

# 文章檢查 Checklist

## 1. 文章檢查: 脈絡通順 (a)、編排有層次 (b)、首中尾段主旨明確 (c)

- a. 檢查各段標題, 確保由因至果
- b. 檢查各段標題, 確保由淺入深
- c. 檢查首中尾段主旨, 確保首段破題、中段論證、尾段總結

## 2. 段落檢查: 段題結構一致 (a)、主旨呼應段題 (b)

- a. 檢查段題結構, 確保同層段題排比
- b. 檢查段落內文, 確保依段題橫向展開論述

## 3. 句子檢查: 句型簡單 (a)、用詞簡單 (b)、用詞明確 (c) (d)、文法正確 (e)、標點符號正確 (e)

- a. 以英文五大句型檢查, 確保屬五大句型
- b. 以 Fry List 字庫檢查, 確保屬字庫用詞
- c. 以 DeepL 翻譯檢查, 確保譯文與原文意思一致
- d. 以 Ludwig 工具檢查, 確保用字、片語達 Exact 判定
- e. 以 Grammarly 工具檢查, 確保達 99 分判定

## 4. 格式檢查: 排版格式 (a)、文字格式 (b)、圖表格式 (c)

- a. 檢查排版格式, 確保各段間格一行, 邊界留白
- b. 檢查文字格式, 確保同層文字行距、縮排、字體、字體大小一致
- c. 檢查圖表格式, 確保圖片、表格皆標齊對正、格式一致

# 程式碼檢查 Checklist

## 1. 程式功能: 功能需求檢查 (a)、功能執行檢查 (b)、例外處理功能檢查 (c)

- a. 檢查功能實作, 確保工作項目要求之功能皆有實作
- b. 檢查功能執行, 確保所有功能執行時無 Error 和 Warning
- c. 檢查例外處理, 確保程式能捕捉並處理所有例外情況

## 2. 程式設計: 函式重造檢查 (a)、函式功能檢查 (b)、測試檢查 (c) (d)、品質檢查 (e)

- a. 檢查函式功能, 確保專案或函式庫中沒有重複的功能
- b. 檢查函式功能, 確保一個函式只做一件事
- c. 檢查測試功能, 確保涵蓋各修改函式
- d. 檢查測試功能, 應驗證預期、非預期輸入下的結果
- e. 以 SonarLint 檢查程式碼, 確保解決所有偵測到的問題

## 3. 程式寫作: 命名檢查 (a)、Coding Style 檢查 (b) (c)

- a. 檢查變數與函式命名, 確保變數、類別、文件以名詞/名詞短語命名, 函式以動詞開頭, 且格式符合專案之程式碼規範 (e.g., PEP8, Google Java Style)
- b. 以 Code Spell Checker 工具檢查, 確保程式碼拼寫無誤
- c. 以 PyLint / CheckStyle 等工具檢查, 確保符合專案之程式碼規範 (e.g., PEP8, Google Java Style)

## 4. PR 準備: Commit 合併檢查 (a)、Commit 訊息檢查 (b)、專案 PR 規範檢查 (c)

- a. 檢查 Commit 合併, 確保 PR 前合併所有 Commit 至一個
- b. 檢查 Commit Message 內容, 確保符合《文章檢查 Checklist》第3項
- c. 檢查專案規定之 PR 程序, 確保完成專案規定之 PR 準備作業

# 清單如何優化工作流程

## 兩份清單如何優化工作流程：

1. 檢查清單可作為寫作和寫程式的標準，減少錯誤
2. 檢查工作會非常具體、清楚、沒有模糊空間，減少反覆檢查的時間

## 兩份清單將工作流程簡化成 6 步驟：

1. **描述工作項目：** 依照《文章檢查 Checklist》的標準，寫出《工作項目描述》
2. **實作程式：** 依照《程式檢查 Checklist》的標準與《工作項目描述》，實作出程式
3. **檢查程式：** 依照《程式檢查 Checklist》檢查並修正程式碼
4. **補充實作結果：** 實作結果補充至《工作項目描述》，當成PR 文章
5. **檢查 PR：** 依照《文章檢查 Checklist》檢查並修正 PR 文章
6. **提交 PR：** 提交 PR 至專案