Welcome to

Digital Senses 數位通感

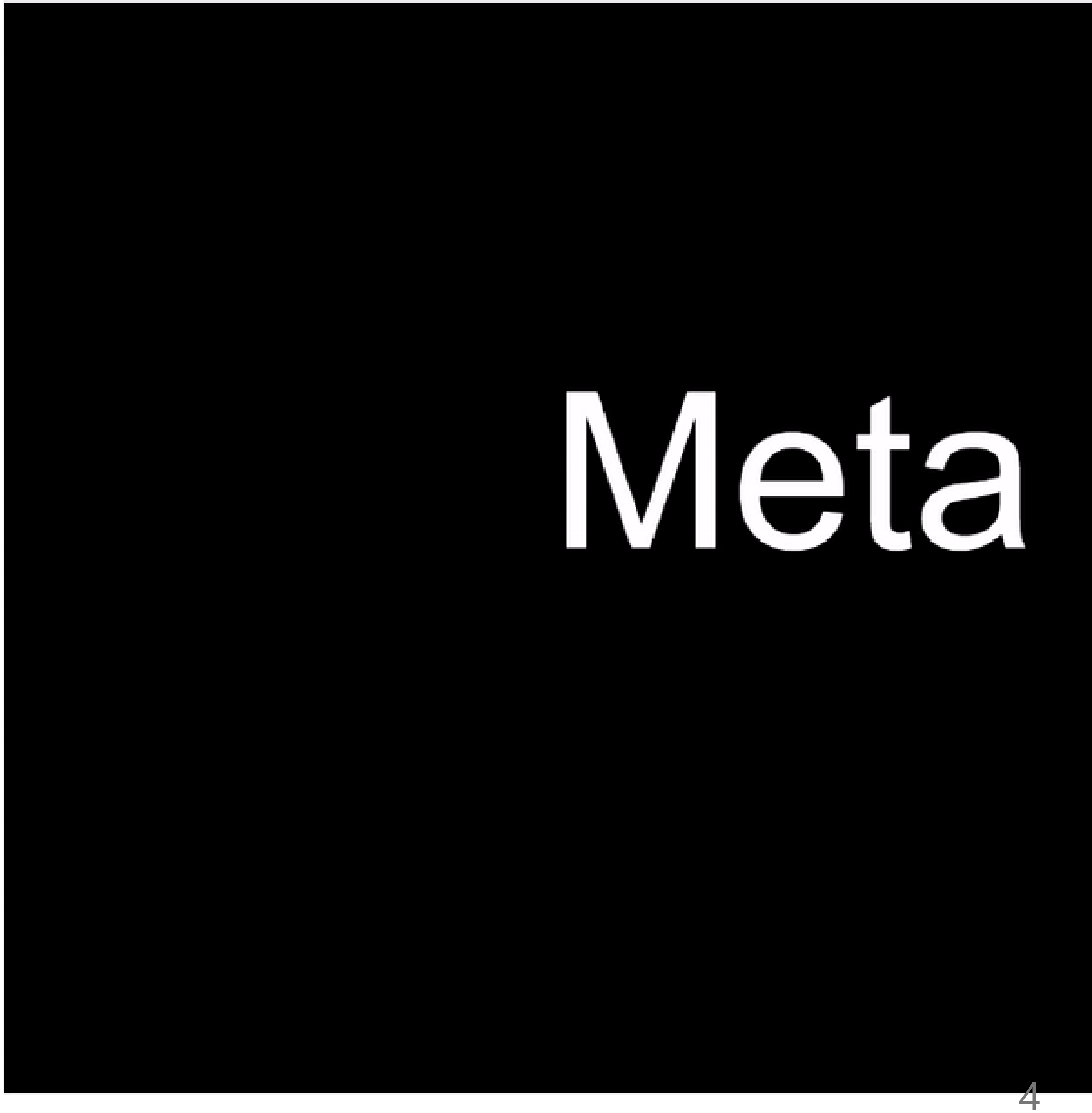
四) 互動：創作與聲音互動的攝影軟件

如何在 p5.js 使用鍵盤互動?

答案 : 使用 系統變量 key 得到當前 按下的 鍵盤按鍵數值

4.1 useKey

```
function setup() {
  createCanvas(500,500);
}

function draw() {
  background(0);
  fill(255);
  textSize(100);
  text(key, width/2, height/2);
}
```



Meta

使用 text(); 在顯示框内顯示文字資訊

不明白text( );内的數值有甚麼意義?

text(文字字串, x座標 , y座標);

試一下 使用鍵盤與單聲道合成器互動

4.2 checkKey

```
let monoSynth;

function setup() {
  createCanvas(500, 500);
  background(0);
  monoSynth = new p5.MonoSynth();
}

function keyPressed() {
  if (key == "a") {
    userStartAudio();
    monoSynth.play("C4", 1, 0, 0.25);
  }
  if (key == "s") {
    userStartAudio();
    monoSynth.play("F4", 1, 0, 0.25);
  }
}
```

function keyPressed() { }部份

在 function keyPressed() 內的編碼

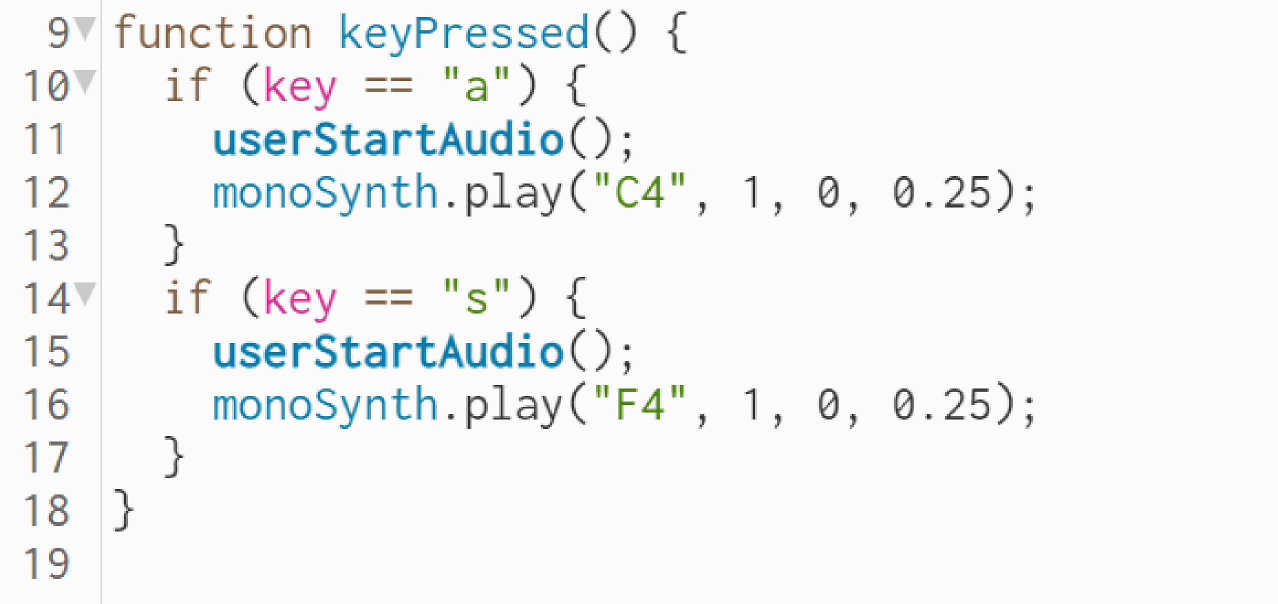
會在 按鍵 被按下後再執行

在這例子中

如果 if () 按鍵 "a" 被按下

if(key == "a")

monoSynth **就會播放音符** C4

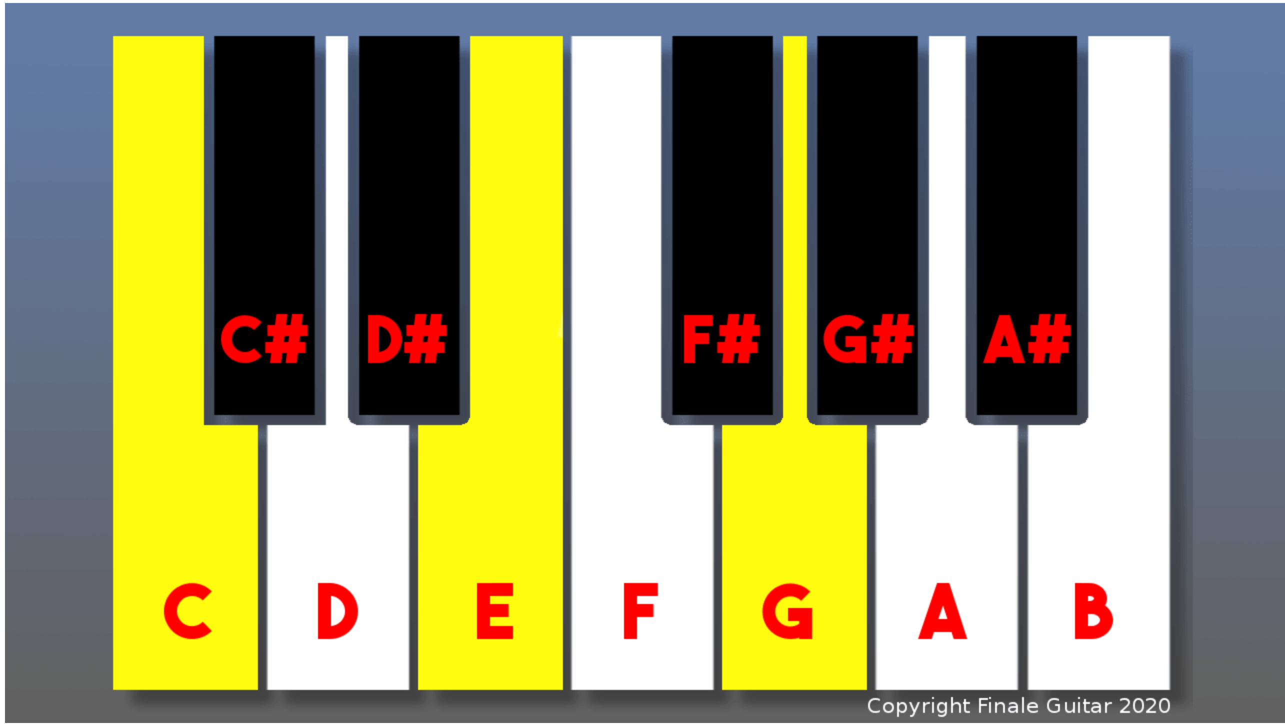


```
function keyPressed() {
  if (key == "a") {
    userStartAudio();
    monoSynth.play("C4", 1, 0, 0.25);
  }
  if (key == "s") {
    userStartAudio();
    monoSynth.play("F4", 1, 0, 0.25);
  }
}
```

尋求進階編碼的同學可以打開例子

4.3 advanceCheckKey

內裡使用 switch 條件式 增加了 按鍵數量至 一個八度的琴鍵



Copyright Finale Guitar 2020

4.3 advanceCheckKey

```
let monoSynth;

function setup() {
  createCanvas(500, 500);
  background(0);
  monoSynth = new p5.MonoSynth();
}

function keyPressed() {
  switch (key) {
    case "a":
      userStartAudio();
      monoSynth.play(midiToFreq(60), 1, 0, 0.25);
      break;
    case "w":
      userStartAudio();
      monoSynth.play(midiToFreq(61), 1, 0, 0.25);
      break;
    case "s":
      userStartAudio();
      monoSynth.play(midiToFreq(62), 1, 0, 0.25);
      break;
    case "e":
      userStartAudio();
      monoSynth.play(midiToFreq(63), 1, 0, 0.25);
      break;
    case "d":
      userStartAudio();
      monoSynth.play(midiToFreq(64), 1, 0, 0.25);
      break;
    case "f":
      userStartAudio();
      monoSynth.play(midiToFreq(65), 1, 0, 0.25);
      break;
    case "t":
      userStartAudio();
      monoSynth.play(midiToFreq(66), 1, 0, 0.25);
      break;
    case "g":
      userStartAudio();
      monoSynth.play(midiToFreq(67), 1, 0, 0.25);
      break;
    case "y":
      userStartAudio();
      monoSynth.play(midiToFreq(68), 1, 0, 0.25);
      break;
    case "h":
      userStartAudio();
      monoSynth.play(midiToFreq(69), 1, 0, 0.25);
      break;
    case "u":
      userStartAudio();
      monoSynth.play(midiToFreq(70), 1, 0, 0.25);
      break;
    case "j":
      userStartAudio();
      monoSynth.play(midiToFreq(71), 1, 0, 0.25);
      break;
  }
}
```

使用 midiToFreq();

把 音符的 MIDI 數值轉為 電腦能理解的 Frequency 聲音頻率數值

不明白 midiToFreq(); 内的數值有甚麼意義?

midiToFreq(MIDI音符數值);

MIDI音符數值

https://www.inspiredacoustics.com/en/MIDI_note_numbers_and_center_frequencies

如果想同時按下多個音鍵

就要使用 Poly Synthesizer 複音合成器

打開例子 4.4 polySynth

4.4 polySynth (poly synthesizer 複音合成器)

```
let polySynth;

function setup() {
  createCanvas(500, 500);
  background(0);
  polySynth = new p5.PolySynth();
}

function keyPressed() {
  switch (key) {
    case "a":
      userStartAudio();
      polySynth.play(midiToFreq(60), 1, 0, 0.25);
      break;
    case "w":
      userStartAudio();
      polySynth.play(midiToFreq(61), 1, 0, 0.25);
      break;

...
```

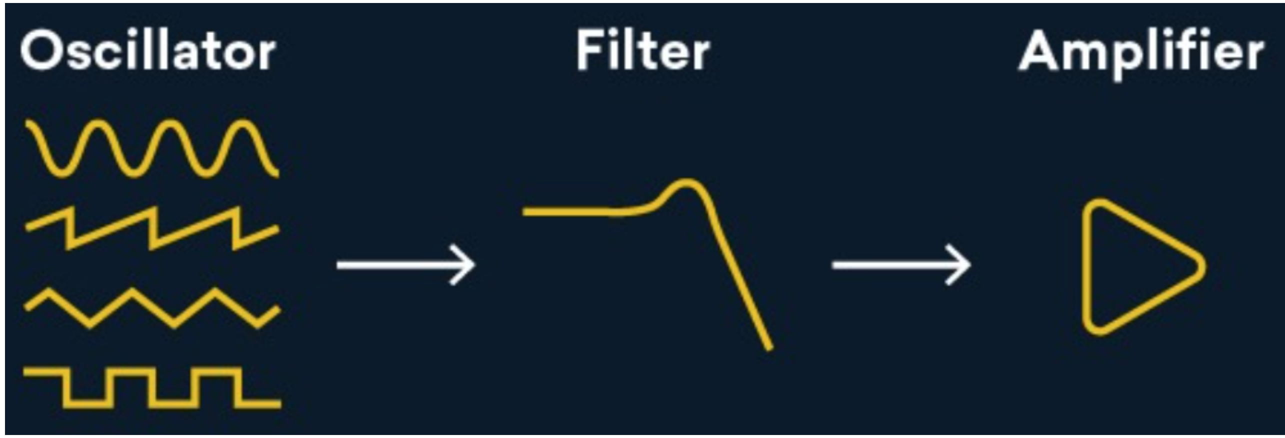
在上一節課提到

聲音是振動產生的聲波

而 Oscillator (or VCOs) **就是 Synth 合成器的基本組件**

打開例子 4.5 simpleOscillator

學習在 p5.js 使用 p5.Oscillator();



4.5 simpleOscillator (oscillator 振盪器)

```
let osc;

function setup() {
  createCanvas(500, 500);
  background(0);
  osc = new p5.Oscillator();
  osc.freq(220);
  osc.setType("sine");
  osc.amp(0.5);
  osc.start();
}
```

先宣告 osc 變量，在 setup() 中

把 osc 定義為 新的(new) p5.Oscillator();

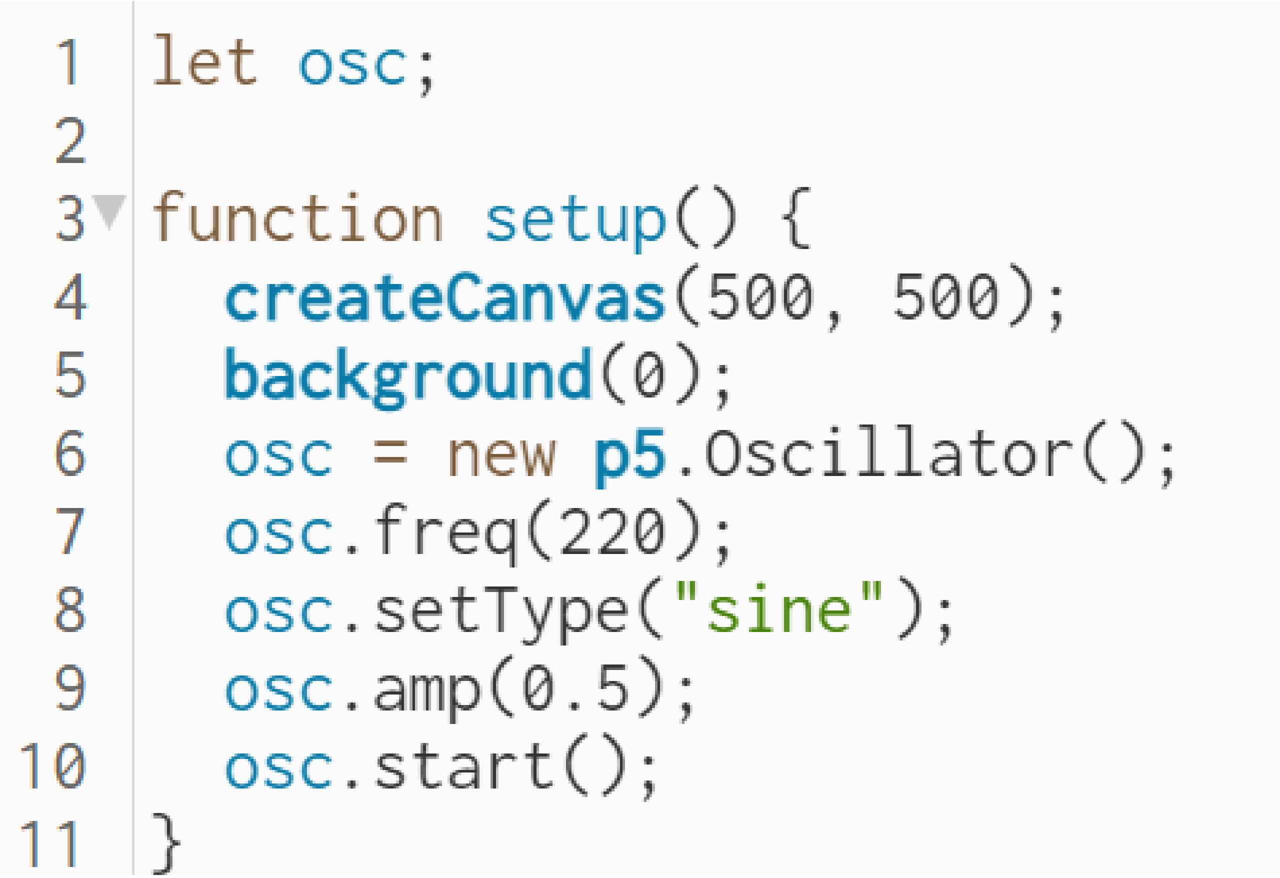
osc.freq(聲音頻率)

osc.setType(振盪器 發生 聲音的方法)

'sine' (default), 'triangle', 'sawtooth', 'square'

osc.amp(音頻功率放大數值) 最小為0 最大為1

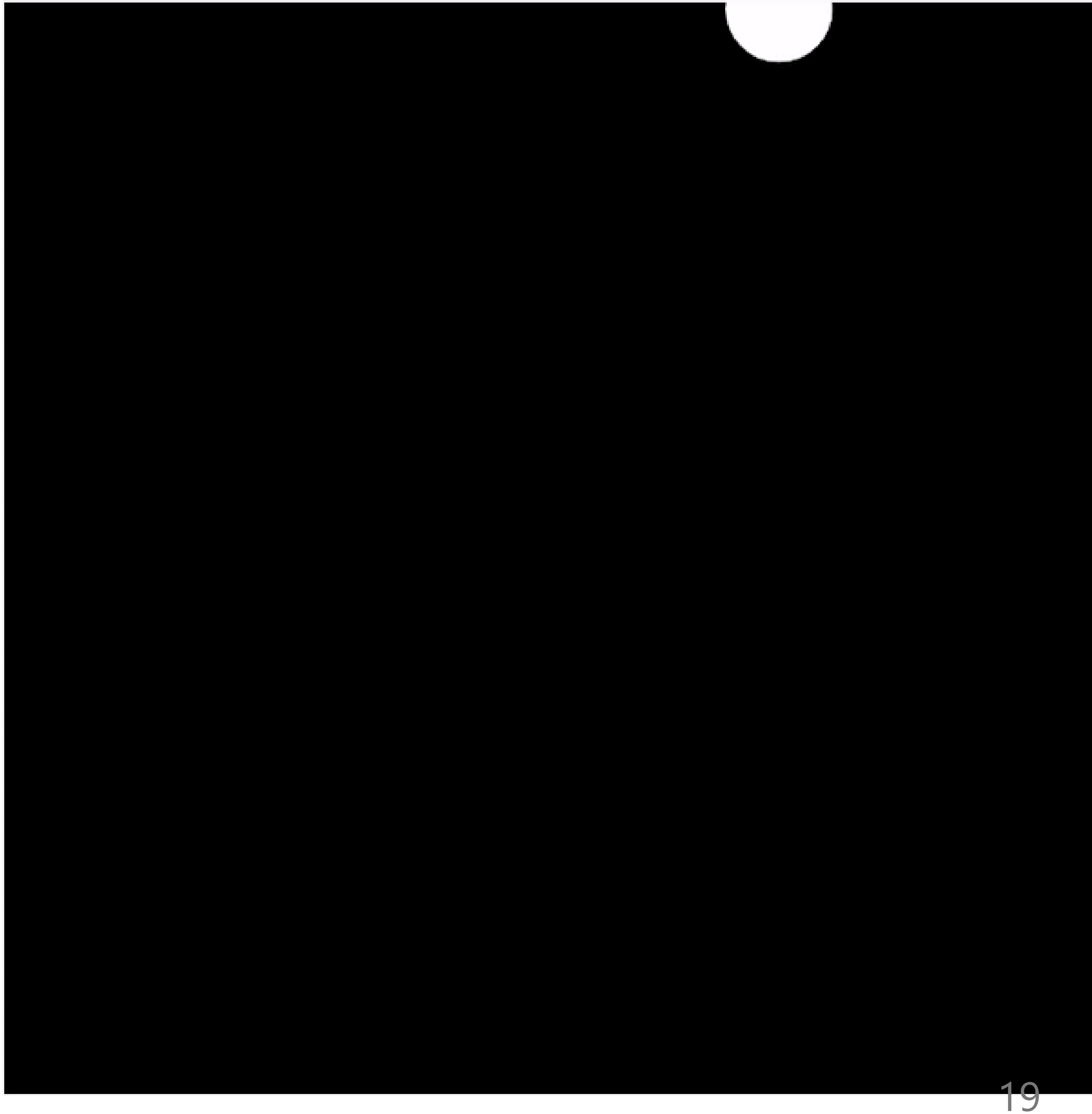
osc.start() 開始使用 p5.Oscillator(); 播放聲音



```
let osc;

function setup() {
  createCanvas(500, 500);
  background(0);
  osc = new p5.Oscillator();
  osc.freq(220);
  osc.setType("sine");
  osc.amp(0.5);
  osc.start();
}
```

4.6 advanceOscillator

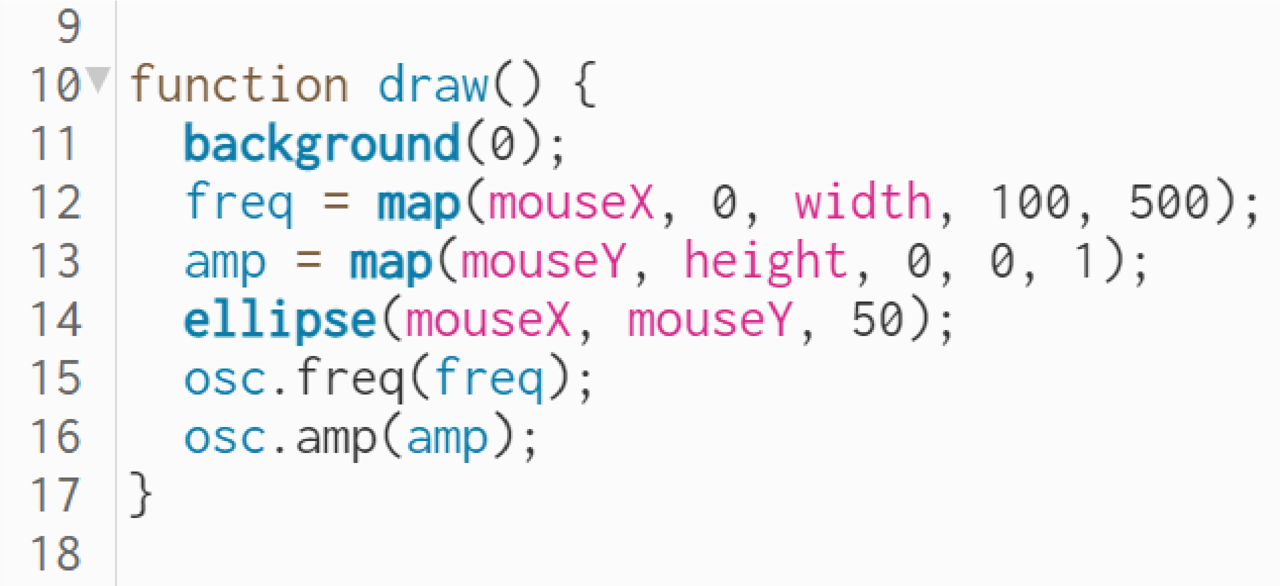
```
let osc, amp, freq;

function setup() {
  createCanvas(500, 500);
  osc = new p5.Oscillator();
  osc.setType("sine");
  osc.start();
}

function draw() {
  background(0);
  freq = map(mouseX, 0, width, 100, 500);
  amp = map(mouseY, height, 0, 0, 1);
  ellipse(mouseX, mouseY, 50);
  osc.freq(freq);
  osc.amp(amp);
}
```



在 draw() 之內 定義變量 freq 及 amp

令 聲音頻率 及 聲音功率放大數值 與 系統變量 mouseX 及 mouseY 互動

但我們需要使用 map() 來把數值 從一個範圍 re-map 重新映射 到另一個範圍



```

function draw() {
  background(0);
  freq = map(mouseX, 0, width, 100, 500);
  amp = map(mouseY, height, 0, 0, 1);
  ellipse(mouseX, mouseY, 50);
  osc.freq(freq);
  osc.amp(amp);
}

```

那我們要學一下 map() 這個函數了

map(value 目標修改數值, start1 數值原來的最小值, stop1 數值原來的最大值, start2 重新映射後的最小值, stop2 重新映射後的最大值);

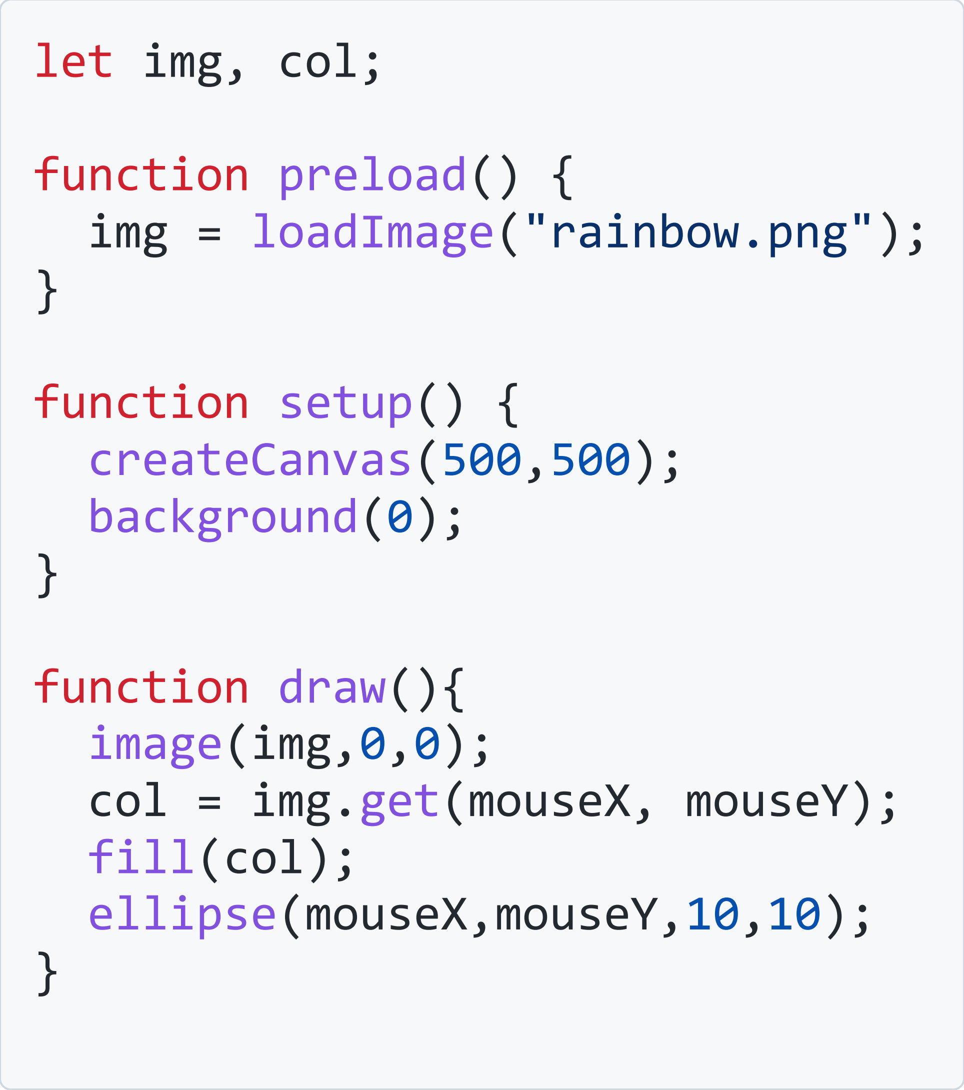
map(mouseX, 0 , width, 100, 500);

以上例子 我們需要 重新映射 系統變量 mouseX, mouseX的最小值為 0, mouseX的最大值為 是顯示框的闊度

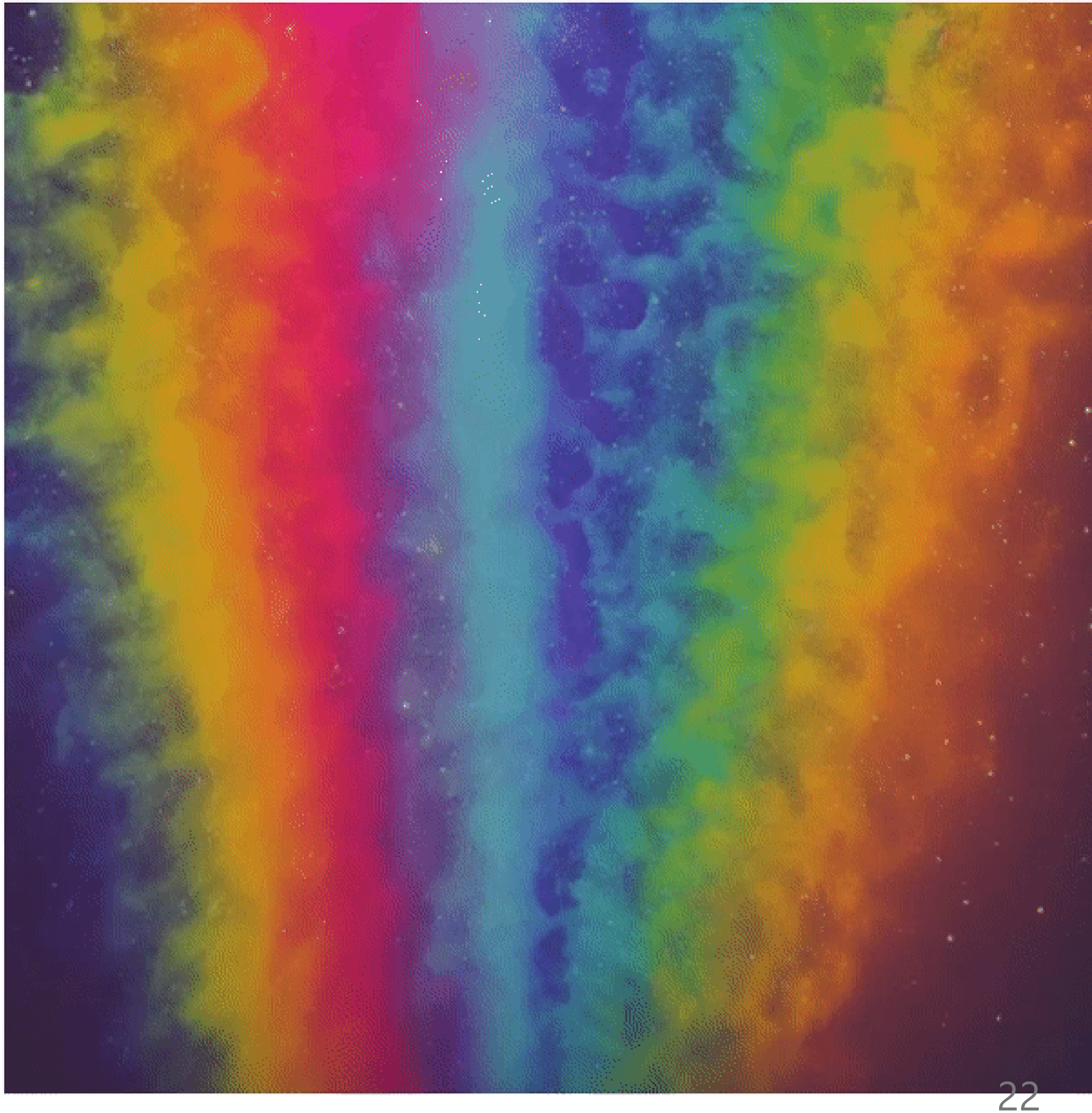
重新映射後的最小值為 100, 重新映射後 mouseX的最大值為 500

例如 : mouseX 在顯示框的中間 這時 變量 freq 就會等於 200

4.7 getColour



```
let img, col;

function preload() {
  img = loadImage("rainbow.png");
}

function setup() {
  createCanvas(500,500);
  background(0);
}

function draw(){
  image(img,0,0);
  col = img.get(mouseX, mouseY);
  fill(col);
  ellipse(mouseX,mouseY,10,10);
}
```



createVideo(影片檔案路徑)asd

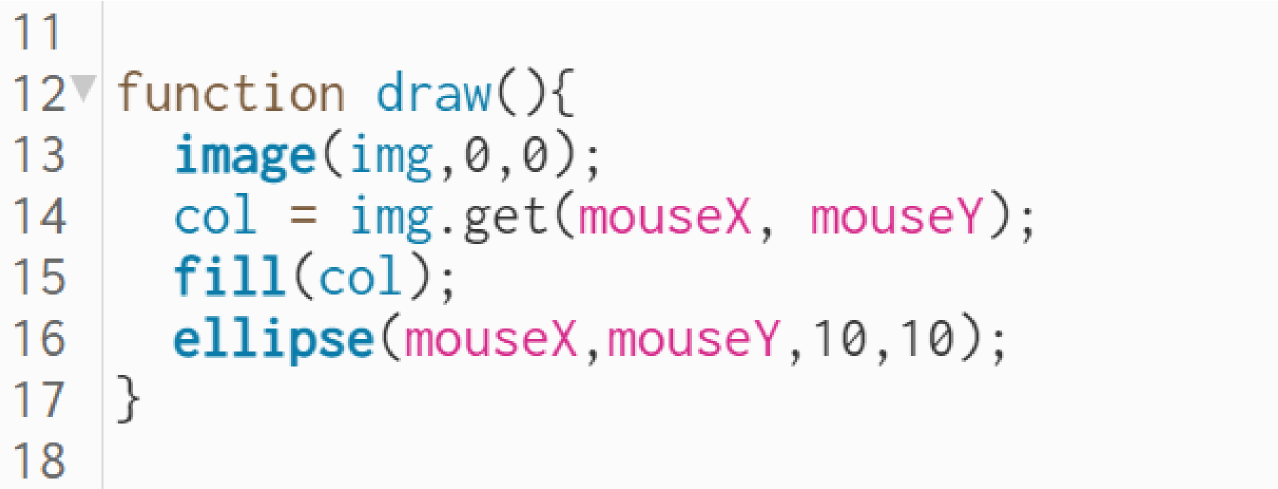
先宣告 video 變量，在 setup() 中使用 createVideo()

video.size(影片闊aasdsd度，影片高度)

video.volume(影片聲音asd大小) 0為無聲 1為最大聲

video.loop() 重複播放影片

video.hide() 隱藏HTML内的影像元素，免去重複顯示影像。



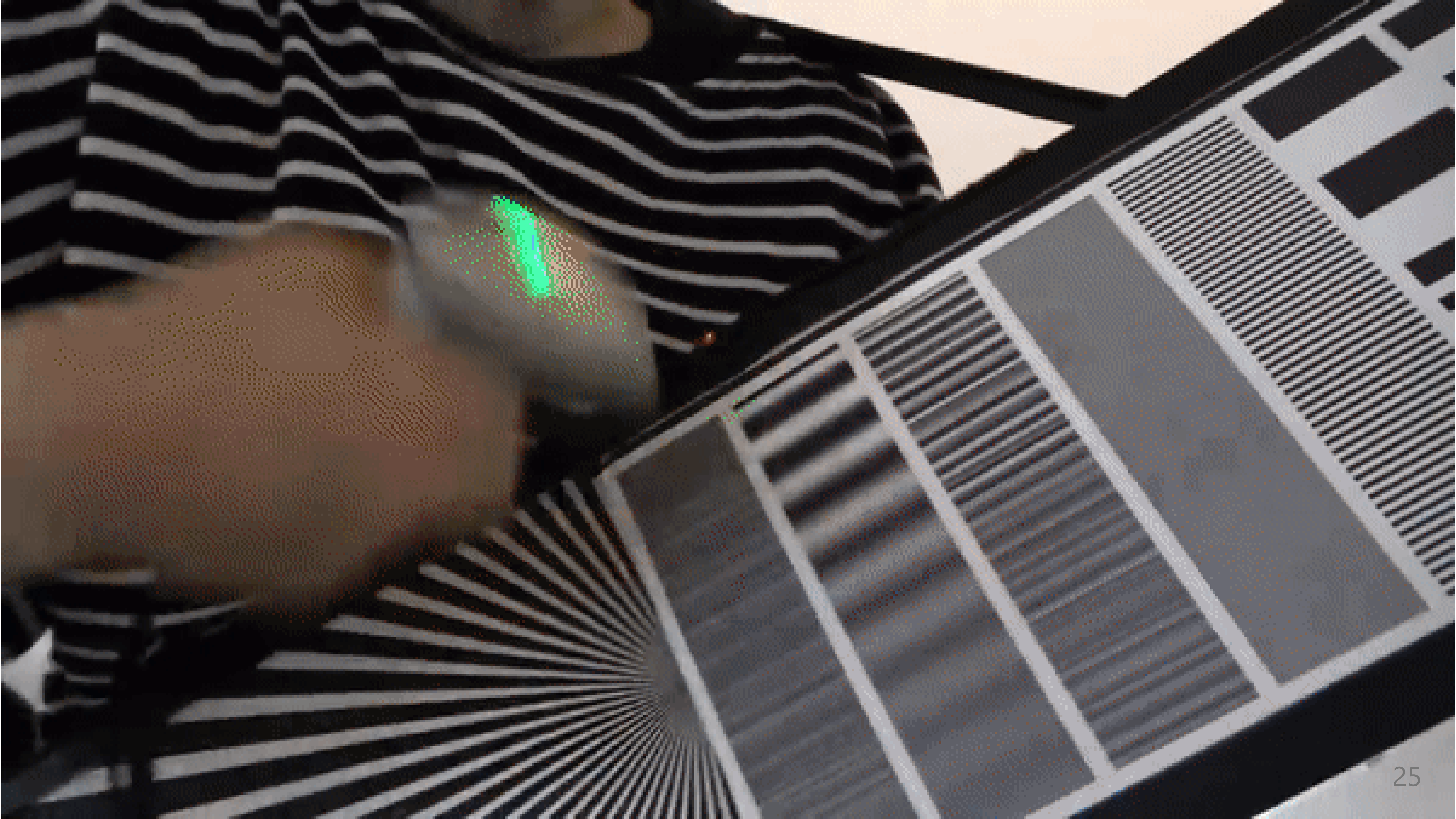
```

function draw(){
  image(img,0,0);
  col = img.get(mouseX, mouseY);
  fill(col);
  ellipse(mouseX,mouseY,10,10);
}

```

Sonification 資訊聲音化

把畫面轉化可聽的 聲音 或 音樂

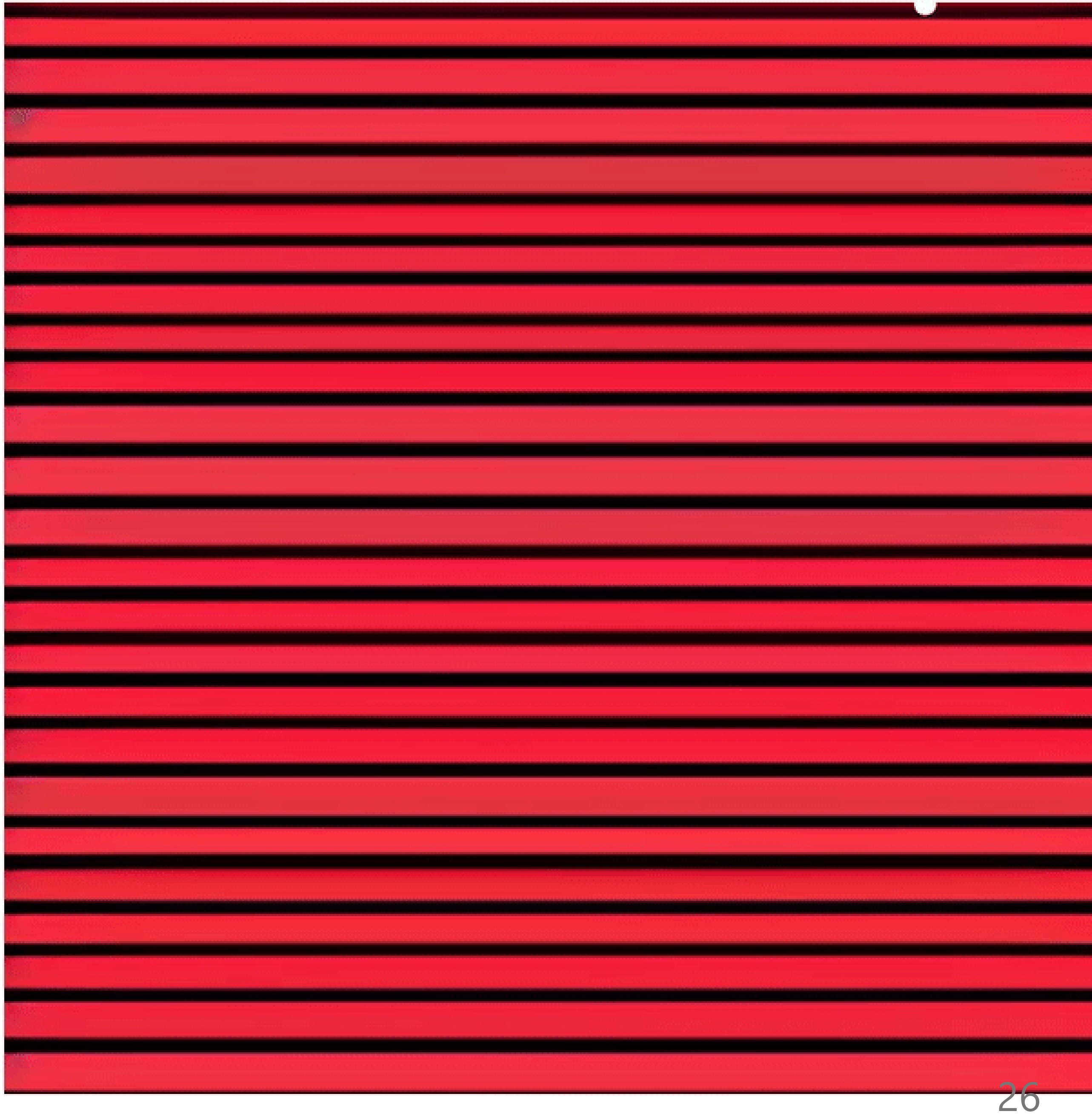
電磁祭囃子

https://www.youtube.com/watch?v=xlOjmLeQe3w

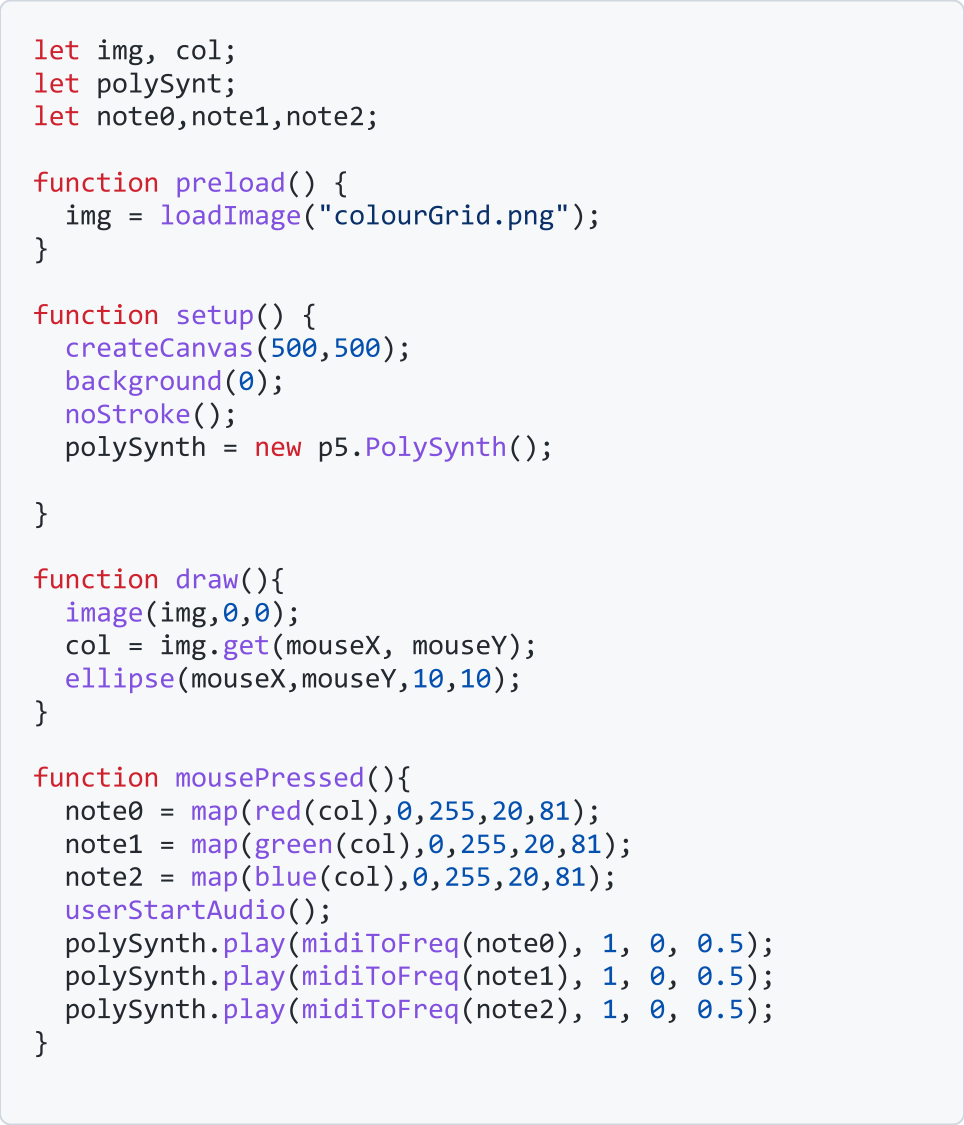


4.8 simpleSonification

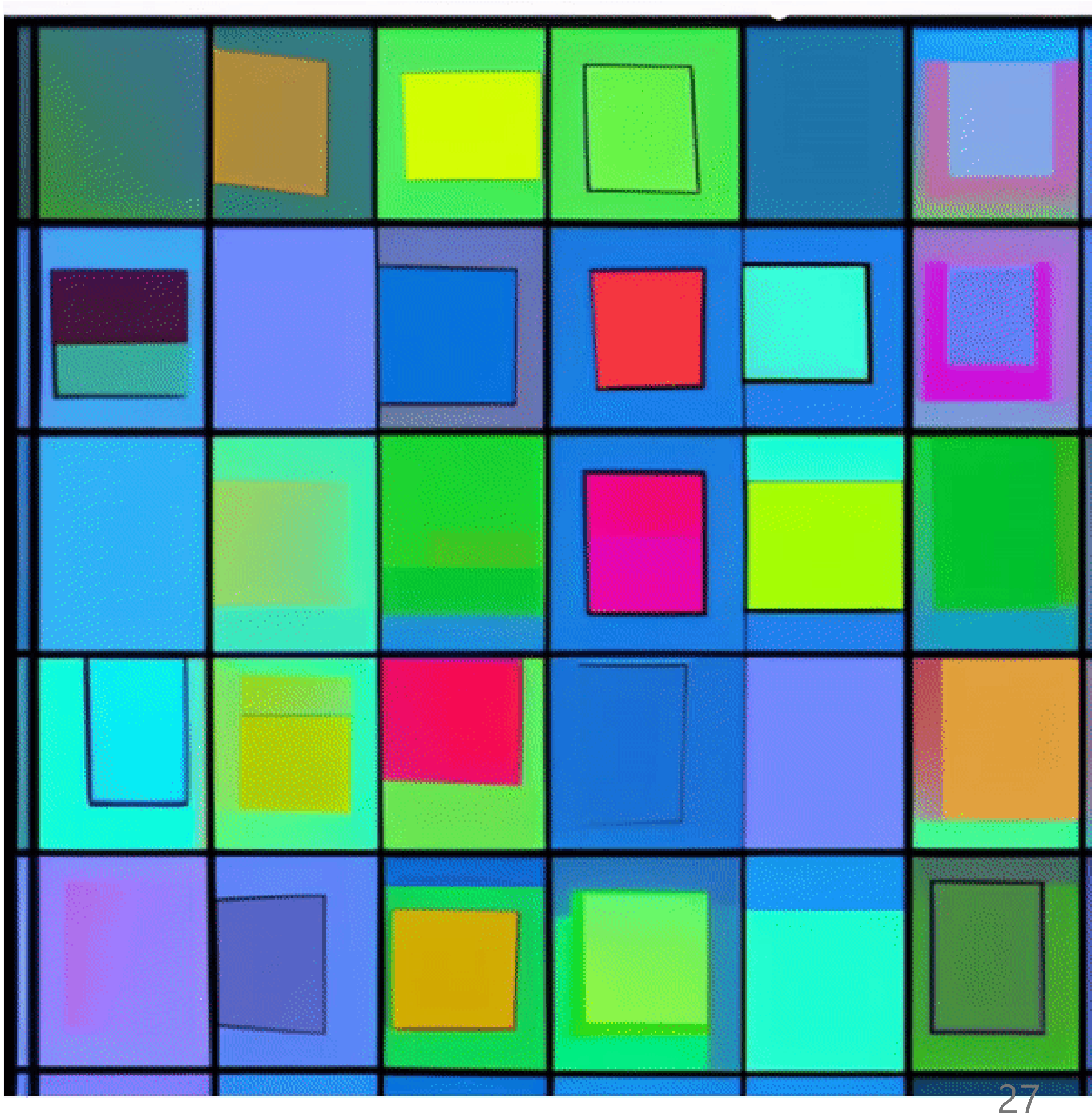
```
let img, col;
let osc;

function preload() {
  img = loadImage("redline.png");
}

function setup() {
  createCanvas(500,500);
  background(0);
  noStroke();
  osc = new p5.Oscillator();
  osc.setType("sine");
  osc.amp(0.5);
  osc.start();
}

function draw(){
  image(img,0,0);
  col = img.get(mouseX, mouseY);
  osc.freq(red(col));
  ellipse(mouseX,mouseY,10,10);
}
```



4.9 advanceSonification



```
let img, col;
let polySynt;
let note0,note1,note2;

function preload() {
  img = loadImage("colourGrid.png");
}

function setup() {
  createCanvas(500,500);
  background(0);
  noStroke();
  polySynth = new p5.PolySynth();

}

function draw(){
  image(img,0,0);
  col = img.get(mouseX, mouseY);
  ellipse(mouseX,mouseY,10,10);
}

function mousePressed(){
  note0 = map(red(col),0,255,20,81);
  note1 = map(green(col),0,255,20,81);
  note2 = map(blue(col),0,255,20,81);
  userStartAudio();
  polySynth.play(midiToFreq(note0), 1, 0, 0.5);
  polySynth.play(midiToFreq(note1), 1, 0, 0.5);
  polySynth.play(midiToFreq(note2), 1, 0, 0.5);
}
```



使用 red(); green(); blue(); 在當前顏色內抽取 紅、綠 、藍 的數值

red(顏色變量); -> 顏色變量中紅色所佔的數值

4.10 advanceSonification withTriadChord and Reverb

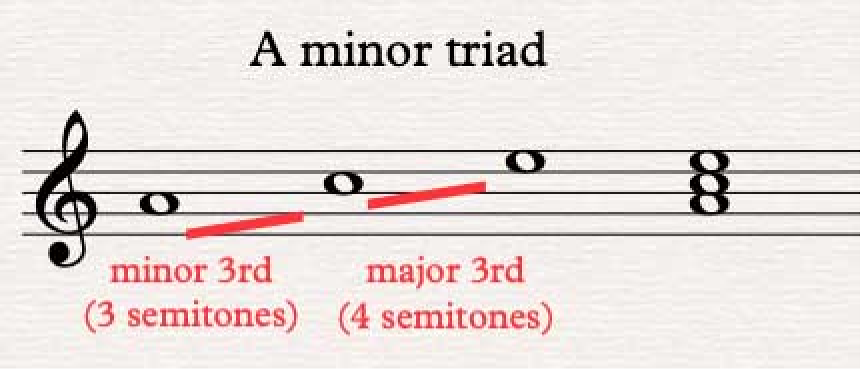
```
let img, col;
let polySynt;
let note0,note1,note2;
let reverb;

function preload() {
  img = loadImage("hkWindow.png");
}

function setup() {
  createCanvas(500,500);
  background(0);
  noStroke();
  polySynth = new p5.PolySynth();
  reverb = new p5.Reverb();
  reverb.process(polySynth, 3, 2);
}
```

```
function draw(){
  image(img,0,0);
  col = img.get(mouseX, mouseY);
  ellipse(mouseX,mouseY,10,10);
}

function mousePressed(){
  userStartAudio();
  note0 = map(brightness(col),0,100,20,61);
  note1 = note0 + 4;
  note2 = note1 + 3;
  polySynth.play(midiToFreq(note0), 1, 0, 0.5);
  polySynth.play(midiToFreq(note1), 1, 0, 0.5);
  polySynth.play(midiToFreq(note2), 1, 0, 0.5);
}
```

使用 brightness(); 取得顏色變量中顏色的光度

再利用 map(); 令畫面 最暗的部份 觸發琴鍵中的低音部份

```
note0 = map(brightness(col),0,100,20,61);
note1 = note0 + 4;
note2 = note1 + 3;
```



4.11 sonificationWithCamera

```
let col;
let polySynt;
let note0;
let reverb;
let capture;

function setup() {
  createCanvas(320, 240);
  background(0);
  noStroke();
  polySynth = new p5.PolySynth();
  reverb = new p5.Reverb();
  reverb.process(polySynth, 3, 2);
  capture = createCapture(VIDEO);
  capture.size(320, 240);
  capture.hide();
}
```

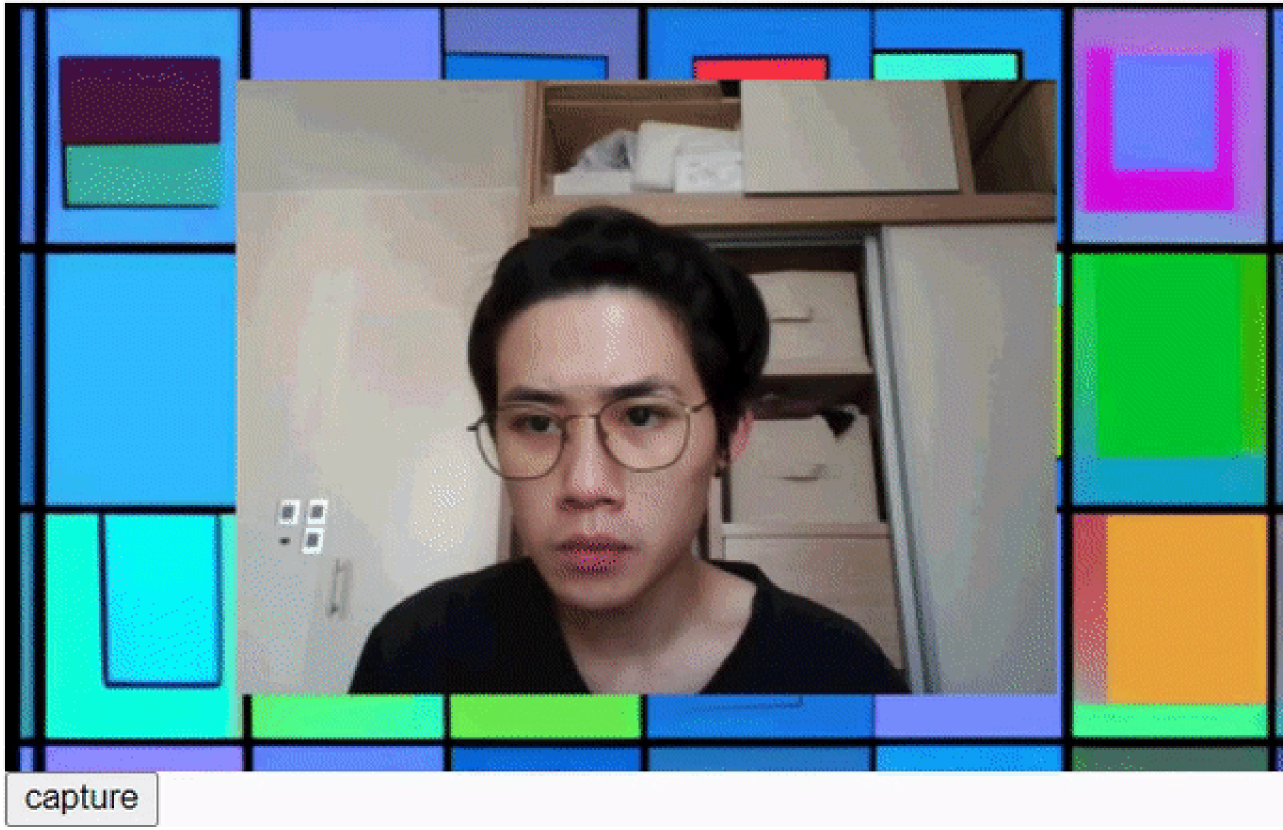
```
function draw() {
  image(capture, 0, 0);
  col = capture.get(mouseX, mouseY);
  ellipse(mouseX, mouseY, 10, 10);
}

function mousePressed() {
  userStartAudio();
  note0 = map(brightness(col), 0, 100, 20, 61);
  polySynth.play(midiToFreq(note0), 1, 0, 0.5);
}
```

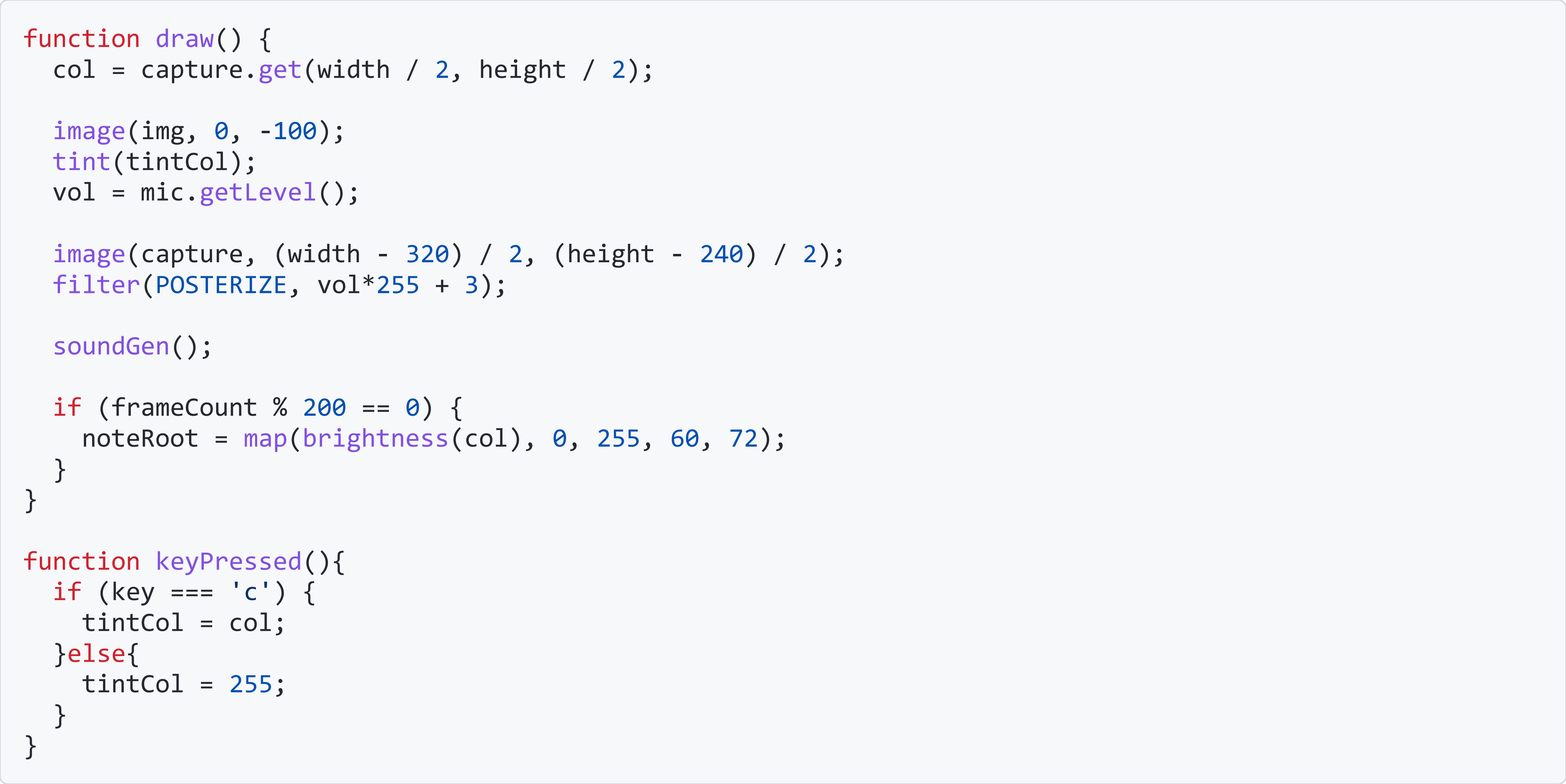
整合過去 課堂 學到的編碼

做一個與聲音互動的攝影軟件！

4.12 finalSoftware

```
let capture, col, tintCol;
let img;
let cameraFrame;
let captureButton;

let polySynth,
  reverb,
  noteRoot = 60,
  note;

let mic,vol;

function preload() {
  img = loadImage("colourGrid.png");
}

function setup() {
  createCanvas(500, 300);
  background(0);
  capture = createCapture(VIDEO);
  capture.size(320, 240);
  capture.hide();
  captureButton = createButton("capture");
  captureButton.mousePressed(takePicture);

  mic = new p5.AudioIn();
  mic.start();

  polySynth = new p5.PolySynth();
  reverb = new p5.Reverb();
  reverb.process(polySynth, 3, 2);

  tintCol = 255;
}
```





```
function draw() {
  col = capture.get(width / 2, height / 2);

  image(img, 0, -100);
  tint(tintCol);
  vol = mic.getLevel();

  image(capture, (width - 320) / 2, (height - 240) / 2);
  filter(POSTERIZE, vol*255 + 3);

  soundGen();

  if (frameCount % 200 == 0) {
    noteRoot = map(brightness(col), 0, 255, 60, 72);
  }
}

function keyPressed(){
  if (key === 'c') {
    tintCol = col;
  }else{
    tintCol = 255;
  }
}
```

```
function soundGen() {
  note = random([
    midiToFreq(noteRoot),
    midiToFreq(noteRoot + 2),
    midiToFreq(noteRoot + 4),
    midiToFreq(noteRoot + 5),
    midiToFreq(noteRoot + 7),
    midiToFreq(noteRoot + 9),
    midiToFreq(noteRoot + 11),
    midiToFreq(noteRoot + 12),
  ]);

  if (frameCount % 25 == 0) {
    userStartAudio();
    polySynth.play(note, 1, 0, 0.25);
  }
}

function takePicture() {
  save();
}
```

4.13 finalSoftwareMp3

```
let capture, col, tintCol;
let img;
let cameraFrame;
let captureButton;

let mySound,
  reverb;

let mic,vol;

function preload() {
  img = loadImage("sea.png");
  mySound = loadSound("YeYe.mp3");
}
```

```
function setup() {
  createCanvas(500, 300);
  background(0);
  capture = createCapture(VIDEO);
  capture.size(320, 240);
  capture.hide();
  captureButton = createButton("capture");
  captureButton.mousePressed(takePicture);

  mic = new p5.AudioIn();
  mic.start();

  userStartAudio();
  mySound.loop();
  reverb = new p5.Reverb();
  reverb.process(mySound, 1, 2);

  tintCol = 255;
}
```

```
function draw() {
  col = capture.get(width / 2, height / 2);

  image(img, 0, -100);
  tint(tintCol);
  vol = mic.getLevel();

  image(capture, (width - 320) / 2, (height - 240) / 2);
  filter(POSTERIZE, vol*255 + 6);

}

function keyPressed(){
  if (key === 'c') {
    tintCol = col;
  }else{
    tintCol = 255;
  }
}

function takePicture() {
  save();
}
```