

توضیحات خلاصه مربوط به گیت

گیت ابزار کنترل سیستم پروژه هستش و میتون ورژن های مختلف پروژه رو با وسیله ی آن نگهداری کرد که این ویژگی باعث میشود که به راحتی ورژن های مختلف رو باهم مقایسه کرد یا فهمید چه کسی روی کدوم قسمت کد چه تغییراتی ایجاد کرده یا میتوان هر بلایی سر کد بدون نگرانی به هم ریختن کد آورد. خاصیت `codereviwe` که هر شخصی میتواند کد شمارو مشاهده و تغییراتی در اون ایجاد میکنند.

برخی از دستورات:

• **Git init**

مشخص میکنیم از این دایرکتوری به بعد مربوط به پروژه من هست و یک ریپازیتوری یا یک مخزن لوکال به پروژه اضافه میکند و فایل وارد `working directory` میشود.

• **Git add -A & Git add 'name of file'**

برای افزودن فایل به یک اتاق انتظار که با هم جمع بشن و منتظر کامیت کردن بمونن و در `staging area` میروند

• **Git clone 'http...'**

ا درس ریپازیتوری را میتوان از گیت هاب کپی کرد و با این دستور پروژه مد نظر به همراه تمام فایل ها اضافه یا کپی میشن. در واقع دستور اول `git init` انجام شده.

• **Git status**

از دستور `git status` برای یافتن اطلاعات فایل هایی که ادیت شده اند و یافتن فایل هایی که در `staging area` قرار دارند استفاده کنید. این دستور اطلاعات دیگری را هم نمایش می دهد برنچ ها و...

• **Git commit -m 'discription of difference'**

میتوان از `staging area` چند تا فایل جمع شده را با این دستور بعنوان یک کامیت یا اسنپ شات ذخیره میکنیم.

• Git log

از دستور git log برای نمایش تمامی commit های انجام شده تا حال استفاده کنید. این دستور نشان می دهد چه کسی دستور را وارد کرده، در چه تاریخی و با چه پیامی آن commit را انجام داده است.

• Git push

برای push کردن تمام کدها از روی مخزن لوکال به مخزن ریموت سیستم ما از دستور زیر استفاده کنید:

git push -u origin master

• Git pull

از git pull برای pull کردن یا دریافت آخرین تغییرات از روی مخزن ریموت به مخزن لوکال استفاده می شود. کدهای مخزن ریموت توسط توسعه دهندگان متفاوت در هر لحظه در حال آپدیت شدن است، از این رو git pull بسیار واجب است.

git pull origin master

• Git branch test

برای کار کردن چند شخص متفاوت برای یک پروژه از شاخه اصلی میتوان چند برنج بگیریم و روی پروژه تغییراتی رو ایجاد کنیم

• Git merge test

اگر تغییرات روی برنج تست مورد قبول مدیر پروژه باشه اون رو مرج میکنه و به پروژه اضافه میکنه.

• Git tag -a v number

برای تگ زدن ورژن های مختلف پروژه وبعد ها میتوان به راحتی ورژن های مختلف برنامه روبه راحتی مشاهده کرد.

• Git blame name pf file -l number , number

برای پیدا کردن تغییرات کامل کد حتی توسط چه شخصی

Git bisect هم برای پیدا کردن باگه

• Git checkout

برای بررسی اینکه کدام برنچ هستیم یا به کدام برنچ وارد بشیم.

• Git rm name of file

برای حذف یک فایل از لوکال و ریموت استفاده میشود.