* 修改用戶

Git config --global user.name “name”

Git config –global user.email “email”

查看: git config user.name

Git config user.email

* 清空屏幕 clear
* 文件夾變成管理庫文件夾(文件夾目錄下)

Git init

可顯示隱藏文件ls –a

打開 open .git

創建 touch test.py

* 查看git內容

Git status

添加到管理庫 git add test.py/一次添加 git add .



* 修改紀錄log

日誌Git log (--oneline)

Stage提交 git commit –m “description”

狀態 git status (-s)

修改的不同處(unstage與commit狀態的不同) git diff

如果已經在stage狀態 git diff --cached

如過在stage上修改 git diff HEAD

* 回到從前Reset、commit之間往返

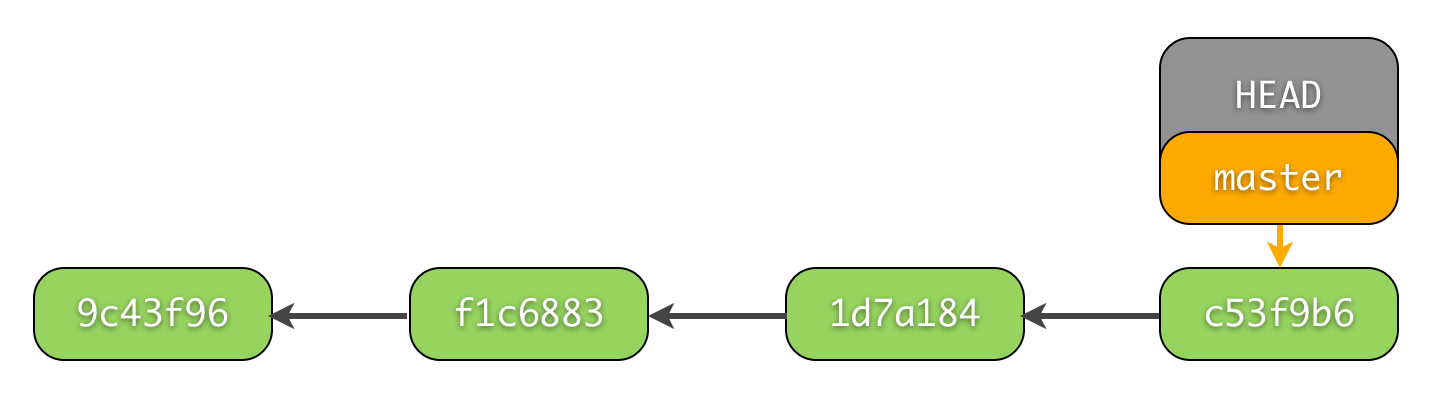
原有commit增加的修改(還是需add，修改後仍是同一個commit，commit ID不一樣) git commit --amend --no-edit

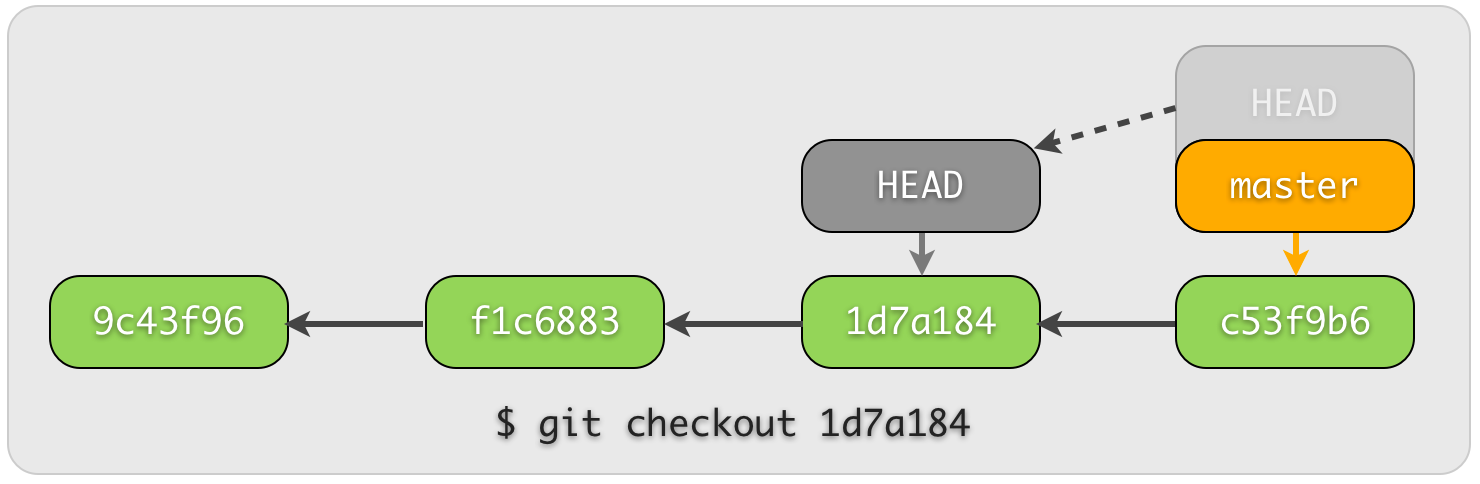
Stage返回modify狀態 git reset test.py

Unmodify返回(指定回到的狀態) git reset --hard HEAD(^ or ~1\2\3)

Git reset --hard ID

可以看到每一步的變化Git reflog





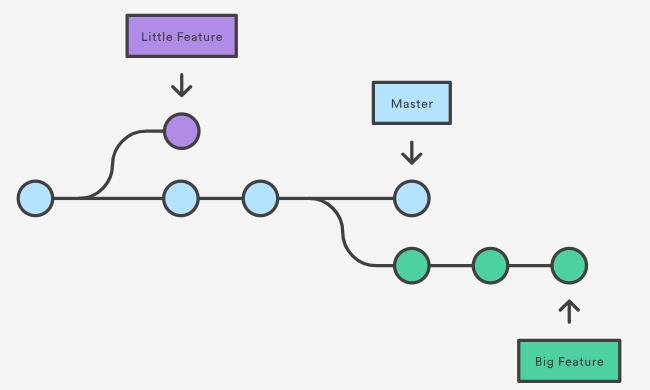
* 回到從前checkout

對單個文件進行

Git checkout ID -- test.py

回到過去並修改文件後執行git add🡪git commit

* 分支branch



Git log –online –graph

建立分支 git branch name

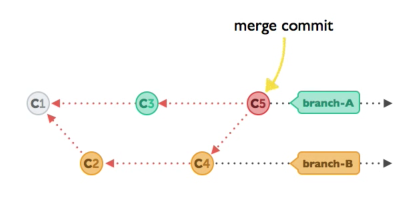
查看所有分支 git branch git checkout –b name(同時移動到分支上)

切換分支 git checkout 分支名

刪除 git branch –d name

分支推到master版本(在master上) git merge --no-ff --m “description” name

* 分支衝突merge



方法:人工合併(修改文件)



* Github

關聯origin(當前文件夾): git remote add origin [url]

Master版本上傳到origin: git push –u origin master(或分支)

本地更新: git pull

下載git clone [url]