

GitHub 履歷趨勢

現在有些求職的公司會要求將履歷表及自傳，以給Github的連結完成嗎？

是的，現在確實有部分公司會接受，甚至偏好用 GitHub 連結來看履歷與作品，但要看產業與職缺類型。

我幫你分成幾種情況說明：

✓ 哪些公司「會」接受 GitHub 履歷？

1 軟體工程 / AI / 資料科學相關職缺

像是：

- Python / AI / Machine Learning 工程師
- Data Scientist
- 後端工程師
- DevOps 工程師

這類公司通常本來就會看 GitHub。

例如：

- Google
- Microsoft
- Amazon
- TSMC (部分IT/資料職缺)

👉 他們常在履歷欄位直接寫：

GitHub / Portfolio URL

這種情況，用 GitHub 放：

- README 自我介紹
- 專案說明
- 成果展示
- PDF 履歷下載

是加分項目。

⚠️ 哪些公司「不會只接受 GitHub」？

1 傳統產業

2 行政 / 管理 / 醫療行政

3 公家機關

這類公司通常：

- 仍要求 PDF 履歷
- 或使用人力銀行平台 (104 / 1111)

他們可能：

- 不熟 GitHub
- 不會點外部連結
- 甚至公司電腦封鎖 GitHub

🎯 現在主流做法是什麼？

👉 最佳策略：兩者並行

✓ 標準做法

1. 提供 PDF 履歷 (正式版本)
2. 在履歷上附上 GitHub 連結

例如：

GitHub Portfolio:
<https://github.com/yourname/Resume>

這樣：

- HR 可以先看 PDF
- 技術主管可以看 GitHub 專案
- 顯得專業且數位化

💡 GitHub 履歷的優勢

如果你是技術背景 (像你之前提到 AI segmentation、TensorFlow 等)，GitHub 有這些優點：

- 可展示程式碼
- 可展示專案架構
- 可放模型結果圖
- 可持續更新 (不用重寄履歷)
- 顯示你熟悉 Git / 版本控制

對 AI / 醫療影像領域尤其加分。

📌 2026年的實際趨勢（簡單總結）

產業	只給 GitHub 可以嗎？
科技業	✓ 可以，甚至加分
AI / Data	✓ 非常建議
一般企業	⚠️ 建議還是附 PDF
公家機關	✗ 不建議只給 GitHub

如果你願意告訴我：

👉 你是想投「醫療AI相關」還是「一般企業資訊職」？

我可以幫你規劃：

- GitHub 履歷架構設計
- README 專業版配置
- 如何讓 HR 也看得懂技術內容

這樣會更精準。

大學剛畢業求職的（資管系）學生，是不是將：履歷，自傳，作品專案分目錄存放於 GitHub 中是這個意思嗎？

是的 👍 你理解的方向是對的。

對於**資管系剛畢業學生**來說，把：

- 履歷
- 自傳
- 作品專案

放在 GitHub 上「分目錄管理」，這是**專業且加分的做法**，但我建議用「更清楚的結構」，而不是單純亂放檔案。

我幫你設計一個**最適合新鮮人的 GitHub 履歷架構**。

✓ 建議做法（一個主 Repository）

建立一個專門的 Repository，例如：

Resume-Portfolio

然後目錄這樣設計：

```
Resume-Portfolio/
  └── README.md      ← 首頁（重點）
  └── Resume/
    └── Resume.pdf
    └── Resume.md

  └── Autobiography/
    └── Autobiography.md

  └── Projects/
    └── Project1-xxx/
    └── Project2-xxx/
    └── Project3-xxx/
```

🔥 關鍵重點：README 才是核心

HR 或主管不會點很多層資料夾。

他們會：

1. 打開 GitHub
2. 看到首頁 README
3. 決定要不要繼續看

所以 README 要像「網站首頁」。

📌 正確觀念

✗ 錯誤做法

- 只放檔案
- 沒說明
- 沒專案介紹
- 讓人自己找

✓ 正確做法

README 要包含：

- 個人簡介
 - 技能表
 - 專案縮圖與連結
 - PDF履歷下載連結
-

🎯 新鮮人建議比例

對資管系畢業生來說：

內容	重要度
專案作品	★★★★★
技術能力	★★★★★
履歷PDF	★★★
自傳	★★

👉 主管更在意「你會做什麼」，不是你寫了多少自傳。

💡 進階做法（更專業）

很多科技公司（像 TSMC IT 職缺、ASUS、Google）更喜歡看到：

- 每個專案獨立一個 Repository
- README 清楚說明：
 - 使用技術
 - 架構圖
 - Demo 截圖
 - 成果數據

🏆 最佳新鮮人版本（推薦）

👉 做兩層架構：

1 主 Repository：Resume-Portfolio

（總覽）

2 各專案獨立 Repository

例如：

Student-Management-System
AI-Image-Classification
Ecommerce-Website

主頁 README 用連結串起來。

這樣會看起來像：

一個有系統的工程師，而不是學生交作業。

🎓 以資管新鮮人來說

GitHub 履歷代表：

- 你會 Git
- 你有版本控制概念
- 你會寫技術文件
- 你專案是真的做過

這比只給 Word 或 PDF 強很多。

如果你願意，我可以幫你設計：

-  新鮮人專用 GitHub README 模板
-  資管系三個必備專案建議
-  HR 喜歡的排版格式
-  技術型 vs 一般企業型兩種版本

你比較想往哪個方向發展？

- 1 程式工程師
- 2 資料分析
- 3 IT管理 / MIS
- 4 研究所

告訴我方向，我幫你做「精準版規劃」。