

## B07901103 電機三 陳孟宏

### [Prerequisites]

1. Python 執行
2. Import 套件: hashlib, json, sys, os, time, pickle (都不用 pip install)

### [How to use my pseudo bitcoin]

1. 進到 b07901103BTC 資料夾 (cd b07901103BTC)
2. 執行 block.py (主函數在這, python block.py)
3. 他一開始會請我們輸入指令 (選項如下, 每個步驟都有防待處理, 若沒打出 valid 的指令, 會繼續跑出相同的提示)
  - -h: (help 指令, 列出所有可執行指令)
  - exit: (終止此次執行檔)
  - printheight: 印出目前高度
  - addblock -transaction {任何字串}: 開始挖 genesis block
    - i. 再來他會問要不要再挖下一個 block (y/n), 若選'y (yes)', 繼續挖, 若選'n (no)', 停止這次的挖區塊, 並將這次所有的區塊存成區塊鍊, 建一個 json 檔記錄到 database 的資料夾中, 我存了一個 json 一個 pickle。  
Json 是視覺化好處理, pickle 是 load 資料很方便
    - ii. 結束完每次的挖區塊會跳回前面請我們輸入指令的

### 步驟 (第 3 步)

- printchain
  - i. 他會先列出 database 裡面所有的 block
- printblock -height {(int)height}
  - i. 印出某個特定高度的 block, 如果數字不是介於 0~高度-1 之間會跳 error message

### [Functionality]

- Block: 建立一個 block, 存進 height, prev\_block\_hash, timestamp, bits, nonce, transaction, hash 的值
- Block chain: 建立一個 genesis block, 並用 list 將所有 block 存起來
- PoW: 將所有 block 資訊做 encode 然後 hash, 再用 run 函數去跑 nonce, 加到產生的 hash 比 target 小
- Database: 將區塊鍊存成 json 跟 pickle, 前者是方便視覺化觀看, 後者是處理 load 資料比較方便
- Client: 執行介面 (見上點 how to use). 主要執行三個規定功能—addblock, printchain, printblock