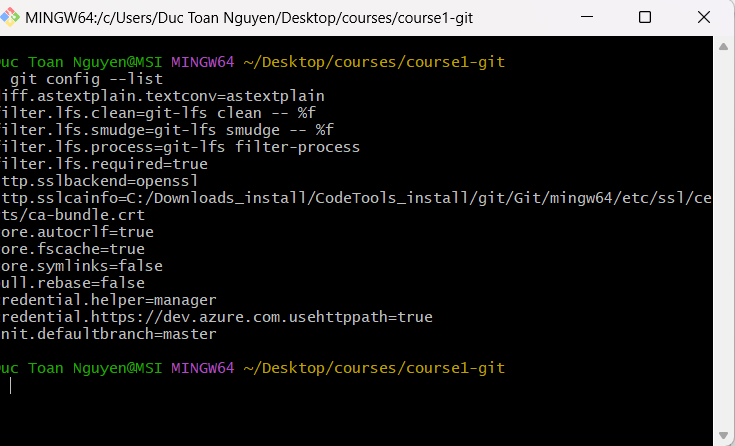
Khóa học git và github: https://www.youtube.com/playlist?list=PLncHg6Kn2JT6nWS9MRjSnt6Z-9Rj0pAlo

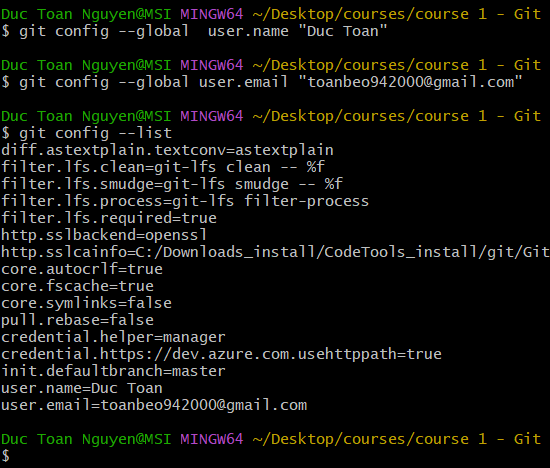
Thiết lập username và email điều này làm mỗi khi tạo 1 repository mới trên github

* git config –-list



Nếu chưa thấy mình sẽ thiết lập nó, lưu ý email ở đây là email mình dùng để tạo tài khoản github

* git config –-global user.name “Tên người dùng”
* git config –global user.email “Email người dùng”



Để xóa thông tin

* git config --global --unset user.name
* git config --global --unset user.email

Cách để kéo code về

* git clone “link kéo code về”

Cách để đẩy code lên github

Trường hợp tạo 1 reponsitory mới hoàn toàn và đẩy code lên

Cách 1:

* git init
* git add README.md
* git commit -m "first commit"
* git branch -M main
* git remote add origin “link nơi lưu code”
* git push -u origin main

Cách 2:

* git init
* git commit -m "first commit"
* git remote add origin “link nơi lưu code”
* git push origin master

Cách 1 khác cách 2 là cách 1 nó tạo nhánh main còn cách 2 thì tạo nhánh master

Trường hợp đã có code trên github rồi nhưng muốn đẩy code mới hoặc chỉnh sửa hoặc xóa

* git add .
* git commit -m “đánh giá”
* git push

các câu lệnh thường gặp:

* git config –-global user.name “Tên người dùng”
* git co nfig –global user.email “Email người dùng”
* git init
* git clone “link”
* git status
* git commit -m “comman”
* git push

git pull

kéo code trên github về nó khác gì với git clone.

git clone là kéo cả cục code lớn về

còn git pull là khi mình tham gia dự án có 1 đứa mới cập nhập code mới kêu mình kéo code về vậy không lẽ mình kéo cả cục về hay sao? Mình chỉ nên kéo code mới thôi đó là điểm khác biệt giữa clone và pull

ví dụ: mình code trang home tôi code phần header và thằng kia nó code navbar vậy khi dùng pull thì sẽ kéo code phần navbar về. Không cần clone cả cục code

* git pull origin “nhánh muốn kéo về”

git merge

là 1 cấu trúc phân nhánh trên github

ví dụ: khi code xong trang home thì muốn phát triển trang detail sản phẩm

tôi sợ trang detail sẽ ảnh hưởng đến trang home nên tôi sẽ tách nó ra làm nhánh 2 và sau đó khi trang detail chạy ổn thì tôi sẽ ghép nó lại với trang home

git branch (kiểm tra coi mình có mấy nhánh và mình đang ở nhánh nào)

git branch “tên nhánh mới” (tạo 1 nhánh mới)

git checkout “tên nhánh” (chuyển qua 1 nhánh khác)

git checkout -b “tên nhánh” (tạo 1 nhánh mới và chuyển sang nhánh mới tạo ngay lập tức)

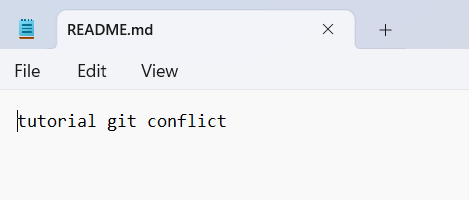
git conflict

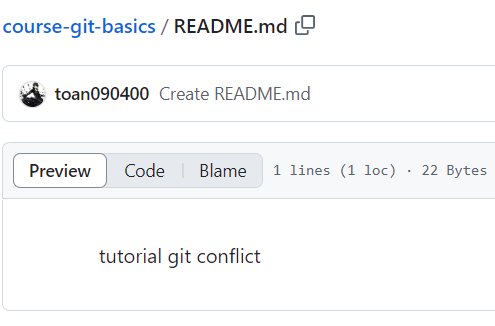
là trường hợp nhiều đứa cùng sửa 1 file

ví dụ: sửa trang home page 4,5 đứa sửa

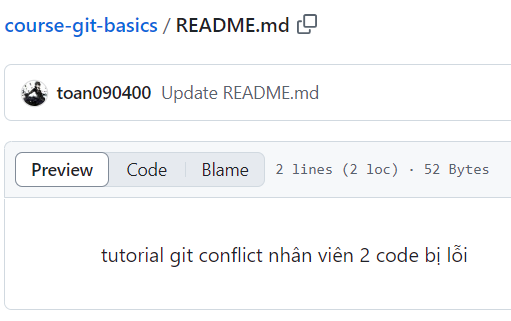
ví dụ thực tế:

đây là 1 flie đã hoàn thành và đã push lên git

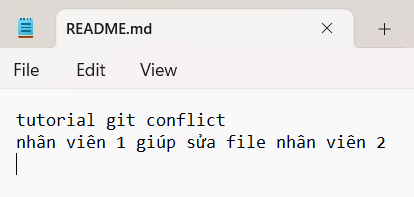




Sau đó 1 thằng khác nó muốn là thêm chức năng nhưng không biết sao nó bị lỗi (đây chỉ là trường hợp ví dụ). Và code lỗi này nó đã push lên github rồi

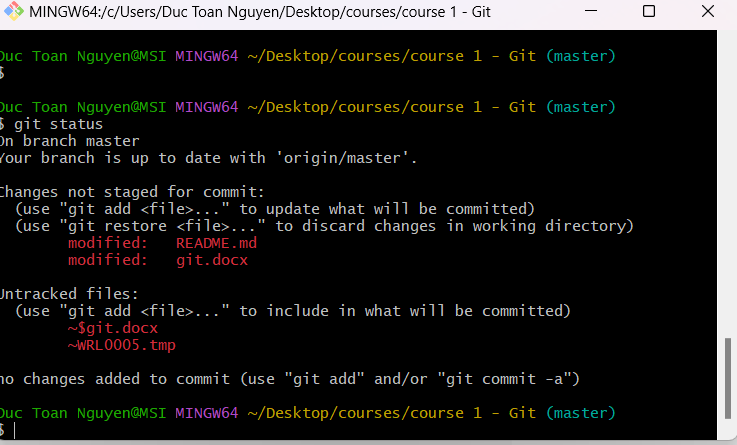


Tiếp theo nhân viên 1 sửa lỗi cho nhân viên 2 (chưa push lên github)

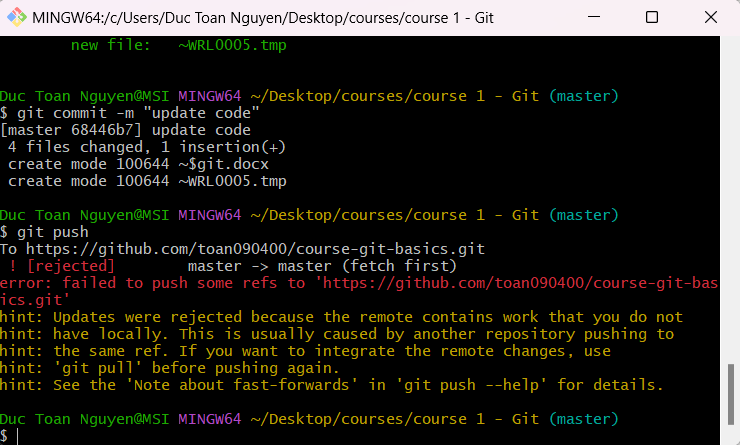


Nhân viên 1 đẩy code lên github

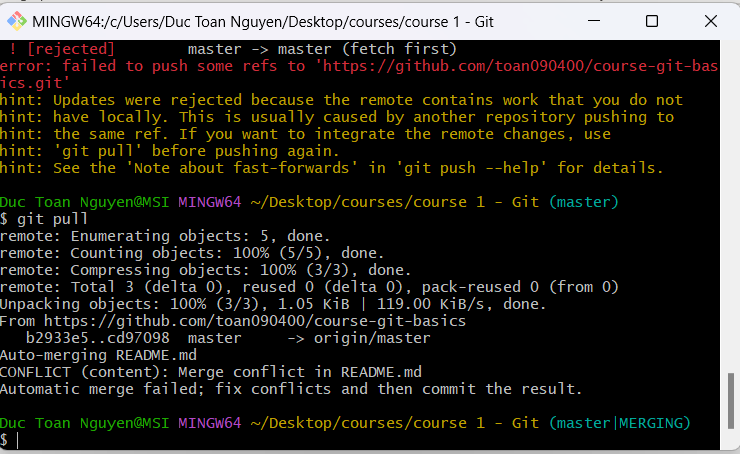
Kiểm tra git status



Sau khi thực hiện các thao tác đẩy code lên thì nó sẽ thông báo là bị lỗi



Gặp trường hợp này nó sẽ yêu cầu mình sửa sai bằng git pull

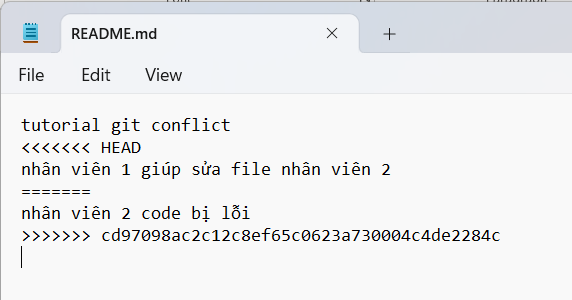


Thực hiện xong nó sẽ hiện cái thông báo này và nó cho biết là lỗi ở đâu nhìn ở chỗ để biết file cần sửa

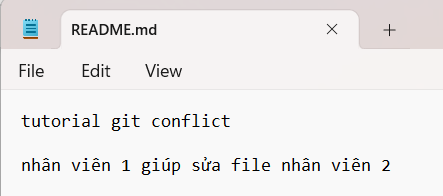
CONFLICT (content): Merge conflict in README.md

Chú ý sau khi thực hiện xong nhánh của mình không còn chỉ là master nữa chuyển sang (master|MERGING)

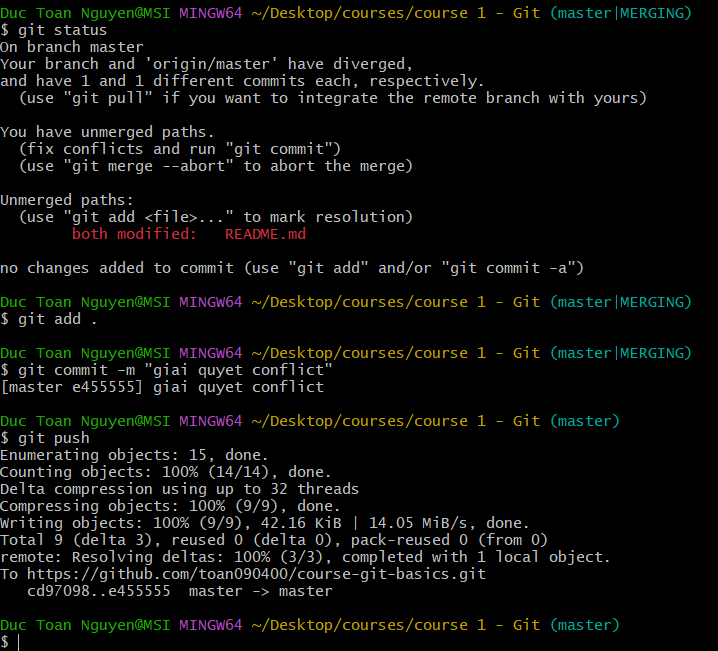
Sau đó chúng ta vào file bị lỗi và mở lên



Nhìn vào đây để sửa là xóa các ký tự đặc biệt và code cũ là được



Đây là nội dung chúng ta muốn sửa và sau đó chúng ta đẩy lên github như bình thường



Lên github xem lại file bị lỗi đã sửa xong

