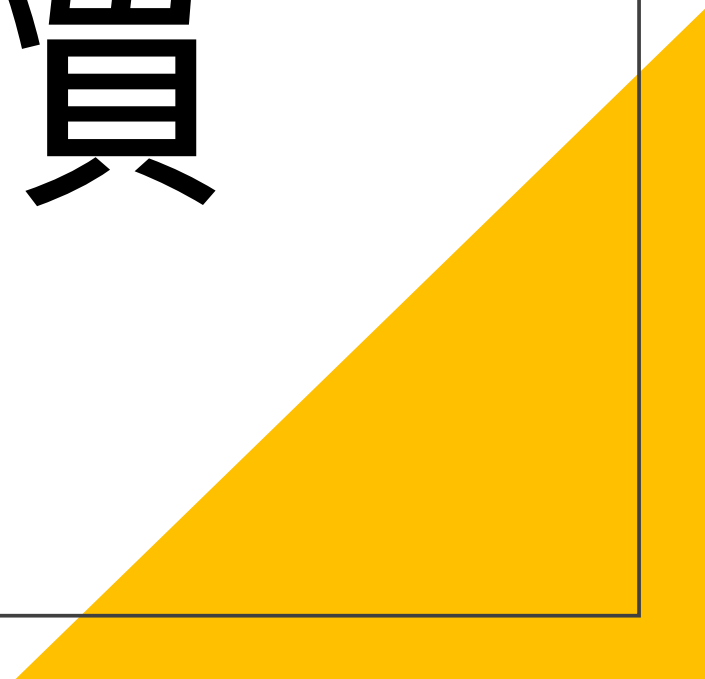


Coding 習慣

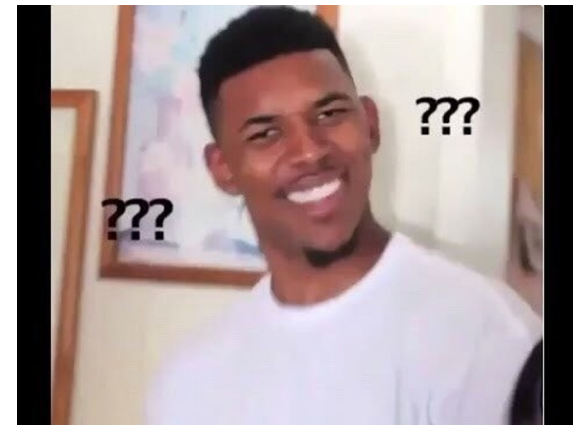


Coding 習慣

好的 Coding 習慣能讓程式在開發 / 維護 / 觀看上更方便也更快，不一定要去完全模仿寫 Code 大神的寫法，可以根據開發團隊的討論來決定。
但是以下幾點是建議各位一定要遵守。

1. 命名方式

命名要清楚，讓人一眼就看得出來這個是什麼（例如：使用者名稱 userName），不要自創縮寫（真的需要就打上註解），讓後面一起開發 / 維護的夥伴有辦法一眼就清楚這是什麼東西。



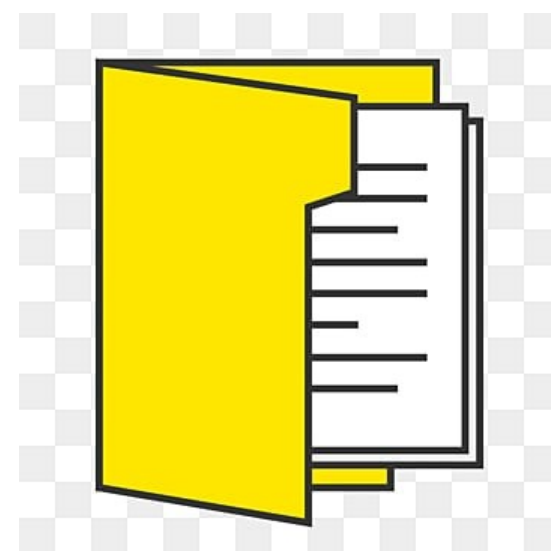
Coding 習慣

2. 檔案放置位置

專案裡有需多檔案，我們稱呼為 layout。

舉例來說 RESTful API 的檔案就會固定放在 RESTful API 檔案的資料夾中，如果每個人都隨便擺放那後面需要修改或者維護時就會有極高的機率找不到檔案或者少修改某隻檔案。

檔案的放置位置可以團隊統一說好就好，已團隊開發的便利性為主。

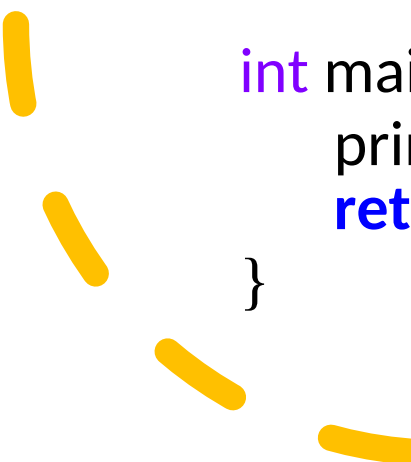


Coding 習慣

3. 排版

排版對於開發與除錯都有很大的幫助。已下面兩張圖來說，你們覺得左邊還是右邊比較好閱讀呢？

這還只是一個方法而已，當程式越開發越大你的 code 會越來越多跟越來越長，到時候就會變得非常可怕了！



```
int main() {  
    printf("Hello World");  
    return 0;  
}
```

```
int main() {printf("Hello  
World");  
    return 0;}
```

Coding 習慣

4. 註解

註解對後期維護與新開發者接手時都會有非常大的幫助，我們會將這段程式碼的出處 / 開發中尚未處理的問題 / 參考的資料出處 / 誰開發了哪段程式碼與開發日期，寫進註解讓大家有跡可循。

但是要請注意！註解需要讓大家都能理解，而不是自己明白就好，否則就喪失了寫註解原本的用意了。



Coding 習慣

5. 盡量避免重複程式碼

當你發現你的 code 中有越來越多一模一樣或者 8~9 成像的程式碼出現，那你的 code 就已經是個不健康的 code 了。

當你發現這段 code 很多地方都會使用到，而你每次都是把這段複製貼上時，請你將這段 code 取出並且將其變成共用方法，而不是每次都複製貼上。

否則將來如果有要修改或者這段程式碼有錯誤時，就很有遺漏到沒有修改到其中某個檔案而導致系統出現問題。



分享練

