GitHub 多人協作全流程筆記

一、初始化專案並上傳 GitHub（第一人）

```
# 1. 建立並進入專案目錄
mkdir museum-project
cd museum-project

# 2. 初始化 Git 倉庫
git init

# 3. 建立 .gitignore 文件（建議）
touch .gitignore
# 編輯 .gitignore 加入不需要版本控制的文件
# 例如：node_modules/, .env, .DS_Store 等

# 4. 提交初始代碼
git add .
git commit -m "Initial commit: 專案初始化"
```

建立 GitHub Repo 並關聯

1. 在 GitHub 建立新倉庫（不要勾選 Initialize README）
2. 執行以下命令：

```
git remote add origin https://github.com/使用者名稱/專案名稱.git
git branch -M main
git push -u origin main
```

二、建立開發分支（第一人）

```
# 建立並切換到 dev 分支
git checkout -b dev
git push -u origin dev
```

三、團隊成員加入專案（大家從這裡開始）

1. 克隆專案

```
# 克隆專案到本地
git clone https://github.com/使用者名稱/專案名稱.git
cd 專案名稱
```

```
# 切換到開發分支
git checkout dev
git pull origin dev
```

2. 建立功能分支

```
# 確保在 dev 分支上
git checkout dev
git pull origin dev

# 建立功能分支（建議使用 feature/ 或 fix/ 前綴）
git checkout -b feature/功能名稱
# 或
git checkout -b fix/問題修復
```

3. 開發並提交

```
# 開發完成後提交
git add .
git commit -m "feat: 新增功能描述"
git push -u origin feature/功能名稱
```

四、合併到開發分支（大家做到這步驟）

```
# 1. 切換到 dev 分支並更新
git checkout dev
git pull origin dev

# 2. 合併功能分支
git merge --no-ff feature/功能名稱 -m "feat: 合併功能描述"

# 3. 推送到遠端
git push origin dev
```

五、發布版本（第一人去更新就好!!其他人不要動到！！）

```
# 1. 切換到主分支
git checkout main
git pull origin main

# 2. 合併開發分支
git merge --no-ff dev -m "release: 發布版本 v1.0.0"
```

```
# 3. 推送到遠端
git push origin main
```

六、版本標籤

```
# 建立標籤
git tag -a v1.0.0 -m "版本 1.0.0 發布"
git push origin v1.0.0
```

分支命名規範

- `main`: 主分支，用於生產環境
- `dev`: 開發分支，用於整合功能
- `feature/*`: 新功能分支
- `fix/*`: 問題修復分支
- `hotfix/*`: 緊急修復分支

常用 Git 指令

```
# 分支操作
git branch -a                    # 查看所有分支
git branch -v                    # 查看分支詳細信息
git checkout -b dev origin/dev   # 拉取遠端 dev 分支

# 查看提交歷史
git log --oneline --graph --all  # 圖形化顯示提交歷史
git log --stat                   # 顯示文件變更統計

# 分支管理
git branch -d feature/xxx        # 刪除本地分支
git push origin --delete feature/xxx  # 刪除遠端分支

# 暫存操作
git stash                        # 暫存當前修改
git stash pop                    # 恢復暫存的修改

# 重置操作
git reset --soft HEAD^           # 撤銷最近一次提交
git reset --hard HEAD^           # 撤銷最近一次提交並丟棄修改
```

最佳實踐建議

1. 經常進行小規模提交，保持提交信息清晰
2. 在合併前先更新本地分支

3. 使用有意義的分支名稱
4. 定期清理已合併的分支
5. 重要操作前先備份
6. 使用 .gitignore 管理不需要版本控制的文件