

組譯器結構：

- 由 Python 語言編寫
- 組譯過程：每讀到一個新的 label 就更新一次 symbol table
- 輸出的 symbol table 會按照 label 出現的先後排序

(也就是 Source statement 的第 1 個 column)

實作過程中學到或發現的東西：

- 發現只要那行指令有 label，則那行指令的 Source statement 裡，label, instruction, value 都會存在，沒有空值。
- 發現所有的假指令(assembly directives)後面的 value 全都是常數
- 發現指令後面的 value 可能是 1 byte 或 3 byte 的常數，像 EOF 是 3 byte，而其他則為 1 byte。
- 學到如何將 python 程式打包成 exe 檔
 - 這份 code 我是用 macbook 寫的，原本想要直接在 macbook 上面將 code 變成 exe 檔，我先是安裝 pyinstaller、docker，然後再下指令編譯，結果這整個過程花了將近一個小時，而且最後產生的執行檔不是 exe 檔，也無法執行，後來從網路上爬文，試了幾個方法也依然失敗。
 - 所以我便放棄用 macbook，改用 windows 電腦將 python code 變成 exe 檔，步驟變少了，編譯的速度也提升很多，不用一分鐘就能產生 exe 檔，而且執行結果是正確的。

實作記錄：

1/6 開始進行

完成字串處理

完成 symbol table 的生成

1/7

將程式打包成 exe 檔

撰寫這份 report

程式功能：

生成 symbol table

版權聲明：

這個 SIC/XE 組譯器 100%由我生成，且沒有參考任何同學的程式碼。

程式執行：

Windows：執行 exe 檔

macOS：需要安裝 Python，然後在終端機移到該程式所在的目錄，並輸入 `python fp.py`。

開始執行後，會印出 symbol table，請使用者按下 enter 鍵離開程式。