# Git Commit Message 這樣寫會更好，替專案引入規範與範例

https://wadehuanglearning.blogspot.com/2019/05/commit-commit-commit-why-what-commit.html

Commit Message 跟寫程式註解還蠻像的，
最好可以寫下「為什麼」你要作這樣的異動，
而不是單單只記錄下你做了「什麼」異動。

> Commit Message 最好兼俱 Why 及 What，讓日後進行維護人員更快進入狀況。

## Commit Message 這樣寫會更好：

- 做 issue 的時候，不應該一次 Commit 所有異動！應該獨立 Commit 每個不同意義的異動，這樣 commit 訊息才會跟異動的程式碼有關聯。
- 每次 Commit 都是針對異動的檔案做說明：Why & What。這樣的 Commit Message 能讓日後的維護人員更快進入狀況
- 每次 Commit 都加上 issue 編號，方便追蹤相關的程式異動原因。

若 Commit Message 寫得妥當，在閱讀追蹤程式碼的意圖會相當容易。如果只把 Git 當作版本控制，隨意撰寫 Commit Message 就太可惜了！

> 不能只把 Git 當作程式碼的 FTP，要把 Git 當作歷史查閱的工具才拿發揮 Git 的功能。

## 好與不好的真實案例

用一個小插曲證實 Commit 訊息的重要性
上面 PPT 是我在工作中遇到的兩個案例，範例中包含「好的 Commit Message」與「不良的 Commit Message」。
在範例中可見：

- 良好的 Commit Message: **如何在「一年後」讓維護人員進入狀況**
- 不良的 Commit Message: **如何在「一個月內」讓維護人員找不出程式異動的原因。**

# Commit Message 之規範

以下為這套訊息規範的展示與說明:
Commit Message 規範範例:

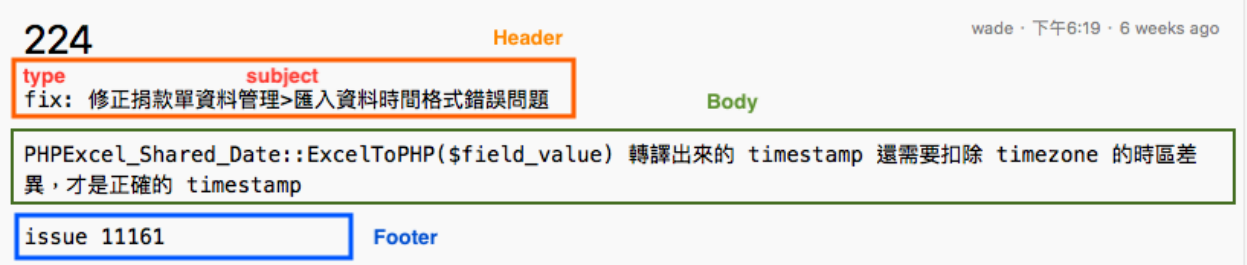
```
224                                                    wade · 下午6:19 · 6 weeks ago

fix: 修正捐款單資料管理>匯入資料時間格式錯誤問題

PHPExcel_Shared_Date::ExcelToPHP($field_value) 轉譯出來的 timestamp 還需要扣除 timezone 的時區差
異，才是正確的 timestamp

issue 11161
```

Commit Message 規範範例解析:




# Commit Message 規範組成:

```
Header: ():
 - type: 代表 commit 的類別：feat, fix, docs, style, refactor, test, chore，必要欄位。
 - scope 代表 commit 影響的範圍，例如資料庫、控制層、模板層等等，視專案不同而不同，為可選欄位。
 - subject 代表此 commit 的簡短描述，不要超過 50 個字元，結尾不要加句號，為必要欄位。

Body: 72-character wrapped. This should answer:
 * Body 部份是對本次 Commit 的詳細描述，可以分成多行，每一行不要超過 72 個字元。
 * 說明程式碼變動的項目與原因，還有與先前行為的對比。

Footer:
 - 填寫任務編號（如果有的話）.
 - BREAKING CHANGE（可忽略），記錄不兼容的變動，
   以 BREAKING CHANGE: 開頭，後面是對變動的描述、以及變動原因和遷移方法。
```

type: subject 是簡述不要超過 50 個字元

### type 只允許使用以下類別：

- feat: 新增/修改功能 (feature)。
- fix: 修補 bug (bug fix)。
- docs: 文件 (documentation)。
- style: 格式 (不影響程式碼運行的變動 white-space, formatting, missing semi colons, etc)。
- refactor: 重構 (既不是新增功能，也不是修補 bug 的程式碼變動)。
- perf: 改善效能 (A code change that improves performance)。
- test: 增加測試 (when adding missing tests)。
- chore: 建構程序或輔助工具的變動 (maintain)。
- revert: 撤銷回覆先前的 commit 例如：revert: type(scope): subject (回覆版本：xxxx)。

Type 是用來告訴進行 Code Review 的人應該以什麼態度來檢視 Commit 內容。
例如：

- 看到 Type 為 fix，進行 Code Review 的人就可以用「觀察 Commit 如何解決錯誤」的角度來閱讀程式碼。
- 若是 refactor，則可以放輕鬆閱讀程式碼如何被重構，因為重構的本質是不會影響既有的功能。

利用不同的 Type 來決定進行 Code Review 檢視的角度，可以提升 Code Review 的速度。因此開發團隊應該要對這些 Type 的使用時機有一致的認同。

## Commit 訊息範例

### 範例 fix：

```
fix: 自訂表單新增/編輯頁面，修正離開頁面提醒邏輯

問題：
1. 原程式碼進入新增頁面後，沒做任何動作之下，離開頁面會跳提醒
2. 原程式碼從新增/編輯頁面回到上一頁後（表單列表頁面），離開頁面會跳提醒
```

```
原因：
1. 新增頁面時，頁面自動建立空白題組會調用 sort_item，造成初始化 unload 事件處理器。
2. 回到上一頁後，就不需要監聽 unload 事件，應該把 unload 事件取消。

調整項目：
1. 初始化 unload 事件處理器：排除新增表單時，頁面自動建立空白題組調用 sort_item 的情境
2. 回到上一頁後，復原表單被異動狀態且清除 unload 事件處理器

issue #1335
```

```
fix: 意見反應，信件看不到圖片問題

問題：
1. 客戶反應：意見反應的信件都看不到圖片。

原因：
1. 目前程式碼都會要求先登入後才可查看使用者上傳的檔案，
   造成在信件上會看不見圖片的問題。

調整項目：
1. File.php，經討論後，開放讓意見反應頁面上傳的檔案，不用登入就可以查看/下載。

issue #1229
```

## 範例 feat:

```
feat: message 信件通知功能

因應新需求做調整：
 通知和 message 都要寄發每日信件，
 通知和 message 都用放在同一封信裡面就好，
 不然信件太多可能也不會有人想去看。

調整項目：
1. mail_template.php，新增 message 區塊。
2. Send_today_notify_mail.php，新增 取得每日 Message 邏輯。
3. Message_model_api.php，新增 $where 參數，以便取得每日訊息。
4. Message_api.php、Message_group_user_model_api.php，新增 **取得訊息使用者** 邏輯，以便撈取每日訊息。

issue #863
```

```
feat: 表單統計，多顯示計畫名稱欄位

因應需求做調整：
```

```
1. 列表資訊多加「計畫名稱」欄位，以利後續匯出資料處理。

調整項目：
1. Assessment_form.php，匯出表單統計時，新增訓練計畫名欄位。
2. customize.php，表單統計查詢時，多顯示訓練計畫名欄位。
3. Complex_assessment_form_api.php、Complex_assessment_form_model_api.php：
  - 取得表單統計資料時，多取得計畫名稱。

issue #1200
```

## 範例 chore:

```
chore: 更新 testing 環境

更新 ci-phpunit-test 套件 0.16 => 0.17

for Request GET 帶參數功能。
```

```
chore:調整單元測試環境

調整項目：
1. MX/Modules
將客製化 Testing 的邏輯移除，否則在測試環境中無法正確存取檔案。
2. 加入 tests/unit 與 tests/integration 目錄，並將測試檔案移至合宜的位置。
3. AdminTestCase.php，繼承 TestCase，實作登入邏輯、setUp 與 tearDown，供其他測試案例繼承使用。
4. Bootstrap.php，引入 AdminTestCase.php 共測試案例繼承用。
5. Login.php，因測試案例中不能有 header 的設定，更動系統登入邏輯，在測試環境中改用 redirect 轉址。
6. phpunit.xml，取消嚴謹宣告覆蓋模式，避免造成測試不通過（若需知道你的測試案例覆蓋了哪些類別或邏輯，可自行打開）。

## 備註： unit 與 integration 目錄
分別為「單元測試目錄」與「整合測試目錄」，單元測試目錄負責測試 Api 與 Model，整合測試目錄則負責測試 Controller。

issue #709
```

## 範例 style:

```
style: message 頁面，對 Component 做 Beautifier

經 IE 瀏覽器測試後發現 Component 裡面仍然夾帶 ES6 語法，
但是目前 Component 的程式碼都被壓縮成一行，
為了日後修改程式方便，故先對所有被壓縮的程式碼做 Beautifier

調整項目：
```

```
1. 針對所有被壓縮的程式碼做 Beautifier
2. 移除被註解的程式碼，原本被註解的程式碼應該是壓縮前的程式碼，但是經測試後發現這些被註解的程式碼都是舊 Code，故移除。

issue #1219
issue #1028
```

```
style: 統一換行符號 CRLF to LF

統一換行符號
```

```
style: 調整 HTML 縮排

issue #964
```

## 範例 refactor:

```
refactor: 重構取得「簽核流程種類名稱」邏輯

原程式碼取得流程名稱的邏輯散落在多個檔案，
為了讓未來新增/修改種類名稱時，不必到多個檔案找查程式，
現在統一透過 Process::get_type_name($process_type) 方法，
取得流程種類名稱。

調整項目：
1. Process.php，新增 get_type_name() 方法，供取得流程名稱稱用。
2. workflow_type_name.php，此 View 檔案只是為了取得流程名稱，現在以 Process:: get_type_name() 取代，故刪除。
3. Workflow_api.php，get_process_name() 方法是為了取得流程名稱，現在以 Process:: get_type_name() 取代，故刪除。
4. 其他檔案：改用 Process:: get_type_name()  取得流程名稱。

issue #1253
```

```
refactor: 表單統計，語意化調整

匯出表單時的表單答題資料，
應該是以表單 $assessment_result  為基準做統計，
故更改變數名稱 $user => $assessment_result

issue #1200
```

```
refactor: 每日通知信件，重構程式結構

考量將來可能會需要寄送多種資訊給使用者，
故重構程式結構，讓未來擴充功能時比較方便。

調整內容：
1. Send_today_notify_mail：
  - 把取得「系統通知」邏輯搬移至 System_notify_handler.php
  - 把取得「站內訊息」邏輯搬移至 Message_handler.php
  - 引入 Pipeline，把取得各種系統資訊的邏輯注入進 Pipeline。
  - 透過 Pipeline 取得每日通知信件內容，並建立信件 HTML
2. Daily_email 介面：
  - 定義 每日信件處理器 Xxx_handler 的方法
  - 之後要擴充新的功能，必須按照 Daily_email 介面的定義，實作方法。
3. message.php、system_notify.php：
  - 將「系統通知」與「站內訊息」的 Email 頁面獨立出來。

issue #1308
```

## 範例 perf:

```
perf: 評核表單列表，優化取得受評者速度

原本取得受評者的邏輯會造成載入頁面緩慢（開發機約 52 秒），故做優化。

調整方式：
原程式碼每個表單迴圈進入 DB 取得受評者資料。
改成
進 DB 一次撈取全部受評者資料，再回到 PHP 分配資料。

結果：
開發機載入頁面時間  52 秒 => 5秒

issue #1272
```

## 範例 docs:

```
docs: 新增註解
```

```
docs: 修正型別註解

讓 IDE 可以讀取到正確的類別
```

```
docs: 移除過期的註解

issue #1229
```