

## به نام خدا

گیت و گیت هاب

خیییییییی سریع



راهنمای خیلی سریع و بدون هیچ حرف اضافه  
برای بی حوصله های عزیز ، به زبان خودمونی و  
به دور از حرف های عجیب و غریب ❤️

## فهرست مطالب

- 1 – گیت چیه ؟
- 2 – مسیر پروژه ها
- 3 – گیت رو نصب کن
- 4 – دایرکتوری رو تحت گیت ببر
- 5 – مشخصات رو وارد کن
- 6 – git status
- 7 – git add
- 8 – git commit
- 9 – گیت هاب
- 10 – ریپوزیتوری بساز
- 11 – صفحه ی عجیب و غریب ولی مهم
- 12 – ReadMe
- 13 – تغییر در گیت هاب دادیم، لوکال چی ؟
- 14 – تغییرات لوکال رو بفرست روی گیت هاب

## درس اول : گیت چیه ؟

گیت چیه ؟ گیت یه سیستم بسیار سمج و گییییره !

یعنی چی ؟ یعنی هر کاری میکنی رو میبینی و ثبت میکنه 😬 😬

خب به چه درد میخوره؟ خیلی به درد میخوره 😬 وقتی داری پروژه انجام میدی و نیاز داری تا تغییرات پروژه ات ثبت بشه، وقتی داری مقاله مینویسی و دوست داری مراحل مختلف تکمیل شدن مقاله ات رو بتونی خیلی خوشگل و مرتب بنویسی، گیت دوست و همراه تو میشه توی این مسیر.

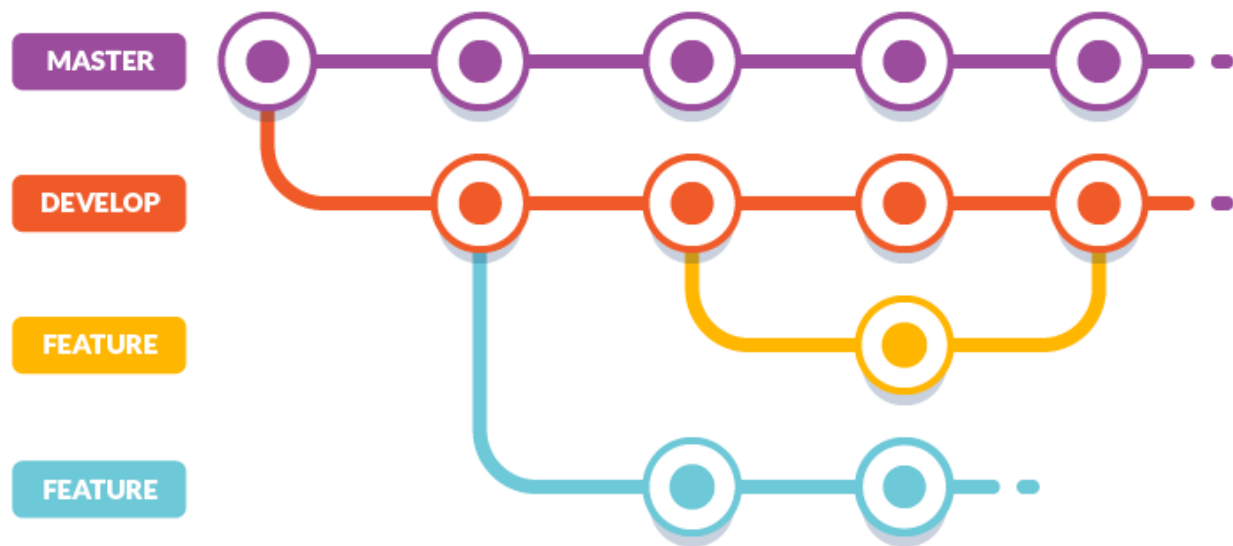
پس گیت میتونه دوست خوبی باشه برات ❤️

## درس دوم : مسیر پروژه ها

توی گیت یک مسیر صاااااااااااااااااااا یا همون صراط مستقیم وجود داره که اسمش main است.

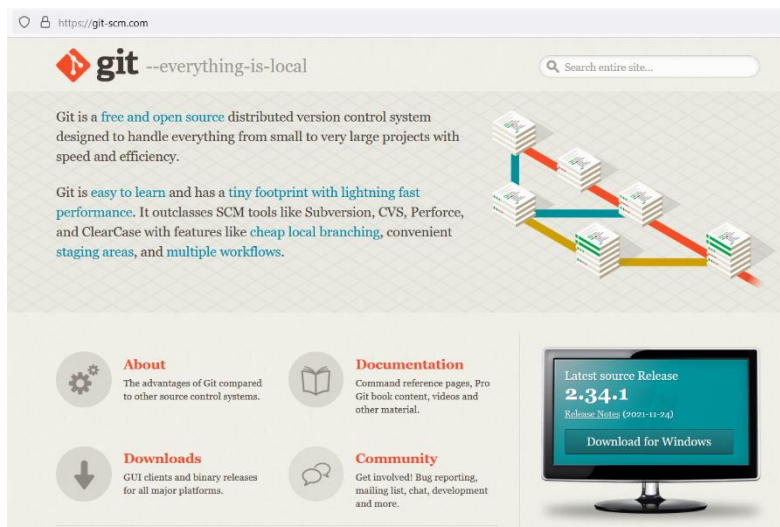
(تقریبا یک سال میشه که master به main تغییر نام داده)

همون راه اصلی پروژهته، همون چیزیه که روش زوم کردی و داری روش کارا انجام میدی. حالا ممکنه در کنار موضوع اصلیت، بیای و روی یک موضوع فرعی کار کنی ( هر چیزی میتونه باشه) به این موضوع فرعی میگن branch.



## درس سوم : گیت رو نصب کن

اگه لینوکسی هستی که گیت رو اتوماتیک داری ولی اگه ویندوزی هستی، میری توی سایت اصلی گیت، فایلش رو دانلود میکنی و next رو میزنی تا اaaaaا نصب بشه 😊👍 مطمئن باشید خودتون میتونید انجامش بدید و وقتی نصب شد با یک صفحه ی سیاه خوشگل روبرو میشید.



## درس چهارم : دایرکتوری رو تحت گیت بر

ما گیت رو نصب کردیم پس این حله.

دستور cd رو استفاده میکنیم تا از طریق گیت بش وارد پوشمون بشیم.

cd desktop و بعد هم cd FileName اینطوری وارد پوشه ای میشیم که دوست داریم اون رو تحت گیت ببریم.

خب بین خود گیت که نمیفهمه باید تغییرات کدوم پوشه یا دایرکتوری رو بررسی کنه؛ پس ما اول باید بهش بفهمونیم که اقا یا خانم گیت، شما باید تغییرات فایل X رو بررسی کنید.

وقتی وارد فایل شدیم از طریق دستور های بالا، از دستور git init استفاده میکنیم.

با این دستور، به اصطلاح ما پوشه یا دایرکتوری خودمون رو تحت گیت بردیم و از این به بعد هرررر تغییری توی این پوشه اتفاق بیوفته، توسط گیت ثبت میشه.

(برای دیدن مثال ها و توضیحات بیشتر پیشنهاد میکنم تا فایل GIT رو مطالعه کنید)

```
User@DESKTOP-02A6BKH MINGW64 ~  
$ cd desktop  
  
User@DESKTOP-02A6BKH MINGW64 ~/desktop  
$ cd FrontEnd  
  
User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd  
$ git init  
Initialized empty Git repository in C:/Users/User/Desktop/FrontEnd/.git/  
  
User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)  
$
```

## درس پنجم : مشخصات رو وارد کن

بعد از اینکه یک پوشه یا دایرکتوری رو تحت گیت بردید، نیاز داریم تا مشخص کنیم این فایل ها توسط چه کسی ساخته شده اند و تغییر کرده اند که به این کار می‌گیم کانفیگ کردن 😊

یعنی اگه 100 سال دیگه یکی اومد سراغ ریپوزیتوری ما، بدونه چه کسی با چه ایمیلی این تغییرات رو اعمال کرده 😊

```
git config --global user.name "your name"
```

```
git config --global user.email "your email"
```

```
User@DESKTOP-02A6BKH MINGW64 ~  
$ cd desktop  
  
User@DESKTOP-02A6BKH MINGW64 ~/desktop  
$ cd FrontEnd  
  
User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd  
$ git init  
Initialized empty Git repository in C:/Users/User/Desktop/FrontEnd/.git/  
  
User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)  
$ git config --global user.name "Ali Moeinian"  
  
User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)  
$ git config --global user.email "eng.alimoeinian1379@gmail.com"  
  
User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)  
$ |
```

## درس ششم : git status

این دستور وaaaaaaاقعا از به درد بخور ترین و کاربردی ترین دستور های گیت است که خیلی از شما استفاده خواهیم کرد 😊 و میاد آخرین وضعیت ریپوزیتوری شما رو میگیره.

از خوبی های این دستور اینه که شما میتونید با خوندن اون متنی که در جواب به شما میده، بفهمید دقیقا در چه مرحله ای هستید و درمراحل بعدی بهتره چیکار کنید.

```
User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd
$ git init
Initialized empty Git repository in C:/Users/User/Desktop/FrontEnd/.git/

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)
$ git config --global user.name "Ali Moeinian"

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)
$ git config --global user.email "eng.alimoeinian1379@gmail.com"

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    main.txt

nothing added to commit but untracked files present (use "git add" to track)

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)
$ |
```

خب فرض کن یه دایرکتوری ساختی ! تغییراتی هم توش دادی و الان نیاز درای تا



تغییرات خودت رو به گیت هم بفهمونی

بین الان گیت تغییرات تورو تحت نظر گرفته و بهت میگه که خب شما فلان فایل رو تغییر دادی و مثلا 4 تا خط بهش اضافه کردی.

الان وقتشه که شما بیای و تغییرات رو ثبت کنی و به گیت بگی که این تغییرات رو درون خودش ثبت کنه که این کار با دستور git add انجام میگیره.

**git add FileName**

```
User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)
$ git config --global user.name "Ali Moeinian"

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)
$ git config --global user.email "eng.alimoeinian1379@gmail.com"

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    main.txt

nothing added to commit but untracked files present (use "git add" to track)

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)
$ git add main.txt

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)
$
```



تا الان ما دایرکتوری ساختیم، اوردیمش تحت گیت، مشخصات خودمون رو وارد کردیم، یه سری تغییرات دادیم توی اون فایل و تغییرات رو ثبت کردیم.

اما کافی نیست 😞

گیت قانون های خودش رو داره و برای مرتب و تمیز بودن کار، توی این مرحله باید به وضوح توضیح کوتاهی بدید که این تغییراتی که شما دادید چی بوده ؟ همونطور که بالاتر هم گفتم، مثلا 4 خط به اخر یه فایلی اضافه کردید.

`git commit -m "Add 4 lines at the end of file"`

خب کار ما اینجا با گیت به پایان میرسه و دیگه کار خاصی با خود گیت نداریم و باید بریم سراغ گیت هاب 😊

MINGW64:/c:/Users/User/desktop/FrontEnd

```
User@DESKTOP-02A6BKH MINGW64 ~
$ cd desktop

User@DESKTOP-02A6BKH MINGW64 ~/desktop
$ cd FrontEnd

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd
$ git init
Initialized empty Git repository in C:/Users/User/Desktop/FrontEnd/.git/

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)
$ git config --global user.name "Ali Moeinian"

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)
$ git config --global user.email "eng.alimoeinian1379@gmail.com"

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    main.txt

nothing added to commit but untracked files present (use "git add" to track)

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)
$ git add main.txt

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)
$ git commit -m "main.txt added to start a make a new Repo, its not a important file"
[master (root-commit) 859e18f] main.txt added to start a make a new Repo, its not a important file
1 file changed, 4 insertions(+)
create mode 100644 main.txt

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)
$
```

## درس نهم : گیت هاب

توی این قسمت نیاز داریم تا اکانت گیت هاب داشته باشیم !

چرا ؟ گیت هاب یه سایت خیلی خفنه بر مبنای گیت.

یعنی شما میتونی کد ها، مقاله ها، یا هر چیز دیگه ای رو بزاری اونجا (بیشتر در موضوع برنامه نویسی)

اما چه فایده ای داره ؟ فایده اش اینه که از کل دنیا، هر کسی که علاقه داشته باشه میتونه بیاد و روی پروژه ی شما کار کنه و اون رو گسترش بده.

اما صرفا قرار نیست که پروژه ی کاملی روی گیت هاب قرار بگیره.

دقیقا همین فایل از وقتی هیچیش شروع نشده بود روی گیت هاب من قرار گرفت و تا به حال که داره کم کم کامل میشه هم روی اونجاست.

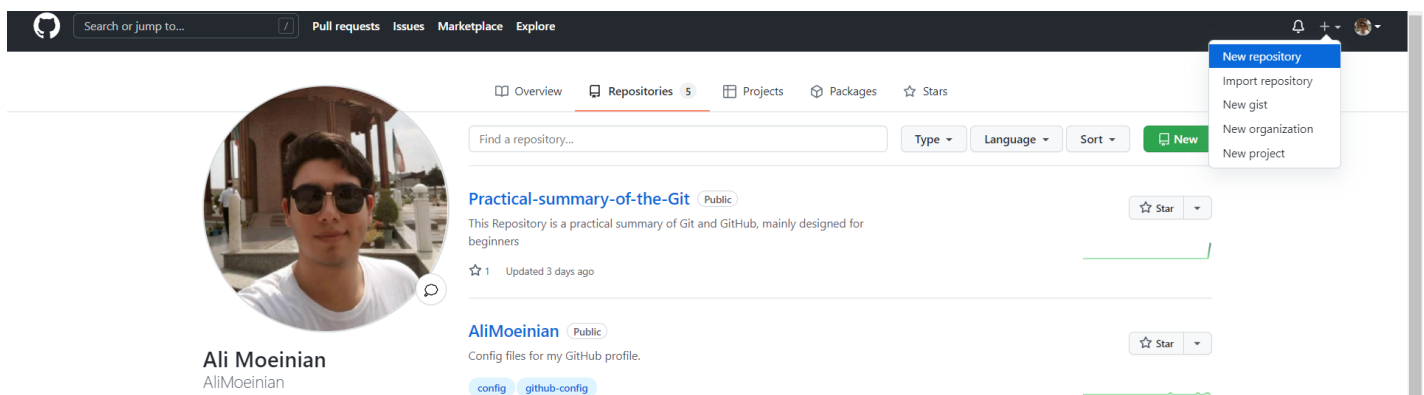
با یک سرچ بسیار ساده میرید توی سایت GitHub و اونجا برای خودتون اکانت میسازید.

## درس دهم : ریپوزیتوری بساز

هرررر فایل مجزایی که دوست دارید تحت گیت باشه و بیاریدش روی گیت هاب، نیاز به یک ریپوزیتوری جدید داره.

ریپوزتوری عین یک مخزنه که کد های شما یا هرررر چیزی که روی گیت هابتون هست رو داخل خودش نگهداری میکنه.

برای ساخت یک ریپوزیتوری، باید روی گزینه ی `new repository` کلیک کنید و به راحتی میتونید یه دونه اسم و یه سری توضیحات به دلخواه بنویسید و در نهایت هم گزینه ی `Create repository` رو بزنید و تمام 😊



توی صفحه ی جدیدی که برات میاد اسم ریپوزیتوریت رو انتخاب کن و تمام.

## درس یازدهم : صفحه ی عجیب و غریب ولی مهم

وقتی ریپوزیتوریتون رو ساختید، با صفحه ی زیر روبرو میشدید! خیلی مهمه ردش نکن بره 😊

بین الان فرض بکن یه پوشه ی خالی داری روی گیت هابت.

یه سری فایل آماده هم داری که با گیت همه ی کار هاش رو کردی.

فایل آماده ات روی لپ تاپته ( local ) و الان باید یه جوری این فایل های آماده ات رو بفرستی روی گیت هابت.

پس باید از دستور هایی که روی این صفحه ات میبینی استفاده کنی.

راستی برای امنیت بیشتر بهتره از ssh استفاده کنی 😊

تمام دستور های بخش مشخص شده رو کپی کن و توی git bash، پیست کن.

و در نهایت یه دونه enter خیلی زیبا بزن و تمام!!!!!!

حالا کافیه صفحه ی گیت هابت رو رفرش کنی تا فایل هاتو ببینی.

...or push an existing repository from the command line

```
git remote add origin git@github.com:AliMoeinian/Way-to-learn-FrontEnd.git
git branch -M main
git push -u origin main
```



با کپی کردن کل این قسمت در گیت بش و زدن enter کار ما تموم میشه و نتیجه میشه :

MINGW64/c/Users/User/desktop/FrontEnd

```
User@DESKTOP-02A6BKH MINGW64 ~
$ cd desktop

User@DESKTOP-02A6BKH MINGW64 ~/desktop
$ cd FrontEnd

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd
$ git init
Initialized empty Git repository in C:/Users/User/Desktop/FrontEnd/.git/

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)
$ git config --global user.name "Ali Moeinian"

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)
$ git config --global user.email "eng.alimoeinian1379@gmail.com"

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        main.txt

nothing added to commit but untracked files present (use "git add" to track)

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)
$ git add main.txt


User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)
$ git commit -m "main.txt added to start a make a new Repo, its not a important file"
[master (root-commit) 859e18f] main.txt added to start a make a new Repo, its not a important file
1 file changed, 4 insertions(+)
create mode 100644 main.txt

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)
$ git remote add origin git@github.com:AliMoeinian/Way-to-learn-FrontEnd.git

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (master)
$ git branch -M main

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 379 bytes | 75.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:AliMoeinian/Way-to-learn-FrontEnd.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.

User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (main)
$
```

 Search or jump to... Pull requests Issues Marketplace Explore

AliMoeinian / Way-to-learn-FrontEnd Public Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

Ali Moeinian main.txt added to start a make a new Repo, its not a important file 859e18f 5 minutes ago 1 commit

main.txt

main.txt added to start a make a new Repo, its not a important file 5 minutes ago

Help people interested in this repository understand your project by adding a README. Add a README

About

This Repository is my way to learn frontEnd.

0 stars 1 watching 0 forks

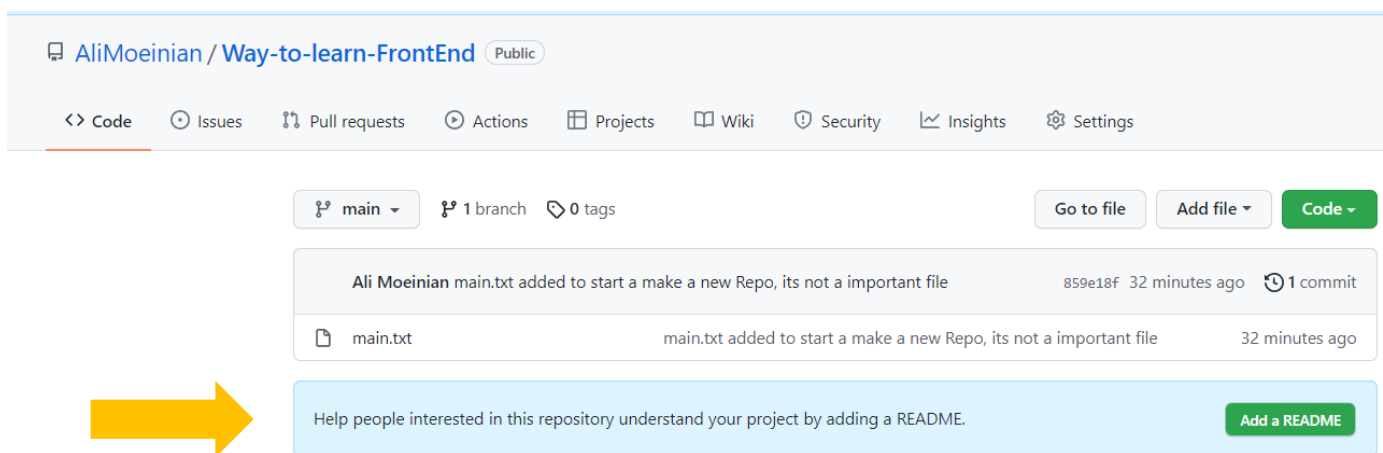
Releases

## درس دوازدهم : ReadMe

وقتی فایلتون رو روی گیت هاب قرار دادید، خود گیت هاب بهتون پیشنهاد میکنه تا به فایل جدید به ریپوزیتوریتون اضافه کنید که اسمش ReadMe است.

این فایل خوبه که باشه توی ریپوزیتوریتون و در اصل توضیحات کلی شما در ارتباط با این ریپوزیتوری و هدف شما از توسعه ی اون میتونه باشه.

در کل به چیزیه که وقتی یکی سراغ ریپوزیتوری شما میاد، بفهمه موضوعش چیه.



AliMoeinian / Way-to-learn-FrontEnd Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

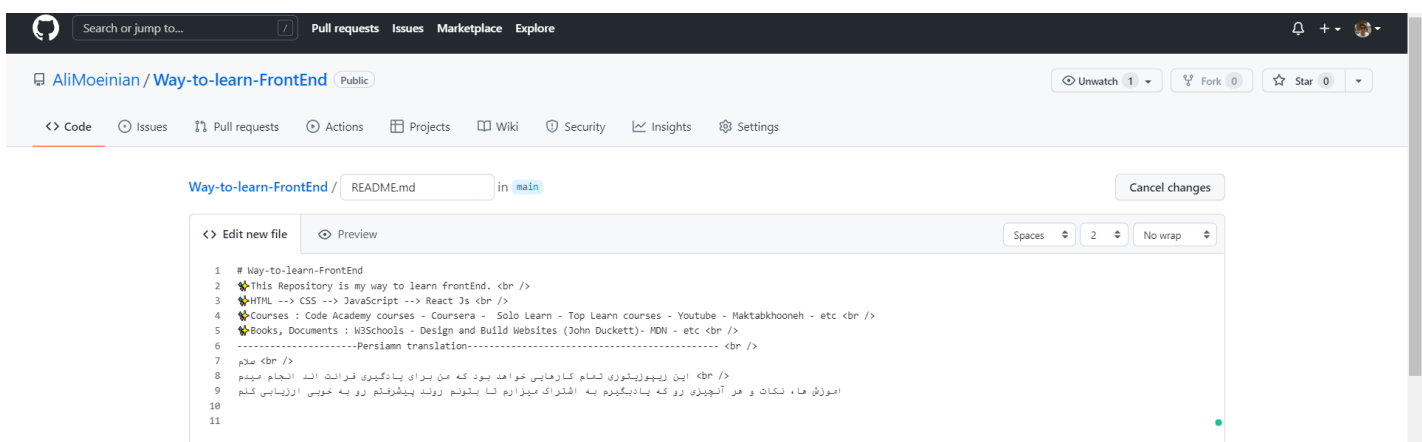
main 1 branch 0 tags

Go to file Add file Code

Ali Moeinian main.txt added to start a make a new Repo, its not a important file 859e18f 32 minutes ago 1 commit

main.txt main.txt added to start a make a new Repo, its not a important file 32 minutes ago

Help people interested in this repository understand your project by adding a README. Add a README



Search or jump to... Pull requests Issues Marketplace Explore

AliMoeinian / Way-to-learn-FrontEnd Public

Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Way-to-learn-FrontEnd / README.md in main

Cancel changes

Edit new file Preview

Spaces 2 No wrap

```
1 # Way-to-learn-FrontEnd
2 This Repository is my way to learn frontEnd. <br />
3 HTML --> CSS --> JavaScript --> React Js <br />
4 Courses : Code Academy courses - Coursera - Solo Learn - Top Learn courses - Youtube - Maktabkhooneh - etc <br />
5 Books, Documents : W3Schools - Design and Build Websites (John Duckett)- MDN - etc <br />
6 -----Persian translation----- <br />
7 سلام <br />
8 این ریپوزیتوری تمام کتابهای خواهد بود که من برای یادگیری فرانت اند انجام میدم
9 آموزش ها، نکات و هر آنچهی رو که یاد بگیرم به افکارم میزارم تا بتوانم روند پیشرفتم رو به خوبی ارزیابی کنم
10
11
```


## درس سیزدهم : تغییر در گیت هاب دادیم، لوکال چی؟

داستان اینه که وضعیت دایرکتوری روی کامپیوتر شما دقیقا باید عین وضعیت ریپوزیتوریتون باشه، اگر که بخواهید تغییرات رو کامیت کنید البته.




مثلا همیشه که شما یه تغییری رو از طریق گیت هاب روی پروژتون اعمال کنید و اون تغییر روی فایل لوکال شما اعمال نشه !

به طور مثال ما الان فایل README رو اضافه کردیم ولی هنوز روی لوکالمون نیست 😊

اینجاست که دستور git pull به کمکتون میاد و تغییرات رو از روی گیت هاب روی فایل لوکالتون اعمال میکنه.

 .git	12/22/2021 10:30 AM	File folder	
 main.txt	12/22/2021 10:20 AM	Text Document	1 KB

```
User@DESKTOP-02A6BKH MINGW64 ~/desktop/FrontEnd (main)
$ git pull
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), 1.75 KiB | 4.00 KiB/s, done.
From github.com:AliMoeinian/Way-to-learn-FrontEnd
 859e18f..5e00a41 main    -> origin/main
Updating 859e18f..5e00a41
Fast-forward
 README.md | 10 ++++++++
 1 file changed, 10 insertions(+)
 create mode 100644 README.md
```

 .git	12/22/2021 11:10 AM	File folder	
 main.txt	12/22/2021 10:20 AM	Text Document	1 KB
 README.md	12/22/2021 11:10 AM	Markdown Source ...	1 KB

## درس چهاردهم : تغییرات روی لوکال رو بفرست روی گیت هاب

یه بار از اول لازمه مرور کنیم :

یه فایل ساخته شده و رفته روی گیت هاب.

حالا پس فردا شما اومدی سراغش یه سری تغییر روش اعمال کردی و الان نیاز داری تا تغییرات جدیدت هم منتقل بشه روی گیت هابت؛ پس مراحل زیر رو باید به ترتیب بری :

git add FileName - 1

git commit -m "Commit" - 2

git push origin main - 3

دوری از عذاب جهنم : اگه موقع کامیت کردن وارد vim شدی، با دستور !q: میتونی ازش خارج بشی.

