

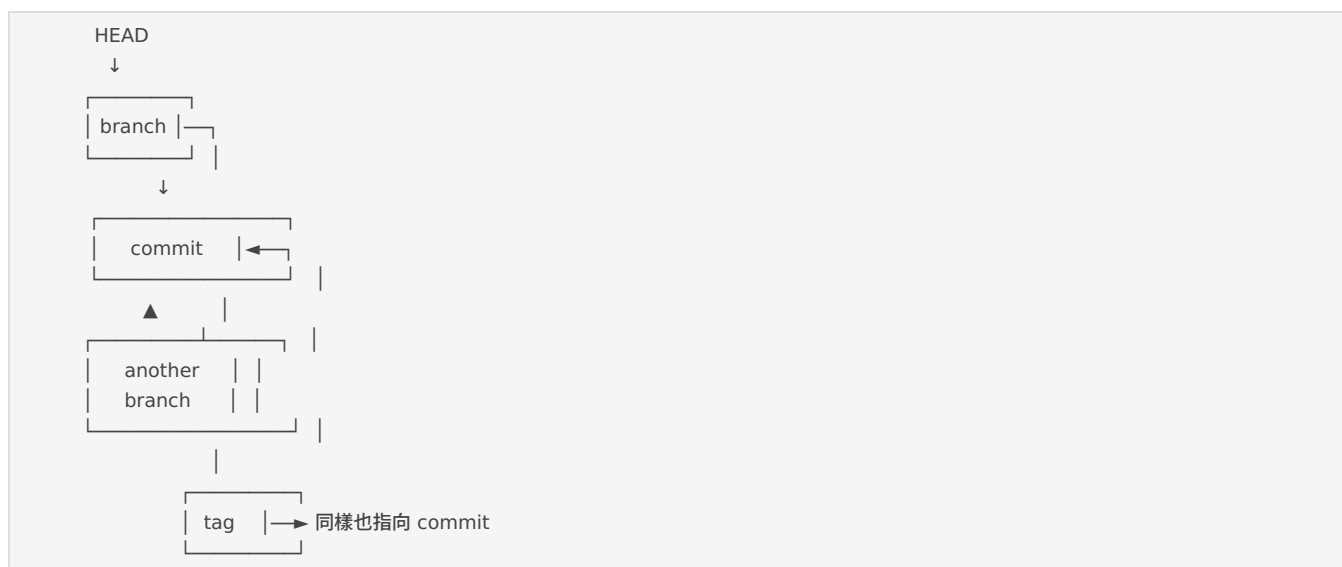
# 【Git】head, tag, commit, branch 的關係

這是一個非常關鍵的 Git 理解核心：

**HEAD**、**tag**、**commit**、**branch** 彼此的關係。

我幫你用「定義 + 關聯圖 + 對照表 + 比喻」完整整理如下：

## □ 一張圖：理解 Git 中 HEAD、分支、Tag、Commit 的關係



## □ 核心定義

名稱	定義說明
<b>commit</b>	Git 的基本單位，是一個「版本快照」，有父子連結，不會變動
<b>branch</b>	指向某個 commit 的「可移動指標」，像捷徑，可以新增、移動
<b>HEAD</b>	代表你目前工作的「游標」，通常指向一個分支（或直接指向 commit）
<b>tag</b>	指向某個 commit 的「不可變名稱」（常用來標記版本，如 <code>v1.0.0</code> ）

## □ 它們的關係總結

關係	說明
<code>branch</code> → <code>commit</code>	分支就是指向某個 commit 的「指標」
<code>HEAD</code> → <code>branch</code> or <code>commit</code>	HEAD 是你「目前所在的位置」，可以指向分支或直接指向 commit (detached HEAD)
<code>tag</code> → <code>commit</code>	Tag 永遠指向某個 commit，是不可變的
commit 之間	commit 是鏈狀連結（每個有一個 parent commit）

# □ 類比說明（像檔案系統）

Git 概念	類比	說明
commit	檔案版本	是實際的內容快照（不可變）
branch	資料夾捷徑	可以移動、可以新建、可以切換
HEAD	滑鼠游標	你目前的編輯焦點（在哪個捷徑/檔案上）
tag	唯一標籤貼紙	永久貼在某個版本上，表示這是一個重要的版本點

# □ 補充：變動性比較

元件	是否可變動	常見操作
commit	□ 不可變	一旦建立就不會改變
branch	□ 可變動	可以移動、刪除、重新建立
tag	□ 可刪除（內容不可變）	可以刪除再重建，但本身是不可移動的指標
HEAD	□ 一直在動	跟著你切換分支、commit 而變動

# □ 常見操作對應關係

操作說明	HEAD 變動	branch 變動	commit 變動	tag 變動
<code>git switch dev</code>	□	□	□	□
<code>git commit</code>	□	□（會跟著 HEAD 一起動）	□（新增）	□
<code>git reset --hard HEAD^</code>	□	□	□（可能丟失）	□
<code>git tag v1.0</code>	□	□	□	□
<code>git checkout &lt;commit&gt;</code>	□（分離）	□	□	□
<code>git switch -c new-feature</code>	□	□（新建）	□	□

# □ 總結一句話：

- “
- **commit** 是內容
  - **branch/tag** 是「名字」，用來指向某個 commit
  - **HEAD** 是「你目前在哪裡」，通常跟著分支動，也可以離開分支進入 detached 狀態

🕒 修訂版本 #1

★ 由 treeman 建立於 8 🕒 2025 09:49:21

🔪 由 treeman 更新於 14 🕒 2025 10:13:29