

آموزش Git :

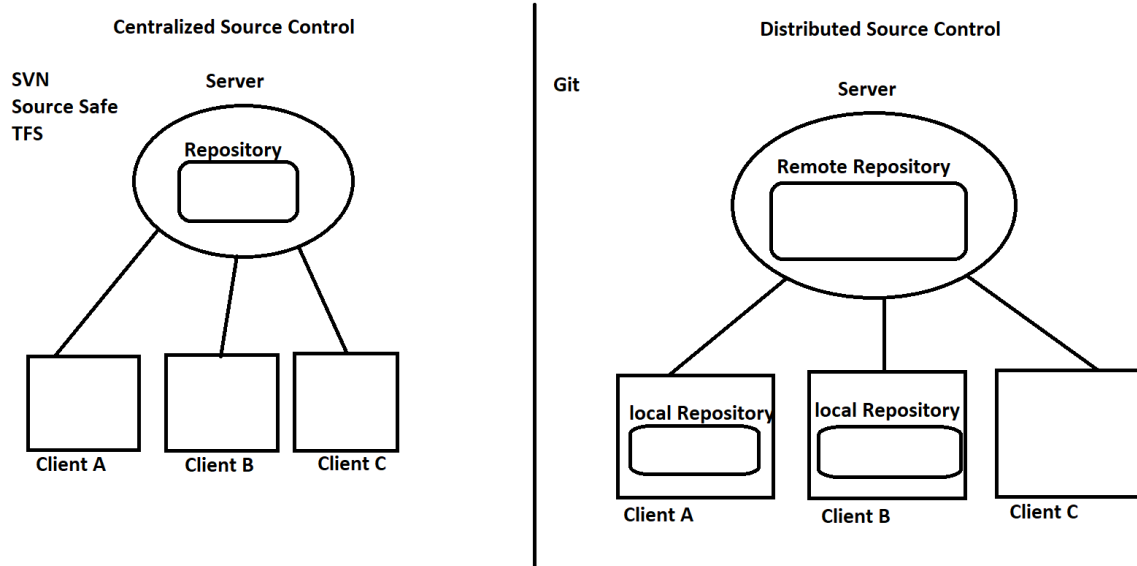
فصل اول :

کاربرد و نقش Source Control :

- اینکه History کارها و تغییرات رو داشته باشیم
- امکان کار تیمی روی یک پروژه
- ورژن بندی پروژه

انواع Source Control :

- **Centralized** : ما یک سرور یا یک کامپیوتر خواهیم که سورس پروژه روی آن قرار می گیرد و همه برنامه نویس ها به آن متصل شده و یک کپی از سورس روی سیستم خود دریافت کرده و روی آن کار می کنند. (سورس پروژه : Remote Repository Code). نمونه source control ها از این نوع:
 - SVN
 - Source Safe
 - Team Foundation Server - TFS
- **Distributed** : در این حالت دو Repository با نام های Remote Repository بر روی سرور و Local Repository بر روی کلاینت.



معرفی Git و ویژگی های آن :

- در سال ۲۰۰۵ شروع شده
- توسط Linus Torvalds ایجاد شده
- یک Distribute Version-Control
- امکان ثبت تغییرات روی local repository : commit
- امکان ارسال تغییرات از local repository بر روی remote repository : push
- دریافت تغییرات از روی remote repository بر روی local repository : pull
- ویژگی branch : یک مسیر جدید برای اعمال تغییرات مورد نظر و سپس این مسیر ایجاد شده را با مسیر اصلی (master) merge می کنیم.
- برای استفاده از گیت نیاز به دیتابیس خاصی نداریم به صورت file base تغییرات ذخیره می شوند.

نصب Git در Windows : دانلود و نصب از روی لینک زیر

<https://git-scm.com/>

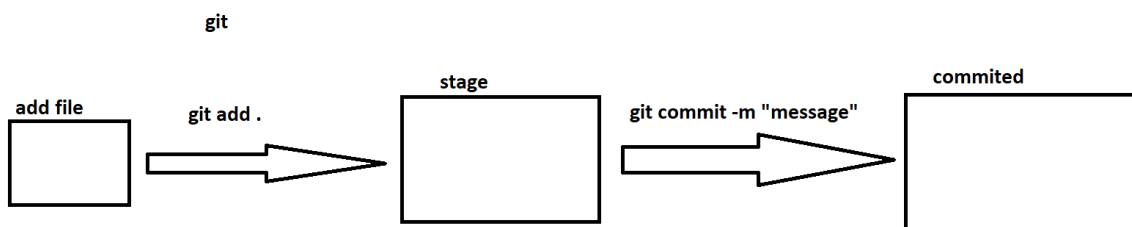
افزودن اولین پروژه به Git : گیت یک فولدر را به عنوان فولدر اصلی پروژه در نظر می گیرد و محتوای داخل آن را رصد می کند.

از طریق روش های زیر :

- از طریق command prompt : در مسیر فولدر مورد نظر باید command prompt را باز کنید. سپس از دستور git init برای افزوده شدن گیت به فولدر مورد نظر استفاده کنید.
- از طریق vs code : از طریق منوی Source Control و Initial Repository در محیط vs code

فصل دوم :

مراحل افزودن فایل به git و نحوه commit کردن :



مراحل کار در Commit کردن یک یا چند تغییر در git :

- ابتدا باید همه تغییرات یا تغییرات مورد نظر را در git با دستور git add اضافه کنیم :
 - git add . : یعنی همه تغییرات اتفاق افتاده در حالت stage اضافه شوند
 - git add file1.txt : فقط تغییرات فایل مورد نظر در حالت stage اضافه شود.
- برای نهایی شدن تغییرات در حالت stage به حالت committed از دستور git commit -m "message" استفاده می کنیم. در local repository

کاربرد حالت stage : یکی از کاربردها stage این است که اگر شما بخواهید تغییرات جاری به طور مثال در ۵ فایل اتفاق افتاده است را در ۲ Commit جداگانه داشته باشی، از امکان stage استفاده می کنیم.

دستور git status : لیست تغییراتی که هنوز در git به صورت untracked - modify می باشند و همینطور لیست تغییراتی که در حالت stage می باشند ولی هنوز commit نشده اند.

دستور git log : لیست Commit ها را نشان می دهد.

کاربرد دستورات checkout, revert و reset در Git :

- **reset :** برای حذف تغییرات local ای که stage شده اند یا stage نشده اند استفاده می شود. :
git reset -hard HEAD

* نکته اینکه reset نمی تواند فایل یا فولدر های track نشده را حذف کند

این کار باید با دستور clean انجام شود

- **clean :** این دستور برای حذف فایل ها و فولدرهای Track نشده استفاده می شود.

○ **git clean -n** : لیست فایل هایی که می توانید حذف کنید را نمایش می دهد.

○ **git clean -f** : حذف فایل های track نشده

○ **git clean -df** : حذف فولدرهای track نشده که داخل آنها فایل یا فولدری وجود دارد به

همراه محتویات داخل آنها

• **revert** : زمانی استفاده می کنیم که بخواهیم تغییرات یک commit را برگردانیم. این دستور تغییرات commit را برمی گرداند و خود commit در git log قابل مشاهده با وضعیت reverted می باشد.

`git revert commitId`

• **checkout** : حرکت بین Commit ها یا branch ها می باشد.

`git checkout commitId` : برگشت به Commit مورد نظر

`git checkout branchName` : برگشت به آخرین وضعیت که داشتیم

* برای مشاهده خلاصه لاگ همه کامیت ها از دستور : **git log --oneline**

دستور **diff** : برای مشاهده تغییراتی که انجام شده است و هنوز stage نشده اند
`git diff`

دستور **blame** : چه تغییراتی توسط چه کسانی تا به حال روی فایل مورد نظر انجام داده است

`git blame File01.txt`

فایل **.gitignore** : فایلی است که در آن مشخص می کنیم که git چه فولدرها و چه فایل هایی را Track نکند. مثل فولدرهای :

• `node_modules`

• `build`

• `test`

فصل سوم : Azure DevOps (Team Foundation Server - TFS)

Remote Repository های آنلاین :

- Azure DevOps
- GitHub
- GitLab

Azure DevOps نام جدید محصول TFS مایکروسافت می باشد که دارای سرویس ها و خدمات زیادی شامل :

- Source Control :
 - Distributed – Git
 - Centralized – Team Foundation Server
- Project Task Management
- CI/CD : continuous integration and continuous delivery
- Test
- و ...

لینک دسترسی به سرویس Azure DevOps :

<https://azure.microsoft.com/en-us/services/devops/>

نحوه ارسال پروژه بر روی Remote Repository :

```
git remote add origin
```

```
https://DevTube@dev.azure.com/DevTube/GitTrainingProject/ git/GitTrainingProject
```

```
git push -u origin --all
```

```
git push origin master
```

فصل چهارم : GitHub

- کاربرد GitHub به صورت یک Host برای پروژه ها می باشد.
- فضایی برای قراردادن Remote Repository ها بر روی آن
- مخزن پروژه های Open Source در دنیای برنامه نویسی ها

استفاده هایی که ما می توانیم از GitHub داشته باشیم :

- از نمونه پروژه های رون آن استفاده کنیم
- ما می توانیم بر روی GitHub دو نوع Remote Repo داشته باشیم :
 - Public Repo : که به صورت عمومی برای همه قابل مشاهده می باشد
 - Remote Repo : این امکان را به شما می دهد که پروژه های خود را روی آن داشته باشید. ولی کاربران ایرانی به دلیل تحریم امکان دسترسی به این سرویس را فعلا ندارند.

نحوه ایجاد Repository جدید و وصل کردن git local به Github :

مشاهده لیست Remote repo های یک پروژه در Git می باشد :

```
git remote -v
```

نحوه pull کردن از سرور remote :

```
git pull
```

لیست Remote ها : git remote -list

نحوه مشاهده لیست config ها : git config -list

نحوه ست کردن نام و ایمیل در گیت :

```
git config --global user.name "Mohsen Derambakht"
```

```
git config global user.email "derambakht@gmail.com"
```

نحوه کپی کردن سورس از remote repo رو سیستم خودمان : git clone address

فصل پنجم : GitLab

وظیفه اصلی یک Remote Repo هستش که امکانات جانبی دیگری همراه خود دارد مانند امکان CI/CD

فصل ششم – کاربرد Branch – Merge – Tag

Branch : به صورت پیش فرض در git ما یک branch با نام master داریم.

کاربرد : مهم ترین کاربرد اینکه شما بتوانید تغییرات خود را روی یک نسخه به غیر از نسخه اصلی انجام دهید و در صورت اطمینان از تغییرات انجام شده آن را روی شاخه اصلی اعمال کنید.

Branch Name : feature-my-file-uploader

مشاهده لیست branch ها : **git branch**

ایجاد یک branch جدید : **git branch BranchName**

نحوه سوئیچ کردن بین branch ها : **git checkout BranchName**

نحوه ایجاد کردن branch و سوئیچ روی branch ایجاد شده : **git checkout -b BranchName**

نحوه merge کردن branch مورد نظر در branch دیگر : از source به destination

git checkout destinationBranch => git checkout master

git merge sourceBranch => git merge feature-english-version

مقایسه تغییرات دو branch نسبت به یکدیگر :

git diff branch-1 branch2

کاربرد **stash** در **git**: ذخیره موقت تغییرات Commit نشده با یک نام مشخص و restore کردن آنها در زمان دلخواه :

git stash save contact-en-page-changes

git stash save contact-en-page-changes --include-untracked

برگرداندن تغییرات آخرین stash ایجاد شده در branch جاری :

git stash pop

مشاهده لیست stash ها : **git stash list**

گرفتن تغییرات از stash و اعمال رو branch بدون حذف کردن از سابقه stash :

git stash apply

کاربرد **Tag** و نحوه استفاده از آن : برای مدیریت Version ها عموماً از امکان tag استفاده می شود.

نمایش لیست تگ ها : **git tag**

ایجاد تگ جدید : **git tag TagName**

حذف تگ مورد نظر در local repo : **git tag -d TagName**

ارسال تگ های رو Remote repository : **git push --tags**

حذف تگ مورد نظر از روی Remote repository : **git push --delete origin tagName**

نحوه سوئیچ کردن و حرکت به tag مورد نظر : **git checkout tags/tagName**

فصل هفتم – فصل هفتم – راه اندازی Git Server به صورت لوکال

مزایای راه اندازی git server به صورت لوکال : git دارای دو repository زیر می باشد :

- Local repository
- Remote repository :
- استفاده از سرویس های آنلاین (git server online) :
 - GitHub
 - GitLab
 - Bitbucket
- استفاده از git server local : برای استفاده از این حالت دو روش پایه ای داریم :
 - نصب git server به صورت دستی :
 - استفاده از نرم افزارهایی که این کار را برای ما انجام می دهند مانند GitStack

نصب git server به صورت دستی :

۱. ایجاد یک فولدر در مسیر دلخواه برای git server
۲. سپس اجرای برنامه git bash و حرکت به مسیر فولدر ایجاد شده در بخش یک مربوط به git server
۳. اجرای دستور : git init name.git –bare
۴. باید فولدر را Share کنیم
۵. باید بخش Network & Sharing سیستم عامل خود را از طریق Control Panel در حالت Turn on network discovery قرار می دهیم

حالا client ها می توانند از روی repository مورد نظر clone کنند.

git clone computer-name/git_server/devtube.git

استفاده از **GitStack** و راه اندازی آن: ابزاری تحت وب که ابتدا باید فایل نصب آن را دانلود کنیم و روی سیستم عامل خود نصب کنیم. بعد از نصب یک پنل تحت وب در آدرس localhost/gitstack خواهیم داشت که می توانیم بان git server و repository های آن را مدیریت کنیم.

<https://gitstack.com/>

خسته نباشید دوستان عزیز

امیدوارم دوره مفیدی بوده باشه برای شما عزیزان