Version Control System (VCS) =>سیستم کنترل ورژن

 : VCSوظیفه مدیریت و ردیابی تغییرات در کدهای پروژه را برعهده دارد.

GitHub: مکانی است که در آن، تمامی کارهای مدیران پروژه و برنامه‌نویسان ثبت و ذخیره می‌شوند و در‌نهایت، عملیات‌های پروژه کاملاً واضح باقی می‌مانند.

کارایی :Gitگیت از سیستم توزیع خوشه‌ای یا درختی استفاده می‌کند. برخلاف برخی از نرم‌افزارهای کنترل ورژن، Git هنگام ذخیره تاریخچه و نسخه‌های فایل، نام فایل‌ها را در نظر نمی‌گیرد و روی محتوای فایل متمرکز است. همچنین، مخزن گیت از ترکیب رمزگذاری دلتا برای ذخیره تفاوت‌های کد استفاده و درادامه با فشرده‌سازی، فایل‌ها را کاملاً ذخیره می‌کند.

امنیت :Gitدر مخزن Git با الگوریتم هش رمزنگاری امنی به نام SHA1 ذخیره می‌شوند. گیت کدهای شما را بدون هیچ تغییری بازیابی می‌کند و دراختیارتان قرار می‌دهد. با Git ، می‌توانید مطمئن باشید که تاریخچه معتبری از سورس خود دارید.

## : Git index

محل مرحله‌بندی بین فهرست فایل‌های سیستم شما و مخزن گیت است. سه نقطه در Git وجود دارد که فایل‌های تغییر‌یافته شما در آن‌ها قرار می‌گیرند: ۱. دایرکتوری فایل‌های سیستم شما (Working Directory)؛ ۲. منطقه مرحله‌بندی (Staging Area)؛ ۳. مخزن .(Repository)

: Working directory وقتی پروژه‌تان را در سیستم ذخیره کرده‌اید و روی آن کار می‌کنید، با فهرست کاری پروژه خود سروکار دارید. این فهرست همان فایل‌های کامپیوتر شماست. تمام تغییراتی که اعمال می‌کنید، تا زمانی که آن‌ها را به قسمت مرحله‌بندی اضافه نکنید، در فهرست کاری باقی می‌مانند.

: Staging areaمنطقه مرحله‌بندی را می‌توان به‌عنوان منطقه‌ای در نظر گرفت که Git تغییرات را در آن ذخیره می‌کند.

(مخزن)Repository :مخزن شامل مجموعه‌ای از فایل‌ها و تاریخچه تغییرات ایجاد‌شده در آن فایل‌هاست. هر پروژه می‌تواند یک یا چند مخزن داشته باشد که اطلاعات آن پروژه در آن ذخیره شده است.

|  |  |  |  |
| --- | --- | --- | --- |
| شماره | دستور | Syntax | توضیحات |
| 1 | ایجاد مخزن گیت | git init | این دستور یک مخزن گیت جدید در پوشه فعلی ایجاد می‌کند. |
| 2 | کلون کردن یک مخزن | git clone <repository\_url> | این دستور یک نسخه از مخزن گیت از آدرس مشخص‌شده به سیستم شما دانلود می‌کند. |
| 3 | مشاهده وضعیت مخزن | git status | این دستور وضعیت فایل‌ها و تغییرات در مخزن را نشان می‌دهد؛ از جمله فایل‌های تغییر یافته، فایل‌های آماده برای **commit** و فایل‌های جدید. |
| 4 | افزودن تغییرات به stage | git add <file\_name>  git add. | این دستور فایل‌های تغییر یافته را برای **commit** آماده می‌کند. با استفاده از git add . می‌توانید همه فایل‌های تغییر یافته را به مرحله **stage** اضافه کنید. |
| 5 | ثبت تغییرات (commit) | git commit -m "توضیح تغییرات" | این دستور تغییرات را با یک پیام توضیحی ثبت می‌کند. پیام باید کوتاه و گویای تغییرات انجام‌شده باشد. |
| 6 | ارسال تغییرات به مخزن محلی | git push origin <branch\_name> | این دستور تغییرات را به شاخه (Branch) مشخص‌شده در مخزن اصلی (Remote) ارسال می‌کند. معمولاً branch\_name برابر main یا master است. |
| 7 | دریافت آخرین تغییرات | git pull | این دستور آخرین تغییرات از مخزن اصلی را دریافت و با کدهای محلی ادغام می‌کند. |
| 8 | مشاهده تاریخچه تغییرات | git log | این دستور تاریخچه تمامی **commit** ها را نشان می‌دهد. می‌توانید با جزئیات زمان و پیام‌های توضیحی هر commit را مشاهده کنید. |
| 9 | ایجاد شاخه جدید (Branch) | git branch <branch\_name> | این دستور یک شاخه جدید با نام مشخص‌شده ایجاد می‌کند. |
| 10 | تغییر به شاخه دیگر | git checkout <branch\_name> | با استفاده از این دستور می‌توانید به شاخه مورد نظر سوئیچ کنید. |
| 11 | ادغام شاخه‌ها (Merge) | git merge <branch\_name> | این دستور شاخه مورد نظر را با شاخه فعلی ادغام می‌کند. معمولاً برای ادغام تغییرات از شاخه‌های فرعی به شاخه اصلی استفاده می‌شود. |
| 12 | حذف شاخه | git branch -d <branch\_name> | این دستور شاخه مورد نظر را حذف می‌کند. از این دستور پس از ادغام تغییرات شاخه استفاده می‌شود. |
| 13 | نادیده گرفتن تغییرات | git reset –hard | این دستور تمام تغییرات محلی را نادیده گرفته و پروژه را به آخرین commit بازمی‌گرداند. |
| 14 | ایجاد تگ (Tag) | git tag <tag\_name> | این دستور یک تگ با نام مشخص‌شده ایجاد می‌کند که می‌تواند برای نشانه‌گذاری نسخه‌های خاصی از پروژه استفاده شود (مثلاً نسخه 1.0.0). |
| 15 | بررسی تفاوت‌ها | git diff | این دستور تفاوت بین فایل‌های تغییر یافته و آخرین commit را نشان می‌دهد. |
| 16 | حذف فایل از stage | git reset <file\_name> | این دستور فایل مشخص‌شده را از مرحله **stage** خارج می‌کند، ولی فایل محلی تغییرکرده باقی می‌ماند. |
| 17 | مشاهده شاخه‌های موجود | git branch | این دستور لیستی از شاخه‌های موجود در پروژه را نمایش می‌دهد. |
| 18 | کلون کردن فقط یک شاخه | git clone -b <branch\_name> <repository\_url> | این دستور یک شاخه مشخص را از مخزن گیت کلون می‌کند. |