

# 【Git】 git revert <commit> 和 git checkout <commit> 差異

`git revert <commit>` 和 `git checkout <commit>` 雖然都能「回到某個 commit」，但本質上用途完全不同。以下為詳細說明與比較。

## 一、 `git revert <commit>`

### 功能說明：

- 用來\*\*「反向操作」某個已提交的 commit\*\*。
- Git 會建立一個新的 commit，內容是將指定的 commit 所做的變更還原回去。
- 原本的 commit 仍然會保留在歷史中。

### 特性：

- 安全，適合多人協作情境。
- 保留歷史紀錄。
- 可選擇 revert 多個 commit（可搭配 `-n` 進行批次處理）。

### 使用情境：

- 已經 push 到遠端，但發現某個 commit 有誤，需要「撤銷」那次變更，但不想刪除歷史。
- 緊急修復 bug，並希望讓其他人看見這次的「撤銷行為」。

### 範例：

```
git revert a1b2c3d
```

會產生一個新的 commit，把 `a1b2c3d` 的內容還原。

## 二、 `git checkout <commit>`（或 `git switch --detach <commit>`）

### 功能說明：

- 將當前工作目錄切換到某個過去的 commit（進入 **detached HEAD** 狀態）。
- 不會建立新的 commit，純粹只是瀏覽某個歷史狀態。
- 如果在此狀態下修改檔案並 commit，會產生「無分支」的 commit（容易遺失）。

### 特性：

- 不會變動歷史，也不會建立新 commit。
- 適合測試、比較、build 老版本。

### 使用情境：

- 想檢查過去某次版本的內容、debug、測試、build。
- 需要比較不同版本的行為。

## 範例：

```
git checkout a1b2c3d
# 或推薦用法 (Git 2.23+)
git switch --detach a1b2c3d
```

## 三、兩者比較表

指令	動作類型	是否建立新 commit	是否修改歷史	是否可安全推送遠端	常見用途
<code>git revert &lt;commit&gt;</code>	建立反向 commit	<input type="checkbox"/> 是	<input type="checkbox"/> 否	<input type="checkbox"/> 是	撤銷已推送的錯誤 commit
<code>git checkout &lt;commit&gt;</code>	切換版本	<input type="checkbox"/> 否	<input type="checkbox"/> 否	<input checked="" type="checkbox"/> 不適合推送 detached commit	查看歷史、測試、debug

## 四、補充說明：若想基於某個舊 commit 建立新分支

```
git checkout -b new-branch a1b2c3d
```

這樣可以保留該版本的狀態，並在其上繼續開發，而不會讓 HEAD 處於 detached 狀態。

如果你有實際情境（例如：某個 commit 上錯 production，或想回復某功能），可以提供情況，我能幫你判斷該用 revert 還是 checkout 或其他方案。

🕒 修訂版本 #1

★ 由 treeman 建立於 14 🕒 2025 10:55:52

🔧 由 treeman 更新於 14 🕒 2025 10:56:18