**1. Tạo web bằng github:**

<https://www.youtube.com/watch?v=zWZDH9L1jMg>

Các step có thể hiểu đơn giản như sau:

B1: Tạo repository VD: Exercise1

B2: Tạo branch có tên **gh-pages**.

B2: Vào Setting của repo Exercise1, tìm mục Branch ở menu bên trái và chọn **default branch là gh-pages**.

B3. git clone về máy, đưa nội dung web page của mình vào thư mục vừa clone.

Lưu ý, phải có file **index.html** ở ngoài cùng của folder.

B4: push lên server.

**=>**  Repository của Github như 1 server chứa toàn bộ file html, ảnh, audio,…

**2. Git command: Luôn luôn xem xét chỗ đứng hiện tại là folder nào !!!**

**- git checkout <tên branch>:** switch branch => VD: git checkout branch1, git checkout master,…

**- git branch <tên branch>:** tạo branch

**- git checkout –b <tên branch><tên branch hiện tại>:** vừa tạo vừa checkout code từ branch hiện tại

**- Hiển thị branch :**

**-** Các branch local **: git branch**

**-** Các branch local + server : **git branch –a**

**- git branch -d <branch\_name>:** remove local branch

**- Sửa commit message :**

Ghi đè commit mới nhất bằng option : **git commit –amend**

Add thêm file mà không muốn thay đổi commit message : **git add &lt;tên file&gt; → git commit –amend –no-edit**

**- Hủy commit cũ :**

- Nếu đã commit những thay đổi, có thể undo lại commit đó : git revert &lt;commit&gt;

Ví dụ :

git revert 068b9b9 → tạo ra commit thứ 2, undo thay đổi của commit chỉ định

- Có thể dùng để undo một commit cũ hơn, tuy nhiên có thể dẫn đến conflict nội dung code nếu các

commit mới hơn nó cũng sửa trùng nội dung đó

**- Theo dõi log :**

git log → Xem lịch sử commit

git log --pretty=oneline –abbrev-commit → Xem lịch sử commit dạng 1 dòng ngắn gọn

git log --author=<name> → Xem những thay đổi của 1 thành viên trong project

git log --name-status → Xem lịch sử những file đã thay đổi

**- git add:**

## git add .  vs  git add -A

Both of these will stage all files, including new files (which git commit -a misses) and deleted files.

The difference is that git add -A also stages files in higher directories that still belong to the same git repository. Here's an example:

/my-repo

.git/

subfolder/

nestedfile.txt

rootfile.txt

If your current working directory is /my-repo, and that's the root of this git repo (thus the .git/ folder), these two commands will do the same thing.

But if you rm rootfile.txt, then cd subfolder, git add . will not stage the change that ../rootfile.txt has been deleted, because git add . only affects the current directory and subdirectories. git add -A, on the other hand, will stage this change.

**- git reset HEAD <tên\_file>:** xóa bỏ file ra khỏi Staging Area.

**3. Một vài lưu ý về Git:**

1. Khi clone từ github về, ta đã tự động kết nối local working với github.

Khi ta thêm file gì đó, và push lên, trên github sẽ tự động cập nhật.

2. Trước khi checkout, phải tạo 1 folder trước. VD:

helloworld

- test

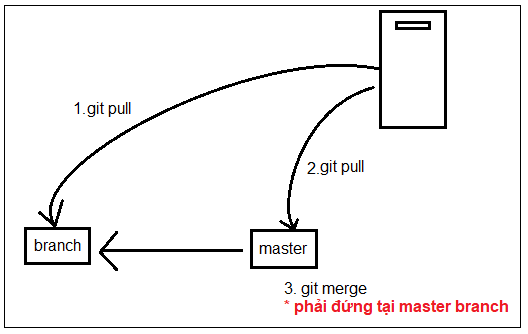
ta phải đứng ở folder helloworld, rồi dùng lênh **git checkout –b test <tên branch>**

3. merge source bằng lệnh:

B1: (đứng ở branch phụ) git pull

B2: (đứng ở master) git pull (**update source mới nhất**)

B3: (đứng ở master) git merge <tên branch phụ>



4. Lưu ý git checkout:

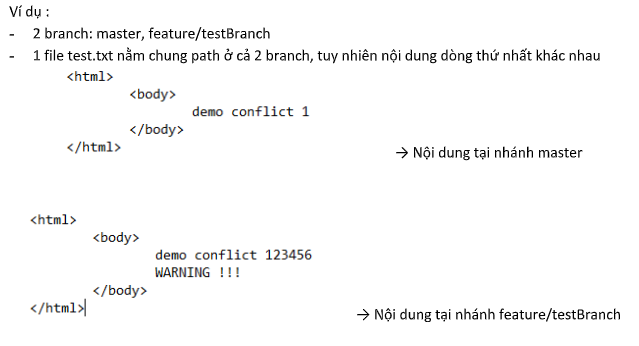
Tạo branch mới và lấy toàn bộ source từ master đưa vào new\_branch:

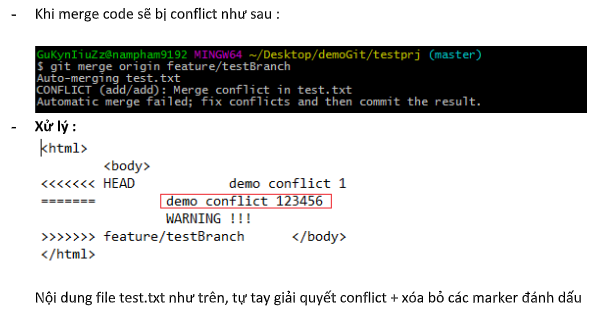
đứng từ master dùng git checkout –b new\_branch master

Nếu ta thêm 1 file nào đó bên new\_branch và switch branch master, file đó cũng sẽ được tạo ra trong master (master và new branch có cùng số lượng và nội dung file).

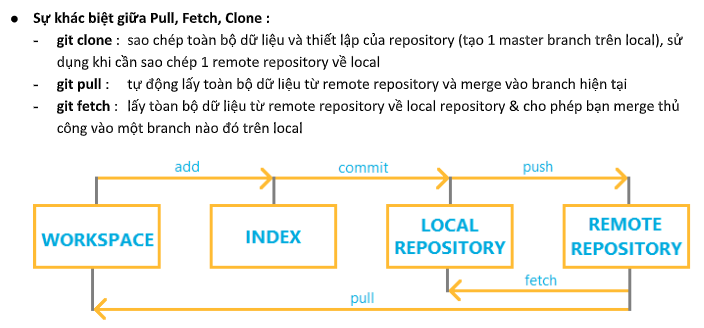
Tuy nhiên nếu ta cho push file lên server ở new branch, thì khi switch branch master, file đó sẽ ko xuất hiện trong master, vì khi này hệ thống **đã phân biệt được 2 branch.**

Sau khi file dc push lên ở new branch, trong **branch master** ta cũng add file **trùng tên và tạo nội dung chung 1 dòng** với file ở new branch, khi merge sẽ gây ra conflict





5. Sự khác biệt pull, fetch, clone.



6. Trước khi Merge phải chắc rằng **không xảy ra lỗi.**

7. git commit –amend:

Giả sử ta đã commit rồi, nhưng lại thiếu file, ta thêm file mới vào và commit lần nữa , ta dùng **commit –amend –m “message”**.

8. Liết kết working directories với Github:

C1: git clone [link github]

C2: git pull [link github]

git remote add origin [link github]

9. Liên kết local với remote branch:

Giả sử ta tạo 1 local branch:

**git checkout –b localbr\_1**

Sau đó liên kết branch đó với remote branch có tên remotebr\_1

**(đứng tại branch localbr\_1)git pull origin remotebr\_1**

10. git merge

- git merge chỉ dành cho **local** repository.

- git merge chỉ được dùng cho branch khác branch đang được chỉnh sửa.

VD:

branch testbr:

add file test.txt

Muốn merge vào master thì phải:

**switch qua branch master**

**git merge testbr**

11. Nếu chỉ muốn thay đổi nội dung 1 branch của local repo thì dùng git commit, không dùng git push.

12. git checkout – [tên file]:

[tên folder]/\* : download folder và các file con

13. Branch <branch\_name> set up to track remote branch <branch\_name> from 'origin'.

Để hiện dòng trên, ta cần **git fetch** để lấy thông tin từ remote repo đưa vào local repo, sau đó tạo 1 branch dùng **git checkout <branch\_name>**

14.

Git: private/ public key (<https://thachpham.com/linux-webserver/huong-dan-ssh-key.html>)

15. -init Github repo **by using command line** => ko có cách (Xem thêm <https://thachpham.com/tools/cach-tao-repository-cho-git.html>)

<https://kbroman.org/github_tutorial/pages/init.html>

Nói chung cần phải tạo 1 repository trên Github hay 1 remote server nào đó trước rồi clone về, ko có lệnh tạo remote repo nào hết.

**4. Git Flow**

<https://kipalog.com/posts/Ban-da-hieu-ro-ve-git-flow-chua>

**Tham khảo:**

Xem thêm các link trên trên browser.

**Câu hỏi:**

**1.Clone duy nhất 1 folder từ remote repo?**

**2.Nếu remote server down, làm sao làm việc với local server.**

**VD: Delete file ở máy, muốn lấy folder chưa file đó từ local server thì làm như thế nào?**

**3. Ta có Update trong SVN, vậy trong Git Update như thế nào ?**

**4. Nếu push lên Remote server (thường là push lên branch phụ) để mọi người review trước khi merge thì lệnh git merge đâu còn tác dụng vì merge thường lên server merge luôn?**

**5. Các TH conflict:**

<https://docs.google.com/presentation/d/19ohvMxiQH0nnMCJFzAFuTi96zKNbd8Tuok0XDK0rD5c/edit#slide=id.ga2b4b13e1_4_0>