

Welcome to
Digital Senses 數位通感

P5.js 基礎教學及基本編程技巧

編程(coding)作為藝術(art)創作的媒介(medium)

世界(world)、系統(system)、代理(agents)、自我組織(self organization)

甚麼是 P5.js?

p5.js是一個用作創意編碼的JavaScript函式庫，重點是讓藝術家、設計師、教育工作者、初學者和其他任何人都能獲得編碼的機會和包容！p5.js是免費和開源的，因為我們相信軟件和學習軟件的工具應該是人人都能獲得的。

<https://p5js.org/>



第一步

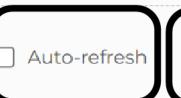
在網頁上使用 p5.js editor 準備創作環境

整合開發環境 IDE - Integrated Development Environment

- 輔助程式開發人員開發軟體的應用軟體，在開發工具內部就可以輔助編寫原始碼文字、並編譯打包成為可用的程式，有些甚至可以設計圖形介面。

https://en.wikipedia.org/wiki/Integrated_development_environment

p5.js editor 的介面與操作



by chanlongfunglazarus



程序名稱 program name (雙擊滑鼠左鍵以更改程序名稱)
Double-click the left mouse button to change the program name

自動更新程序 auto-refresh

停止程序運行 stop

運行程序 play

p5*

Auto-refresh 1.1 Basic_Run by chanlongfunglazarus

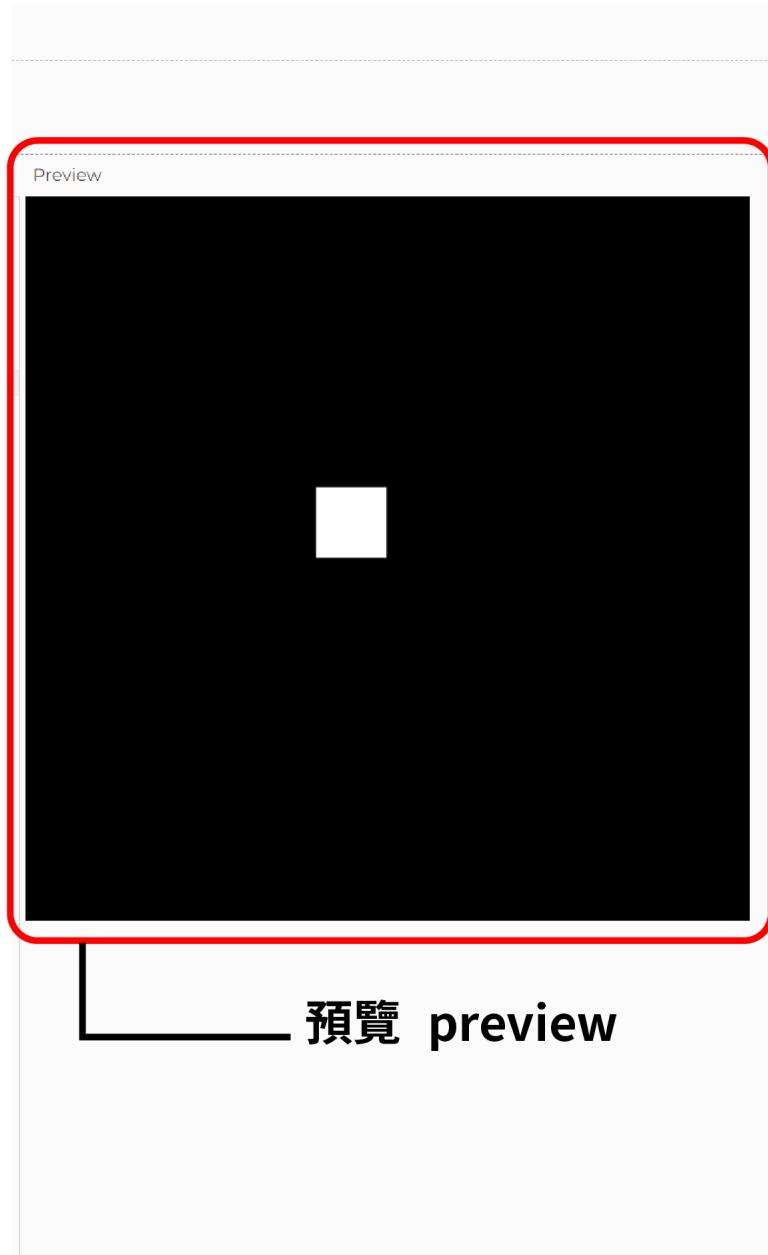
sketch.js

```
function setup() {
  createCanvas(500,500);
  background(0);
  rect(200, 200, 50, 50);
}
```

編輯器 editor

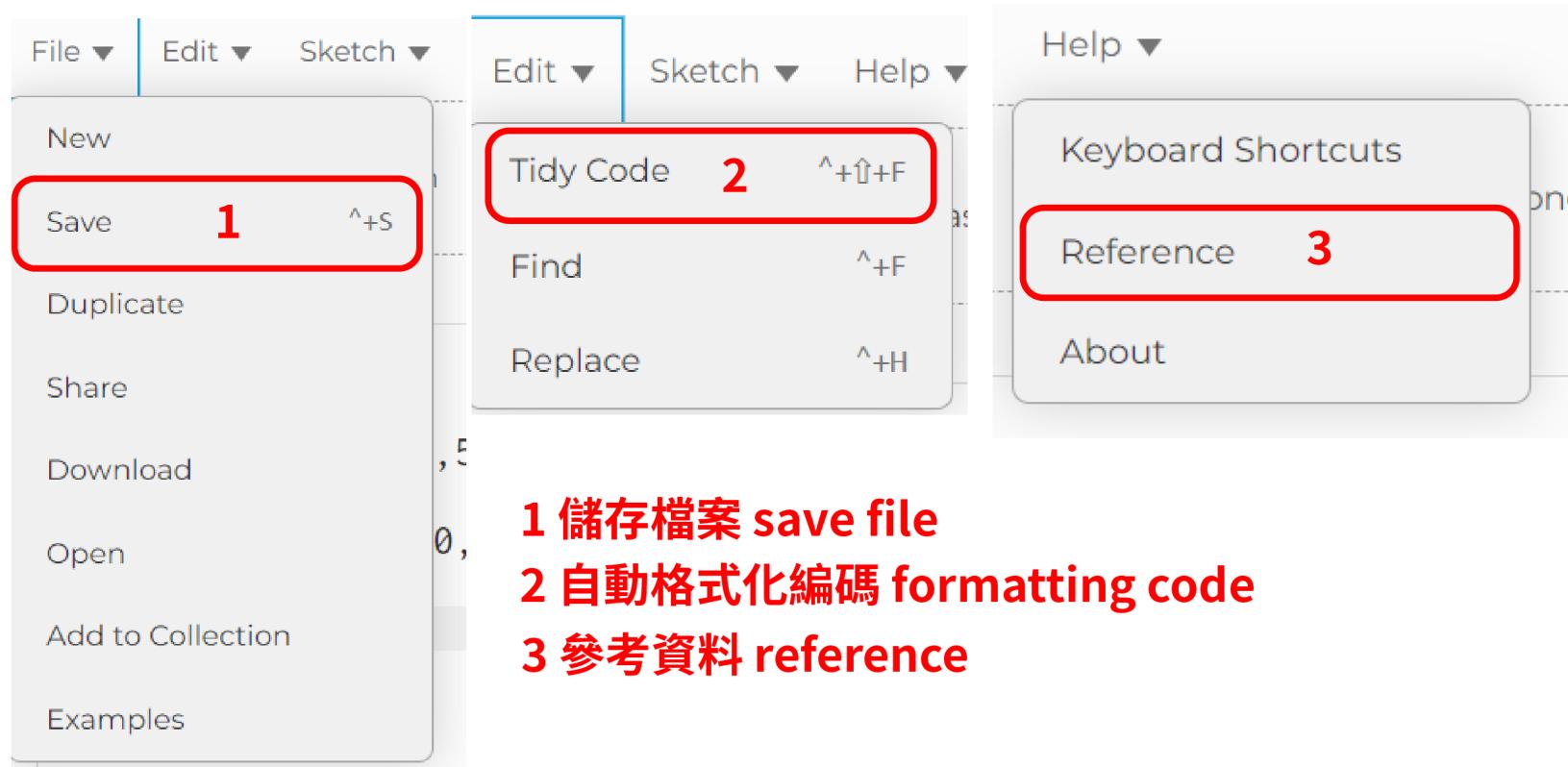
程式碼行號 line number

The screenshot shows the p5.js web-based code editor. At the top left is the p5* logo. To its right are two circular buttons: a pink one with a white play icon and a grey one with a white square icon. Next is a checkbox labeled "Auto-refresh". To the right of the checkbox is the file name "1.1 Basic_Run" followed by a small icon and the author's name "by chanlongfunglazarus". Below this is the code area titled "sketch.js". It contains a single function "setup" with code to create a canvas of size 500x500, set the background to black, and draw a white rectangle at position (200, 200) with dimensions 50x50. Line numbers 1 through 6 are displayed on the left side of the code. At the bottom right of the code area is a grey button with a white play icon, labeled "Preview".





工具欄 tool bar



1 儲存檔案 save file

2 自動格式化編碼 formatting code

3 參考資料 reference

Can't find what you're looking for? You may want to check out [p5.sound](#).

You can also download an offline version of the reference.

3D
Color
Constants
DOM

Data
Environment
Events
Foundation

IO
Image
Math
Rendering

Shape
Structure
Transform
Typography

Environment Color

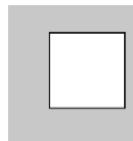
[describe\(\)](#)
[describeElement\(\)](#)
[textOutput\(\)](#)
[gridOutput\(\)](#)
[print\(\)](#)
[frameCount](#)
[deltaTime](#)
[focused](#)
[cursor\(\)](#)
[frameRate\(\)](#)
[noCursor\(\)](#)
[displayWidth](#)
[displayHeight](#)

[Creating & Reading](#)
[alpha\(\)](#)
[blue\(\)](#)
[brightness\(\)](#)
[color\(\)](#)
[green\(\)](#)
[hue\(\)](#)
[lerpColor\(\)](#)
[lightness\(\)](#)
[red\(\)](#)
[saturation\(\)](#)
[p5.Color](#)

[Setting](#)
[background\(\)](#)
[clear\(\)](#)
[colorMode\(\)](#)
[fill\(\)](#)
[noFill\(\)](#)
[noStroke\(\)](#)
[stroke\(\)](#)
[erase\(\)](#)
[noErase\(\)](#)

rect()

Examples



```
// Draw a rectangle at location (30, 20) with a
// width and height of 55.
rect(30, 20, 55, 55);
describe('white rect with black outline in mid-
right of canvas');
```

[edit](#) [reset](#) [copy](#)

Description

Draws a rectangle on the canvas. A rectangle is a four-sided closed shape with every angle at ninety degrees. By default, the first two parameters set the location of the upper-left corner, the third sets the width, and the fourth sets the height. The way these parameters are interpreted may be changed with the [rectMode\(\)](#) function.

Syntax

```
rect(x, y, w, [h], [tl], [tr], [br], [bl])
```

```
rect(x, y, w, h, [detailX], [detailY])
```

Sketch Files

index.html

sketch.js

style.css

Sketch Files

Create folder

Create file

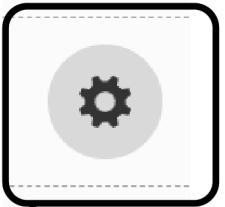
Upload file

按此打開文件欄 click here to open Sketch Files

Upload File

Drop files here or click to use the file browser

按此上載文件(聲音/圖片) click here to upload files (sound/image)



設定
setting

Settings

General settings Accessibility

Theme
Light Dark High Contrast

Text Size
18
- Decrease + Increase

Autosave
On Off

Autoclose Brackets and Quotes
On Off

Word Wrap
On Off

Settings

General settings Accessibility

Line numbers
On Off

Lint warning sound
On Off Preview sound

Accessible text-based canvas
Used with screen reader

Plain-text Table-text



登入/申請帳戶以儲存檔案
Login/Sign up an account to save sketch

Log In

Email or Username

Password

Log In

or

 Login with GitHub

 Login with Google

Don't have an account? [Sign Up](#)

第一步

在網頁上使用 p5.js editor 開始創作

今天課堂將會使用的例子

<https://editor.p5js.org/chalongfunglazarus/collections/-UFmPbjGw>

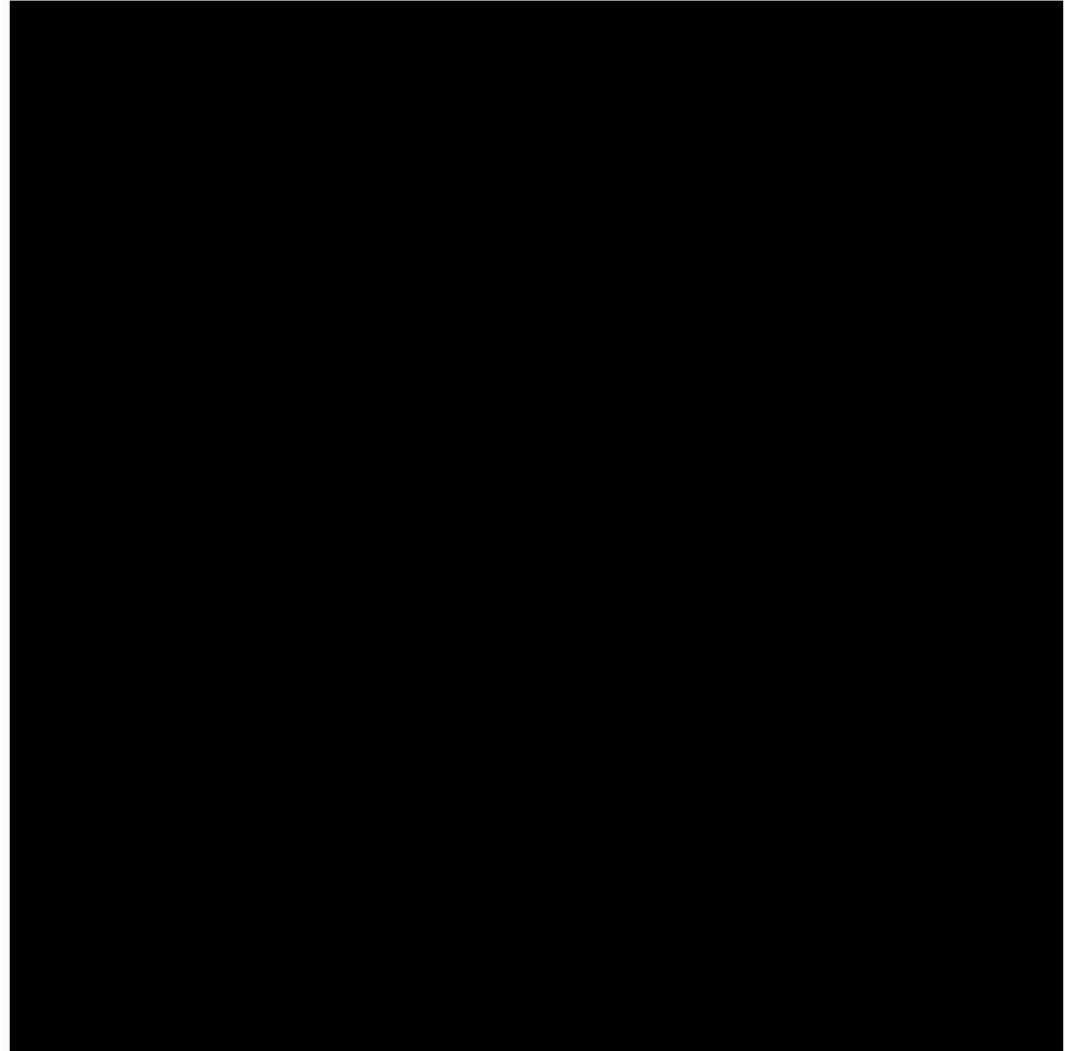
(1)

二維平面圖形、顏色與滑鼠

1.1 setupCanvas

```
//function setup()  
//在程式運行的第一幀 執行  
function setup() {  
  
    //設定一個500px*500px大的顯示框  
    createCanvas(500,500);  
  
    //設定顯示框的背景色為黑色  
    background(0);  
}
```

Preview



有機會遇上的問題

1. 在編碼中遺漏了特定符號 如 ;(),
2. 編碼不在 {} 之內

留意!!! Syntax 句法

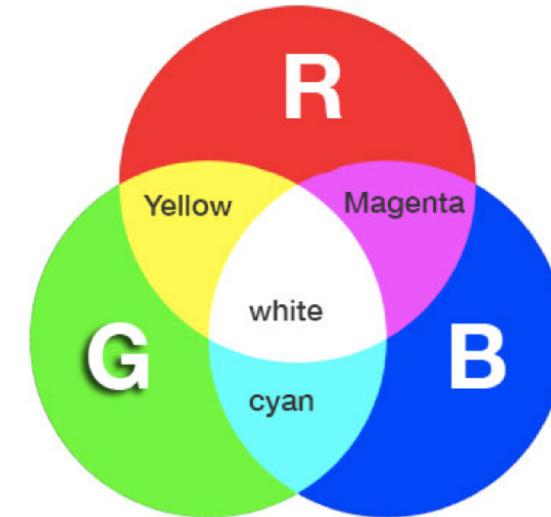
- 指一門語言裡支配句子結構, 直到組成句子的規則或過程。

<https://en.wikipedia.org/wiki/Syntax>

電腦如何理解顏色？

三原色光模式 (RGB color model)

- 一種加色模型，將紅、綠、藍三原色的色光以不同的比例相加，以合成產生各種色彩光。主要目的是在電子系統中檢測，表示和顯示圖像，比如電視和電腦，利用大腦強制視覺生理模糊化，將紅綠藍三原色子像素合成為一色彩像素，產生感知色彩。



RED : #FF0000 or rgb(255,0,0)
GREEN : #00FF00 or rgb(0,255,0)
BLUE : #0000FF or rgb(0,0,255)

tutorial.techaltum.com



```
// R, G & B integer values  
background(255, 204, 0);  
describe('canvas with yellow background');
```

[edit](#) [reset](#) [copy](#)

https://en.wikipedia.org/wiki/RGB_color_model

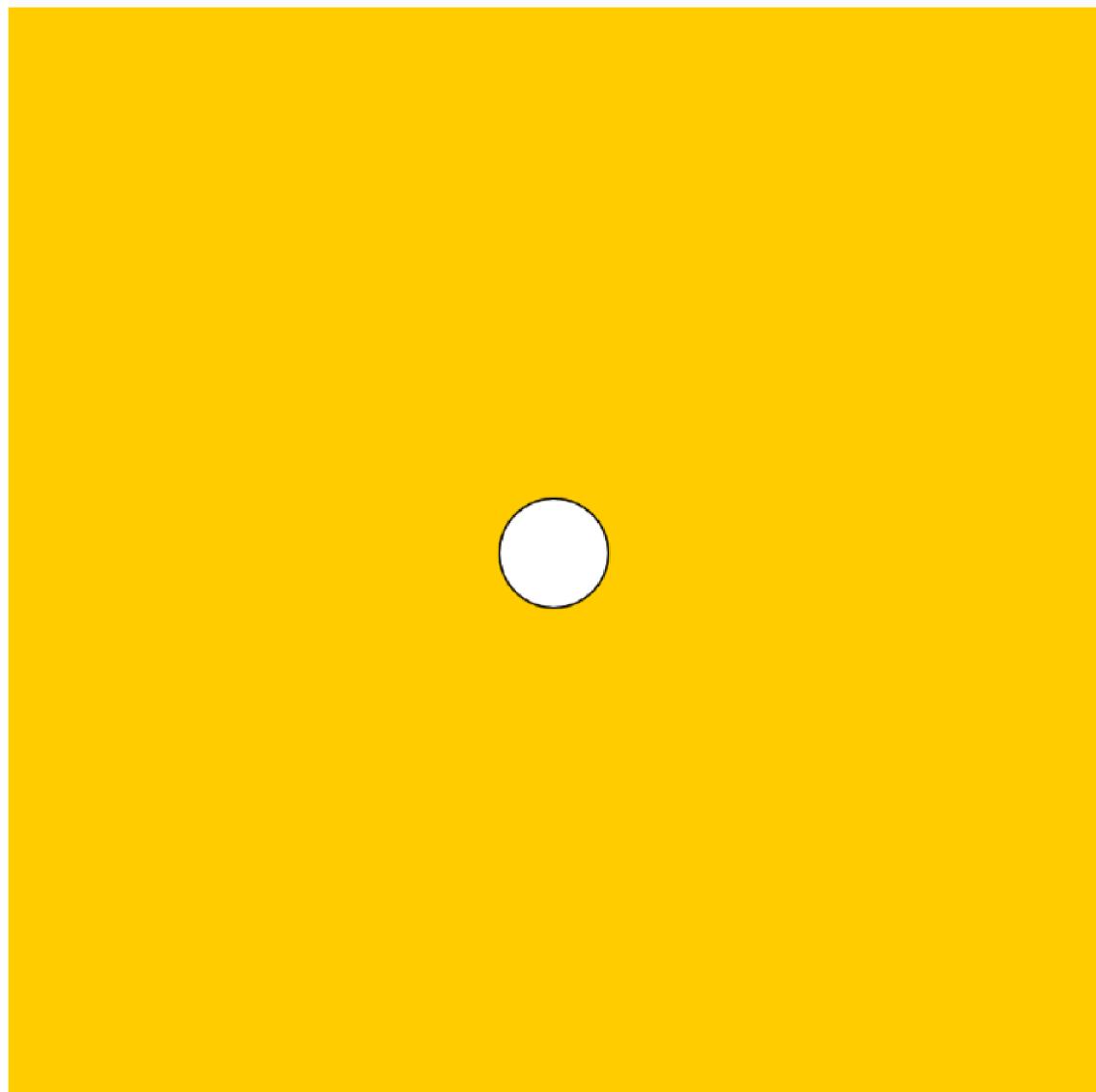
試一下! 改變顯示框的背景色

background(red , green , blue);

1.2 setupEllipse

```
function setup() {  
  createCanvas(500, 500);  
  
  //更改顯示框的背景色  
  background(255, 204, 0);  
  
  //在顯示框的  
  //X:250px 和 Y:250px 位置  
  //添加一個圖形  
  ellipse(250, 250, 50, 50);  
}
```

Preview



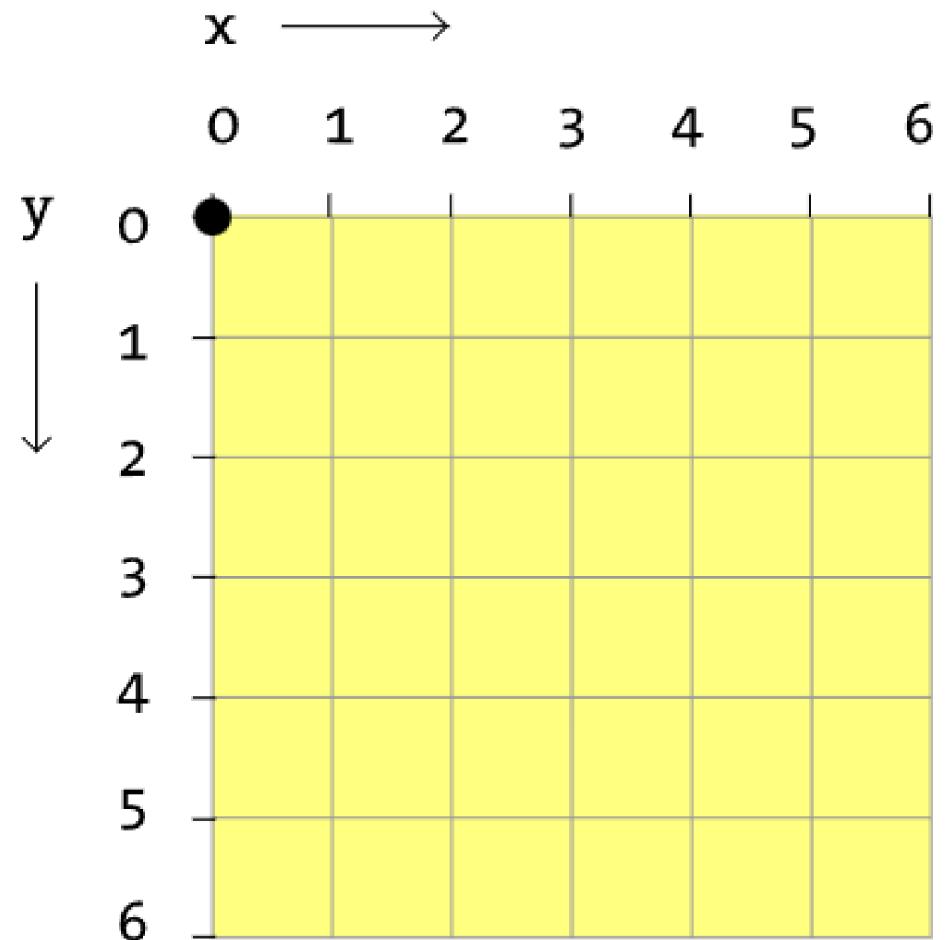
不明白ellipse();內的數值有甚麼意義？

ellipse(x座標 , y座標 , width圓形的闊度 , height圓形的高度);

p5.js 的座標系統

左上角的座標為 (0,0)

以像素為基本單位



Computer

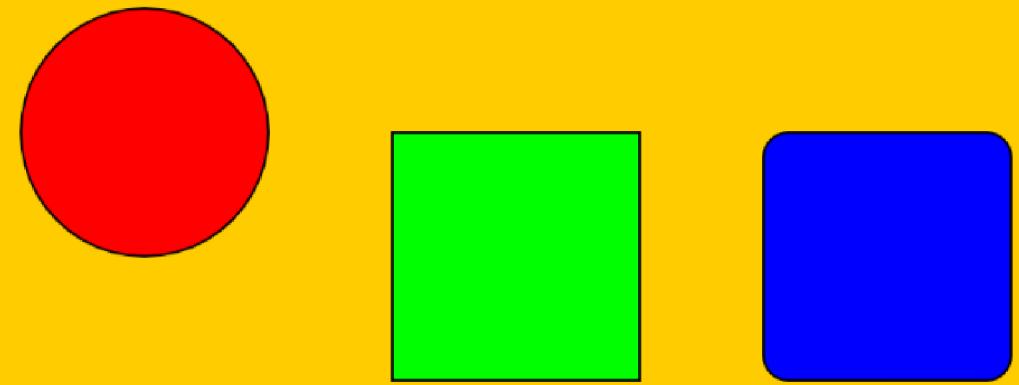
來試一下

添加不同形狀 及 改變形狀的顏色吧!

接下來

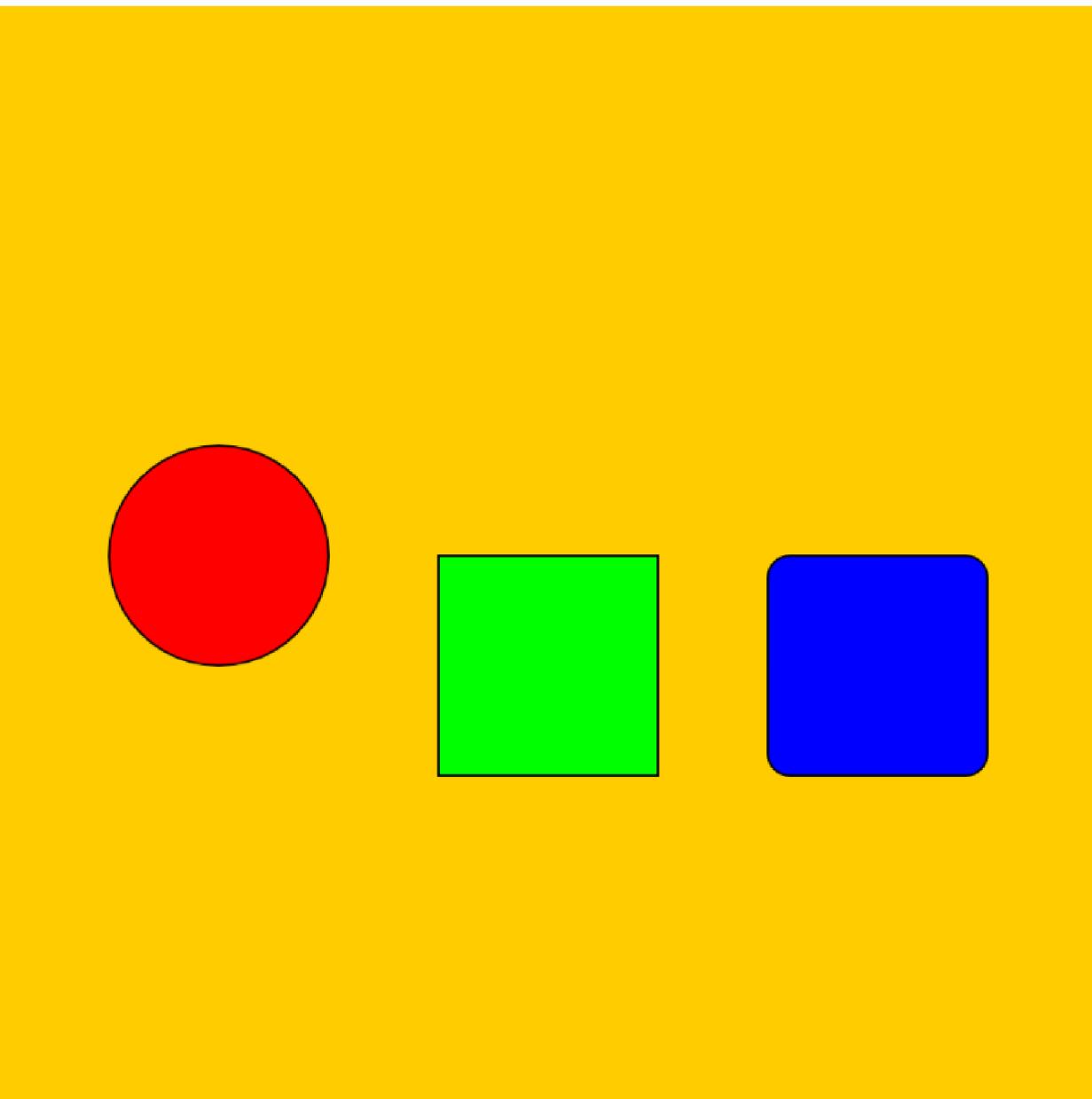
一起到 Reference 頁面查找
rect(); 和 fill();

試一下使用編碼編寫出這個
畫面



1.3 setupShape&Colours

```
function setup() {  
  createCanvas(500, 500);  
  background(255, 204, 0);  
  fill(255,0,0);  
  ellipse(100,250,100,100);  
  fill(0,255,0);  
  rect(200,250,100,100);  
  fill(0,0,255);  
  rect(350,250,100,100,10);  
}
```



人機 簡易互動，隨機

setup() 以後另一個 *重要的 函数

draw()

draw()函數持續執行其 {} 內包含的編碼

在 draw() {} 內的編碼會循環執行

- 在右面的例子 background() 這句編碼會循環執行

```
1 function setup() {  
2   createCanvas(400, 400);  
3 }  
4  
5 function draw() {  
6   background(255, 204, 0);  
7 }
```

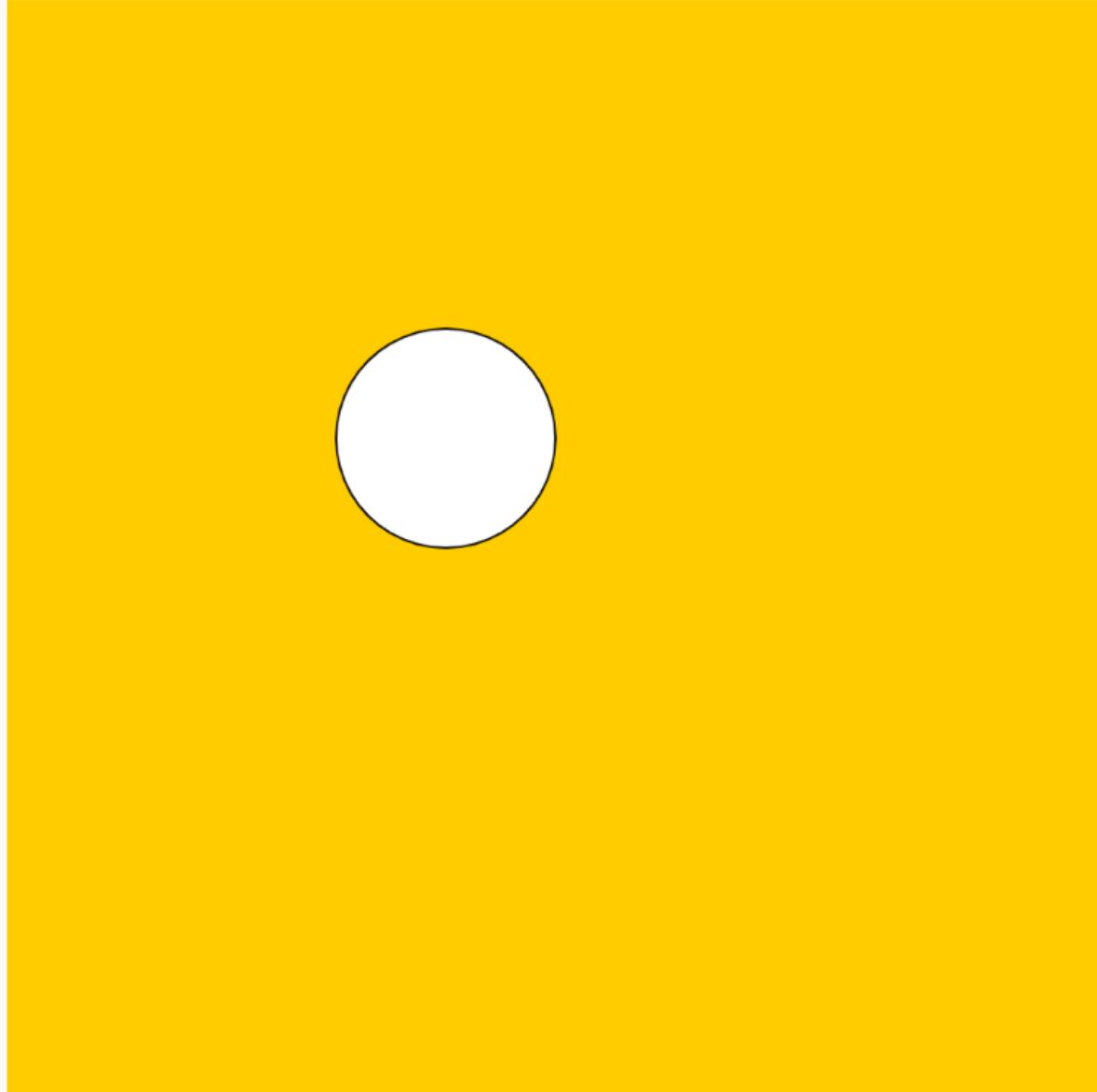
>

sketch.js

Saved: just now

Preview

```
1▼ function setup() {  
2  createCanvas(500, 500);  
3 }  
4  
5▼ function draw() {  
6  background(255, 204, 0);  
7  ellipse(200,200,100,100);  
8 }
```



為什麼沒有分別？

試一下 在XY座標位置使用 mouseX 和 mouseY

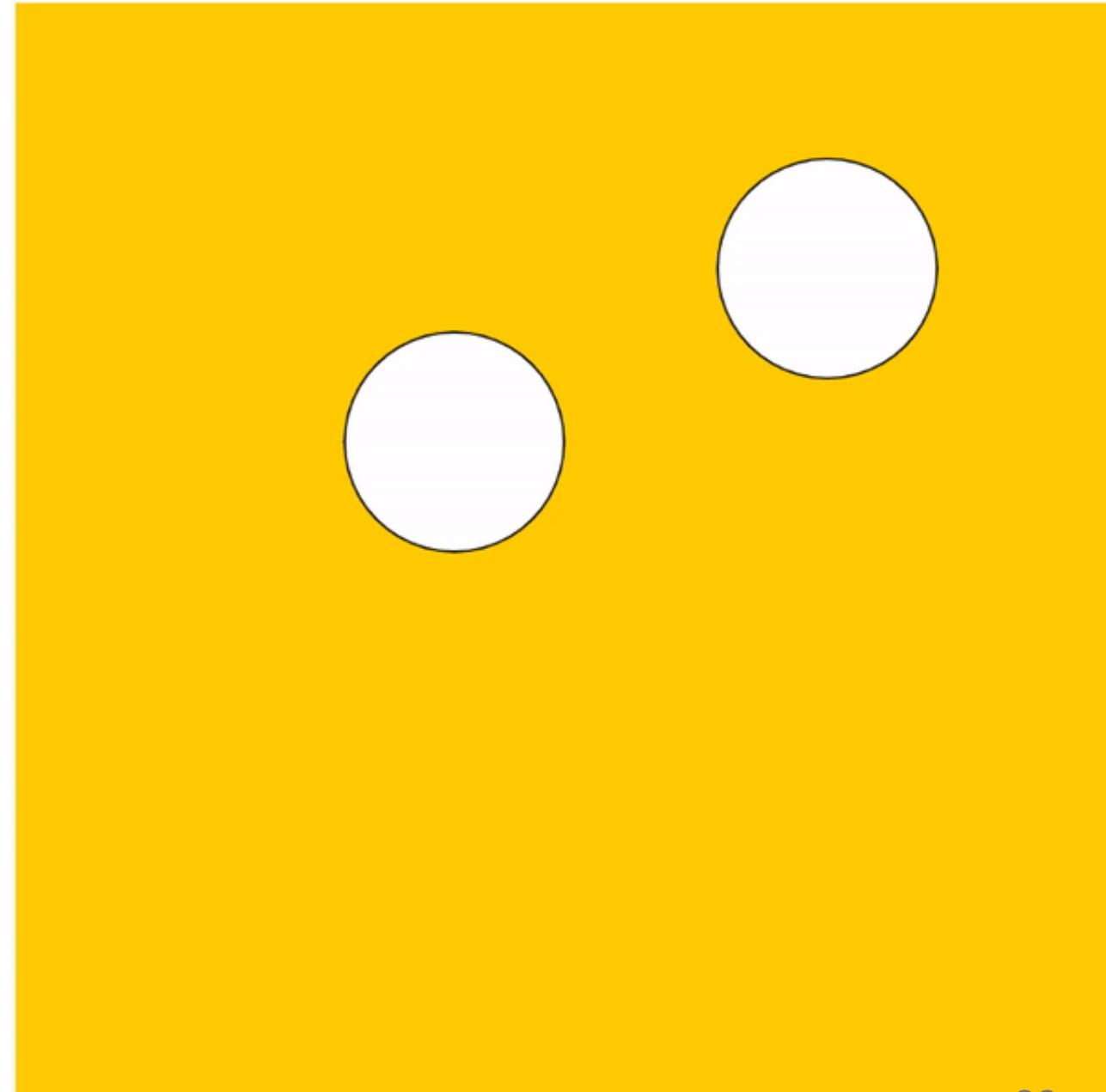
有關 mouseX 與 mouseY

mouseX 是一種系統變量，mouseX包含滑鼠當前的 X座標位置

mouseY 同樣是系統變量，mouseY包含滑鼠當前的 Y座標位置

1.4 drawLoop

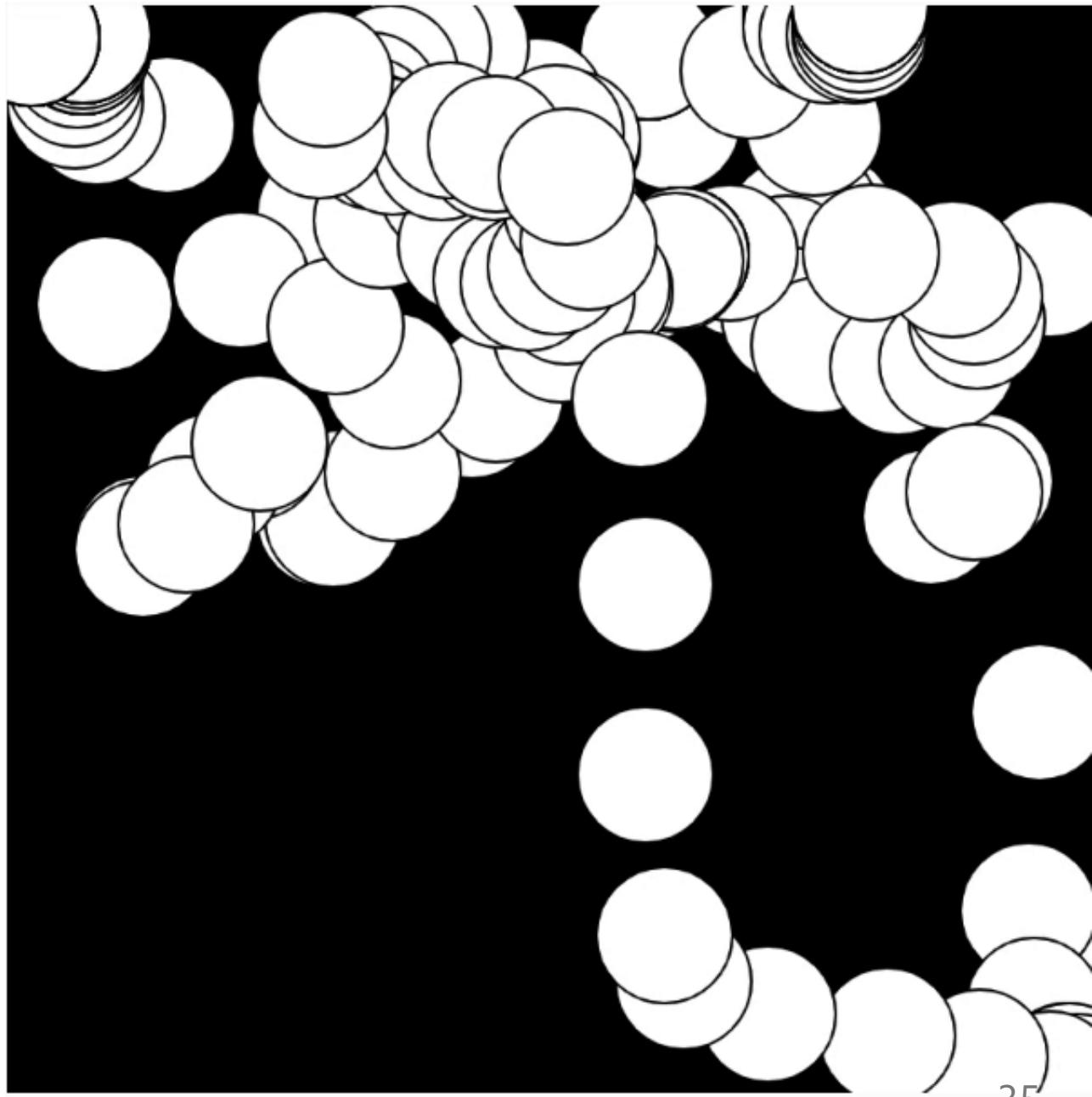
```
function setup() {  
  createCanvas(500, 500);  
  
}  
  
function draw() {  
  background(255, 204, 0);  
  ellipse(200,200,100,100);  
  ellipse(mouseX,mouseY,100,100);  
}
```



**若果只 更新 圖形部分
不更新 背景 效果會如何呢？**

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

function draw() {
  ellipse(mouseX,mouseY,50,50);
}
```



接下來 做個彩虹刮刮卡吧!

彩虹刮刮卡

可以分解為~

1. 畫面的背景是黑色的
2. 一個跟著滑鼠移動的圓形
3. 顏色要是變化多端

那我們要學一下 random() 隨機 這個函數了

random() 返回一個 隨機的浮點數 (float)

使用 random() 我們需要一個數值的範圍

例如：由 0 至 1 之間返回一個隨機數值

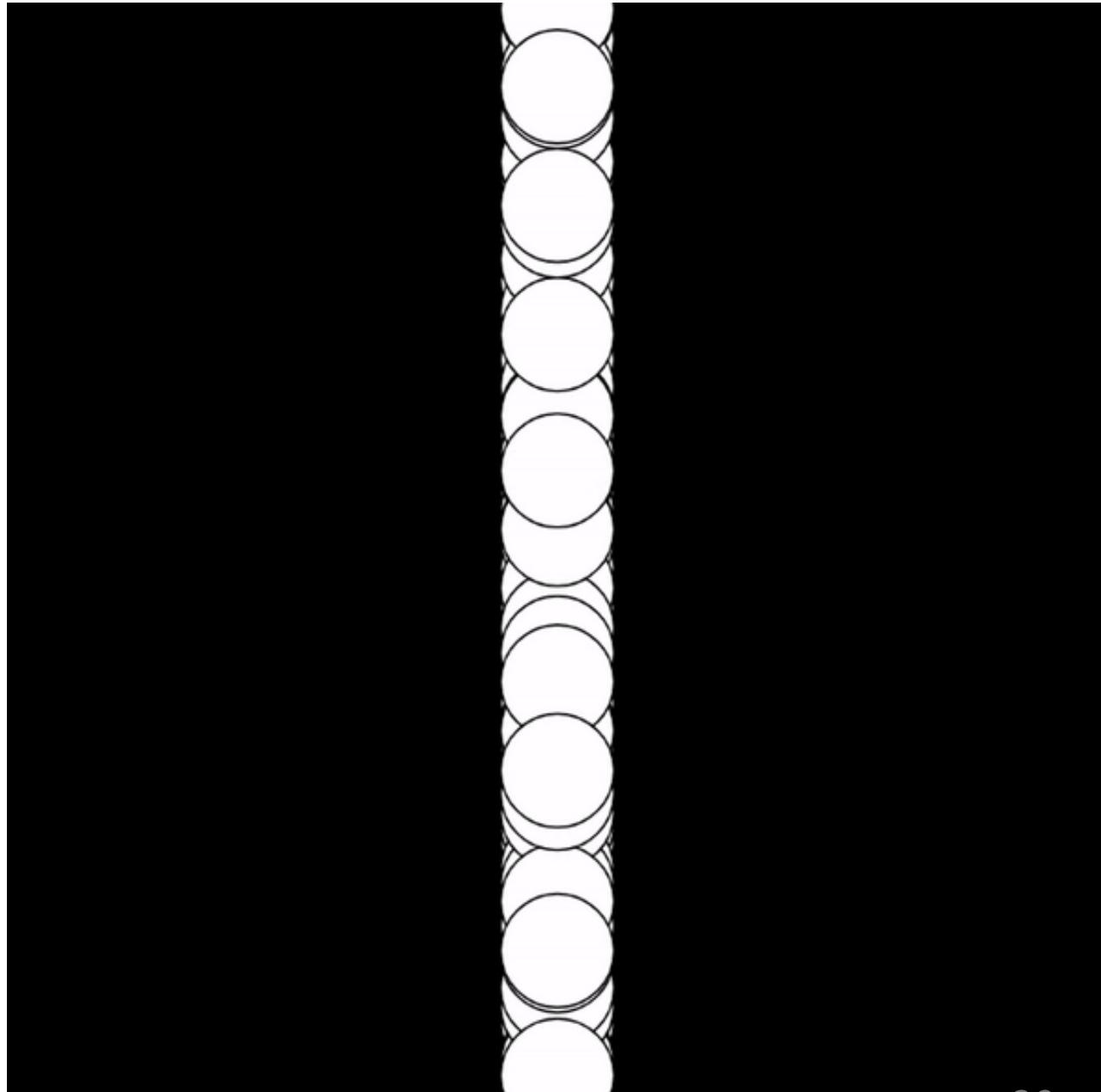
`random(0, 1);`

`random([min], [max]);`

1.5 random

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

function draw() {
  ellipse(250,random(0,500),50,50);
}
```



RGB 的數值範圍為 0 - 255, 256色

那 random(0, 255) 加到 fill(); 又會如何?

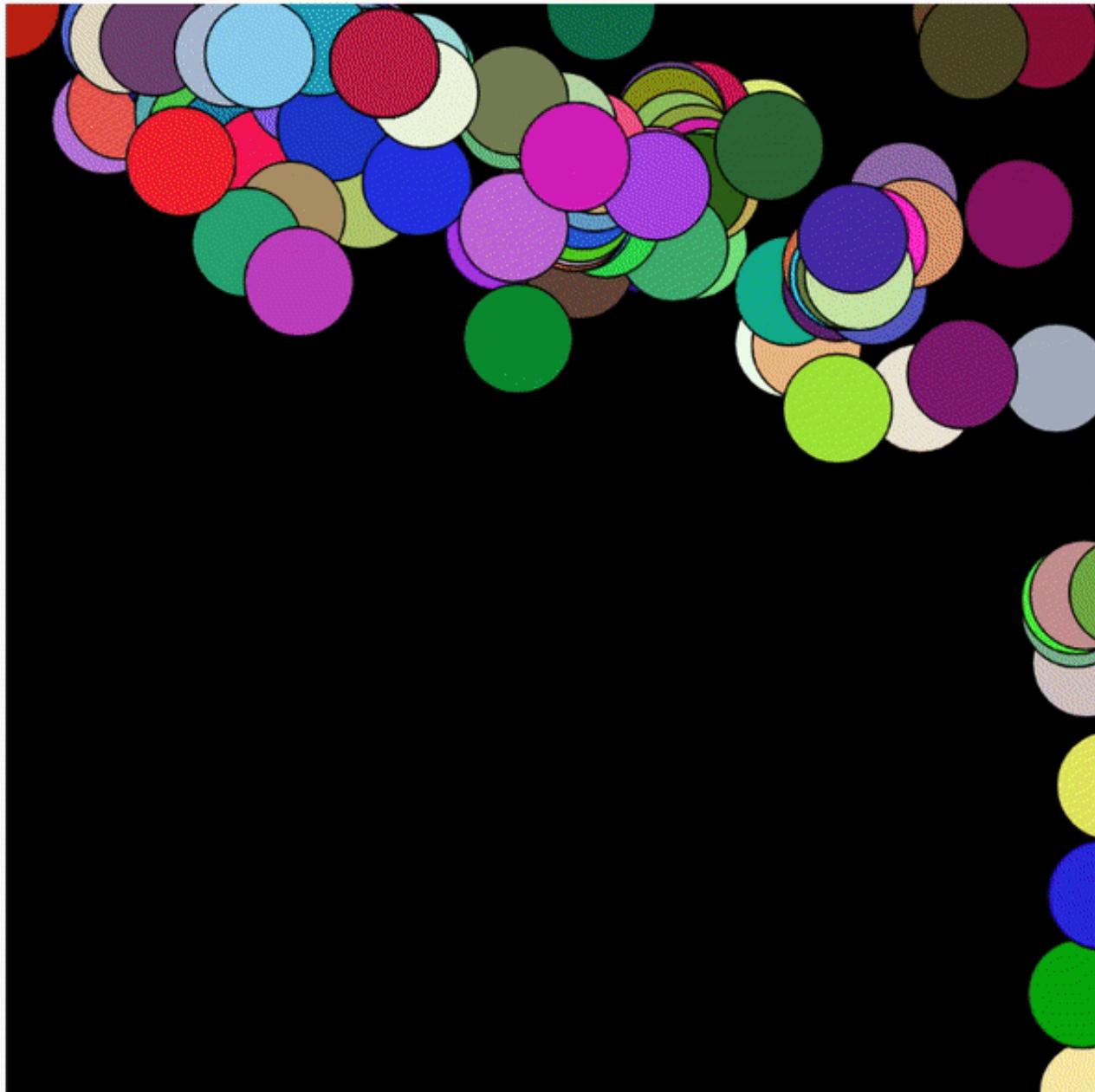
試一下這句編碼

`fill(random(0, 255),random(0, 255),random(0, 255));`

1.6 drawRainbow

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

function draw() {
  fill(random(0, 255),
    random(0, 255),
    random(0, 255));
  ellipse(mouseX, mouseY, 50, 50);
}
```



**來使用 random(), mouseX, mouseY 及更多學到的
發揮你的創意吧 ~!**

Welcome to
Digital Senses 數位通感

影像：自製攝影及影像載入軟件

(2)

圖像與影片

**大家都有使用相機拍攝
的經驗嗎？**

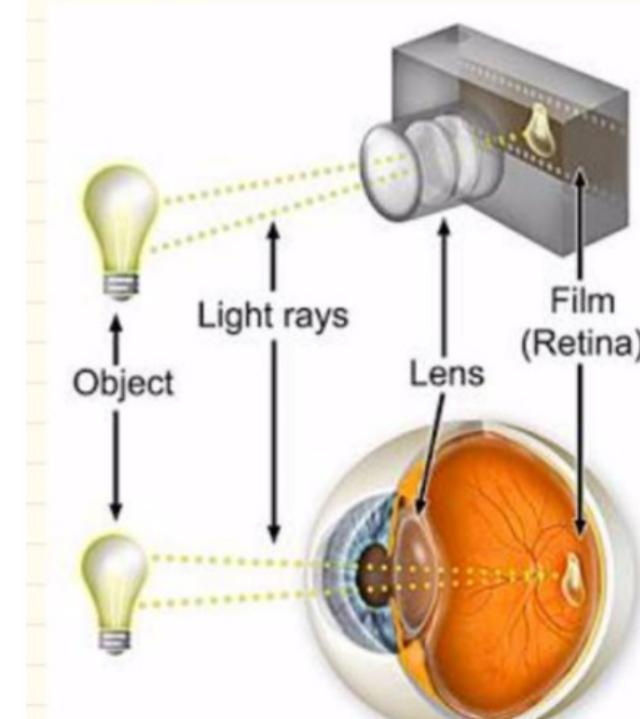
對相機的了解又有幾多呢？



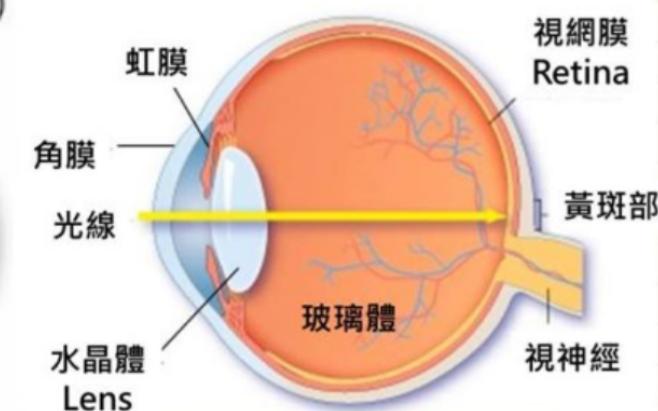
照相機的成像原理類近我們的肉眼

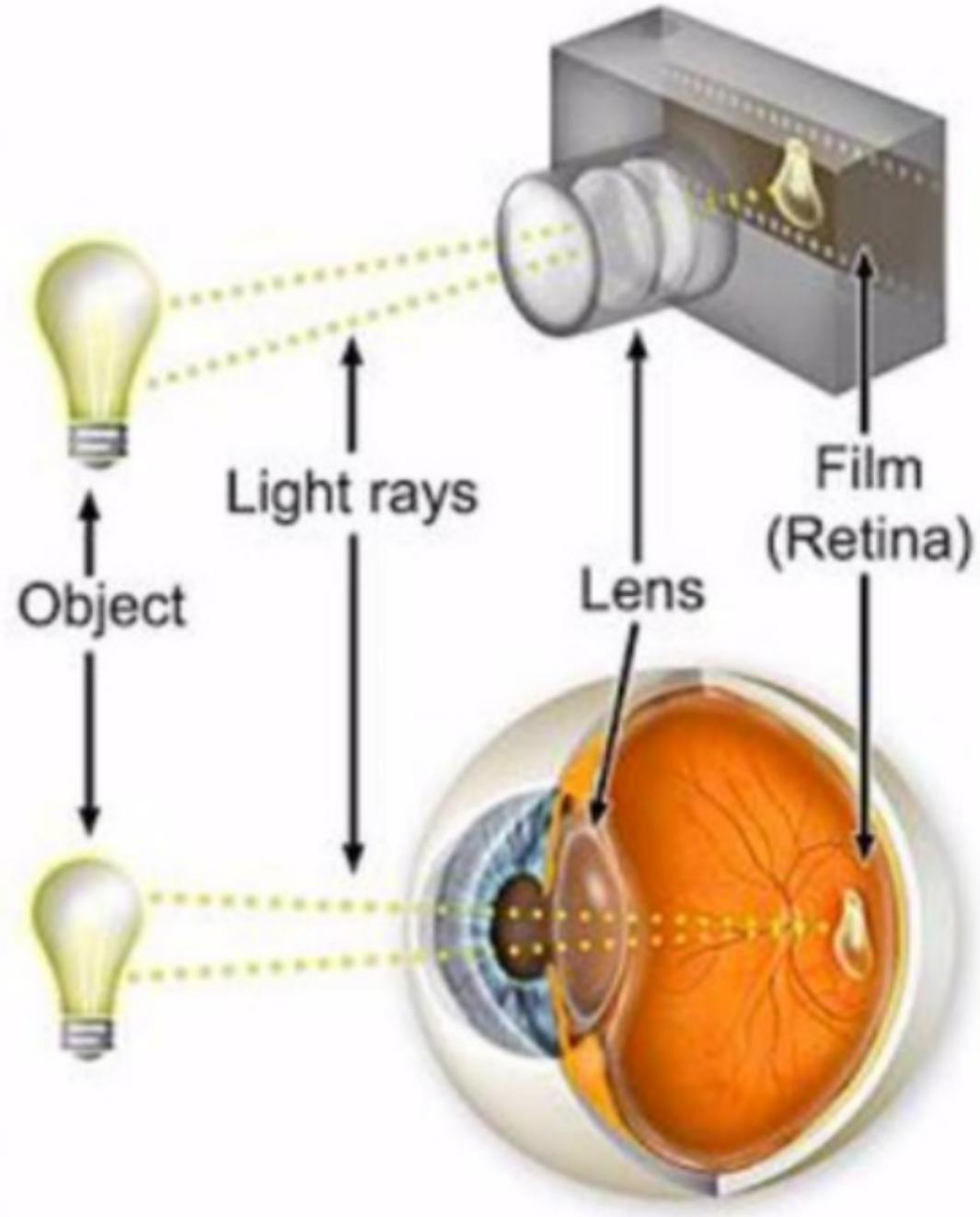
景物所反射的光線進入眼睛，眼睛虹膜會通過調節瞳孔大小來控制進入眼睛的光線強度，然後經過水晶體的折射，在眼底視網膜上形成影像。大腦會把視感神經所感知的信息記憶下來。

景物的光線進入相機以後，就會受到光圈的調節控制，會被鏡片折射會聚成影像，並將影像投射在感光材料上。於是，影像就被記錄下來了。

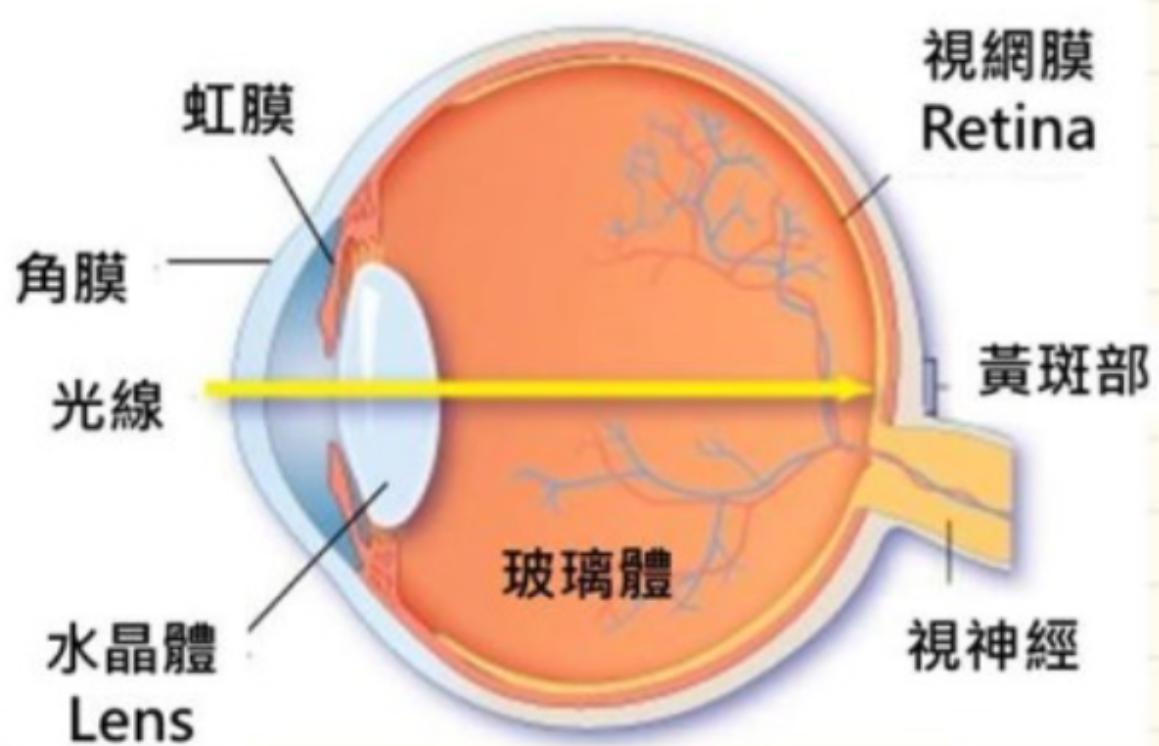


眼睛 V.S 相機





眼睛 V.S 相機



那如何在 p5.js 載入圖像呢？

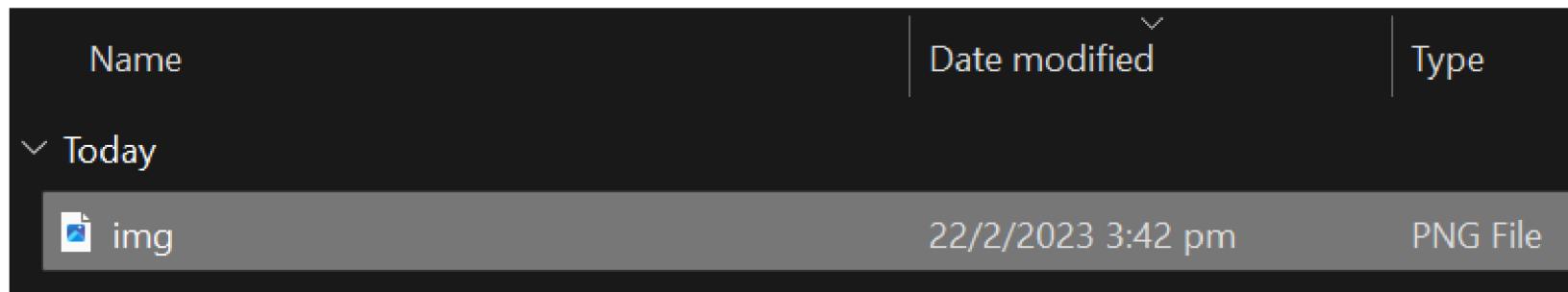
答案：使用 p5.Image loadImage() 載入圖片

loadImage() 載入圖片 這個函數的 Syntax 句法

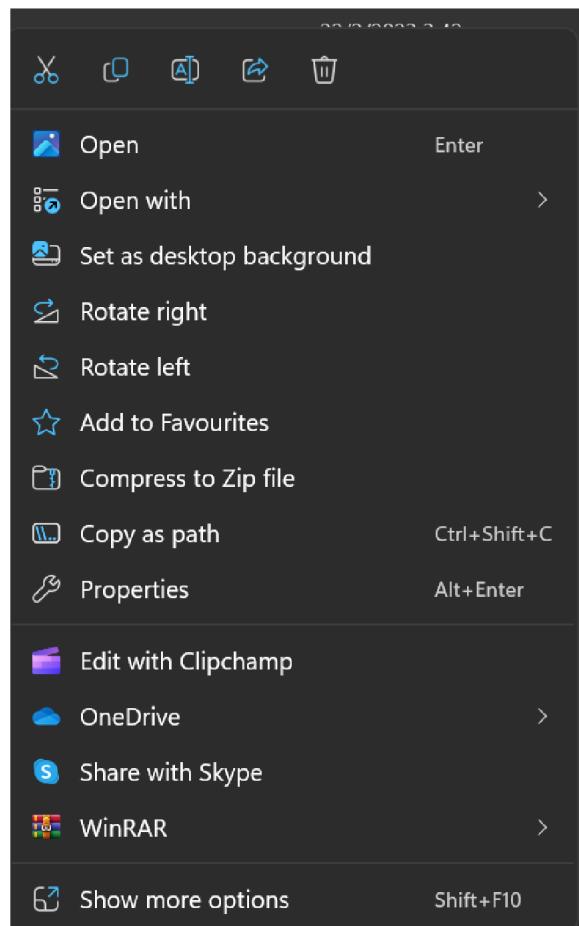
loadImage(圖片檔案路徑)

圖片檔案路徑 需要使用 String 字串 這個 Data type 資料類型
所以圖片檔案路徑需要放在 " " 內

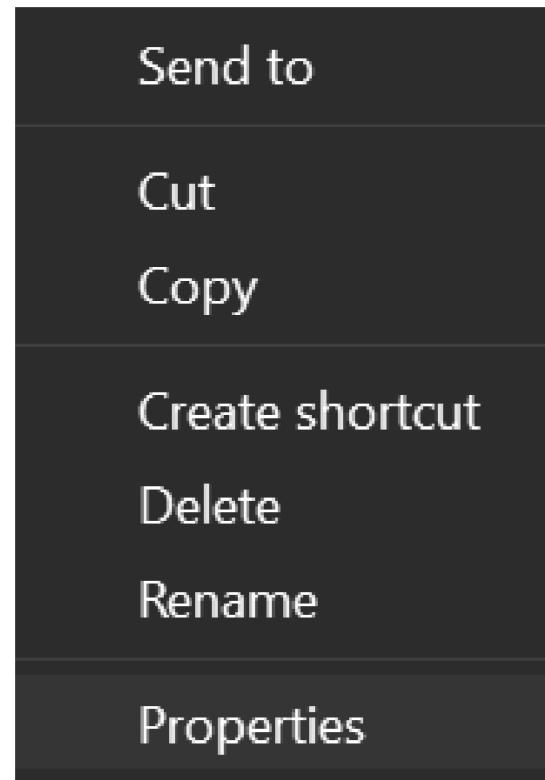
** 如需要知道圖片像素大小請看下一頁 ~



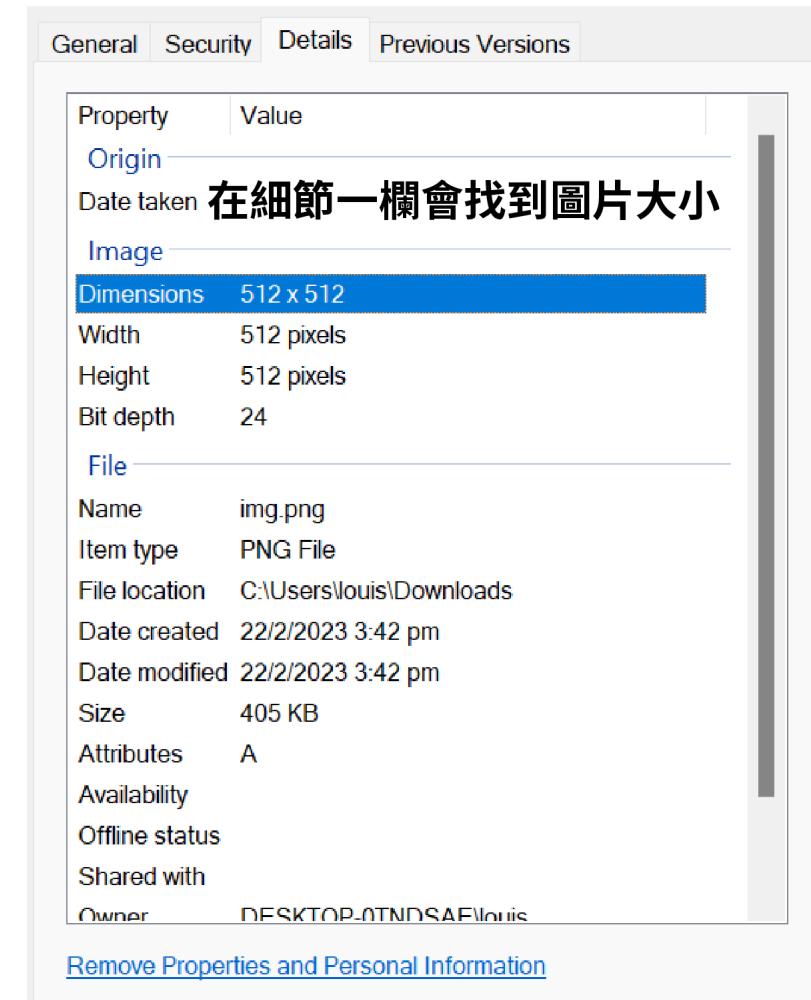
1 先下載圖片檔案



2 在檔案
按下滑鼠右鍵



3 在選單按下內容



2.1 loadImage

```
let img;  
  
function preload() {  
  img = loadImage("img.png");  
}  
  
function setup() {  
  createCanvas(512, 512);  
  image(img, 0, 0);  
}
```

** 如忘記如何上載圖片請看下一頁 ~



不明白image();內的數值有甚麼意義？

image(圖片的變量，圖片左上角的x座標，圖片左上角的y座標);

image(圖片的變量，圖片左上角的x座標，圖片左上角的y座標，圖片右上角的x座標，圖片右上角的y座標);

Sketch Files

index.html

sketch.js

style.css

Sketch Files

Create folder

Create file

Upload file

按此打開文件欄 click here to open Sketch Files

Upload File

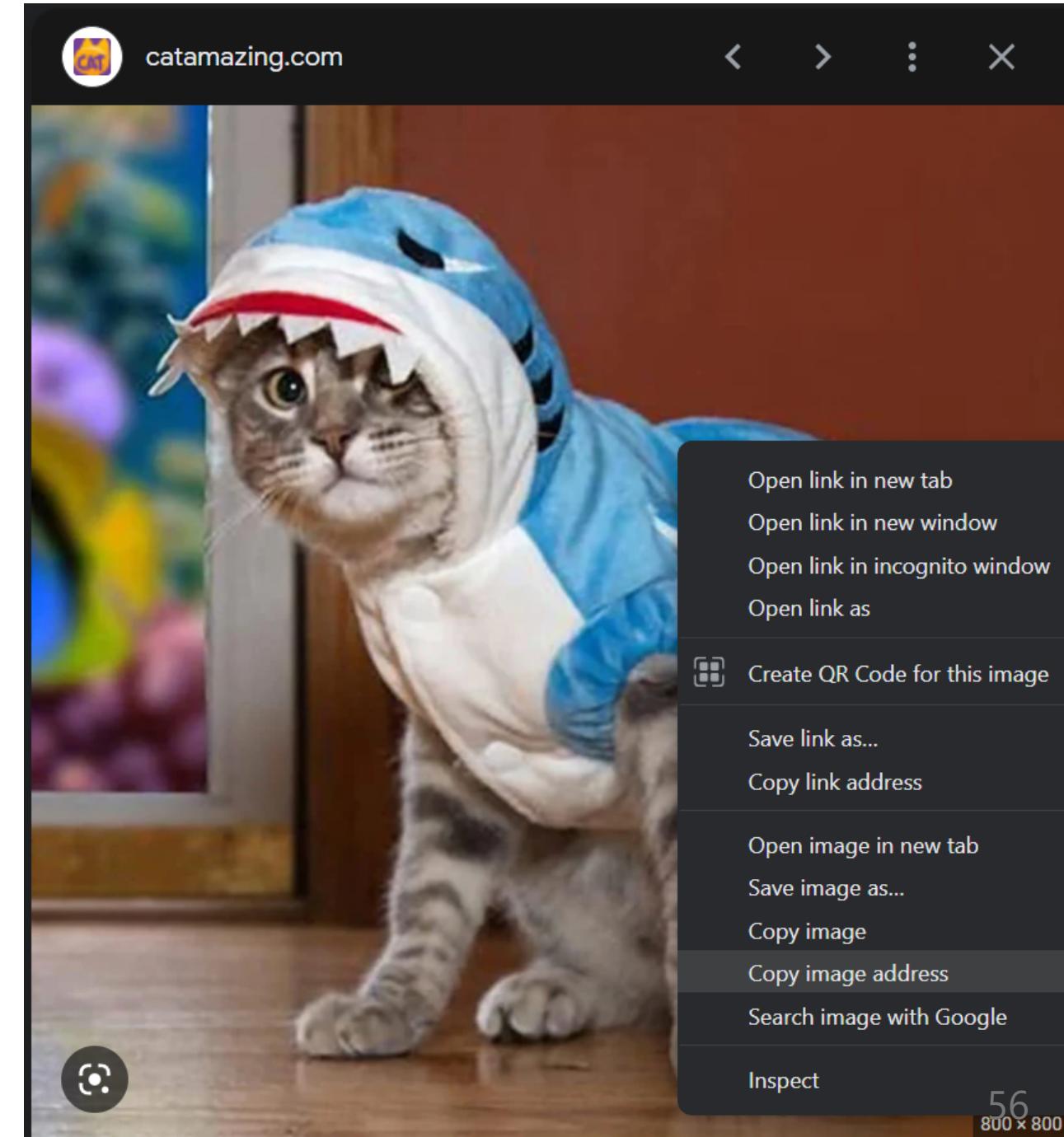
Drop files here or click to use the file browser

按此上載文件(聲音/圖片) click here to upload files (sound/image)

我們也可以使用網上相 片連結載入圖片

在搜尋器的目標圖片上按
下滑鼠右鍵，在選單中選取

複製圖片連結



2.2 loadImage_Link

```
let img;

function preload() {
  img = loadImage("https://cdn.shopify.com/s/files/1/1832/0821/files/catshark.jpg?v=1649869148");
}

function setup() {
  createCanvas(512,512);
  image(img, 0, 0,width,height);
}
```



有關 width 與 height

width 是一種系統變量， mouseX 包含顯示框當前的闊度大小

height 同樣是系統變量， mouseY 包含顯示框當前的高度大小

!! 比 setup() 更先執行 !!

preload()

preload()函數執行一次 {} 內包含的
編碼

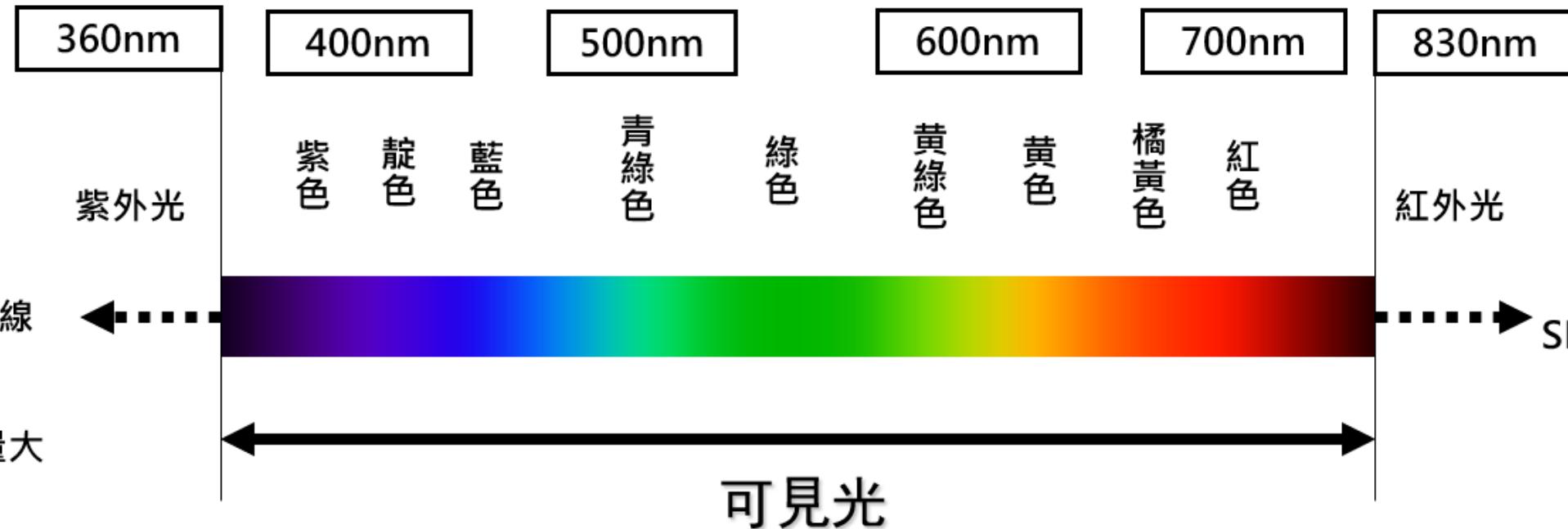
在 preload() {} 內的編碼會執行在
setup() 之前

```
1 let img;  
2  
3 function preload() {  
4   img = loadImage("img.png");  
5 }  
6  
7 function setup() {  
8   createCanvas(512,512);  
9   image(img, 0, 0);  
10 }
```

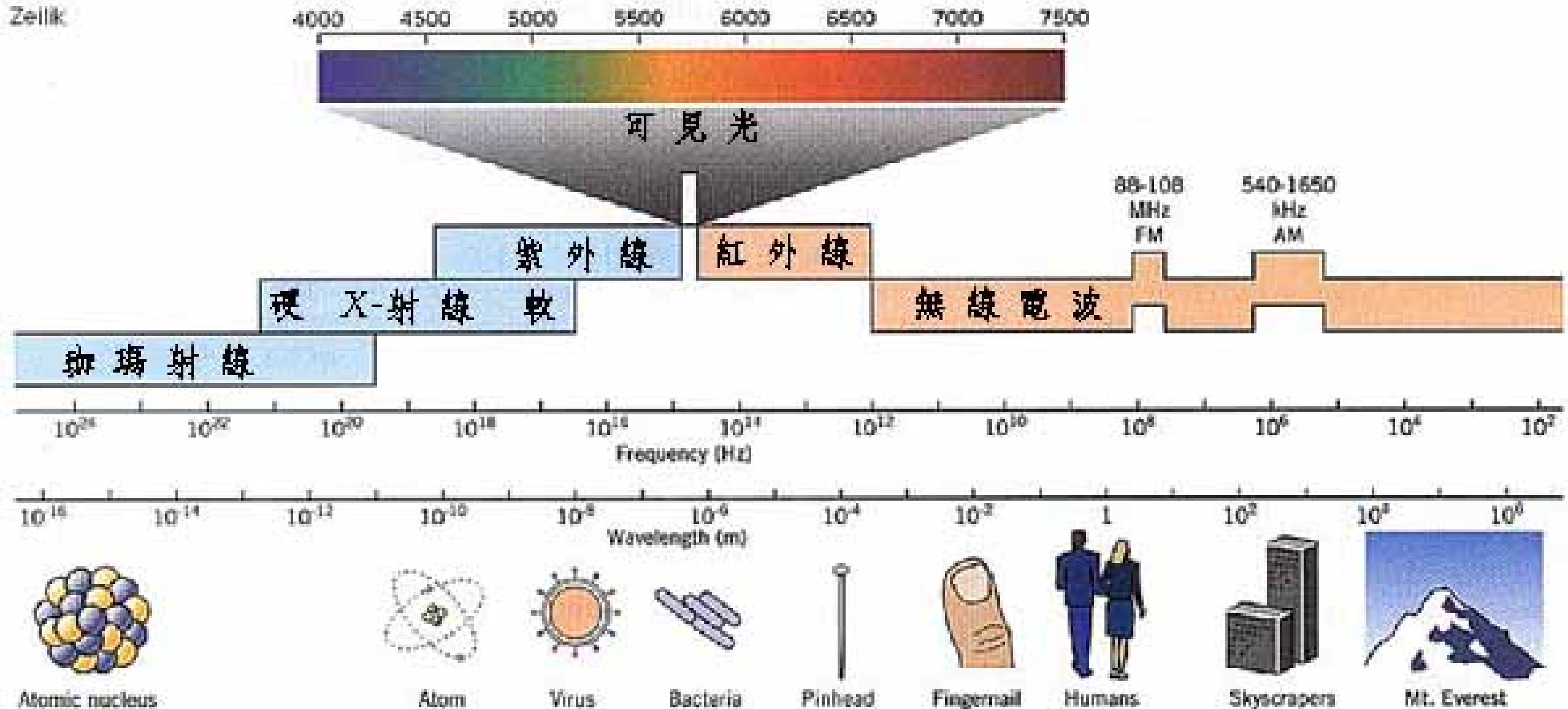
記得光學三原色嗎？

其實光譜不單只有「三原色」的紅、綠和藍。

在「可見光」外還有更廣闊的光譜！



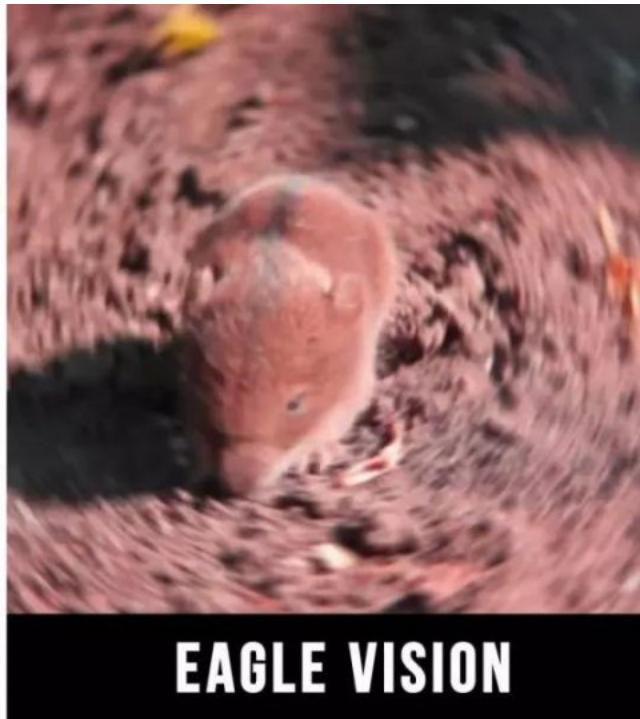
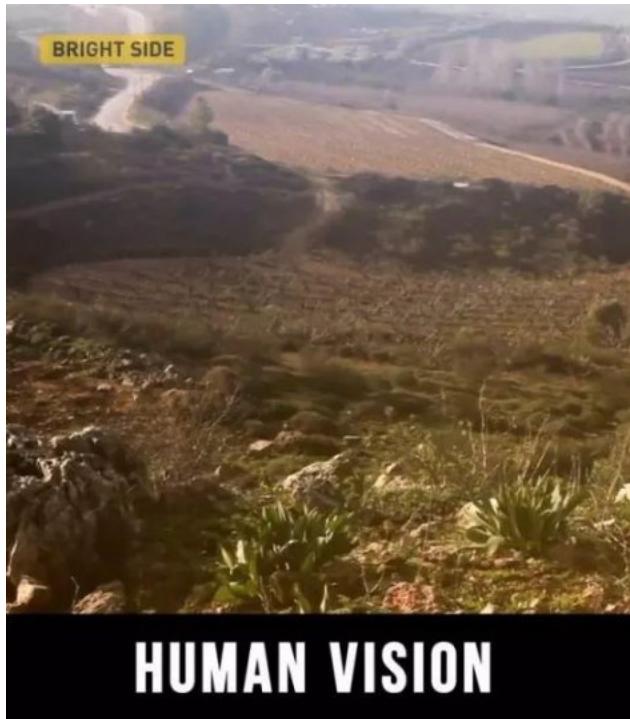
我們再對比一下電磁波譜與可見光



有趣的知識 !!

不同動物與人類看到的世界也大有不同呵~

鳥類在視覺上有相似之處，不同於人類，它們可以分辨紫外線。由於它們的眼睛結構，鳥類可以將視線聚焦在某些地方。例如像獵鷹和鷹這樣的猛禽，它可以將視線聚焦在局部地區。你可能只看到一塊田地，但獵鷹可以把注意力集中在那裏出現的小動物上。



昆蟲擁有成千上萬個視覺受體，借此共同創造廣闊的視野。他們看到世界在緩慢運動，也可以看到紫外線。



每個不同的視覺感官世界也值得尊重呵~

我們也試一下使用濾鏡，
改變影像效果。

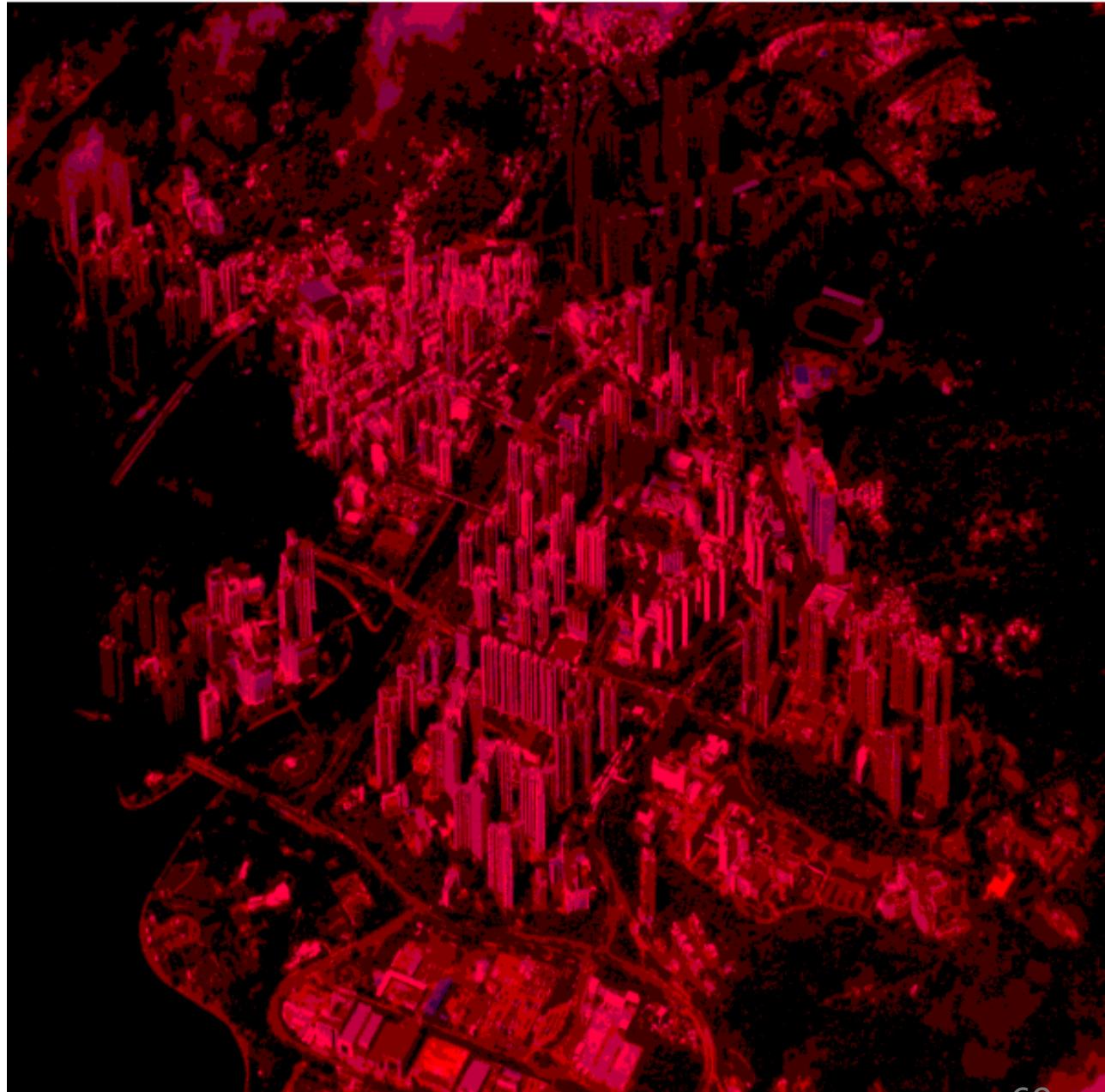


2.3 imageFilter

```
let img;

function preload(){
  img = loadImage('TaiPo_Image.jpg');
}

function setup(){
  createCanvas(400,400);
  background(220);
  tint('#ED225D');
  image(img,0,0,400,400);
  filter(PPOSTERIZE, 5);
}
```



不明白tint();內的數值有甚麼意義？

tint(red , green , blue);

tint(顏色代碼);

不明白filter();內的數值有甚麼意義？

<https://p5js.org/reference/#/p5/filter>

留意 !!

tint() 需要寫在 image() 上方

filter() 需要寫在 image() 下方

```
tint('#ED225D');  
image(img, 0, 0, 400, 400);  
filter(POTERIZE, 5);
```

現在試一下在 p5.js 使用網絡攝影機！

2.4 videoCapture

```
let capture;

function setup() {
  createCanvas(320, 240);
  capture = createCapture(VIDEO);
  capture.size(320, 240);
  capture.hide();
}

function draw() {
  image(capture, 0, 0, 320, 240);
}
```



createCapture(VIDEO)

先宣告 capture 變量，在 setup() 中使用 createCapture()

createCapture(類型) 類型普遍會是
聲音 AUDIO 或影片 VIDEO

createCapture(VIDEO)在HTML上創
造一個影片類型的元素，在網絡攝
影機中取得影像數據。

```
let capture;

function setup() {
  createCanvas(320, 240);
  capture = createCapture('VIDEO');
  capture.size(320, 240);
  capture.hide();
}

function draw() {
  image(capture, 0, 0, 320, 240);
}
```

`capture.size(影片闊度, 影片高度)`

`capture.hide()` 隱藏HTML內的影像元素，免去重複顯示影像。

在 `draw()` 內使用 `image()` 更新攝影影像

```
let capture;

function setup() {
  createCanvas(320, 240);
  capture = createCapture(VIDEO);
  capture.size(320, 240);
  capture.hide();
}

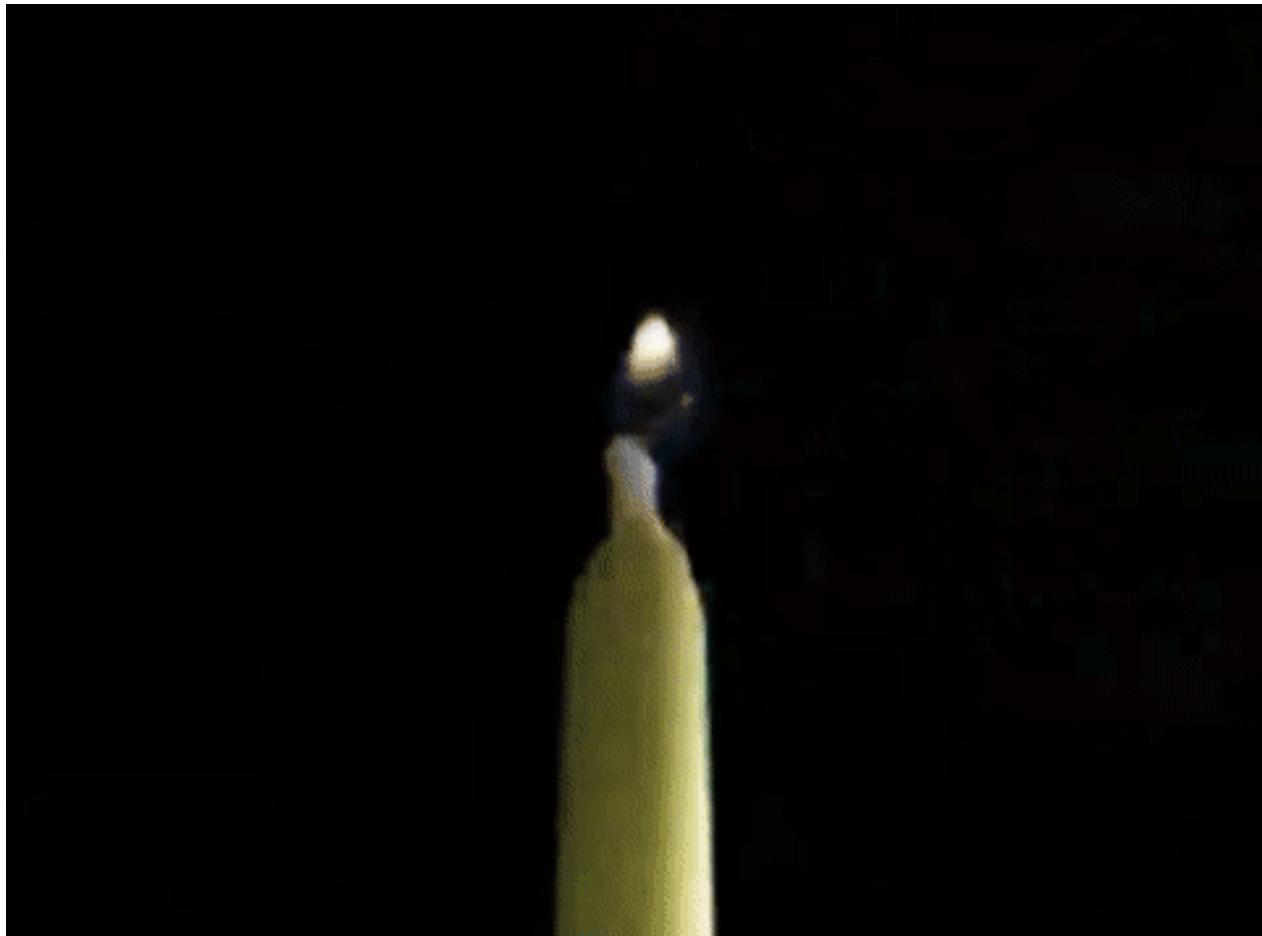
function draw() {
  image(capture, 0, 0, 320, 240);
}
```

2.5 loadVideo

```
let video;

function setup() {
  createCanvas(320, 240);
  video = createVideo("light.mp4");
  video.size(320, 240);
  video.volume(0);
  video.loop();
  video.hide();
}

function draw() {
  let img = video.get();
  image(img, 0, 0);
}
```



createVideo(影片檔案路徑)

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

video.size(影片闊度， 影片高度)

video.volume(影片聲音大小) 0為無聲 1為最大聲

video.loop() 重複播放影片

video.hide() 隱藏HTML內的影像元素，免去重複顯示影像。

```
1 let video;
2
3 function setup() {
4   createCanvas(320, 240);
5   video = createVideo("light.mp4");
6   video.size(320, 240);
7   video.volume(0);
8   video.loop();
9   video.hide();
10 }
11
12 function draw() {
13   let img = video.get();
14   image(img, 0, 0);
15 }
```

draw() { }部份

在draw() 內宣告 img 變量

img 儲存在 video.get() 取得影像數據

使用 image() 把 img 內的影格循環更新畫面

```
1 let video;
2
3▼ function setup() {
4  createCanvas(320, 240);
5  video = createVideo("light.mp4");
6  video.size(320, 240);
7  video.volume(0);
8  video.loop();
9  video.hide();
10 }
11
12▼ function draw() {
13  let img = video.get();
14  image(img, 0, 0);
15 }
```

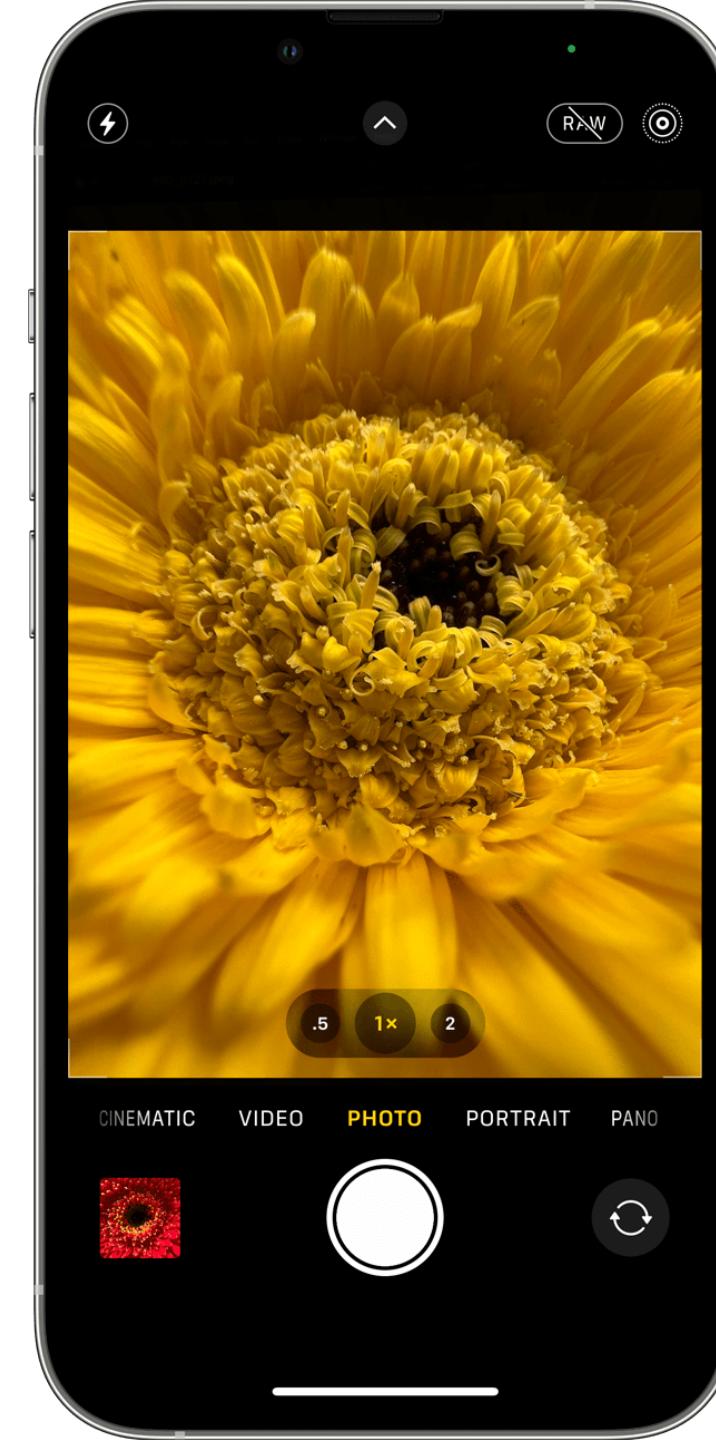
接下來 做個攝影軟件吧!



攝影軟件

可以分解為~

1. 顯示鏡頭捕捉的畫面
2. 拍攝按鈕
3. 儲存相片



2.6 button

```
let captureButton;

function setup() {
  createCanvas(640, 360);
  captureButton = createButton("draw");
  captureButton.mousePressed(drawEllipse);
  background(0);
}

function drawEllipse() {
  fill(random(0,255),
    random(0,255),
    random(0,255));
  ellipse(random(0,width),
    random(0,height),30,30);
}
```



先宣告 captureButton 變量

createButton()

createButton("按鈕的名稱");

captureButton.mousePressed

當滑鼠按下按鈕 -> 執行 custom
function 自訂函式 drawEllipse

```
let captureButton;

function setup() {
  createCanvas(640, 360);
  captureButton = createButton("draw");
  captureButton.mousePressed(drawEllipse);
  background(0);
}
```

custom function

自訂函式 drawEllipse

每次按下按鈕才會執行一次

```
function drawEllipse() {  
    fill(random(0,255), random(0,255), random(0,255));  
    ellipse(random(0,width), random(0,height), 30, 30);  
}
```

更多有關 custom function 自訂函式

創建自訂函式能夠將編碼組織成較小的模塊，並將複雜的任務視為一個單一的步驟。

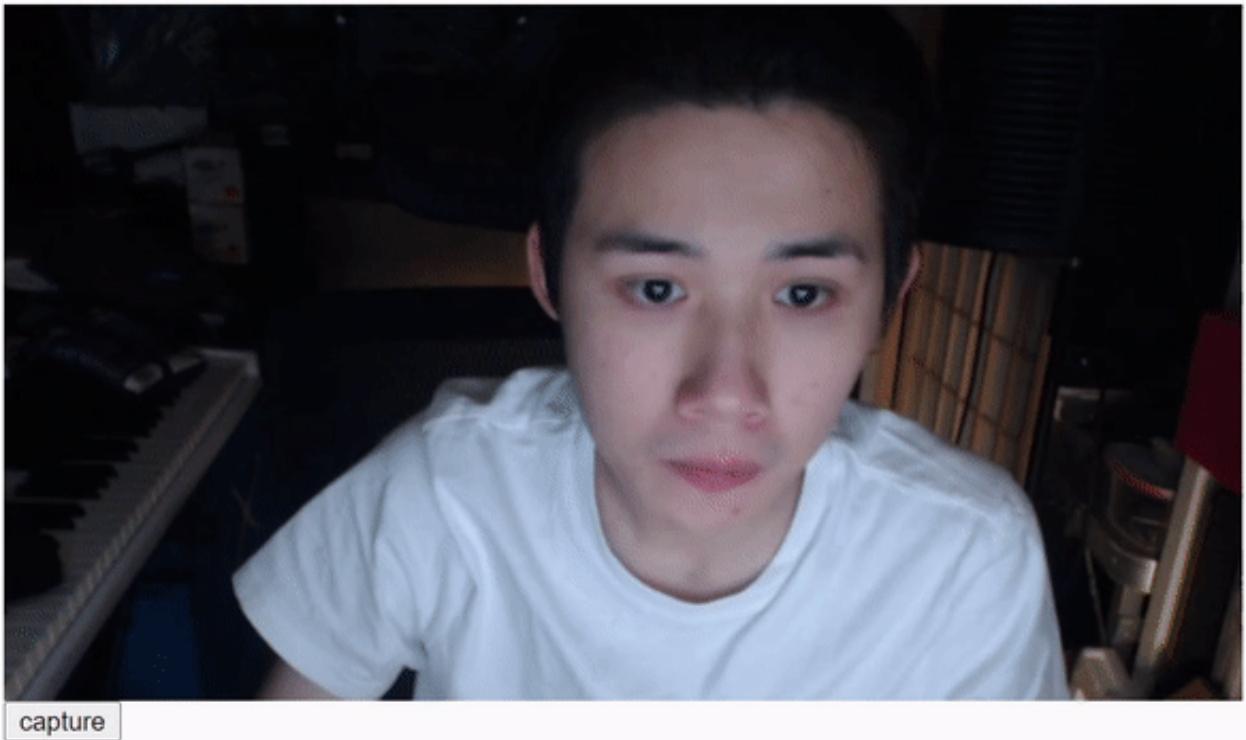
2.7 buildYourCamera

```
let capture;
let captureButton;

function setup() {
  createCanvas(640, 360);
  capture = createCapture(VIDEO);
  captureButton = createButton("capture");
  captureButton.mousePressed(takePicture);
  capture.hide();
}

function draw() {
  background(0);
  image(capture, 0, 0);
}

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



**來使用 image(), createCapture(), createVideo() 及更多
學到的**

發揮你的創意吧 ~!

Welcome to
Digital Senses 數位通感

三) 聲音：自製錄音及聲音載入軟件

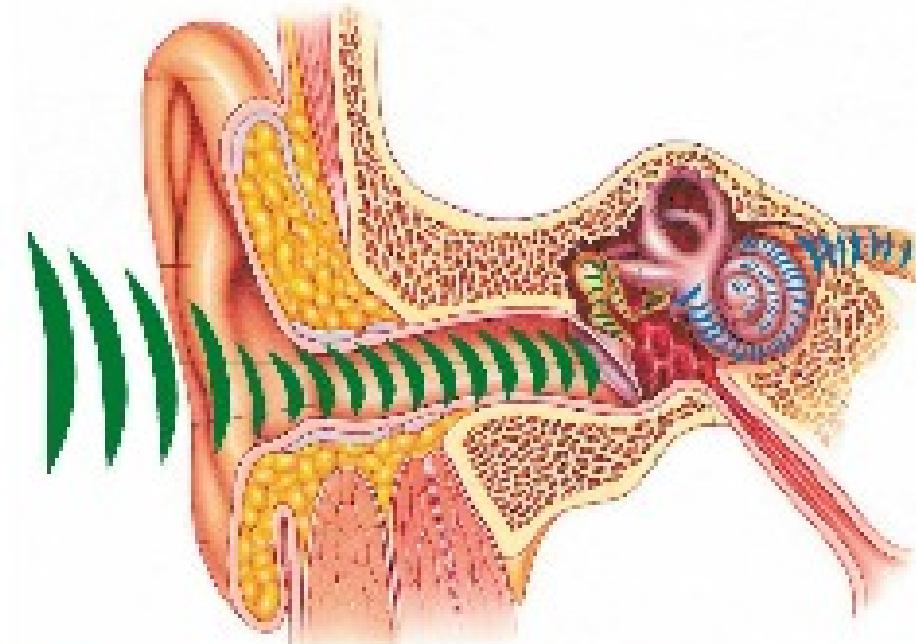
(3)

聲音

聲音是？

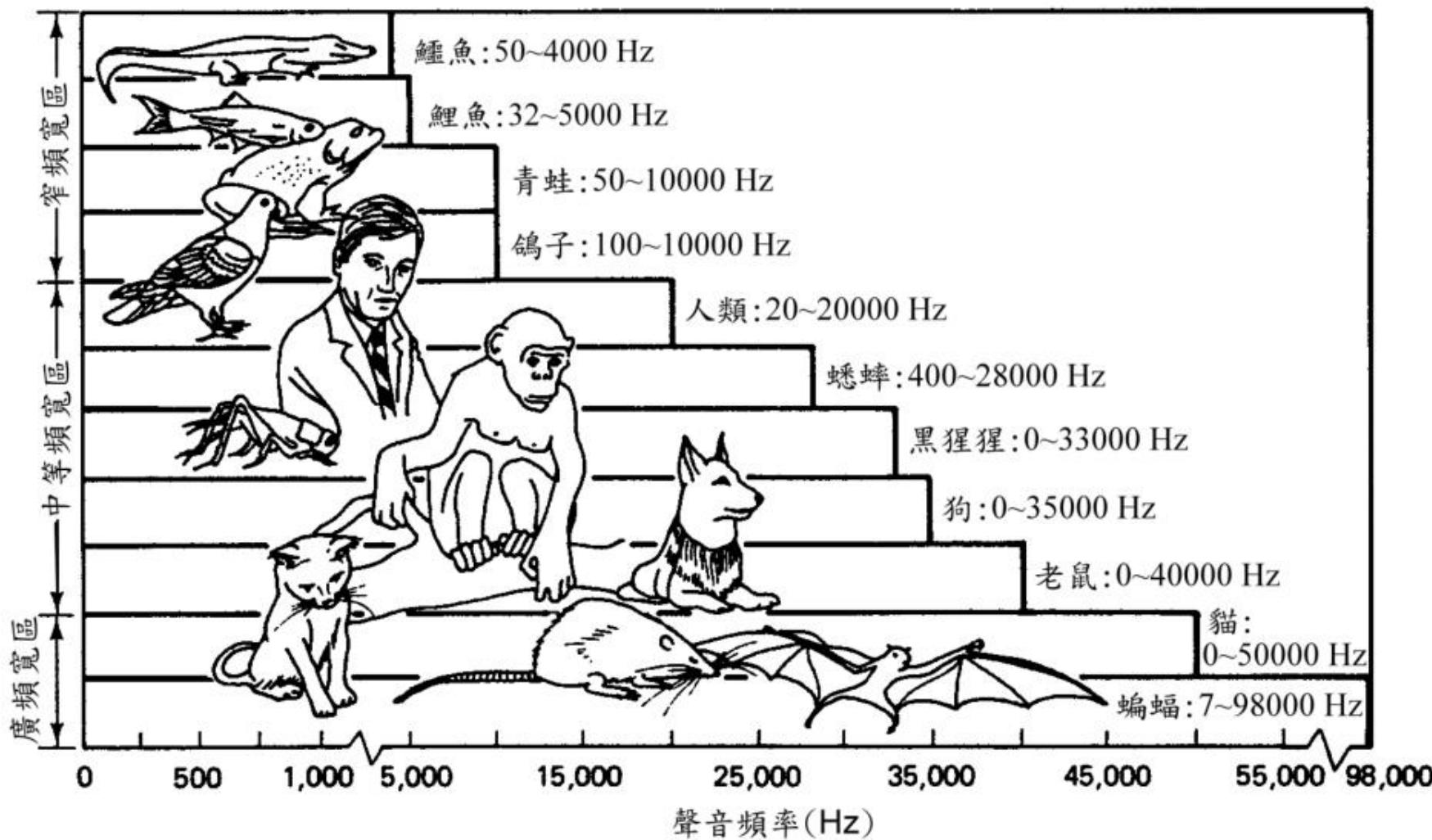
聲音是振動產生的聲波，通過介質（氣體、固體、液體）傳播並能被人或動物聽覺器官所感知的波動現象。

聲音的頻率一般會以赫茲表示，記為Hz，指每秒鐘週期性震動的次數。而分貝是用來表示聲音強度的單位，記為dB。



綠色代表聲波以震動能傳遞到耳膜及聽小骨
藍色代表由內耳轉換為電波能
再經由聽神經傳達至大腦聽覺區

不同動物可聽見之聲音頻率範圍



那如何在 p5.js 載入音頻呢？

答案：使用 p5.sound loadSound() 載入聲音檔案

3.1 loadSound

```
let mySound;  
  
function preload() {  
    mySound = loadSound("Summer.mp3");  
}  
  
function setup() {  
    createCanvas(500, 500);  
    background(0);  
    mySound.play();  
}
```

** 如忘記如何上載聲音請看下一頁 ~

Sketch Files

index.html

sketch.js

style.css

Sketch Files

Create folder

Create file

Upload file

按此打開文件欄 click here to open Sketch Files

Upload File

Drop files here or click to use the file browser

按此上載文件(聲音/圖片) click here to upload files (sound/image)

loadSound(聲音檔案路徑)

先宣告 mySound 變量，在 setup() 中使用 loadSound()

video.play() 播放聲音檔案

```
let mySound;  
  
function preload() {  
    mySound = loadSound("Summer.mp3");  
}  
  
function setup() {  
    createCanvas(500, 500);  
    background(0);  
    mySound.play();  
}
```

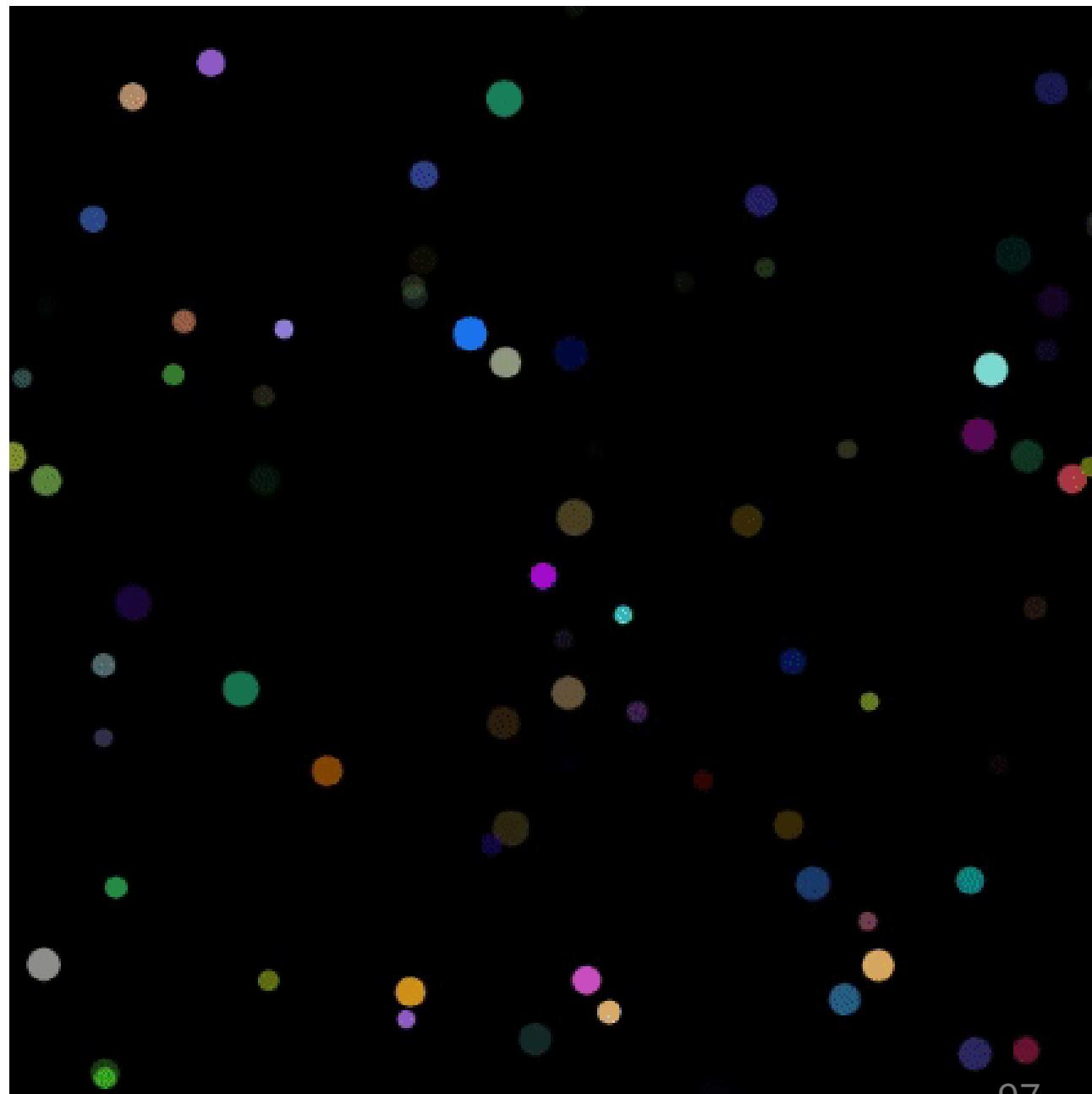
3.2 loadSound2

```
let mySound;

function preload() {
  mySound = loadSound("Summer.mp3");
}

function setup() {
  createCanvas(300, 300);
  background(0);
  noStroke();
  mySound.loop();
}

function draw(){
  fill(random(255),random(255),random(255),random(255));
  ellipse(random(0,width),random(0,height),random(5,10));
}
```



3.2 loadSound2

```
let mySound;

function preload() {
    mySound = loadSound("Summer.mp3");
}

function setup() {
    createCanvas(300, 300);
    background(0);
    noStroke();
    mySound.loop();
}

function draw(){
    fill(random(255),random(255),random(255),random(255));
    ellipse(random(0,width),random(0,height),random(5,10));
}
```

那如何與聲音檔案互動？

那要先學習 Conditional 條件運算式 - if 的結構

3.3 conditional_if

```
let number1 = 5;

function setup() {
  createCanvas(500,500);
  background(0);
  if (number1 > 1) {
    ellipse(250,250,50,50);
  }
}
```

> 大於 Greater than

< 小於 Less than

= = 相等 Equality

宣告及定義 number 1 等於整數數值 5

這個例子在 setup() 內

用 if(需要達成的條件)

如條件達成就會執行 {} 內的編碼

在這個例子中 條件是

如果 number 1 大過 > 1 的話

就會運行 ellipse();

```
let number1 = 5;

function setup() {
  createCanvas(500,500);
  background(0);
  if (number1 > 1) {
    ellipse(250,250,50,50);
  }
}
```

學會使用 if condition 後

那就可以加上 p5.sound 內的 p5.Amplitude(); 來與聲音檔案進行互動

3.4 amplitude

```
let mySound;
let amp, ampLevel;

function preload() {
  mySound = loadSound("Summer.mp3");
}

function setup() {
  createCanvas(500, 500);
  background(0);
  noStroke();
  mySound.loop();
  amp = new p5.Amplitude();
}

function draw() {
  ampLevel = amp.getLevel();
  print(ampLevel);
  if (ampLevel > 0.15) {
    fill(random(255), random(255), random(255), random(255));
    ellipse(random(0, width), random(0, height), random(5, 10));
  }
}
```



3.4 amplitude

```
let mySound;
let amp, ampLevel;

function preload() {
  mySound = loadSound("Summer.mp3");
}

function setup() {
  createCanvas(500, 500);
  background(0);
  noStroke();
  mySound.loop();
  amp = new p5.Amplitude();
}

function draw() {
  ampLevel = amp.getLevel();
  print(ampLevel);
  if (ampLevel > 0.15) {
    fill(random(255), random(255), random(255), random(255));
    ellipse(random(0, width), random(0, height), random(5, 10));
  }
}
```

宣告 mySound, amp 及 ampLevel

amp 及 ampLevel 接下來在 setup() 及 draw() 會用上

在 preload() 內

使用 loadSound(); 載入聲音檔案

```
let mySound;  
let amp, ampLevel;  
  
function preload() {  
    mySound = loadSound("Summer.mp3");  
}
```

在 setup() 內

把 amp 定義為 新的(new)
p5.Amplitude();

這個步驟名為 instantiate

使用 .loop(); 令聲音檔案循環播放

```
function setup() {  
    createCanvas(500, 500);  
    background(0);  
    noStroke();  
    mySound.loop();  
    amp = new p5.Amplitude();  
}
```

在 draw() 內

在 ampLevel 這個變量中儲存 在amp
內使用.getLevel(); 取得的數值

.getLevel(); 會取得聲音檔案中當
前播放的聲音振幅

最小振幅為0， 最大振幅為1

```
function draw() {  
    ampLevel = amp.getLevel();  
    print(ampLevel);  
    if (ampLevel > 0.15) {  
        fill(random(255), random(255), random(255), random(255));  
        ellipse(random(0, width), random(0, height), random(5, 10));  
    }  
}
```

在 draw() 內

如果振幅大於 0.15

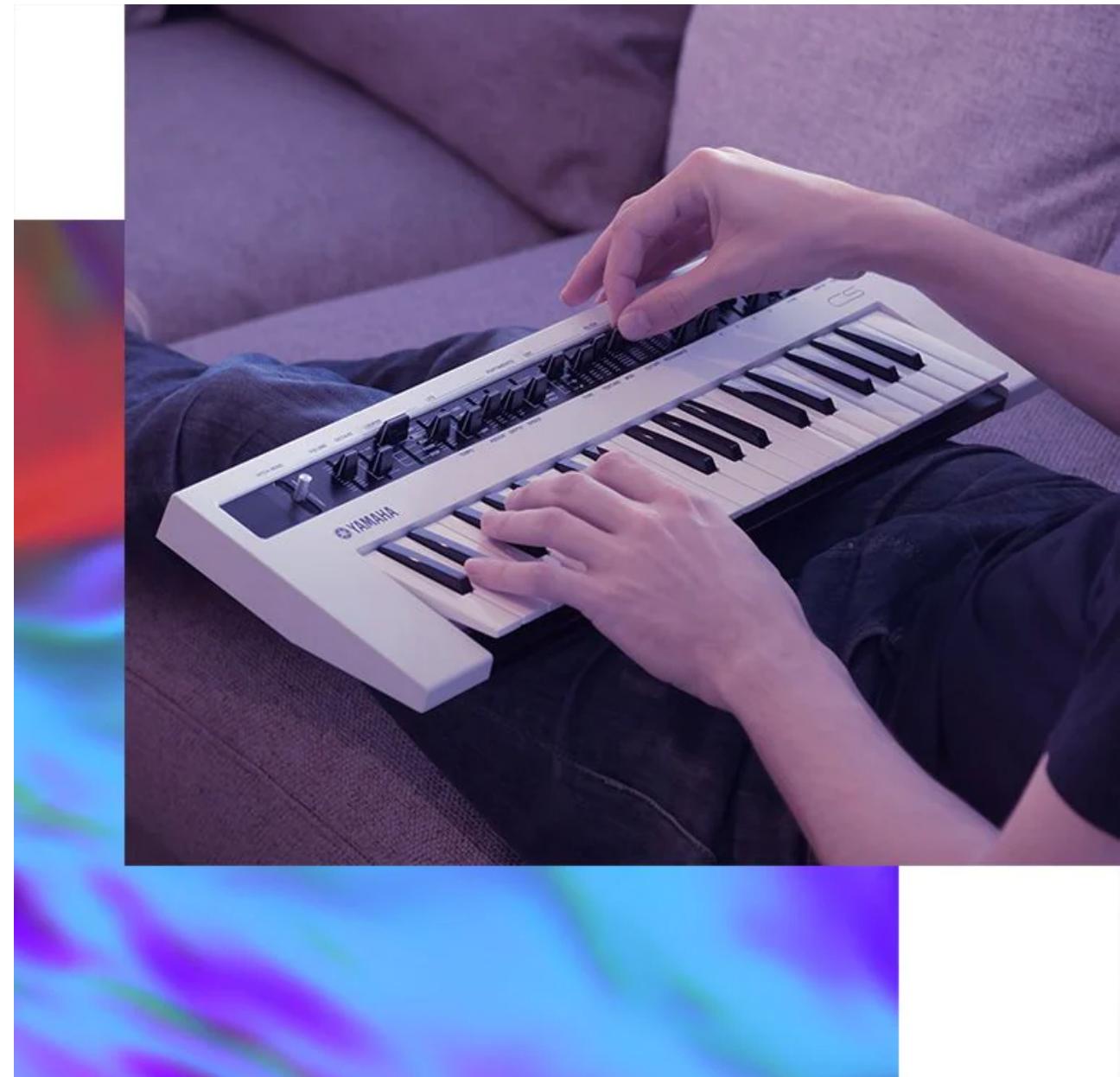
if(ampLevel > 0.15)

才會在顯示框中顯示隨機顏色及位置的圓形

```
function draw() {
    ampLevel = amp.getLevel();
    print(ampLevel);
    if (ampLevel > 0.15) {
        fill(random(255), random(255), random(255), random(255));
        ellipse(random(0, width), random(0, height), random(5, 10));
    }
}
```

除了使用聲音檔案播放
聲音

試一下在 p5.sound 中
使用 Synth 合成器



3.5 monoSynth

```
let monoSynth;

function setup() {
  createCanvas(400, 400);
  background(220);
  monoSynth = new p5.MonoSynth();
}

function mousePressed() {
  userStartAudio();
  monoSynth.play("C4", 1, 0, 0.25);
}
```

宣告變量 monoSynth

在 setup() 內

把 monoSynth 定義為 新的(new)
p5.MonoSynth();

```
let monoSynth;

function setup() {
  createCanvas(400, 400);
  background(220);
  monoSynth = new p5.MonoSynth();
}

function mousePressed() {
  userStartAudio();
  monoSynth.play("C4", 1, 0, 0.25);
}
```

使用 `function mousePressed()`

當用戶按下滑鼠後才會運行
`function mousePressed() {}` 內的編碼

使用 `userStartAudio()` 告訴瀏覽器
用戶要使用聲音內容

```
let monoSynth;

function setup() {
  createCanvas(400, 400);
  background(220);
  monoSynth = new p5.MonoSynth();
}

function mousePressed() {
  userStartAudio();
  monoSynth.play("C4", 1, 0, 0.25);
}
```

不明白monoSynth.play();內的數值有甚麼意義？

monoSynth.play(note 音符 , velocity 音符力度 , secondsFromNow(attack) 觸發時間 , sustainTime(sustain) 維持的時間);

現在來試一下使用 p5.AudioIn();

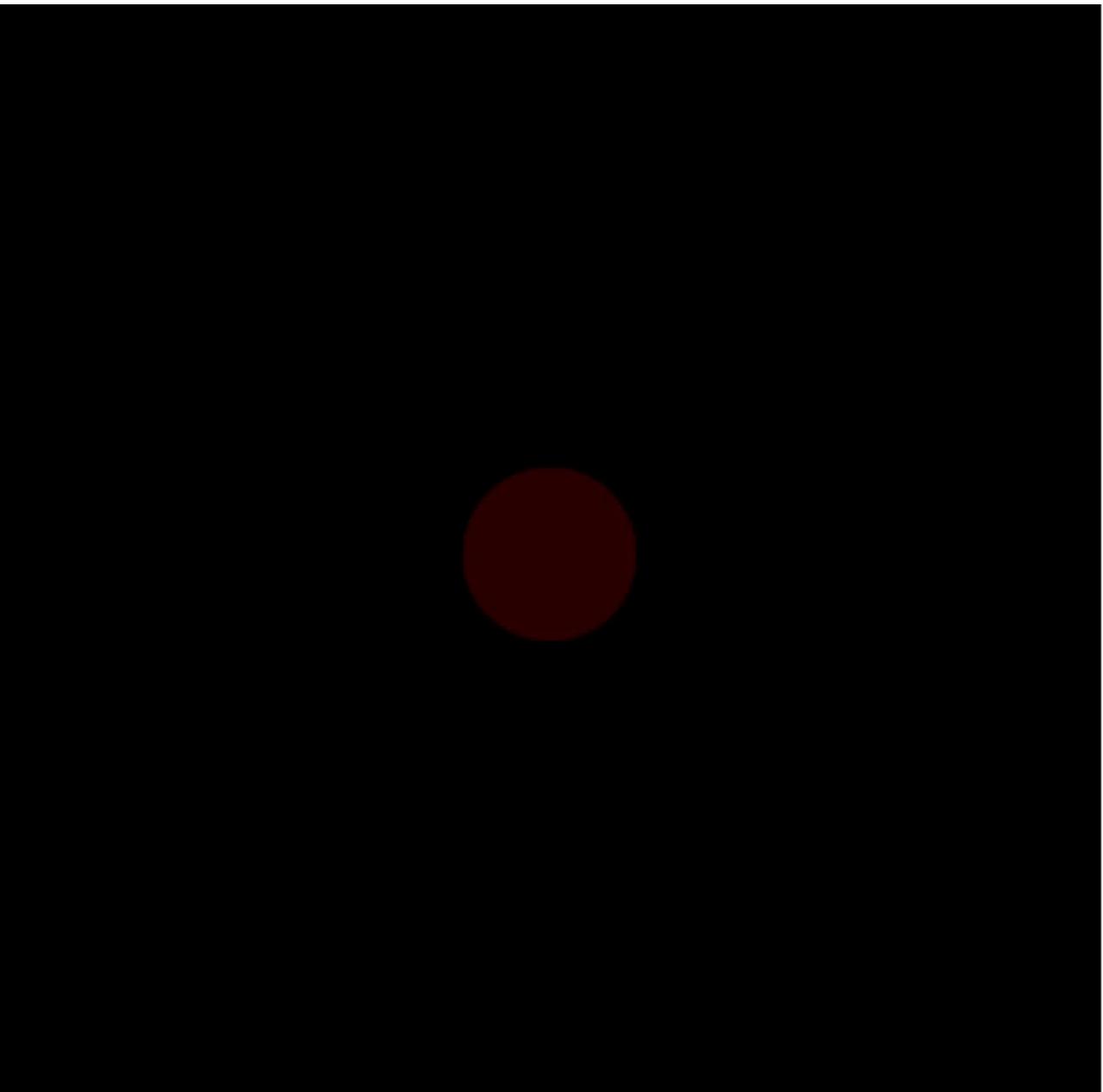
我們可以使用 網絡攝影機 中的 收音咪

3.6 useMic

```
let mic;
let vol;

function setup() {
  createCanvas(400, 400);
  mic = new p5.AudioIn();
  mic.start();
}

function draw() {
  background(0);
  vol = mic.getLevel();
  fill(vol * 255, 0, 0);
  ellipse(width / 2, height / 2, vol * 400, vol * 400);
}
```



3.6 useMic

```
let mic;
let vol;

function setup() {
  createCanvas(400, 400);
  mic = new p5.AudioIn();
  mic.start();
}

function draw() {
  background(0);
  vol = mic.getLevel();
  fill(vol * 255, 0, 0);
  ellipse(width / 2, height / 2, vol * 400, vol * 400);
}
```

宣告變量 mic 及 vol

在 setup() 內

把 mic 定義為 新的(new)
p5.AudioIn();

這個步驟令用戶可以使用網絡攝影機
中的收音咪

```
let mic;
let vol;

function setup() {
  createCanvas(400, 400);
  mic = new p5.AudioIn();
  mic.start();
}
```

在 draw() 內

在變量 vol 中儲存於 mic.getLevel();
取得的聲音音量數值

最大音量為1， 最小音量為0

把聲音振幅使用在 fill(); 及
ellipse(); 中達致互動效果

利用 * 把 vol 數值分別倍大255及
400倍， 達到可見的效果。

```
function draw() {  
    background(0);  
    vol = mic.getLevel();  
    fill(vol * 255, 0, 0);  
    ellipse(width / 2, height / 2, vol * 400, vol * 400);  
}
```

+ 相加 Addition

- 相減 Subtraction

* 相乘 Multiplication

/ 相除 Division

% 餘數 Remainder

聲音作為與時間相關的媒介

我們應該如何在 p5.js 中得到時間相關的數值？

3.7 counter

```
let time = 0;

function setup(){
  createCanvas(500,500);
  frameRate(5);
}

function draw() {
  background(0);
  time = time + 1;
  fill(255,0,0);
  text(time,width/2,height/2-50);
  fill(0,255,0);
  text(frameCount,width/2,height/2+50);
}
```

141

141

3.7 counter

```
let time = 0;

function setup(){
    createCanvas(500,500);
    frameRate(5);
}

function draw() {
    background(0);
    time = time + 1;
    fill(255,0,0);
    text(time,width/2,height/2-50);
    fill(0,255,0);
    text(frameCount,width/2,height/2+50);
}
```

先宣告變量 time 及定義其為 整數 0

在 draw() 內

再定義 time 等 time 自身 加上 1

這樣做可以令 time 在每次循環中比
上一次循環的數值大上 1

未經過 draw() 循環 time = 0

第一次經過循環 time = 0 + 1;

在第一次經過循環結束時 time = 1

第二次經過循環 time = 1 + 1;

在第二次經過循環結束時 time = 2

第三次經過循環 time = 2 + 1;

第三次經過循環 time = 3;

```
let time = 0;
```

```
function draw() {  
    background(0);  
    time = time + 1;  
}
```

不明白frameCount有甚麼意義？

frameCount是系統變量，frameCount包含現在程式總共運行了多少個 iteration
迭代 (簡單可以理解為程式反覆運算了多少次)

下一步 使用frameCount生成音樂吧！

3.8 counterWithSynth

```
let monoSynth;
let note;

function setup() {
  createCanvas(400, 400);
  background(220);
  monoSynth = new p5.MonoSynth();
}

function draw() {
  //C major - Scale
  note = random(["C4", "D4", "E4", "F4", "G4", "A4", "B3"]);

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

為什麼音符會隨機生成？

```
note = random(["C4", "D4", "E4", "F4", "G4", "A4", "B3"]);
```

使用 `random()`; 隨機抽取在 [] 內的字串到

`monoSynth.play();` 內的 `note` 中使用

試一下融合早前課堂中學到的所有編碼

把聲音視覺化



3.9 simpleVisualizer

```
let monoSynth;
let note;
let img0, img1, img2, img3, img4, img5, img6, img7;

function preload() {
  img0 = loadImage("0.png");
  img1 = loadImage("1.png");
  img2 = loadImage("2.png");
  img3 = loadImage("3.png");
  img4 = loadImage("4.png");
  img5 = loadImage("5.png");
  img6 = loadImage("6.png");
  img7 = loadImage("7.png");
}
```

3.9 simpleVisualizer

```
function setup() {
  createCanvas(400, 400);
  background(0);
  monoSynth = new p5.MonoSynth();
}

function draw() {

  //C major - Scale
  note = random(["C4", "D4", "E4", "F4", "G4", "A4", "B4", "C5"]);
}
```

3.9 simpleVisualizer

```
switch (note) {  
    case "C4":  
        image(img0, 0, 0, 400, 400);  
        break;  
    case "D4":  
        image(img1, 0, 0, 400, 400);  
        break;  
    case "E4":  
        image(img2, 0, 0, 400, 400);  
        break;  
    case "F4":  
        image(img3, 0, 0, 400, 400);  
        break;  
    case "G4":  
        image(img4, 0, 0, 400, 400);  
        break;  
    case "A4":  
        image(img5, 0, 0, 400, 400);  
        break;  
    case "B4":  
        image(img6, 0, 0, 400, 400);  
        break;  
    case "C5":  
        image(img6, 0, 0, 400, 400);  
        break;  
    }  
}
```

不明白switch(變數名稱或運算式)是甚麼？

switch 條件判斷適合用於需要得知 一個變量 等於 多個數值 的時候使用

switch(變數名稱或運算式)

case 需要達成的條件 :

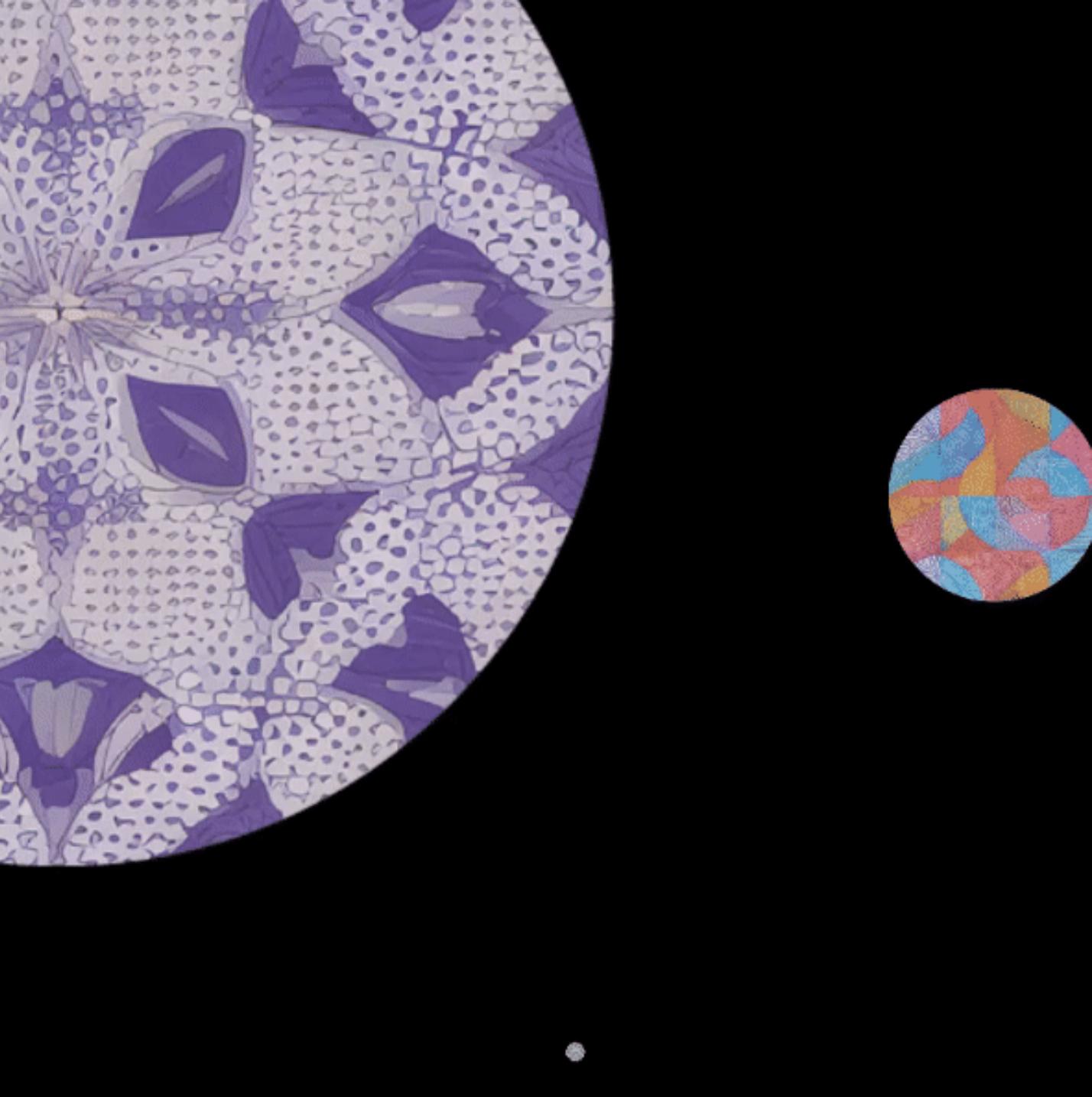
(條件達成的要執行的編碼)

break; (宣告完成離開 switch 區塊)

尋求更進階編碼的同學可以打開例子

3.10 advanceVisualizer

內裡增加了 p5.Reverb() applyMatrix(); translate(); 及 resetMatrix();
下一頁有相應的 reference page 連結給你們課後自學



p5.Reverb();

<https://p5js.org/reference/#/p5.Reverb>

applyMatrix();

<https://p5js.org/reference/#/p5/applyMatrix>

translate();

<https://p5js.org/reference/#/p5/translate>

resetMatrix();

<https://p5js.org/reference/#/p5/resetMatrix>

要進一步使用編碼創作，自學能力是關鍵呵～

最後我們試一下在例子 3.9 中加入 createCapture() 吧！

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(); 在顯示框內顯示文字資訊

不明白text();內的數值有甚麼意義？

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")`

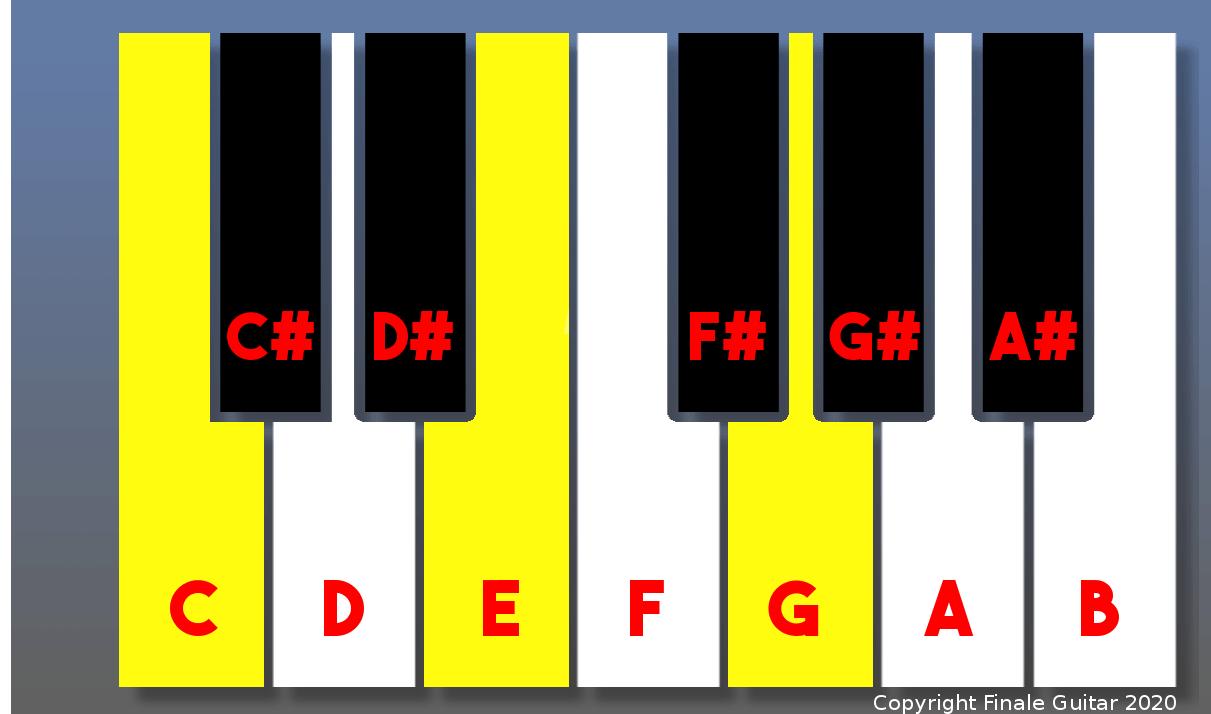
`monoSynth` 就會播放音符 C4

```
9▼ function keyPressed() {  
10▼   if (key == "a") {  
11     userStartAudio();  
12     monoSynth.play("C4", 1, 0, 0.25);  
13   }  
14▼   if (key == "s") {  
15     userStartAudio();  
16     monoSynth.play("F4", 1, 0, 0.25);  
17   }  
18 }  
19 }
```

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



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(); 內的數值有甚麼意義?

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

```
1 let osc;
2
3 function setup() {
4   createCanvas(500, 500);
5   background(0);
6   osc = new p5.Oscillator();
7   osc.freq(220);
8   osc.setType("sine");
9   osc.amp(0.5);
10  osc.start();
11 }
```

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

```
9
10 function draw() {
11   background(0);
12   freq = map(mouseX, 0, width, 100, 500);
13   amp = map(mouseY, height, 0, 0, 1);
14   ellipse(mouseX, mouseY, 50);
15   osc.freq(freq);
16   osc.amp(amp);
17 }
18
```

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

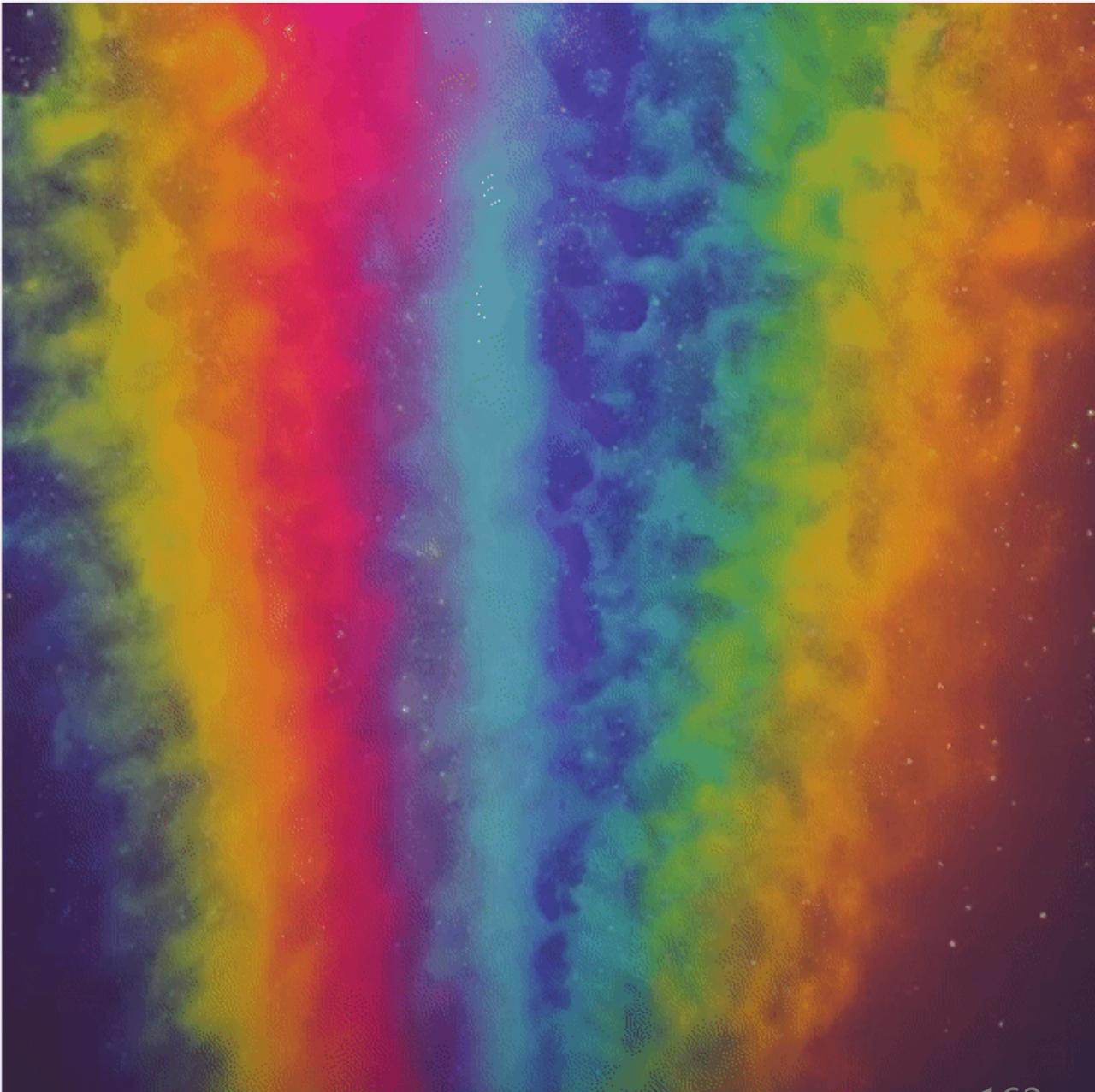
4.7 getColour

```
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(影片闊度，影片高度)

video.volume(影片聲音大小) 0 為無聲 1為最大聲

video.loop() 重複播放影片

video.hide() 隱藏HTML內的影像元素，免去重複顯示影像。

```
11
12 function draw(){
13   image(img,0,0);
14   col = img.get(mouseX, mouseY);
15   fill(col);
16   ellipse(mouseX, mouseY, 10, 10);
17 }
18
```

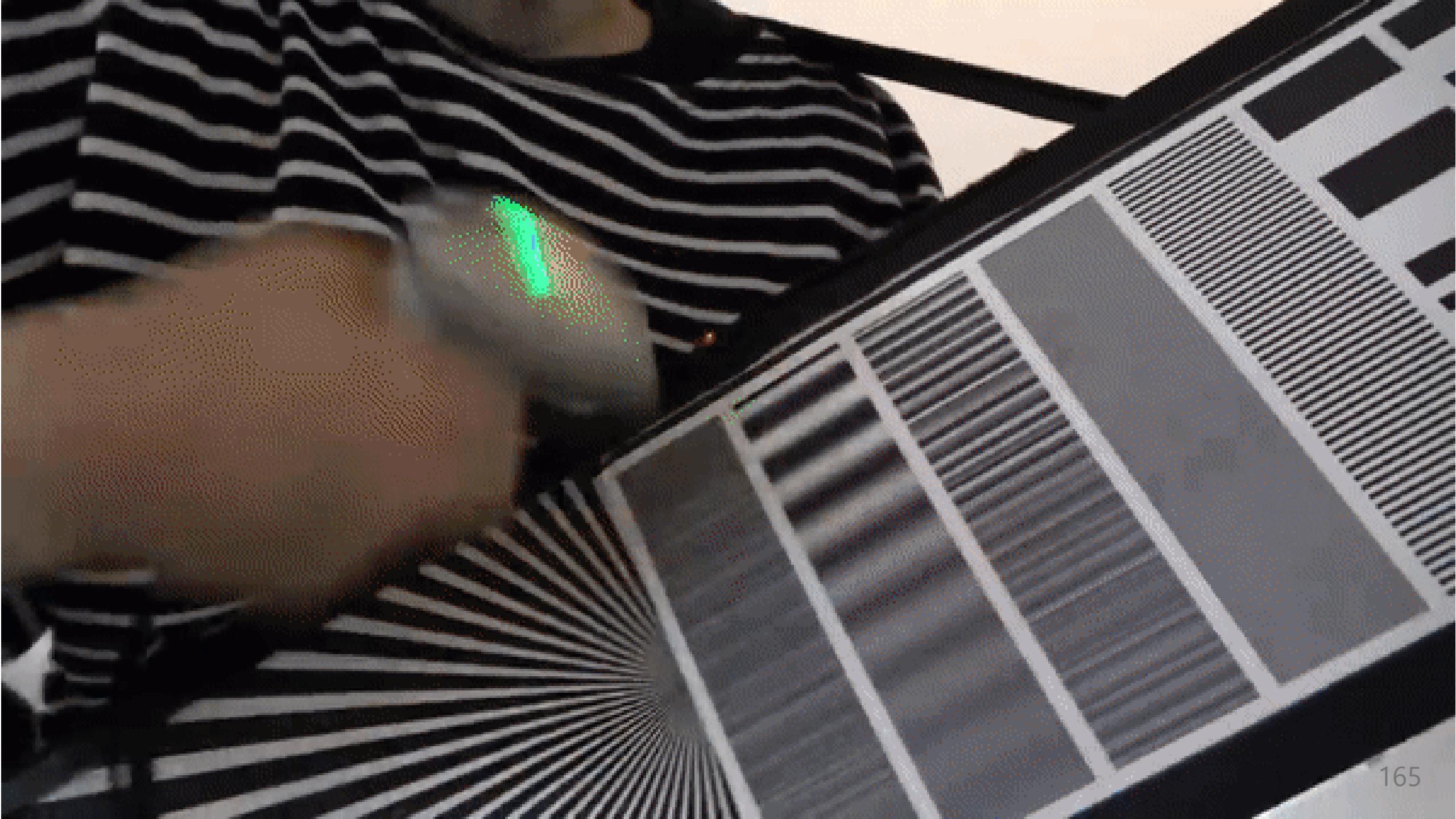
Sonification 資訊聲音化

把畫面轉化可聽的 聲音 或 音樂

電磁祭囉子

[https://www.youtube.com/watch?
v=xlOjmLeQe3w](https://www.youtube.com/watch?v=xlOjmLeQe3w)





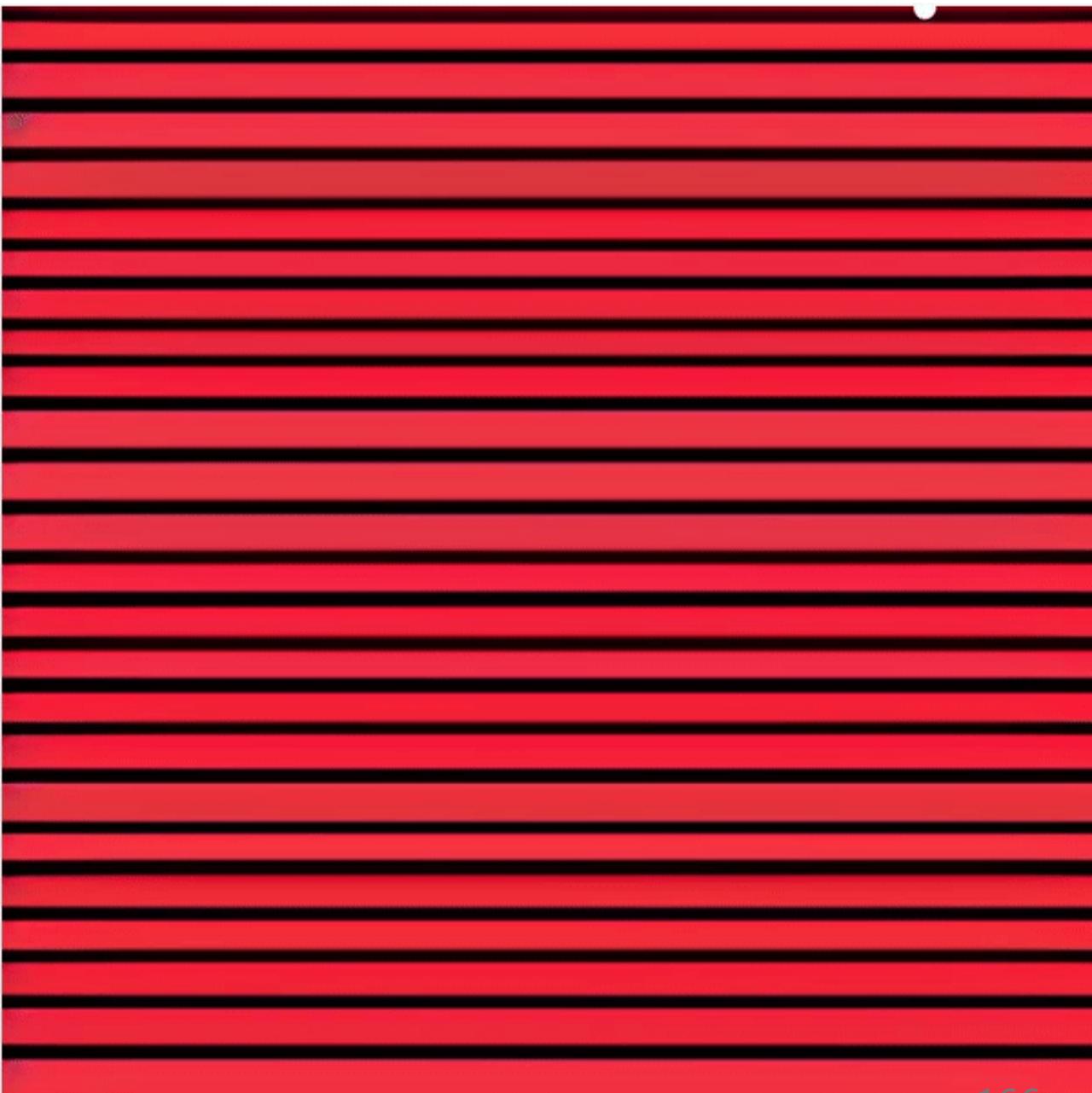
4.8 simpleSonification

```
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

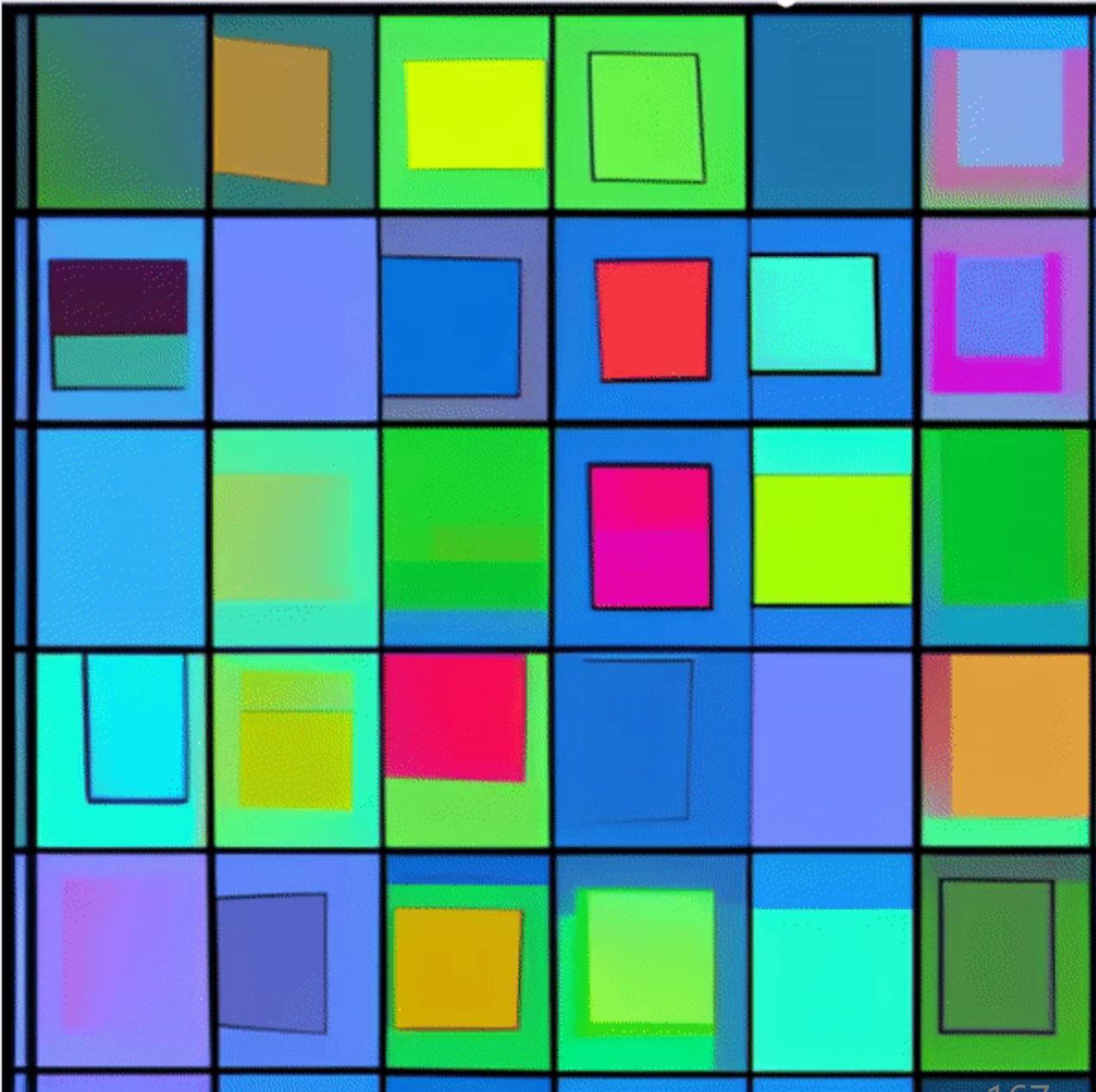
```
let img, col;
let polySynt;
let note0,note1,note2;

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;

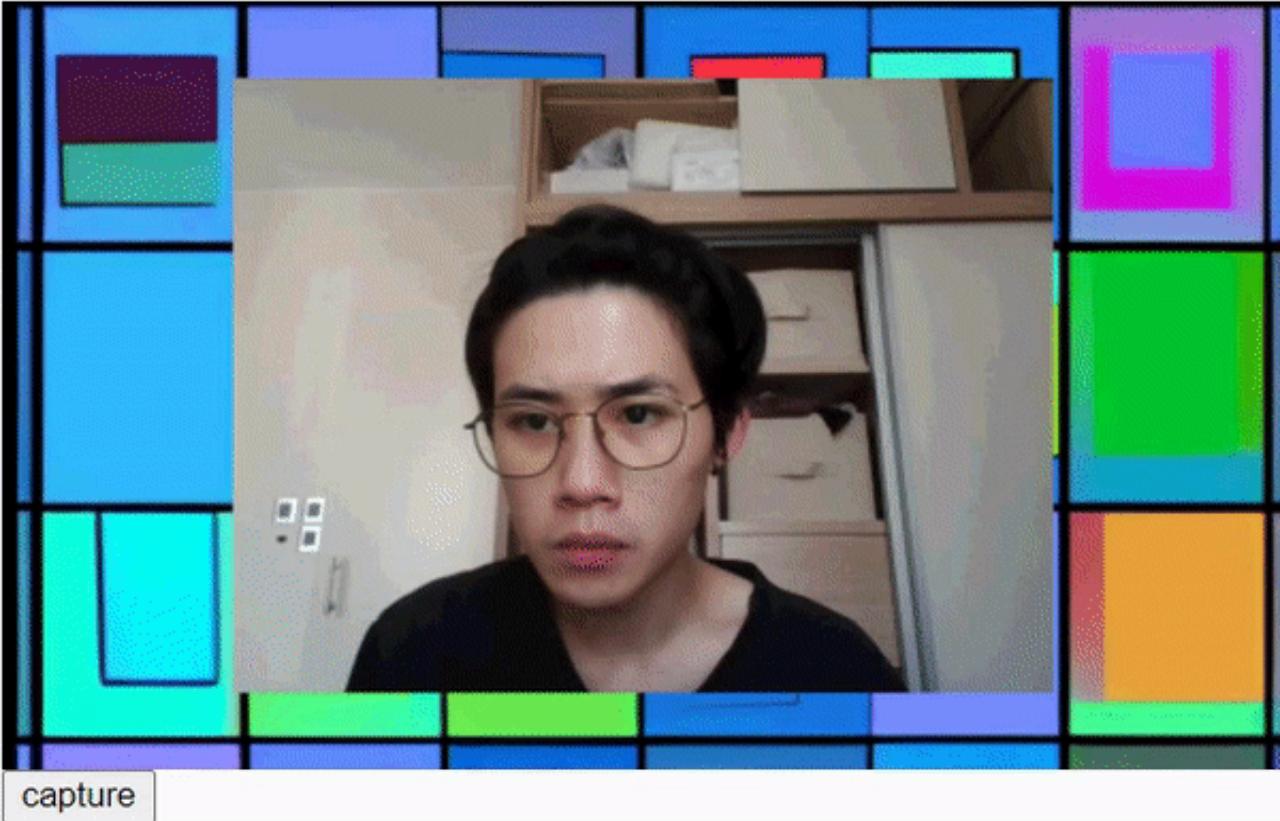
function preload() {
  img = loadImage("colourGrid.png");
}

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(POTERIZE, 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(POTERIZE, vol*255 + 6);

}

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

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