

آزمایشگاه مهندسی نرم افزار

آزمایش اول

علی کریملو ۹۷۱۱۰۳۲۵

رضا شکری ۹۷۱۰۶۰۵۷

۱. پوشه‌ی `git`. یک پوشه‌ی مخفی است که تمام اطلاعات مربوط به گیت برای مدیریت نسخه‌ی کد در آن ذخیره می‌شود.

این پوشه شامل فایل‌ها و دایرکتوری‌های زیر است:

HEAD - این فایل به شاخه‌ی فعلی اشاره می‌کند.

index - این فایل شامل اطلاعاتی در مورد تغییراتی است که در حال حاضر به ریپوزیتوری افزوده شده‌اند (اطلاعات درج شده در `staging area`).

objects - در این دایرکتوری، تمام اشیاء `(blob)`، `tree`، `commit` که تاریخچه‌ی کد شما در آن‌ها می‌تواند ذخیره شود، قرار دارند.

refs - این دایرکتوری شامل رمزهایی است که برای اشاره دادن به `commit` ها به کار می‌روند.

config - این فایل شامل پیکربندی گیت برای این ریپوزیتوری است.

برای ساخت پوشه‌ی `git`، لازم است یک ریپوزیتوری جدید با دستور `git init` ایجاد شود. از آنجا که پوشه‌ی `git`، فقط در داخل پوشه‌ی ریپوزیتوری وجود دارد، در صورت پاک کردن آن، تمام تغییرات ذخیره‌شده‌ی قبلی از دست خواهند رفت.

۲. عبارت `atomic commit` به معنای این است که یک `commit` به صورت اتمی و در یک فرآیند تنها یک تغییر را در بر دارد و هیچ تغییری را نه در خود `commit` و نه در `working tree` نگه نمی‌دارد. بنابراین در صورت اجرای ناموفق، هیچ تغییری در سیستم نگه داشته نمی‌شود.

در مورد `atomic pull-request`، به معنای این است که تمام تغییرات مربوط به درخواست `pull-request` به صورت اتمی و یکجا بررسی و تصدیق شده‌اند قبل از اینکه درخواست مورد نظر برای ادغام در مخزن اصلی قرار گیرد. این به معنای این است که در صورتی که تغییری تأثیرگذار بر دیگر بخش‌های سیستم باشند، آنها به صورت همزمان بررسی و تصدیق می‌شوند و در صورتی که هیچ کدام از آنها تصدیق نشوند، هیچ تغییری به مخزن اصلی اضافه نخواهد شد.

۳. دستورهای `fetch`، `pull` و `merge`، هر سه به دریافت تغییرات از دیگران در یک مخزن `Git` اشاره دارند، اما با دستورات مختلف به کار می‌روند.

fetch: تغییرات از مخازن اصلی (`remote repositories`) به مخزن شما دریافت می‌شود، اما تغییرات به صورت خودکار در مخزن شما اعمال نمی‌شوند. برای اعمال تغییرات، باید از دستور `merge` یا `rebase` استفاده کرد.

pull: تغییرات از مخازن اصلی به مخزن شما دریافت می‌شود و به صورت خودکار در مخزن شما اعمال می‌شود. دستور `pull`، همانند دستور `fetch` نیست و در واقع ترکیبی از دستورهای `fetch` و `merge` است.

merge: با استفاده از این دستور، تغییری که در شاخه‌های مختلف از یک مخزن اتفاق افتاده‌اند، با هم ترکیب می‌شوند. در صورت اضافه شدن تغییرات در دو شاخه مختلف، دستور `merge` باعث ترکیب این تغییرات می‌شود.

۴. Reset:

وقتی دستور reset برای کامیت‌ها در گیت استفاده می‌شود، به معنی این است که با انتخاب یک کامیت خاص، head به آن کامیت منتقل می‌شود و تاریخچه تغییرات بعد از آن کامیت پاک می‌شود. به این ترتیب، می‌توانید به حالت قبلی کد خود بازگشت کنید.

Revert:

وقتی دستور revert استفاده می‌شود، یک کامیت جدید ایجاد می‌شود که تغییرات معکوس شده را به نام کاربر وارد می‌کند. این دستور برای بازگشت به حالت قبلی کد کاربرد دارد ولی تغییرات لحظه‌ای را نیز در تاریخچه ذخیره می‌کند.

Rebase:

با استفاده از این دستور تاریخچه تغییرات ما بر روی شاخه‌ی دیگری اعمال می‌شود. به معنی این است که تاریخچه‌ی تغییرات ما روی شاخه‌ی دیگری (به عنوان مثال، شاخه‌ی master) قرار می‌گیرد. این دستور برای ترکیب تاریخچه‌ی کد ایجاد شده توسط دو شاخه مختلف استفاده می‌شود.

Restore:

دستور restore برای برگرداندن فایل‌ها به حالت قبلی استفاده می‌شود. این دستور فایل‌های تغییر یافته را به حالت خودشان در گذشته بازمی‌گرداند. این دستور تنها برای فایل‌هایی که در گیت ثبت شده‌اند قابل استفاده است.

۵. Stage مرحله‌ای است که قبل از commit انجام می‌شود و تغییراتی که در فایل‌هایی ایجاد شده‌اند و می‌خواهید در commit به آن‌ها اشاره کنید، در این مرحله به آن‌ها اضافه می‌شوند. به عبارت دیگر، در این مرحله با انتخاب تغییراتی که می‌خواهید در commit شامل شود، آن‌ها را برای commit آماده می‌کنید.

دستور stash در git، به شما اجازه می‌دهد تا تغییراتی که انجام داده‌اید را برای استفاده در آینده به صورت موقت ذخیره کنید. به عنوان مثال، اگر بخواهید از یک برنج به برنج دیگری منتقل شوید، تغییراتی که در برنج فعلی انجام داده‌اید را ممکن است نخواهید خواست که در برنج جدید اعمال شود. از stash برای ذخیره تغییرات و پاک کردن آن‌ها از برنامه‌های فعلی استفاده می‌شود.

۶. Snapshots به تصویربرداری از وضعیت فعلی ریپازیتوری در لحظه‌ای خاص گفته می‌شود. به طور کلی، snapshot یک تصویر از همه تغییراتی است که در داده‌ها یا کدهای موجود در ریپازیتوری صورت گرفته است. بنابراین، با استفاده از snapshot می‌توانید در هر زمان به خاطر کامیت‌های قبلی رجوع کنید و تغییرات را با نسخه‌های قبلی مقایسه کنید.