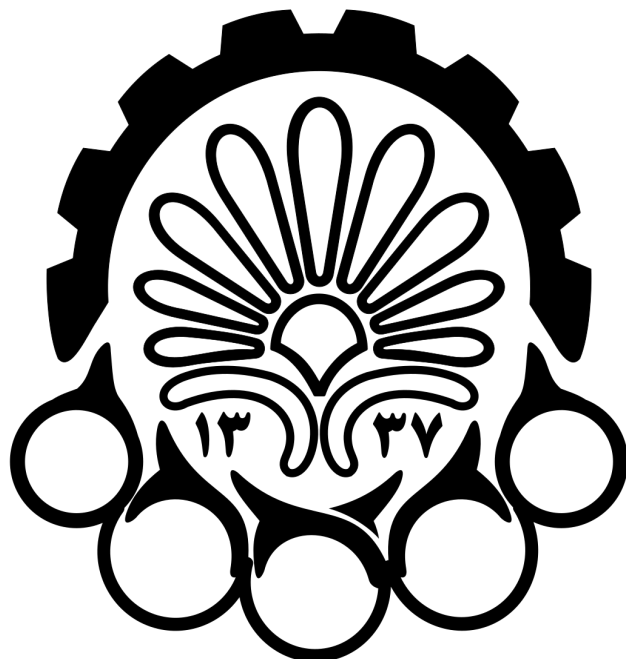


به نام خدا



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

تمرین اول درس مهندسی اینترنت

پاییز ۱۴۰۰ - مهندس الوانی

پیش از انجام این تمرین ابتدا مطمئن شوید که برنامه git را در سیستم خود نصب دارید. برای آموزش نصب این ابزار میتوانید از [این لینک](#)^۱ استفاده کنید.

همچنین در وبسایت [Github](#) یک حساب کاربری ایجاد کنید و مشخصات حساب خود را در git وارد کنید. برای اینکار میتوانید از دستورات زیر استفاده کنید:

```
git config --global user.name "Your name"
git config --global user.email "your@email.com"
```

برای یادگیری دقیق تر و آشنایی با چالش‌های رایج، هنگام کار کردن به صورت فردی و یا گروهی روی پروژه‌ها با گیت، می‌توانید از چالش‌هایی که در این ۲ سایت قرار داده شده استفاده کنید.

این چالش‌های بخشی از تمرین نیستند ولی حل آنها برای استفاده راحت تر و مفیدتر از گیت توصیه می‌شود.

- [learngitbranching](#)^۲
- [katacoda git course](#)^۳

^۱ <https://www.atlassian.com/git/tutorials/install-git>

^۲ <https://learngitbranching.js.org/>

^۳ <https://www.katacoda.com/courses/git>

سوال اول

۱. یکی از پروژه‌های خود را به صورت عمومی داخل اکانت گیت‌هاب منتشر کنید.

۲. با توجه به زبان و تکنولوژی‌های مورد استفاده، برای پروژه خود فایل `gitignore` مناسب بنویسید. یکی از فایل‌های مهم هنگام استفاده از گیت فایل `gitignore` است. بعضی فایل‌ها از جمله تنظیمات IDE، اطلاعات مهم، فایل‌هایی که به صورت خودکار تولید شده‌اند و ... نباید داخل مخزن عمومی قرار بگیرند. فایل گیت‌ایگنور برای جلوگیری از اضافه شدن آن‌ها در زمان استفاده از دستور «git add» استفاده می‌شود.

برای اطلاعات بیشتر و نحوه استفاده از `gitignore` به [این لینک](#)^۴ مراجعه کنید.

۳. برای پروژه خود فایل LICENSE اضافه کنید. برای این کار می‌توانید [از این وبسایت](#)^۵ کمک بگیرید. نرم افزارها و کدهای متن باز هنگام انتشار عمومی باید قوانینی مربوط به استفاده، ویرایش، انتشار و ... را مشخص کنند. این کار را می‌توان با انتخاب یکی از LICENSE‌های از پیش تعیین شده انجام داد.

۴. برای پروژه خود فایل README همراه با توضیحات لازم اضافه کنید. (برای قسمت اصلی خلاصه‌ای مختصر از پروژه و عملکردش کافی است) این فایل در صفحه اول پروژه نمایش داده می‌شود و حاوی اطلاعات قابل خواندن درمورد پروژه است. برای اطلاعات بیشتر درمورد README و زبان Markdown می‌توانید از منابع زیر کمک بگیرید:

- [About READMEs](#)
- [Best README Template](#)
- [Mastering Markdown](#)

موارد امتیازی

- نوشتن README کامل‌تر (برای این منظور می‌توانید از منبع `Best README Template` استفاده کنید)
- نوشتن `gitignore` هدفمند متناسب با محتوی پروژه (زبان برنامه‌نویسی و ابزارهای استفاده شده)
- ساختن یک Github Page برای پروژه خود. برای این کار از [این لینک](#)^۶ و قسمت Project site استفاده کنید.
- برای اکانت شخصی خود یک صفحه توضیحات بسازید مانند [این نمونه](#)^۷.

^۴ <https://www.atlassian.com/git/tutorials/saving-changes/gitignore>

^۵ <https://choosealicense.com>

^۶ <https://pages.github.com>

^۷ <https://github.com/aut-ce>

سوال دوم

در این فعالیت قصد داریم نحوه مشارکت در یک پروژه عمومی را شبیه سازی کنیم. به همین منظور، یک ریپازیتوری عمومی در [این لینک](#)^۸ ساخته شده است. پس از وارد شدن، فعالیت‌های خواسته شده را انجام دهید و سرانجام تغییرات خود را در قالب pull request ثبت کنید.

سوال سوم

در این فعالیت قصد داریم دستورات اولیه گیت را به کار ببریم. ابتدا فایل FileA.txt را ایجاد کنید و آن را با پیام مناسب کامیت کنید. سپس جمله‌ای به دلخواه به آن اضافه کنید و آن را کامیت کنید. توجه داشته باشید که کامیت جدیدی اضافه نکنید و تغییرات جدید به کامیت قبلی اضافه شود. سپس تغییرات را به ریپازیتوری خود پوش کنید.

در قدم بعدی فایلی با پسوند env. بسازید و آن را پوش کنید. میدانیم که در این نوع فایل‌ها ممکن است داده‌های حساس وجود داشته باشد. پس سعی کنید آن را پاک کنید. میدانیم که یک راه این است که کد در ریپازیتوری را برداریم این فایل را پاک کنیم و دوباره پوش کنیم. اما این کار فایل را به طور کامل از گیت پاک نمی‌کند و در history گیت میتوان آن را پیدا کرد. پس با دستور مناسب سعی کنید فایل را از ریپازیتوری گیت خود پاک کنید.

پس از انجام دادن قدم‌های بالا دو فایل دیگر به نام‌های FileB.txt و FileC.txt با محتویات دلخواه ایجاد کنید. فرض کنید تغییرات روی این فایل‌ها تمام نشده است و قابلیت پوش شدن روی ریپازیتوری اصلی را هنوز ندارد ولی شما می‌خواهید این فایل‌ها را با همگروهی خود به اشتراک بگذارید تا بتواند تغییراتی روی آن اعمال کند و سپس روی ریپازیتوری اصلی پوش کند. با انجام دستورات مناسب این شرایط را برای فایل‌ها اعمال کنید.

برنج جدیدی را ایجاد کنید و ۳ کامیت جدید اعمال کنید و آن را پوش کنید. حال به برنج اصلی بروید و از بین ۳ کامیت ایجاد شده تنها کامیت دوم را مرج کنید. در نهایت کامیت‌های بعدی را همه باهم مرج کنید. (راهنمایی: در مورد دستور git rebase تحقیق کنید)

^۸ <https://github.com/aut-ce/git-assignment>

نحوه تحویل تمرین

آدرس ریپازیتوری‌های خود در سوال اول و سوم را در قالب یک فایل pdf با نام «IE_HW1_StudentID.pdf» در صفحه کورسز درس آپلود کنید.

در نهایت هر گونه سوال در مورد تمرین و بخش‌های آن را از طریق ایمیل ie.ce.aut@gmail.com مطرح فرمایید.
تندرست و موفق باشید