Git:可以存取某時點的程式碼狀態，在Git管理檔案的話，更新歷史會保存在Git。

優點

. GIT是分散式版本控制系統

分散式版本: 每個人的電腦是完整的Repository，如果有人的電腦壞掉檔案不見，可以從別人那邊copy，同一份文件各自修改只要傳給對方看對方就可以看到修改的地方。

checkout

->切換分支

指向提交時用~(Tilde)和^(caret)，指向某個提交的相對位置。最常用的就是在HEAD後面加上~(Tilde）可以指向第幾代的父代；^(caret) 可以指向幾世代以前的父代

Stash是暫時儲存檔案修改內容的區域

完成作業的Topic分支，最終都會合併到Integration分支。合併分支有2種方法：使用「merge」或「rebase」。根據使用的方法合併後的分支歷史記錄有很大的差別。

non fast-forward 參數選項，即使是可以 fast-forward 的合併也會建立新的提交並合併喔。

Merge

修改內容的歷史記錄會維持原狀，但是合併後的歷史紀錄會變得更複雜。

Rebase

修改內容的歷史記錄會接在要合併的分支後面，合併後的歷史記錄會比較清楚簡單，但是，會比使用 merge 更容易發生衝突。

快速指令

建立數據庫

$ git init

在要建立數據庫的目錄裡執行init命令。

$ git commit

添加-a參數，就可以檢測出有修改的檔案(不包括新增的檔案)，將其加入索引並提交。這些操作只要一個指令就可以完成了。

加上-m參數，就可以指令提交“提交訊息”。如果不添加-m參數，就會啟動修改提交訊息的編輯器。

$ git status

加上 -s 參數，僅會顯示已修改的檔案名稱。

如果在 -s 後再加上 -b 參數，則會顯示分支的名稱。

$ git diff

僅使用 "diff" 命令時，會顯示工作目錄和索引的差異。

加上 --cached 參數，會顯示索引與 HEAD 的差異。

如果指定特定的HEAD 或提交，則可以顯示工作目錄和指定HEAD/提交之間的差異。

顯示提交記錄

$ git log

僅使用“log" 將顯示分支的提交列表。

如果要查看特定檔案的提交記錄，請指定檔案名稱。

查看提交的詳細記錄

$ git show <commit>

請使用show命令，show命令的參數可以指定log命令參閱的提交與HEAD。

修改/移動一個檔案/目錄的名稱

$ git mv <oldfilename> <newfilename>

Back To Top

刪除檔案

$ git rm <file>

從工作目錄刪除非管理對象的檔案

$ git clean

加上 -n 參數，可以查看將被刪除的檔案。

若添加 -f 參數則會立即刪除檔案。

在預設情況下 .gitignore 檔案中所列之檔案/目錄並不在刪除範圍內。不過如果加上-x 參數， .gitignore 檔案中的檔案/目錄也會從工作目錄中刪除。

Back To Top

還原在工作目錄已更改的檔案

$ git checkout -- <file>

git log

列出所有commit的歷史紀錄

git diff

比較文件或commit間修改的差異

git show

詳細列出該次commit的修改內容