第一份code（檔名：爬類別前100.ipynb）（給類別排名網址，爬書單）：

Input：Amazon第一層類別排名之網址（手動複製貼上）

<https://www.amazon.com/gp/bestsellers/books/ref=s9_bw_cg_bsmpill_1c1_w?ref=bsm_nav_pill_print&pf_rd_m=ATVPDKIKX0DER&pf_rd_s=merchandised-search-1&pf_rd_r=D8X1VSEE7D19EHQVWHZ6&pf_rd_t=101&pf_rd_p=be7d2af7-7d62-473f-ad08-a5445d8bcb9c&pf_rd_i=16857165011>

這是總榜（全部amazon紙本書前100名，請複製左邊各類別排名之網址，三十多個）

Output：

* (檔名：Best\_Sellers\_1dep.csv)書單:Amazon第一層的前100名書籍之書名、商品網址、該書在該類別之排名、所屬類別、作者、format（hardcover, paperback, kindle, …etc）、價格、抓取時間
* (url\_2.txt)第二層排名之網址（與前面Output分開，另外輸出檔案）

如果目的是要爬第二層書單，就要先爬第一層書單(主要是url\_2.txt，Best\_Sellers\_1dep.csv是副產物)

爬第二層書單時，記得改存檔檔名

#（以此類推，可用該code爬取第二、三層書單，以第三層為例，Input為Amazon第三層類別排名之網址，Output為Amazon第三層的前100名書籍之書名、商品網址、該書在該類別之排名、所屬類別、作者、format（hardcover, paperback, kindle, …etc）、價格、抓取時間、第四層排名之網址）

第二份code(檔名：爬商品頁面.ipynb)（給書單，爬詳細商品資訊）：

Input：(檔名：Best\_Sellers\_2dep.csv)第一份code的Output（第二層書單）

# 中間處理：得到第二層書籍的Output之後，可以根據第三份code的Output，找出感興趣的第二層類別，在爬第三層時篩選書籍，篩選每個類別前20名之總排名在10000名以內之類別，然後爬蟲。

Output：

* (檔名: AmazonBS\_other\_info\_2dep.csv)書之總排名、作者、format（hardcover, paperback, kindle, …etc）、打折前價格、打折後價格（如果該書沒打折，只會出現打折後價格）、所屬類別、其他詳細資訊、爬取時間
* (not\_append.txt)沒有讀到商品頁面的書，需再確認

第三份code(找出熱門類別.ipynb)（給書單和詳細商品資訊，篩選類別）：

Input(檔名AmazonBS\_other\_info\_2dep.ipynb)：第二份code的Output

Output：(檔名Paper\_dep1\_agg.csv)第二層類別中每個類別前10、20…名之書籍，在總排名是多少名（由於只會採計一種format，在該類別之第20名，不一定是該類別&format的第20名），

第四份code(爬特定出版社.ipynb)（給HarperCollins書單，找詳細商品資訊）：

Input：HarperCollins書單（其他人提供）

#與第二份code不同，由於HarperCollins書單沒有商品網址，需查詢該書後才能找到商品網址

Output：

* (HarperCollins\_Amazon.csv)書之總排名、作者、format（hardcover, paperback, kindle, …etc）、打折前價格、打折後價格（如果該書沒打折，只會出現打折後價格）、所屬類別、其他詳細資訊、爬取時間
* (not\_append.txt)沒有讀到商品頁面的書，需再確認

（有些書可能還沒上市，會有爬不到排名的狀況）

第五份code(barnesandnoble.ipynb)（爬barnesandnoble類別之網址）：

Input：'https://www.barnesandnoble.com/h/books/browse'

（從類別總覽中爬各類別之網址）

Output：(third\_df.csv)barnesandnoble第三層類別之網址