با دستور زیر میتوان نام فولدری را عوض کرد

mv my\_first\_repository/ my\_even\_cooler\_project

سمت راستی نام جدید است

git init my\_really\_cool\_project

این دستور یک فولدر با نام داده شده ایجاد می کند که در ان یک فایل .git خالی قرار دارد البته اگر فولدر را ایجاد کرده بودی بعد خواستی گیت را وارد کنی کافیست در خط فرمان وارد فولدر پروژه شوی و دستور git init را اجرا کنی فولدری که این دستور داخلش اجرا شود master خواهد بود.

بعد از ان با cd واردش می شویم

فایل readme.me

یک شروع خوب برای ساخت یک پروژه این است که یک readme بسازیم. شاید بهتر است از چیزی جدی تر از یک شروع خوب حرف بزنیم زیرا اگر در push اول readme.md را اضافه نکنیم و بیاییم در push دوم ان را ایجاد کنیم خطایی می گیریم که برای رفع ان باید ابتدا یک pull و سپس مجددا push کنیم. البته در stack دستوری برای push اجباری هم هست که خوشم نیامد. اما اخرش مجبور شدم ازش استفاده کنم:

git push -f origin master

راه حل دیگر استفاده از سه دستور زیر است:

git remote add origin [//your github url]

//pull those changes

git pull origin master

//now, push your work to your new repo

git push origin master

برای ایجاد فایل readme در اموزشی که دیدم با این دستور کار کرد :

Nano README

این دستور البته برای من کار نکرد پس من هم رفتم دستی یک فایل تکست با نوت پد ایجاد کردم و ان را به داخل فولدر مربوطه کپی کردم بعد با این دستور ان را به پروژه افزودم دقت کنیم هرچه در داخل پوشه باشد لزوما داخل GIT نیست و این خوب است :

git add README.txt

بعد از انکه ان را افزودیم باید تغییراتمان را commit کنیم:

Git commit

Git commit –m “hello”

یک بار این کار را کردم سابلایم باز شد. هنوز خوب نگرفته ام چی به چیست.

نحوه ی config کردن git را که در ان اموزش هست اما اگر می خواهی بدانی کدام editor پیش فرض گیت شده این دستور را بنویس خودش بالا می اید:

git config --global –edit

می توان با touch فایل اضافه کرد اما گویا ان یک مرحله قبل از add است یعنی اول touch بعد add . البته می توان از همان اول هم add کرد انجه فعلا یاد گرفتیم این است که add شده ها commit می شوند .

اگر چند فایل را تغییر داده باشیم و می توانیم با دستور

Git commit –a –m “for all”

کامیت ها را برای همه ذخیره کنیم اما می توانیم یکی یکی هم کار کنیم این همان کاری است که به ان staging aria می گوییم یعنی انها را به staging atia اضافه می کنیم که برای اینکار باید ان را add کنیم ان فایلی که میخواهیم را addd می کنیم انگاه دستور commit را اجرا میکنیم.

Git log

Cat name\_of\_file

می خواهم در مورد فایل .ignore مطالبی را بگوییم :

هر سطر از این فایل دستور نادیده گرفته شدن است .

# کامنت است

قرار دادن این سطر موجب می شود که هر فایلی که پسوند sqlite3 داشته باشد را نادیده بگیرد

'\*.sqlite3'

* هرچه که ارسل شده یا در اصطلاح گیت track شده دیگر با گذاشتن نامشان در .ignore باز نمی گردند انها دیگر رفته اند و برای بازگرداندشان احتمالا راهی هست اما این راه هرجه هست میدانم که .ignore نیست
* ----------

این خطوطی است که داد:

git remote add origin https://github.com/sky1048576/mysecendrepositort.git

git push -u origin master

در مورد آن -u باید بگویم که در آموزشی دیدم که می‌گفت آن -u رو فقط برای بار اول که می‌خواهیم کد ها را پوش کنیم باید بگزاریم و در پوش های بعدی دیگر نیازی به این کار نیست که -u بگزاریم یا بهتر از آن حتی لازم نیست دیگر می‌توانیم فقط با دستور git push اطلاعات رو پوش کنیم به احتمال زیاد دلیل اینکه دستور git push خالی هیچ وقت برای من کار نکرده این است که هیچ گاه -u رو نگذاشته ام

با این دستوور میتوان فولدر جاری را در سابلایم تکست باز کرد:

subl -an .

فایل هایی که پسوند pyc دارند در زمان اجرا تولید می شوند و نیازی نیست که commit شوند.

در سایت به ادرس

<https://www.gitignore.io/>

می توان درخواست داد که فایل gitignore را برایمان درست کند.

من داخل این سایت رفتم و پایتون و جنگو را وارد کردم دیدم در فایلی که برایم create کرده همچین سطری وجود دارد:

\*.pyc

که این خود نشان می دهد که این حرف در مورد فایل های با پسوند pyc صحیح است.

با این دستور خواهیم دانست که مخزن محلی ما چه مخزن های انلاینی را می شناسد:

git remote -v

میتوان فقط یک فایل را add کرد:

git add base.html

میتوان فقط یک فایل را commit کرد:

git commit templates/base.html -m "this commit only add a date to base.html”

البته اگر قرار است محدودیتی باشد بگزار روی add کردن باشد معقولتر است.

تمام شد همه چیز را یاد گرفتم:  
دستور git log و همچنین انچه در گیتاب هست را اموختم

حالا میتوانم با git checkout sha به ان بروم

در ان هر کاری دلم میخواهد بکنم حتی کامیت کنم اما هیچند مگر اینکه شاخه ایجاد کنم با این دستور git checkout –b name اینجا هر نامی که میخواهم میتوانم بدهم بعد از ان دیگر مثل یک شاخه است و کافی است تغییرات را merge کنیم که اگر تداخل داشت هم که وضعیت مشخص است.

دستورات گیتاپ:

افزودن پروژه به گیت برای اولین بار :

git remote add origin https://github.com/sky1048576/sh\_p401

مخزن محلی ما چه مخزن های انلاینی می شناسد

git remote –v

push کردن اجباری که گویا دوباره از صفر push می کند چون خیلی طول کشید

git push -f origin master

افزودن فایل readme

subl readme.md

git push -u origin feature/VIRA-94

git push -u origin feature/edit