

系統分析

關於系統分析

- 系統分析的說明與目標
 - 相關名詞說明
 - 分析練習
-

系統分析的說明與目標

- 系統開發生命週期 (System Development Life Cycle , SDLC)
 - 系統規劃
 - 系統分析 (System Analysis, SA)
 - 系統設計 (System Design, SD)
 - 系統建置
 - 系統維護
-

系統分析相關名詞說明 – 1/2

➤ 什麼是「系統」？

- 一組能夠產生特定結果的相關元件

➤ 什麼是「資料」？

- 未經過處理的文字數字等，呈現基本事實

➤ 什麼是「資訊」？

- 透過相同準則整理資料，使其對使用者有用或有意義
-

系統分析相關名詞說明 – 2/2

➤ 什麼是「規劃」？

- 定義目標、決定情境、探討可行性
- 建立達成目標之整體策略(計畫)

➤ 什麼是「分析」？

- 把複雜的事物或內容逐漸拆分、檢查和評估的過程

➤ 什麼是「設計」？

- 設想和計劃，一種有目的的創作行為
-

分析練習

➤ 稍後會有一些問答題，請依照個人平時的習慣，
底下清單擇一準備即可

- 一張可以書寫的紙、一枝筆
 - 開啟電腦中的任何一套編輯軟體
 - 開啟手機中的任何一套編輯軟體
-

系統規劃

- 組織專案成員/參與專案
 - 定義系統目標
-

組織專案成員/參與專案

➤ 組織專案成員 ➔ 大專題分組

- a. 意願、信得過、技術互補、談話合得來
- b. 不太認識、不太熟悉、沒有相處過

➤ 參與專案 ➔ 結訓後就業

- a. 意願、信得過、技術互補、談話合得來
 - b. 不太認識、不太熟悉、沒有相處過
-

定義系統目標

1. 了解問題或機會
 2. 定義專案範圍(限制)
 3. 發現事實(搜集資料)
 4. 評估可行性
 5. 估算時間及成本
 6. 向主管展示結果
-

定義系統目標→了解問題或機會

➤ 大專

- 現在市面上沒有 _ _ _ 類型的系統
- 我曾經使用過 _ _ _ 系統但是不好用
- 我們組員都的興趣都是 _ _ _

➤ 就業

- 為什麼系統需要增加/修改 _ _ _ 功能
-

定義系統目標→定義專案範圍

- 在有限的時間內，做有限的事
 - 界定核心功能
 - 條列有可能影響結果的因素
 - 不違反法律條款
-

定義系統目標→發現事實

- 進行採訪（單一、團體）
 - 查閱現有文件
 - 觀察現有作業情況
 - 問卷調查
 - 不知道該怎麼調查時，請讓這些關鍵字輔助你
「who, what, where, when, why, how（5W1H）」
-

定義系統目標→其他

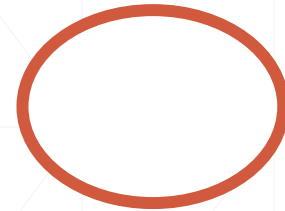
- 評估可行性
 - 避免過度不切實際
 - 估算時間及成本
 - 需要的技術目前是否已經具備？
 - 向主管展示結果（職場）
 - 需要定期向直屬主管回報工作進度
-

分析工具 UML 基本介紹

- 使用案例圖說明與練習
 - 活動圖說明與練習
 - 循序圖說明與練習
-

使用案例圖 - Use Case Diagram

- 描述系統功能
 - 盡量以動作 + 名詞描述功能
- 界定系統邊界
 - 那些功能應該包含在系統中
- 找出參與角色或參與系統
 - 誰執行功能，或者誰進行回應



使用案例圖 - 描述系統功能

* 使用案例圖並不包含編號，下列為了討論方便而使用

1 插入卡片

4 辨識人臉

7 查詢

6 存款

2 提款

3 優惠券

5 指紋辨識

8 明細列印

使用案例圖 - 界定系統邊界

* 使用案例圖並不包含編號，下列為了討論方便而使用

1 插入卡片

4 辨識人臉

7 查詢

6 存款

2 提款

3 優惠券

5 指紋辨識

8 明細列印

使用案例圖 - 參與人員或參與系統

* 使用案例圖並不包含編號，下列為了討論方便而使用

1 插入卡片

4 辨識人臉

7 查詢

6 存款

2 提款

3 優惠券

5 指紋辨識

8 明細列印

使用案例圖 - ATM提款



活動圖 - Activity Diagram - 1/3

➤ 標示活動起點和終點

- 原則上有一個起點、一個終點



起點



終點

➤ 決策

- 判斷或者控制活動走向
- 描述說明條件或限制



活動圖 - Activity Diagram - 2/3

➤ 描述活動或者行為

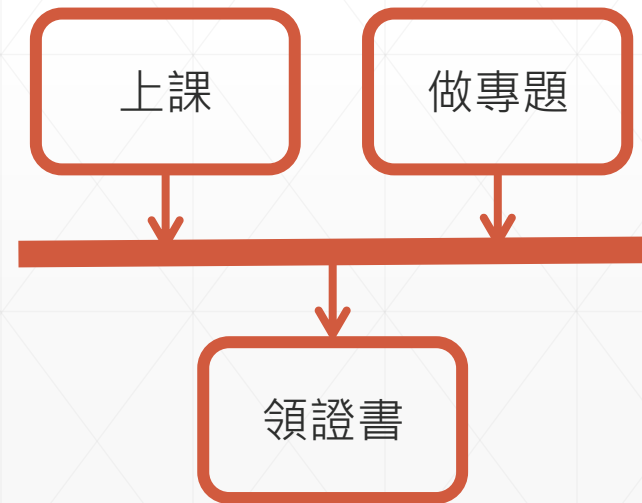
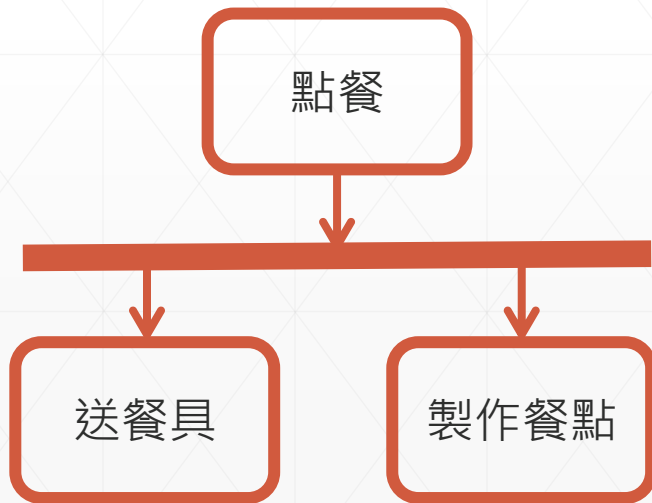
- 描述盡量具體、單一、不可再分割



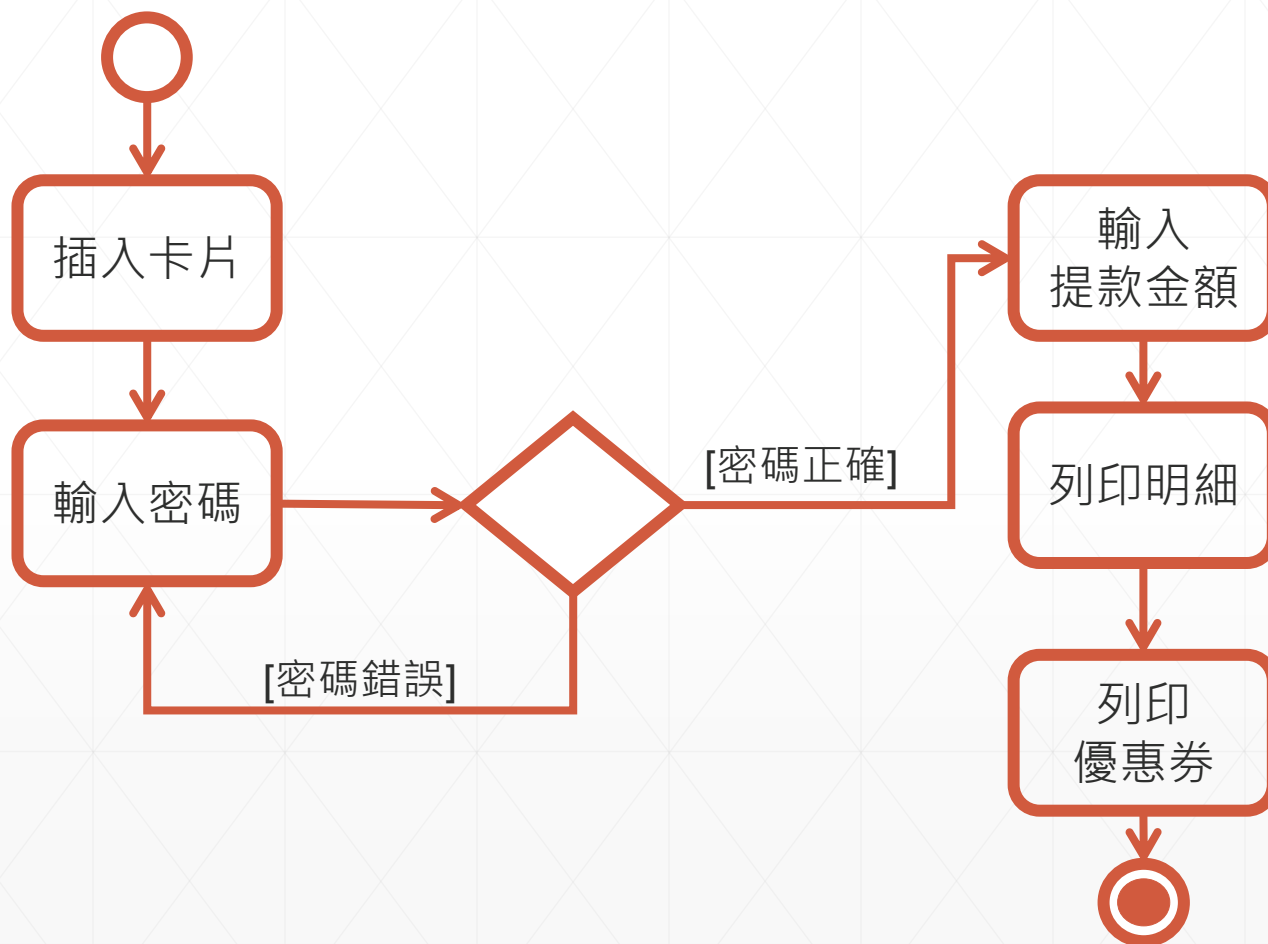
活動圖 - Activity Diagram - 3/3

➤ 分支與合併

- 分支：在活動結束後，有多項活動可分散進行
- 合併：需要等待多項活動結束後才可進行的活動



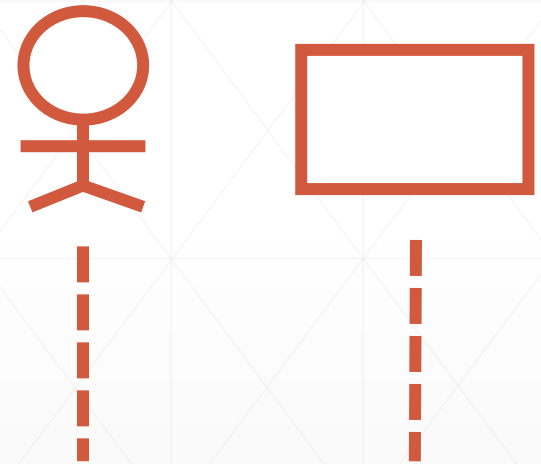
活動圖 - ATM提款



循序圖 - Sequence Diagram - 1/2

➤ 角色或物件

- 角色：若循序圖是根據特定使用案例圖而來
- 物件：物件可能是某項服務、某個系統、某個表單等等



➤ 時間軸：直條虛線

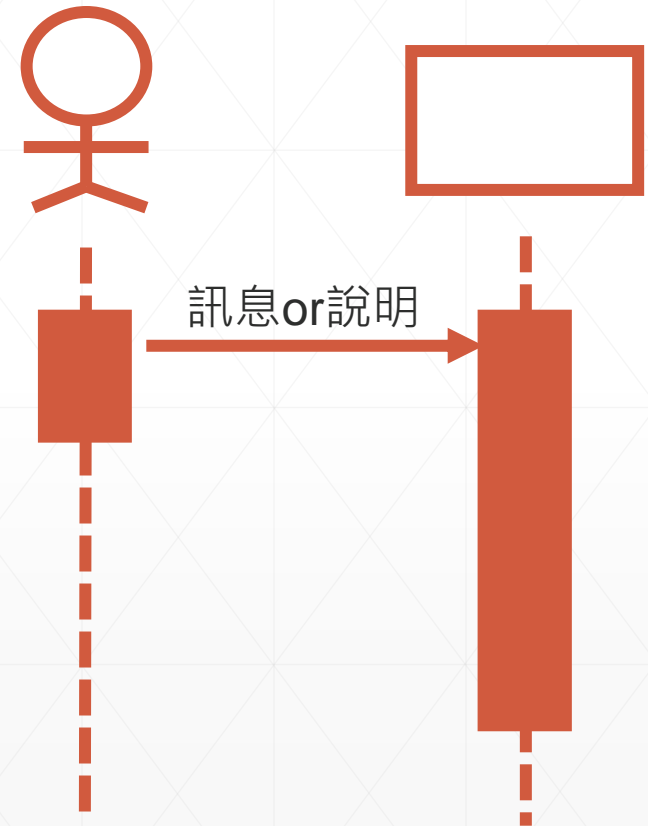
循序圖 - Sequence Diagram - 2/2

➤ 活動

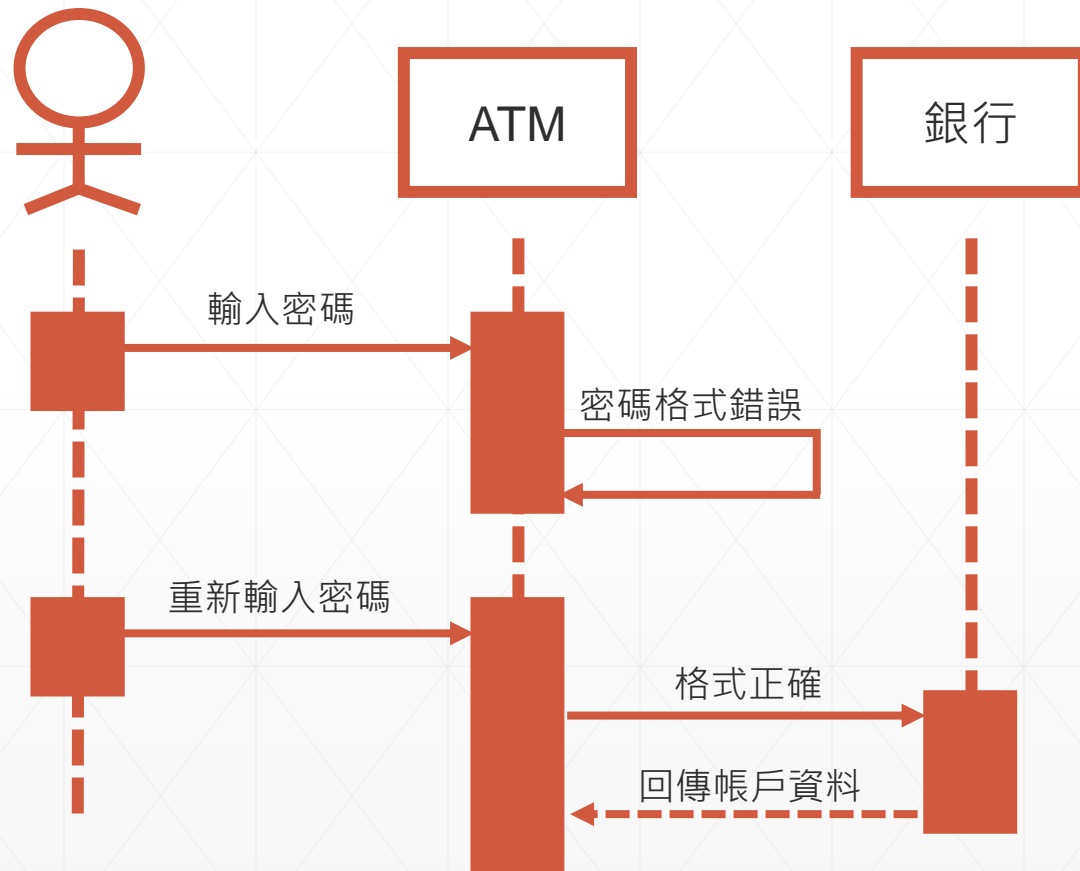
- 一項操作所需要的時間
- 會覆蓋在時間軸虛線之上

➤ 訊息箭頭

- 描述角色和物件，或者物件和物件之間的訊息傳遞



循序圖 - ATM提款



實體關係圖

- 實體關係圖的圖形符號
 - 案例討論
 - 將分析後的圖形轉換為資料表
-

實體關係圖 – Entity Relationship Model

➤ 方形：描述實體(Entity)



➤ 橢圓形：描述屬性(Attribute)



➤ 菱形：描述關係(Relationship)



菱形：描述關係

➤ 一對一的關係



➤ 一對多的關係



➤ 多對多的關係



案例討論

➤ 需求收集時，從文字紀錄轉換成實體關係模型的過程，通常可以這樣對應：

- 名詞：對應為「實體」➔方形
 - 動詞：對應為「關係」➔菱形
 - 形容詞：對應為「屬性」➔橢圓形
-

案例討論：遊客購買門票

➤ 找出文字描述中的名詞、動詞、形容詞：

遊客(名詞)購買(動詞) 門票(名詞)

遊客

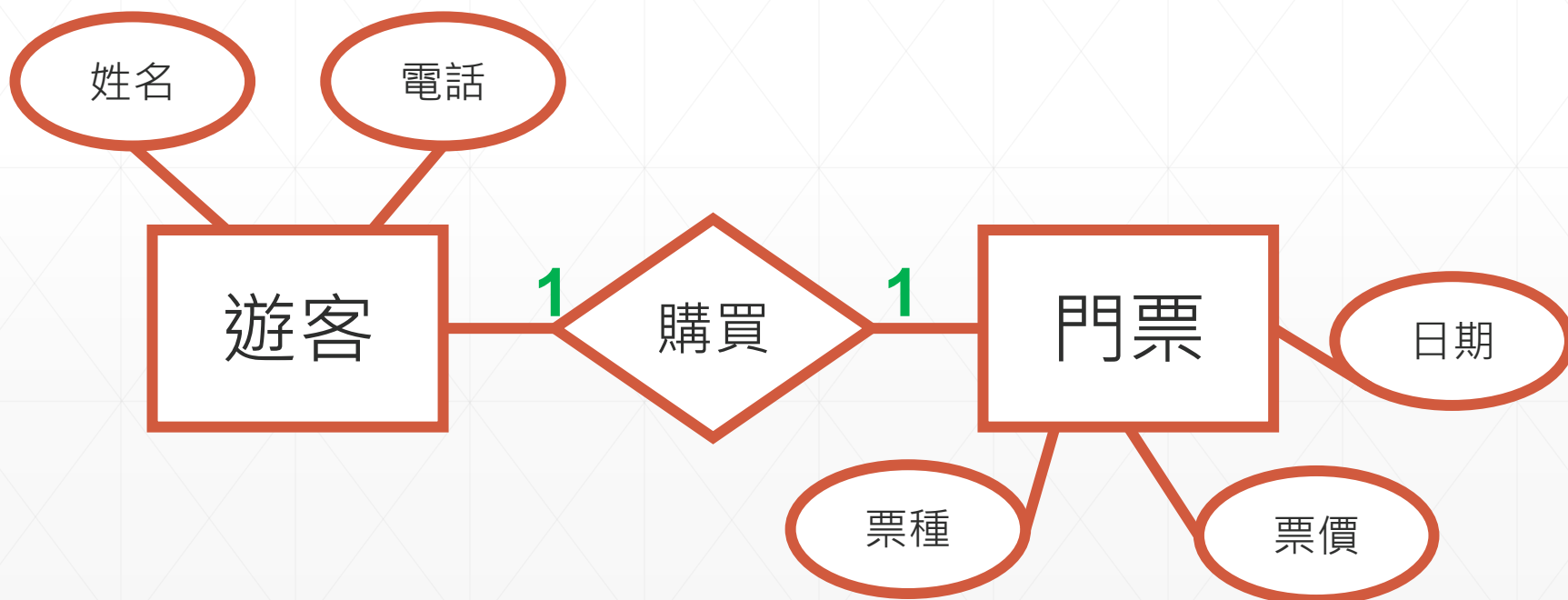
購買

門票

案例討論：遊客購買門票

➤ 建立關係並且添加屬性

* 隨著需求分析或結合日常生活經驗，可能會繪製出不同圖形



將分析後的圖形轉換為資料表

➤ 一對一的關係

- 一對一的情況下，可以考慮將資料表合併

<u>姓名</u>	<u>電話</u>	<u>日期</u>	票種	票價
言午許	0912345678	2025/12/31	成人票	100
金刀劉	0968222888	2025/12/31	孩童票	50
...

程式碼版本控制

- Git & GitHub
 - 環境檢查、設定
 - GitHub多人專案操作
 - Git 指令說明
 - 解決程式碼版本衝突
-

Git & GitHub

➤ Git 是軟體(指令)、GitHub是平台(服務)

- SQL & MySQL
 - SQL & SQL Server
 - 七分SO & 七分褲
 - 大象 & 大象杯
-

環境檢查、設定

➤ 【檢查電腦中的使用者認證】

- 控制台 > 使用者帳戶 > 認證管理員 >

windows認證 > 一般認證 > (有git|github相關字樣)

- 教室的電腦中，如果有先前使用者的設定，記得要先移除

➤ 【下指令前，請務必確認當前目錄是否正確】

環境檢查、設定

➤ git version

- 確認自己的電腦是否已經安裝了git軟體
- 若無，則下載安裝
- 若當前版本為 2.16.1 以上，可以使用

`git update-git-for-windows` 進行更新

環境檢查、設定

➤ git config --list

- 觀察 user.name 、 user.email 是否有值(存在)
- 輸入Q可以離開
- 若顯示的資料不是自己的，可透過下列指令刪除

```
git config --global --unset user.name
```

```
git config --global --unset user.email
```

環境檢查、設定

- `git config --global user.name "你的帳號"`
 - `git config --global user.email "你的信箱"`
 - 設定電腦的環境配置
 - 建議與GitHub註冊時填寫的資料一致
-

GitHub多人專案操作

➤ 組長

- 在GitHub建立專案Repository
- Repository > Settings > Manage access > Invite a collaborator 加入組員 GitHub 帳號

➤ 組員

- 請收信，同意加入組長建立的專案
-

Git 指令說明

➤ `git clone` GitHub專案url

- 複製遠端GitHub數據庫到本地端工作目錄
 - 這個指令通常在一開始使用一次
 - 後續取得遠端GitHub數據庫會採用 `git pull` 指令
-

Git 指令說明

➤ git status

- 了解工作目錄現況



Git 指令說明

➤ `git add *.副檔名` || `git add .`

- 將指定檔案||全部檔案放到「暫存區」，git開始列管 (追蹤)這些檔案

➤ `git commit -m "註解訊息"`

- 將列管中的檔案放到「本地數據庫」
-

Git 指令說明

➤ git pull

- 取得遠端GitHub的檔案

➤ git push

- 可能會跳出輸入GitHub帳密的視窗
 - 將本地端程式碼提交到GitHub「遠端數據庫」
-

Git 指令說明

➤ git branch

- 查看當前工作目錄有幾個分支

➤ git branch 分支名稱

- 建立工作分支

➤ git checkout 分支名稱

- 切換到指定分支
-

Git 指令說明

➤ git diff 分支名稱

- 查看分支之間的差異

➤ git merge 分支名稱

- 合併分支
 - 合併時，若「遠端數據庫」和「本地數據庫」有落差，則需要手動處理差異處
-

Git 指令說明

➤ git log

- 查看檔案修改人員、版本更新歷史等相關訊息

➤ git log --oneline

- 將每一筆commit顯示成單獨一行
-

Git 指令說明

➤ `git reset --soft SHA || HEAD~N`

- 保留當前修改，將「本地數據庫」移至「暫存區」

➤ `git reset --mixed SHA || HEAD~N`

- 保留當前修改，將「本地數據庫」移至「工作目錄」

➤ `git reset --hard SHA || HEAD~N`

- 丟棄當前所有修改，取得上一個commit結果
-