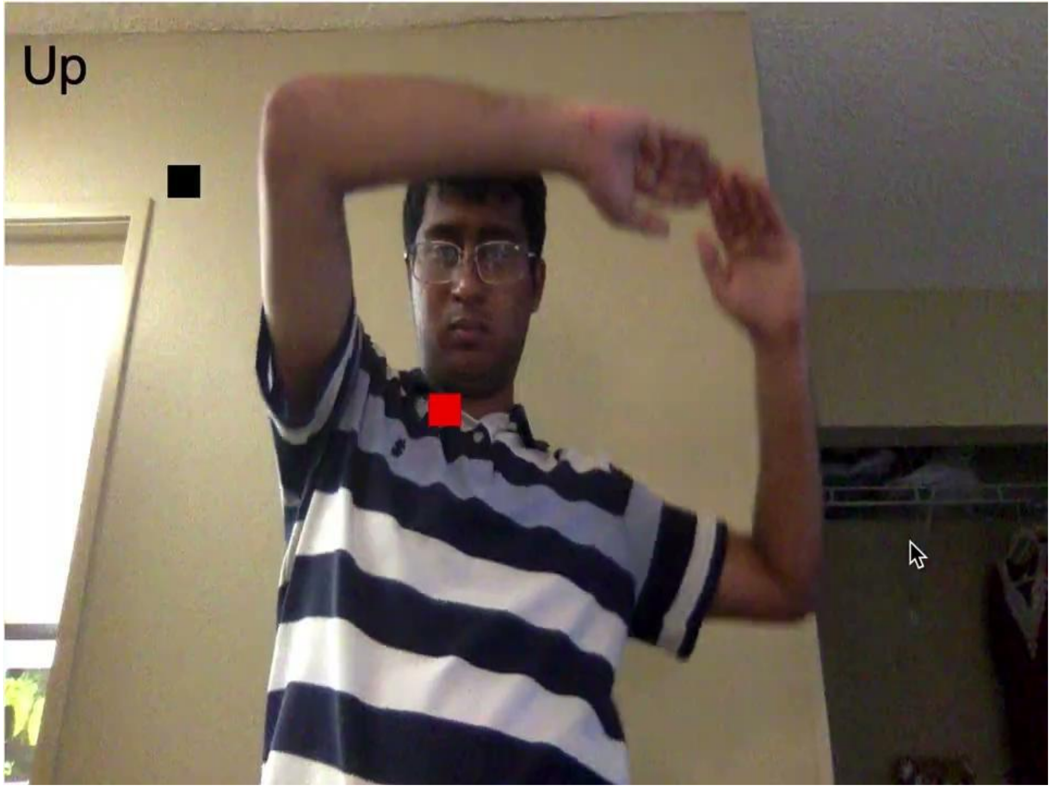
Intro to Computer Vision

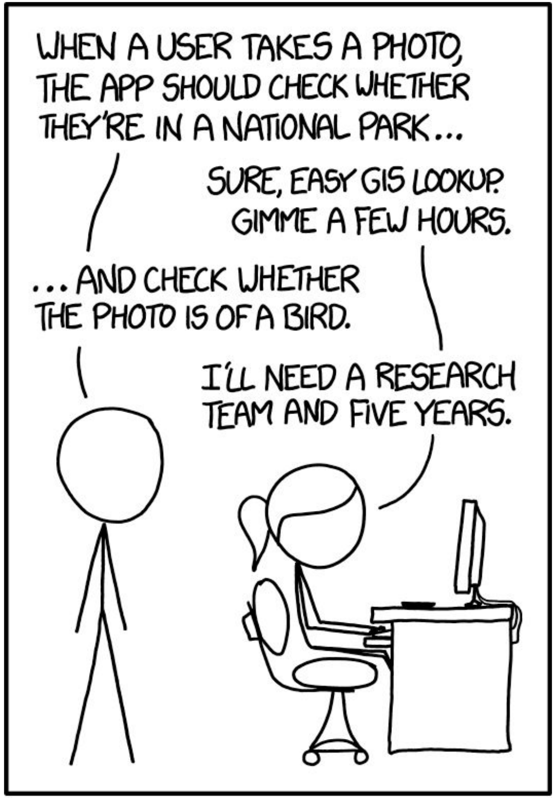
WaffleHacks 2023 Workshop

Agenda

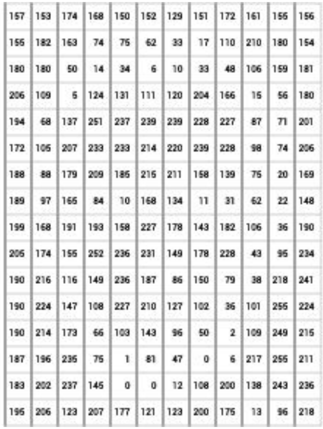
1. Basics of Computer Vision
2. Teachable Machine
3. Snake Game in p5.js
4. Incorporating Pose Estimation model into Snake Game

Final Result:

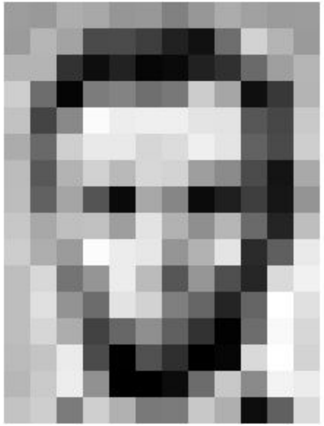


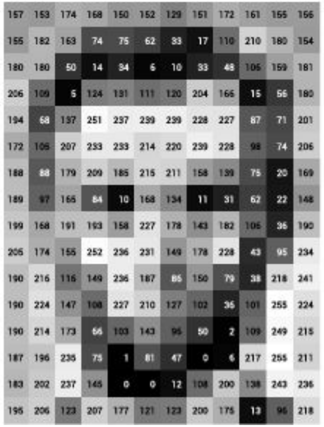

WHEN A USER TAKES A PHOTO, THE APP SHOULD CHECK WHETHER THEY'RE IN A NATIONAL PARK...
SURE, EASY GIS LOOKUP. GIMME A FEW HOURS.
...AND CHECK WHETHER THE PHOTO IS OF A BIRD.
I'LL NEED A RESEARCH TEAM AND FIVE YEARS.

IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218





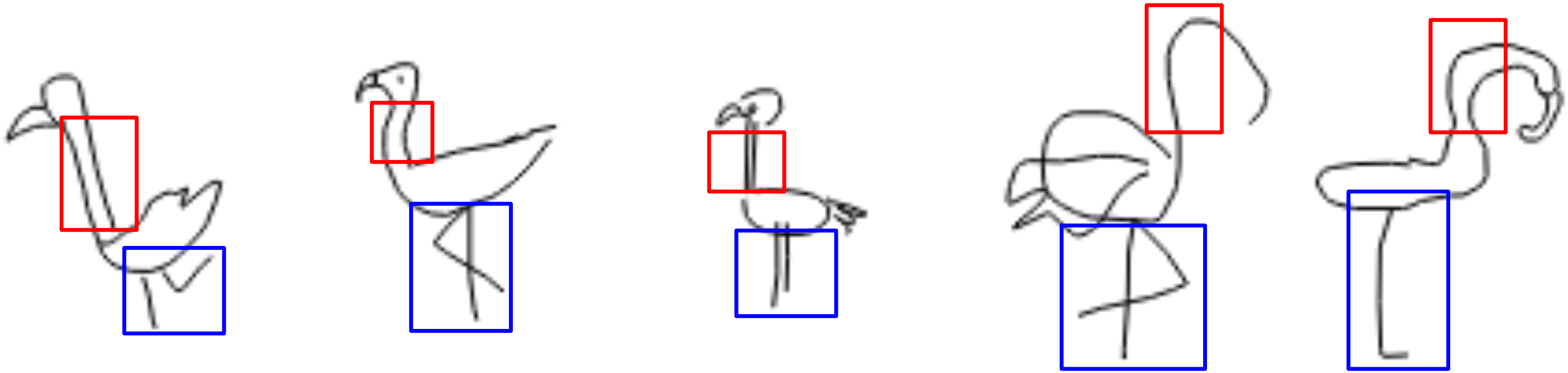
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

What is Computer Vision?

- Computers can’t process images like we can directly
  - Convert an image into a set of numerical values (such as RGB per pixel)
- Algorithms extract features from labeled images and use this to build models
  - Can think of features like patterns or correlations between specific characteristics of an image and its label
- Models can then be used to predict or analyze new content
- Not possible until recently, with large amounts of data and computational power available

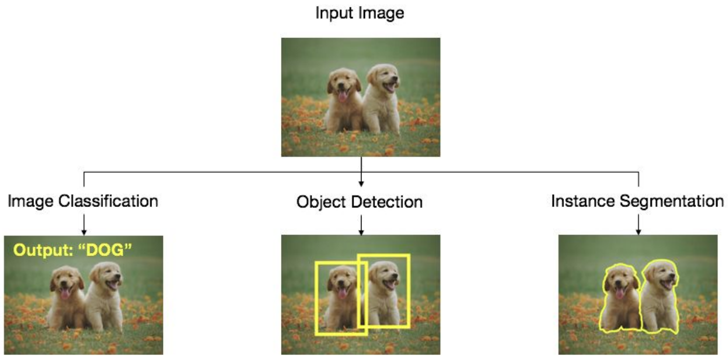
How does it work?

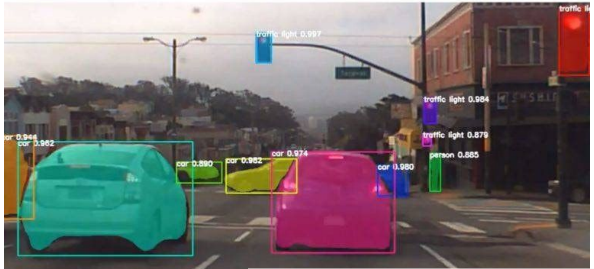
Uncovering Patterns in Data:

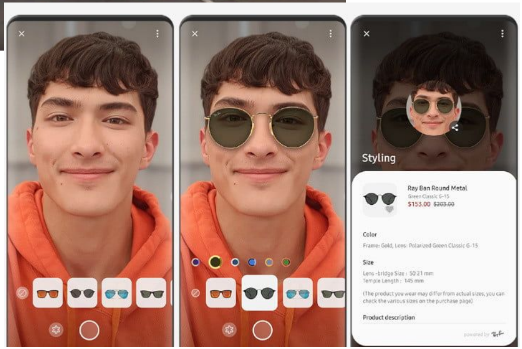


Applications

- Classification
  - Identify category of image
- Identification/Detection
  - Identify location of specific object or all objects in an image
- Segmentation
  - Dividing image into segments, usually based on objects
- Object Tracking
  - Follow the path of one or more objects in a video

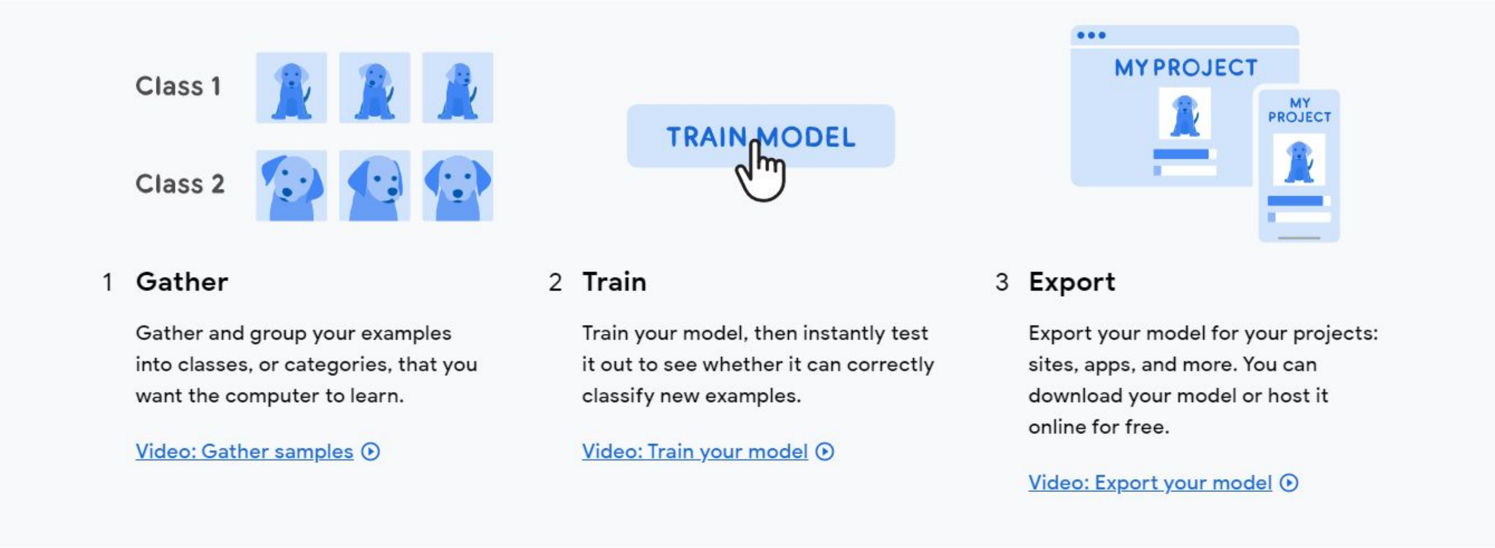






Teachable Machine

- https://teachablemachine.withgoogle.com/
- Platform by Google that allows you to train models without machine learning code



1 **Gather**

Gather and group your examples into classes, or categories, that you want the computer to learn.

Video: Gather samples

2 **Train**

Train your model, then instantly test it out to see whether it can correctly classify new examples.

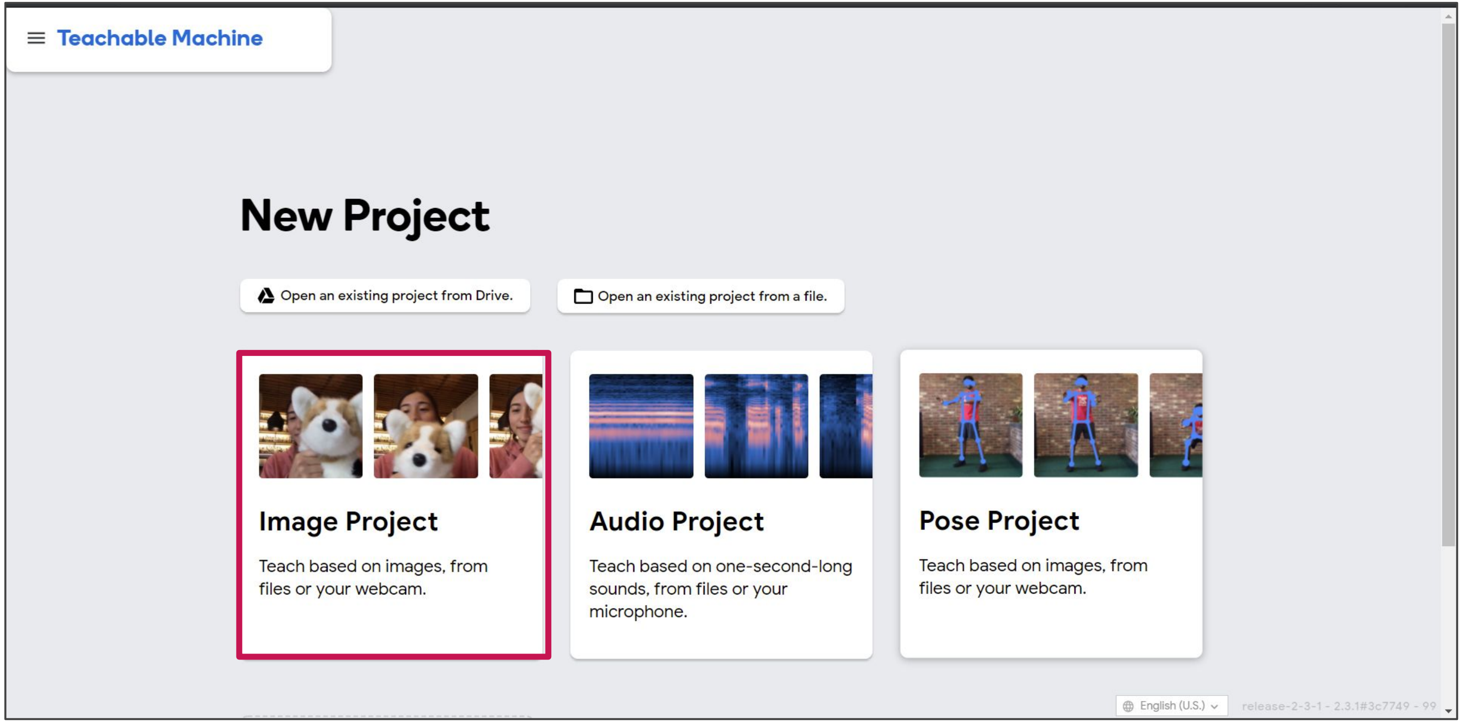
Video: Train your model

3 **Export**

Export your model for your projects: sites, apps, and more. You can download your model or host it online for free.

Video: Export your model

Object Classification



Teachable Machine

New Project

Open an existing project from Drive.

Open an existing project from a file.

Image Project

Teach based on images, from files or your webcam.

Audio Project

Teach based on one-second-long sounds, from files or your microphone.

Pose Project

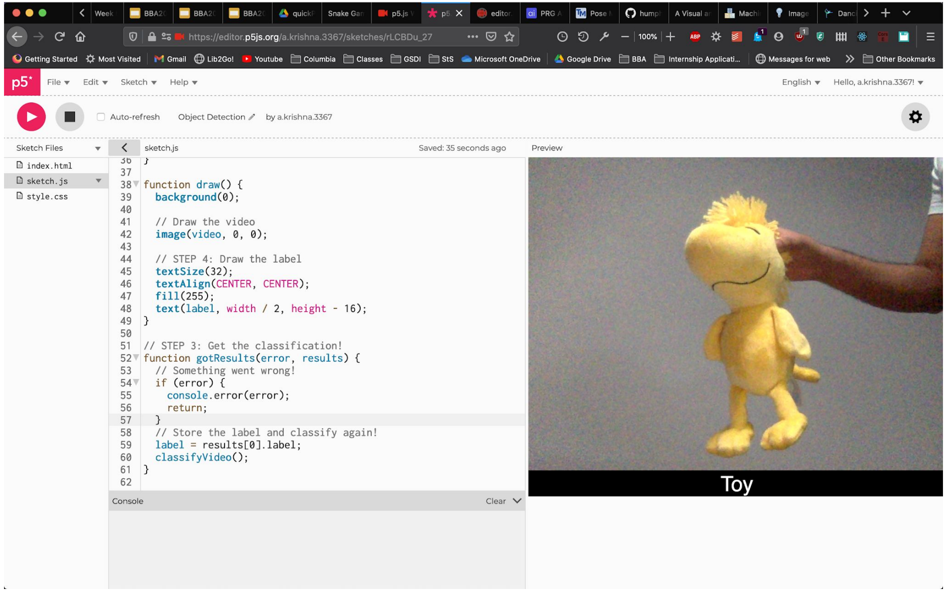
Teach based on images, from files or your webcam.

Training your Model

Under the Hood

Incorporating into p5.js

https://editor.p5js.org/ArjunK_Wafflehacks/sketches/vUER6iN7L

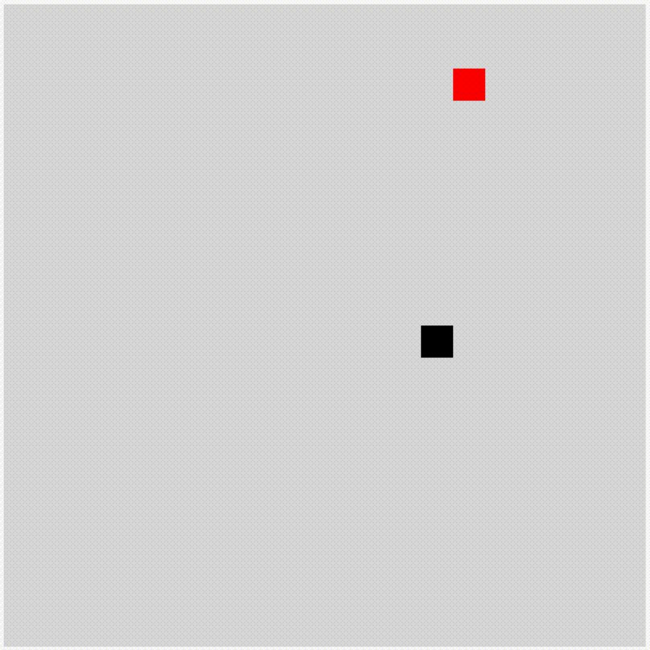


```
}

function draw() {
  background(0);

  // Draw the video
  image(video, 0, 0);

  // STEP 4: Draw the label
  textSize(32);
  textAlign(CENTER, CENTER);
  fill(255);
  text(label, width / 2, height - 16);
}

// STEP 3: Get the classification!
function gotResults(error, results) {
  // Something went wrong!
  if (error) {
    console.error(error);
    return;
  }
  // Store the label and classify again!
  label = results[0].label;
  classifyVideo();
}

```

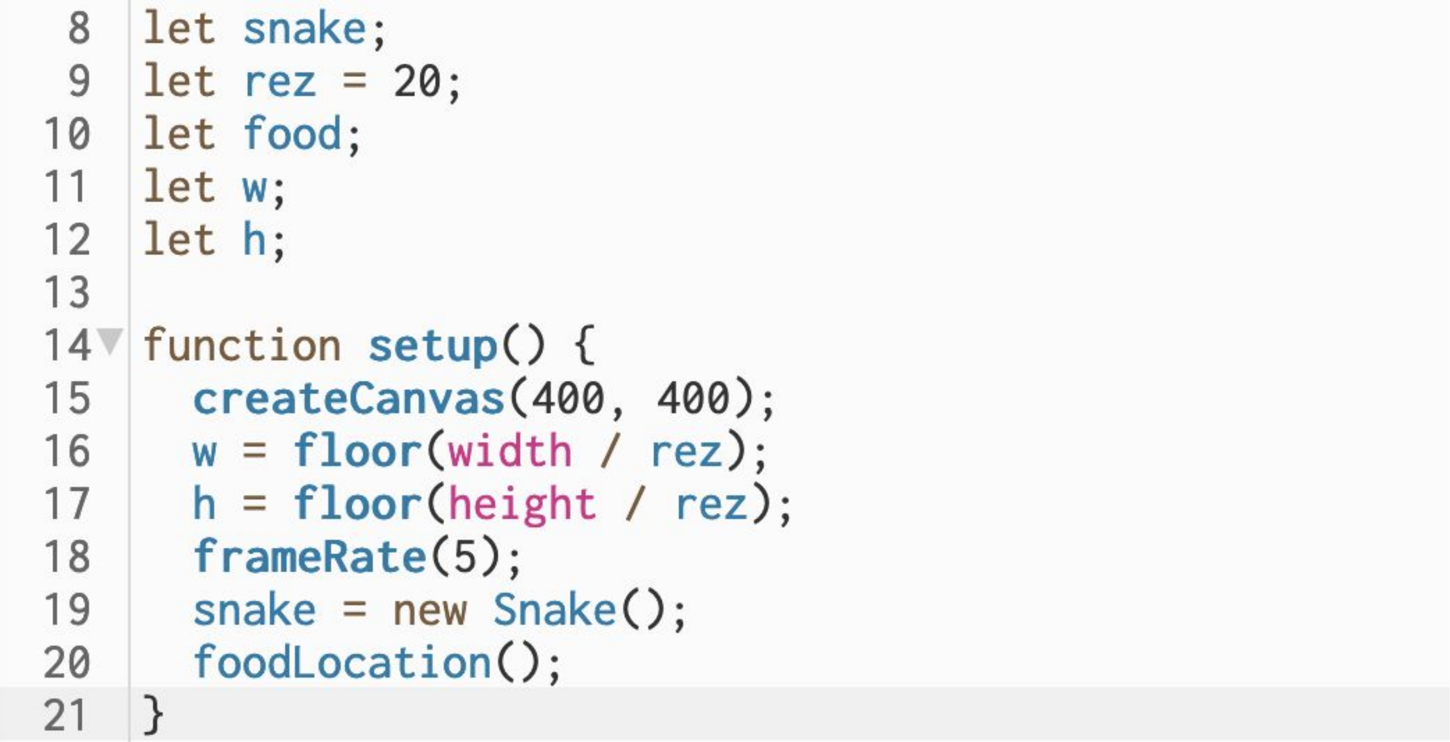
Snake Game

https://editor.p5js.org/ArjunK_Wafflehacks/sketches/UgmVJEqmX

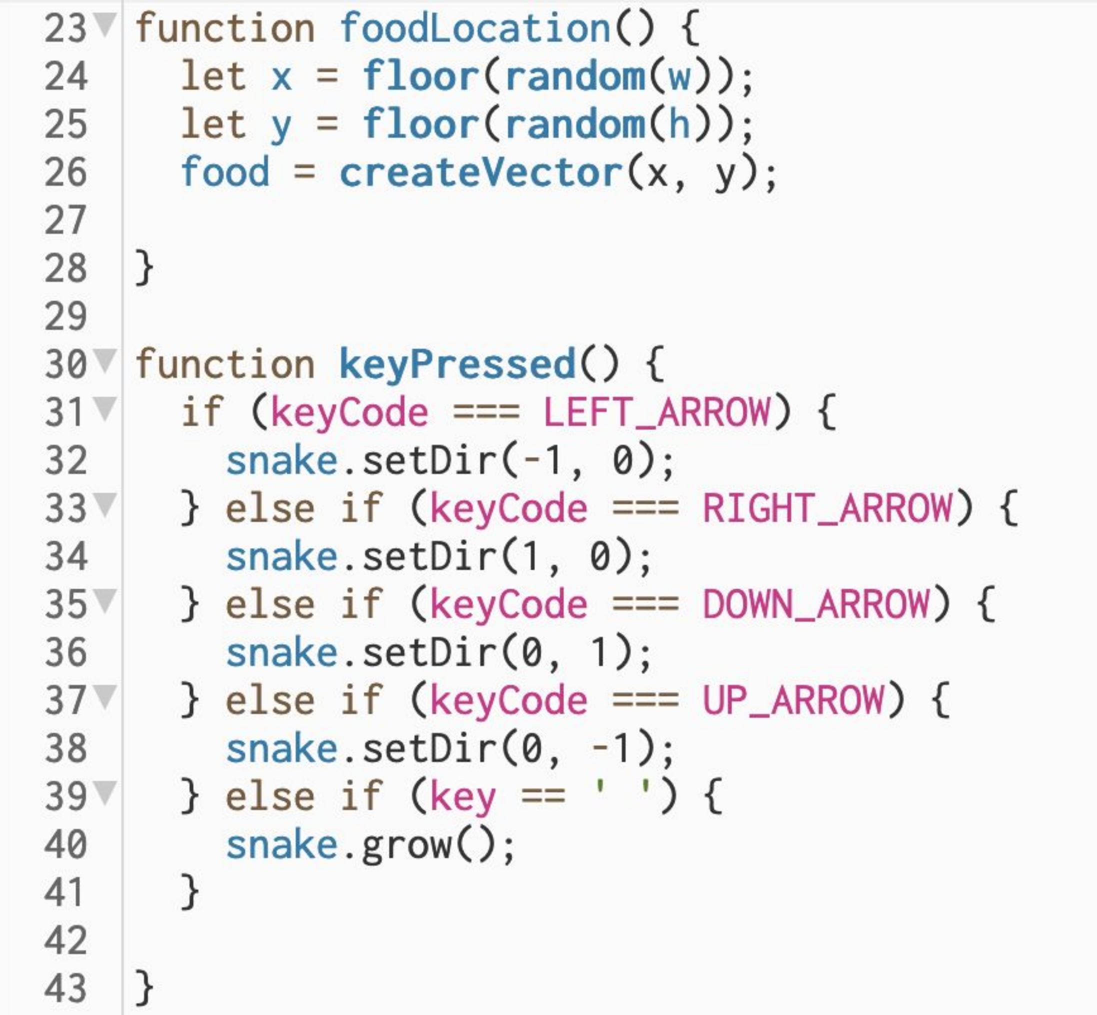
Adapted from Coding Train’s Tutorial



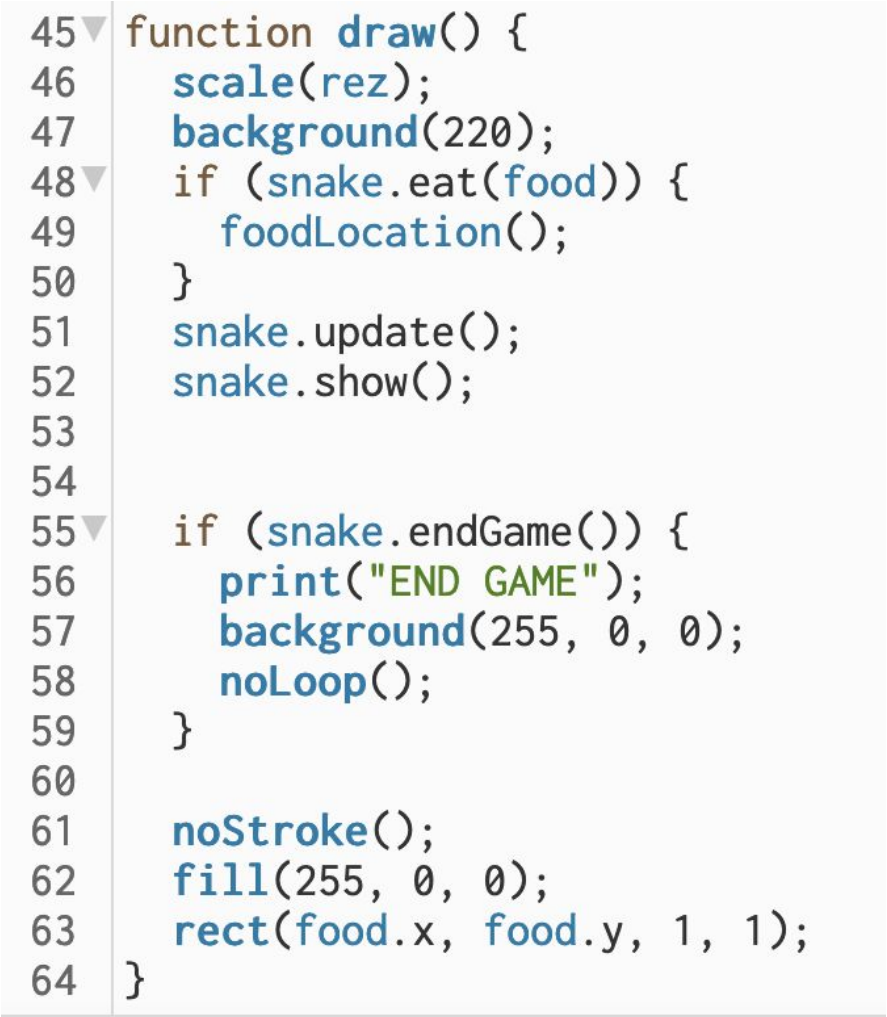
Walking Through Snake Game Code



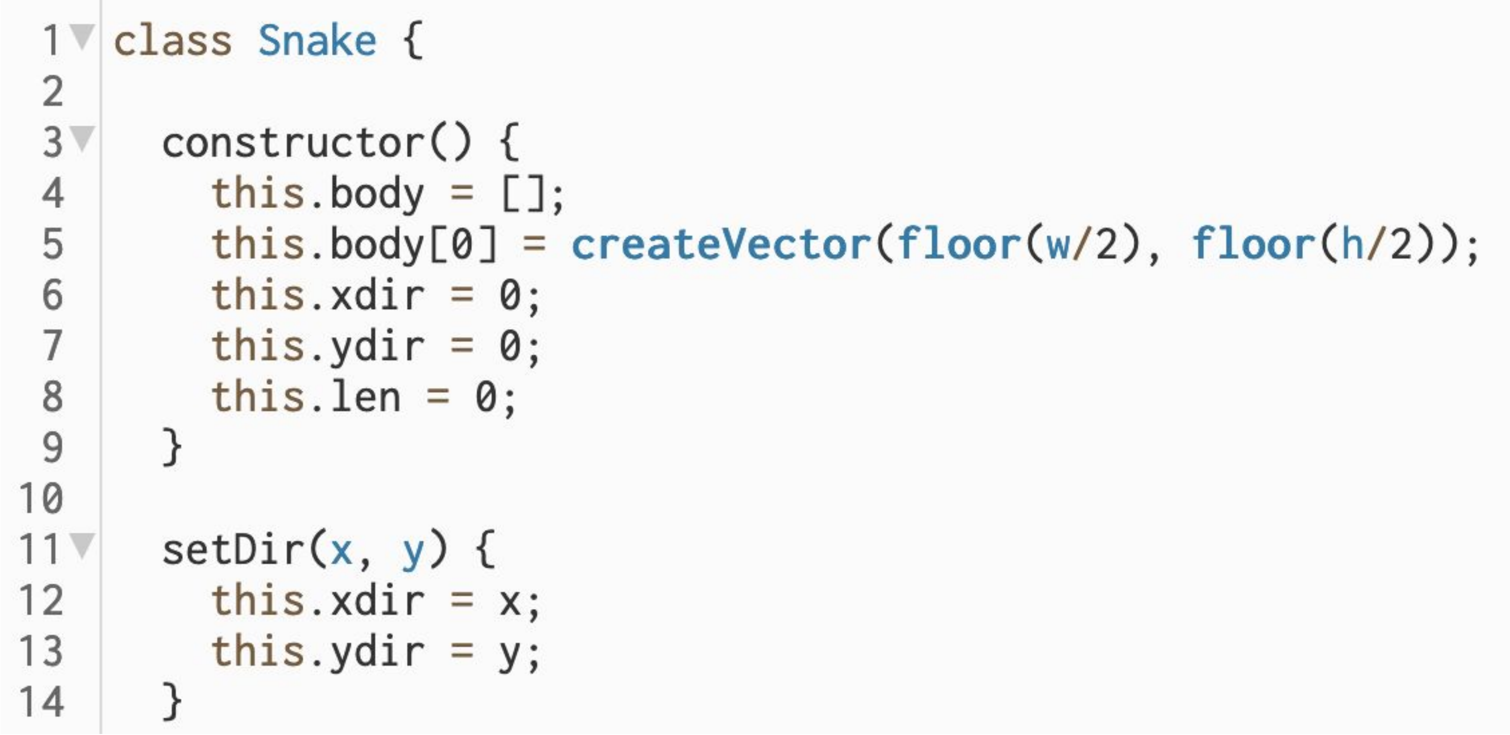
```
let snake;
let rez = 20;
let food;
let w;
let h;

function setup() {
  createCanvas(400, 400);
  w = floor(width / rez);
  h = floor(height / rez);
  frameRate(5);
  snake = new Snake();
  foodLocation();
}
```



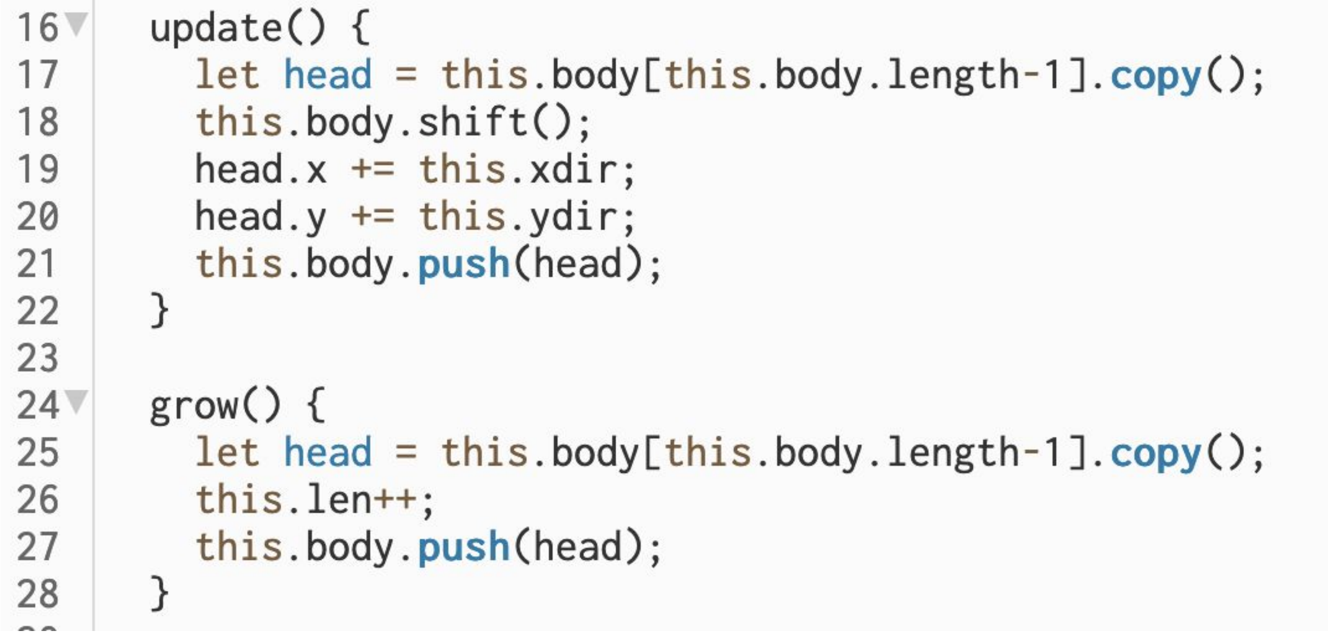
```
function foodLocation() {
  let x = floor(random(w));
  let y = floor(random(h));
  food = createVector(x, y);

}

function keyPressed() {
  if (keyCode === LEFT_ARROW) {
    snake.setDir(-1, 0);
  } else if (keyCode === RIGHT_ARROW) {
    snake.setDir(1, 0);
  } else if (keyCode === DOWN_ARROW) {
    snake.setDir(0, 1);
  } else if (keyCode === UP_ARROW) {
    snake.setDir(0, -1);
  } else if (key == ' ') {
    snake.grow();
  }

}
```



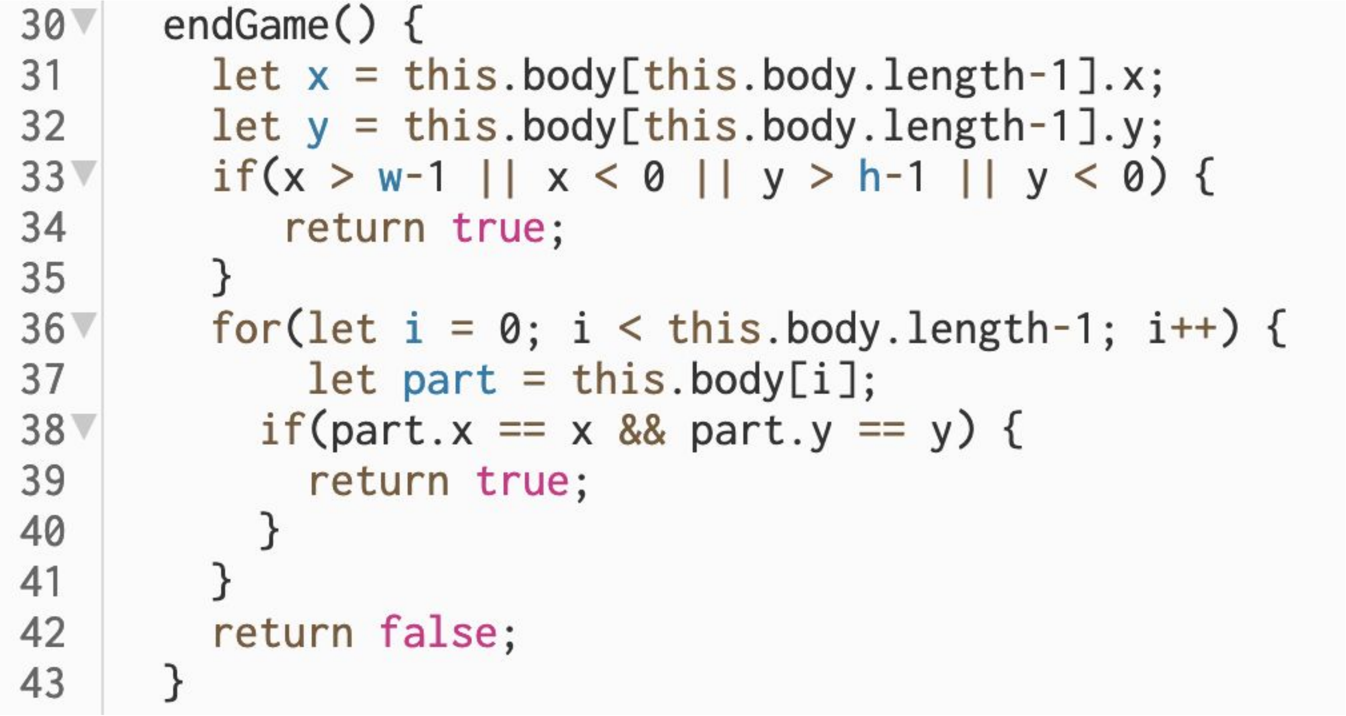
```
function draw() {
  scale(rez);
  background(220);
  if (snake.eat(food)) {
    foodLocation();
  }
  snake.update();
  snake.show();


  if (snake.endGame()) {
    print("END GAME");
    background(255, 0, 0);
    noLoop();
  }

  noStroke();
  fill(255, 0, 0);
  rect(food.x, food.y, 1, 1);
}
```



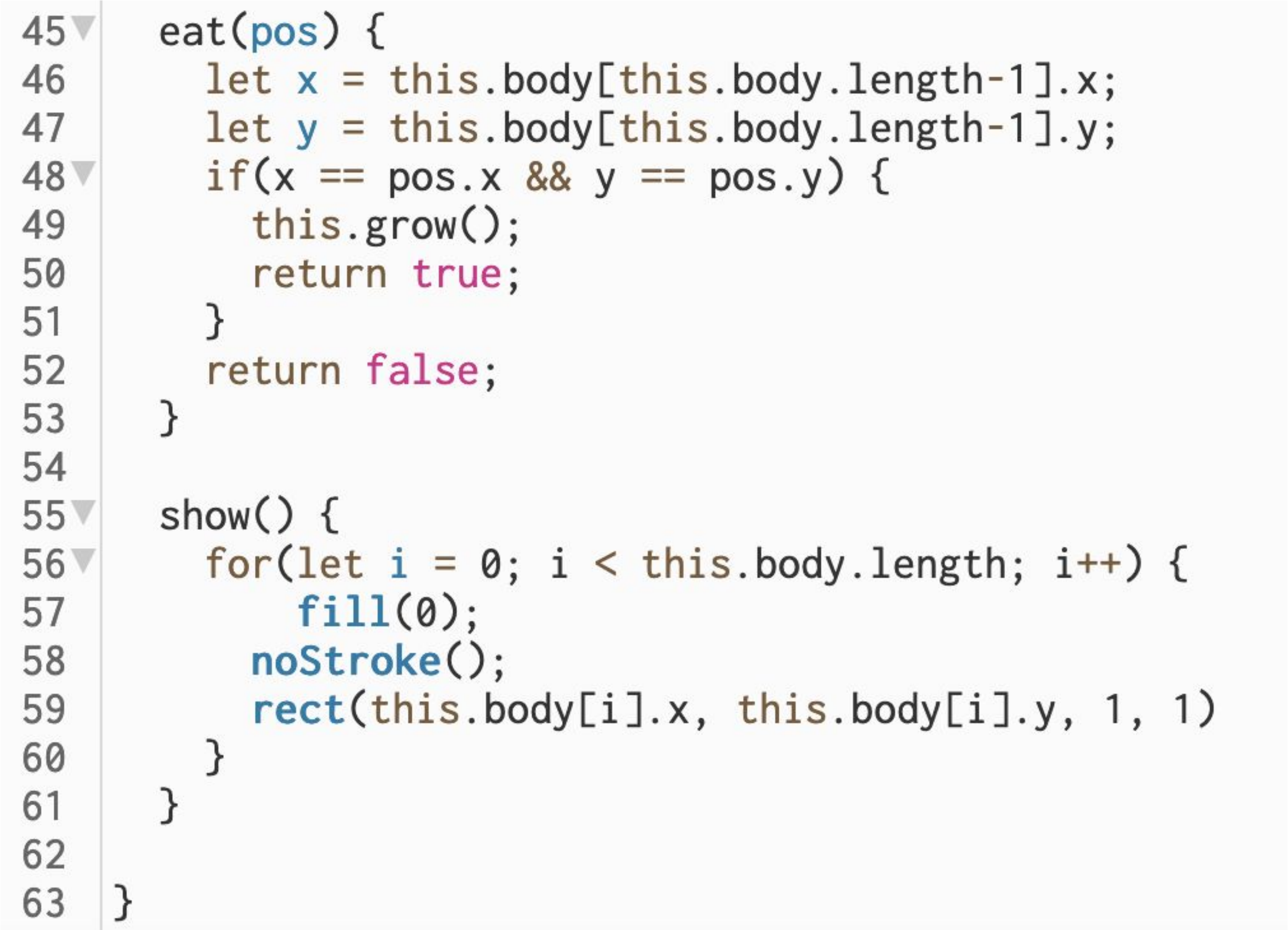
```
class Snake {

  constructor() {
    this.body = [];
    this.body[0] = createVector(floor(w/2), floor(h/2));
    this.xdir = 0;
    this.ydir = 0;
    this.len = 0;
  }

  setDir(x, y) {
    this.xdir = x;
    this.ydir = y;
  }
```



```
  update() {
    let head = this.body[this.body.length-1].copy();
    this.body.shift();
    head.x += this.xdir;
    head.y += this.ydir;
    this.body.push(head);
  }

  grow() {
    let head = this.body[this.body.length-1].copy();
    this.len++;
    this.body.push(head);
  }
```



```javascript
  endGame() {
    let x = this.body[this.body.length-1].x;
    let y = this.body[this.body.length-1].y;
    if(x > w-1 || x < 0 || y > h-1 || y < 0) {
       return true;
    }
    for(let i = 0; i < this.body.length-1; i++) {
        let part = this.body[i];
      if(part.x == x && part.y == y) {
        return true;
      }
    }
    return false;
  }
```



```
  eat(pos) {
    let x = this.body[this.body.length-1].x;
    let y = this.body[this.body.length-1].y;
    if(x == pos.x && y == pos.y) {
      this.grow();
      return true;
    }
    return false;
  }

  show() {
    for(let i = 0; i < this.body.length; i++) {
        fill(0);
      noStroke();
      rect(this.body[i].x, this.body[i].y, 1, 1)
    }
  }

}
```

Adding Computer Vision!

Instead of controlling the snake with the arrow keys, we will incorporate an object detection model to direct the snake based on our pose.

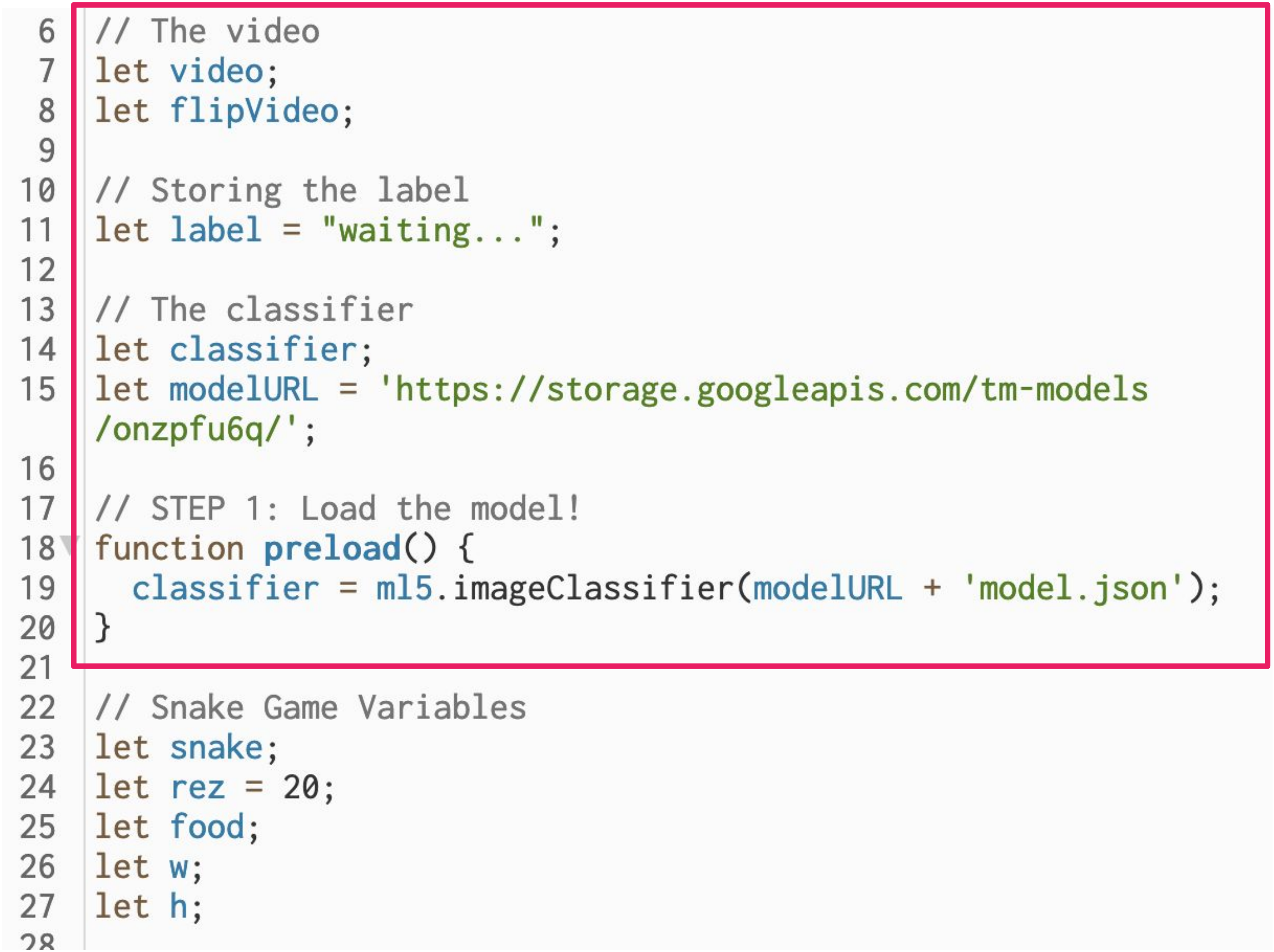
Need poses for 5 directions - up, down, left, and right, and neutral.

Creating New Model

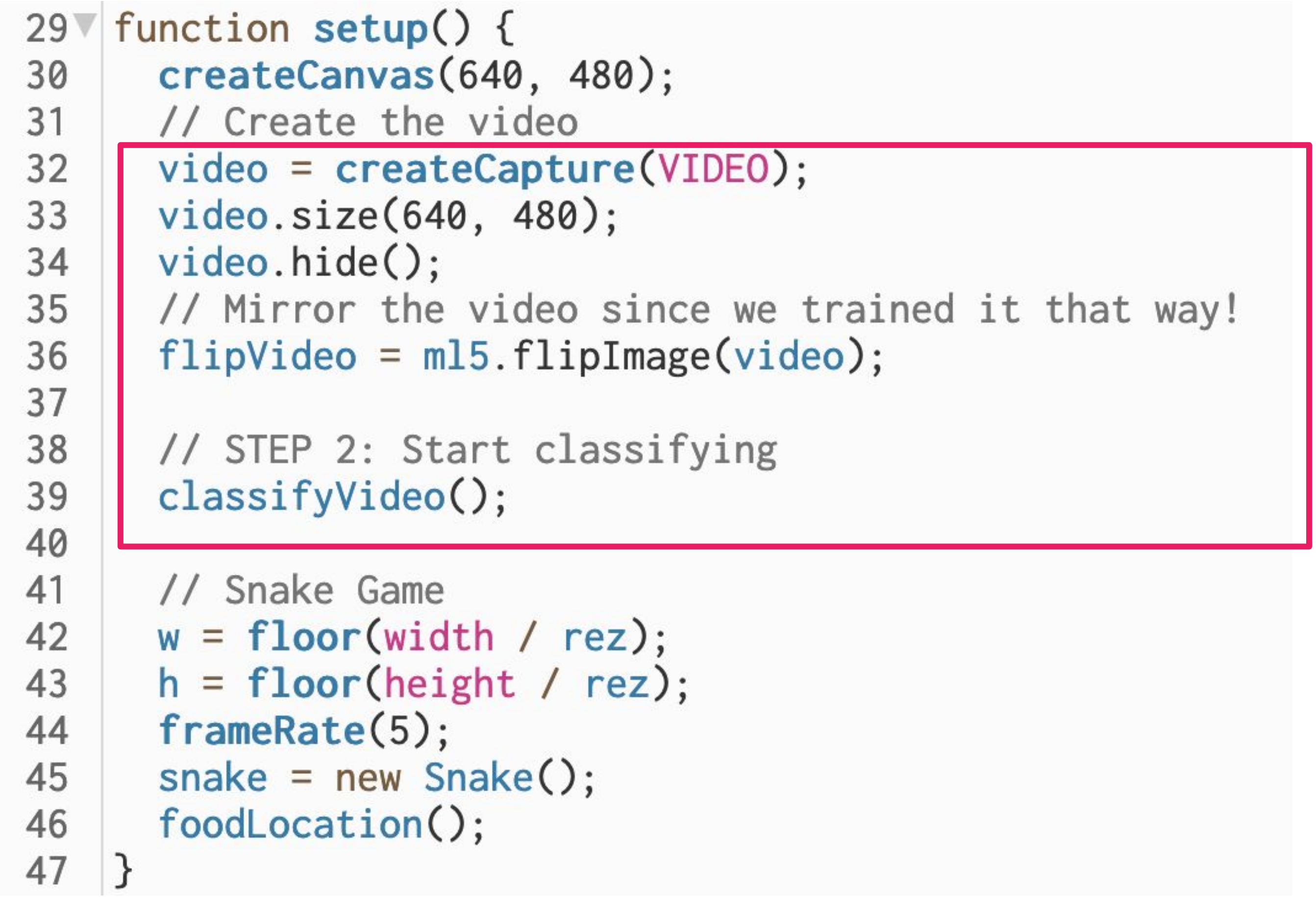
Controlling the Snake

4 steps to modify the current Snake Game:

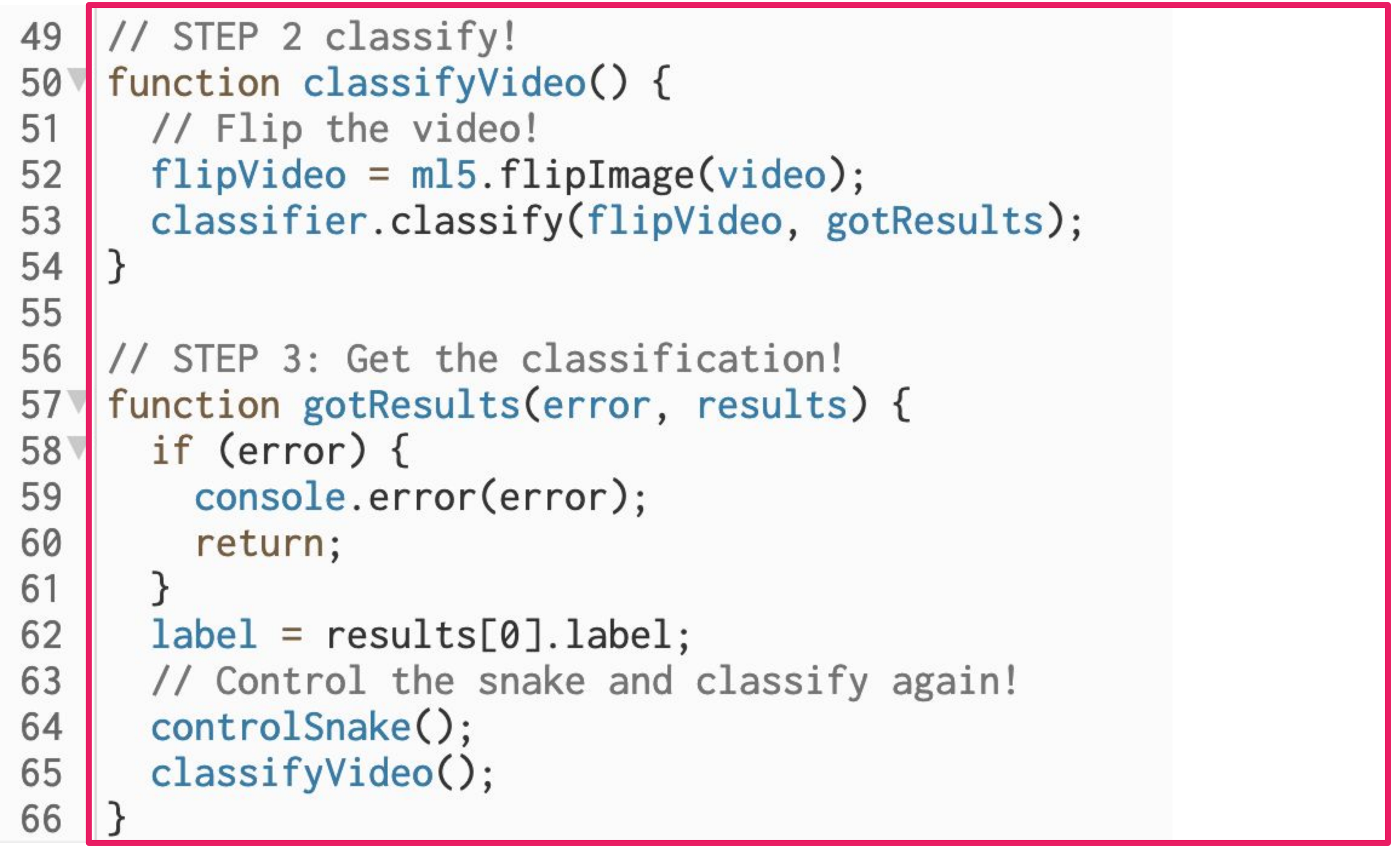
1. Import Model
2. Get video feed from player
3. Send image to model to classify user action as a direction command
4. Use predicted label to move snake



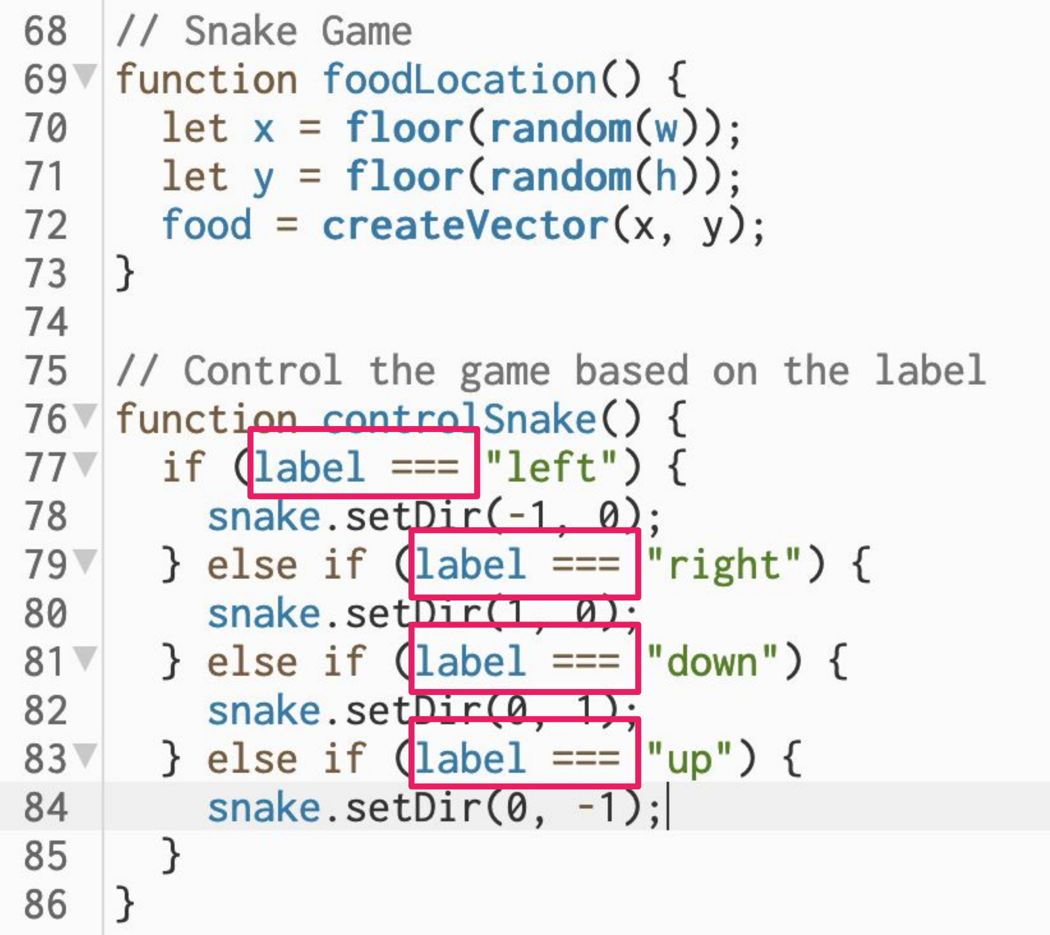
```
// The video
let video;
let flipVideo;

// Storing the label
let label = "waiting...";

// The classifier
let classifier;
let modelURL = 'https://storage.googleapis.com/tm-models/onzpfu6q/';

// STEP 1: Load the model!
function preload() {
  classifier = ml5.imageClassifier(modelURL + 'model.json');
}

// Snake Game Variables
let snake;
let rez = 20;
let food;
let w;
let h;
```



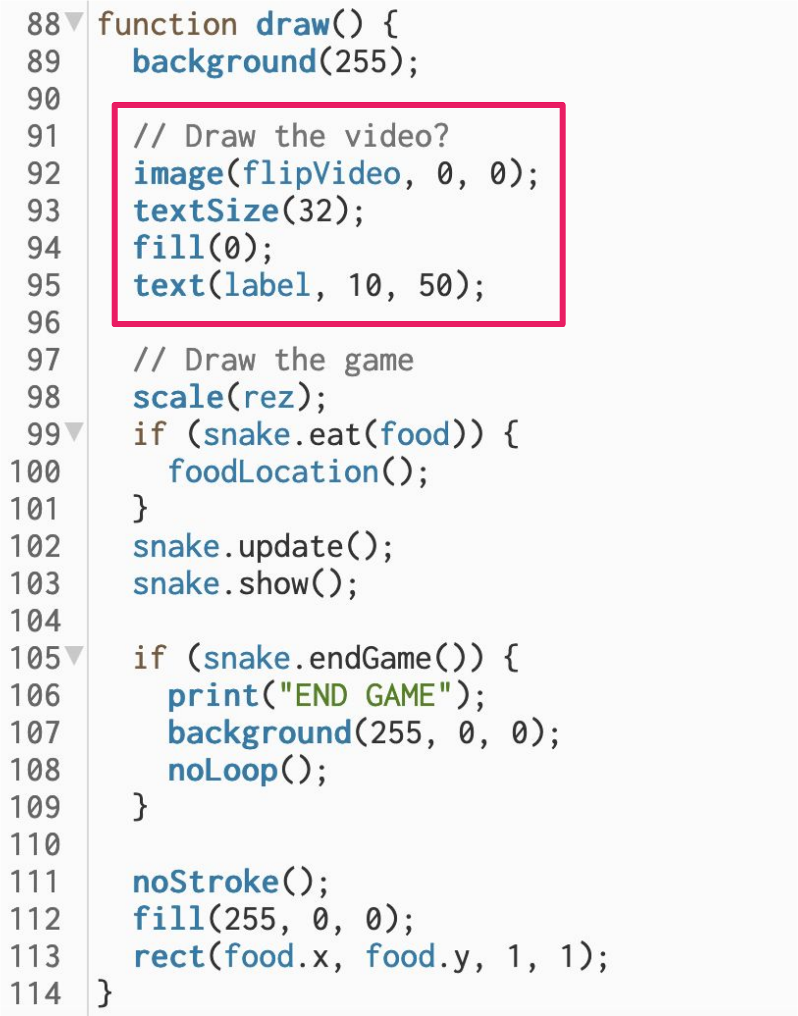
```
function setup() {
  createCanvas(640, 480);
  // Create the video
  video = createCapture(VIDEO);
  video.size(640, 480);
  video.hide();
  // Mirror the video since we trained it that way!
  flipVideo = ml5.flipImage(video);

  // STEP 2: Start classifying
  classifyVideo();

  // Snake Game
  w = floor(width / rez);
  h = floor(height / rez);
  frameRate(5);
  snake = new Snake();
  foodLocation();
}
```



```
// STEP 2 classify!
function classifyVideo() {
  // Flip the video!
  flipVideo = ml5.flipImage(video);
  classifier.classify(flipVideo, gotResults);
}

// STEP 3: Get the classification!
function gotResults(error, results) {
  if (error) {
    console.error(error);
    return;
  }
  label = results[0].label;
  // Control the snake and classify again!
  controlSnake();
  classifyVideo();
}
```



```javascript
// Snake Game
function foodLocation() {
  let x = floor(random(w));
  let y = floor(random(h));
  food = createVector(x, y);
}

// Control the game based on the label
function controlSnake() {
  if (label === "left") {
    snake.setDir(-1, 0);
  } else if (label === "right") {
    snake.setDir(1, 0);
  } else if (label === "down") {
    snake.setDir(0, 1);
  } else if (label === "up") {
    snake.setDir(0, -1);
  }
}
```



```
function draw() {
  background(255);

  // Draw the video?
  image(flipVideo, 0, 0);
  textSize(32);
  fill(0);
  text(label, 10, 50);

  // Draw the game
  scale(rez);
  if (snake.eat(food)) {
    foodLocation();
  }
  snake.update();
  snake.show();

  if (snake.endGame()) {
    print("END GAME");
    background(255, 0, 0);
    noLoop();
  }

  noStroke();
  fill(255, 0, 0);
  rect(food.x, food.y, 1, 1);
}
```

More Resources

More experiments with Teachable Machine:
https://experiments.withgoogle.com/search?tag=Teachable+Machine

Teachable Machine Code:
https://github.com/googlecreativelab/teachablemachine-community/

Intro to CV Course:
https://www.coursera.org/learn/introduction-computer-vision-watson-opencv

Final Code

https://editor.p5js.org/ArjunK_Wafflehacks/sketches/CsXo7MPo2