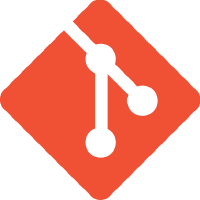
**به نام خدا**

**گیت و گیت هاب**

**خیییلییییی سریع**

****

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

**درس اول : گیت چیه ؟**

**گیت چیه ؟ گیت یه سیستم بسیار سمج و گییییره !**

**یعنی چی ؟ یعنی هر کاری میکنی رو میبینه و ثبت میکنه 🙄🤔**

**خب به چه درد میخوره؟ خیلی به دردت میخوره 🤨 وقتی داری پروژه انجام میدی و نیاز داری تا تغیرات 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 صااااااف یا همون صراط مستقیم وجود داره که اسمش MASTER است.**

**همون راه اصلی پروژته، همون چیزیه که روش زوم کردی و داری روش کارا نجام 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 تااااااا نصب بشه😀👍**

**مطمن باشید خودتون میتونید انجامش بدید و وقتی نصب شد با یک صفحه ی سیاه خوشگل روبرو میشید.**

**درس چهارم : دایرکتوری رو تحت گیت ببر**

**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 ببین 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 هررررر 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)**

**درس پنجم : مشخصاتت 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”**

**درس ششم : git status**

**این دستور واااااااااقعا از به درد بخور ترین و کاربردی ترین دستور های گیت است که خیلی ازش استفاده خواهیم کرد😃 و میاد آخرین وضعیت ریپوزیتوری شما رو میگه.**

**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.**

**درس هفتم : git add**

**خب فرض کن یه دایرکتوری ساختی ! تغییراتی هم توش دادی و الان نیاز درای تا تغییرات خودت رو به گیت هم بفهمونی 😁**

**ببین 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 تغییراتت 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**

**درس هشتم : git commit**

**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 🥰**

**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 😊**

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

**وقتی ریپوزیتوریتون رو ساختید، با صفحه ی زیر روبرو میشید! خیلی مهمه ردش نکن بره 😁**

**ببین 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 اماده 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 تمااااااااااااااام.**

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

**درس دوازدهم : ReadMe**

**وقتی فایلتون رو روی گیت هاب قرار دادید، خود گیت هاب بهتون پیشنهاد میکنه تا یه 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 یه چیزیه که وقتی یکی سراغ ریپوزیتوری شما میاد، بفهمه موضوعش چیه.🎈**

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

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

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

**یه بار از اول لازمه مرور کنیم :**

**یه فایلی ساخته شده و رفته روی گیت هاب.**

**حالا پس فردا شما اومدی سراغش یع 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 :**

**1 – git add FileName**

**2 – git commit -m “Commit”**

**3 – git push origin main**

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

**تقریبا مطالبی که دوست داشتم توی این فایل بیان بشه تمام شد**

**فقط اضافه کردن عکس های هر قسمت مونده که سر حوصله عکس ها رو هم اضافه میکنم تا یک فایل بسیار خلاصه و به درد بخور داشته باشید.**