人工智慧

手部辨識系統製作 Mediapipe

吳智鴻 教授

國立臺中教育大學 數位內容科技學系

HTTP://120.108.221.55/PROFCHWU/DCTAI

資料來源

參考網站：

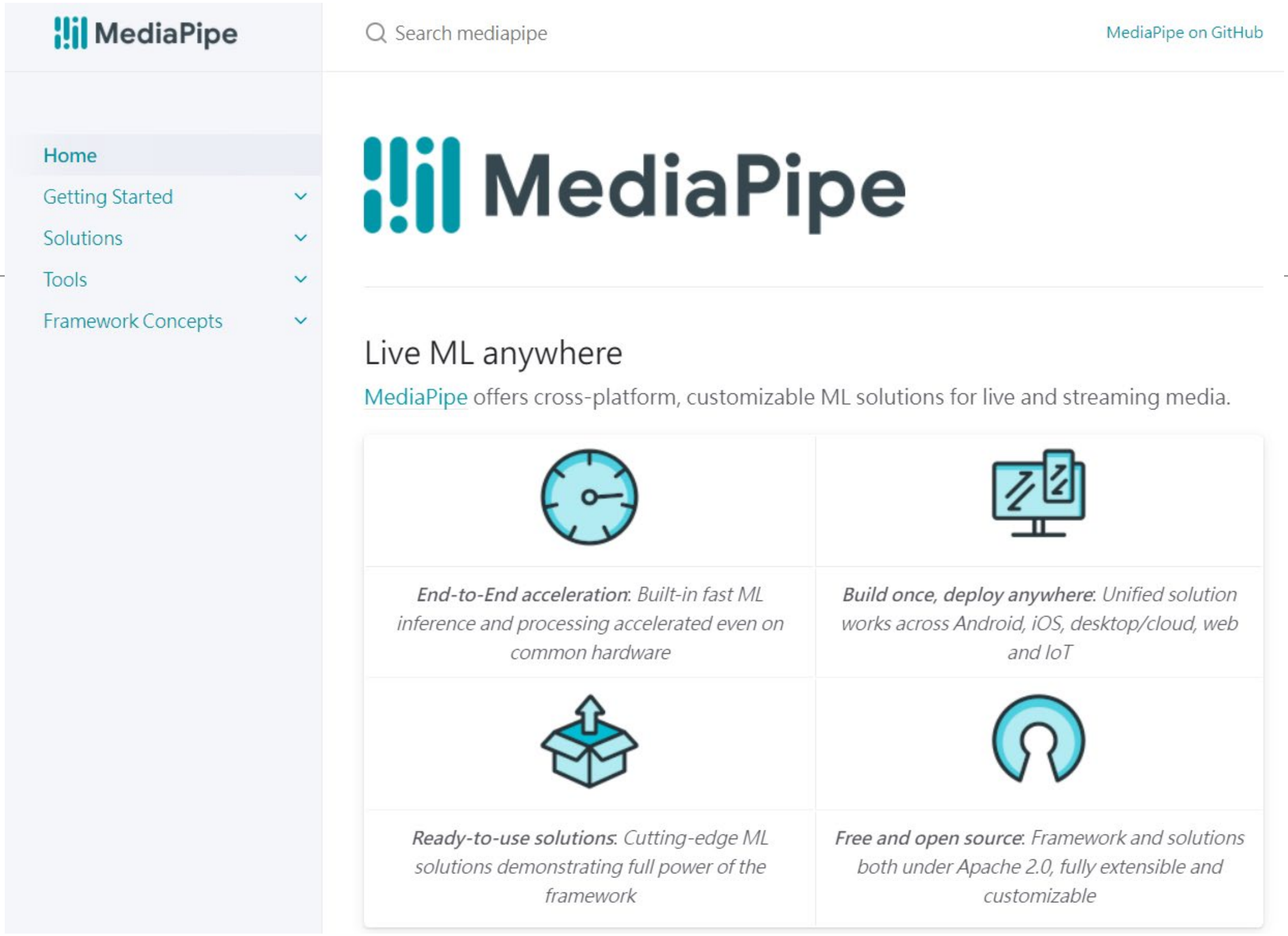
https://www.youtube.com/watch?v=x4eeX7WJIuA

程式碼:

https://github.com/hibyby/GrandmaCan_python_hand_tracking/blob/main/handTracking.py

Google MediaPipe

Google在跨平臺人工智慧工作管線框架MediaPipe，推出採用最新技術的身體姿勢追蹤功能BlazePose，能夠在手機上即時精確地定位身體姿勢關鍵點，可廣泛用於運動應用程式上，偵測健身或是瑜伽等姿勢。



Live ML anywhere

MediaPipe offers cross-platform, customizable ML solutions for live and streaming media.

End-to-End acceleration: *Built-in fast ML inference and processing accelerated even on common hardware*

Build once, deploy anywhere: *Unified solution works across Android, iOS, desktop/cloud, web and IoT*

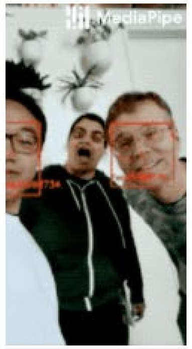
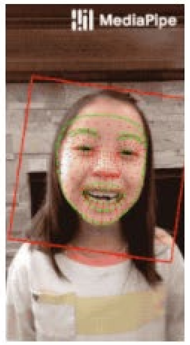
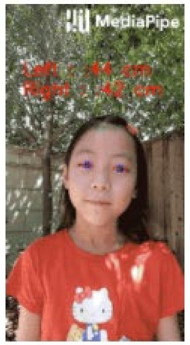
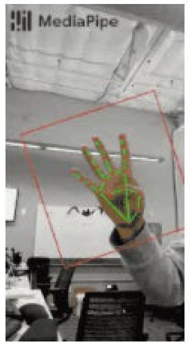
Ready-to-use solutions: *Cutting-edge ML solutions demonstrating full power of the framework*

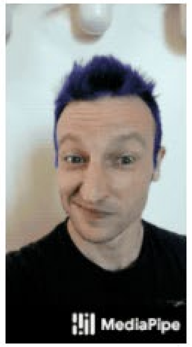
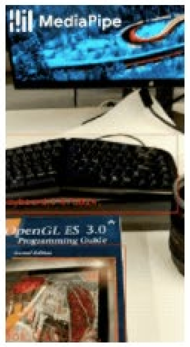
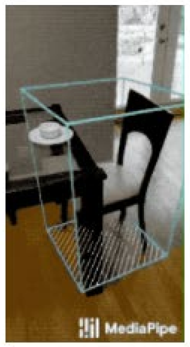
Free and open source: *Framework and solutions both under Apache 2.0, fully extensible and customizable*

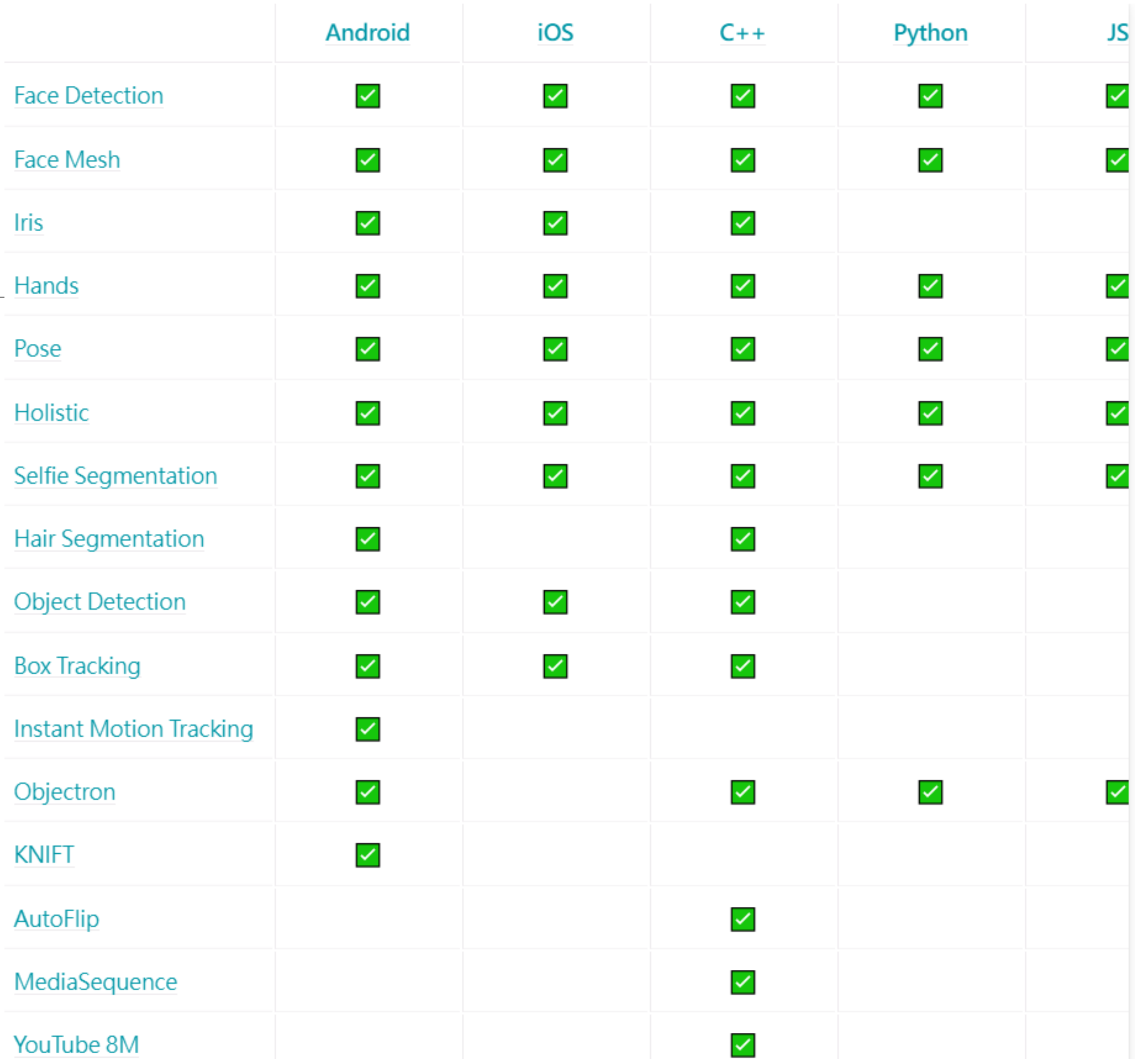
https://google.github.io/mediapipe/

提供多種的 ML Solutions

ML solutions in MediaPipe

Face Detection	Face Mesh	Iris	Hands	Pose	Holistic
					

Hair Segmentation	Object Detection	Box Tracking	Instant Motion Tracking	Objectron	KNIFT
					



	Android	iOS	C++	Python	JS
Face Detection	✓	✓	✓	✓	✓
Face Mesh	✓	✓	✓	✓	✓
Iris	✓	✓	✓		
Hands	✓	✓	✓	✓	✓
Pose	✓	✓	✓	✓	✓
Holistic	✓	✓	✓	✓	✓
Selfie Segmentation	✓	✓	✓	✓	✓
Hair Segmentation	✓		✓		
Object Detection	✓	✓	✓		
Box Tracking	✓	✓	✓		
Instant Motion Tracking	✓				
Objectron	✓		✓	✓	✓
KNIFT	✓				
AutoFlip			✓		
MediaSequence			✓		
YouTube 8M			✓		

安裝環境

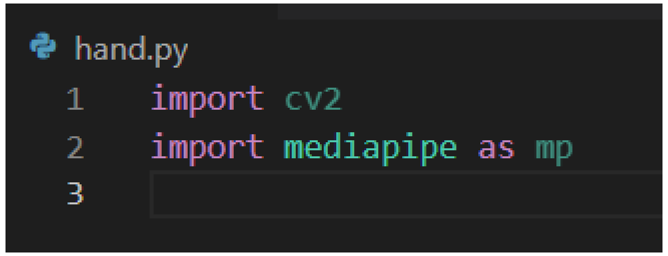
安裝opencv

```
pip install opencv-python
```

安裝mediapipe

```
pip install mediapipe
```

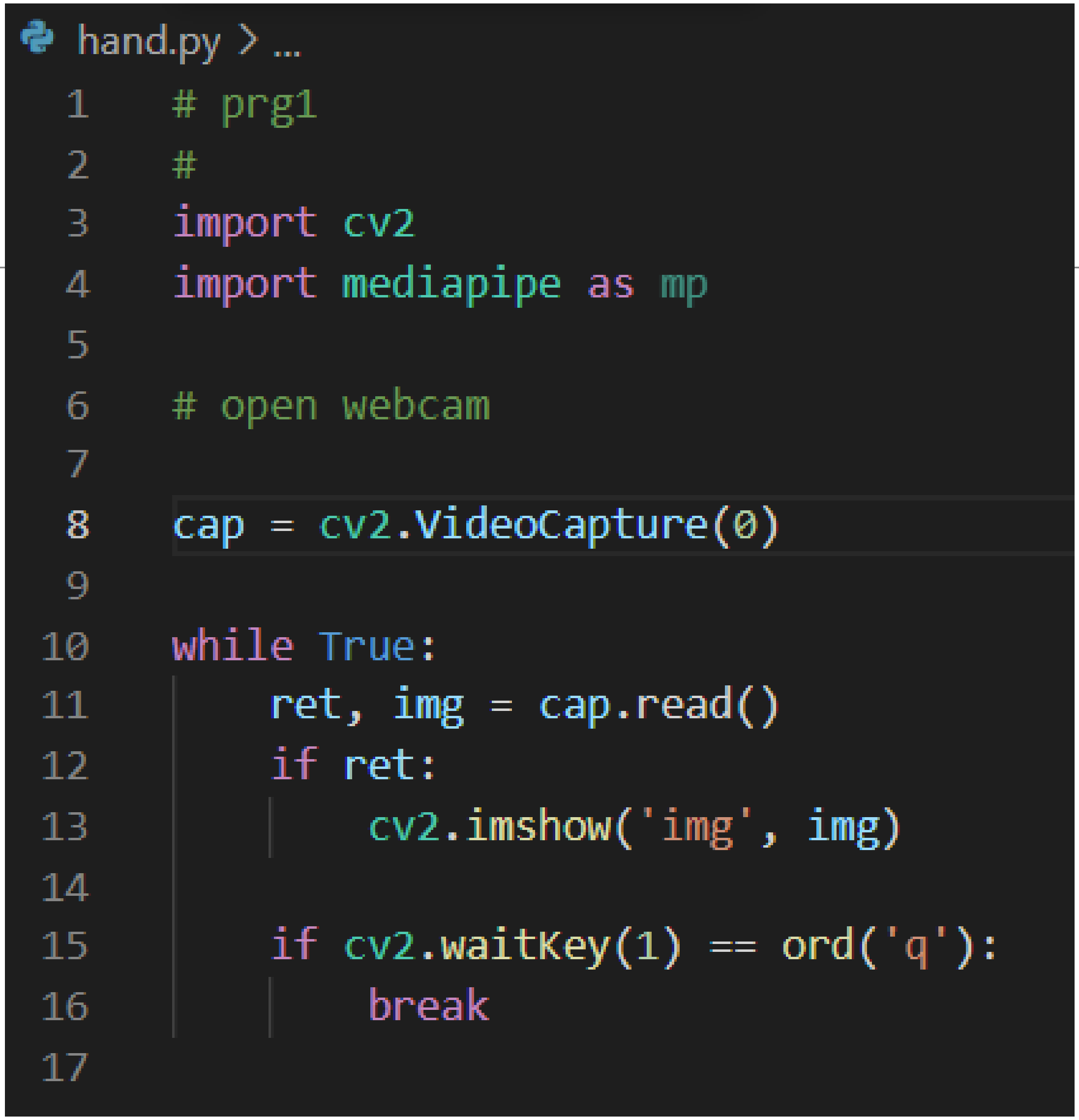
Visual Studio Code
開新檔案 hand.py



hand.py

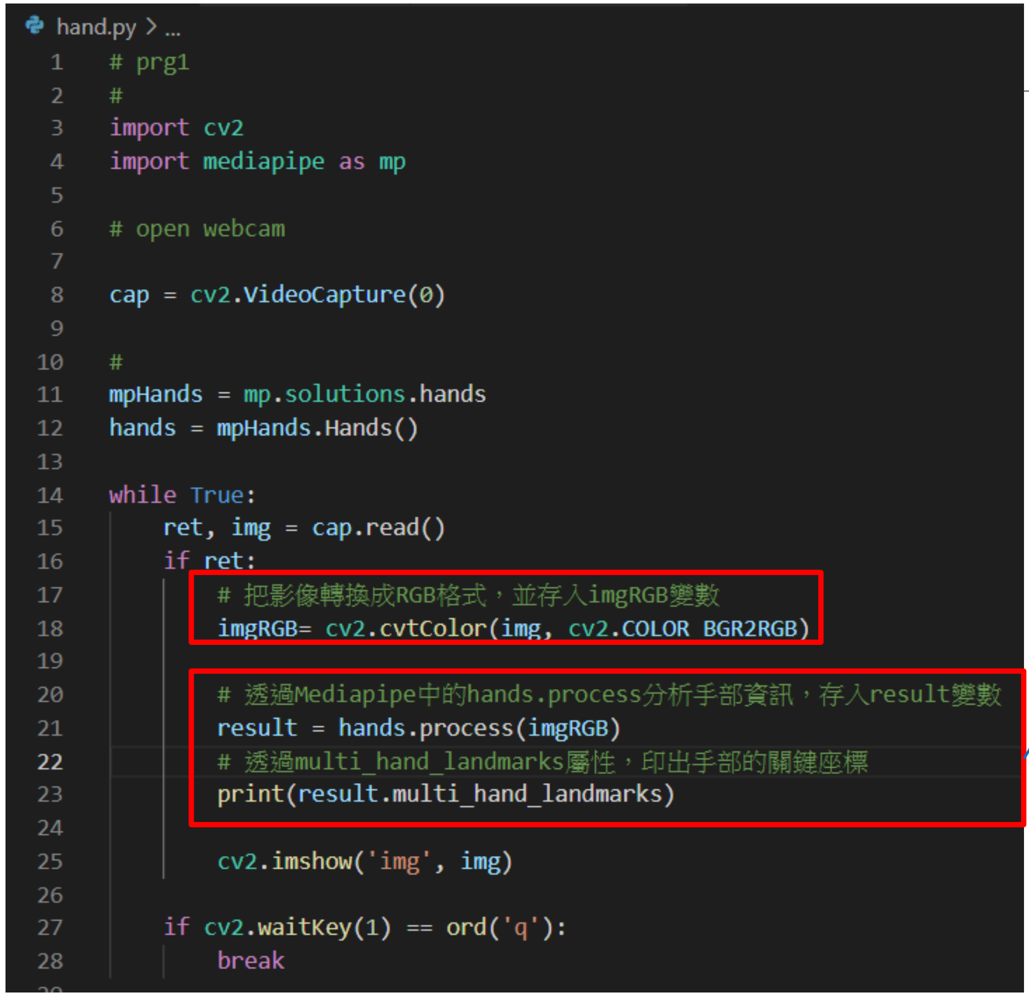
```
import cv2
import mediapipe as mp

```

Prg1
打開webcam


hand.py > ...

```python
# prg1
#
import cv2
import mediapipe as mp

# open webcam

cap = cv2.VideoCapture(0)

while True:
    ret, img = cap.read()
    if ret:
        cv2.imshow('img', img)

    if cv2.waitKey(1) == ord('q'):
        break

```

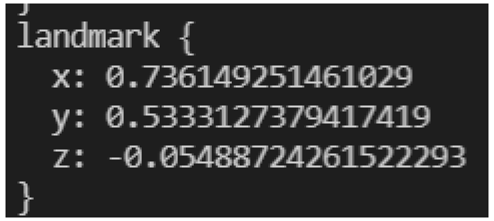
Prg2 偵測影像並顯示手部座標



hand.py > ...

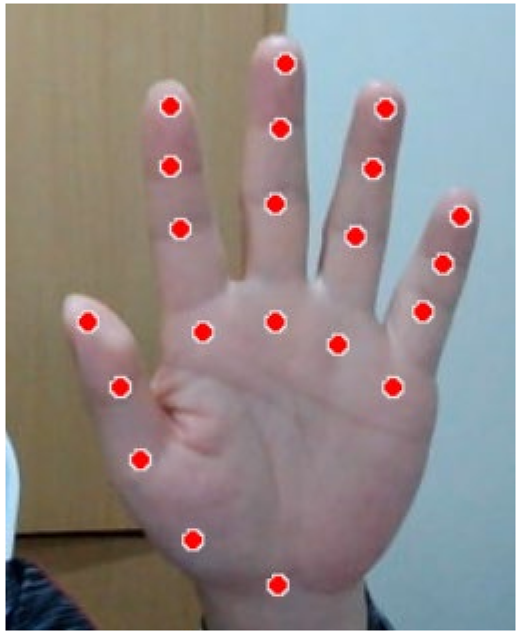
```python
# prg1
#
import cv2
import mediapipe as mp

# open webcam

cap = cv2.VideoCapture(0)

#
mpHands = mp.solutions.hands
hands = mpHands.Hands()

while True:
    ret, img = cap.read()
    if ret:
        # 把影像轉換成RGB格式，並存入imgRGB變數
        imgRGB= cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

        # 透過Mediapipe中的hands.process分析手部資訊，存入result變數
        result = hands.process(imgRGB)
        # 透過multi_hand_landmarks屬性，印出手部的關鍵座標
        print(result.multi_hand_landmarks)

        cv2.imshow('img', img)

    if cv2.waitKey(1) == ord('q'):
        break
```

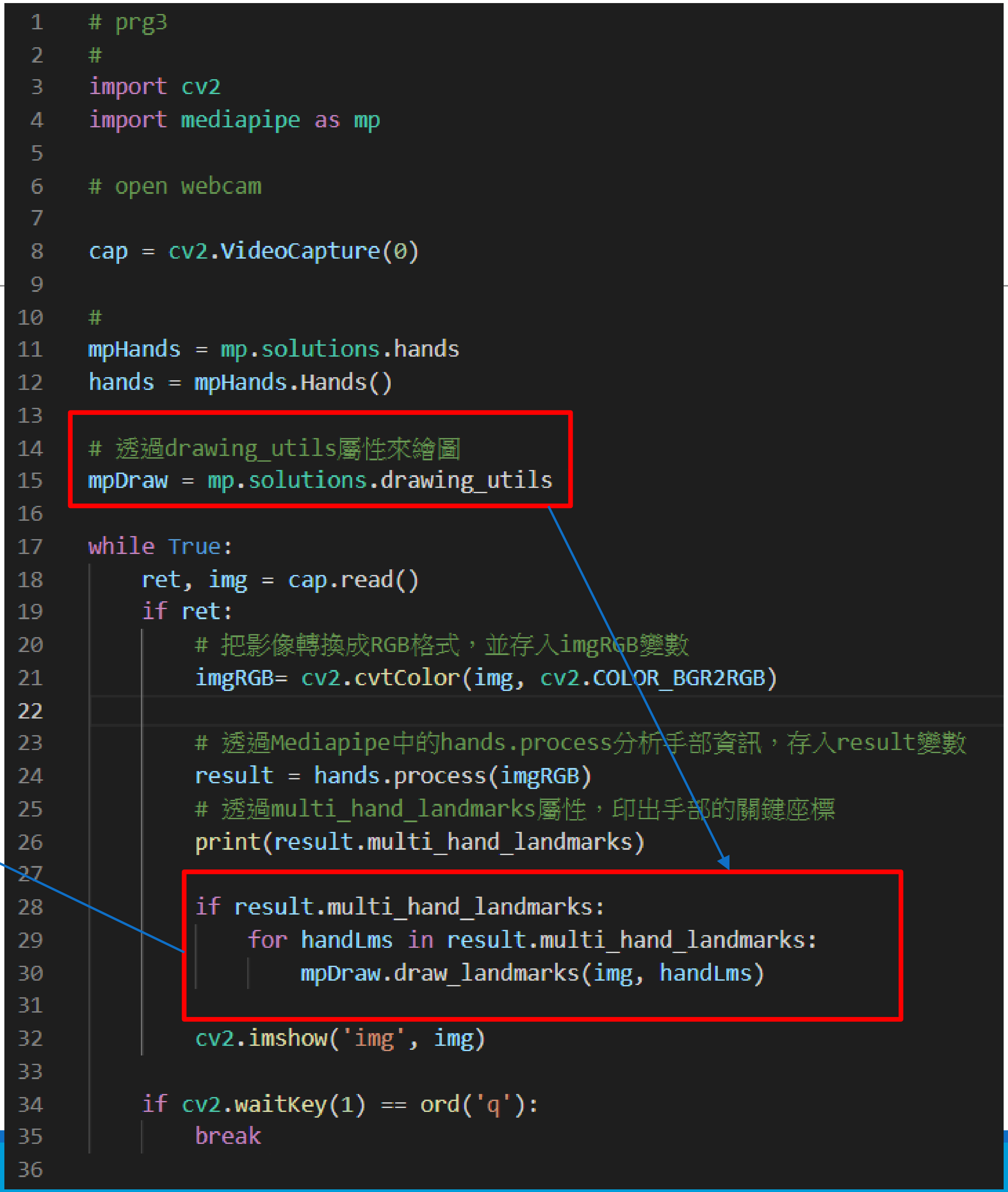
手部的關鍵點座標 (x, y, z)



```
landmark {
  x: 0.736149251461029
  y: 0.5333127379417419
  z: -0.05488724261522293
}
```

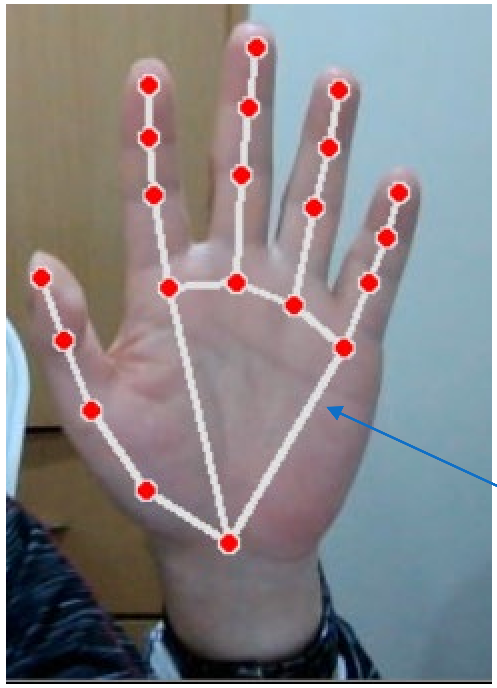
Prg3
繪製手部關鍵點

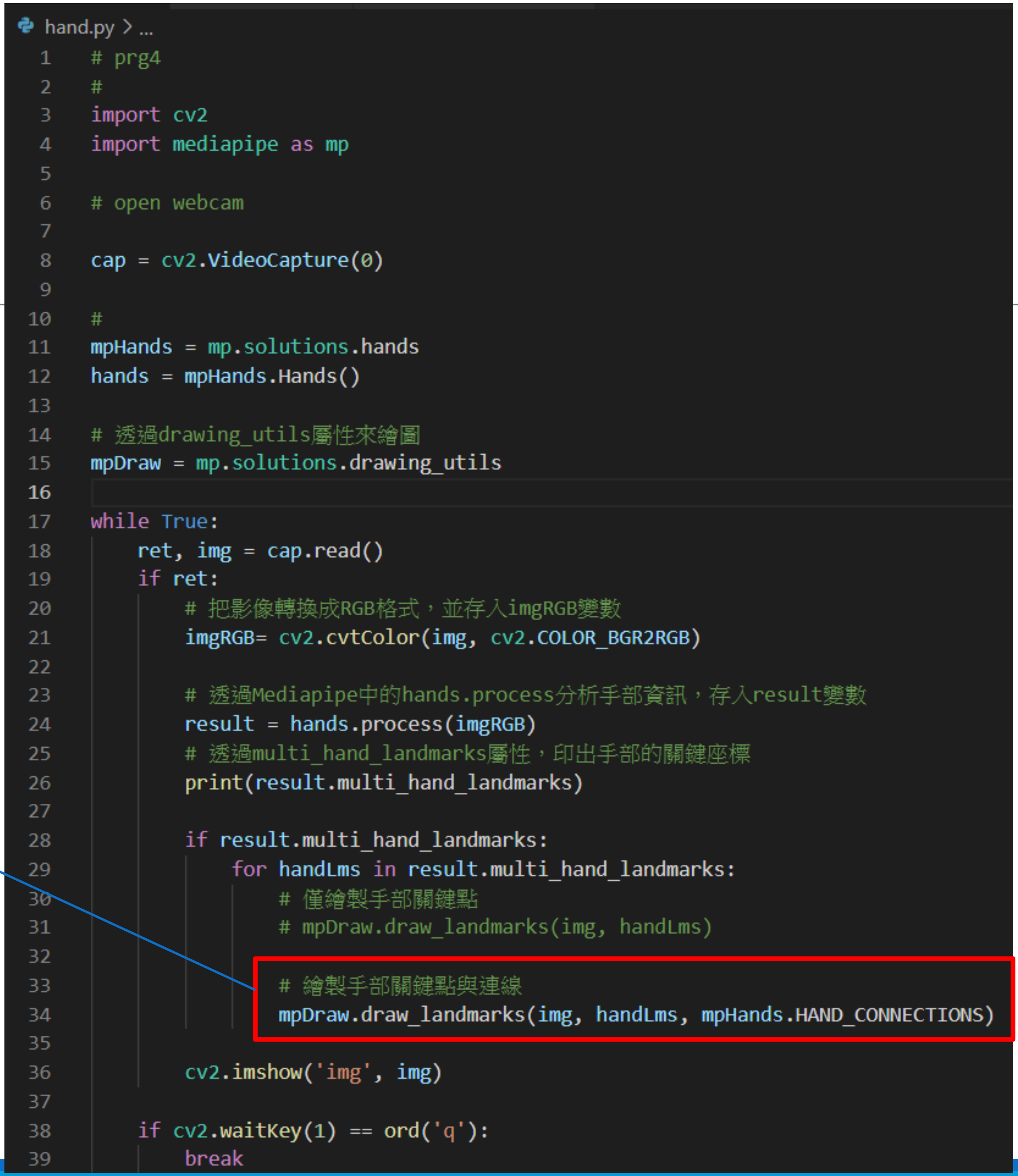




```
# prg3
#
import cv2
import mediapipe as mp

# open webcam

cap = cv2.VideoCapture(0)

#
mpHands = mp.solutions.hands
hands = mpHands.Hands()

# 透過drawing_utils屬性來繪圖
mpDraw = mp.solutions.drawing_utils

while True:
    ret, img = cap.read()
    if ret:
        # 把影像轉換成RGB格式，並存入imgRGB變數
        imgRGB= cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

        # 透過Mediapipe中的hands.process分析手部資訊，存入result變數
        result = hands.process(imgRGB)
        # 透過multi_hand_landmarks屬性，印出手部的關鍵座標
        print(result.multi_hand_landmarks)

        if result.multi_hand_landmarks:
            for handLms in result.multi_hand_landmarks:
                mpDraw.draw_landmarks(img, handLms)

        cv2.imshow('img', img)

    if cv2.waitKey(1) == ord('q'):
        break
```

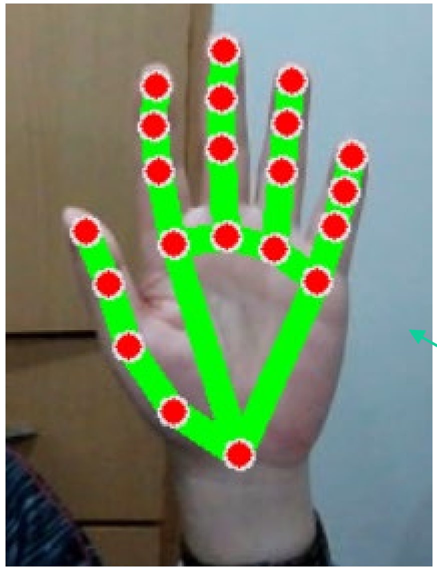
Prg4 畫出手部關鍵點連線

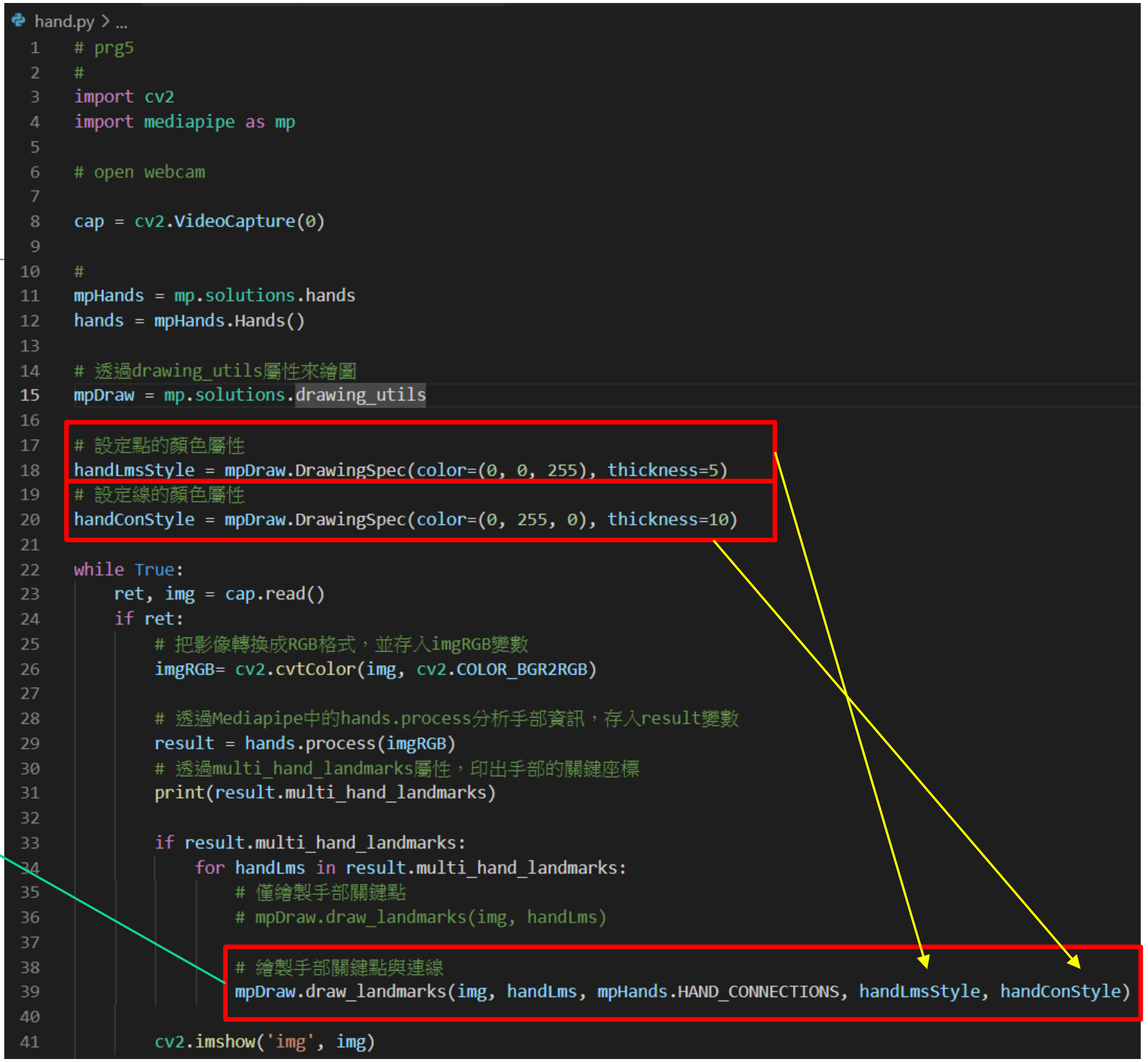



hand.py > ...

```python
# prg4
#
import cv2
import mediapipe as mp

# open webcam

cap = cv2.VideoCapture(0)

#
mpHands = mp.solutions.hands
hands = mpHands.Hands()

# 透過drawing_utils屬性來繪圖
mpDraw = mp.solutions.drawing_utils

while True:
    ret, img = cap.read()
    if ret:
        # 把影像轉換成RGB格式，並存入imgRGB變數
        imgRGB= cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

        # 透過Mediapipe中的hands.process分析手部資訊，存入result變數
        result = hands.process(imgRGB)
        # 透過multi_hand_landmarks屬性，印出手部的關鍵座標
        print(result.multi_hand_landmarks)

        if result.multi_hand_landmarks:
            for handLms in result.multi_hand_landmarks:
                # 僅繪製手部關鍵點
                # mpDraw.draw_landmarks(img, handLms)

                # 繪製手部關鍵點與連線
                mpDraw.draw_landmarks(img, handLms, mpHands.HAND_CONNECTIONS)

        cv2.imshow('img', img)

    if cv2.waitKey(1) == ord('q'):
        break
```

Prg5
設定點&線的顏色

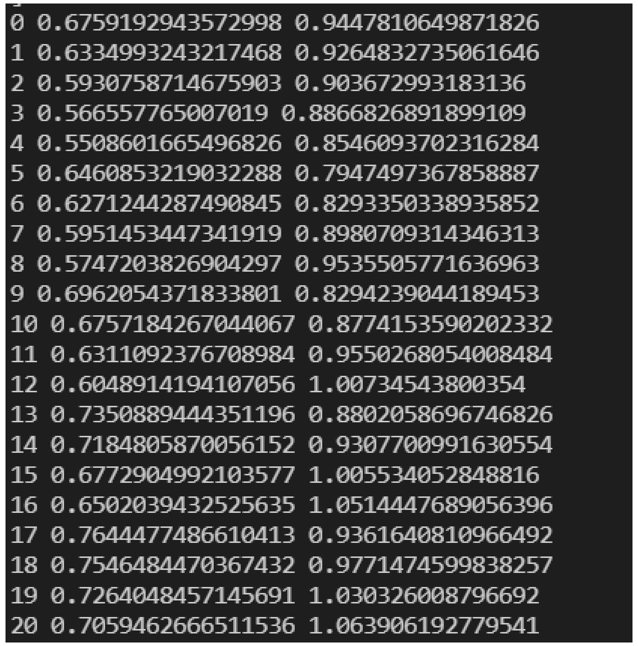




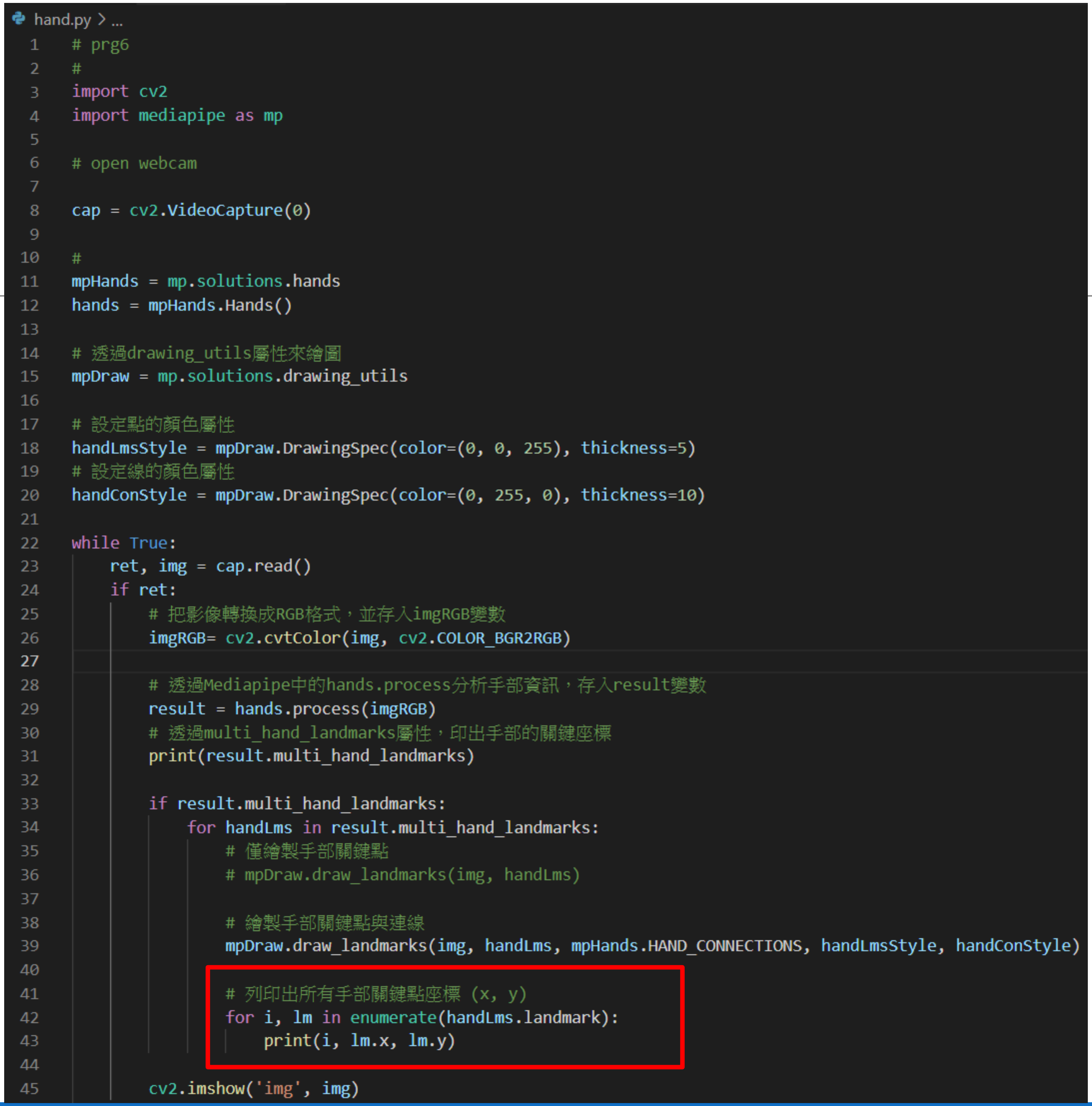
```
# prg5
#
import cv2
import mediapipe as mp

# open webcam

cap = cv2.VideoCapture(0)

#
mpHands = mp.solutions.hands
hands = mpHands.Hands()

# 透過drawing_utils屬性來繪圖
mpDraw = mp.solutions.drawing_utils

# 設定點的顏色屬性
handLmsStyle = mpDraw.DrawingSpec(color=(0, 0, 255), thickness=5)
# 設定線的顏色屬性
handConStyle = mpDraw.DrawingSpec(color=(0, 255, 0), thickness=10)

while True:
    ret, img = cap.read()
    if ret:
        # 把影像轉換成RGB格式，並存入imgRGB變數
        imgRGB= cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

        # 透過Mediapipe中的hands.process分析手部資訊，存入result變數
        result = hands.process(imgRGB)
        # 透過multi_hand_landmarks屬性，印出手部的關鍵座標
        print(result.multi_hand_landmarks)

        if result.multi_hand_landmarks:
            for handLms in result.multi_hand_landmarks:
                # 僅繪製手部關鍵點
                # mpDraw.draw_landmarks(img, handLms)

                # 繪製手部關鍵點與連線
                mpDraw.draw_landmarks(img, handLms, mpHands.HAND_CONNECTIONS, handLmsStyle, handConStyle)

        cv2.imshow('img', img)
```

Prg6
繪製手部關鍵點座標

手部關鍵點座標比率
第i個 (x, y)
Ex. 0.5表示是視窗位置的比率



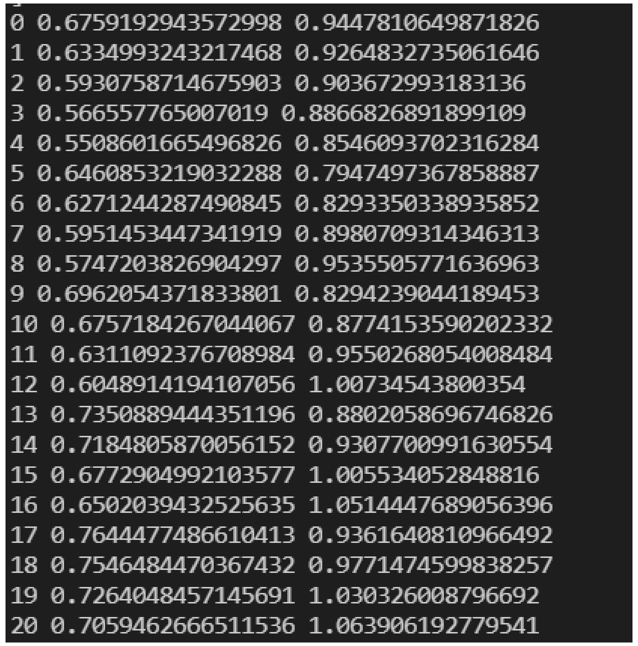
```
0 0.6759192943572998 0.9447810649871826
1 0.6334993243217468 0.9264832735061646
2 0.5930758714675903 0.903672993183136
3 0.566557765007019 0.8866826891899109
4 0.5508601665496826 0.8546093702316284
5 0.6460853219032288 0.7947497367858887
6 0.6271244287490845 0.8293350338935852
7 0.5951453447341919 0.8980709314346313
8 0.5747203826904297 0.9535505771636963
9 0.6962054371833801 0.8294239044189453
10 0.6757184267044067 0.8774153590202332
11 0.6311092376708984 0.9550268054008484
12 0.6048914194107056 1.00734543800354
13 0.7350889444351196 0.8802058696746826
14 0.7184805870056152 0.9307700991630554
15 0.6772904992103577 1.005534052848816
16 0.6502039432525635 1.0514447689056396
17 0.7644477486610413 0.9361640810966492
18 0.7546484470367432 0.9771474599838257
19 0.7264048457145691 1.030326008796692
20 0.7059462666511536 1.063906192779541
```



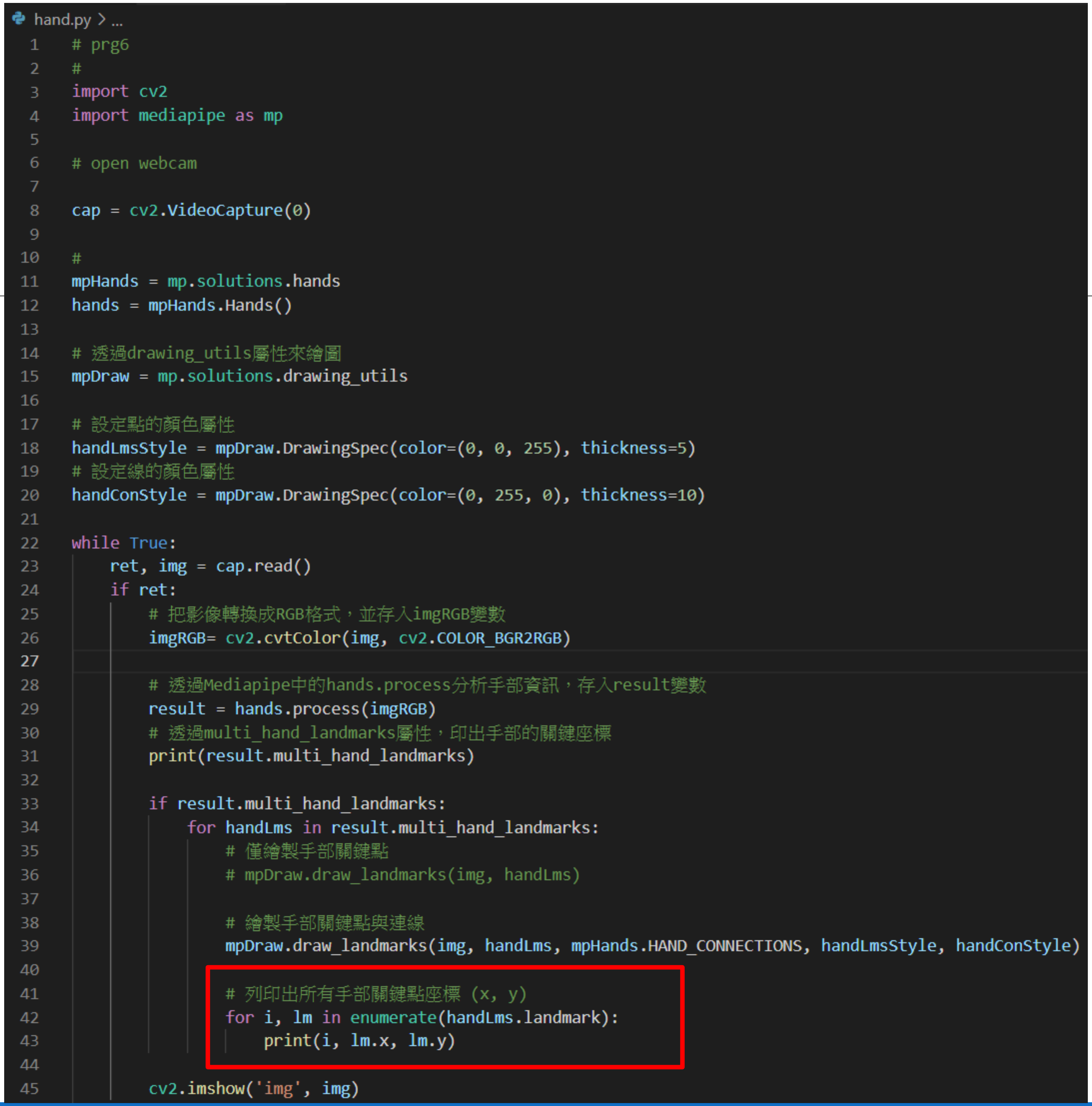
```python
# prg6
#
import cv2
import mediapipe as mp

# open webcam

cap = cv2.VideoCapture(0)

#
mpHands = mp.solutions.hands
hands = mpHands.Hands()

# 透過drawing_utils屬性來繪圖
mpDraw = mp.solutions.drawing_utils

# 設定點的顏色屬性
handLmsStyle = mpDraw.DrawingSpec(color=(0, 0, 255), thickness=5)
# 設定線的顏色屬性
handConStyle = mpDraw.DrawingSpec(color=(0, 255, 0), thickness=10)

while True:
    ret, img = cap.read()
    if ret:
        # 把影像轉換成RGB格式，並存入imgRGB變數
        imgRGB= cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

        # 透過Mediapipe中的hands.process分析手部資訊，存入result變數
        result = hands.process(imgRGB)
        # 透過multi_hand_landmarks屬性，印出手部的關鍵座標
        print(result.multi_hand_landmarks)

        if result.multi_hand_landmarks:
            for handLms in result.multi_hand_landmarks:
                # 僅繪製手部關鍵點
                # mpDraw.draw_landmarks(img, handLms)

                # 繪製手部關鍵點與連線
                mpDraw.draw_landmarks(img, handLms, mpHands.HAND_CONNECTIONS, handLmsStyle, handConStyle)

                # 列印出所有手部關鍵點座標 (x, y)
                for i, lm in enumerate(handLms.landmark):
                    print(i, lm.x, lm.y)

        cv2.imshow('img', img)
```

Prg7#1
繪製手部關鍵點座標

手部關鍵點座標比率
第i個 (x, y)
Ex. 0.5表示是視窗位置的比率



```
0 0.6759192943572998 0.9447810649871826
1 0.6334993243217468 0.9264832735061646
2 0.5930758714675903 0.903672993183136
3 0.566557765007019 0.8866826891899109
4 0.5508601665496826 0.8546093702316284
5 0.6460853219032288 0.7947497367858887
6 0.6271244287490845 0.8293350338935852
7 0.5951453447341919 0.8980709314346313
8 0.5747203826904297 0.9535505771636963
9 0.6962054371833801 0.8294239044189453
10 0.6757184267044067 0.8774153590202332
11 0.6311092376708984 0.9550268054008484
12 0.6048914194107056 1.00734543800354
13 0.7350889444351196 0.8802058696746826
14 0.7184805870056152 0.9307700991630554
15 0.6772904992103577 1.005534052848816
16 0.6502039432525635 1.0514447689056396
17 0.7644477486610413 0.9361640810966492
18 0.7546484470367432 0.9771474599838257
19 0.7264048457145691 1.030326008796692
20 0.7059462666511536 1.063906192779541
```



hand.py > ...

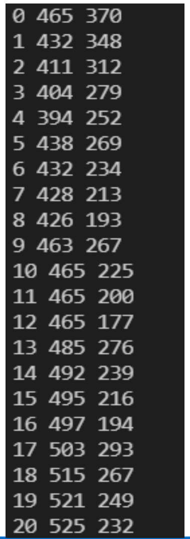
```python
# prg6
#
import cv2
import mediapipe as mp

# open webcam

cap = cv2.VideoCapture(0)

#
mpHands = mp.solutions.hands
hands = mpHands.Hands()

# 透過drawing_utils屬性來繪圖
mpDraw = mp.solutions.drawing_utils

# 設定點的顏色屬性
handLmsStyle = mpDraw.DrawingSpec(color=(0, 0, 255), thickness=5)
# 設定線的顏色屬性
handConStyle = mpDraw.DrawingSpec(color=(0, 255, 0), thickness=10)

while True:
    ret, img = cap.read()
    if ret:
        # 把影像轉換成RGB格式，並存入imgRGB變數
        imgRGB= cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

        # 透過Mediapipe中的hands.process分析手部資訊，存入result變數
        result = hands.process(imgRGB)
        # 透過multi_hand_landmarks屬性，印出手部的關鍵座標
        print(result.multi_hand_landmarks)

        if result.multi_hand_landmarks:
            for handLms in result.multi_hand_landmarks:
                # 僅繪製手部關鍵點
                # mpDraw.draw_landmarks(img, handLms)

                # 繪製手部關鍵點與連線
                mpDraw.draw_landmarks(img, handLms, mpHands.HAND_CONNECTIONS, handLmsStyle, handConStyle)

                # 列印出所有手部關鍵點座標 (x, y)
                for i, lm in enumerate(handLms.landmark):
                    print(i, lm.x, lm.y)

        cv2.imshow('img', img)
```

Prg7#2 繪製手部關鍵點座標

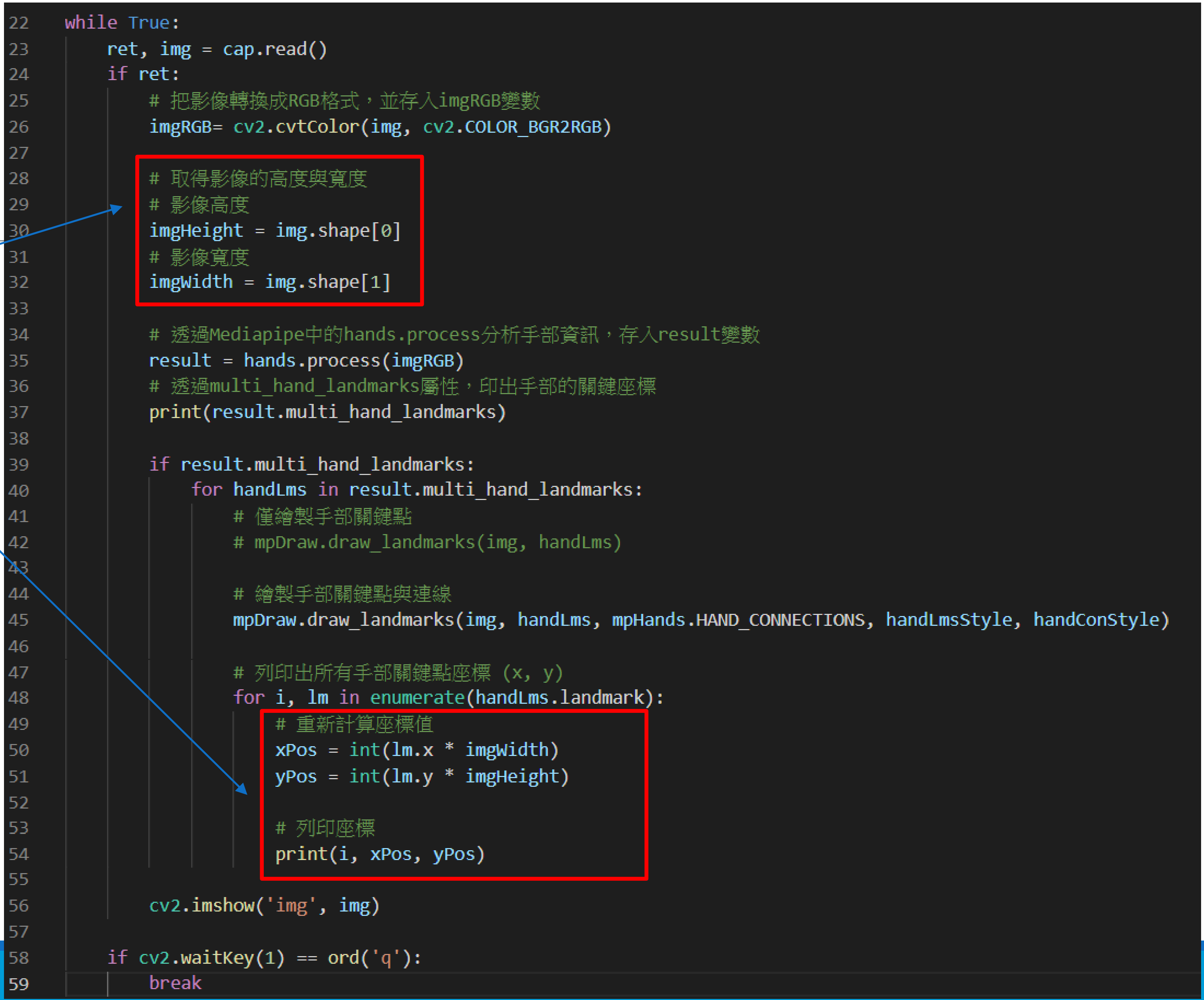
步驟1. 取得影像的高度與寬度

步驟2. 把相對位置*影像寬與高
轉換為實際的座標
lm.x * imgWidth
lm.y * Img.Height

步驟3. 計算後之真正座標結果

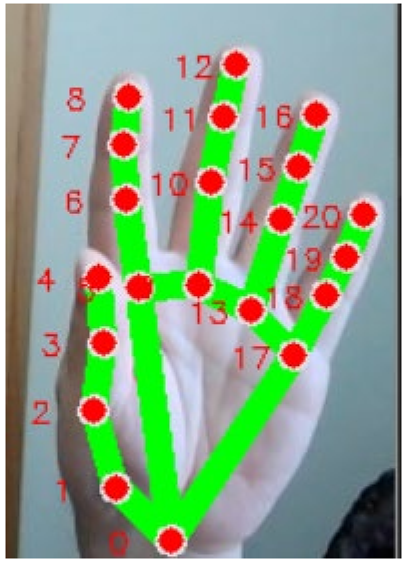


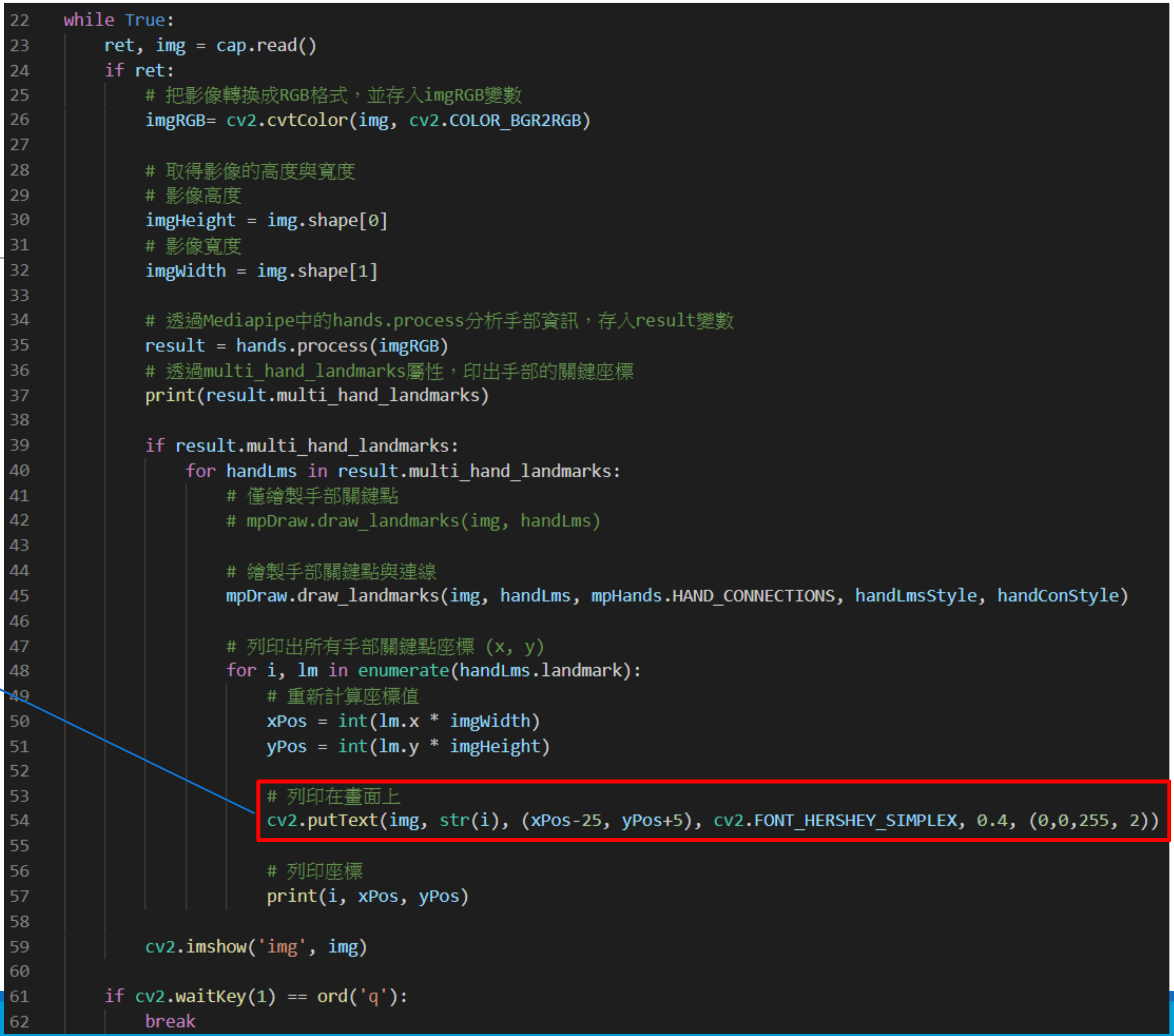
```
0 465 370
1 432 348
2 411 312
3 404 279
4 394 252
5 438 269
6 432 234
7 428 213
8 426 193
9 463 267
10 465 225
11 465 200
12 465 177
13 485 276
14 492 239
15 495 216
16 497 194
17 503 293
18 515 267
19 521 249
20 525 232
```



```python
while True:
    ret, img = cap.read()
    if ret:
        # 把影像轉換成RGB格式，並存入imgRGB變數
        imgRGB= cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

        # 取得影像的高度與寬度
        # 影像高度
        imgHeight = img.shape[0]
        # 影像寬度
        imgWidth = img.shape[1]

        # 透過Mediapipe中的hands.process分析手部資訊，存入result變數
        result = hands.process(imgRGB)
        # 透過multi_hand_landmarks屬性，印出手部的關鍵座標
        print(result.multi_hand_landmarks)

        if result.multi_hand_landmarks:
            for handLms in result.multi_hand_landmarks:
                # 僅繪製手部關鍵點
                # mpDraw.draw_landmarks(img, handLms)

                # 繪製手部關鍵點與連線
                mpDraw.draw_landmarks(img, handLms, mpHands.HAND_CONNECTIONS, handLmsStyle, handConStyle)

                # 列印出所有手部關鍵點座標 (x, y)
                for i, lm in enumerate(handLms.landmark):
                    # 重新計算座標值
                    xPos = int(lm.x * imgWidth)
                    yPos = int(lm.y * imgHeight)

                    # 列印座標
                    print(i, xPos, yPos)

        cv2.imshow('img', img)

    if cv2.waitKey(1) == ord('q'):
        break
```

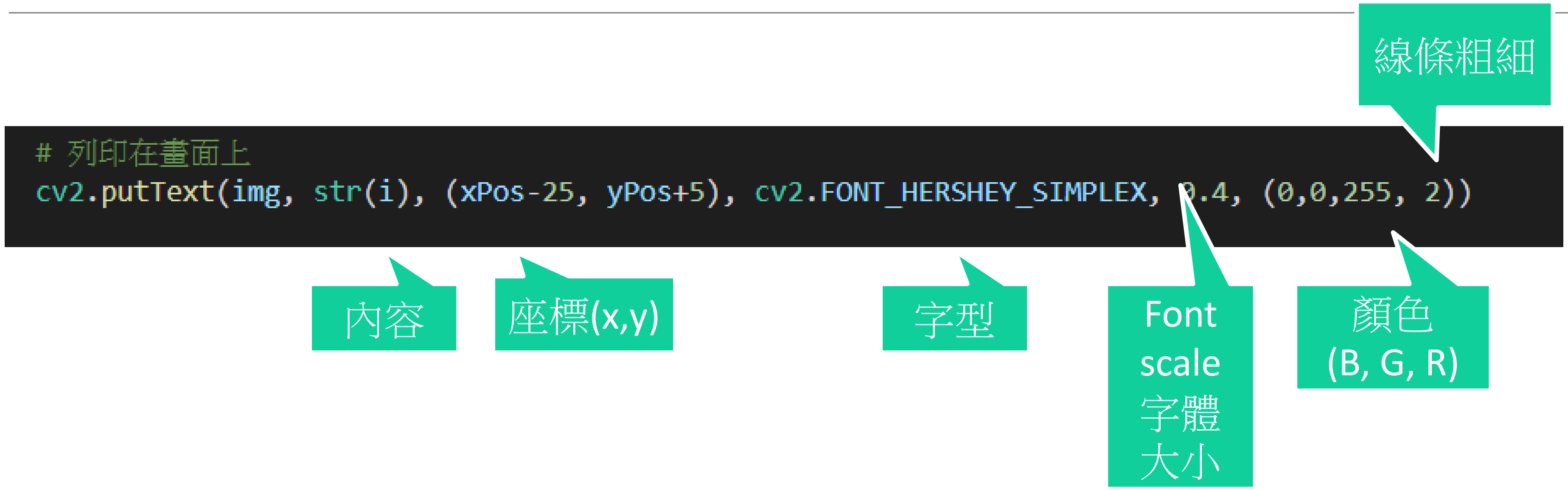
Prg7#3 繪製手部關鍵點座標





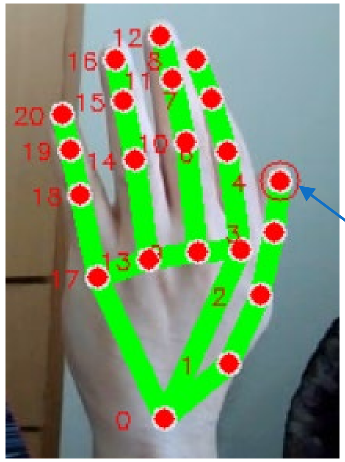
```python
while True:
    ret, img = cap.read()
    if ret:
        # 把影像轉換成RGB格式，並存入imgRGB變數
        imgRGB= cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

        # 取得影像的高度與寬度
        # 影像高度
        imgHeight = img.shape[0]
        # 影像寬度
        imgWidth = img.shape[1]

        # 透過Mediapipe中的hands.process分析手部資訊，存入result變數
        result = hands.process(imgRGB)
        # 透過multi_hand_landmarks屬性，印出手部的關鍵座標
        print(result.multi_hand_landmarks)

        if result.multi_hand_landmarks:
            for handLms in result.multi_hand_landmarks:
                # 僅繪製手部關鍵點
                # mpDraw.draw_landmarks(img, handLms)

                # 繪製手部關鍵點與連線
                mpDraw.draw_landmarks(img, handLms, mpHands.HAND_CONNECTIONS, handLmsStyle, handConStyle)

                # 列印出所有手部關鍵點座標 (x, y)
                for i, lm in enumerate(handLms.landmark):
                    # 重新計算座標值
                    xPos = int(lm.x * imgWidth)
                    yPos = int(lm.y * imgHeight)

                    # 列印在畫面上
                    cv2.putText(img, str(i), (xPos-25, yPos+5), cv2.FONT_HERSHEY_SIMPLEX, 0.4, (0,0,255, 2))

                    # 列印座標
                    print(i, xPos, yPos)

        cv2.imshow('img', img)

    if cv2.waitKey(1) == ord('q'):
        break
```

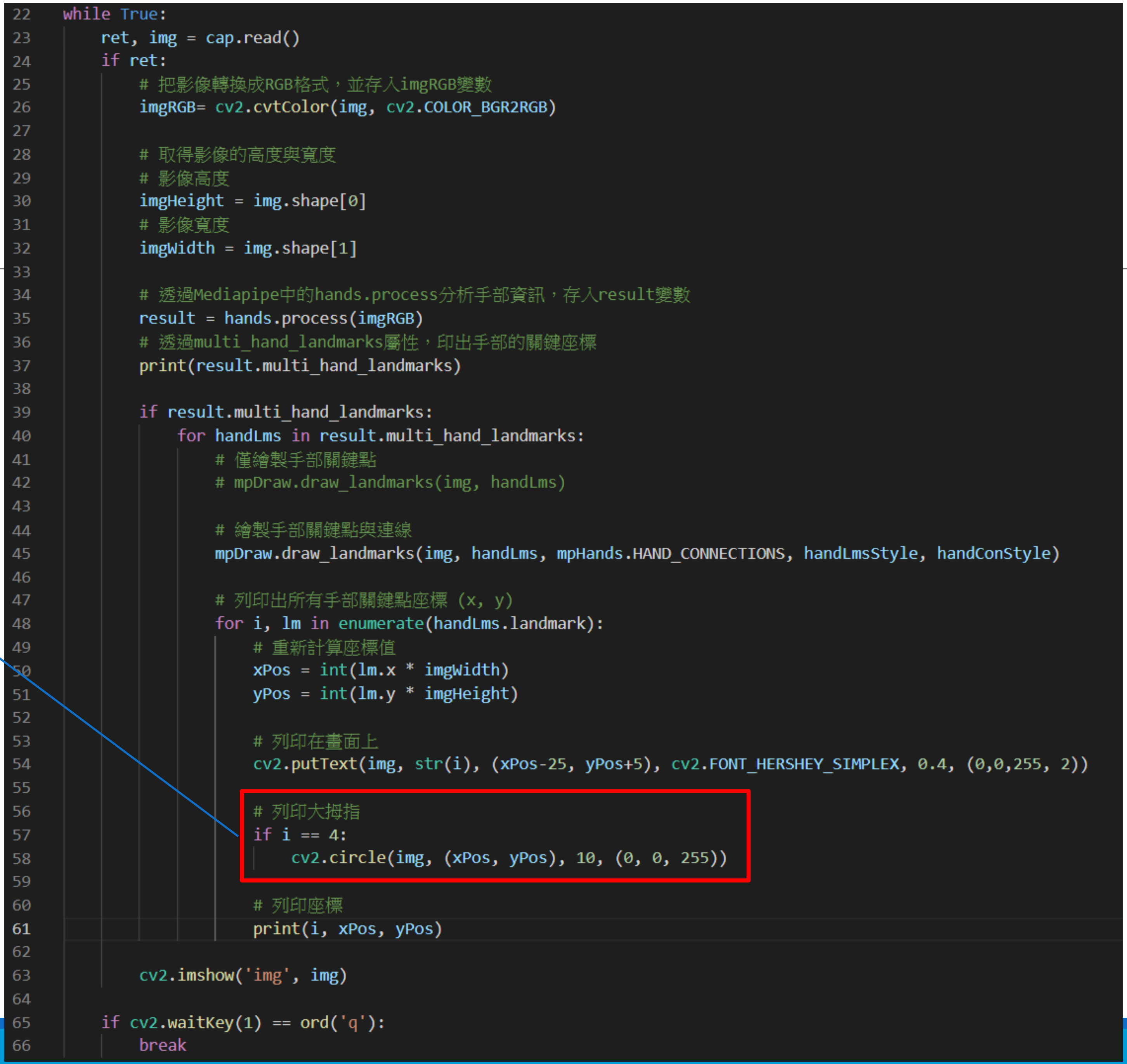
cv2提供的列印文字功能



```
# 列印在畫面上
cv2.putText(img, str(i), (xPos-25, yPos+5), cv2.FONT_HERSHEY_SIMPLEX, 0.4, (0,0,255, 2))
```

Prg8
指定設定
大拇指畫圓

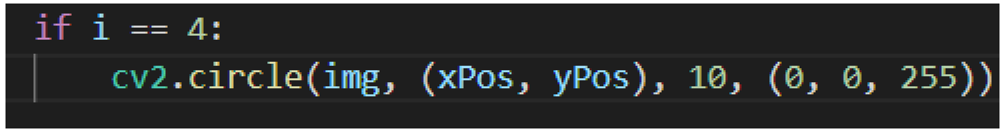




```python
while True:
    ret, img = cap.read()
    if ret:
        # 把影像轉換成RGB格式，並存入imgRGB變數
        imgRGB= cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

        # 取得影像的高度與寬度
        # 影像高度
        imgHeight = img.shape[0]
        # 影像寬度
        imgWidth = img.shape[1]

        # 透過Mediapipe中的hands.process分析手部資訊，存入result變數
        result = hands.process(imgRGB)
        # 透過multi_hand_landmarks屬性，印出手部的關鍵座標
        print(result.multi_hand_landmarks)

        if result.multi_hand_landmarks:
            for handLms in result.multi_hand_landmarks:
                # 僅繪製手部關鍵點
                # mpDraw.draw_landmarks(img, handLms)

                # 繪製手部關鍵點與連線
                mpDraw.draw_landmarks(img, handLms, mpHands.HAND_CONNECTIONS, handLmsStyle, handConStyle)

                # 列印出所有手部關鍵點座標 (x, y)
                for i, lm in enumerate(handLms.landmark):
                    # 重新計算座標值
                    xPos = int(lm.x * imgWidth)
                    yPos = int(lm.y * imgHeight)

                    # 列印在畫面上
                    cv2.putText(img, str(i), (xPos-25, yPos+5), cv2.FONT_HERSHEY_SIMPLEX, 0.4, (0,0,255, 2))

                    # 列印大拇指
                    if i == 4:
                        cv2.circle(img, (xPos, yPos), 10, (0, 0, 255))

                    # 列印座標
                    print(i, xPos, yPos)

        cv2.imshow('img', img)

    if cv2.waitKey(1) == ord('q'):
        break
```

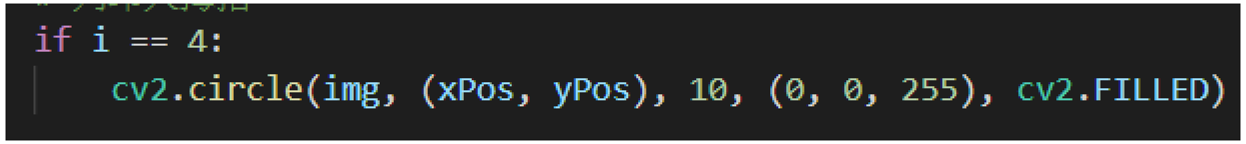
cv2畫圓功能

大拇指畫空心圓



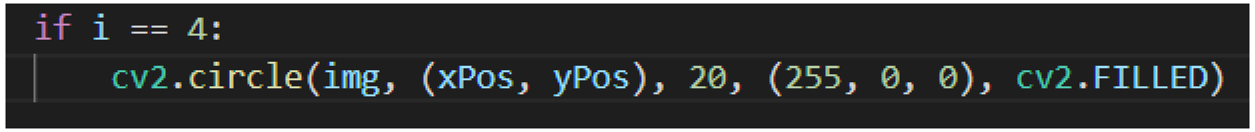
```python
if i == 4:
    cv2.circle(img, (xPos, yPos), 10, (0, 0, 255))
```

大拇指畫實心圓



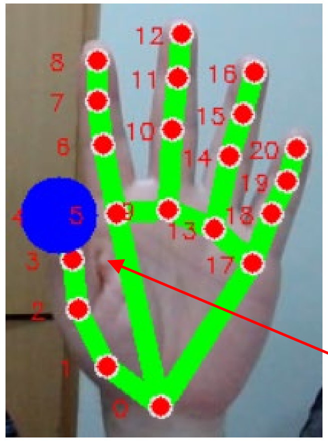
```python
if i == 4:
    cv2.circle(img, (xPos, yPos), 10, (0, 0, 255), cv2.FILLED)
```

大拇指畫大顆的藍色實心圓

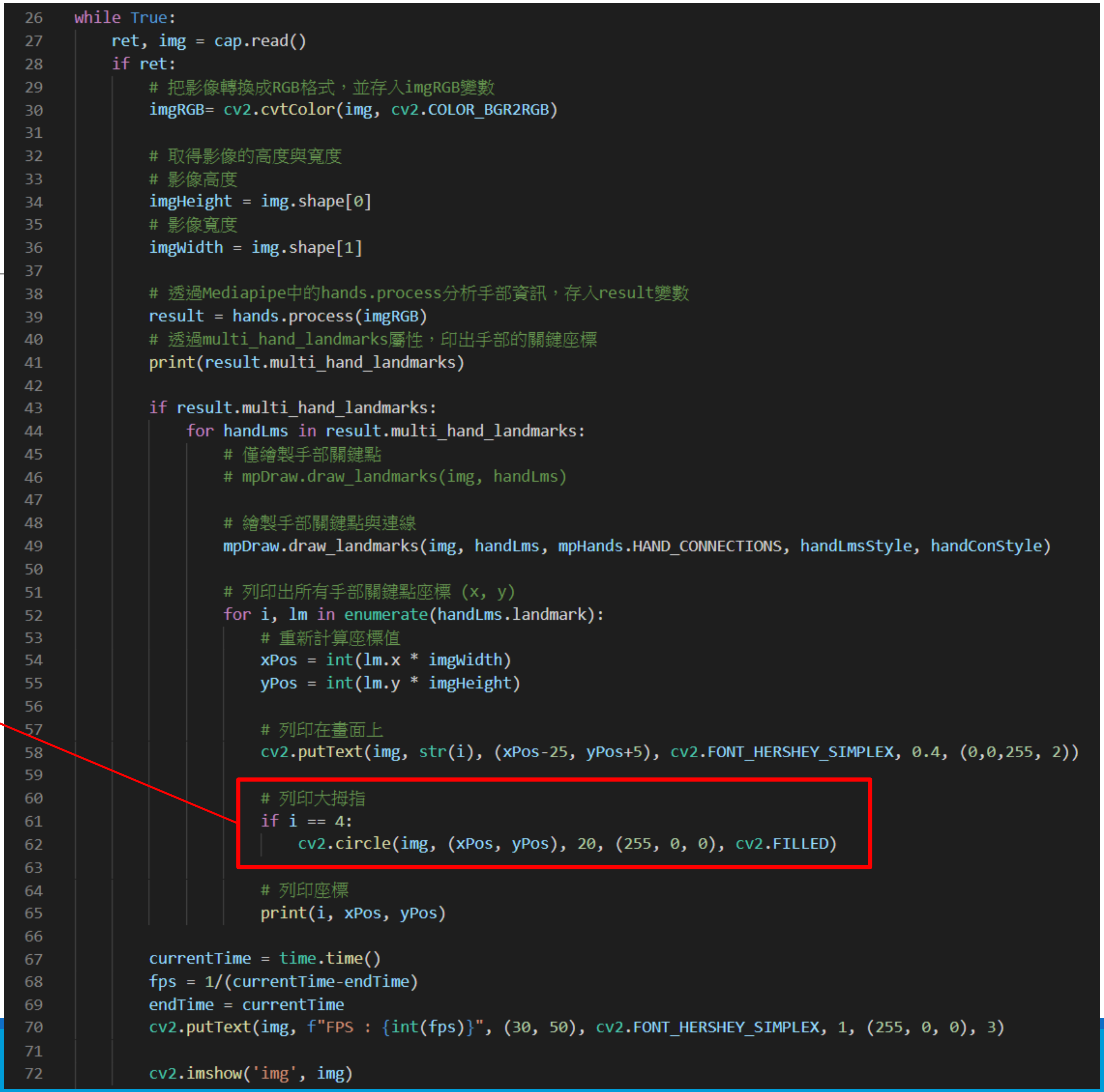


```python
if i == 4:
    cv2.circle(img, (xPos, yPos), 20, (255, 0, 0), cv2.FILLED)
```

Prg8#2
繪製大拇指實心圓

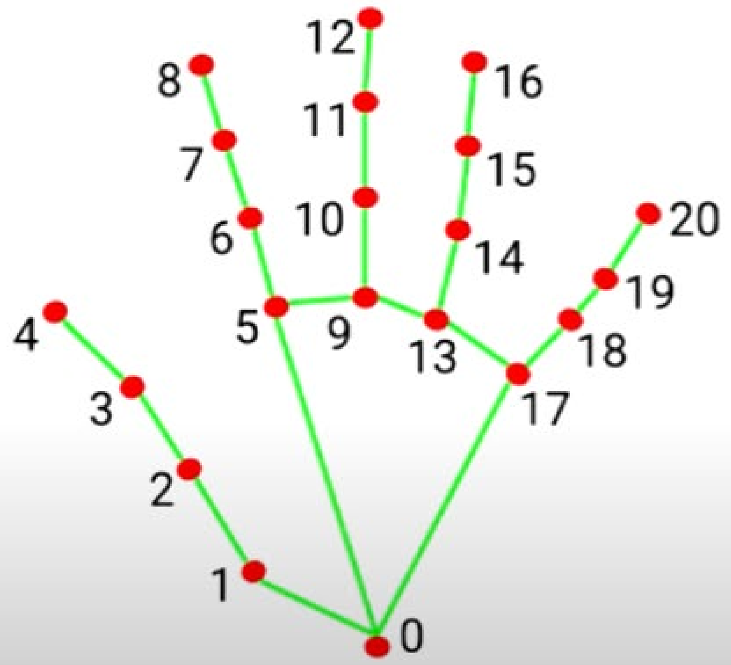


大拇指為關鍵點4



```python
while True:
    ret, img = cap.read()
    if ret:
        # 把影像轉換成RGB格式，並存入imgRGB變數
        imgRGB= cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

        # 取得影像的高度與寬度
        # 影像高度
        imgHeight = img.shape[0]
        # 影像寬度
        imgWidth = img.shape[1]

        # 透過Mediapipe中的hands.process分析手部資訊，存入result變數
        result = hands.process(imgRGB)
        # 透過multi_hand_landmarks屬性，印出手部的關鍵座標
        print(result.multi_hand_landmarks)

        if result.multi_hand_landmarks:
            for handLms in result.multi_hand_landmarks:
                # 僅繪製手部關鍵點
                # mpDraw.draw_landmarks(img, handLms)

                # 繪製手部關鍵點與連線
                mpDraw.draw_landmarks(img, handLms, mpHands.HAND_CONNECTIONS, handLmsStyle, handConStyle)

                # 列印出所有手部關鍵點座標 (x, y)
                for i, lm in enumerate(handLms.landmark):
                    # 重新計算座標值
                    xPos = int(lm.x * imgWidth)
                    yPos = int(lm.y * imgHeight)

                    # 列印在畫面上
                    cv2.putText(img, str(i), (xPos-25, yPos+5), cv2.FONT_HERSHEY_SIMPLEX, 0.4, (0,0,255, 2))

                    # 列印大拇指
                    if i == 4:
                        cv2.circle(img, (xPos, yPos), 20, (255, 0, 0), cv2.FILLED)

                    # 列印座標
                    print(i, xPos, yPos)

        currentTime = time.time()
        fps = 1/(currentTime-endTime)
        endTime = currentTime
        cv2.putText(img, f"FPS : {int(fps)}", (30, 50), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 3)

        cv2.imshow('img', img)
```

Hand Landmarks



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Fig 2. 21 hand landmarks.

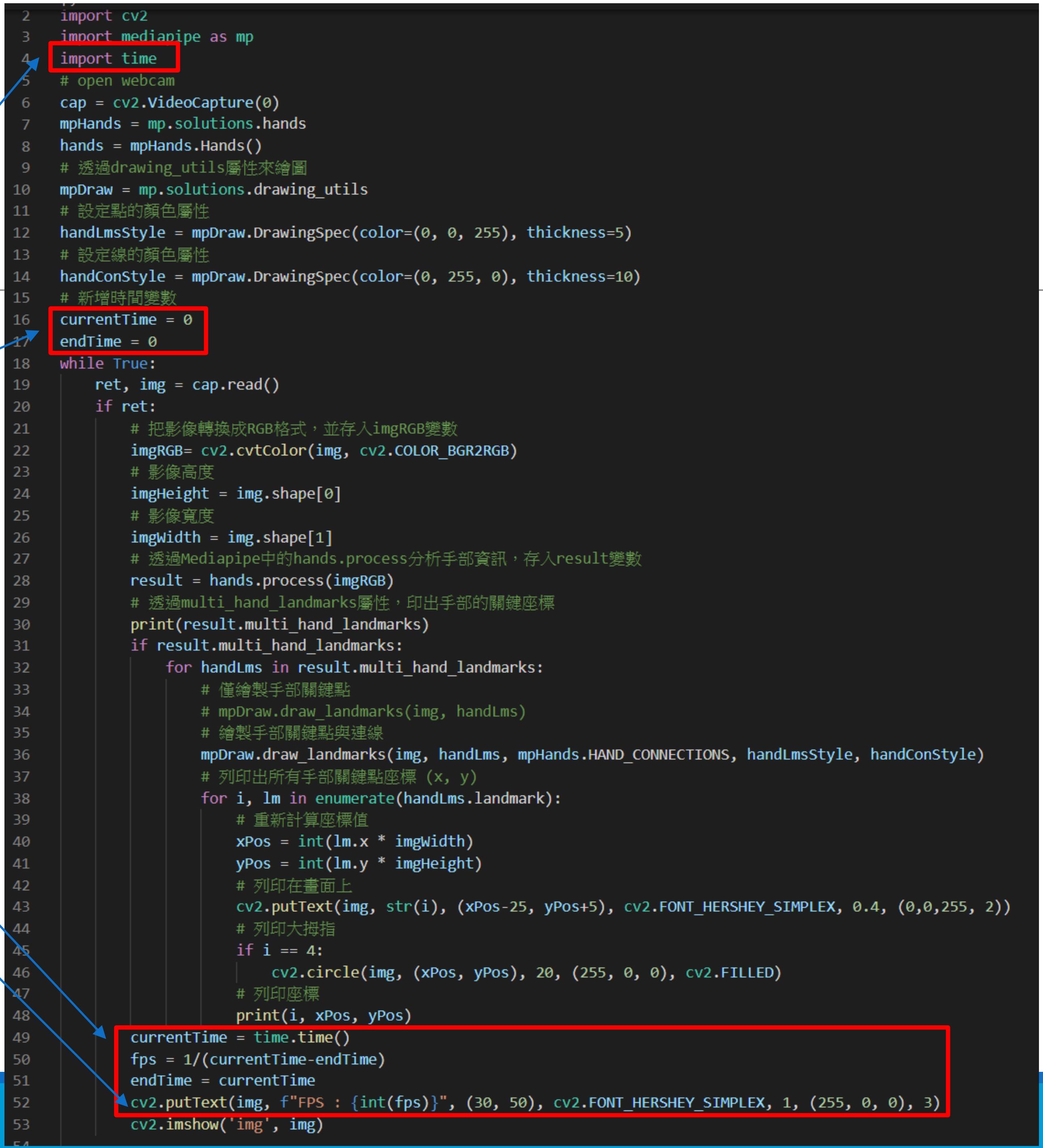
Prg9
FPS

1. 匯入 import time

2.新增兩個時間變數

3. 計算fps

4. 顯示在畫面上



```python
import cv2
import mediapipe as mp
import time
# open webcam
cap = cv2.VideoCapture(0)
mpHands = mp.solutions.hands
hands = mpHands.Hands()
# 透過drawing_utils屬性來繪圖
mpDraw = mp.solutions.drawing_utils
# 設定點的顏色屬性
handLmsStyle = mpDraw.DrawingSpec(color=(0, 0, 255), thickness=5)
# 設定線的顏色屬性
handConStyle = mpDraw.DrawingSpec(color=(0, 255, 0), thickness=10)
# 新增時間變數
currentTime = 0
endTime = 0
while True:
    ret, img = cap.read()
    if ret:
        # 把影像轉換成RGB格式，並存入imgRGB變數
        imgRGB= cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        # 影像高度
        imgHeight = img.shape[0]
        # 影像寬度
        imgWidth = img.shape[1]
        # 透過Mediapipe中的hands.process分析手部資訊，存入result變數
        result = hands.process(imgRGB)
        # 透過multi_hand_landmarks屬性，印出手部的關鍵座標
        print(result.multi_hand_landmarks)
        if result.multi_hand_landmarks:
            for handLms in result.multi_hand_landmarks:
                # 僅繪製手部關鍵點
                # mpDraw.draw_landmarks(img, handLms)
                # 繪製手部關鍵點與連線
                mpDraw.draw_landmarks(img, handLms, mpHands.HAND_CONNECTIONS, handLmsStyle, handConStyle)
                # 列印出所有手部關鍵點座標 (x, y)
                for i, lm in enumerate(handLms.landmark):
                    # 重新計算座標值
                    xPos = int(lm.x * imgWidth)
                    yPos = int(lm.y * imgHeight)
                    # 列印在畫面上
                    cv2.putText(img, str(i), (xPos-25, yPos+5), cv2.FONT_HERSHEY_SIMPLEX, 0.4, (0,0,255, 2))
                    # 列印大拇指
                    if i == 4:
                        cv2.circle(img, (xPos, yPos), 20, (255, 0, 0), cv2.FILLED)
                    # 列印座標
                    print(i, xPos, yPos)
        currentTime = time.time()
        fps = 1/(currentTime-endTime)
        endTime = currentTime
        cv2.putText(img, f"FPS : {int(fps)}", (30, 50), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 3)
        cv2.imshow('img', img)
```

資料來源

1. Youtube

https://www.youtube.com/watch?v=x4eeX7WJIuA

2. Mediapipe

https://google.github.io/mediapipe/

Congratulations.

You can design your real-time Hand Detection System now！