1. Git與TFS有什麼不同

簡述如下，完整請見<https://www.ithome.com.tw/node/77088>

* 1. TFS

集中式版控，同一份程式碼的歷程統一保存在TFS Server上

* 1. Git

分散式版控，在本機電腦與版控Server上各有一份版本控制，可以透過Push(將本機歷程紀錄推送到Server) Pull(將Server歷程記錄拉回到本機)等指令雙向同步本機與Server上的版本歷程

1. 分散式版控的優點
   1. 適合多人合作開發

每個人都能有自己的工作副本，方便多人協同合作

* 1. 離線開發

在不方便連網時，本機版控可以協助開發者在自己電腦做到版本控制，待有連網時，再將歷程上傳

* 1. 多次微調時同時兼具版控歷程保護與Server歷程清楚明瞭

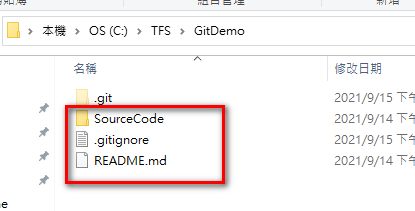
本機開發時可能會因為一份檔案多次微調而簽入多次，但對專案歷程來說其實只需要知道最終結果，所以上傳時能針對簽入歷程進行調整

* 1. 可同時針對同一支程式進行多種版本開發

一些實驗性功能或測試，可以在本機建立N個不同的分支，同時開發卻互不干擾

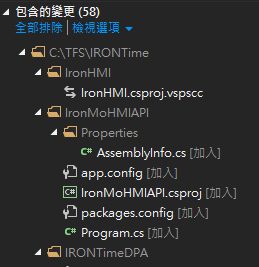
1. Git架構
   1. 工作目錄(Working Directory)

以GitDemo這一個目錄來說，紅框處就是工作目錄了，其實就是我們日常開發的,cs等…檔案存放路徑



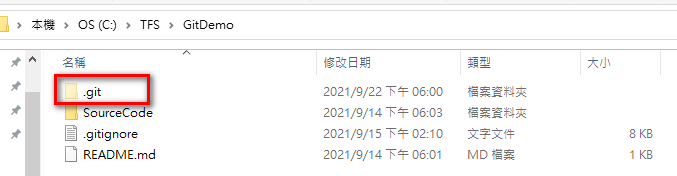
* 1. 暫存區(Staging / Index)

Git在把檔案加入版控前，會先加入暫存區，有點像下圖的TFS等待簽入的區域，執行git add.(將所有工作目錄已變更檔案加入暫存區)或是git add abc.cs(將abc.cs加入暫存區)



* 1. 儲藏庫(Repository)

這就是Git的版控儲藏區，執行git commit “註解”，就會將暫存區的檔案簽入儲藏庫，下圖紅框處就是本機儲藏庫位置，相關的設定檔也會在裡面



* 1. 忽略檔案

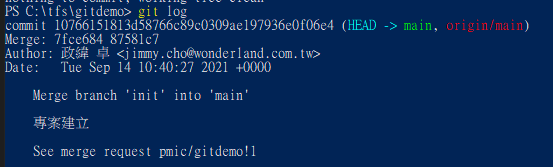
如果有檔案不需要簽入版控(例如dll)，可以增加.gitignore檔，此網站可以輸入關鍵字產生.gitignore檔<https://www.toptal.com/developers/gitignore>

如果無法下載的話, 用這個檔案 "R:\PMICIT\技術文件\Git\gitignore"



* 1. Commit

每簽入(commit)一次就會產生一個commit記錄(同時會產生commit blob tree三種檔案，blob是簽入的檔案，tree是紀錄檔名與資料夾結構，commit是記錄本次簽入包含哪些tree與blob檔案)，這也就是TFS的歷程記錄，commit會是一個sha1檔名的物件，如下圖1076615…0f06e4，在利用commit下指令時可以只輸入前幾碼(通常用前7碼)，在做一些復原或是一些進階操作例如部分合併時，會經常用到commit物件



* 1. Branch(分支)

在搭配Git開發時，通常會從主線建立分支，在分支上開發，開發完成後才會合併到主線，可以達到同時多人或多項需求同時開發(建分支開發新功能的同時，可以再建分支修改bug) 。

Branch在Git中實際上是一個指標檔案，指向某一個commit檔案，而每次commit時，Branch就會自動指向新的commit檔案。

當現在只有main主線分支時，結構會是branch(main)指標指向commit(c1)，此時建立新的分支test1，會產生一個branch(test1)指標指向commit(c1)，接下來我們切換到branch(test1)分支後，修改檔案執行commit，會產生commit(c2)，branch(test1)會自動改指到commit(c2)，branch(main)則停留在commit(c1)。

* 1. Tag

Tag指的是對某一個commit下一個註解，通常會用在標註上線版本，例如這一次的commit是1.2.4版

* 1. 物件關係圖

關係如下圖，如果想要更了解可以參考此[網址](https://github.com/doggy8088/Learn-Git-in-30-days/blob/master/zh-tw/06.md)

