座位表 -請同學填入 自己名字(有看到的提醒隔壁的同學 填寫 感謝!!

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **白板** | | | | | |
| 楊凱元 | 陳瑀萱 | 翁紹捷⭐ | 吳承祐 | 吳晉宇 | 助教 |
| 戴中航 | 沈咨名 | 班代:翁尉珊 | 程韋鈞 | 趙冠宇 | 譚峻 |
| 連彥鈞 | 劉樾 | 許瑀芹 | 王俐斐 | 葉濠銘 | 張雲藝 |
| 張書榜 | 王祥鈞 | 張又心 | 許媁涵 | 方俊程 | 蔡定芝 |
| 😂😂😂 | | 蘇煜㭿 | 林苔英 | 簡緯翔 | 曾祥易 |
| 黃玟勝(+100) | 吳柏均 | 張竣豪 | 黨敏伶 |

# 

# 課綱

圖文選單的製作

official account的方式

程式碼的方式

下載新教材

016-018

Chatbot重溫

做一個蒐集image的機器人

AI全民化

結合模型建模做一個圖像辨識機器人

-----

GCP使用快市場半步，市場追我們

AI與雲端

雲原生概念

帳號建置

帳單預警(暫略)

核心服務介紹

CloudStorage - 無限資料儲存

IAM

Cloud SDK integrate CloudStorage - 用程式碼操作儲存空間

Chatbot integrate GCP cloudstorage - 用機器人結合儲存空間，照片無限上傳

# Chatbot 圖文選單

### 官方平台版

<https://account.line.biz/login?redirectUri=https%3A%2F%2Fmanager.line.biz%2F>

### Line Messaging api 版

<https://github.com/BingHongLi/line_chat_bot_tutorial/blob/master/material/step1/Step1_UserRichMenu.ipynb>

圖文選單與用戶跟line的關係

準備菜單的圖面設定檔

讀取安全設定檔上的參數

將菜單設定檔透過line\_bot\_api傳給Line   
 第一步create\_rich\_menu api to line

第二步set\_rich\_menu\_image api to 圖文選單

第三步link\_rich\_menu\_to\_user 用戶 and 圖文選單

對Line上傳菜單照片 檢視現有的菜單

將菜單與用戶做綁定

將菜單與用戶解除綁定

刪除菜單

rich\_menu包含 (CRUD)

Create-製造

Read-關聯

Update-檢查用戶關聯

Delete-解除用戶關聯

我要把圖文選單刪掉

<https://github.com/line/line-bot-sdk-python#delete_rich_menuself-rich_menu_id-timeoutnone>

手把手寫一套

https://codeshare.io/21dbjX

ai班資料夾

<https://reurl.cc/9XNk2Y>

<https://drive.google.com/file/d/1T3JlskbBn-zY8ZfA8CCmgNVBKNoAJYib/view?usp=sharing>

# 圖像模型與chatbot結合

教材016

<https://github.com/BingHongLi/ai-chatbot-tutorial/blob/main/016_teachable-machine_chatbot.ipynb>

下載，貼回自己的drive內，更換channel secret與channel access token

將上午的模型壓縮檔置入，運行，更換webhook

程式碼解析

建置流程:

1. 蒐集照片(資料集)

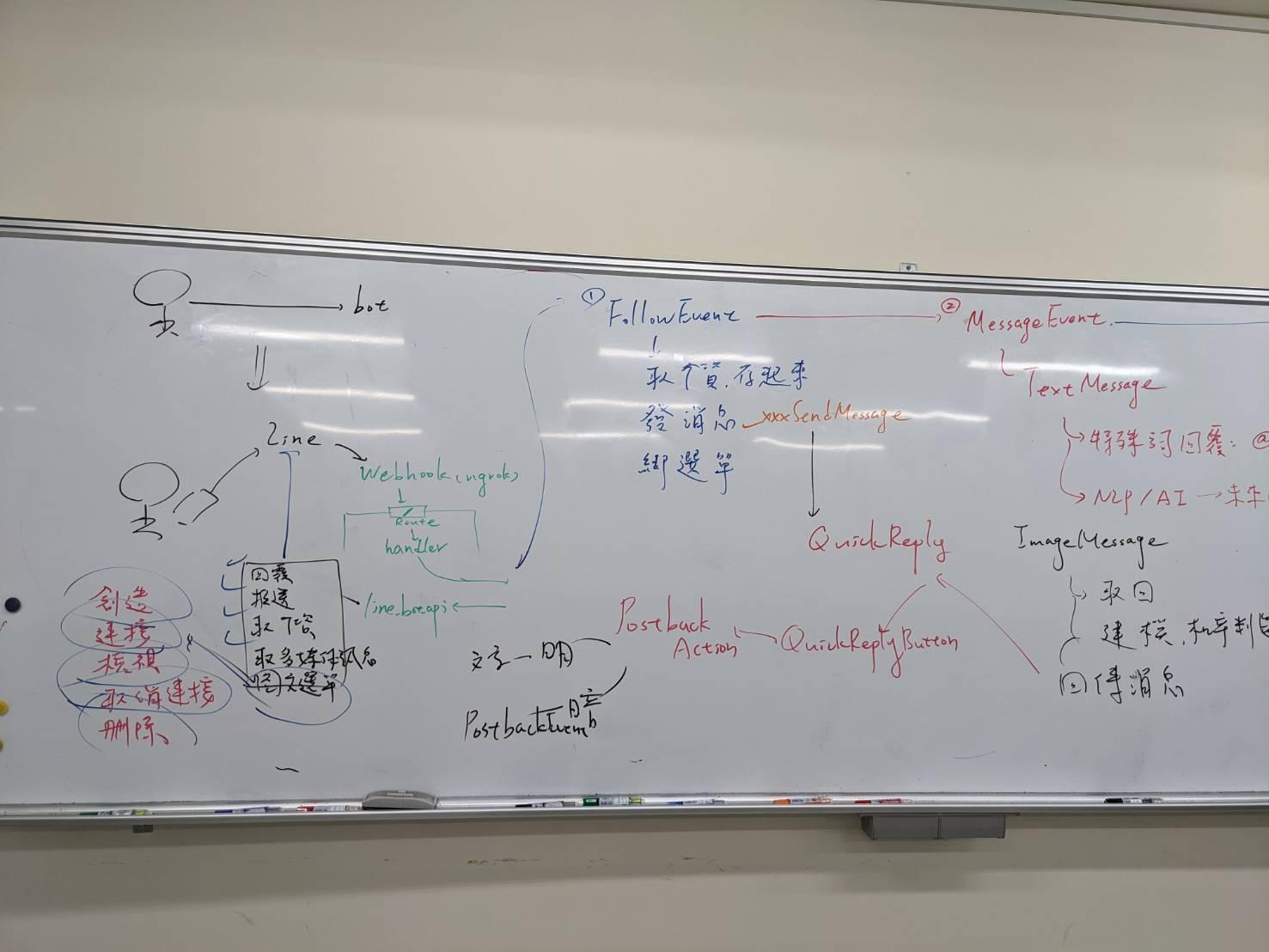
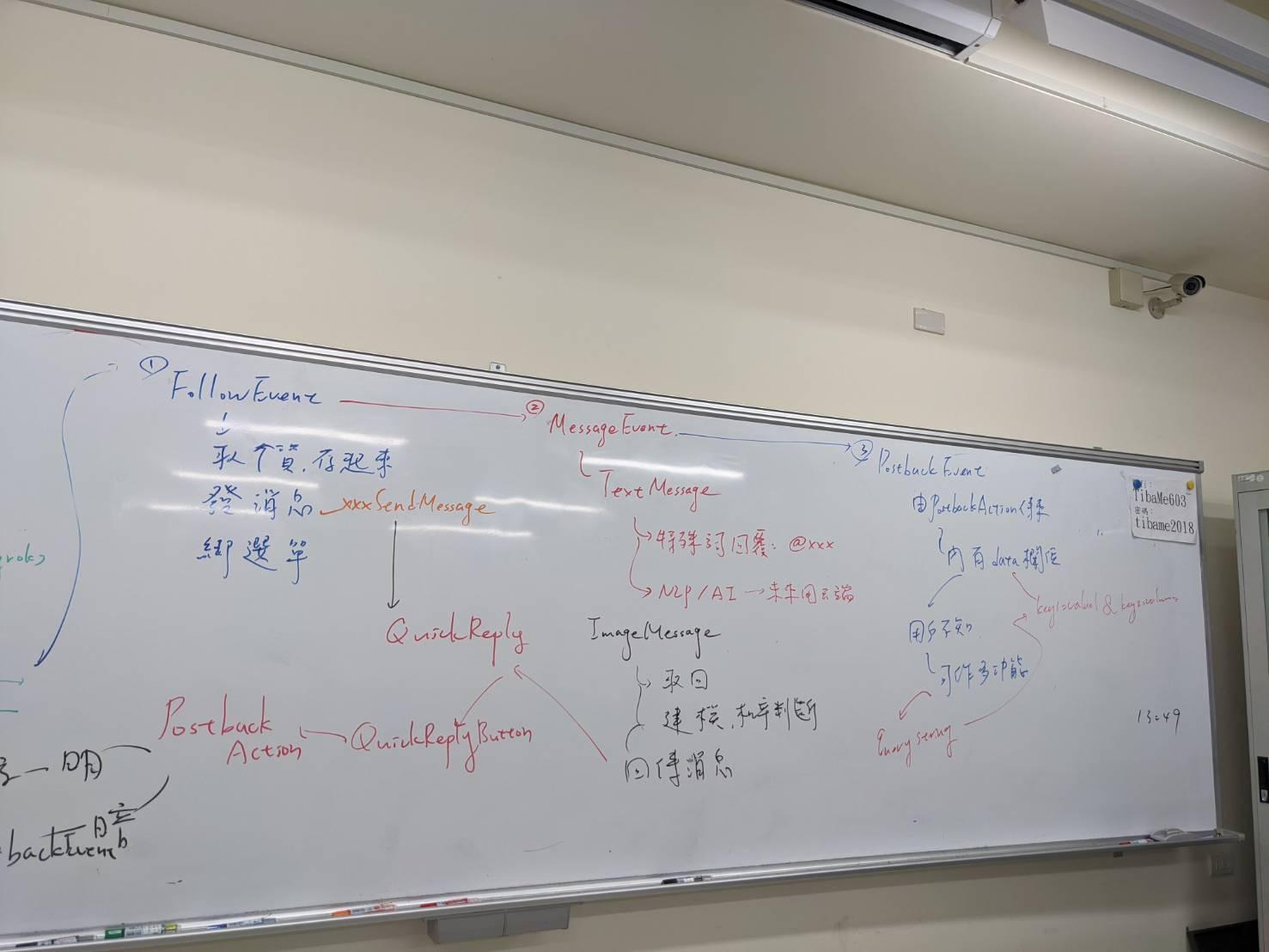
2. 建置模型(teachable machine) <https://teachablemachine.withgoogle.com/train>

3. 模型+LINE機器人

應用場景:

- 破冰遊戲(?

- 城市探險



AI跟雲服務有很大的相關性

雲原生

# 雲服務體驗流程

google 搜尋gcp

申辦gcp帳號

帶大家看一下帳單預警

gcp的三大核心服務

cloud storgae:碰到巨量資料時，需要無限空間(一般硬碟最多2T)

cloud compute engine

iam

# Cloud Storage

## 核心名詞:

所有檔案都是object，裝載object的是bucket

## 實作

### 目標

讓大家對CloudStorage有具體的想像，再進一步解析

### 具體流程

建置bucket

資料所在地

region

dual-region

multi-region

儲存類別(Storage-level)

上傳物件

下載物件

內部人員

外部人員 - acl - 精細或統一

<https://cloud.google.com/storage/docs/how-to>

cloud shell

#執行flask app server

#用pip 安裝flask

#google搜尋flask quickstart

pip3 install flask

**from** flask **import** Flask

app = Flask**(**\_\_name\_\_**)**

@app.route**(**'/'**)**

**def** hello\_world**():**

**return** 'Hello, World!'