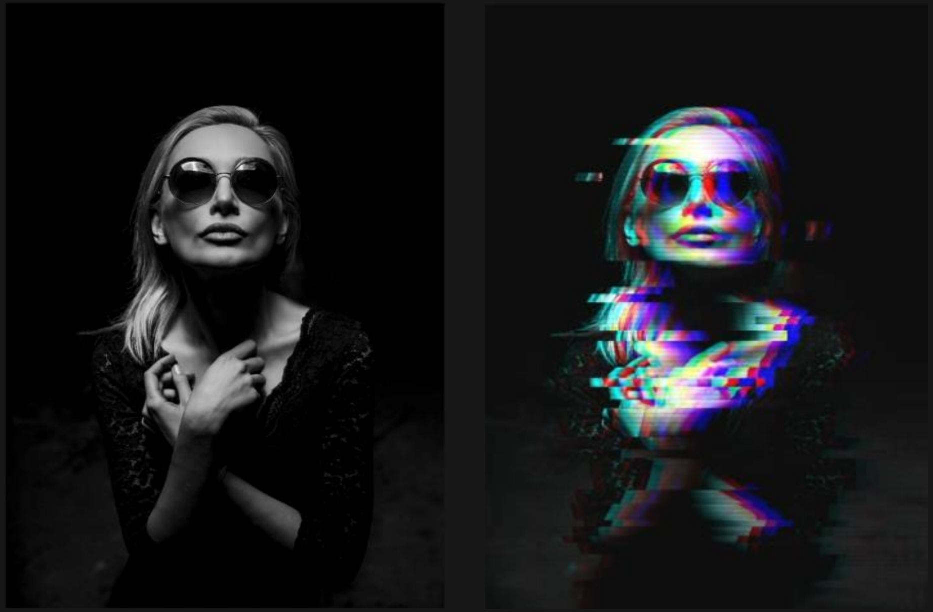
Tasks 2/2

Image manipulation:

- Create an image glitch effect
- Create a random dithering effect
    - For each value in the image, simply generate a random number 1..256; if it is greater than the image value at that point, plot the point white, otherwise plot it black.



https://www.visgraf.impa.br/Courses/ip00/proj/Dithering1/random_dithering.html



https://www.youtube.com/watch?v=57GQ1rS0yU0