Motion Capture and Future Interaction Technology Research

Introduction to MoCap & Developer Environments

Speaker: Fu-Song Hsu

Developer Environments

- Python Virtual Environments
  - **Anaconda**

- Code editor
  - **Juputer notebook**
  - Visual Studio Code (VS Cdoe)
  - Sublime

How to Install

Anaconda Python

on Windows and MacOS ?

ANACONDA

Google

anaconda

影片 圖片 安裝 下載 中文 下載 是 什麼 教學 官網 蛇

約有 144,000,000 項結果 (搜尋時間：0.29 秒)

Anaconda
https://www.anaconda.com › download

Free Download

Conda is an open-source package and environment management system that runs on Windows, macOS, and Linux. Conda quickly installs, runs, and updates packages and ...

Code in the Cloud · Open Source · Excel · Enterprise

Anaconda
https://www.anaconda.com

Anaconda | The World's Most Popular Data Science Platform

Anaconda is the birthplace of Python data science. We are a movement of data scientists, data-driven enterprises, and open source communities.

ANACONDA.

Distribution

Free Download*

Register to get everything you need to get started on your workstation including Cloud Notebooks, Navigator, AI Assistant, Learning and more.

- Easily search and install thousands of data science, machine learning, and AI packages
- Manage packages and environments from a desktop application or work from the command line
- Deploy across hardware and software platforms
- Distribution installation on Windows, MacOS, or Linux

Provide email to download Distribution

Email Address:

Agree to receive communication from Anaconda regarding relevant content, products, and services. I understand that I can revoke this consent here at any time.

By continuing, I agree to Anaconda's Privacy Policy and Terms of Service.

Submit >

Skip registration

ANACONDA.

Anaconda Distribution Download

Thank you for choosing Anaconda and Conda packages.

Continue with your download.

Download Now

ANACONDA.

Windows

Python 3.12

- 64-Bit Graphical Installer (912.3M)

Mac

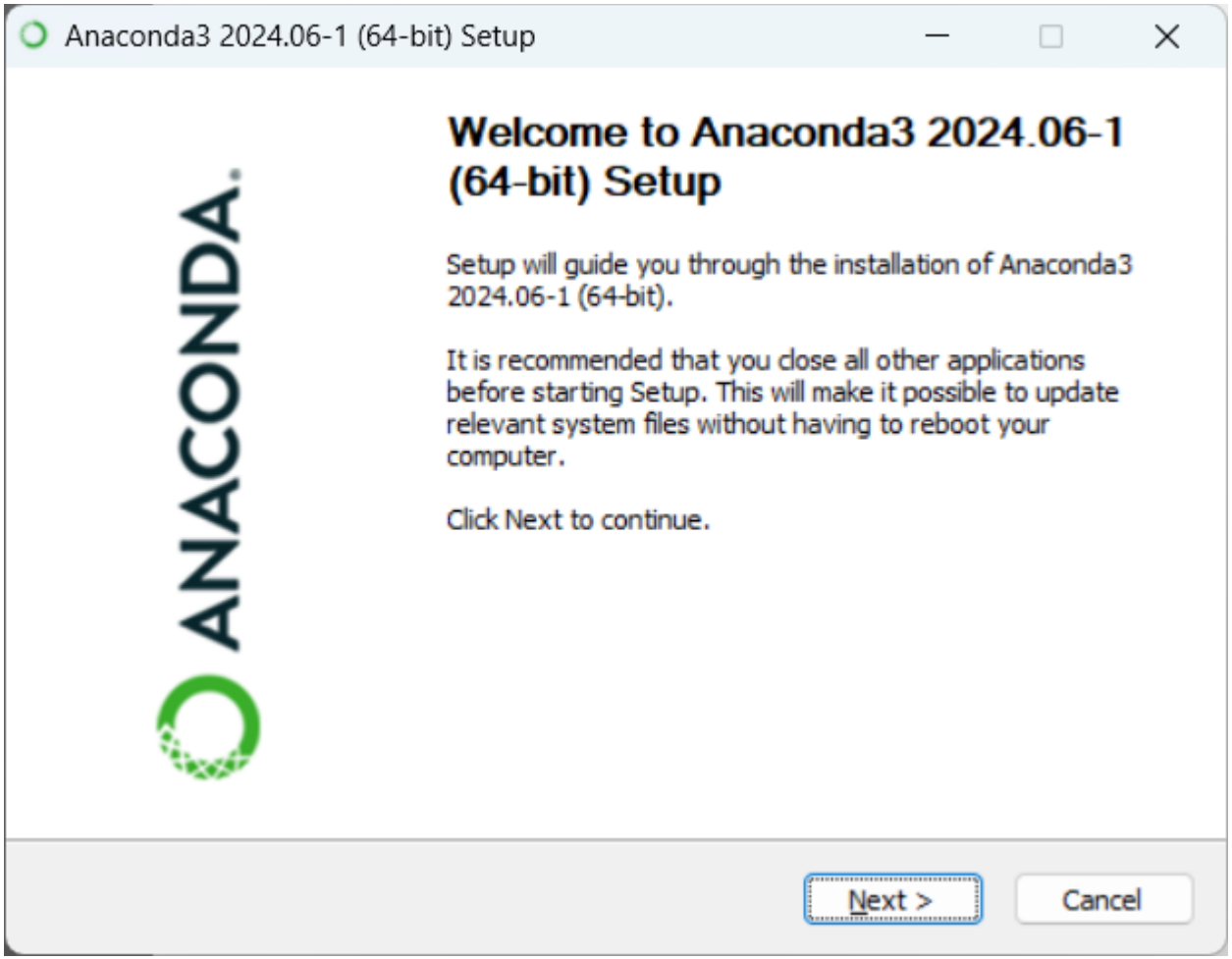
Python 3.12

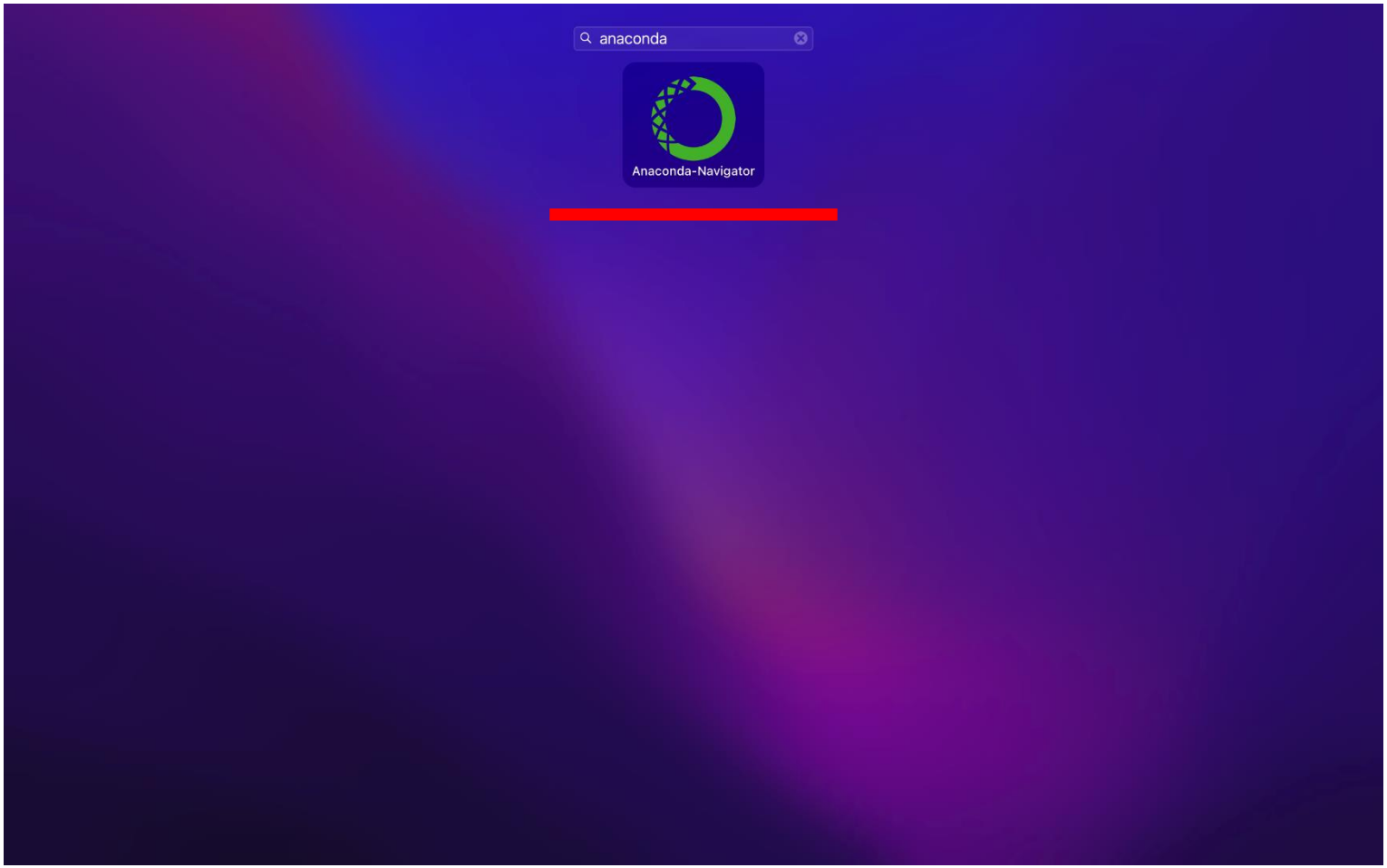
- 64-Bit (Apple silicon) Graphical Installer (704.7M)
- 64-Bit (Apple silicon) Command Line Installer (707.3M)
- 64-Bit (Intel chip) Graphical Installer (734.7M)
- 64-Bit (Intel chip) Command Line Installer (731.2M)

Linux

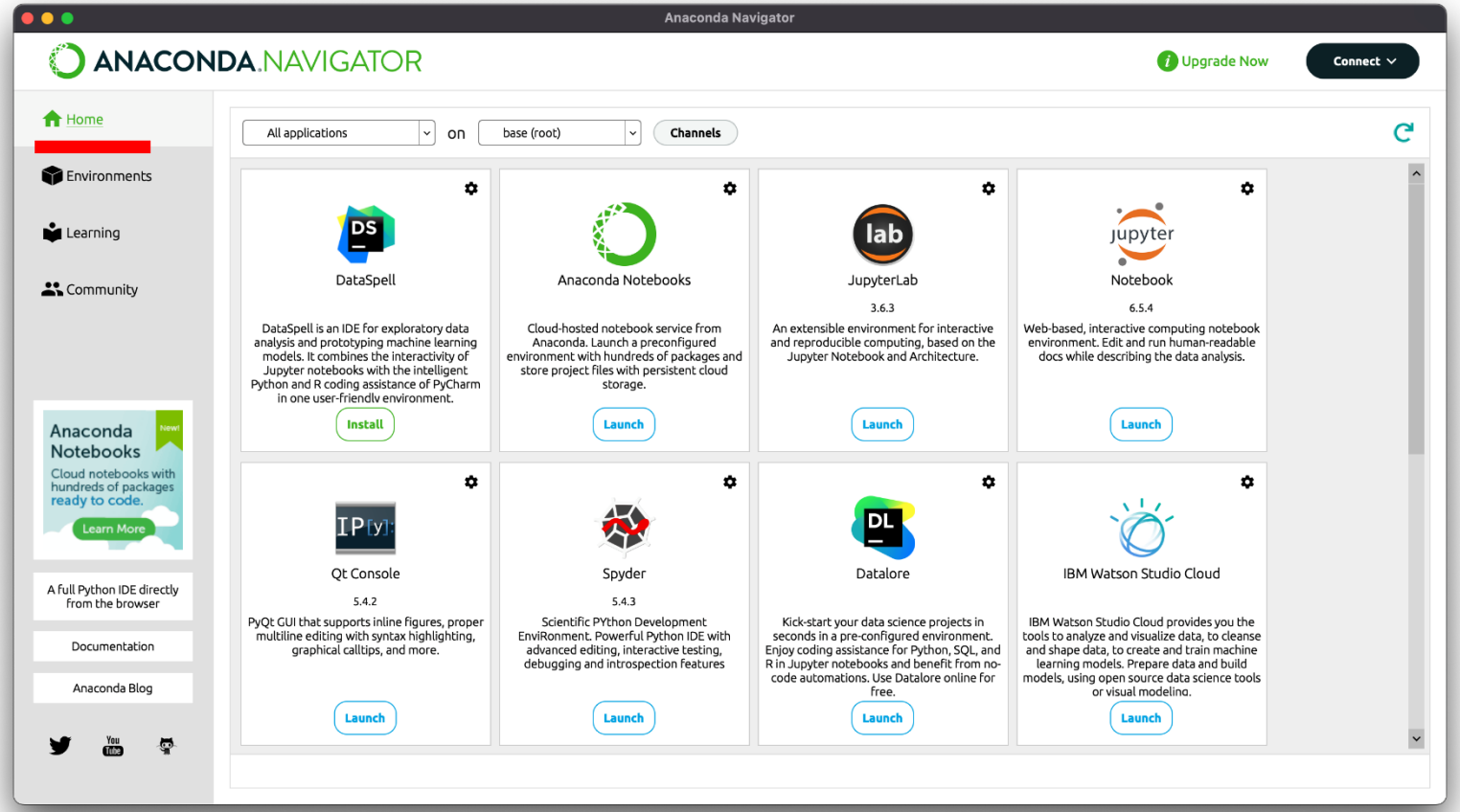
Python 3.12

- 64-Bit (x86) Installer (1007.9M)
- 64-Bit (AWS Graviton2 / ARM64) Installer (800.6M)
- 64-bit (Linux on IBM Z & LinuxONE) Installer (425.8M)


Anaconda3 2024.06-1 (64-bit) Setup
ANACONDA.
Welcome to Anaconda3 2024.06-1 (64-bit) Setup
Setup will guide you through the installation of Anaconda3 2024.06-1 (64-bit).
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.
Click Next to continue.
Next >
Cancel


anaconda
Anaconda-Navigator

Install software on the environment



Anaconda Navigator

File Help

Manage your environments when you have multiple buniness projects.

ANACONDA.NAVIGATOR

Upgrade Now

Connect

Home

Environments

Learning

Community

Anaconda Notebooks

New!

Cloud notebooks with hundreds of packages ready to code.

Learn More

A full Python IDE directly from the

Documentation

Anaconda Blog

Search Environments

base (root)

MoCap

py36

pytorch

Create Clone Import Backup Remove

Installed

Channels

Update index...

Search Packages

Name	T	Description	Version
_ipyw_jlab_nb_ex...		A configuration metapackage for enabling anaconda-bundled jupyter extensions	0.1.0
aiohttp		Async http client/server framework (asyncio)	3.8.1
aiosignal		Aiosignal: a list of registered asynchronous callbacks	1.2.0
alabaster		Configurable, python 2+3 compatible sphinx theme.	0.7.12
anaconda		Simplifies package management and deployment of anaconda	2022.05
anaconda-client		Anaconda.org command line client library	1.9.0
anaconda-project		Tool for encapsulating, running, and reproducing data science projects	0.10.2
anyio		High level compatibility layer for multiple asynchronous event loop implementations on python	3.5.0
appdirs		A small python module for determining appropriate platform-specific dirs.	1.4.4
argon2-cffi		The secure argon2 password hashing algorithm.	21.3.0
argon2-cffi-bindings		Low-level python cffi bindings for argon2	21.2.0
arrow		Better dates & times for python	1.2.2
astroid		A abstract syntax tree for python with inference support	2.6.6

430 packages available

Create a new conda environment

CONDA

conda | conda-build | conda.org

Search Ctrl + K

- conda create
- conda deactivate
- conda doctor
- conda env
  - conda env config
  - conda env create
  - conda env remove
  - conda env update
- conda export
- conda info
- conda init
- conda install
- conda list
- conda notices

Commands > conda env >

conda env create

Create an environment based on an environment definition file.

conda create --name mediapipe python=3.9

name in the CLI command using the -n/--name argument. The name specified in the CLI will override the name specified in the environment.yml file.

Unless you are in the directory containing the environment definition file, use -f to specify the file path of the environment definition file you want to use.

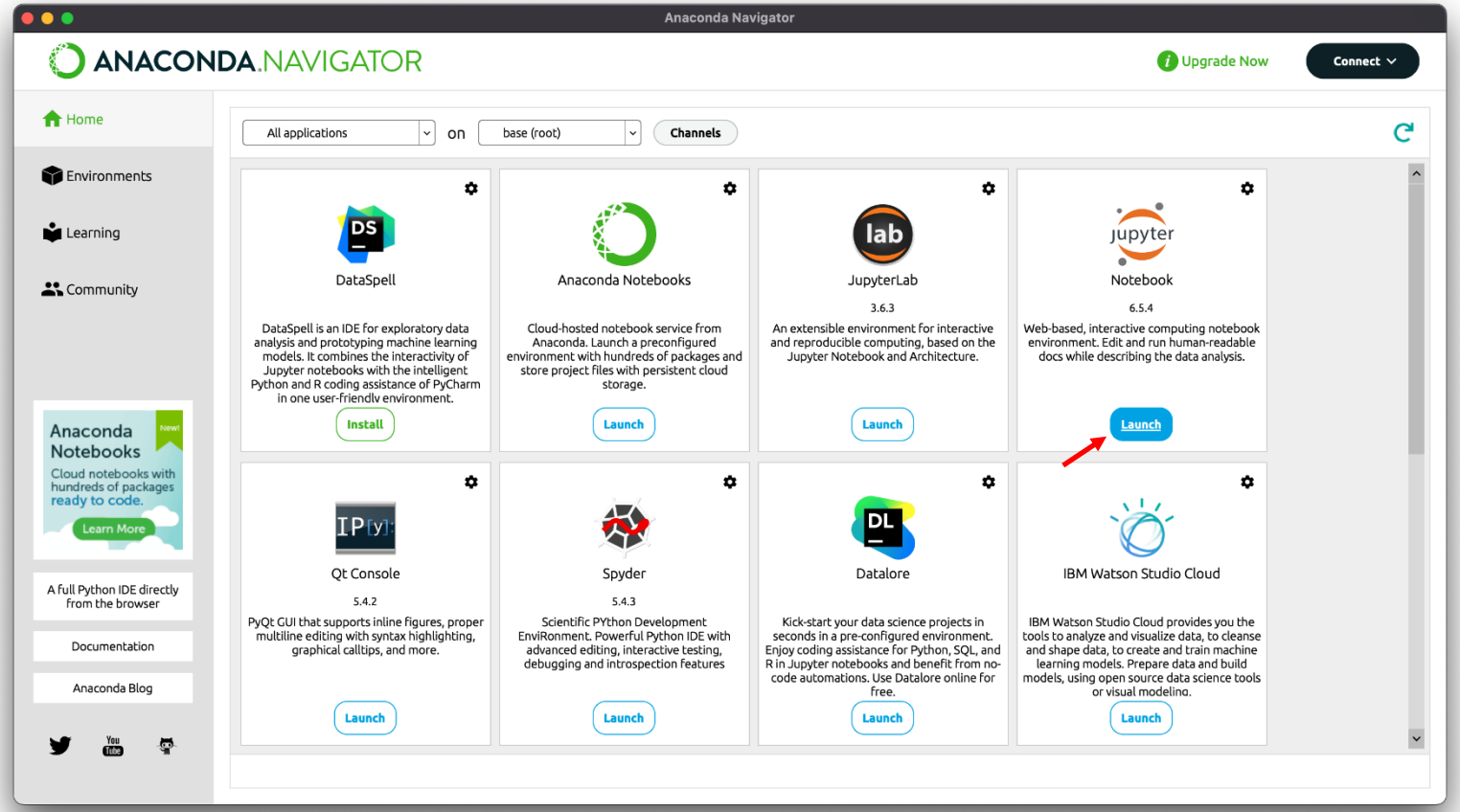
```
usage: conda env create [-h] [-f FILE] [-n ENVIRONMENT | -p PATH] [-C] [-k]
                        [--offline]
                        [--environment-specifier {binstar,environment.yml,exp
                        [--no-default-packages] [--json] [--console CONSOLE]
                        [-v] [-q] [-d] [-y] [--solver {classic,libmamba}]
                        [--subdir SUBDIR]
```

On this page

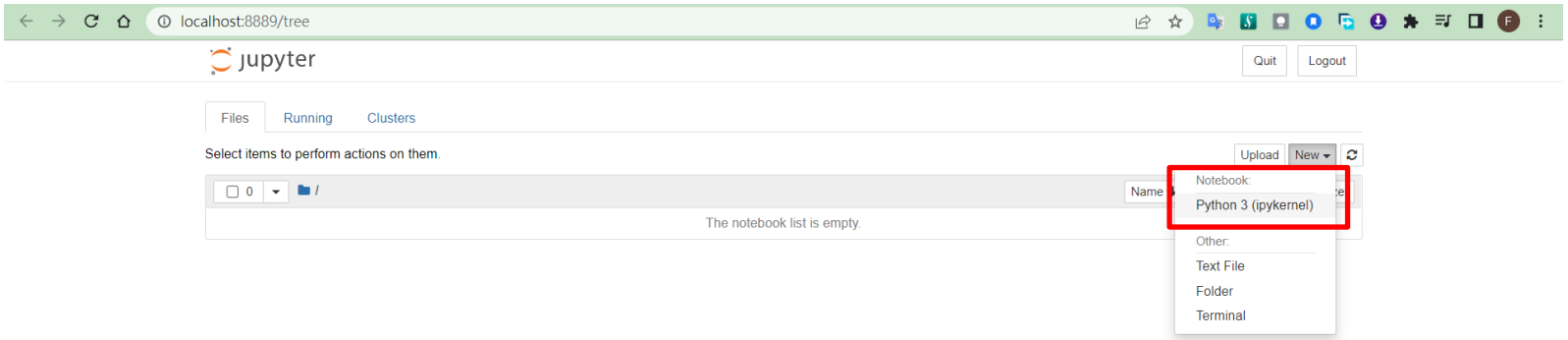
- Positional Arguments
- Named Arguments
- Target Environment Specification

Edit on GitHub

Show Source


Anaconda Navigator
ANACONDA.NAVIGATOR
Upgrade Now
Connect
Home
Environments
Learning
Community
All applications
on
base (root)
Channels
DataSpell
DataSpell is an IDE for exploratory data analysis and prototyping machine learning models. It combines the interactivity of Jupyter notebooks with the intelligent Python and R coding assistance of PyCharm in one user-friendly environment.
Install
Anaconda Notebooks
Cloud-hosted notebook service from Anaconda. Launch a preconfigured environment with hundreds of packages and store project files with persistent cloud storage.
Launch
JupyterLab
3.6.3
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Launch
Notebook
6.5.4
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch
Qt Console
5.4.2
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Launch
Spyder
5.4.3
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Launch
Datalore
Kick-start your data science projects in seconds in a pre-configured environment. Enjoy coding assistance for Python, SQL, and R in Jupyter notebooks and benefit from no-code automations. Use Datalore online for free.
Launch
IBM Watson Studio Cloud
IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.
Launch
Anaconda Notebooks
New!
Cloud notebooks with hundreds of packages ready to code.
Learn More
A full Python IDE directly from the browser
Documentation
Anaconda Blog

Juputer Notebook



localhost:8889/tree#

Home Page - Select or create a | Untitled - Jupyter Notebook

localhost:8889/notebooks/Untitled.ipynb?kernel_name=python3

jupyter Untitled Last Checkpoint: 2 分鐘前 (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

In [ ]
```
print("Welcome to Motion Capture")
```

How to Install

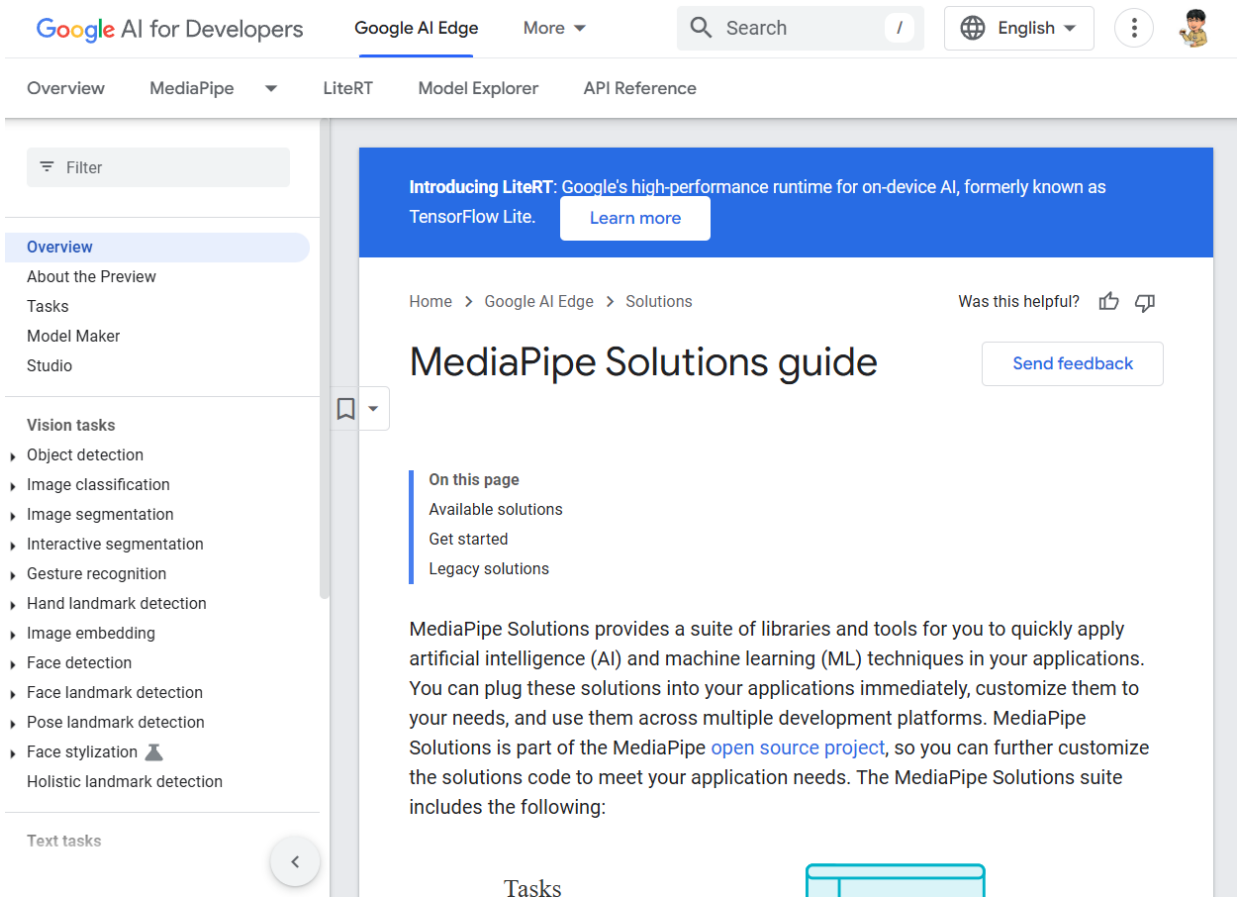
Mediapipe

in python?

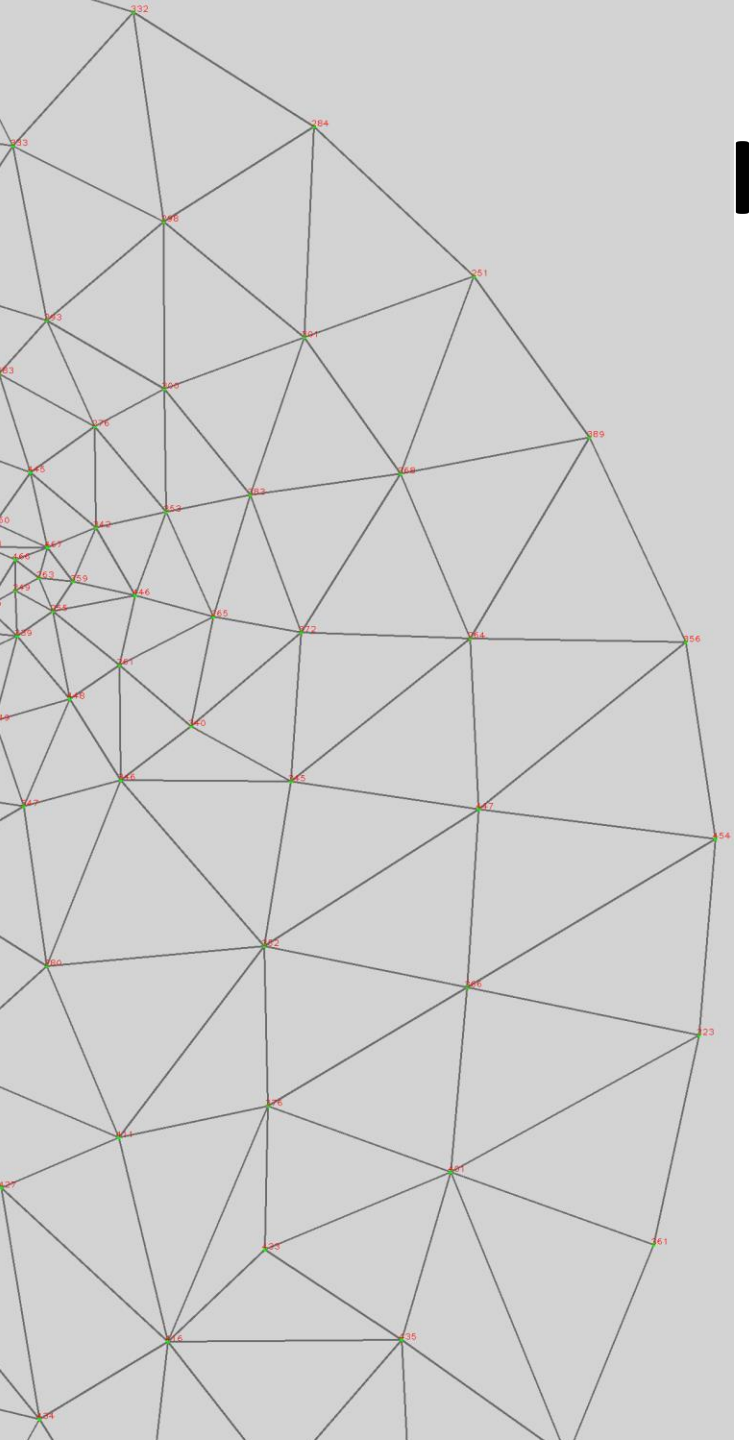
MediaPipe

ANACONDA

Mediapipe

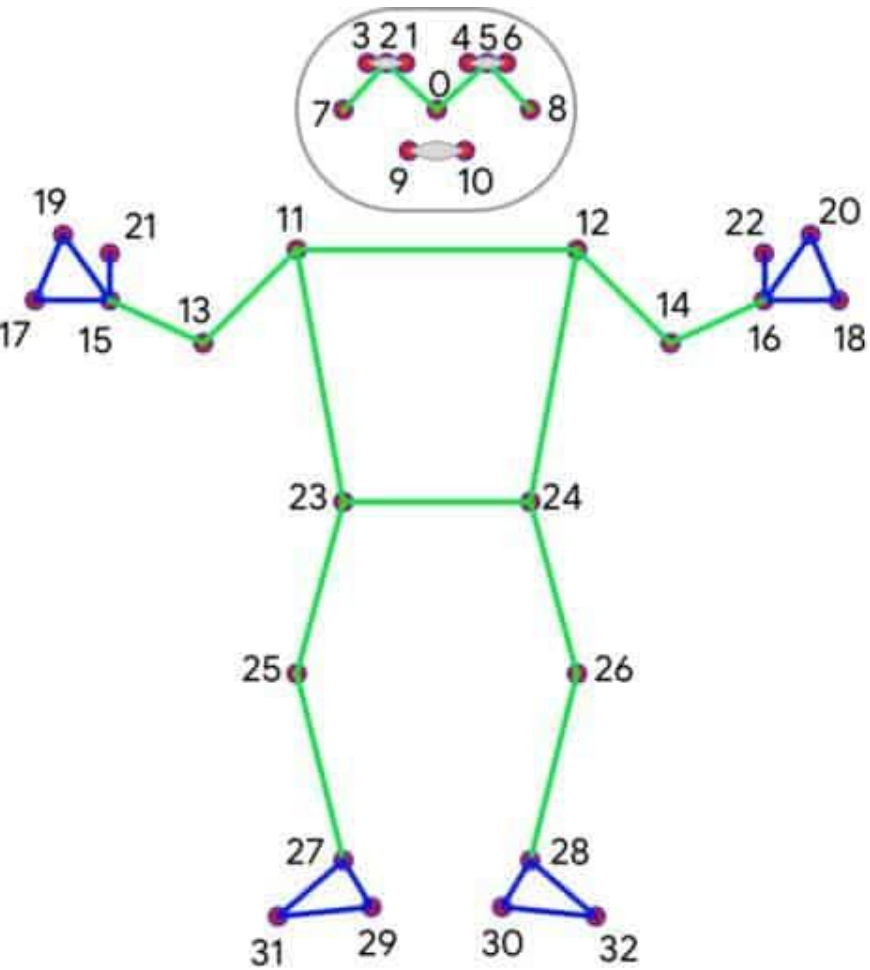


https://ai.google.dev/edge/mediapipe/solutions/guide?hl=zh-tw



l with 468 landmarks

Body Posture



0. nose
1. right eye inner
2. right eye
3. right eye outer
4. left eye inner
5. left eye
6. left eye outer
7. right ear
8. left ear
9. mouth right
10. mouth left
11. right shoulder
12. left shoulder
13. right elbow
14. left elbow
15. right wrist
16. left wrist
17. right pinky knuckle #1
18. left pinky knuckle #1
19. right index knuclke #1
20. left index knuckle #1
21. right thumb knuckle #2
22. left thumb knuckle #2
23. right hip
24. left hip
25. right knee
26. left knee
27. right ankle
28. left ankle
29. right heel
30. left heel
31. right foot index
32. left foot index

What’s Covered

- Setting up Media Pipe
- Tracking face, hands of a human
- Visualizing detections to the screen (class practise)

jupyter Media Pipe Holistic Tutorial Last Checkpoint: 1 天前 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Run Markdown

0. Install and Import Dependencies

In [ ]:
```
!pip install mediapipe opencv-python
```

Install the libraries

In [1]:
```
import mediapipe as mp
import cv2
```

Import the libraries into your program

In [2]:
```
mp_drawing = mp.solutions.drawing_utils
mp_holistic = mp.solutions.holistic
```

1. Get Realtime Webcam Feed

In [ ]:
```
cap = cv2.VideoCapture(0)
while cap.isOpened():
    ret, frame = cap.read()
    cv2.imshow('Raw Webcam Feed', frame)

    if cv2.waitKey(10) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

we're getting our video capture device

frame is the image from your webcam

Breaking out of that loop, hitting 'q' on our keyboard

In [17]:
```
cap.release()
cv2.destroyAllWindows()
```

jupyter Media Pipe Holistic Tutorial Last Checkpoint: 幾秒前 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Markdown

2. Make tracking from webcam ¶

1. track Facial Landmarks
2. track Hand Poses
3. track Body Poses

open up our holistic model (pre-trained model in mediapipe)

In [11]:
```
cap = cv2.VideoCapture(0)
# Initiate holistic model
with mp_holistic.Holistic(min_detection_confidence=0.5, min_tracking_confidence=0.5) as holistic:

    while cap.isOpened():
        ret, frame = cap.read()

        # Recolor Feed
        image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        # Make Detections
        results = holistic.process(image)
        # print(results.face_landmarks)

        # face_landmarks, pose_landmarks, left_hand_landmarks, right_hand_landmarks

        # Recolor image back to BGR for rendering
        image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

        # Draw face landmarks
        #mp_drawing.draw_landmarks(image, results.face_landmarks, mp_holistic.FACEMESH_TESSELATION)

        # Right hand
        #mp_drawing.draw_landmarks(image, results.right_hand_landmarks, mp_holistic.HAND_CONNECTIONS)

        # Left Hand
        #mp_drawing.draw_landmarks(image, results.left_hand_landmarks, mp_holistic.HAND_CONNECTIONS)

        # Pose Detections
        mp_drawing.draw_landmarks(image, results.pose_landmarks, mp_holistic.POSE_CONNECTIONS)

        cv2.imshow('Raw Webcam Feed', image)

        if cv2.waitKey(10) & 0xFF == ord('q'):
            break

cap.release()
cv2.destroyAllWindows()
```

confidence: higher tracking accuracy with higher value

pass it to our holistic model

process that image to be able to display

Working with holistic model

jupyter Media Pipe Holistic Tutorial Last Checkpoint: 1 分鐘前 (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

3. Apply Styling

In [13]:
```
mp_drawing.DrawingSpec(color=(0,0,255), thickness=2, circle_radius=2)
```

Out[13]:
```
DrawingSpec(color=(0, 0, 255), thickness=2, circle_radius=2)
```

In [14]:
```
mp_drawing.draw_landmarks??
```

In [7]:
```
cap = cv2.VideoCapture(0)
# Initiate holistic model
with mp_holistic.Holistic(min_detection_confidence=0.5, min_tracking_confidence=0.5) as holistic:

    while cap.isOpened():
        ret, frame = cap.read()

        # Recolor Feed
        image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        # Make Detections
        results = holistic.process(image)
        # print(results.face_landmarks)

        # face_landmarks, pose_landmarks, left_hand_landmarks, right_hand_landmarks

        # Recolor image back to BGR for rendering
        image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

        # 1. Draw face landmarks
        mp_drawing.draw_landmarks(image, results.face_landmarks, mp_holistic.FACEMESH_TESSELATION,
                                  mp_drawing.DrawingSpec(color=(80,110,10), thickness=1, circle_radius=1),
                                  mp_drawing.DrawingSpec(color=(80,256,121), thickness=1, circle_radius=1)
                                  )

        # 2. Right hand
        mp_drawing.draw_landmarks(image, results.right_hand_landmarks, mp_holistic.HAND_CONNECTIONS,
                                  mp_drawing.DrawingSpec(color=(80,110,10), thickness=1, circle_radius=1),
                                  mp_drawing.DrawingSpec(color=(80,256,121), thickness=1, circle_radius=1)
                                  )

        # 3. Left Hand
        mp_drawing.draw_landmarks(image, results.left_hand_landmarks, mp_holistic.HAND_CONNECTIONS,
                                  mp_drawing.DrawingSpec(color=(80,110,10), thickness=1, circle_radius=1),
                                  mp_drawing.DrawingSpec(color=(80,256,121), thickness=1, circle_radius=1)
                                  )
```

Motion Capture and Future Interaction Technology Research

MoCap with hand tracking

Speaker: Fu-Song Hsu

What’s Covered

- Setup hand tracking module
- Two-Hand Tracking
- Hand Pose Recognition (class work)
- Interactive Game

MediaPipe

Hand Tracking

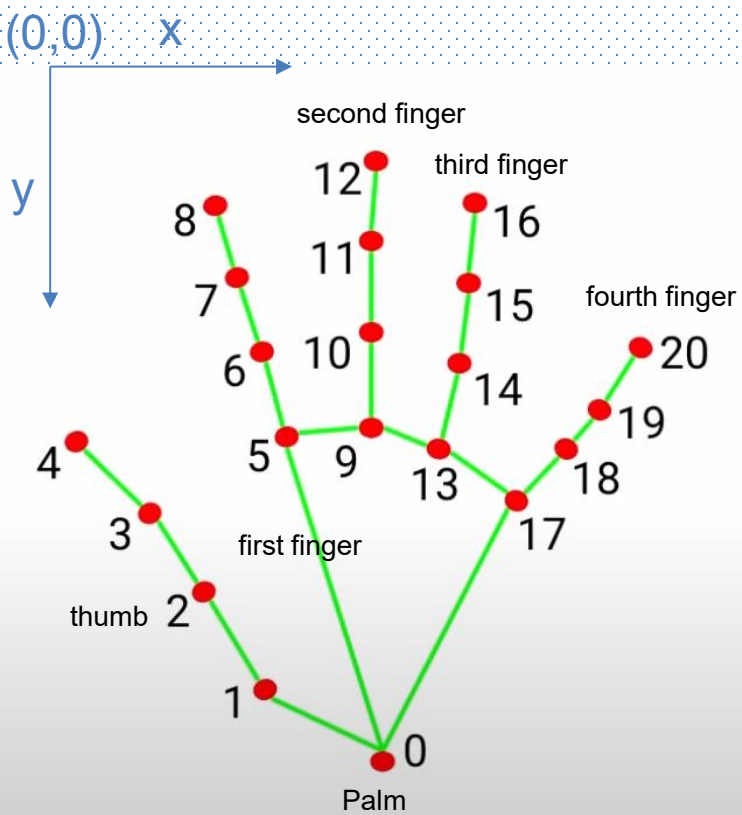
Palm Detection

complete image

Hand Landmarks

cropped image

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

Source: Media Pipe Website

Two-Hand Tracking

File Edit View Insert Cell Kernel Help Not Trusted Python 3 (ipykernel)

Run Code

In [2]:
```
import mediapipe as mp
import cv2
import time
import math
import numpy as np
```

Import libraries

In [3]:
```
mpDraw = mp.solutions.drawing_utils
mpHands = mp.solutions.hands
```

Two-Hand Tracking

In [4]:
```
hands = mpHands.Hands(
    static_image_mode=False,
    #model_complexity=0,
    max_num_hands=2,
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5
    )
```

create a module for tracking hands

In [5]:
```
cap = cv2.VideoCapture(0)
while cap.isOpened():
    stime=time.time()
    ret, frame = cap.read()
    h, w, c = frame.shape
    frame=cv2.flip(frame,1)
    imgRGB = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = hands.process(imgRGB)

    if results.multi_hand_landmarks:
        for i in range(len(results.multi_handedness)):
            thisHandType=results.multi_handedness[i].classification[0].label
            thisHand=results.multi_hand_landmarks[i]
            mpDraw.draw_landmarks(frame, thisHand, mpHands.HAND_CONNECTIONS)

            for id, lm in enumerate(thisHand.landmark):
                hx, hy = int(lm.x * w), int(lm.y * h)
                cv2.circle(frame, (hx, hy), 5, (255, 0, 0), cv2.FILLED)
                cv2.putText(frame,str(id),(hx,hy), cv2.FONT_HERSHEY_PLAIN, 1, (255, 0, 255), 1)
                if id==0:
                    cv2.putText(frame,thisHandType,(hx,hy-30), cv2.FONT_HERSHEY_PLAIN, 2, (0, 255, 0), 2)
    etime=time.time()
    fps=round(1/(etime-stime),2)
    cv2.putText(frame,"FPS:" + str(fps),(10,50), cv2.FONT_HERSHEY_PLAIN, 3, (0, 0, 255), 3)
```

```
    #model_complexity=0,
    max_num_hands=2,
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5
    )
```

send our webcam image to hand module

```
In [5]: cap = cv2.VideoCapture(0)
        while cap.isOpened():
            stime=time.time()
            ret, frame = cap.read()
            h, w, c = frame.shape
            frame=cv2.flip(frame,1)
            imgRGB = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
            results = hands.process(imgRGB)

            if results.multi_hand_landmarks:
                for i in range(len(results.multi_handedness)):
                    thisHandType=results.multi_handedness[i].classification[0].label
                    thisHand=results.multi_hand_landmarks[i]
                    mpDraw.draw_landmarks(frame, thisHand, mpHands.HAND_CONNECTIONS)

                    for id, lm in enumerate(thisHand.landmark):
                        hx, hy = int(lm.x * w), int(lm.y * h)
                        cv2.circle(frame, (hx, hy), 5, (255, 0, 0), cv2.FILLED)
                        cv2.putText(frame,str(id),(hx,hy), cv2.FONT_HERSHEY_PLAIN, 1, (255, 0, 255), 1)
                        if id==0:
                            cv2.putText(frame,thisHandType,(hx,hy-30), cv2.FONT_HERSHEY_PLAIN, 2, (0, 255, 0), 2)
            etime=time.time()
            fps=round(1/(etime-stime),2)
            cv2.putText(frame,"FPS:" + str(fps),(10,50), cv2.FONT_HERSHEY_PLAIN, 3, (0, 0, 255), 3)
            cv2.imshow('Webcam',frame)
            key=cv2.waitKey(1)
            if key==ord('a'):
                cv2.imwrite('webcam.jpg',frame)
            if key==ord('q'):
                break
        cap.release()
        cv2.destroyAllWindows()
```

apply custom colors

get hand label

get hand landmark

draw hand landmarks

get FPS

Hand Pose Recognition

Hand Pose Recognition

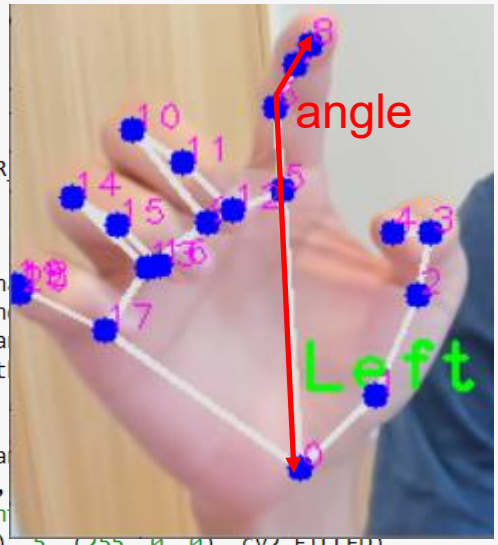
In [29]:
```
hands = mpHands.Hands(
    static_image_mode=False,
    model_complexity=0,
    max_num_hands=1,
    min_detection_confidence=0.7,
    min_tracking_confidence=0.5
    )
```

In [30]

create a function for getting degree

```
AngleTH=130
def findAngleF(a,b,c):
    ang = math.degrees(math.atan2(c[2]-b[2], c[1]-b[1]) - math.atan2(a[2]-b[2], a[1]-b[1]))
    print(ang)
    if ang<0 :
      ang=ang+360
    if ang >= 360- ang:
        ang=360-ang
    return round(ang,2)
```

In [31]:
```
cap = cv2.VideoCapture(0)
while cap.isOpened():
    stime=time.time()
    ret, frame = cap.read()
    h, w, c = frame.shape
    frame=cv2.flip(frame,1)
    imgRGB = cv2.cvtColor(frame, cv2.COLOR
    results = hands.process(imgRGB)

    if results.multi_hand_landmarks:
        for i in range(len(results.multi_h
            thisHandType=results.multi_han
            thisHand=results.multi_hand_la
            mpDraw.draw_landmarks(frame, t

            thisHandLMList = []
            for id, lm in enumerate(thisHa
                thisHandLMList.append([id,
                hx, hy = int(lm.x * w), in
                cv2.circle(frame, (hx, hy), 5, (255, 0, 0), cv2.FILLED)
                cv2.putText(frame,str(id),(hx,hy), cv2.FONT_HERSHEY_PLAIN, 1, (255, 0, 255), 1)
```



For an instance,
a=0
b=6
c=8

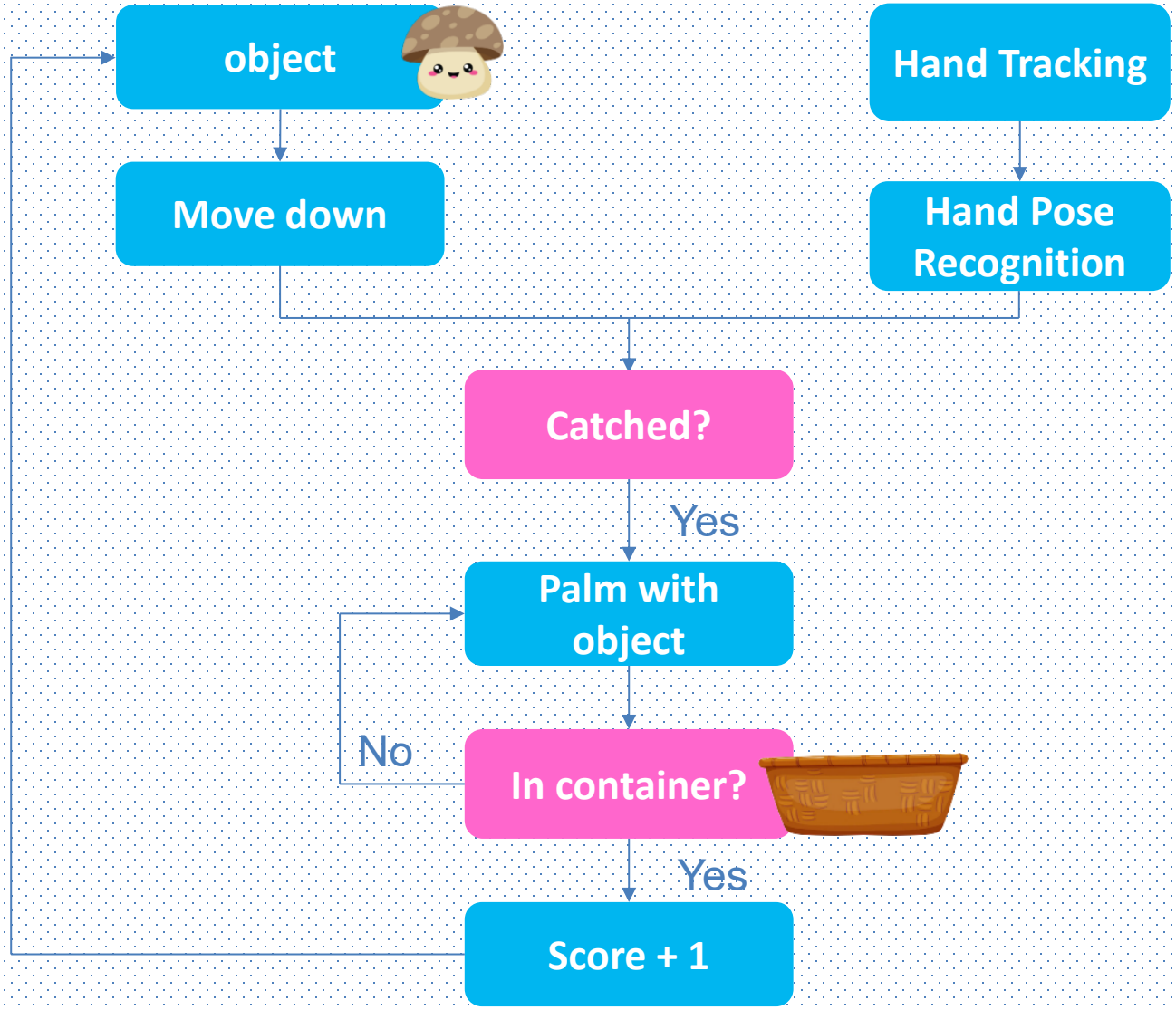
In [31]:
```
cap = cv2.VideoCapture(0)
while cap.isOpened():
    stime=time.time()
    ret, frame = cap.read()
    h, w, c = frame.shape
    frame=cv2.flip(frame,1)
    imgRGB = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = hands.process(imgRGB)

    if results.multi_hand_landmarks:
        for i in range(len(results.multi_handedness)):
            thisHandType=results.multi_handedness[i].classification[0].label
            thisHand=results.multi_hand_landmarks[i]
            mpDraw.draw_landmarks(frame, thisHand, mpHands.HAND_CONNECTIONS)

            thisHandLMList = []
            for id, lm in enumerate(thisHand.landmark):
                thisHandLMList.append([id, lm.x, lm.y,lm.z])
                hx, hy = int(lm.x * w), int(lm.y * h)
                cv2.circle(frame, (hx, hy), 5, (255, 0, 0), cv2.FILLED)
                cv2.putText(frame,str(id),(hx,hy), cv2.FONT_HERSHEY_PLAIN, 1, (255, 0, 255), 1)
                if id==0:
                    cv2.putText(frame,thisHandType,(hx,hy-30), cv2.FONT_HERSHEY_PLAIN, 2, (0, 255, 0), 2)
            finger=[0,0,0,0,0]
            if (findAngleF(thisHandLMList[0],thisHandLMList[3],thisHandLMList[4])>AngleTH):
                finger[0]=1
            if (findAngleF(thisHandLMList[0],thisHandLMList[6],thisHandLMList[8])>AngleTH):
                finger[1]=1
            if (findAngleF(thisHandLMList[0],thisHandLMList[10],thisHandLMList[12])>AngleTH):
                finger[2]=1
            if (findAngleF(thisHandLMList[0],thisHandLMList[14],thisHandLMList[16])>AngleTH):
                finger[3]=1
            if (findAngleF(thisHandLMList[0],thisHandLMList[18],thisHandLMList[20])>AngleTH):
                finger[4]=1
            print(finger)


            text=""
            if (finger==[0,0,0,0,0]):
                text="Zero"
            if (finger==[1,1,1,1,1]):
                text="Hi"
```

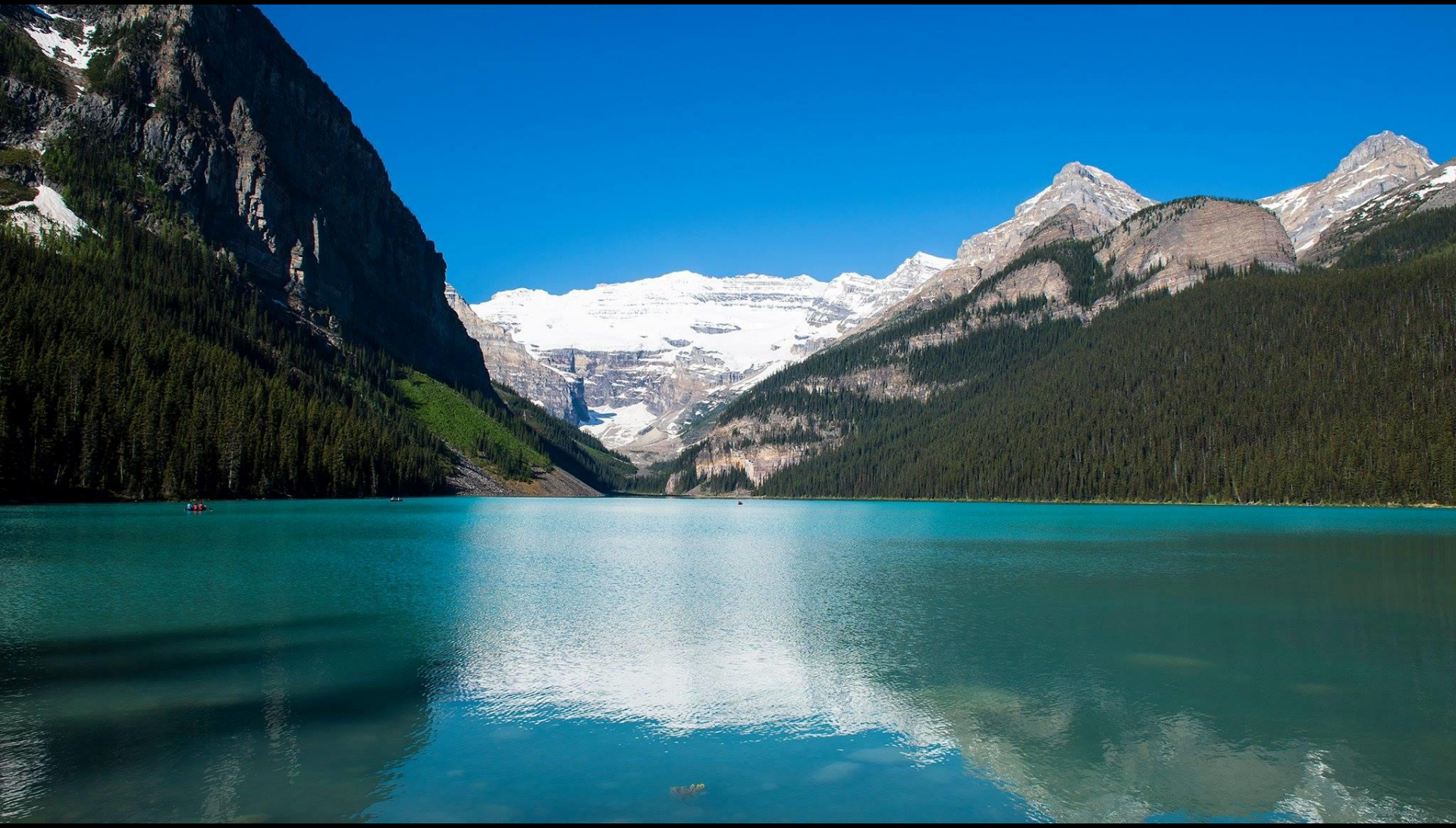
Hand Pose Recognition
By calculating angles

Class work

Five hand gestures: victory, hi, spider-man and so on

Interactive Game


object
Hand Tracking
Move down
Hand Pose Recognition
Catched?
Yes
Palm with object
No
In container?
Yes
Score + 1



Q&A