Mediapipe:pose

211062016 謝宗璇

0.6
0.2
-0.2
-0.6
-0.6
-0.2
0.2
0.6
0.6
0.2
-0.2
-0.6

4個階段！

1.基本架構

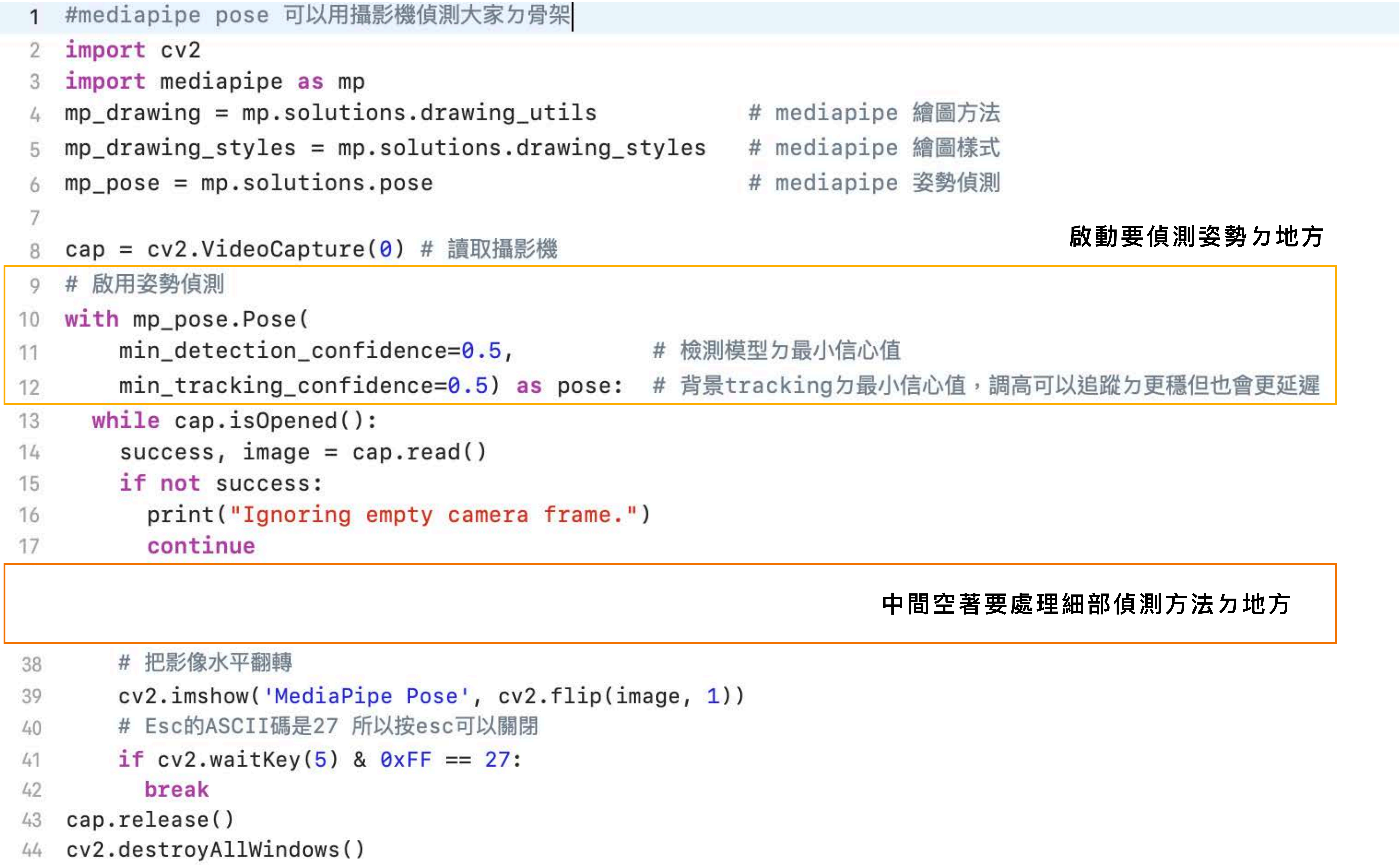
2.去背

3.抓關節角度

4.整合！

階段1:基本ㄉmediapipe:pose

階段1:基本ㄉmediapipe



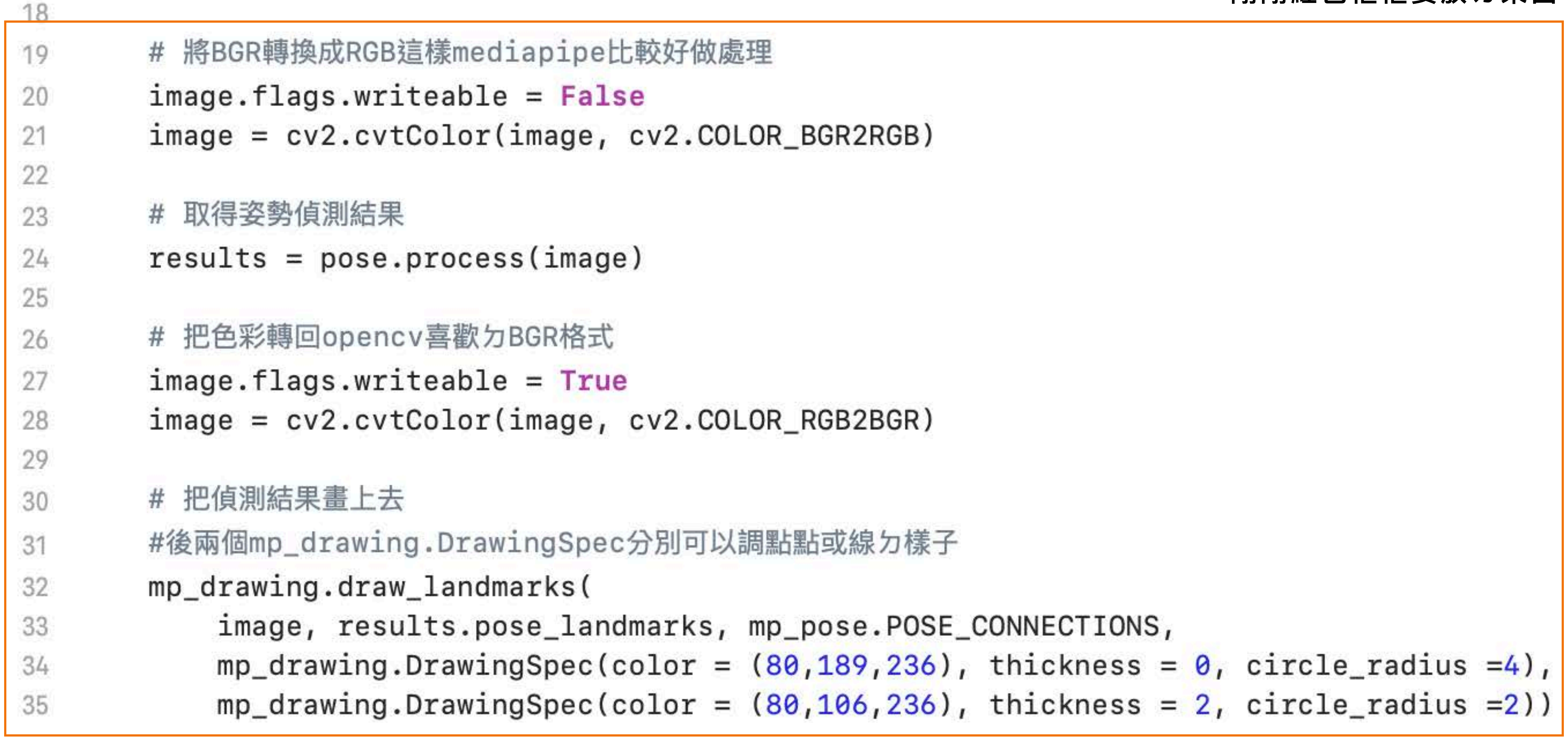
```python
#mediapipe pose 可以用攝影機偵測大家ㄉ骨架
import cv2
import mediapipe as mp
mp_drawing = mp.solutions.drawing_utils               # mediapipe 繪圖方法
mp_drawing_styles = mp.solutions.drawing_styles   # mediapipe 繪圖樣式
mp_pose = mp.solutions.pose                           # mediapipe 姿勢偵測

cap = cv2.VideoCapture(0) # 讀取攝影機
# 啟用姿勢偵測
with mp_pose.Pose(
    min_detection_confidence=0.5,          # 檢測模型ㄉ最小信心值
    min_tracking_confidence=0.5) as pose:  # 背景trackingㄉ最小信心值，調高可以追蹤ㄉ更穩但也會更延遲
  while cap.isOpened():
    success, image = cap.read()
    if not success:
      print("Ignoring empty camera frame.")
      continue

    # 把影像水平翻轉
    cv2.imshow('MediaPipe Pose', cv2.flip(image, 1))
    # Esc的ASCII碼是27 所以按esc可以關閉
    if cv2.waitKey(5) & 0xFF == 27:
      break
cap.release()
cv2.destroyAllWindows()
```

啟動要偵測姿勢ㄉ地方

中間空著要處理細部偵測方法ㄉ地方

階段1:基本ㄉmediapipe

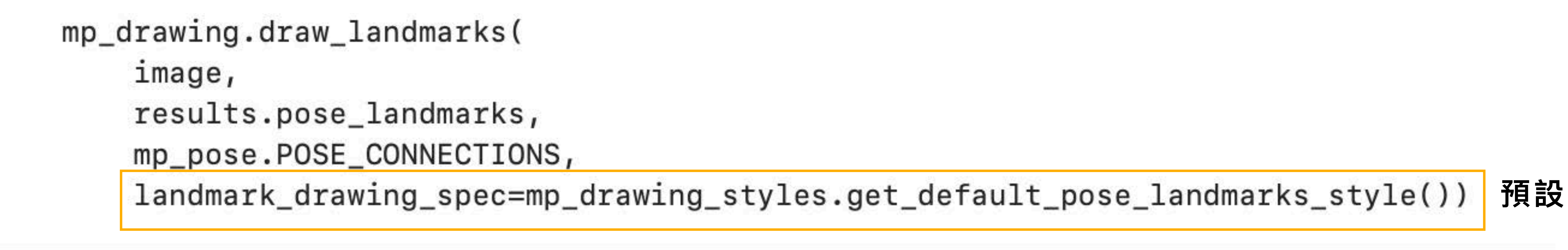
剛剛紅色框框要放ㄉ東西



```
# 將BGR轉換成RGB這樣mediapipe比較好做處理
image.flags.writeable = False
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# 取得姿勢偵測結果
results = pose.process(image)

# 把色彩轉回opencv喜歡ㄉBGR格式
image.flags.writeable = True
image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

# 把偵測結果畫上去
#後兩個mp_drawing.DrawingSpec分別可以調點點或線ㄉ樣子
mp_drawing.draw_landmarks(
    image, results.pose_landmarks, mp_pose.POSE_CONNECTIONS,
    mp_drawing.DrawingSpec(color = (80,189,236), thickness = 0, circle_radius =4),
    mp_drawing.DrawingSpec(color = (80,106,236), thickness = 2, circle_radius =2))
```

調點點樣式

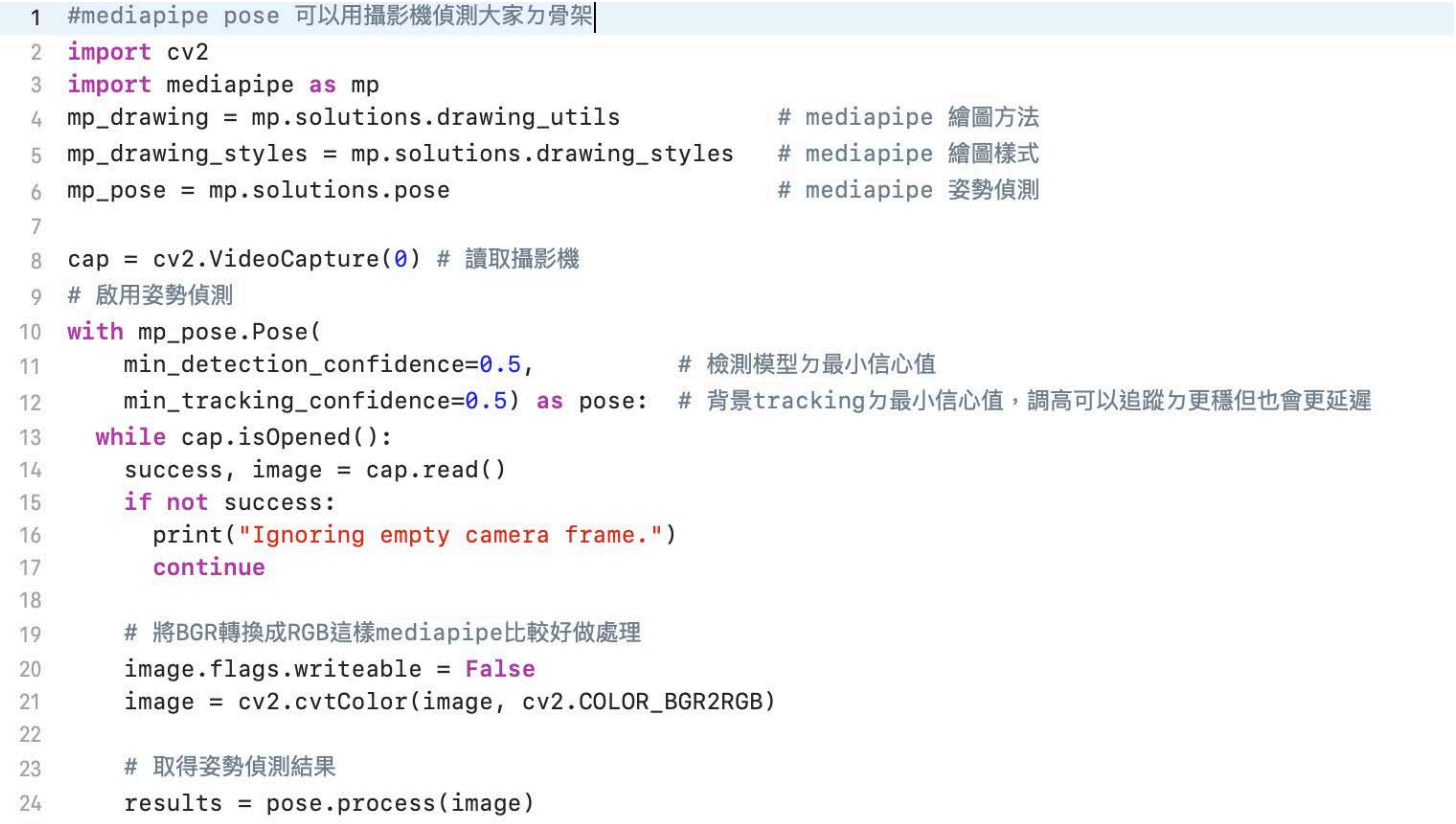
調線樣式



```
mp_drawing.draw_landmarks(
    image,
    results.pose_landmarks,
    mp_pose.POSE_CONNECTIONS,
    landmark_drawing_spec=mp_drawing_styles.get_default_pose_landmarks_style())
```

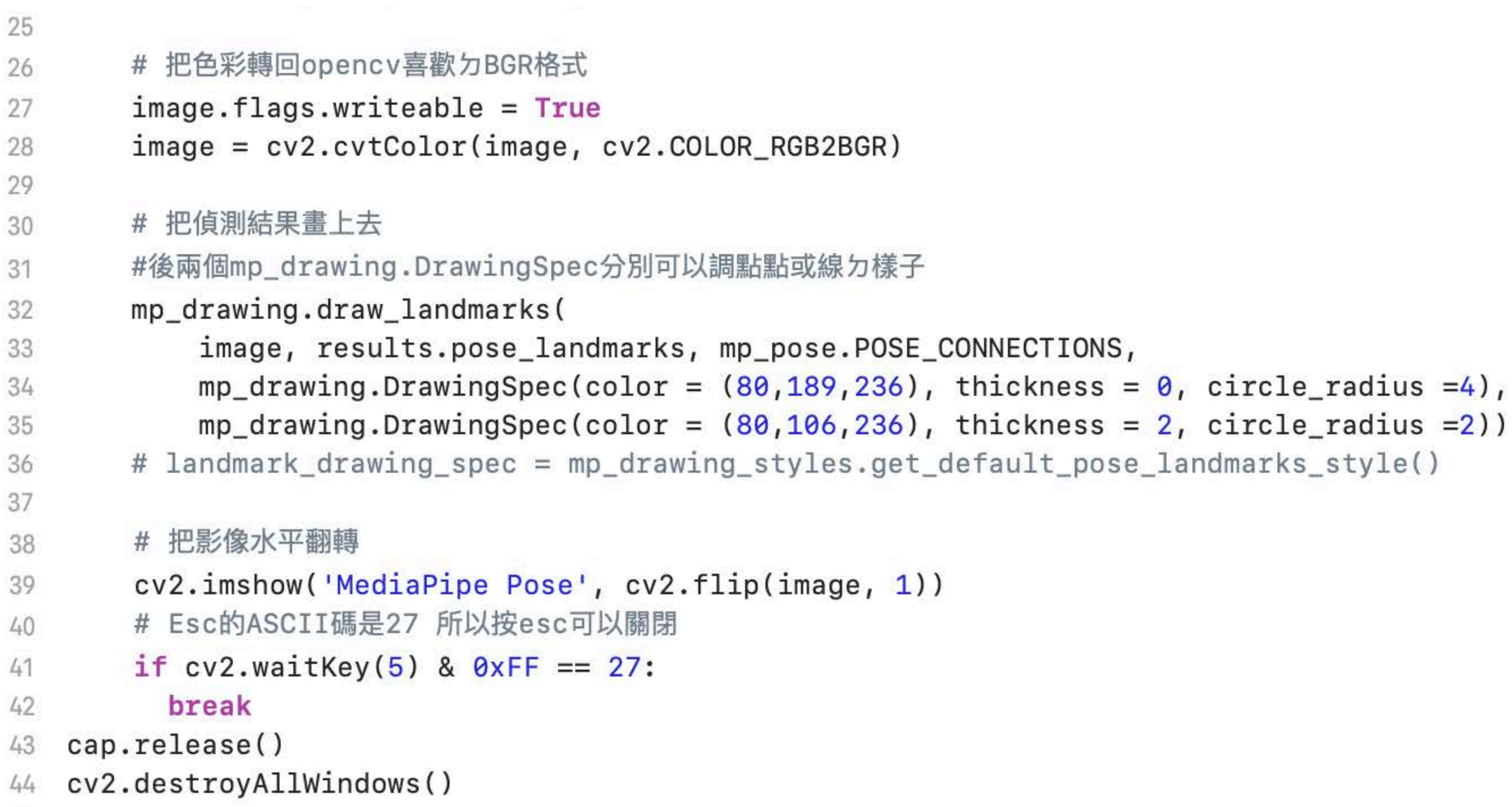
預設ㄉ點線樣式

Full Code(1/2)



```
#mediapipe pose 可以用攝影機偵測大家ㄉ骨架
import cv2
import mediapipe as mp
mp_drawing = mp.solutions.drawing_utils           # mediapipe 繪圖方法
mp_drawing_styles = mp.solutions.drawing_styles   # mediapipe 繪圖樣式
mp_pose = mp.solutions.pose                       # mediapipe 姿勢偵測

cap = cv2.VideoCapture(0) # 讀取攝影機
# 啟用姿勢偵測
with mp_pose.Pose(
    min_detection_confidence=0.5,          # 檢測模型ㄉ最小信心值
    min_tracking_confidence=0.5) as pose:  # 背景trackingㄉ最小信心值，調高可以追蹤ㄉ更穩但也會更延遲
  while cap.isOpened():
    success, image = cap.read()
    if not success:
      print("Ignoring empty camera frame.")
      continue

    # 將BGR轉換成RGB這樣mediapipe比較好做處理
    image.flags.writeable = False
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    # 取得姿勢偵測結果
    results = pose.process(image)
```

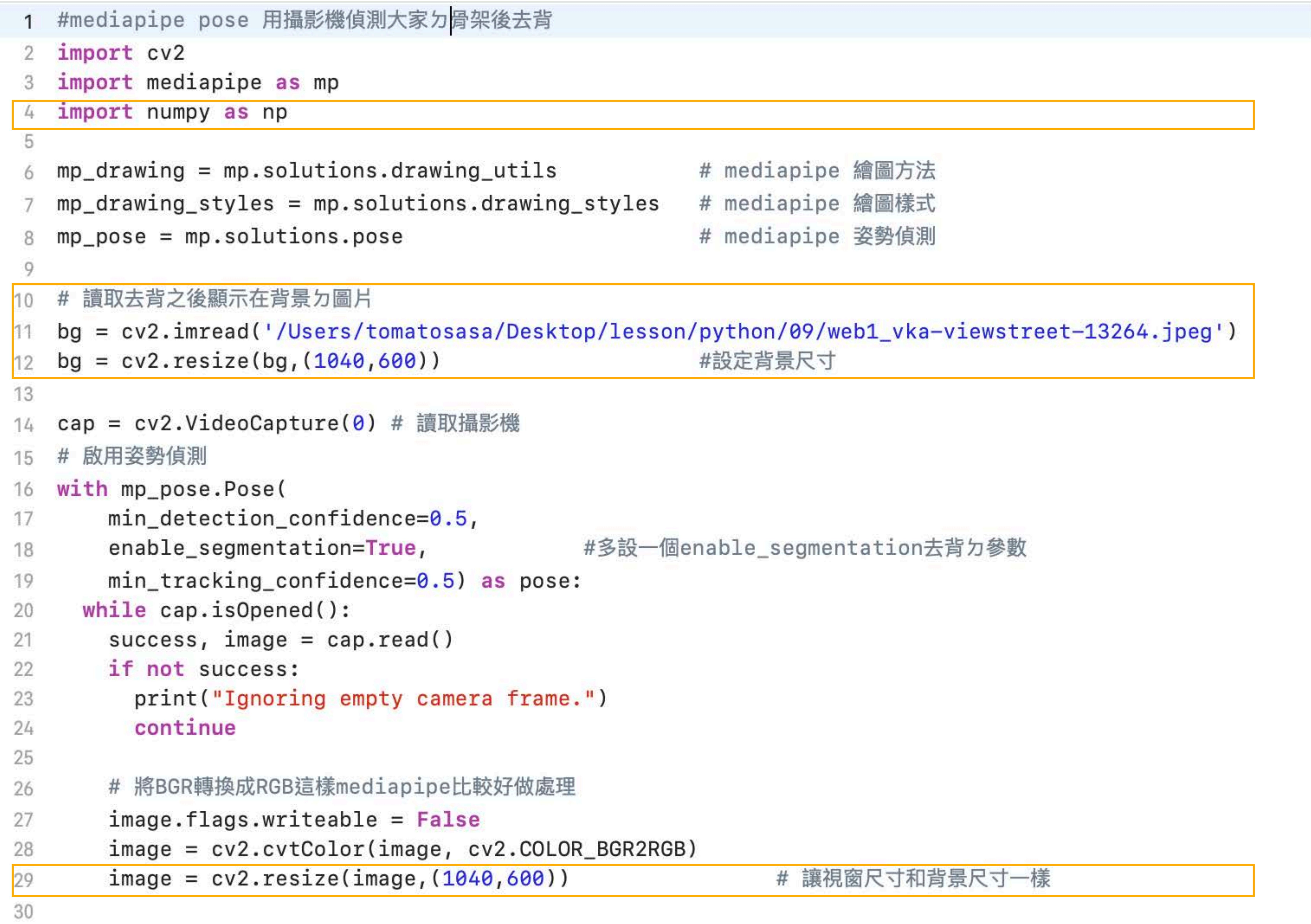
Full Code(2/2)



```
    # 把色彩轉回opencv喜歡ㄉBGR格式
    image.flags.writeable = True
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

    # 把偵測結果畫上去
    #後兩個mp_drawing.DrawingSpec分別可以調點點或線ㄉ樣子
    mp_drawing.draw_landmarks(
        image, results.pose_landmarks, mp_pose.POSE_CONNECTIONS,
        mp_drawing.DrawingSpec(color = (80,189,236), thickness = 0, circle_radius =4),
        mp_drawing.DrawingSpec(color = (80,106,236), thickness = 2, circle_radius =2))
    # landmark_drawing_spec = mp_drawing_styles.get_default_pose_landmarks_style()

    # 把影像水平翻轉
    cv2.imshow('MediaPipe Pose', cv2.flip(image, 1))
    # Esc的ASCII碼是27 所以按esc可以關閉
    if cv2.waitKey(5) & 0xFF == 27:
      break
cap.release()
cv2.destroyAllWindows()
```

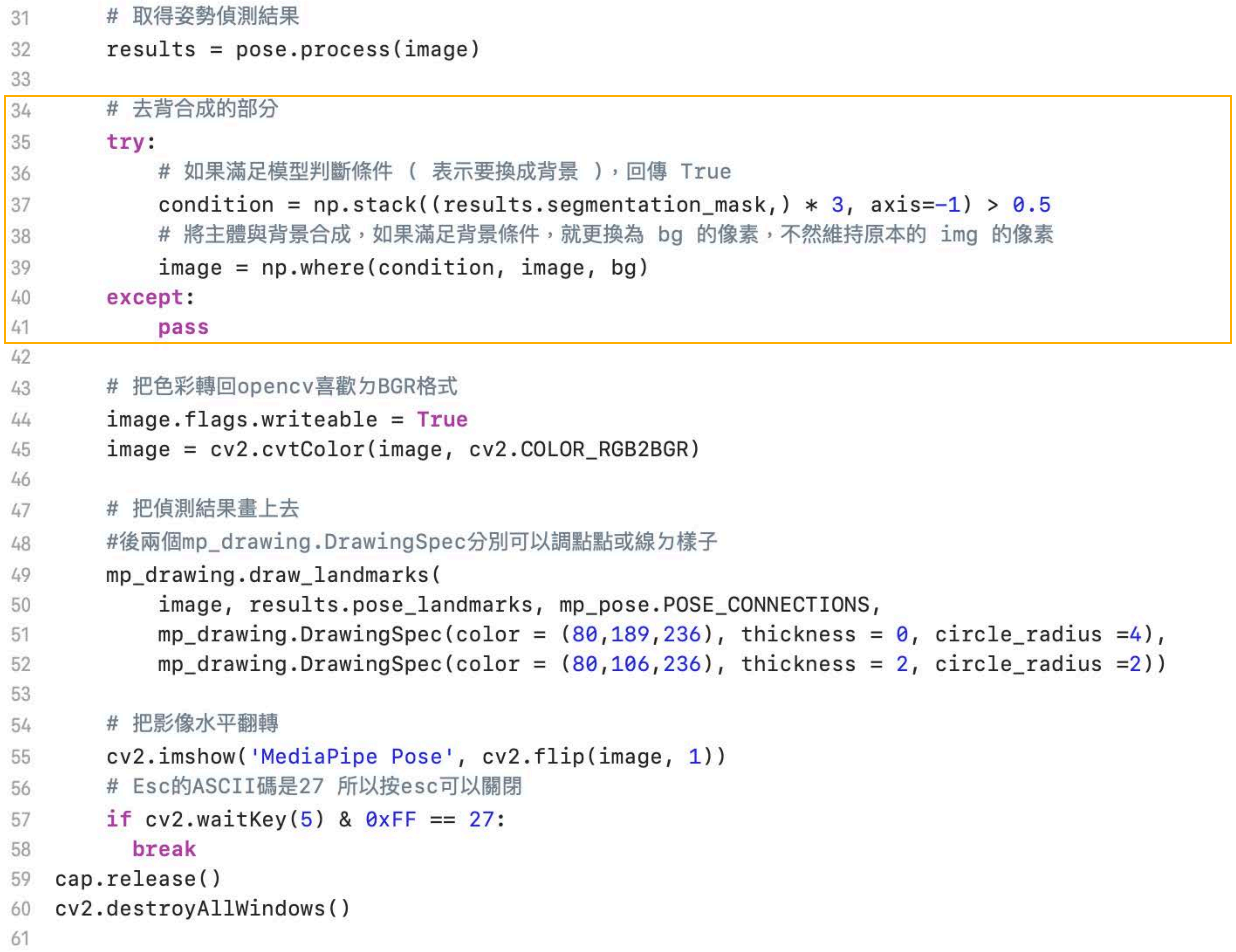
階段2:mediapipe:pose去背！

Full Code(1/2)



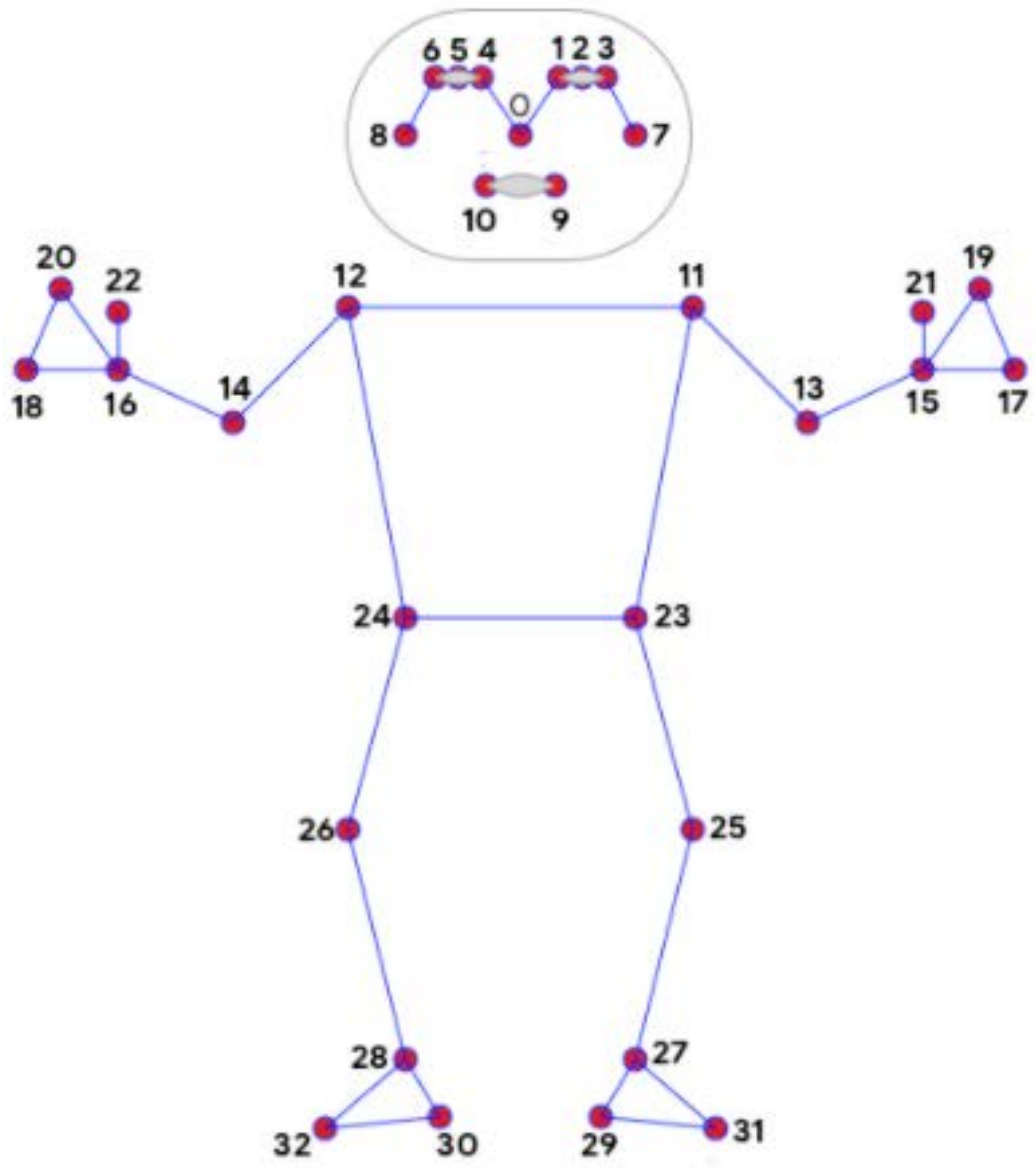
```
#mediapipe pose 用攝影機偵測大家ㄉ骨架後去背
import cv2
import mediapipe as mp
import numpy as np

mp_drawing = mp.solutions.drawing_utils           # mediapipe 繪圖方法
mp_drawing_styles = mp.solutions.drawing_styles   # mediapipe 繪圖樣式
mp_pose = mp.solutions.pose                       # mediapipe 姿勢偵測

# 讀取去背之後顯示在背景ㄉ圖片
bg = cv2.imread('/Users/tomatosasa/Desktop/lesson/python/09/web1_vka-viewstreet-13264.jpeg')
bg = cv2.resize(bg,(1040,600))                    #設定背景尺寸

cap = cv2.VideoCapture(0) # 讀取攝影機
# 啟用姿勢偵測
with mp_pose.Pose(
    min_detection_confidence=0.5,
    enable_segmentation=True,            #多設一個enable_segmentation去背ㄉ參數
    min_tracking_confidence=0.5) as pose:
  while cap.isOpened():
    success, image = cap.read()
    if not success:
      print("Ignoring empty camera frame.")
      continue

    # 將BGR轉換成RGB這樣mediapipe比較好做處理
    image.flags.writeable = False
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    image = cv2.resize(image,(1040,600))                # 讓視窗尺寸和背景尺寸一樣

```

Full Code(2/2)



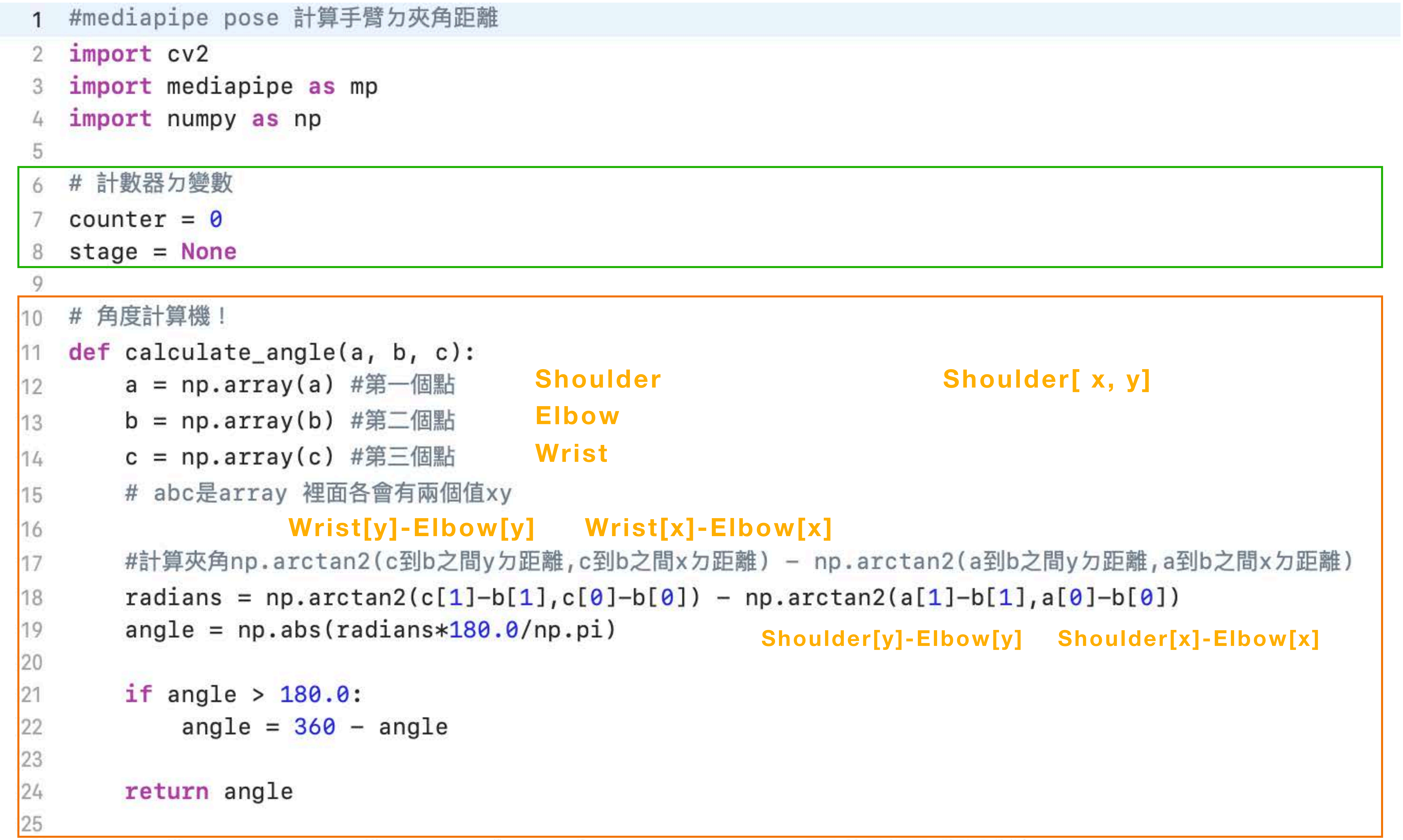
```
        # 取得姿勢偵測結果
        results = pose.process(image)

        # 去背合成的部分
        try:
            # 如果滿足模型判斷條件 ( 表示要換成背景 )，回傳 True
            condition = np.stack((results.segmentation_mask,) * 3, axis=-1) > 0.5
            # 將主體與背景合成，如果滿足背景條件，就更換為 bg 的像素，不然維持原本的 img 的像素
            image = np.where(condition, image, bg)
        except:
            pass

        # 把色彩轉回opencv喜歡ㄉBGR格式
        image.flags.writeable = True
        image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

        # 把偵測結果畫上去
        #後兩個mp_drawing.DrawingSpec分別可以調點點或線ㄉ樣子
        mp_drawing.draw_landmarks(
            image, results.pose_landmarks, mp_pose.POSE_CONNECTIONS,
            mp_drawing.DrawingSpec(color = (80,189,236), thickness = 0, circle_radius =4),
            mp_drawing.DrawingSpec(color = (80,106,236), thickness = 2, circle_radius =2))

        # 把影像水平翻轉
        cv2.imshow('MediaPipe Pose', cv2.flip(image, 1))
        # Esc的ASCII碼是27 所以按esc可以關閉
        if cv2.waitKey(5) & 0xFF == 27:
          break
    cap.release()
    cv2.destroyAllWindows()
```

階段3:Mediapipe:pose抓角度！



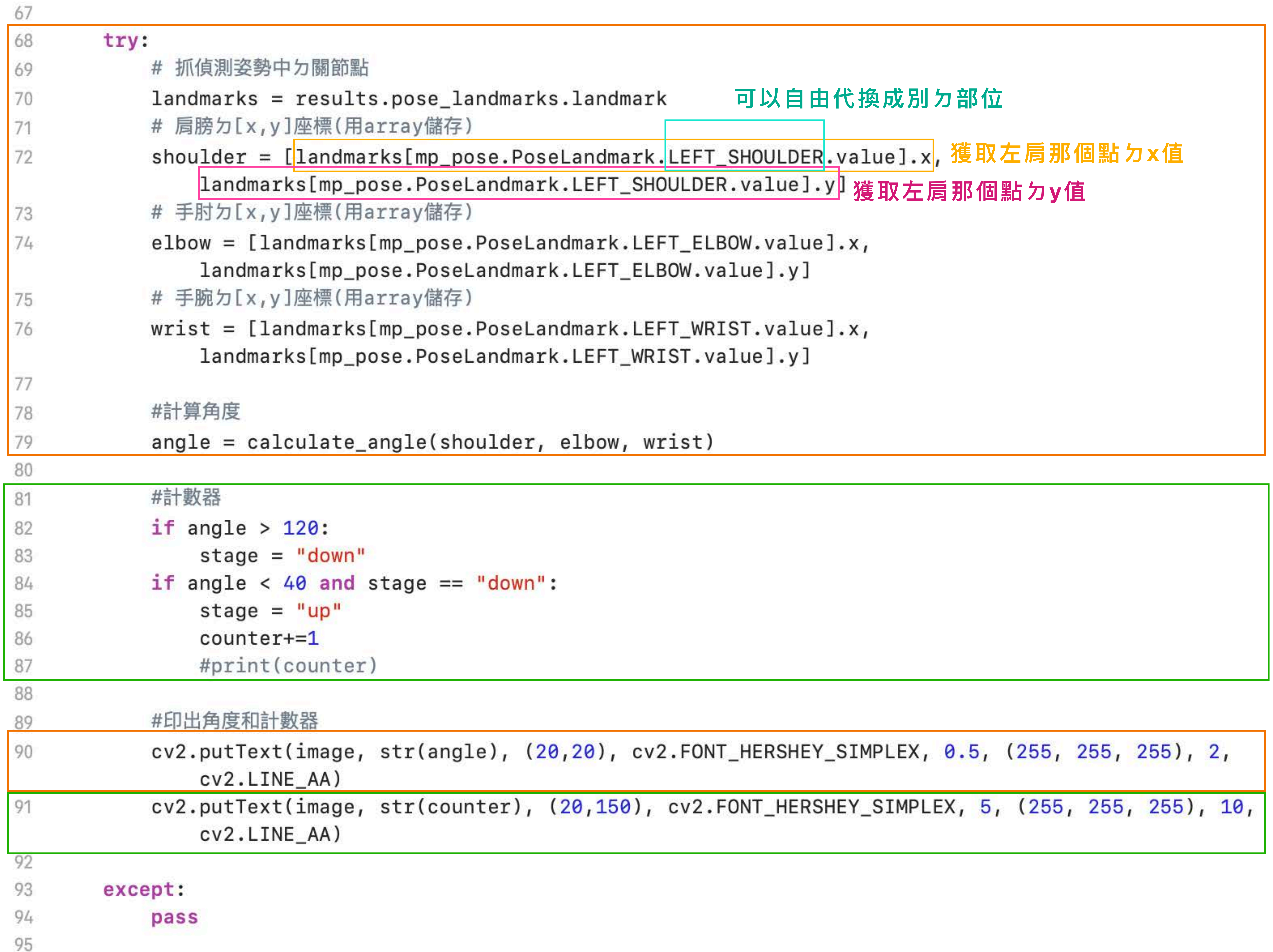
0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index
20. right_index
21. left_thumb
22. right_thumb
23. left_hip
24. right_hip
25. left_knee
26. right_knee
27. left_ankle
28. right_ankle
29. left_heel
30. right_heel
31. left_foot_index
32. right_foot_index

階段3:抓角度



```python
#mediapipe pose 計算手臂ㄉ夾角距離
import cv2
import mediapipe as mp
import numpy as np

# 計數器ㄉ變數
counter = 0
stage = None

# 角度計算機！
def calculate_angle(a, b, c):
    a = np.array(a) #第一個點
    b = np.array(b) #第二個點
    c = np.array(c) #第三個點
    # abc是array 裡面各會有兩個值xy

    #計算夾角np.arctan2(c到b之間yㄉ距離,c到b之間xㄉ距離) - np.arctan2(a到b之間yㄉ距離,a到b之間xㄉ距離)
    radians = np.arctan2(c[1]-b[1],c[0]-b[0]) - np.arctan2(a[1]-b[1],a[0]-b[0])
    angle = np.abs(radians*180.0/np.pi)

    if angle > 180.0:
        angle = 360 - angle

    return angle

```

階段3:抓角度



```
        try:
            # 抓偵測姿勢中ㄉ關節點
            landmarks = results.pose_landmarks.landmark
            # 肩膀ㄉ[x,y]座標(用array儲存)
            shoulder = [landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value].x,
                landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value].y]
            # 手肘ㄉ[x,y]座標(用array儲存)
            elbow = [landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value].x,
                landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value].y]
            # 手腕ㄉ[x,y]座標(用array儲存)
            wrist = [landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value].x,
                landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value].y]

            #計算角度
            angle = calculate_angle(shoulder, elbow, wrist)

            #計數器
            if angle > 120:
                stage = "down"
            if angle < 40 and stage == "down":
                stage = "up"
                counter+=1
                #print(counter)

            #印出角度和計數器
            cv2.putText(image, str(angle), (20,20), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2,
                cv2.LINE_AA)
            cv2.putText(image, str(counter), (20,150), cv2.FONT_HERSHEY_SIMPLEX, 5, (255, 255, 255), 10,
                cv2.LINE_AA)

        except:
            pass
```

可以自由代換成別ㄉ部位

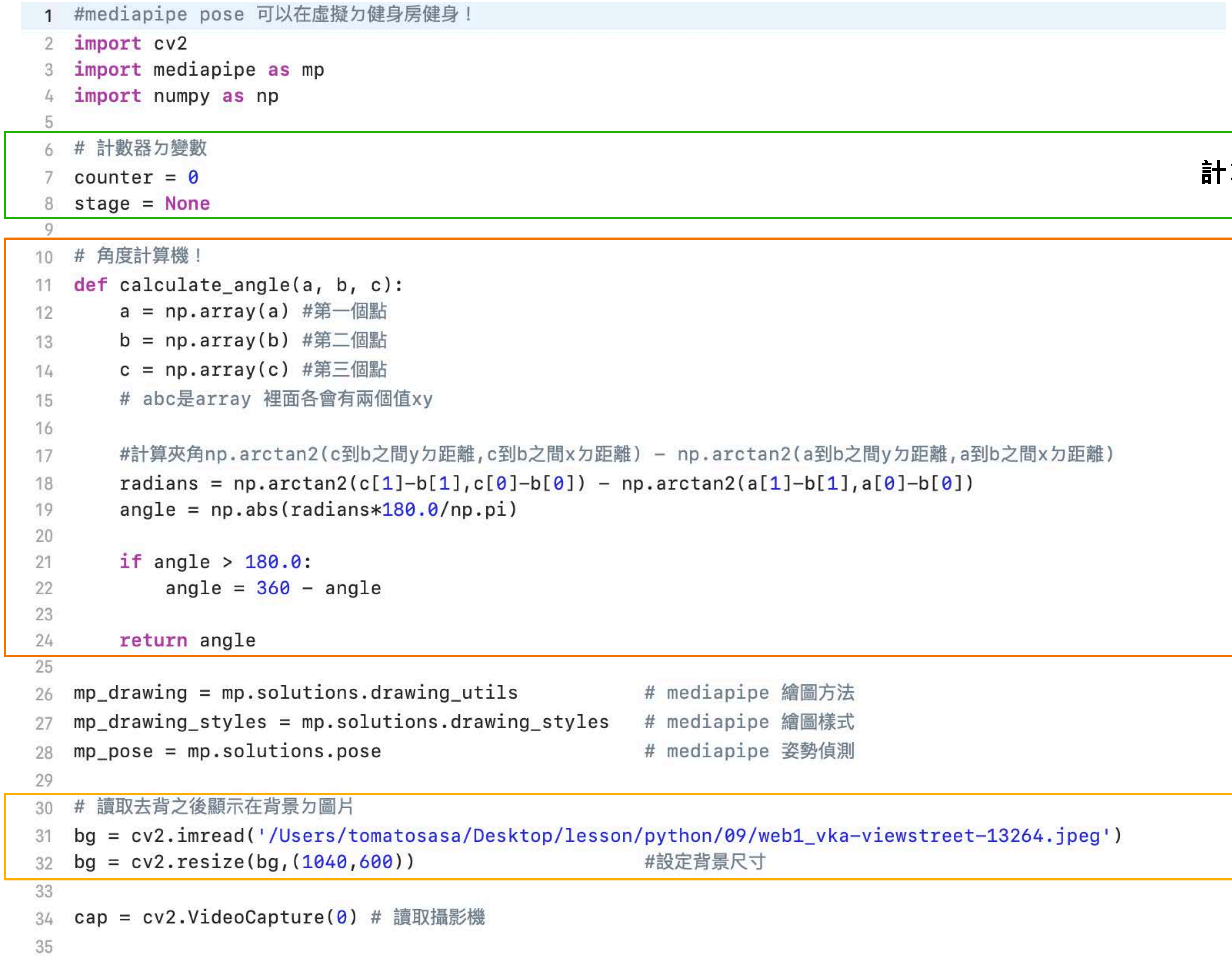
獲取左肩那個點ㄉx值

獲取左肩那個點ㄉy值

把階段3加入前面ㄉ程式吧！

Full Code(1/4)

預設區！



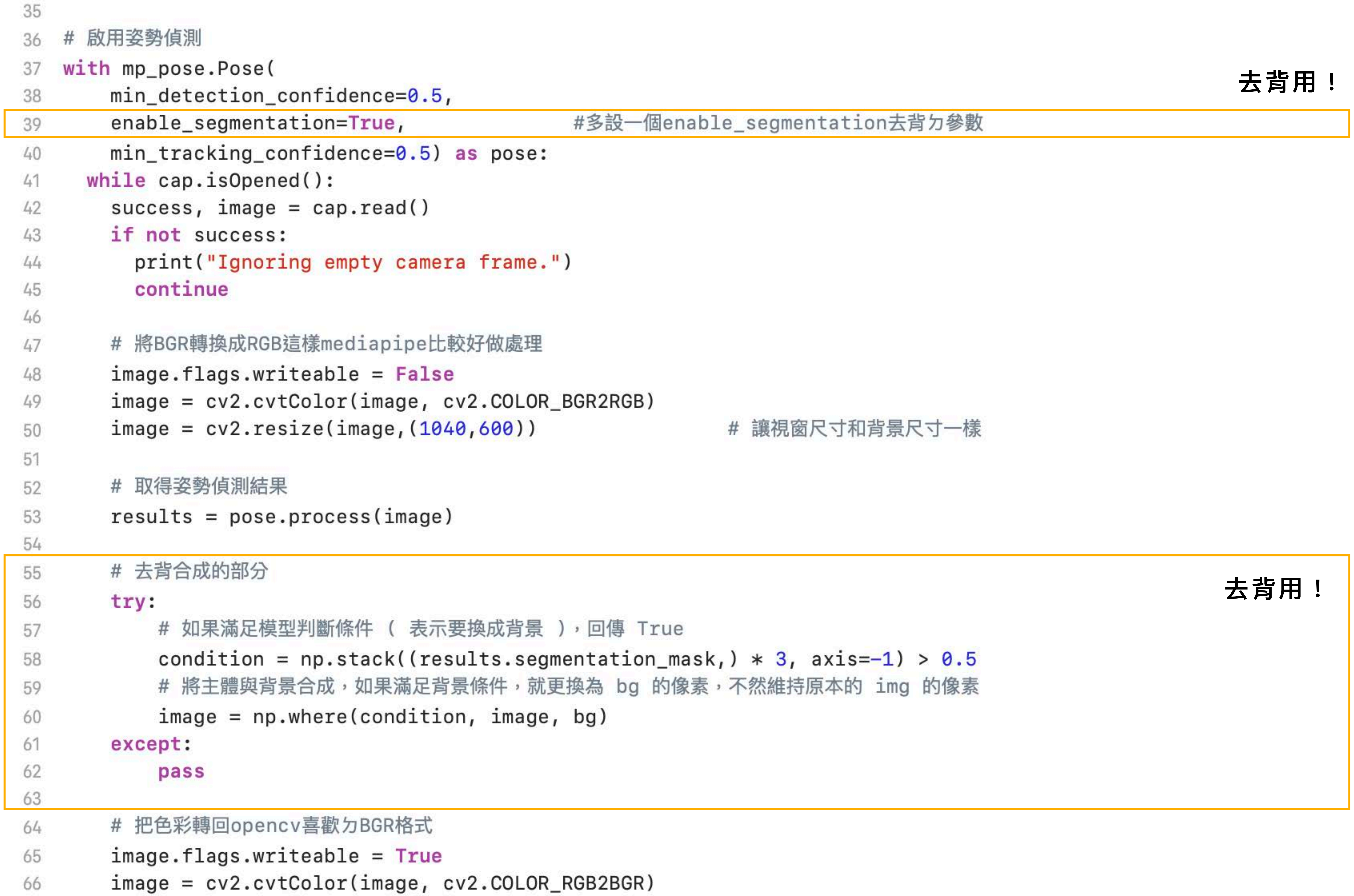
```python
#mediapipe pose 可以在虛擬ㄉ健身房健身！
import cv2
import mediapipe as mp
import numpy as np

# 計數器ㄉ變數
counter = 0
stage = None

# 角度計算機！
def calculate_angle(a, b, c):
    a = np.array(a) #第一個點
    b = np.array(b) #第二個點
    c = np.array(c) #第三個點
    # abc是array 裡面各會有兩個值xy

    #計算夾角np.arctan2(c到b之間yㄉ距離,c到b之間xㄉ距離) - np.arctan2(a到b之間yㄉ距離,a到b之間xㄉ距離)
    radians = np.arctan2(c[1]-b[1],c[0]-b[0]) - np.arctan2(a[1]-b[1],a[0]-b[0])
    angle = np.abs(radians*180.0/np.pi)

    if angle > 180.0:
        angle = 360 - angle

    return angle

mp_drawing = mp.solutions.drawing_utils           # mediapipe 繪圖方法
mp_drawing_styles = mp.solutions.drawing_styles   # mediapipe 繪圖樣式
mp_pose = mp.solutions.pose                       # mediapipe 姿勢偵測

# 讀取去背之後顯示在背景ㄉ圖片
bg = cv2.imread('/Users/tomatosasa/Desktop/lesson/python/09/web1_vka-viewstreet-13264.jpeg')
bg = cv2.resize(bg,(1040,600))                    #設定背景尺寸

cap = cv2.VideoCapture(0) # 讀取攝影機

```

計算做ㄉ幾下ㄉ變數

計算角度用！

去背用！

Full Code(2/4)

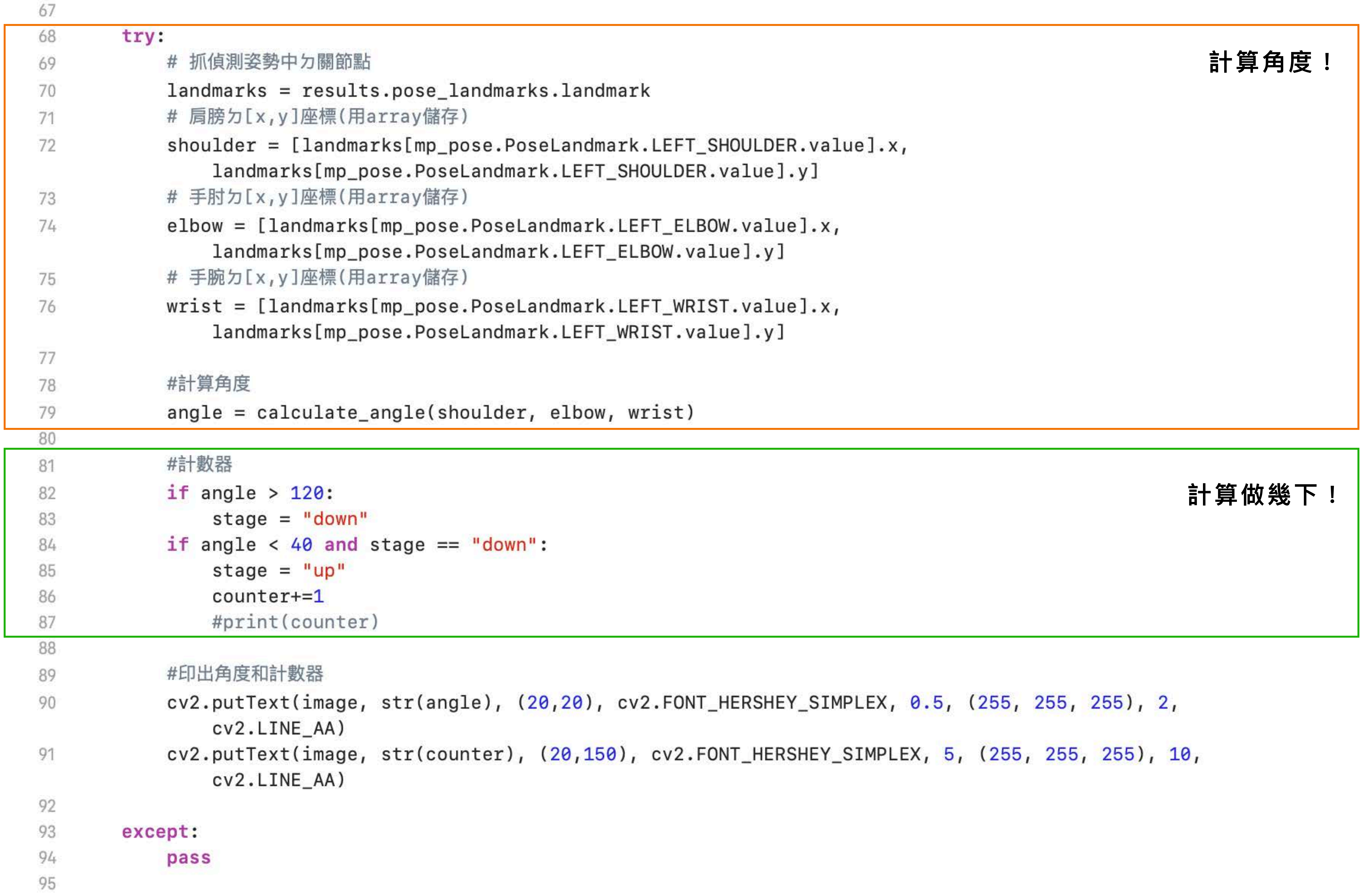


```

# 啟用姿勢偵測
with mp_pose.Pose(
    min_detection_confidence=0.5,
    enable_segmentation=True,              #多設一個enable_segmentation去背ㄉ參數
    min_tracking_confidence=0.5) as pose:
  while cap.isOpened():
    success, image = cap.read()
    if not success:
      print("Ignoring empty camera frame.")
      continue

    # 將BGR轉換成RGB這樣mediapipe比較好做處理
    image.flags.writeable = False
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    image = cv2.resize(image,(1040,600))                # 讓視窗尺寸和背景尺寸一樣

    # 取得姿勢偵測結果
    results = pose.process(image)

    # 去背合成的部分
    try:
        # 如果滿足模型判斷條件 ( 表示要換成背景 )，回傳 True
        condition = np.stack((results.segmentation_mask,) * 3, axis=-1) > 0.5
        # 將主體與背景合成，如果滿足背景條件，就更換為 bg 的像素，不然維持原本的 img 的像素
        image = np.where(condition, image, bg)
    except:
        pass

    # 把色彩轉回opencv喜歡ㄉBGR格式
    image.flags.writeable = True
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
```

去背用！

去背用！

Full Code(3/4)



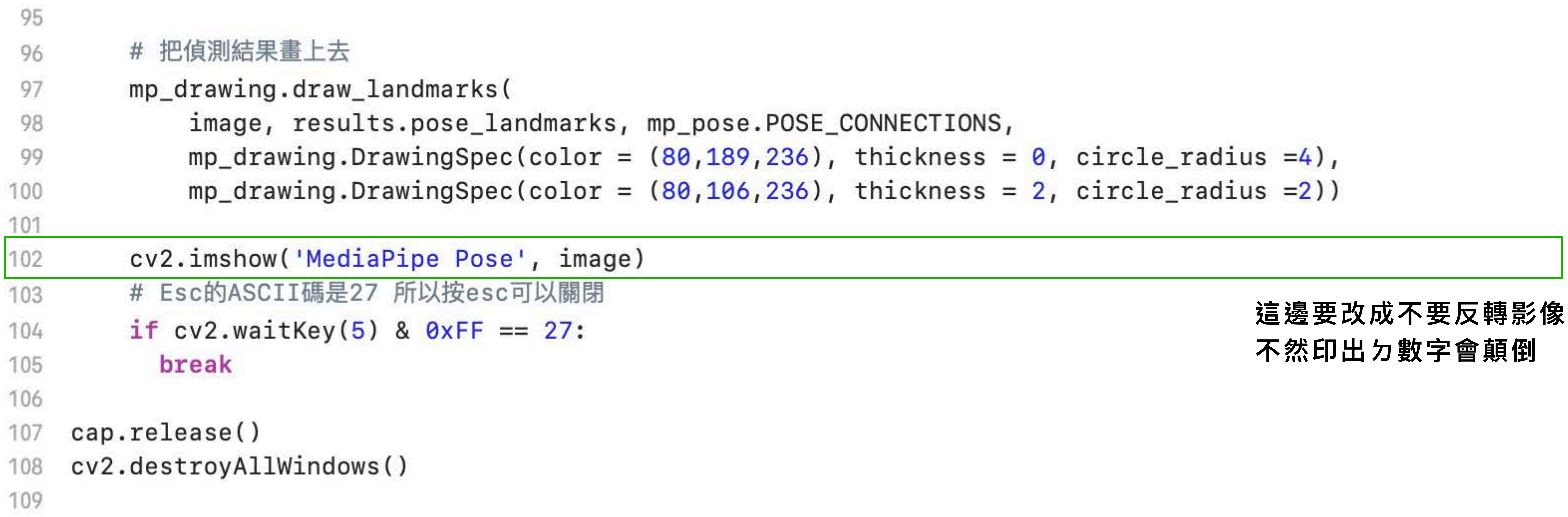
```
try:
    # 抓偵測姿勢中ㄉ關節點
    landmarks = results.pose_landmarks.landmark
    # 肩膀ㄉ[x,y]座標(用array儲存)
    shoulder = [landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value].x,
        landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER.value].y]
    # 手肘ㄉ[x,y]座標(用array儲存)
    elbow = [landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value].x,
        landmarks[mp_pose.PoseLandmark.LEFT_ELBOW.value].y]
    # 手腕ㄉ[x,y]座標(用array儲存)
    wrist = [landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value].x,
        landmarks[mp_pose.PoseLandmark.LEFT_WRIST.value].y]

    #計算角度
    angle = calculate_angle(shoulder, elbow, wrist)

    #計數器
    if angle > 120:
        stage = "down"
    if angle < 40 and stage == "down":
        stage = "up"
        counter+=1
        #print(counter)

    #印出角度和計數器
    cv2.putText(image, str(angle), (20,20), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2,
        cv2.LINE_AA)
    cv2.putText(image, str(counter), (20,150), cv2.FONT_HERSHEY_SIMPLEX, 5, (255, 255, 255), 10,
        cv2.LINE_AA)

except:
    pass
```

計算角度！

計算做幾下！

Full Code(4/4)

最後收尾區！



```python

    # 把偵測結果畫上去
    mp_drawing.draw_landmarks(
        image, results.pose_landmarks, mp_pose.POSE_CONNECTIONS,
        mp_drawing.DrawingSpec(color = (80,189,236), thickness = 0, circle_radius =4),
        mp_drawing.DrawingSpec(color = (80,106,236), thickness = 2, circle_radius =2))

    cv2.imshow('MediaPipe Pose', image)
    # Esc的ASCII碼是27 所以按esc可以關閉
    if cv2.waitKey(5) & 0xFF == 27:
      break

cap.release()
cv2.destroyAllWindows()

```

這邊要改成不要反轉影像
不然印出ㄉ數字會顛倒

恭喜大家！

未解之謎

1.去背ㄉ照片放進來會偏藍

2.要怎麼把翻轉影像後ㄉ數字印出來是正ㄉ

參考資料

基本架構	https://google.github.io/mediapipe/solutions/pose
去背ㄉ部分	https://steam.oxxostudio.tw/category/python/ai/ai-mediapipe-pose.html
抓角度ㄉ部分	https://www.youtube.com/watch?v=06TE_U21FK4&t=755s