

Command Line 常用指令

Shell 與 Command Line

作業系統介面就稱為 `shell`。 `shell` 提供讓你與作業系統通訊的方法，藉由提示您輸入指令，進而啟動和控制其他程式

當你登入作業系統後， `shell` 就會顯示指令提示，一般通常會是 `$` 符號顯示。當你在指示上輸入指令並按 `Enter` 鍵後， `shell` 就會評估指令並執行。這個讓你可以輸入指令的介面就叫命令提示介面 (Command-Line Interface, *CLI*)

相對的名詞就是圖形化介面 (Graphical User Interface, *GUI*)，也就是我們慣常使用的 Windows 操作介面

終端機 Terminal

在圖形化介面的操作環境下，我們會需要透過終端機 (terminal) 才能提供輸入指令 (command) 的介面

Windows 環境建議使用 `Git Bash`，`PowerShell` 或 `Windows Terminal`
Mac 環境建議使用 `iTerm2`

其實在 GUI 下的操作，例如拖曳、點擊等動作，這些操作會被作業系統轉換成相應的 command，再進行實際的動作。例如我們在 Windows 透過檔案總管，將某個檔案從「下載」資料夾移到「我的文件」資料夾，其實背後就是執行類似 `move` 指令來實現檔案移動

位置相關的指令

列出目前所在位置

\$ pwd

列出目前所在位置的檔案與資料夾列表

\$ ls

列出目前所在位置的檔案與資料夾列表(包含隱藏檔)

\$ ls -a

列出目前所在位置的檔案與資料夾列表(包含隱藏檔與詳細資訊)

\$ ls -al

切換路徑

跳到其他資料夾

\$ cd

跳到上一層資料夾

\$ cd ..

跳到目前的資料夾

\$ cd .

複製檔案

使用 `cp` 指令，將 來源檔案 或 來源資料夾 所指定的檔案或目錄內容，複製到 目的檔案 或 目的資料夾 所指定的檔案或目錄中

```
# 複製檔案
```

```
$ cp prog.c prog.bak
```

```
# 複製資料夾
```

```
$ cp -R 來源資料夾 目的資料夾
```

遷移並重新命名檔案

使用 `mv` 指令將檔案和目錄從一個目錄移至另一個目錄，或重新命名檔案或目錄

```
# 搬移檔案
```

```
$ mv 來源檔案 目的檔案
```

```
# 重新命名
```

```
$ mv 原始檔名 新的檔名
```

刪除檔案

使用 `rm` 指令移除不需要的檔案或資料夾

```
# 刪除檔案
```

```
$ rm 欲刪除的檔案
```

```
# 刪除資料夾
```

```
$ rm -r 欲刪除的資料夾及其包含的所有檔案
```

```
# 刪除所有檔案 (千萬不要做)
```

```
$ rm *.*
```


顯示檔案內容

使用 `cat` 指令會讀取指定的檔案內容，並將它寫入標準輸出（通常是在終端機或命令提示介面的視窗）

```
# 讀取並顯示檔案
```

```
$ cat 檔案
```

```
# 印出公鑰內容
```

```
$ cat ~/.ssh/id_ed25519.pub
```

啟動 Visual Studio Code

如果已經有安裝 [VSCode](#)，可以使用 `code` 指令來啟動 Visual Studio Code 編輯器

```
# 啟動 vs code
```

```
$ code
```

```
# 啟動 vs code 並以目前資料夾為專案起始路徑 (推薦)
```

```
$ code .
```

CLI 心智模型（初學者版）

- **Shell**：像是 `bash` / `zsh` / `fish`，負責解讀你輸入的指令
- **Terminal**：顯示輸入/輸出的視窗
- **Command**：`程式` + `參數(args)` + `選項`，例如：`ls -al /usr`
- **目前路徑**：`pwd` 顯示；相對路徑(`./`，`..`) vs 絕對路徑(`/`，`~/`)
- **Tab 補全**：輸入部分名稱後按 `Tab` 補齊，大幅減少打字與錯字

安全操作與常見選項

- ⚠ 刪除：優先用 `rm -i`（互動確認），避免誤刪；`rm -r` 會遞迴刪資料夾
- 常見選項：
 - `ls -a` 顯示隱藏檔、`ls -l` 詳細列表、`cp -R` 複製資料夾
- 路徑小幫手：`~` 家目錄、`.` 目前、`..` 上一層

更多常用指令（精選）

```
# 檔案/資料夾
mkdir demo && cd demo      # 建立資料夾並進入
touch notes.txt            # 建立空檔案
rmdir empty-dir            # 刪除空資料夾

# 檢視內容
less notes.txt             # 可翻頁檢視，q 離開
head -n 5 notes.txt        # 前 5 行
tail -n 5 notes.txt        # 後 5 行；tail -f 可即時追蹤

# 搜尋
grep -R "TODO" .           # 由目前路徑遞迴搜尋文字
find . -name "*.md"        # 依檔名樣式尋找

# 管線與重導向
cat notes.txt | wc -l      # 統計行數
echo "hello" >> notes.txt  # 追加寫入

# macOS 小技巧
open .                    # Finder 開啟目前資料夾
pbcopy < notes.txt        # 複製到剪貼簿；pbpaste 反向

# 資訊與協助
which node                # 程式所在路徑
whoami; date              # 使用者與現在時間
man ls                    # 手冊；多數指令也支援 --help
```

迷你練習（可直接複製貼上）

```
mkdir play && cd play  
echo "line1" > a.txt && echo "line2" >> a.txt  
cp a.txt b.txt && mv b.txt notes.txt  
grep -n "line" a.txt  
tail -n 1 a.txt  
cd .. && rm -r play # 若不確定，改用 rm -ri play 逐一確認
```

CLI Cheat Sheet (10 行)

```
pwd; ls -al          # 目前位置與清單
cd ..; cd -          # 上一層; 返回前一路徑
mkdir x; touch f      # 建資料夾/檔案
cp -R src dst        # 複製資料夾
mv a b               # 重新命名/搬移
rm -ri folder        # 安全刪除資料夾
cat/less/head/tail   # 檢視檔案
grep -R "text" .      # 搜尋文字
open .               # Finder 開啟
man cmd / cmd --help # 查看說明
```