**Hướng dẫn học GIT**

**1. Mục tiêu**

* Hiểu Git và các nguyên tắc cơ bản của VCS (version control software)
* Cài đặt và cấu hình Git
* Sử dụng Git để kiểm soát phiên bản trong suốt vòng đời phát triển ứng dụng
* Thao tác với GitHub
* Sử dụng các công cụ Git khác

**2. Thời gian học:**

2 ngày (Bao gồm các buổi review kiến thức tổng hợp, cá nhân tự ôn tập và làm bài test.)

**3. Link tài liệu:**

[Slide Overview] [GIT.pptx](https://orientsoftware365.sharepoint.com/:p:/s/Training.DN/EQncQQT57yNJg2j1lI2TqaABAbtco46bN6Hnva3A7pQNfg?e=CfNWBR)

[References]

* <http://git-scm.com/book> - 4 chương đầu
* <http://backlogtool.com/git-guide/vn/intro/intro1_1.html>
* <https://learngitbranching.js.org/> - Tool thực hành git branch
* <http://chris.beams.io/posts/git-commit/>

[Note]

* Các bạn tạo account ở link [<https://github.com/>] để thực hành nhé

**Một số tip khi làm việc với GIT**

* Đôi bạn cùng tiến
  + trước khi git checkout thì git status
  + trước khi git commit thì git log --oneline
* khi nào sử dụng commit -m và commit --amend

git commit --amend // đối với lần commit thứ 2 trở đi.

Vậy làm sao biết được đây phải là lần commit đầu hay ko (dùng cho não cá vàng) -> các bạn chạy lệnh git log --oneline

* Xem nội dung chỉnh sửa mới nhất: git diff file\_name

Thực hành: step1: thêm 1 dòng text vào file a -> git diff a xem sự khác biệt step 2: thực hiện tiếp tục:

git add .

git commit -m "test git dif"

// thêm 1 line trống trong file a

git diff a

Xem thay đổi (chưa được add) của những file hiện tại: git diff

Xem thay đổi (đã được add, chưa commit): git diff --cached

Xem những files thay đổi giữa hai commits: git diff --name-only COMMIT1\_ID COMMIT2\_ID

Xem thay đổi trước khi push git diff --cached origin/master

* Lỗi n commit trong 1 pull request: Thực hiện git rebase 2 commit
* git rebase -i HEAD~2
* // đọc và làm theo hướng dẫn :p
* Trước khi push code lên các bạn nhớ rebase với branch develop - mới nhất nhóe

// đứng tại branch của mình git rebase develop

* + có conflict: fix conflict -> git add . -> git rebase --continue
  + không có conflict git push origin name\_branch
* Lỡ push 1 file/folder không cần thiết lên pull request

// file

git rm --cached file\_name

// folder

git rm -r --cached folder

* Vô tình fix comment nhầm branch (yaoming) 1 Nếu qua đầu là bờ: tức là chưa chạy lệnh git commit thì chỉ việc git checkout sang branch bị comment

2 Lỡ phóng lao thì phải theo lao: Lỡ bấm commit thì git reflog

// chọn commit mà mình muốn quay trở lại git reset id\_commit //sau đó làm theo số 1 :D

// Ngoài ra, các bạn có thể tìm hiểu thêm git reset (\*) reset

**Happy coding!!!**