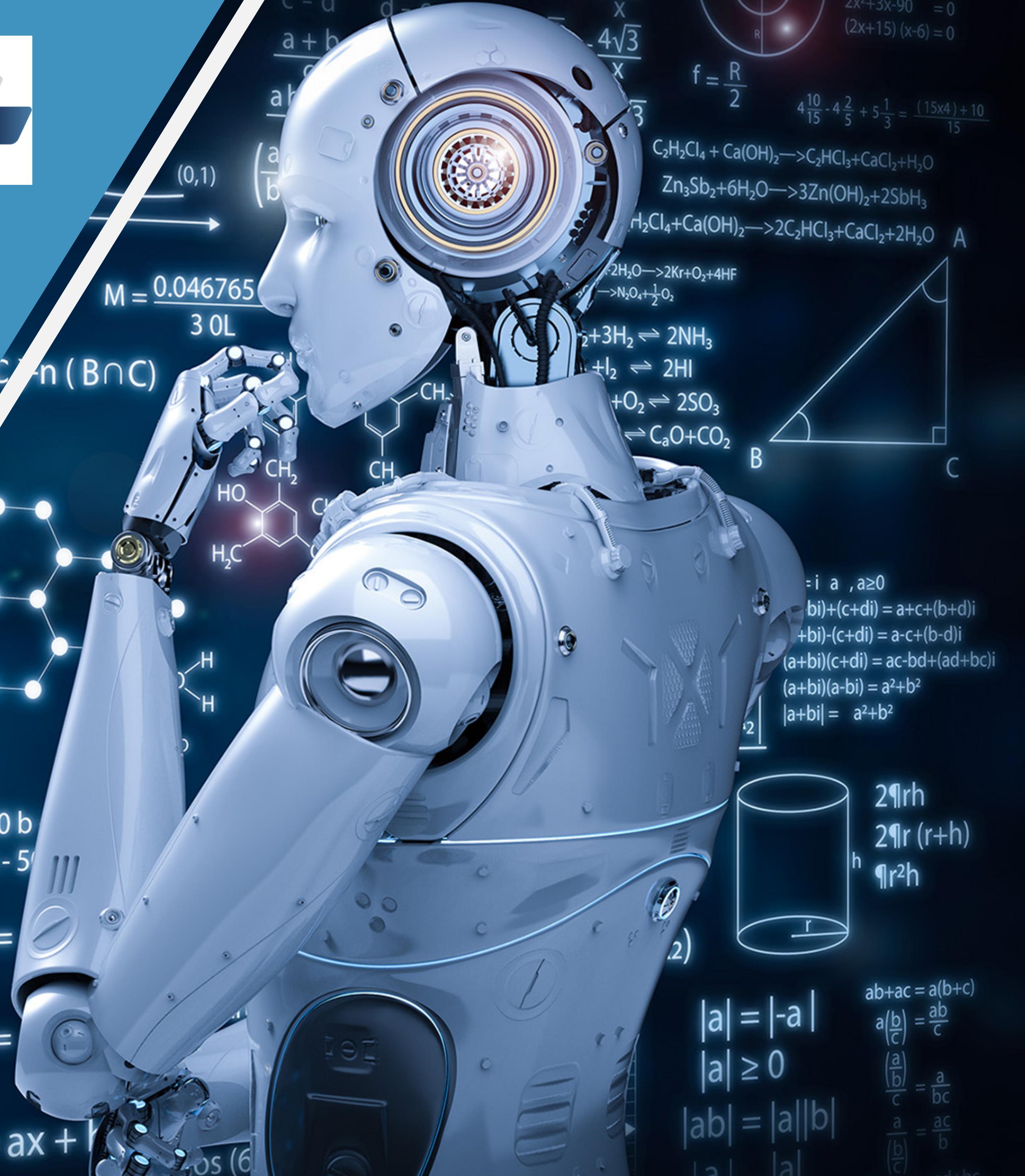




# Day 05

## 深度學習與電腦視覺 學習馬拉松

Cupay 陪跑專家：楊鎮銘





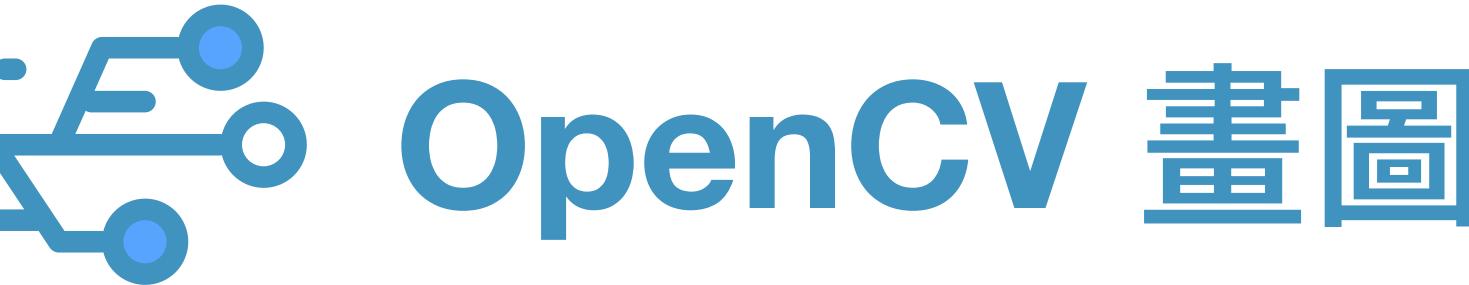
# 基礎影像處理

## OpenCV 畫圖-影像處理

# 重要知識點



熟悉透過 OpenCV 來畫圖的過程

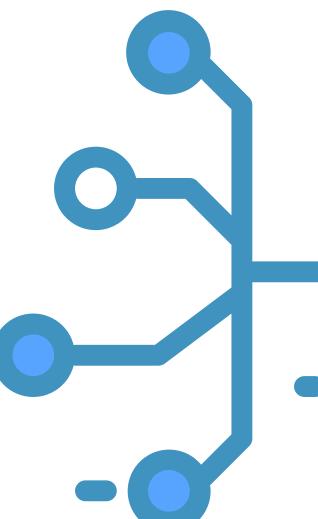


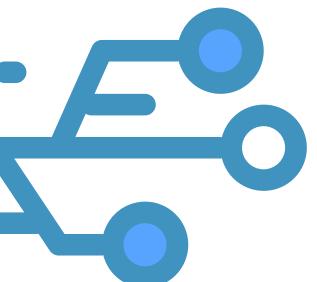
# OpenCV 畫圖



我們經常使用 OpenCV 來做電腦視覺相關的任務  
適當的在圖片上做標記會讓我們更容易了解結果的正確性

- 偵測：標記預選框 / 偵測框等
- 遮罩：標記遮照等
- 追蹤：標記時間內移動軌跡 / 輪廓等
- 預處理：檢查經過 transformation 後的標記是否正確等





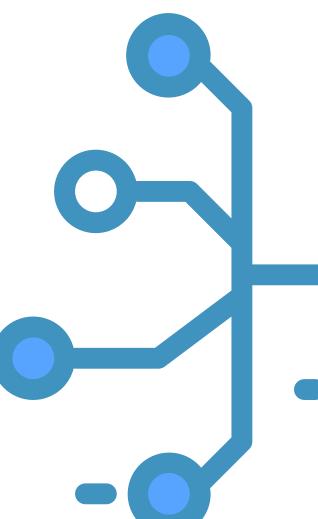
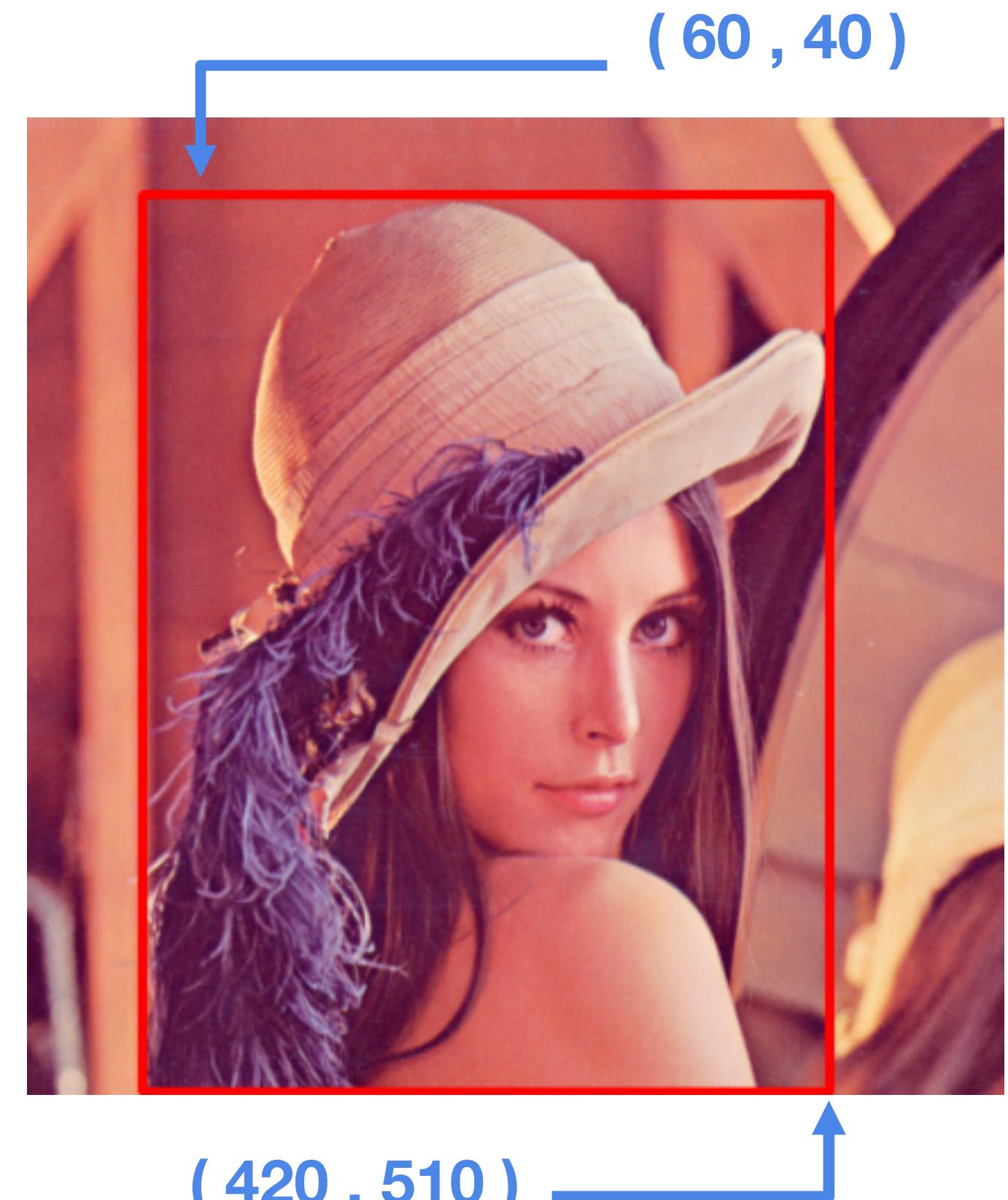
# OpenCV 畫圖 - 矩形

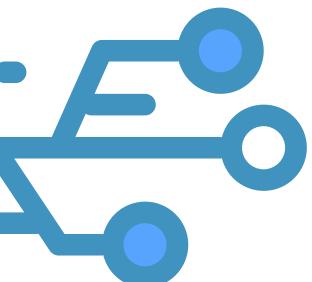


通常在定位目標位置的時候會使用矩形

```
cv2.rectangle(  
    image,          # 圖片  
    (60, 40),      # 左上角的座標  
    (420, 510),    # 右上角的座標  
    (0, 0, 255),   # BGR 顏色  
    3)             # 線的粗細程度
```

BGR 格式的  $(0, 0, 255)$  代表紅色的值是 255





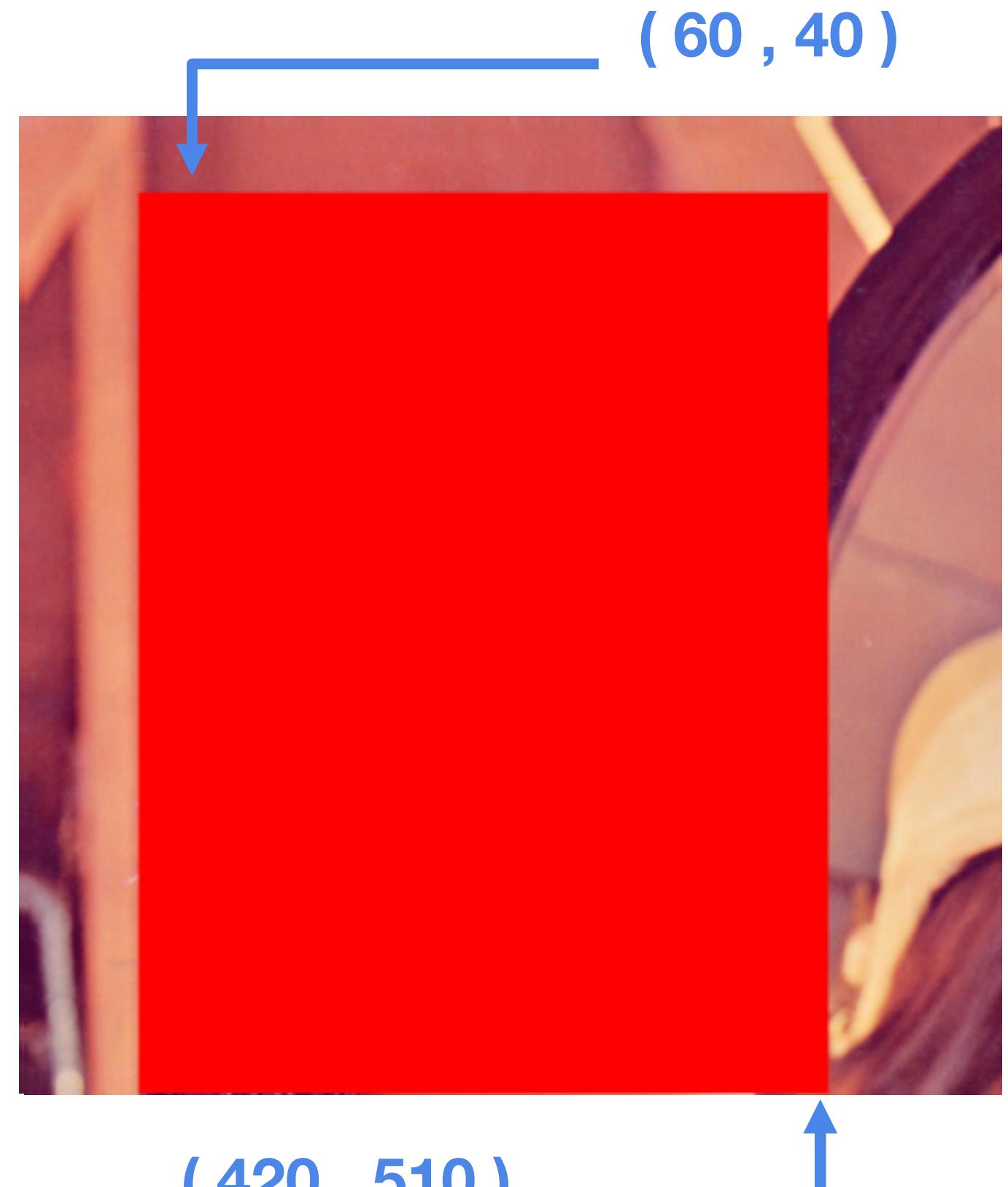
# OpenCV 畫圖 - 矩形



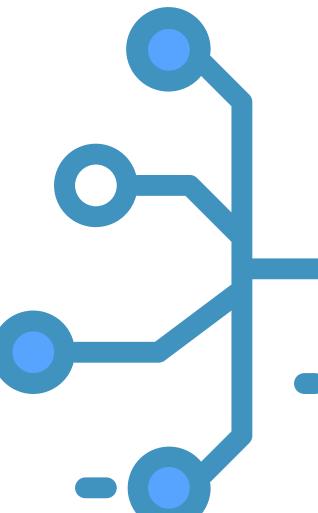
通常在定位目標位置的時候會使用矩形

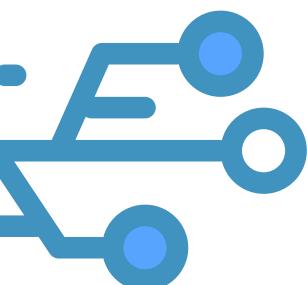
```
cv2.rectangle(  
    image,          # 圖片  
    (60, 40),      # 左上角的座標  
    (420, 510),    # 右上角的座標  
    (0, 0, 255),   # BGR 顏色  
    -1)            # 線的粗細程度
```

假如填上負值則代表將矩型填滿



Note : OpenCV 畫圖是直接畫在圖片上面，並不是回傳一個畫好圖的圖片



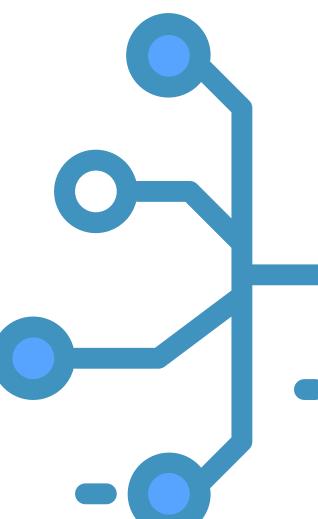
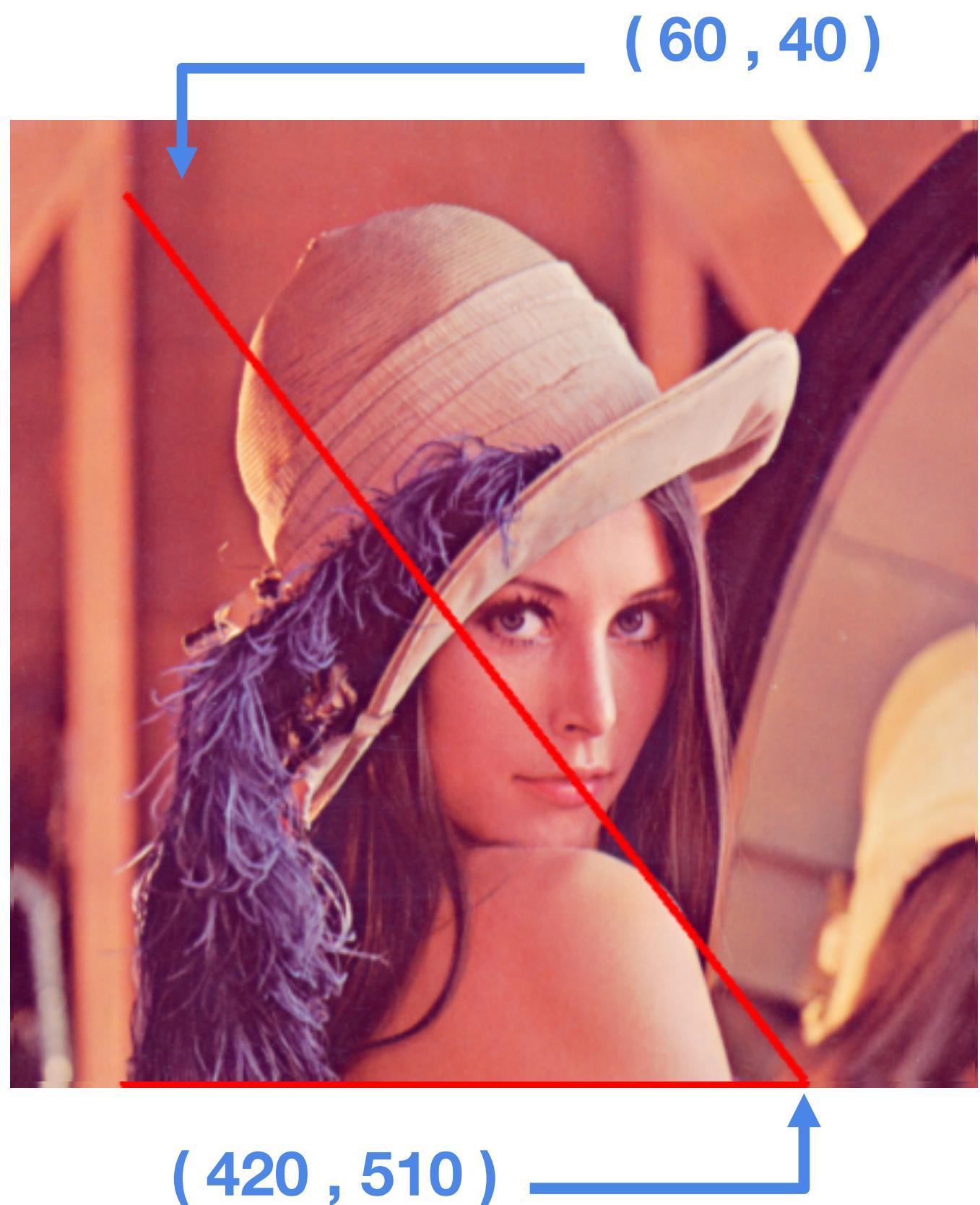


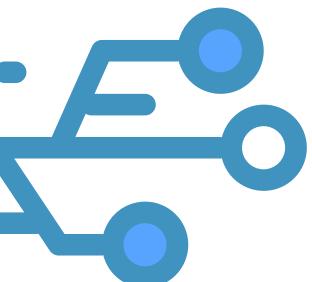
# OpenCV 畫圖 - 線



通常在標記路徑 / 輪廓時會使用線  
這邊簡單舉例 OpenCV 的寫法

```
cv2.line(  
    image,          # 圖片  
    (60, 40),      # 左上角的座標  
    (420, 510),    # 右上角的座標  
    (0, 0, 255),   # BGR 顏色  
    3)             # 線的粗細程度
```



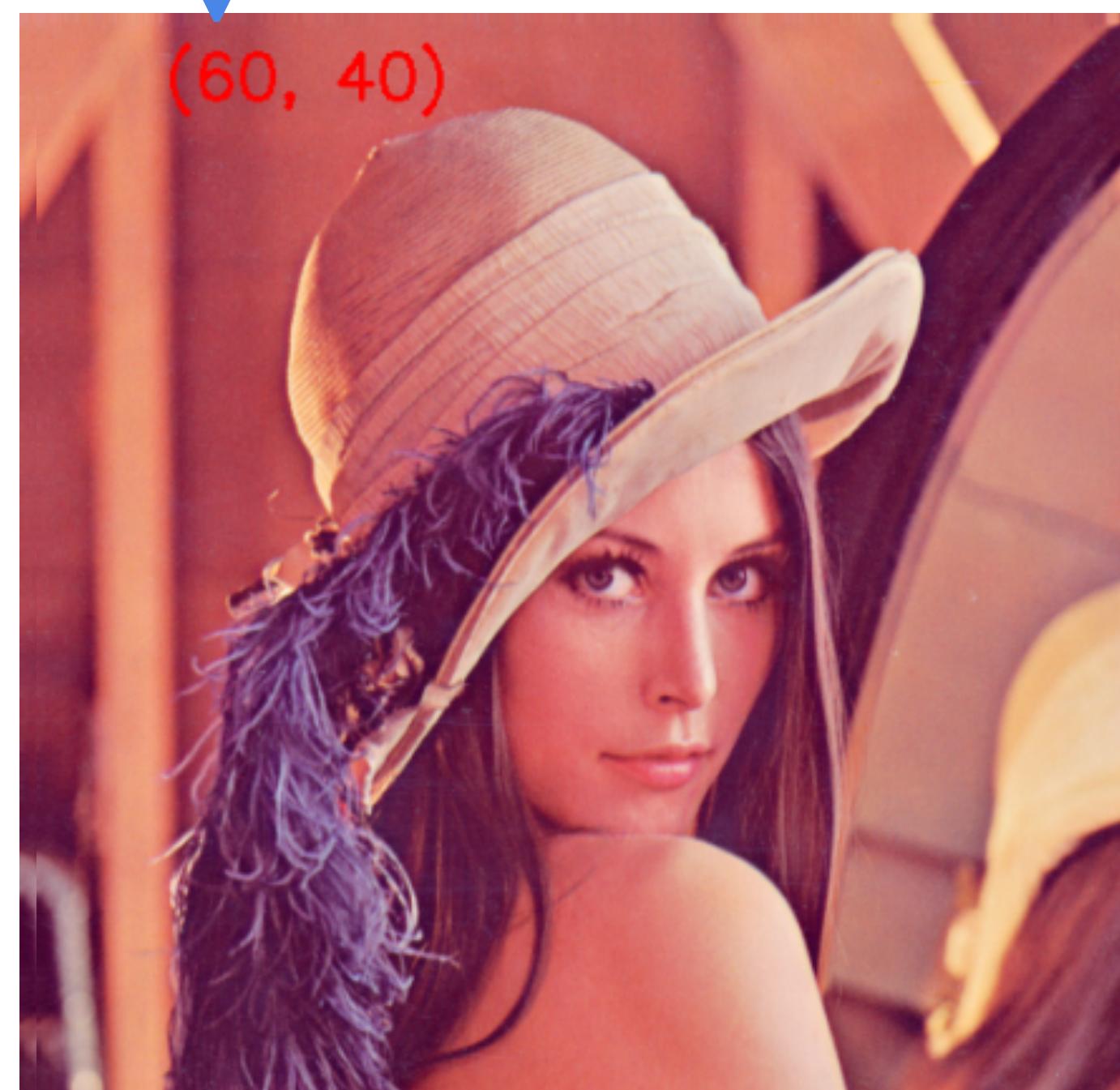


# OpenCV 畫圖 - 文字

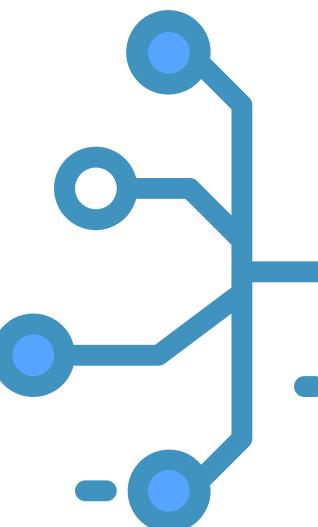


通常在標記座標或是檔名時需要加上文字

```
cv2.putText(  
    image,          # 圖片  
    '(60, 40)',   # 要加的文字  
    (60, 40),      # 文字左下角位置  
    0,              # 字型 (參考連結)  
    3,              # 字體大小  
    (0, 0, 255),   # 字體顏色  
    2)              # 字體粗細
```



標記文字

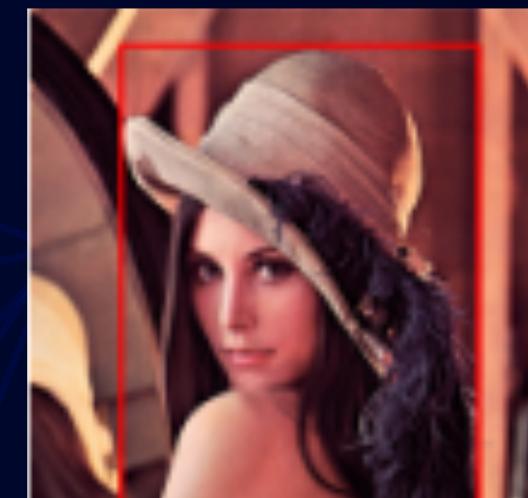
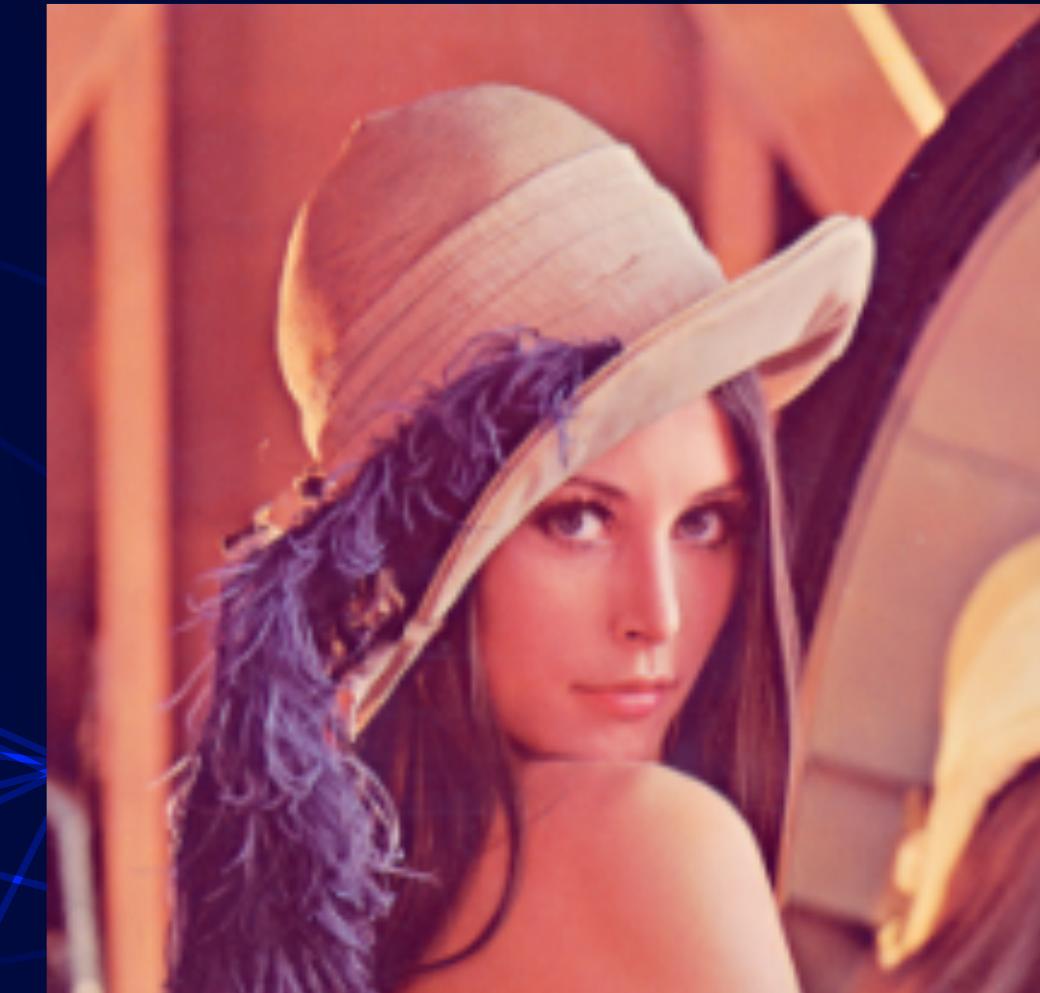


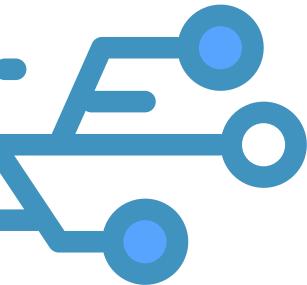
# 知識點回顧

- 了解 OpenCV 畫圖的方式
  - \* 直接畫在圖片上，並不會回傳結果圖
  - \* 除了幾何圖形也可以印字

# 範例

- 假設我們希望先對圖片做以下幾點預處理，請印出最後結果
- Hint: 注意先後順序，人物原始邊框座標  $(60, 40), (420, 510)$ 
  - 對明亮度做直方圖均衡處理
  - 水平鏡像 + 縮放處理 (0.5 倍)
  - 畫出人物矩形邊框





# 推薦延伸閱讀



The screenshot shows the OpenCV Python Tutorials page for 'Drawing Functions in OpenCV'. The page includes a 'Goal' section with learning objectives, a 'Code' section with common arguments, and examples for 'Drawing Line' and 'Drawing Rectangle' with accompanying code snippets.

**Goal**

- Learn to draw different geometric shapes with OpenCV
- You will learn these functions : `cv2.line()`, `cv2.circle()` , `cv2.rectangle()`, `cv2.ellipse()`, `cv2.putText()` etc.

**Code**

In all the above functions, you will see some common arguments as given below:

- img : The image where you want to draw the shapes
- color : Color of the shape. for BGR, pass it as a tuple, eg: (255,0,0) for blue. For grayscale, just pass the scalar value.
- thickness : Thickness of the line or circle etc. If `**-1**` is passed for closed figures like circles, it will fill the shape. `default thickness = 1`
- lineType : Type of line, whether 8-connected, anti-aliased line etc. *By default, it is 8-connected.* `cv2.LINE_AA` gives anti-aliased line which looks great for curves.

**Drawing Line**

To draw a line, you need to pass starting and ending coordinates of line. We will create a black image and draw a blue line on it from top-left to bottom-right corners.

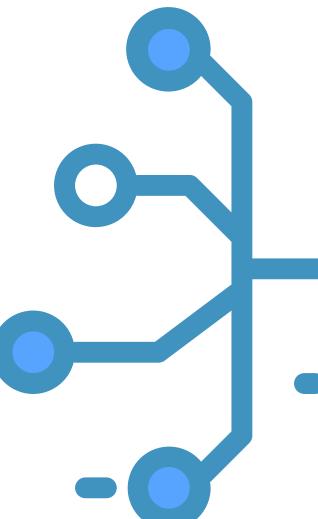
```
1 import numpy as np
2 import cv2
3
4 # Create a black image
5 img = np.zeros((512,512,3), np.uint8)
6
7 # Draw a diagonal blue line with thickness of 5 px
8 cv2.line(img,(0,0),(511,511),(255,0,0),5)
```

**Drawing Rectangle**

To draw a rectangle, you need top-left corner and bottom-right corner of rectangle. This time we will draw a green rectangle at the top-right corner of image.

<https://docs.opencv.org/3.1.0/index.html>

**OpenCV 官方網站 - Drawing**  
介紹一些額外的畫圖功能，包含多邊形與圓形等方式  
[連結](#)



# 解題時間 Let's Crack It



請跳出 PDF 至官網 Sample Code & 作業開始解題