

◆ 企業與教育 AI 實務課程

# Gemini 自動化 workflow 與 Apps Script 程式碼生成

不用程式背景，讓 AI 幫你寫程式——打造個人專屬的全自動辦公系統

📊 試算表

▲ 雲端硬碟

📄 表單

📅 日曆

✉ Gmail

◆ Gemini 串起全部

---

講師 **賴家榮** 甫東科技 創辦人 | 輔英科大 · 樹德科大 · 屏東大學 講師



開場

# 關於今天的講師

從系統工程到 AI 導入教學——實務出身的自動化教練

賴

賴家榮

甫東科技 創辦人

SGS ISO 42001

AI 管理系統認證

勞動部人力提升計畫講師

企業內訓經驗 6 年以上

## 現任

- 甫東科技 創辦人——企業 AI 導入與自動化顧問
- 輔英科技大學・樹德科技大學 數位行銷講師
- 屏東大學 資訊工程系 講師

## 曾任

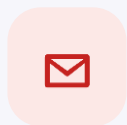
- 文魁 / 松崗 系統工程師
- 2009 高雄世界運動會 計時計分系統工程師
- 勞動部大型 / 中小型人力提升計畫 講師 ( 6 年以上 )



開場

# 你是否也有這些困擾？

每天都在做的「例行公事」，其實正在偷走你的時間



## 一封一封寄通知

成績單、繳費提醒、活動通知.....

30 封信要複製貼上 90 分鐘，還怕貼錯人



## 資料夾亂成一團

收來的作業、照片、報價單全堆在一起

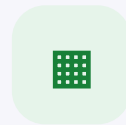
找一個檔案要翻 10 分鐘



## 報名資料人工抄日曆

表單收到預約，還要自己抄進行事曆

一忙起來就漏掉、撞時段



## 每週重做同一份報表

每週一固定彙整資料寄給主管

步驟一模一樣，卻週週從頭做一次

這些工作有一個共同點：**規則固定、重複發生**——而這正是最適合交給自動化的工作



開場

# 今天的學習地圖：由淺入深，四步到位

不用任何程式基礎——每一步都有實作，離開教室就能用



難度



入門：跟著做就會

進階：能自己設計工作流



開場

# 課前準備：只要一個 Google 帳號

全程使用網頁版服務，不需安裝任何軟體、不需付費

## 開課前請確認 ✓



### Google 帳號

一般免費 Gmail 帳號即可 ( @gmail.com )



### 能登入 gemini.google.com

開啟網頁、登入同一個 Google 帳號



### Chrome 或 Edge 瀏覽器

建議使用電腦操作，畫面與課程一致



### 穩定網路

所有工具都在雲端執行

## 今天會用到的服務 ( 皆為網頁版 )



### Google 試算表

存放資料的表格



### 雲端硬碟

檔案的家



### Google 表單

收集回覆的入口



### Google 日曆

行程管理



### Gmail

寄送郵件



### Gemini

幫你寫程式的 AI

以上全部免費，不需要信用卡

# 01

◆ PART 1

## 觀念建立與環境準備

先搞懂「自動化」是怎麼一回事，再把工具箱打開

01

什麼是自動化工作流？  
用生活例子秒懂

02

認識 Google Workspace  
生態系與 Studio

03

Apps Script 是什麼？  
能幫你做哪些事

04

實作：打開編輯器  
跑出第一支程式

# 什麼是「自動化 workflow」？

其實你早就每天在用了——只是還沒用在工作上



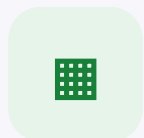
## 生活中早就有的例子



今天要做的事：把「鬧鐘的邏輯」搬進辦公室——表單一有人填 → 程式自動處理 → 日曆自動多一筆行程

# 認識 Google Workspace：你的雲端辦公室

五個免費服務，各司其職——自動化就是讓它們「彼此說話」



## Google 試算表

= 雲端的 Excel

存名單、成績、  
訂單等表格資料

資料倉庫

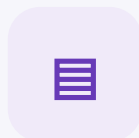


## 雲端硬碟

= 雲端的資料夾

存放所有檔案，  
可分享、可分類

檔案的家

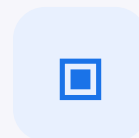


## Google 表單

= 線上問卷

收報名、收回覆，  
結果自動進試算表

收集入口

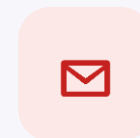


## Google 日曆

= 雲端行事曆

管理行程，  
可邀請、可提醒

時間管家



## Gmail

= 電子郵件

寄送通知信，  
每天可自動寄百封

訊息發送

平常它們是五座孤島，資料要靠你「人工搬運」。

學會自動化之後：**試算表會自己叫 Gmail 寄信、表單會自己往日曆寫行程。**

# Apps Script : Google 內建的「自動化引擎」

它就是讓五座孤島彼此說話的那條「傳令兵」

## 白話認識它

- **Google 官方內建的程式語言** 用一小段「指令稿」指揮試算表、Gmail、日曆做事
- **完全免費、免安裝** 打開瀏覽器就能寫，程式存在雲端
- **天生就認識 Google 全家** 一行指令就能寄信、建行程，不用複雜設定
- **電腦關機也能執行** 程式住在 Google 雲端，時間到自己上工

### 一句話比喻：

Apps Script 就像 Excel 的「巨集」，但住在雲端、而且指揮得動 Google 所有服務。



# Apps Script 能幫你做什麼？

今天的三大實戰 (★) 就從這張清單裡來

## ★ 試算表自動寄信

名單放在試算表，一鍵寄出  
所有通知信、成績單

■ 試算表

✉ Gmail

## ★ 雲端硬碟自動整理

依檔案類型 / 日期自動分類，  
下載夾不再爆炸

▲ 雲端硬碟

## ★ 表單自動排進日曆

有人填預約表單，日曆  
自動出現行程

■ 表單

■ 日曆

## 定時自動報表

每週一 8 點自動彙整資料，  
寄摘要給主管

■ 試算表

✉ Gmail

## 郵件自動回覆 / 分類

關鍵字自動貼標籤，  
自動回覆常見問題

✉ Gmail

## 庫存 / 到期提醒

數量低於安全量、合約快到期  
自動通知承辦人

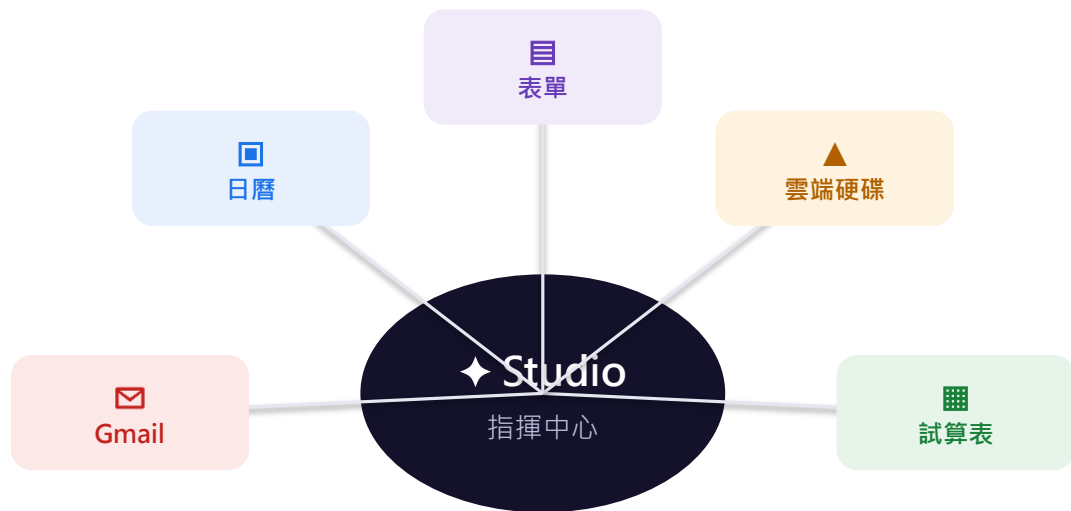
■ 試算表

✉ Gmail

**共同套路：**讀資料 → 依規則處理 → 對另一個服務做動作。看懂這個套路，你就能舉一反三。

# Google Workspace Studio：自動化的指揮中心

用「說的」就能打造 workflow——與 Apps Script 是最佳拍檔



## 它是什麼？

Google 推出的自動化平台：用自然語言描述需求，AI 幫你串聯 Workspace 各服務，產生自動化流程。

## 跟 Apps Script 什麼關係？

Studio = 方向盤（設計流程），Apps Script = 引擎（實際執行）。看得懂引擎，才能微調出精準可靠的流程。

## 本課策略

免費帳號人人可用的路線：Gemini + Apps Script 徒手打造，觀念完全相通。

你只要說「表單有人填就幫我排日曆」，  
Studio 幫你把流程串好，背後的引擎正是 Apps Script。

# 值得嗎？算一筆時間帳

以一位班級導師 / 行政人員的日常為例 ( 每週工時估算 )

| 例行工作      | 手動花費  | 自動化後          | 每週省下 |
|-----------|-------|---------------|------|
| 寄 30 封通知信 | 90 分鐘 | 0 分鐘 ( 自動寄出 ) | 90 分 |
| 整理下載資料夾   | 40 分鐘 | 0 分鐘 ( 自動歸檔 ) | 40 分 |
| 報名資料抄進日曆  | 30 分鐘 | 0 分鐘 ( 即時同步 ) | 30 分 |
| 彙整週報寄主管   | 60 分鐘 | 5 分鐘 ( 過目確認 ) | 55 分 |

\* 首次設定約 30-60 分鐘 ( 有 Gemini 幫忙 )，之後每週純收割。



# 實作 ①：第一次打開 Apps Script

跟著做，兩分鐘進入編輯器

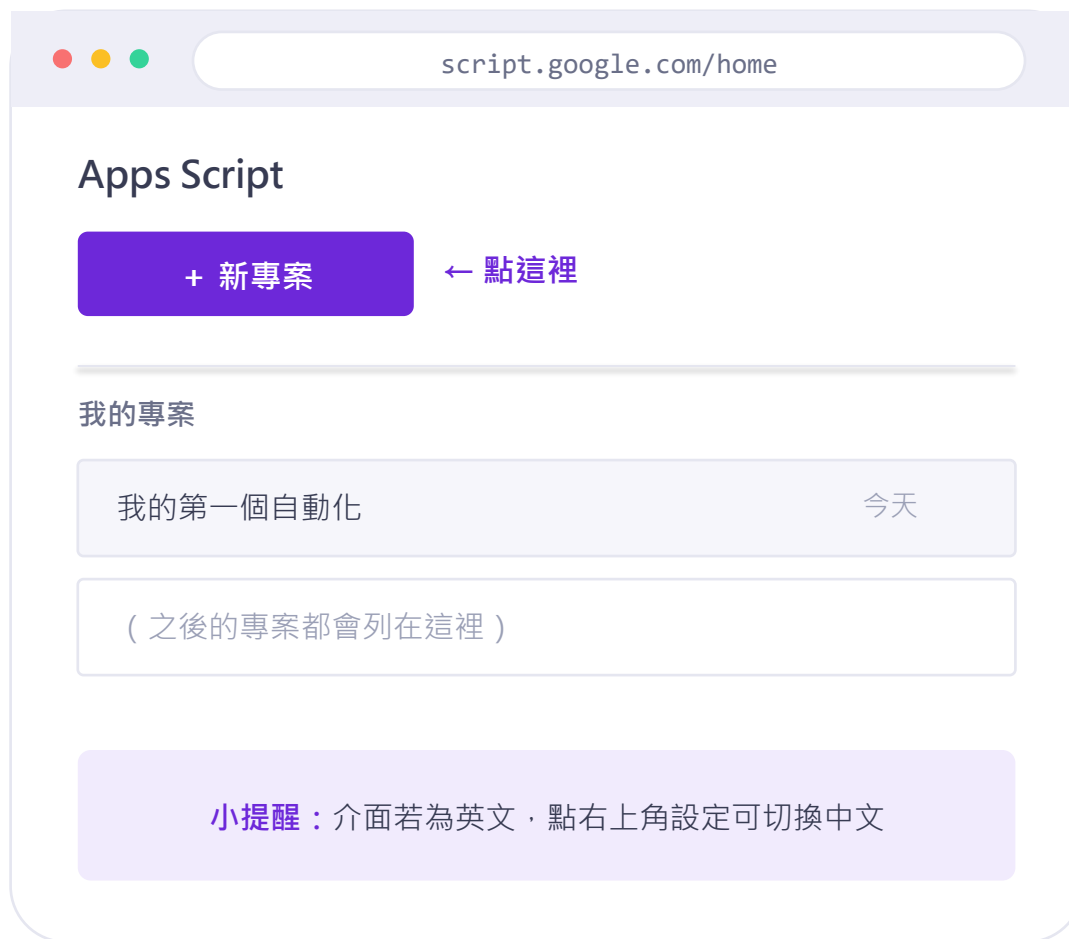
1 打開瀏覽器，登入你的 Google 帳號  
右上角頭像確認是「上課用」的帳號

2 網址列輸入 `script.google.com`  
建議加入書籤，之後天天用得到

3 點左上角「+ 新專案」  
會開出一個空白的程式編輯器

4 點上方「未命名的專案」改名  
例如：我的第一個自動化

另一條路：試算表 ▸ 擴充功能 ▸ Apps Script ( 直接綁定該表格 )



# 認識編輯器：只需要看懂五個地方

其他按鈕先不用管——今天用到的就這五個

The screenshot shows the Google Apps Script editor interface. It features a top bar with a title, a '儲存' (Save) button, and an '執行' (Run) button. On the left, there is a sidebar with '檔案' (Files) and '服務 +' (Services). The main area contains a code editor with a function definition. At the bottom, there is a section for '執行紀錄' (Execution Log).

1 我的第一個自動化

檔案

2 Code.gs

服務 +

觸發條件  
(第四章用)

```
3 function myFunction() {  
2    // 之後程式碼都貼在這裡  
3 }
```

4

5 執行紀錄

執行開始 / 執行完畢，錯誤訊息也會出現在這裡

- 1 專案名稱  
點一下就能改名
- 2 Code.gs 檔案  
程式碼住的檔案
- 3 程式碼區  
Gemini 給的程式碼貼這裡 ( 先全選刪掉原內容 )
- 4 執行 ▶  
按下去程式就開始跑
- 5 執行紀錄  
成功或失敗都看這裡；錯誤訊息從這裡複製

## 實作 ②：跑出你的第一支程式

Hello World——確認引擎發得動，順便看懂「授權」是怎麼回事

Code.gs

```
function sayHello() {  
  // 在執行紀錄印出一句話  
  Logger.log("你好，自動化世界！");  
}
```

### 逐行白話

- `function sayHello() {}` 一包名叫 `sayHello` 的動作，`{}` 裡是要做的事
- `//` 開頭的字 「註解」= 寫給人看的說明，程式會跳過
- `Logger.log("...")` 把引號裡的文字印到執行紀錄

- 1 把左邊程式碼輸入（或貼入）編輯器  
取代原本的 `myFunction`
- 2 按  儲存，再按 ▶ 執行  
第一次執行會跳出「需要授權」視窗
- 3 點「審查權限」→ 選你的帳號  
出現警告畫面是正常的（見下方說明）
- 4 點「進階」→「前往專案（不安全）」→ 允許  
執行紀錄出現：你好，自動化世界！

### 為什麼 Google 會警告「未驗證」？

因為這支程式是「你自己」剛寫的，還沒送 Google 審核。程式只在你的帳號裡、動你自己的資料——自己寫的程式，可以放心允許。

# 02

◆ PART 2

## Gemini 助攻——無痛撰寫 Apps Script

你負責說清楚需求，程式碼交給 AI

01

為什麼讓 Gemini 寫？  
你和 AI 的分工

02

提示詞黃金公式  
四個要素一次學會

03

實作：第一次請 AI  
寫程式並執行成功

04

出錯怎麼辦？  
錯誤訊息丟回去就好

# 為什麼讓 Gemini 寫程式？

從此你的工作不是「學語法」，而是「說清楚需求」



# 跟 Gemini 溝通的黃金公式

好提示詞 = 角色 + 目標 + 細節 + 要求——四塊拼圖拼齊就贏

1

## 角色

告訴它「你是誰」

範例句

你是 Google Apps Script 專家，  
擅長教完全沒有程式基礎的人

2

## 目標

想完成「哪件事」

範例句

我想讓 Google 試算表的名單  
自動寄出個人化郵件

3

## 細節

資料長什麼樣子

範例句

工作表名稱「名單」；A 欄姓名  
、  
B 欄 Email、C 欄成績

4

## 要求

希望怎麼交付

範例句

請給完整程式碼 + 每行中文註解  
，  
並告訴我貼在哪裡、怎麼執行

新手最常漏掉的是 ③ 細節：欄位在哪一欄、工作表叫什麼名字。AI 不是通靈——資料描述得越清楚，程式一次就能跑對。

# 萬用提示詞範本（收藏這一頁）

之後所有實戰都用這個骨架，換掉【】裡的内容就好

💡 提示詞 | 直接複製給 Gemini

你是 Google Apps Script 專家，擅長教完全沒有程式基礎的人。

我想要：【用一句話描述要自動化的事】

我的資料：【工作表 / 資料夾名稱、每一欄放什麼】

請你：

1. 給我完整、可直接執行的程式碼
2. 每一行加上中文註解
3. 一步一步告訴我要貼在哪裡、怎麼執行
4. 告訴我第一次執行會遇到什麼（例如授權）

## ◆ 一次只要一個功能

先讓「寄一封信」成功，再加「個人化稱呼」——  
小步快跑最不容易卡住

## ◆ 欄位名稱照實寫

「B 欄是 Email」比「有email」精準十倍

## ◆ 固定要求中文註解

之後回頭看程式碼，每行都看得懂

## 實作 ③：第一次請 Gemini 寫程式

目標：讓 Apps Script 寄一封測試信到自己的信箱

💬 提示詞 | 直接複製給 Gemini

你是 Google Apps Script 專家，擅長教完全沒有程式基礎的人。

我想要：執行程式時，寄一封主旨為「自動化測試成功！」的郵件  
到我自己的 Gmail 信箱，內容寫「這封信是 Apps Script 自動寄出的」。

請給完整程式碼 + 每行中文註解，並告訴我貼在哪裡、怎麼執行。

### Gemini 會回給你什麼？

- ① 一段程式碼（黑底區塊，右上角有「複製」鈕）
- ② 貼上與執行的步驟說明
- ③ 授權提醒

- 1 開 [gemini.google.com](https://gemini.google.com)  
貼上左邊的提示詞送出
- 2 點程式碼區塊右上「複製」
- 3 貼到 Apps Script 編輯器  
先全選刪掉舊內容再貼
- 4 儲存 → 執行 → 完成授權
- 5 打開 Gmail 收件匣  
看到那封信 = 成功 🎉

**寄給自己** 是最安全的測試方式，  
之後所有實戰都先這樣驗證。

## 拿到程式碼之後：固定三步驟

今天之內會重複做很多次——把肌肉記憶練起來



**小技巧：** Gemini 若一次給好幾段程式碼，先問它「請合併成一個可以直接執行的完整版本」再複製。

# 看懂程式碼的「最低限度」

不用會寫，會「認」這四個東西就夠用一整天

Code.gs

```
function 寄成績通知() {  
  // 打開叫「名單」的工作表  
  var sheet = SpreadsheetApp  
    .getActiveSpreadsheet()  
    .getSheetByName("名單");  
  
  // 寄一封郵件  
  GmailApp.sendEmail(  
    "test@gmail.com",  
    "主旨", "內文");  
}
```

**function 名字( ) { }**

「一包動作」的外殼，執行時選它的名字

**// 開頭的整行**

中文註解 = AI 寫給你的說明書，讀它就知道每段在幹嘛

**"雙引號裡的字"**

可以放心改的部分：郵件主旨、工作表名稱都在這裡

**SpreadsheetApp / GmailApp**

「服務名稱」= 在指揮哪個 Google 工具，一看就知道

# 執行出錯了？三十秒除錯法

紅字不是世界末日，而是 Gemini 最愛吃的線索



🔄 還是錯？再走一次 ③→⑤。實務上 1-2 回合內幾乎都會修好。

## 除錯句型 | 原封不動照抄

我執行你給的程式碼後，出現以下錯誤訊息：

【貼上完整紅字錯誤訊息】

請解釋哪裡出錯（用白話），並給我修正後的完整程式碼。

## 為什麼這招有效？

- 錯誤訊息寫明了「第幾行、什麼問題」
- Gemini 讀得懂這些英文術語
- 連同原程式碼一起看，它能直接改好

# 新手常見錯誤對照表

先認得長相，遇到就不慌——都能用上一頁的除錯句型解決

| 你看到的訊息                                            | 白話意思                       | 怎麼辦                  |
|---------------------------------------------------|----------------------------|----------------------|
| Authorization is required / 需要授權                  | 程式第一次要動你的服務，要你點頭           | 點「審查權限」完成授權（見第 15 頁） |
| Cannot read properties of null ( getSheetByName ) | 找不到那個工作表——名稱打錯了            | 检查工作表名稱是否一字不差（含空格）   |
| ReferenceError: ... is not defined                | 有字打錯或程式碼沒複製完整              | 重新完整複製貼上，或丟回給 Gemini |
| Exceeded maximum recipients ( 配額 )                | 今天寄信額度用完了（免費帳號約 100 封 / 日） | 明天再寄，或減少測試寄送量        |
| Script runtime exceeded / 逾時                      | 資料太多，程式跑超過時間上限             | 請 Gemini 改成「分批處理」版本  |

八成的錯誤都是：**名稱打錯、沒貼完整、忘了授權**——先檢查這三件事。

# 程式會動了，怎麼「加菜」？

在同一個對話串裡追問——Gemini 記得前面的程式碼

與 Gemini 的對話

你

在原本的程式碼加上：寄完信之後，把該列的 D 欄填入「已寄送」

◆ Gemini

好的！以下是加入「寄送後標記」的完整程式碼：

```
sheet.getRange(i+1, 4).setValue("已寄送");
```

( 附完整程式碼與逐行中文註解... )

幫我解釋第 8 行在做什麼？我想確認它不會蓋掉原本的資料

\* 持續同一個對話串 = AI 記得全部脈絡，越改越準。

## 好用的追問句型

加功能

「在原本的程式碼加上.....」

改範圍

「改成只處理 C 欄是空白的那些列」

求解釋

「用白話解釋第 X 行在做什麼」

要安全

「執行前，先告訴我這段程式會動到哪些資料」

# 提示詞的好與壞：一眼看懂差距

同一個需求，兩種問法——結果天差地遠

## ✗ 模糊的問法

「幫我寫一個寄信的程式」

AI 只能猜：寄給誰？內容？從哪拿資料？——猜錯就要來回改十次

## ✓ 清楚的問法

「試算表『名單』工作表：A 欄姓名、B 欄 Email。請逐列寄信，主旨『繳費提醒』，內文開頭稱呼 A 欄的名字」

一次到位——欄位、動作、格式全都講清楚

## 送出前 10 秒自我檢查

✓ 說了資料在哪嗎？

工作表 / 資料夾名稱、欄位位置

✓ 說了要做什麼動作嗎？

寄信？搬檔案？建行程？

✓ 說了細節格式嗎？

主旨、稱呼、資料夾命名規則

✓ 要求中文註解了嗎？

之後看得懂、才敢改

## 第二章小結 + 隨堂練習

帶著這三句心法，進入今天的重頭戲

1

### 需求說得清，程式寫得對

黃金公式：角色 + 目標 + 細節 + 要求

2

### 固定三步驟

複製 → 全選貼上 → 執行；紅字整段丟回給 Gemini

3

### 小步快跑

一次一個功能，跑通了再加菜

### 隨堂練習

請 Gemini 寫一支程式：執行時在「執行紀錄」印出今天日期 + 一句給自己的鼓勵

#### 起手式（可以自己改造）

你是 Apps Script 專家。請寫一支程式：

執行時用 `Logger.log` 印出今天的日期（`yyyy/MM/dd`），

以及一句隨機的中文鼓勵語（從 5 句裡隨機挑一句）。

請附中文註解，並告訴我怎麼執行、去哪裡看結果。

做完的同學：試著追問「改成印出星期幾」

# 03

◆ PART 3

## 跨工具串接實戰（重點核心）

三個馬上能帶回去用的自動化——每一個都從情境到完成

01

實戰一  
試算表自動寄郵件

02

實戰二  
雲端硬碟自動整理資料夾

03

實戰三  
表單回覆自動同步日曆

04

總複習  
看懂三招共同的套路

# 情境：30 封通知信，寄到懷疑人生

主角：班級導師「小榮老師」——每學期都要來一次的惡夢

## 小榮老師的期中夜晚

1. 打開成績試算表，看著 30 位學生的名單
2. 複製第 1 位的成績 → 開新郵件 → 貼上 → 改稱呼 → 寄出
3. 複製第 2 位.....做到第 17 位時，把 A 生成績寄給了 B 家長
4. 重寄道歉信，繼續做到晚上 11 點

人工複製貼上：慢、累，而且一定會出錯

90

分鐘 / 30 封

1

次寄錯 = 一場災難

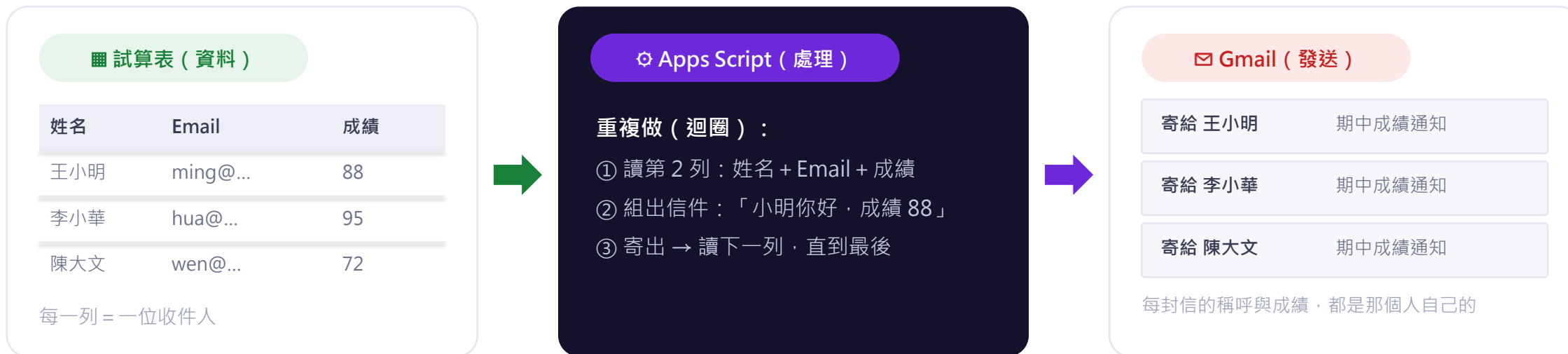
3

秒——自動化後全部寄完

**目標：**名單放在試算表 → 按一次執行（或時間到自動跑）→ 每個人收到「寫著自己名字與成績」的信，寄完自動標記。

# 原理：三個服務怎麼接力？

先看懂流程圖，等一下的程式碼就只是「把這張圖翻譯成文字」



## 關鍵觀念——迴圈 ( Loop )：

「對名單裡的每一列，重複做同一件事」。30 列就自動做 30 次，300 列也一樣輕鬆——這正是電腦最擅長、人類最痛苦的部分。

# 實作 ④-1：先把試算表準備好

資料整齊，自動化就成功了一半

■ 成績通知（試算表） | 工作表：名單

|   | A 姓名 | B Email            | C 成績 | D 狀態 |
|---|------|--------------------|------|------|
| 2 | 王小明  | ming2026@gmail.com | 88   |      |
| 3 | 李小華  | hua.li@gmail.com   | 95   |      |
| 4 | 陳大文  | dawen.c@gmail.com  | 72   |      |
| 5 | 林美琪  | maggie.l@gmail.com | 81   |      |

\* D 欄「狀態」先留空——程式寄完會自動填「已寄送」

## 名單設計三原則

- 1

第 1 列一定是標題

程式從第 2 列開始讀資料
- 2

一格只放一種資料

Email 獨立一欄、不要跟姓名擠在一起
- 3

加一欄「狀態」

記錄已寄過的人，重跑也不會重複寄

**課堂操作：**開一份新試算表，照上圖建好 4 欄，放 3 筆「自己的 Email」當測試資料。

## 實作 ④-2：給 Gemini 的提示詞

套用黃金公式——注意「細節」寫得多具體

💬 提示詞 | 直接複製給 Gemini

你是 Google Apps Script 專家，擅長教沒有程式基礎的人。

我想要：從試算表名單自動寄出個人化成績通知信。

我的資料：工作表名稱「名單」；A 欄姓名、B 欄 Email、C 欄成績、D 欄狀態。第 1 列是標題。

請你：

1. 逐列寄信：主旨「期中成績通知」，內文「〔姓名〕你好，你的期中成績是〔成績〕分，請查收。」
2. 只寄 D 欄是空白的列；寄出後把 D 欄填「已寄送」
3. 完整程式碼 + 每行中文註解 + 執行步驟

### 為什麼要「只寄空白列」？

程式中斷重跑時，已寄過的人不會再收到一次——這叫「防重寄」設計

### 為什麼內文用〔姓名〕？

明確告訴 AI 哪裡要「代入變數」，每封信才會是那個人的名字

### 忘了公式細節？

回看第 19 頁的萬用範本，【】換成你的資料就好

## 實作 ④-3：看懂 Gemini 給的程式碼

左邊是程式碼（精簡版），右邊是每一段的白話翻譯

Code.gs

```
function 寄成績通知() {  
  // ① 打開試算表的「名單」工作表  
  var sheet = SpreadsheetApp.getActive()  
    .getSheetByName("名單");  
  // ② 把所有資料一次讀進來  
  var data = sheet.getDataRange().getValues();  
  // ③ 從第 2 列開始，一列一列處理  
  for (var i = 1; i < data.length; i++) {  
    if (data[i][3] == "") {           // 只處理狀態空白  
      GmailApp.sendEmail(data[i][1],  
        "期中成績通知",  
        data[i][0] + "你好，你的期中成績是"  
        + data[i][2] + "分，請查收。");  
      sheet.getRange(i+1, 4).setValue("已寄送");  
    }  
  }  
}
```

### ① 打開工作表

指名要動哪份資料——名稱一定要跟你的工作表一模一樣

### ② 一次讀入全部

data 就像把整張表格搬進記憶裡

### ③ 迴圈逐列處理

i 從 1 開始 = 跳過標題列；每一圈處理一位

### ④ 條件判斷 if

D 欄空白才寄——這就是「防重寄」

### ⑤ 寄信 + 標記

sendEmail 寄出；setValue 把「已寄送」寫回 D 欄

## 實作 ④-4：執行與驗證

專業做法：先用假名單測試，確認無誤才上真名單

- 1 名單的 Email 全部先填自己的信箱  
3 筆測試資料，收信的都是你
- 2 貼上程式碼 → 儲存 → 執行  
第一次會要求授權（試算表 + Gmail 兩項）
- 3 打開收件匣檢查  
應收到 3 封，稱呼與成績「各自不同」
- 4 回試算表看 D 欄  
3 列都變成「已寄送」
- 5 再按一次執行  
不會再寄 = 防重寄成功；之後換上真名單

### ✉ 收件匣（你的 Gmail）

|       |                                |
|-------|--------------------------------|
| 我（自己） | 期中成績通知 — 王小明你好，你的期中成績是 88 分... |
| 我（自己） | 期中成績通知 — 李小華你好，你的期中成績是 95 分... |
| 我（自己） | 期中成績通知 — 陳大文你好，你的期中成績是 72 分... |

### 上真名單前的三個確認

- Email 欄拼字檢查過一遍（錯的信箱寄不到）
- 內文以自己名字讀一次，語氣沒問題
- 免費帳號每天上限約 100 封，大名單分天寄

## 實戰一進階：三種常用變化

基本版會動之後，用「追問」直接升級——句型都幫你寫好了

### 變化 1

#### 更有溫度的稱呼

依性別欄自動加「先生 / 小姐」，或加上「親愛的」

#### 追問句

在原本程式碼加上：E 欄是稱謂，內文開頭改成「親愛的〔姓名〕〔稱謂〕」

### 變化 2

#### 附上個人附件

每人附上自己的成績單 PDF ( 放在雲端硬碟 )

#### 追問句

加上：F 欄是雲端硬碟檔案連結，寄信時把該檔案作為附件

### 變化 3

#### 指定時間寄送

先寫好，週五 17:00 統一寄出 ( 結合第四章觸發器 )

#### 追問句

加上：如何設定成每週五下午 5 點自動執行？

## 實戰一可以用在哪？

只要是「名單 + 一人一信」的工作，全部適用同一套程式

### 教育現場

- 成績通知、缺曠提醒
- 催交作業（只寄未交名單）
- 家長會通知與報名確認

### 行政職場

- 繳費 / 續約到期提醒
- 會議通知與行前提醒
- 員工生日祝福（當天自動寄）

### 小店經營

- 訂單出貨通知
- 會員點數到期提醒
- VIP 專屬優惠通知

### 生活社群

- 社區繳費通知
- 球隊 / 社團活動點名
- 讀書會場次提醒

換場景時只要改三個地方：欄位對應、郵件主旨、內文模板——提示詞改一改，兩分鐘生出新版本。

## 情境：雲端硬碟 = 數位垃圾山？

收作業、收照片、收報價單.....什麼都往同一個資料夾丟

### ▲ 雲端硬碟 ▶ 「收件匣」資料夾 (現況)

|                                                                                                           |                                                                                                               |
|-----------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
|  IMG_20260712_133045.jpg |  作業-王小明(3).docx              |
|  scan0001.pdf            |  報價單final_v2(1).xlsx         |
|  LINE_ALBUM_0715.jpg     |  作業王小明 最新版.docx              |
|  未命名文件.docx              |  IMG_8842.HEIC               |
|  發票2026-06.pdf           |  新增 Microsoft Excel 工作表.xlsx |

.....以及另外 137 個檔案

### 10 分鐘

找一個檔案的平均時間

### (3)、final、最新版

檔名地獄，不知哪個是對的

### 目標

執行一次程式 (或每天自動跑)：

圖片進「圖片」夾、PDF 進「PDF」夾、文件進「文件」夾  
——收件匣永遠乾乾淨淨。

# 原理：檔案自動分家

一個來源、一套規則、多個去處——「分流」是整理的核心



關鍵觀念——條件判斷 (if)：

「如果檔名以 .pdf 結尾 → 搬到 PDF 資料夾」。規則你用中文告訴 Gemini，它翻譯成程式——規則越清楚，分得越漂亮。

## 實作 ⑤-1：準備資料夾 + 提示詞

先建好資料夾，再把「資料夾名稱」原封不動寫進提示詞

1 到雲端硬碟建 1 個來源資料夾  
命名「收件匣」

2 建 3 個分類資料夾  
「圖片」「PDF」「文件」

3 丟 5-6 個測試檔案進收件匣

### 進階提示：資料夾 ID

同名資料夾可能有很多個。打開資料夾時，網址最後一長串就是它的專屬 ID——告訴 Gemini 用 ID 指定，最不會搬錯家。

💬 提示詞 | 直接複製給 Gemini

你是 Google Apps Script 專家，擅長教沒有程式基礎的人。

我想要：自動整理雲端硬碟「收件匣」資料夾裡的檔案。

規則：

- 圖片檔 (jpg、jpeg、png、heic) → 移到「圖片」資料夾
- PDF 檔 → 移到「PDF」資料夾
- Word 與 Google 文件 → 移到「文件」資料夾
- 其他檔案 → 留在原地不動

請你：給完整程式碼 + 中文註解；執行後在紀錄裡  
印出「共搬移了幾個檔案」，方便我確認。

## 實作 ⑤-2：程式碼解讀 + 執行

又見老朋友：迴圈掃檔案、if 判斷分類——套路完全一樣

Code.gs

```
function 整理收件匣() {  
  // 找到來源與目的資料夾 (依名稱)  
  var src = DriveApp.getFoldersByName("收件匣").next();  
  var imgF = DriveApp.getFoldersByName("圖片").next();  
  var files = src.getFiles(); // 收件匣所有檔案  
  while (files.hasNext()) { // 一個一個檢查  
    var f = files.next();  
    var name = f.getName().toLowerCase();  
    if (name.endsWith(".jpg") || name.endsWith(".png")) {  
      f.moveTo(imgF); // 搬去「圖片」  
    }  
  }  
  // ...PDF、文件的判斷同理  
}
```

### 白話對照

- `getFoldersByName` 用名稱找到資料夾
- `while (hasNext)` 「還有下一個檔案就繼續」的迴圈
- `endsWith(".jpg")` 檔名結尾是不是 .jpg
- `moveTo` 把檔案搬到指定資料夾

### 執行結果

- ▶ 收件匣 **147 → 12 個檔案**
- ▶ 圖片 **0 → 68 個檔案**
- ▶ PDF **0 → 41 個檔案**
- ▶ 文件 **0 → 26 個檔案**

## 實戰二進階：三種常用變化

搬完家還能改名、建月份資料夾——追問就有

### 變化 1

#### 按月份自動歸檔

自動建立「2026-07」這種月份資料夾，照片依拍攝月份進駐

#### 追問句

改成：依檔案的建立日期，自動建立並移動到「yyyy-MM」格式的月份資料夾

### 變化 2

#### 自動統一檔名

「作業-王小明(3).docx」→「20260730\_王小明\_作業.docx」

#### 追問句

加上：搬移前把檔名改成「日期\_人名\_類型」格式，人名從原檔名擷取

### 變化 3

#### 每天自動打掃

睡前 11 點自動整理一次，早上永遠看到乾淨的收件匣

#### 追問句

加上：如何設定每天 23:00 自動執行？（第四章會親手做）

## 實戰二可以用在哪？

所有「檔案一多就亂」的地方，都值得放一支自動整理程式

### 收作業自動歸檔

學生交到共用資料夾，程式依「檔名裡的班級 / 姓名」分班歸檔，缺交名單一目了然

### 相片庫自動分月

活動照片全丟一個夾，程式依日期分成月份資料夾，年底找照片不再大海撈針

### 單據報帳歸檔

發票、收據掃描後，檔名含「發票」自動進當月報帳資料夾，月底一次匯整

### 共用雲端的秩序

部門共用槽訂好命名規則，程式每天巡邏，亂放的檔案自動搬回正確位置

搭配第四章的「每日觸發器」，這支程式會變成 24 小時待命的數位管家。

## 情境：報名表單和日曆之間，隔著一個「你」

表單收單很方便——但行程還是要有人一筆一筆抄進日曆

### 諮商預約的一天（自動化前）

- |       |                  |             |
|-------|------------------|-------------|
| 09:12 | 家長 A 填了預約表單      | → 你在上課，沒看到  |
| 12:30 | 午休打開回覆表，抄 3 筆進日曆 | → 抄錯一個日期    |
| 15:40 | 家長 B 也填了表單       | → 你在開會，又沒看到 |
| 21:00 | 睡前想起來沒抄，爬起來開電腦   | → 隔天還是撞了時段  |

### 人工轉抄的兩大風險

- ① 忘了抄 → 空等或撞時段    ② 抄錯 → 白跑一趟

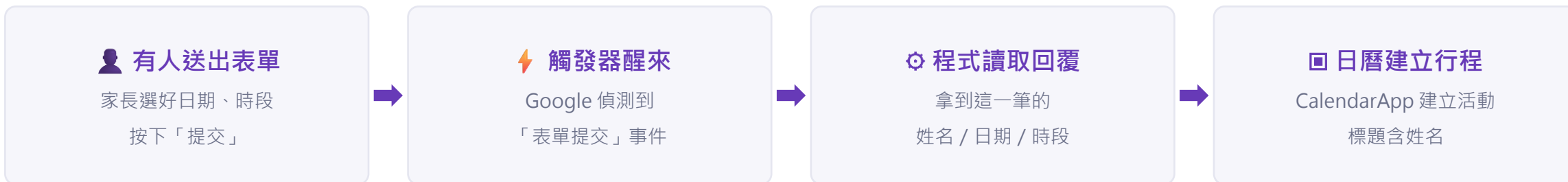
### 目標

有人「送出表單」的那一秒，日曆自動長出一筆行程——標題、日期、時段全部到位，你連電腦都不用開。

新概念登場：這次不是「按執行」，而是「事件觸發」——表單一送出，程式自己醒來。

## 原理：讓「送出表單」變成發令槍

事件觸發 ( onFormSubmit ) —— 自動化從「你來按」進化成「它自己來」



### 「按鈕執行」vs「事件觸發」

#### 實戰一・二 ( 手動執行 )

你按 ▶ 或時間到 → 程式跑一次 → 處理「一批」資料

#### 實戰三 ( 事件觸發 )

事件發生的瞬間自動跑 → 每次處理「觸發它的那一筆」

好消息：程式碼裡多一個參數 e ( 事件內容 ) 而已，其他套路不變——Gemini 都會幫你寫好。

## 實作 ⑥-1：建立預約表單

題型選對，後面的程式就簡單——日期用「日期」題、時段用「選擇題」

### 家長諮詢預約表

選擇方便的時間，我們會安排諮詢

您的姓名 · 簡答

王家長

聯絡 Email · 簡答

parent@gmail.com

希望日期 · 日期 📅

2026/08/05

希望時段 · 選擇題 ●

☒ 14:00–15:00 ☐ 15:00–16:00

### 表單設計要點

1

#### 日期一定用「日期」題型

程式拿到的才是標準日期，不會遇到「8月5號下午」這種要用猜的文字

2

#### 時段用選擇題鎖定選項

14:00–15:00 固定格式，程式才能準確換算成行程的起訖時間

3

#### 表單自動連到試算表

「回覆」分頁按綠色圖示——所有回覆自動存進試算表，等於免費的預約資料庫

課堂操作：花 3 分鐘建好這張表單

## 實作 ⑥-2：提示詞 + 程式碼

注意提示詞裡把「觸發」需求說出來——Gemini 會連設定教學一起給

### 💬 提示詞 | 直接複製給 Gemini

你是 Google Apps Script 專家。

我想要：每當有人送出「家長諮詢預約表」，  
自動在我的 Google 日曆建立行程。

表單題目依序是：

姓名、Email、希望日期（日期題）、  
希望時段（選項如「14:00–15:00」）。

行程要求：

標題「諮詢：〔姓名〕」；日期時段照表單；  
說明欄放家長 Email。請在表單綁定的 Apps Script  
撰寫，並教我設定「表單提交」觸發器。

表單專案 / Code.gs

```
function onFormSubmit(e) { // e = 這筆回覆
  var v = e.values;        // [時間戳, 姓名, Email, 日期, 時段]
  var start = new Date(v[3] + " " + v[4].slice(0,5));
  var end = new Date(start.getTime() + 60*60*1000);
  CalendarApp.getDefaultCalendar()
    .createEvent("諮詢：" + v[1],
      start, end, {description: v[2]});
}
```

### 兩個新面孔

- **e.values** 這筆回覆的所有答案（照題目順序排）
- **createEvent** 在日曆建立行程：標題、開始、結束

## 實作 ⑥-3：綁上觸發器，然後見證奇蹟

這一步做完，「送出表單 → 日曆長行程」就全自動了

- 1 表單 ▶ : ▶ Apps Script  
從表單進入，程式才「認得」這張表單
- 2 貼上程式碼並儲存
- 3 左側點 🕒 「觸發條件」→ 右下「+ 新增觸發條件」
- 4 事件來源選「來自表單」，事件類型選「提交表單時」  
函式選 onFormSubmit
- 5 儲存並完成授權  
表單 + 日曆兩項權限
- 6 自己填一次表單測試  
打開日曆——行程已經在那裡了 🎉

### 新增觸發條件

要執行的函式

onFormSubmit ▼

選取活動來源

來自表單 ▼

選取活動類型

提交表單時 ▼

儲存

### 從此刻起：

凌晨三點有人填表單，日曆照樣自動更新——這就是「事件驅動」的威力。

## 實戰三可以用在哪？

任何「填單→排時間」的流程，都能就地全自動

### 📅 諮詢 / 面談預約

家長諮詢、選課輔導、  
一對一面談，填單即排程

### 🔍 設備 / 場地借用

會議室、球場、器材——  
借用時段直接進公用日曆

### 👨‍🏫 家教 / 課程排課

學生選時段，你的教學  
日曆自動長出課表

### 🔧 到府服務排單

客戶預約維修時段，  
師傅日曆即時更新

### ◆ 組合技預告：三個實戰其實可以串成一條龍

家長填預約表單 → 日曆自動排程（實戰三）→ 自動寄確認信給家長（實戰一）→ 諮詢紀錄檔自動歸檔（實戰二）。

下一章學會觸發器之後，你就能自己設計這種多步驟工作流。

# 三大實戰總複習：看穿同一個套路

表面是三個不同功能，骨子裡是同一條公式

|        | 資料從哪來  | 程式做什麼   | 結果去哪裡     | 怎麼啟動         |
|--------|--------|---------|-----------|--------------|
| 實戰一 寄信 | 試算表名單  | 迴圈逐列組信  | Gmail 寄出  | 按 ▶ ( 可改定時 ) |
| 實戰二 整理 | 收件匣資料夾 | if 判斷分類 | 各分類資料夾    | 按 ▶ ( 可改每日 ) |
| 實戰三 排程 | 表單這筆回覆 | 讀答案組行程  | Google 日曆 | 表單提交自動觸發     |

## 萬用公式



# 04

◆ PART 4

## 打造客製化工作流

設定觸發器，讓程式自己準時上工——徹底擺脫手動

01

觸發器是什麼？  
時間驅動 vs 事件驅動

02

實作：每天 8 點  
自動執行的鬧鐘

03

完整案例：每週一  
自動彙整週報寄主管

04

設計你自己的  
專屬自動化工作流

# 觸發器：幫程式裝上鬧鐘與感應器

你不在電腦前，程式照樣執行——這才叫「全自動」

到目前為止，程式要「你按 ▶」才動。觸發器（Trigger）= 替程式設定「什麼時候自己動」的規則，設定一次、天天有效。

## 時間驅動（像鬧鐘）

「時間到了就執行」——適合定期批次工作

- 每天 08:00 自動整理收件匣資料夾
- 每週五 17:00 自動寄出本週成績通知
- 每月 1 號 自動彙整上月報帳資料
- 每小時 檢查庫存表，低於安全量就通知

## 事件驅動（像感應器）

「事情發生就執行」——適合即時反應

- 提交表單時 實戰三：預約自動進日曆
- 試算表被編輯時 主管改了排班表就通知組員
- 檔案開啟時 打開報表自動更新日期
- 日曆更新時 行程異動同步到試算表

選擇心法：固定週期的批次工作用「時間」；要立刻反應的用「事件」。

## 實作 ⑦：幫「整理收件匣」裝上每日鬧鐘

以實戰二的程式為例——每天早上 8 點自動打掃雲端硬碟

- 1 打開實戰二的 Apps Script 專案  
就是「整理收件匣」那支程式
- 2 左側選單點  「觸發條件」  
進入觸發器管理頁
- 3 右下角「+ 新增觸發條件」
- 4 函式選「整理收件匣」；來源選「時間驅動」
- 5 類型選「日計時器」；時段選「上午 8 點～9 點」  
Google 會在這個區間內執行
- 6 儲存 → 完成  
從明天起，每天自動打掃✓

### 新增觸發條件

要執行的函式

整理收件匣 ▾

選取活動來源

時間驅動 ▾

選取時間型觸發條件類型

日計時器 ▾

選取時段

上午 8 點到 9 點 ▾

儲存

**小知識：**「8 點～9 點」表示會在區間內某一刻執行，  
不是整點準時——排程留彈性即可。

# 觸發器類型總覽（速查表）

拍下這一頁——之後設計 workflow 時照表點菜

| 類型         | 什麼時候執行      | 最適合的工作       | 課堂對應     |
|------------|-------------|--------------|----------|
| 日計時器       | 每天固定時段      | 每日整理資料夾、每日晨報 | 實戰二 + ⑦  |
| 週計時器       | 每週固定日 / 時段  | 週報彙整、每週提醒信   | 第 54 頁案例 |
| 月計時器       | 每月固定日期      | 月報、對帳、月費提醒   | 報帳歸檔     |
| 分鐘 / 小時計時器 | 每 N 分鐘 / 小時 | 庫存監控、緊急信箱檢查  | 進階應用     |
| 提交表單時      | 有人送出表單      | 預約排程、報名確認信   | 實戰三      |
| 編輯 / 變更時   | 試算表被修改      | 排班異動通知、狀態看板  | 進階應用     |
| 開啟時        | 檔案被打開       | 自動更新日期、載入選單  | 進階應用     |

## 完整案例：每週一 8 點，週報自己寄給主管

把「讀試算表 + 寄信 + 定時」三個學過的零件，組成一條自動生產線



### 💬 一次到位的需求描述

你是 Apps Script 專家。請寫一支「每週自動週報」程式：

1. 讀取試算表「業績」工作表 B2:E2 的四個數字  
( 本週營收、訂單數、新客數、客訴數 )
  2. 寄郵件給 boss@company.com，主旨「週報 yyyy/MM/dd」，  
內文條列四個數字，並副本給我自己
  3. 教我設定「每週一上午 8-9 點」的週計時器
- 請附完整程式碼與中文註解。

### 用到的零件

讀試算表

實戰一學過

寄 Gmail

實戰一學過

週計時器

實作⑦學過

沒有任何新知識——只是把學過的積木疊在一起。這就是「 workflow 設計」。

# 換你了：設計自己的工作流

回答四個問題，答案拼起來就是你的提示詞

1

哪件事讓你最煩？

每週重複、規則固定的那件

例：每週要提醒 20 位家長交聯絡簿

2

資料放在哪裡？

試算表？表單？資料夾？

例：家長名單在試算表「聯絡簿」

3

要自動做什麼動作？

寄信？搬檔案？建行程？

例：寄 LINE 風格的簡短提醒信

4

什麼時候執行？

固定時間？還是事件發生時？

例：每週四下午 4 點

💬 四個答案 = 一份完整提示詞

你是 Apps Script 專家。我想自動化：【① 的答案】。我的資料：【② 的答案 + 欄位說明】。

請自動：【③ 的答案】，並教我設定：【④ 的答案】的觸發器。附完整程式碼 + 中文註解 + 步驟。

# 自動化上路前：三張安全檢查表

讓程式「可靠地」替你上班——這三件事顧好就安心

## 額度限制

### 免費帳號寄信

約 100 封收件人 / 天

### 觸發器總時數

約 90 分鐘 / 天

### 單次執行上限

6 分鐘 ( 大量資料要分批 )

### 超過會怎樣？

當天暫停，明天自動恢復

## 安全與隱私

### 授權範圍

程式只動「你帳號裡」的資料

### 別人的程式碼

看不懂的不要亂貼亂授權

### 個資意識

名單含個資，分享試算表要小心

### 隨時反悔

Google 帳號設定可撤銷授權

## 維護與除錯

### 執行紀錄

左側「執行項目」看每次成敗

### 失敗通知

觸發器失敗，Google 會寄信提醒

### 改了要測

任何修改先手動 ▶ 測一次

### 留註解

三個月後的你會感謝現在的你

# 加碼：用 NotebookLM 當你的課後 AI 助教

把今天的講義與程式碼丟進去——之後想不起來就「問它」

- 1 打開 [notebooklm.google.com](https://notebooklm.google.com)  
用同一個 Google 帳號登入

- 2 新增筆記本，上傳今日講義 PDF  
也可以貼上你的程式碼與筆記

- 3 直接用中文問它

## 跟 Gemini 的分工

**Gemini**：幫你「寫新程式」

**NotebookLM**：幫你「複習與查找」上過的內容

## 課後你可以這樣問它：

「防重寄是在程式的哪一段？幫我再解釋一次」

「幫我整理今天三個實戰的提示詞範本」

「觸發器有哪些類型？做成一張複習表」

「把整理資料夾的步驟做成語音摘要」

\* 回答會附「來源引用」，點了直接跳到講義那一段

## 常見疑問，一次說清楚

課堂上最常被問的四個問題

Q

免費帳號真的夠用嗎？

夠。今天所有實作都在免費 Gmail 帳號完成。額度（寄信 100 封 / 日）對個人與班級應用綽綽有餘；學校的 GWS ePlus 帳號額度更高。

Q

完全沒碰過程式，跟得上嗎？

可以。你的工作是「把需求說清楚」+「複製貼上」+「驗收結果」。  
語法細節交給 Gemini——課程設計就是為零基礎打造的。

Q

程式會不會哪天突然罷工？

會動的程式不會自己壞。常見停擺原因：改了工作表名稱、額度用完、授權被撤銷。觸發器失敗時 Google 會寄通知信，照第 23 頁除錯法處理即可。

Q

公司 / 學校帳號能用嗎？

能，而且通常更好用。若管理員關閉了 Apps Script，請 IT 開通；GWS ePlus 用戶還可搭配 Workspace Studio 圖形化管理流程。

# 帶著走的資源與回家作業

今天是起點——回去挑一個「最煩的例行公事」開刀

## 常用連結（存書籤）

[script.google.com](https://script.google.com)

Apps Script 編輯器——自動化的家

[gemini.google.com](https://gemini.google.com)

你的程式代寫夥伴

[notebooklm.google.com](https://notebooklm.google.com)

課後複習 AI 助教

[developers.google.com/apps-script](https://developers.google.com/apps-script)

官方文件（丟給 Gemini 讀也行）

[workspace.google.com/studio](https://workspace.google.com/studio)

Workspace Studio（新一代流程平台）

## 回家作業（三選一，由淺入深）

### 入門

把實戰一改成「生日祝福信」：名單加生日欄，寄給當月壽星

### 進階

實戰二 + 觸發器：讓下載資料夾每天 23:00 自動整理

### 挑戰

用第 55 頁的四問句，從零打造一條自己的工作流，下次課分享

小訣竅：先讓它「會動」，再讓它「漂亮」。自動化跟寫作一樣，先求有、再求好。

◆ 課程總結

# 讓程式替你上班，把時間留給更重要的事

01

## 觀念與環境

自動化 = 觸發 + 動作 + 結果

02

## Gemini 助攻

說清楚需求 · AI 寫程式

03

## 三大實戰

寄信 · 整理 · 同步日曆

04

## 觸發器

設定一次 · 天天自動

記住今天最重要的一句話：規則固定、重複發生的事，就交給程式。

---

謝謝大家！ 賴家榮 | 甫東科技

輔英科技大學 · 樹德科技大學 · 屏東大學 講師 | SGS ISO 42001 AI 認證