

# Alireza Talei

## Git

گیت یکی از قدرتمندترین نرم افزارهای ورژن کنترل برای ردیابی تغییرات فایل های کامپیوتری و هماهنگ سازی کار بر روی این فایل ها هنگامی که چندین نفر روی آنها کار می کنند.

Git در سال ۲۰۰۵ توسط لینوس توروالدز ایجاد شد تا کار بر روی هسته لیناکس و توسعه آن راحت تر باشد.

Git توسط زبانهای برنامه نویسی C, Shell, Perl, Tcl, Python نوشته شده است.

## Version Control

برنامه ای برای مدیریت تغییرات اسناد، برنامه های کامپیوتری، وب سایتهای بزرگ و دیگر مجموعه های اطلاعات می باشد.

## Distributed Version Control

در بحث توسعه نرم افزار ها کنترل نسخه توزیع شده نیز به نوعی Version Control می باشد که کدبیس کامل، شامل تاریخچه کامل تمامی تغییرات و نسخ پیشین منتشر شده آن نرم افزار و تفاوت هر نسخه می باشد. و در نرم افزار های متن باز اجازه ایجاد شاخه جدید و یا درهم تنیدن دو شاخه مجزا فراهم می گردد.

این روش باعث افزایش سرعت کار گشته و توانایی کار به صورت آفلاین بر روی پروژه را داده و علاوه بر آن تاکید و تکیه به یک سرور یا کامپیوتر خاص برای ایجاد بک آپ را منتفی می نماید.

## Github

شرکت Github یک شرکت مبتنی بر وب ارائه خدمات میزبانی برای سیستم ورژن کنترل Git است. گیت هاب به اشخاص اجازه می دهد تا به صورت خصوصی و با پرداخت هزینه و هم به صورت عمومی و رایگان از خدمات آن استفاده نمایند که عمده آن تمامی پروژه های متن باز به همین روش بر روی گیت هاب منتشر می شوند.

در حال حاضر Github به بیش از ۲۸ میلیون نفر ارائه سرویس می دهد و دارا بودن حدود ۵۷ میلیون مخزن (repository) که ۲۸ میلیون آنها عمومی می باشند تبدیل به بزرگترین میزبان متن باز در جهان شده است.

حدود هفت ماه پیش (۴ جون ۲۰۱۸) شرکت مایکروسافت با پرداخت ۵.۷ میلیارد دلار این سرویس را خریداری نمود.

نصب و اجرای گیت

با مراجعه به سایت <https://git-scm.com> و دانلود نسخه هماهنگ با سیستم عامل شما.

# Alireza Talei

## دستورات اولیه برای شروع کار با گیت

master	شاخه اصلی پروژه
- git init	ایجاد یک پروژه گیت
- git status	مشاهده وضعیت پروژه
- git commit -m '...'	ذخیره تغییرات به همراه پیام جهت بازبینی ها
- git add [file name] or [*.*] or [-A]	به حالت استیج بردن (یک فایل) یا (کلیه فایلها)
- git log	مشاهده پیام ها (کامیت مسیج ها)
HEAD -> master	فایلی که روی آن مشغول کار هستید در شاخه اصلی (master) (آخرین کامیت شما)
- git diff HEAD	مشاهده اختلاف فایلی که مشغول به کار روی آن هستید با وضعیت قبلی آن (مشاهده تغییرات فایل)
- git reset [file name]	بیرون آوردن یک فایل از حالت استیج
- git diff - -staged	مشاهده اختلاف فایلی که استیج شده (مشاهده تغییرات فایل) {همان دستور git diff برای فایل استیج شد}
- git checkout - - [file name]	برگرداندن یک فایل به کامیت قبلی (تأیید گرفتن آخرین تغییرات انجام شده)
- git branch	مشاهده تمام شاخه های موجود (و در کدام شاخه هستیم با علامت *)
- git branch [branch name]	ایجاد یک شاخه جدید
- git checkout [branch name]	بیرون آوردن شاخه {رفتن به شاخه از شاخه ای دیگر}
- git merge [branch name]	یکپارچه کردن (مرج کردن) دو شاخه با هم
- git rm [file name]	پاک کردن یک فایل (دیلیت کردن فایل)
- git branch -d [branch name]	پاک کردن یک شاخه (دیلیت کردن یک شاخه)
repository	مخزن (یک پروژه روی گیت)
- git clone [url]	ایجاد یک کپی از یک پروژه گیت
origin	پروژه روی گیت که از آن کپی گرفته ایم
- git push origin master	ارسال تغییرات از لوکال (مستر) به اوریجین
- git pull origin master	دریافت تغییرات از اوریجین به لوکال (مستر)
- git remote add origin [url] & then git push -u origin master	به آدرس گیت هاب این پروژه را اضافه کن (و سپس گیت پوش)
- ...conflict ...	اختلاف - تفاوت بین اوریجین با مستر {هنگامی که یک فایل هم زمان توسط دو نفر تغییر کرده}
reject	هنگامی که گیت نتواند خودش اختلاف را برطرف کند
- git show [hash number]	نمایش تغییرات مربوط به هش نامبر مربوطه {نمایش تغییرات کامیت}
- git tag	نمایش تمامی تگ های پروژه
- git tag -a v[number] -m '...'	تگ زدن به آخرین کامیت پروژه
- git tag -a v[number] [hash number of old logs] -m '...'	تگ زدن به یک لاگ به خصوص {لاگ های قدیمی تر}
- git show [tag number]	نمایش تگ مربوطه {مشاهده پروژه در آن تگ}
- git push origin [tag number]	اضافه کردن تگ نامبر به اوریجین {اضافه کردن تگ نامبر از لوکال به به اوریجین}
- git push origin - - tags	اضافه کردن کلیه تگ نامبرها به اوریجین {اضافه کردن تگ نامبرها از لوکال به به اوریجین}
- git checkout [tag number]	بیرون آوردن {کپی گرفتن از یک ورژن (تگ) برنامه}
- type [file name]	نمایش محتویات فایل