

[老葉說技術-第46期] 秒懂Git：Git到底能幫你做什麼呢？ Git最簡單教程，使用Git三件套：init, add, commit

- **Git** 是一個檔案歷史追蹤系統，很強大。
- 但它可以為你做什麼呢？
 - **Git**可以像時光機一樣，帶你回到過去曾經提交過的檔案狀態。
 - 當多人對同一個文件庫做編輯時，**Git**可以有效的合併彼此的工作成果。
 - 以我為例，我平時會將程式碼上傳到**GitHub**，分享給各位粉絲，讓各位能下載並執行每一集的程式碼。
- 本期就是用最簡單的方法，讓各位體驗**Git**，首先，各位需要在電腦上安裝**Git**。
 - 請到**Git**官網：<https://git-scm.com/downloads>



[老葉說技術-第46期] 秒懂Git：Git到底能幫你做什麼呢？ 最簡單教程，使用Git三件套：init, add, commit

- 安裝完成後，那要如何使用**Git**呢？
- **Git**是以目錄為單位進行管理，並管理該目錄下的所有檔案(包括目錄)，並記錄所有的提交記錄，這個文件庫稱作**Repository**。
- 因此若要將某個目錄納入**Git**的管理，則需用終端機進入該目錄，鍵入 **git init**（代表初始化）即可。
- 初始化後，該目錄下所有新增的檔案並不會自動被**Git**追蹤，你需要用 **git add** 檔名，來將檔案納入**Git**的追蹤範圍，所謂的追蹤就是**Git**會自動偵測檔案是否被修改、刪除，若不想一個一個檔案加入，也可以用**git add --all**，來加入所有檔案(包括所有子目錄的檔案)，也可以用**git add .**，但它最多只能加入當前目錄的下一層子目錄的檔案。



[老葉說技術-第46期] 秒懂Git：Git到底能幫你做什麼呢？ 最簡單教程，使用Git三件套：init, add, commit

- 提交(commit)就是存檔：當你覺得工作到一個段落，或想先暫時存檔時，則需要學習將暫存區的檔案作一次提交，所謂的提交就是把本次的所有修改內容永久記錄起來，在終端機鍵入 `git commit -m '本次提交的說明'`，若沒有做提交，只是僅僅在暫存區，檔案仍有可能會遺失。
- 每一次的提交(commit)都是一次目錄下所有檔案的快照，而Git可以像時光機一樣，帶你回任一的快照狀態，可以任意移動，每一次的移動，該目錄都會回到那一次快照的樣子。
- 對目錄git init 後，會產生一個.git的隱藏目錄，你的所有的存檔記錄都存在那，若把.git目錄刪除，則過去所有的存檔記錄都不見了。



[老葉說技術-第46期] 秒懂Git：Git到底能幫你做什麼呢？ 最簡單教程，使用Git三件套：init, add, commit

- 開一個新目錄叫gittest, 並進入該目錄，執行git init
- 建立一個新檔案叫 a.txt, 內容打a，存檔。
- 鍵入 git status，Git會告訴你a.txt未被追蹤 (也就是未加入暫存區)
- 把 a.txt加入暫存區，鍵入 git add a.txt
- 再鍵入 git status, 發現a.txt已經被追蹤 (變綠色)
- 我們先來做一次提交，git commit -m 'add a.txt'
- 鍵入 git log

```
[jackieyeh@JackiedeiMac gittest % git log
commit 44c0dc47f2e39d893f3a34f31c4abd545f40c2f7 (HEAD -> main)
Author: Jack Yeh <realjackyeh@gmail.com>
Date:   Tue Nov 8 10:06:26 2022 +0800

    add a.txt
```



[老葉說技術-第46期] 秒懂Git：Git到底能幫你做什麼呢？ 最簡單教程，使用Git三件套：init, add, commit

- 開一個新目錄叫**fb**，並執行 `git status`，發現並無檔案需被追蹤。
- 在**fb**目錄下新增一個**b.txt**，內容為**b**，執行 `git status`，Git會告訴你**fb**目錄下有檔案未被追蹤。
- 執行 `git add --all`，再執行 `git status`，發現**fb**目錄下的**b.txt**已被追蹤。
- 我們再提交一次 `git commit -m 'add fb/b.txt'`。
- 執行 `git log`，會發現已經有二次提交記錄。
- 現在我們用Git時光機回到第一次的提交狀態，`git reset` 第一次提交的SHA1碼 `--hard`，這時你會發現目錄下只有**a.txt**。

```
jackieyeh@JackiedeiMac gittest % git reset b5b5 --hard
HEAD is now at b5b5d07 add a.txt
jackieyeh@JackiedeiMac gittest %
```



[老葉說技術-第46期] 秒懂Git：Git到底能幫你做什麼呢？ 最簡單教程，使用Git三件套：init, add, commit

- 這時，我們查看a.txt的內容，`cat a.txt`。
- 要怎麼回到有fb/b.txt的狀態呢？一樣使用Git時光機回到最後一次的提交狀態，`git reset` 最後一次提交的SHA1碼 `--hard`
- `git reset`共有三個參數可供使用`--hard/soft/mixed`，以我個人來說`hard`最常用，其它二個參數各位可以查一下是什麼意思，自己實際操作一下就可以明白。
- 預設分支叫**Main**，它是一個檔案，它會指向某一次的提交，而**HEAD**（你所在位置）則會指向分支。**HEAD → branch → Commit**
- 分支就像標籤，可以隨意增加，當有多人共同編輯一個文件庫時，就各自開各自的分支，最後用**merge** 或 **rebase**來合併彼此的內容。
- 只要提交過的記錄(檔案狀態)，就很難被刪除，不管怎麼操作分支，過去的記錄都還在，不用擔心

