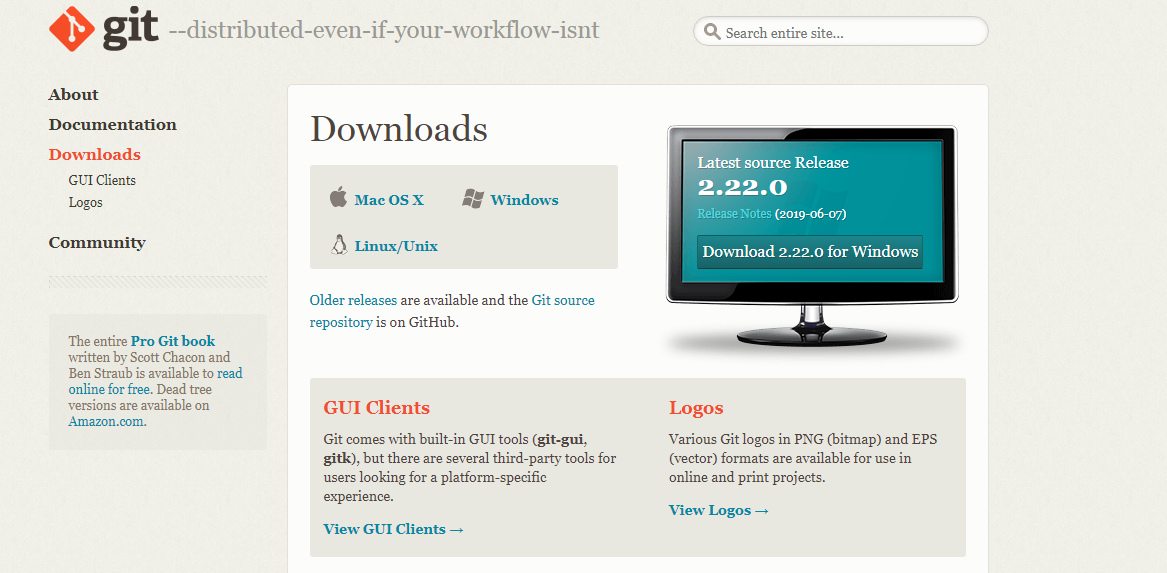
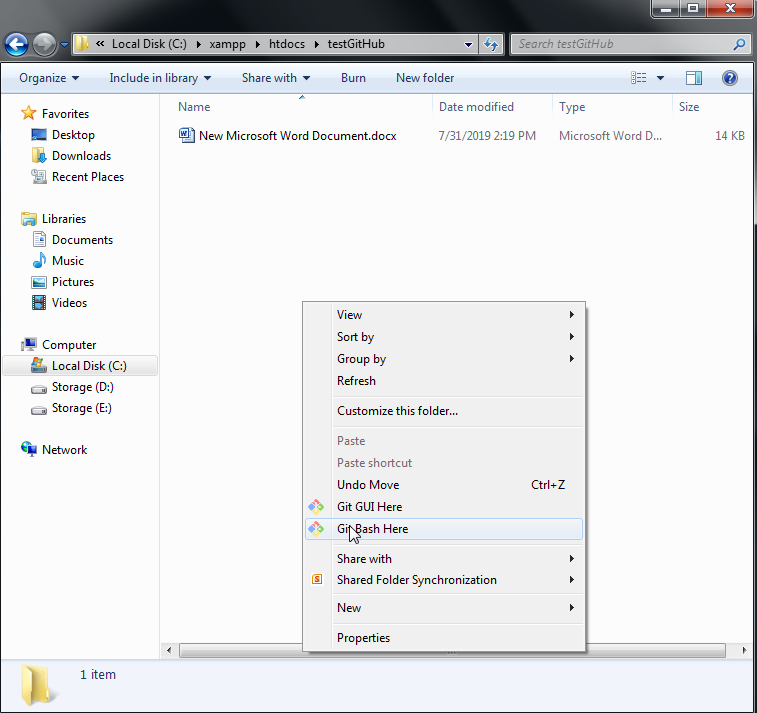
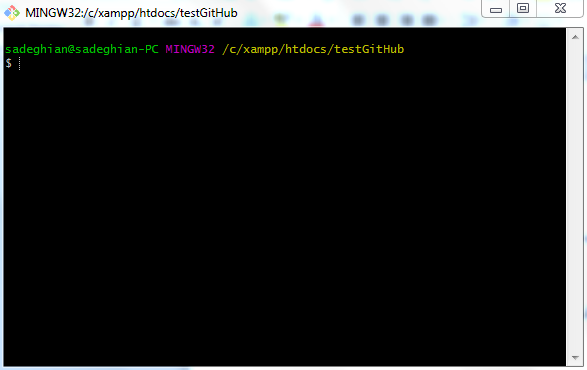
**نصب و راه‌اندازی گیت**

در ابتدا از اینجا : <https://git-scm.com/download> گیت مورد نیاز خود را برای سیستم عامل موردنظرتان دانلود و نصب کنید.



در مرحله دوم به <https://github.com/> بروید و عضو سایت شوید و آدرس ایمیل و رمزعبور خود را فراموش نکنید.

سپس پوشه پروژه موردنظر خود را باز کنید و توی پوشه کلیک راست کنید. گزینه Git Bash Here را انتخاب کنید.

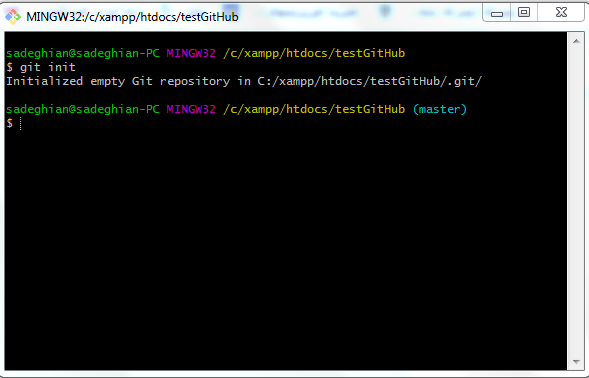
پنجره‌ای باز می‌شود همانند Cmd که می‌توان دستورات گیت را در آن وارد کرد. به این محیط، خط فرمان گیت (Git Command Line) گویند.

در این پنجره خود را با کدهای زیر به گیت معرفی می‌کنید:

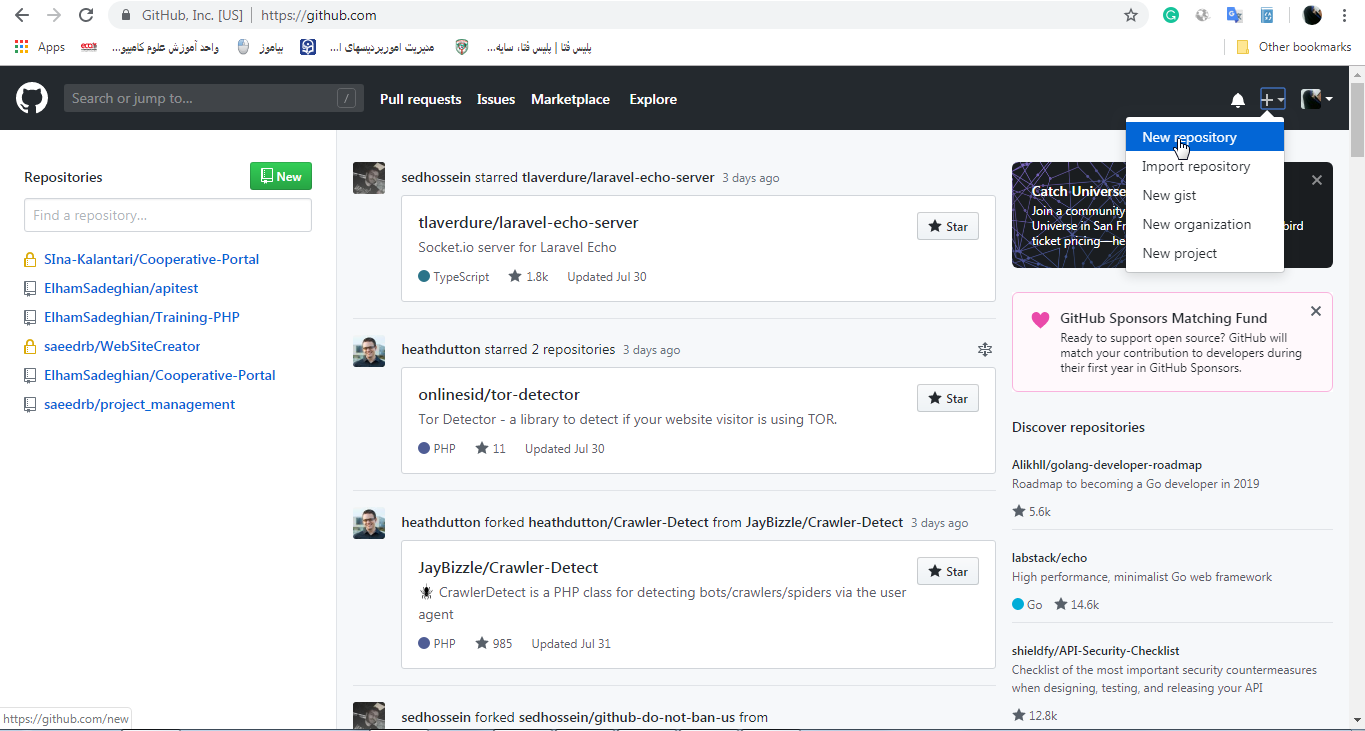
git config --global user.name "Your Name Here"

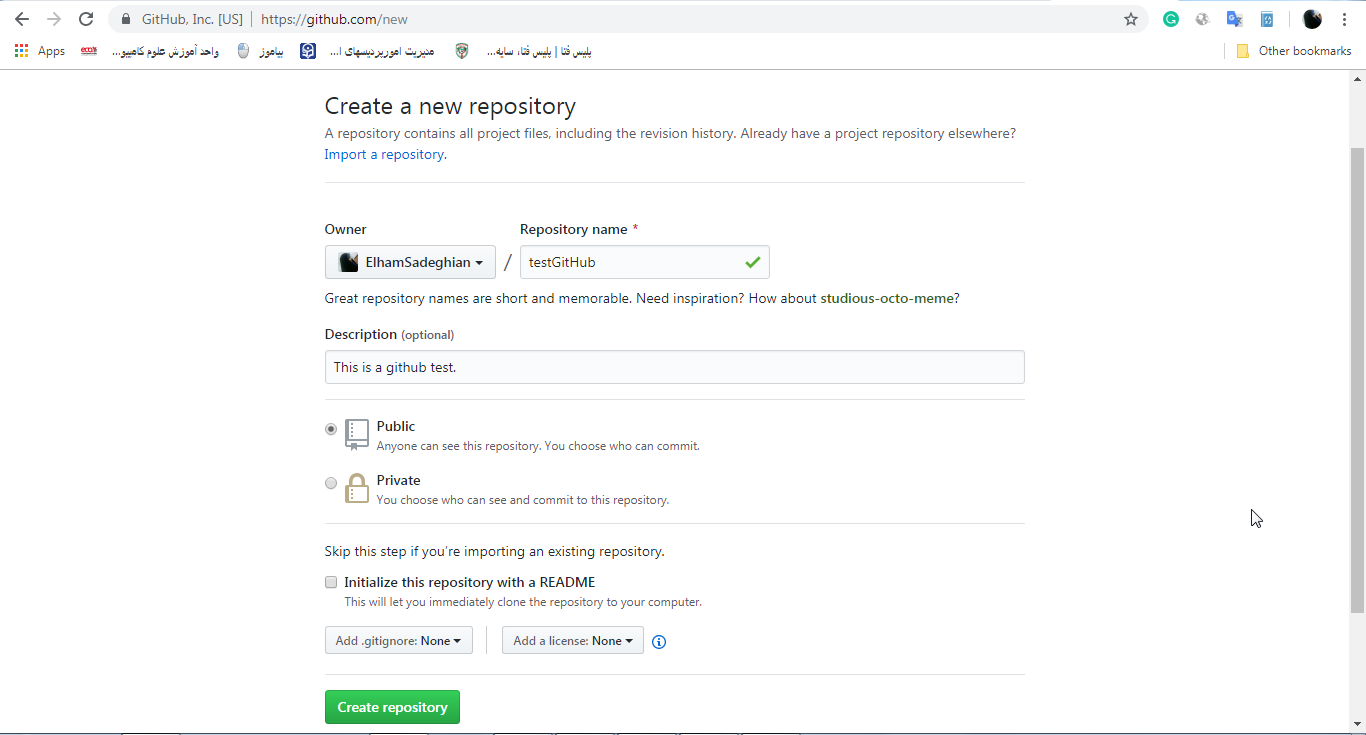
git config --global user.email "[your\_email@youremail.com](mailto:your_email@youremail.com)"

به جای عبارت‌های داخل " " به ترتیب اسمی که در گیت تعریف کردید و آدرس ایمیلی که با آن عضو گیت شدید را وارد می‌کنید.

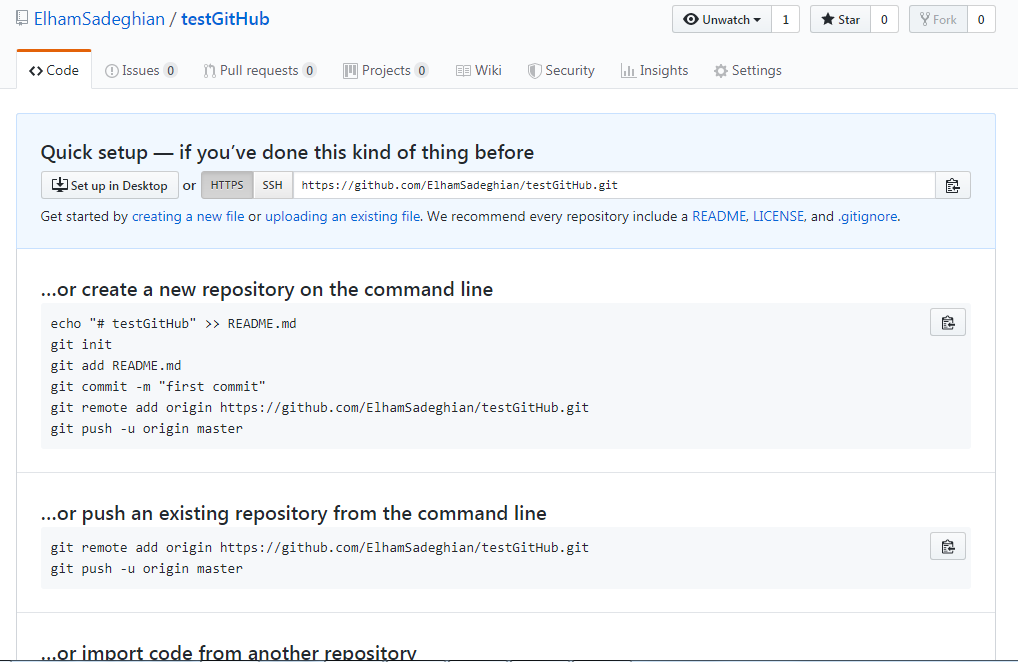
ابتدا git init را می‌نویسیم و enter می‌کنیم. با انجام این کار، راه اندازی اولیه گیت را انجام داده‌ایم و فولدری به نام .git (بصورت hidden) در فولدر اصلی پروژه ساخته می‌شود.

بعد از همه این مراحل به گیت‌هاب برمیگردید و در آنجا از سمت راست صفحه در بالا روی دکمه **+** یک مخزن Repo ایجاد می‌کنید.



در صفحه جدید یک اسم برای مخزن انتخاب می‌کنید. بخش توضیحات اختیاریست و می‌تواند خالی باشد. گزینه Initialize this repository with a README یک فایل ایجاد می‌کند تا اطلاعات پروژه خود را به طور اختصار در آن بنویسید و فعال کردن تیک آن اجباری نیست.

بعد از زدن دکمه create repository وارد صفحه جدید می‌شوید.



در اینجا چون قبلا ریپازیتوری را ایجاد کردید از کدهای بخش دوم استفاده می‌کنید.

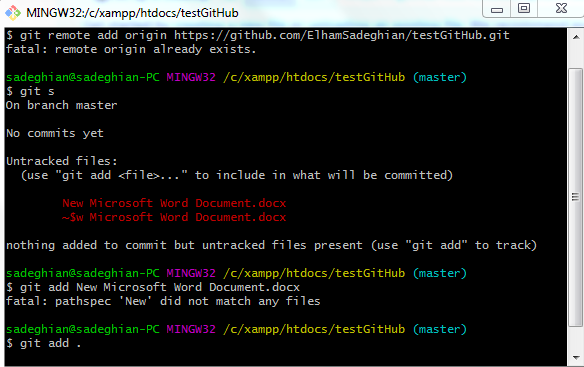
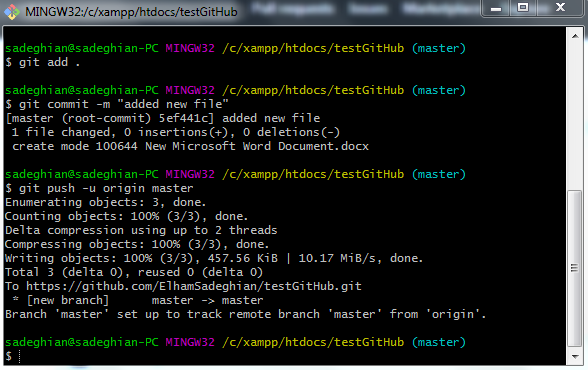
git remote add origin <https://github.com/ElhamSadeghian/testGitHub.git>

git push -u origin master

با این کد، از command line یک ریپازیتوری موجود را push می‌کنید.

ابتدا کد اول را وارد می‌کنید سپس با git status آخرین تغییرات را مشاهده کنید و با git add فایل‌های جدید و تغییر یافته را ادد کنید. برای اینکار هم می‌توان از git add filename استفاده کنید - که فقط فایل مشخصی را اضافه می‌کند- هم از git add . استفاده کنید که همه فایل های موجود را اضافه می‌کند. سپس با git commit –m “write description” توضیحات مربوط به تغییرات را اضافه می‌کنید.

در نهایت کد دوم در توضیحات بالا را استفاده می‌کنید تا این تغییرات در گیت ذخیره شود.



**بررسی و اعمال تغییرات پروژه در گیت**

اگر از پروژه خارج شده‌اید، ایتدا با استفاده از کامند لاین به مسیر پروژه موردنظر بازمی‌گردید. گیت می‌داند که یک مخزن آنلاینی وجود داره که شما میخواهید تغییرات محلی به آنجا منتقل بشود. برای اطمینان این دستور را وارد کنید:

git remote –v

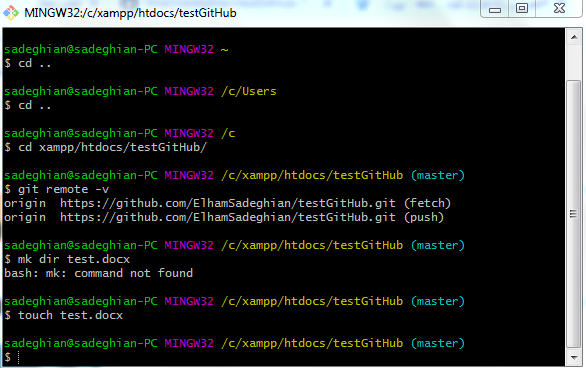
این دستور لیستی از همه‌یorigin هایی که مخزن محلی می‌شناسد را به شما می‌دهد. اگر تا اینجا با من همراه بوده‌اید فقط باید یک testGitHub.git وجود داشته باشد که دوبار لیست شده‌است. که یعنی هم می‌توانید اطلاعات را به آن push کنید و هم می‌توانید اطلاعات را از آن fetch کنید.

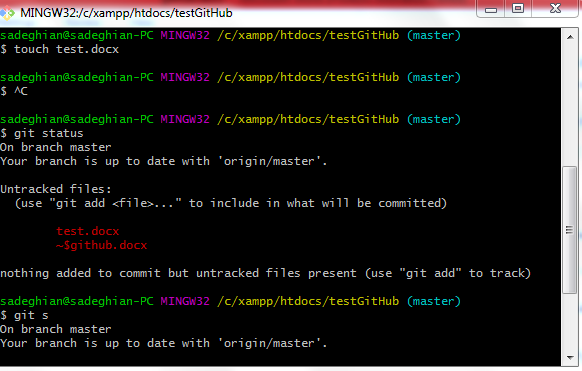
حال فرض کنید قصد اضافه کردن فایل ورد جدیدی را به پروژه دارید. از یکی از دستورهای زیر استفاده می‌کنید تا فایل ایجاد شود:

mk dir test.docx

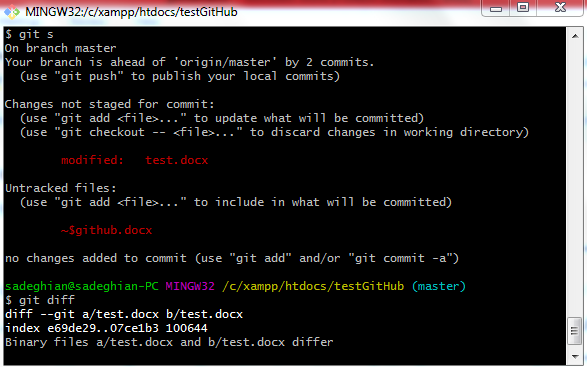
touch test.docx

در سیستم عامل ویندوز دستور دوم قابل استفاده است.



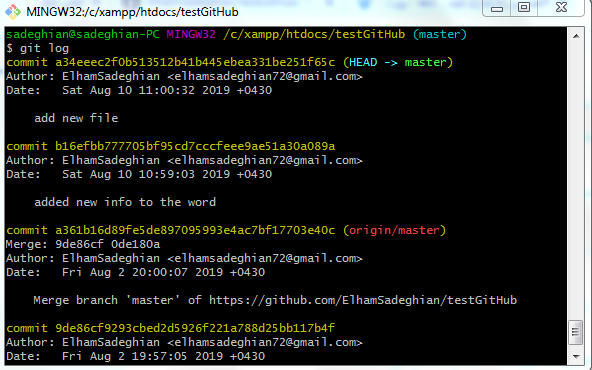
پس از اعمال تغییرات در پروژه قصد افزودن آن به گیت را دارید. پس در ابتدا از git status استفاده می‌کنید تا فایل‌های تغییر یافته یا افزوده شده را مشاهده کنید. در پروژه من، فایل وردی که توضیحات گیت را در آن می‌نویسم و فایل test.docx که حین آموزش اضافه کردیم در تغییرات قابل مشاهده می‌باشد.

برای مشاهده تفاوت بین آخرین commit انجام شده با وضعیت حال حاضر فایل‌های پروژه، ابتدا دستور git status  را اجرا کرده تا وضعیت فعلی را ببینید و سپس دستور git diff را اجرا کنید. با انجام اینکار، فایل‌هایی که دچار تغییر شده‌اند را به همراه تفاوت‌های آن‌ها با وضعیت فعلی آن فایل‌ها را مشاهده خواهید کرد.



با اجرای دستور git diff –help دستورالعمل جامع استفاده از دستور diff برای شما نمایش داده خواهد شد.

برای مشاهده تاریخچه commitهای انجام شده، کافیست دستور git log را اجرا کنید. با انجام اینکار، تمام commitهای انجام شده به ترتیب آخرین کامیت نمایش داده می‌شوند. آیتم‌هایی که در هر لاگ نمایش داده می‌شوند، عبارتند از:

1. آیدی کامیت(Commit ID)که عددی منحصر بفرد و یکتا می‌باشد.
2. نام نویسنده(Author)که آدرس ایمیل کاربر را به همراه نام او نمایش می‌دهد.
3. تاریخ کامیت(Date)که تاریخ و زمان دقیق اجرای دستور commit را اعلام می‌کند.
4. پیغام(message)که تعریف آن برای هر commit الزامی می‌باشد.

در انتها برای خروج از این لاگ کافیست فقط q را وارد کنید.

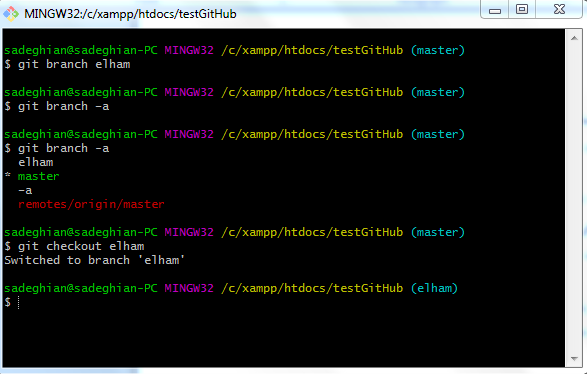
**مفهوم merge در گیت**

فرض کنید یک گروه در حال توسعه یک وب‌سایت می‌باشد و این تیم شامل ۴ نفر می‌باشد. این افراد بر روی ۴ قسمت مجزا کار می‌کنند. هنگامی که کار این ۴ نفر به اتمام برسد، باید این ۴ بخش را با هم ادغام (merge) کنند و یک وب‌سایت را تشکیل دهند.

**استفاده از دستور branch در Git**

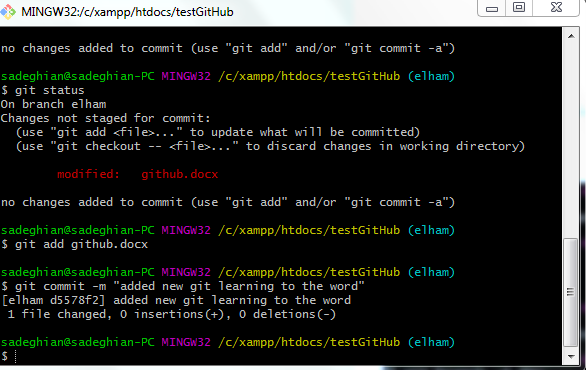
به منظور تعریف یک branch کافیست دستور ساده git branch elham را اجرا کنید که در اینجا elham یک کلمه دلخواه است. یعنی نام اولین branch پروژه elham قرار گرفت.

برای مشاهده لیست تمامbranch ها، دستور git branch –a را اجرا می‌کنید. در هر پروژه‌ای که از گیت بهره می‌برد،branch ی بنام master بصورت پیش فرض وجود دارد که شاخه اصلی هر پروژه می‌باشد. بنابراین در این مثال پس از اجرای دستور git branch –a لیست شاخ‌ها به صورت master و elham نمایش داده‌خواهندشد.

به منظور جابجایی بینbranch های مختلف در یک پروژه، دستور git checkout elham را اجرا کنید. در این دستور، می‌توانید بجای elham نام شاخه دلخواه خود را بنویسید تا به آن branch سوئیچ کنید.

برای اینکه دو عمل ایجاد شاخه و سوئیچ به آن شاخه را به صورت همزمان انجام دهید باید دستور git checkout –b elham  را اجرا کنید. با اجرای این دستور باعث می‌شوید هم شاخهelham  را ایجاد کرده باشید و هم به آن شاخه سوئیچ کرده باشید.

دقت داشته باشید که در صورت اعمال تغییرات در شاخه مثلا ویرایش یک فایل و افزودن دو فایل جدید این تغییرات تنها در همین شاخه قابل مشاهده است. بنابراین اگر به شاخه master سوئیچ کنید، پروژه به حالت قبل بازمی‌گردد. یکی از کاربردهای تعریف branch اینست که شما بخواهید یک نسخه تستی از پروژه را در کنار نسخه اصلی داشته باشید و کدهای جدیدی که می‌خواهید روی پروژه خود تست کنید را در ورژن تستی پروژه اعمال کنید. بدون اینکه نسخه اصلی پروژه شما ضربه‌ای بخورد. پس از اینکه کدهای جدیدتان را در شاخه تستی امتحان کردید و جواب داد، می‌توانید این کدها را به شاخه اصلی پروژه خود اضافه کنید. شما می‌توانید به راحتی با استفاده از دستور merge این دو نسخه را ادغام کنید.

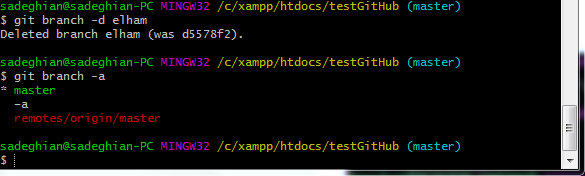
توجه داشته باشید که حتما قبل از استفاده از دستور merge و ادغام نسخه‌ها با یکدیگر، توسط دستور commit، تغییرات انجام شده در شاخه تستی را به repository اضافه کنید. بنابراین با اجرای دستور  git commit –a –m “add changes to test branch”  می‌توان اینکار را انجام داد.

فرض کنید تست کدها موفقیت آمیز بود؛ اکنون می‌خواهید این کدها را نیز در شاخه master داشته باشید. بدین منظور، ابتدا به شاخه master سوئیچ کرده و سپس دستور git merge elham را اجرا کنید. با اجرای این دستور، کدهای جدید و احتمالا فایل‌هایی که در شاخه elham ایجاد کرده‌اید به master منتقل خواهند شد.

برای اینکه مطمئن شوید این انتقال به درستی صورت گرفته‌است می‌توانید کد git log را در شاخه master اجرا کنید و لاگ‌های آن را مشاهده کنید. برای مشاهده لاگ‌های انجام شده به صورت گراف، کافیست دستور git log –graph  را اجرا کنید.



**حذف شاخه در گیت**

برای پاک کردن شاخه elham باید ابتدا از آن branch خارج شوید، سپس آن را توسط دستور git branch –d dev حذف کنید.