這是個很常見的需求

如果使用gitlab or github or 其他工具，確實會有很多功能可以幫你合併！但是出現衝突（Conflict）時，多半還是要人為去排查問題，此篇是針對我認為最單純簡單的過程來處理。

因為選字關係，請不要直接複製使用指令的部分，會出錯，手打練習不會花多少時間。

工具
（你可以有自己的一套，但怕麻煩易出錯、新手，可以直接用我的方法）

軟體	目的
terminate	使用git 指令，可以使用VC Code 的 Terminate 視窗也可以。
file editer	使用IDE也可以，像是VC Code，會主動標出有衝突的檔案，有更動的檔案
耐心	不著急，這很容易... 心急弄破碗

作法 (以下方式不包含刪除分支，有需要此動作可以靠自行查詢或版控工具)

$ git clone REPO. -b Branch-A
當你是要把 Branch-B 合併到 Branch-A，如果已經有抓下Repo. 跳過此步驟 此處經常會有誤解，我們習慣性思維是“把Branch-B 推進 Branch-A 進行合併”(x) 而實際上在動作是 “Branch-A 把 Branch-B 拉近自己的內容紀錄中，達到合併的效果” (v)

$ cd REPO./ $ git checkout Branch-A
移動到分支 Branch-A

$ git fetch origin Branch-A
把Branch-A 的內容紀錄同步到本機中，採用的遠端機器是記錄在origin 的。 基本上origin 是預設的位置，除非你了解如何改動和其目的，不建議更改。

$ git checkout Branch-B $ git fetch origin Branch-B
移動到分支 Branch-B, 把Branch-B 的內容紀錄同步到本機中

$ git checkout Branch-A $ git merge Branch-B ```bash Auto-merging xxx CONFLICT (content): Merge conflict in xxxx Automatic merge failed; fix conflicts and then commit the result.

```

跳回Branch-A 開始拉動 Branch-B 內容進入 Branch-A 做合併
如果被發現有兩邊皆修改的檔案，就會跳出 Auto-merging xxx
如果發現有衝突會記錄下來，並且告知... 那接下來就需要改變。

請到標注的檔案中去改有 (以下是常見情況)
<<<< Head (current Branch-A)
hello world~
==========
good morning.
>>>> Branch-B

從中把你要的確認下來，紅字的部分選好後，藍字部分全部刪除
儲存好檔案，完成衝突解決。

```

```
$ git add . –all
$ git commit -m “merge Branch-B into Branch-A: sync”
// or, “merge Branch-A with Branch-B: sync”
$ git push -u origin Branch-A
```

更新修改表，然後附上description，建議一定要有 Merge 的字樣，
後面可以補上是為了同步（sync）、覆蓋(cover)、退版/反悔（revert）...
最後上傳更新，完成合併和解衝突的動作。

範例 merge master into dev（dev 同步 master 的新資料）

```
% git clone REPO -b dev
% cd repo
% git fetch origin dev
% git checkout master
% git fetch origin master
% git checkout dev
% git merge master

～fix conflict ～
% git add . --all
% git commit -m "merge master into dev: sync"
% git push -u origin dev
```

之後去remote git 圖像化分支瀏覽