**回溯到之前的版本**

Git 的一大優勢就是可以隨時回溯到任何一個歷史版本，就像擁有一個「時間機器」一樣。

要回溯版本，你需要找到你想回到的那個版本的 **commit ID**（一串由數字和字母組成的代碼）。你可以使用 git log 指令來查看你所有的提交歷史：

Bash

git log

你會看到類似以下的訊息：

commit 96e5e8e3c54c03842c26f0c4516a2b9033a39e76 (HEAD -> main, origin/main)

Author: Your Name <you@example.com>

Date: Mon Aug 19 10:00:00 2025 +0800

Updated the UI and fixed a bug in the TTS feature

commit a1b2c3d4e5f67890abcdef1234567890abcdef12 (Initial commit)

Author: Your Name <you@example.com>

Date: Sun Aug 18 15:30:00 2025 +0800

Initial commit for the Gemini TTS web app

在這串歷史記錄中，commit 後面的那串代碼就是 **commit ID**。你可以複製前幾個字元就足夠了，例如 a1b2c3d。

找到你想回溯的 commit ID 後，有兩種主要的回溯方式：

**1. 只在本地回溯：git checkout**

這個指令會將你的工作目錄切換到某個特定的提交狀態。這是一個很好的方式，讓你可以在本地檢視舊版本，但不會影響到你的主要分支。

Bash

# 將 <commit-id> 替換成你想回溯的版本 ID

git checkout <commit-id>

如果你想回到最新的版本，只需要：

Bash

git checkout main

**2. 永久回溯並上傳：git revert**

git revert 會創建一個新的提交，將某個舊版本提交的變更「反向」套用。這種方式的好處是它不會抹除歷史記錄，是比較安全的做法。

Bash

# 創建一個新的提交，來反轉之前的變更

git revert <commit-id>

**將 index.html 改名為 index1.html**

**不建議**。雖然在本地這麼做沒有問題，但在 Git 的版本控制和 Netlify 的自動部署中，將檔案改名或複製為 index1.html 會帶來一些麻煩：

1. **Git 歷史記錄混亂**：你將有兩個檔案的歷史記錄，而不是一個單一檔案的連續歷史。
2. **部署問題**：Netlify 或任何網頁伺服器都**只會自動偵測和部署名為 index.html 的檔案**作為網站的首頁。如果你將它改名，你的網站可能會變成一個空白頁面，或是顯示錯誤。

**正確的記錄方式：使用 git commit**

git commit 的訊息就是用來記錄你做了什麼變更的。這比手動改名檔案來得更有效率、更乾淨。

你可以使用 git log 隨時查看每次提交的訊息，了解你在每個時間點做了什麼。這正是 Git 的強大之處。

**總結一下：**

* **回溯版本**：使用 git log 找到 commit ID，然後用 git checkout <commit-id> 或 git revert <commit-id>。
* **記錄版本**：不要改動檔案名稱，而是使用 **git commit -m "你的變更內容"**。這會將所有變更整齊地記錄在你的 Git 歷史中。

這樣，你就可以輕鬆地管理你的專案，而不需要手動創建多個檔案來記錄不同版本了。