:ساخت دایرکتوری

mkdir project-name

:ساخت فایل

touch page1.html

touch page2.html page3.html ساخت همزمان چند فایل

:حذف فایل

git rm footer.html

:شروع کردن تنظیمات اولیه

git init ……..یک دایرکتوری می سازد

:بررسی وضعیت

git status…...چی کا را کردی

وقتی فایل ها تغییر می کنند شما آن ها را کامل کرده پ به وضعیت استیج(پیش از کامیت) می برید و آماده کامیت می شویم

گیت اد فایل را به گیت می شناساند و به وضعیت استیج می برد میتوان همه یا فقط یکی را شناساند

git add filename.format

git add ”\*.html” یا فقط همه ی فایل ها بایک فرمت مشخص

git add -A

کامیت کردن و ثبت یک نقطه از تغییرات

git commit -m ‘comment’

:دیدن تاریخچه ی کامیت ها

هرکامیت کدی یونیک دارد می توان به هر کدام از آنها بازگشت

git log

گاهی در پروژه پیش می آید که حرکت پروژه دیگر به صورت خطی نیست به طور مثال در پروژه ی بانکی تا سه کامیت پیش رفته ایم و به ما میگویند کپجا را به پروژه بیافزا در اینجا مثلا ما از کامیت دوم یک برنچ جدا میکنیم و به گسترش

کپجا در آنجا می پردازیم و پس از تکمیل با پروژه ی اصلی در کامیت مثلا پنجم مرج می کنیم

برای بازگشت به یک کامیت برای برنچ اصلی و ادامه دادن:

git reset --hard 245a7cdbbbd5136f38be7fcad199434844453610

Dual lang com1

Dual lang com2

commit3

commit1

commit5

commit2

commit4

Capca com1

Capca com1

Capca com1

git branch fixpages ساخت برنچ

git checkout fixpages :change the branch

زمانی که یک فایل را از استیج حذف می کنیم

git reset file.format : for delete a file of the add and after that do commit

:در برنچ اصلی خود هستیم و می خواهیم برنچ فرعی را به برنچ اصلی مرج کنیم پس می نویسیم

git merge fixpages

git branch -d branchname حذف برنچ

: می توان از آدرس پروژه های گیتهاب برای ادیت آن پروژه در سیستم خود با دستور زیر استفاده کرد

git clone https://github.com/jadijadi/titap\_mystry.git

اد کردن یک اورجین جدید

git remote add origin <https://github.com/test-repo/>

git remote

git diff HEAD

gtt diff –staged

git push origin master

git push -u origin master

git pull origin master

git remote -v : for more informations

git push -u origin master

git tag -a v2.0 -m ‘first version with tag .this is what we are runnig for a long time’

git tag: it say your versions

git tag -a v1.8 d32432db343124 -m ‘comments’

git show v1.8

git push orogin v1.8 :for push versions

git checkout v1.8

gpg –gen-key

gpg --list-key

git config --global user.name

git config --global user.signingKey

gpg --list-secret-keys --keyid-format LONG

git config --global user.signingKey 3ED6683D36501F9F

git tag -s v2.1 -m ‘this is me’

git show v2.1

git tag -v v2.1

git commit -S -m ‘ssada’

git help command

git blame filename.format -L23

git blame filename.format ~~-L2,30~~

git bisect start

git log

git biscote bad

git biscot good

https://github.com/schahrijar/django\_manual\_test.git