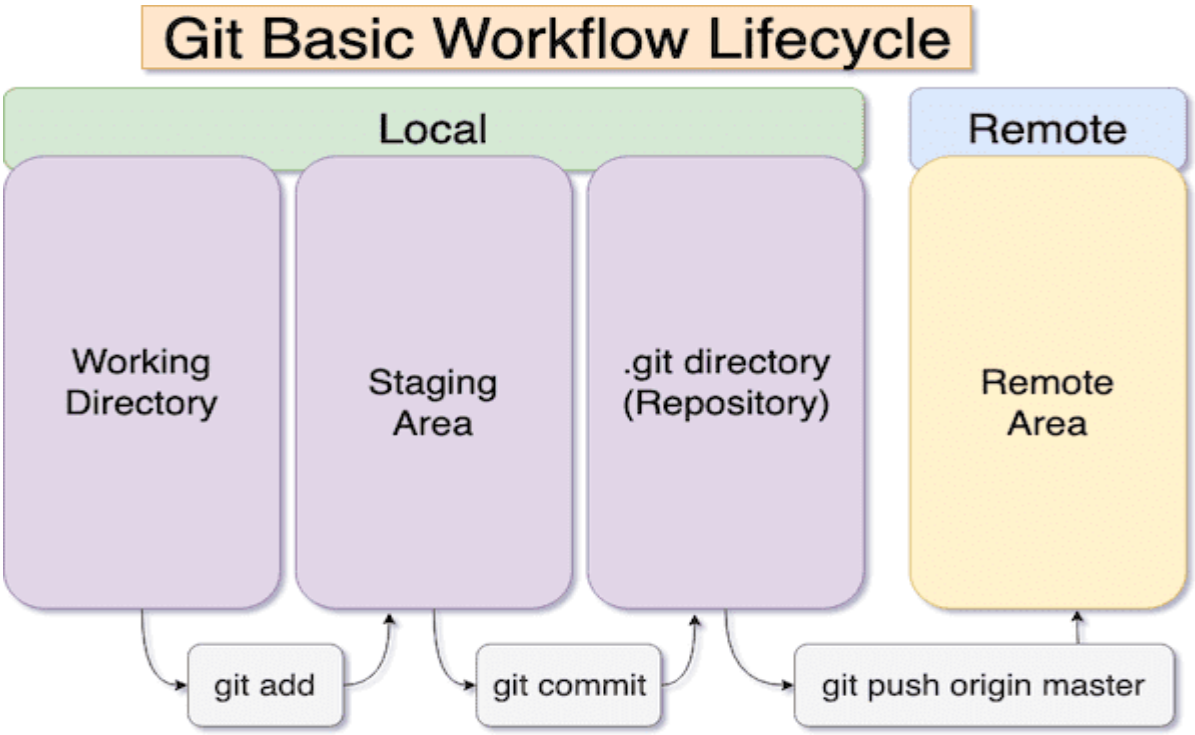
GIT

- To Install Git … → https://git-scm.com/download/win



شويه إعدادات مُهمه لمعرفه مين عدل علي الكود فيما بعد..

git config --global is a command in Git that is used to configure settings globally for your user account. These settings apply to all repositories on your machine.

The git config --global user.name "Dev_Nourhan" command is used to set the username associated with your commits in Git. It is a configuration option that allows you to identify yourself as the author of the commits made in a repository.

The git config --global user.email "your-email@example.com" command is used to set the email address associated with your Git commits. It is an essential configuration step when setting up Git for the first time on your system.

The git config --local command is used to set or get configuration options that are specific to the current Git repository. It allows you to configure settings on a per-repository basis, which overrides the global or system-wide Git configurations.

The git config --system command is used to configure Git on a system-wide basis. It sets configuration options that will be applied to all users and repositories on the system.

init.defaultbranch=master

git init is a command in Git that is used to create a new, empty Git repository or initialize an existing directory as a Git repository. It sets up all the necessary files and data structures required for version control with Git.

- initial-branch is created(master/main) الفرع الرئيسي الخاص بالبروجيكت

The git status command displays the state of the working directory and the staging area.

git add <File name> It adds files to the staging area. We can add single or multiple files at once in the staging area.

git rm -r --cached [directory] command removes a file from the Git index and keeps it in the working area.

git-reset … reset is the command we use when we want to move the repository back to a previous commit, discarding any changes made after that commit.

Commit State For Tracked files only*

git commit a commit is a snapshot of your project's files at a specific point in time. It is used to save your changes and create a new version of your project. Each commit has a unique identifier called a commit hash, which allows you to refer to it later.

git commit –m "any msg"

git log is a command in Git that displays a list of commit history in reverse chronological order. It shows information like the commit hash, author, date, and commit message for each commit.

git reset --hard is a command in Git that allows you to discard all changes.

Remote Repository refers to a repository that is hosted on a remote server, allowing multiple users to collaborate on a project. It acts as a central location where team members can push their changes and pull the latest changes made by others.

* عليها أخر تحديثات للكود ويقدر الديفلوبرز ياخدوا نسخه منها ويقدروا يخزنوا عليه آخر تعديلاتهم.

عندي أكتر من بلاتفورم تُتيح لي حوار الريموت ريبوزاتوري زي مثلا:-

Github(أشهرهم) – Gitlab – gitbucket - Azure Devops

* كل ريبوزاتوري بيكون له ديفولت برنش.

cls ….. To clear terminal.

Clone Repository

git clone <repository_url> علشان اخد نسخه من الريموت ريبو وأضافها ع اللوكل ريبو واشتغل عليها.

git remote –v To see the list of remote repositories associated with your local repository.

بيظهر اللينك بتاع الريبو اللي أنا بسحب أو بضيف عليه.

git branch –a is used to list all the branches in a Git repository, including both local and remote branches.

git fetch remote command is used to retrieve the latest changes from a remote repository without merging them into your current branch.

لحد هنا أنا سحبت فقط التعديلات اللي حصلت ... طب لو عاوزه اضفها عندي هقوله ↓

git merge <branch-name>

git pull origin branch_name ولكن توجد طريقه آخر ... إني استخدم الأمر بدلاً من الأمرين السابقين

git pull is a Git command used to fetch and merge the changes from a remote repository to your local repository. It combines the git fetch command (which downloads the latest changes from the remote repository) and the git merge command (which merges those changes into your current branch).

WHEN SHOULD I CREATE A NEW BRANCH IN GIT?

- New feature,
- Fix a bug,
- Experiment with some changes without affecting the main branch.
- Collaboration: When working on a project with multiple developers, each developer can create their own branch to work on a specific task or feature. This way, everyone can work independently without conflicts, and later merge their branches into the main branch.

VERSION CONTROL SYSTEM (VCS) OR SOURCE CONTROL MANAGEMENT

ال Version Control معناها أنه يكون عندي كنترول علي ال History بتاع التغيرات اللي حصلت في Project files بتاعتي.

- عندي 3 أنواع من ال Version Control وهما:-

1- Local Version Control (LVC)

من اسمه ف هو عباره عن حاجه لوكل علي الجهازي بتاعي مش متشيره مع حد ولا مرفوع علي حاجه ولا حد شايفه.

2- Central Version Control (CVC)

البروجيكت مثلا علي سيرفر خاص بالشركه بتاعتي وانا بتعديلاتي شغالين لايف علي السيرفر .

3- Distributed Version Control (DVC)

شبه ال central ولكن في فرق هو مرفوع shared ومتاح للناس أو لمجموعه علي حسب أنا عاوز اي ولكن هنا أنا ب clone نسخه من البروجيكت ده علي جهازي واقدر اشتغل لوكل علي جهازي بس لما بخلص شغلي أقدر اعملها Push أو أعمل Pull للتعديلات اللي حصلت من أشخاص تانيه شغالين علي البروجيكت. علشان كده في الأول قولت إني ب clone نسخه من البروجيكت مش ب download لان ال clone هينزلي نسخه من المشروع وهفضل متصل بالسيرفر بحيث لو في

تعديلات حصلت علي البروجيكت أقدر اعملها pullعندي أو العكس لما أخلص تعديلاتي أقدر أرفعها واللي معي بالبروجيكت يقدروا يشوفوها وينزلوها عندهم.

- ال Git بتاعي من النوع ال Distributed Version Control

GIT ARCHITECTURE(SIMPLE ARCHITECTURE)

في السكشن ده بنحاول نوضح شويه من ال git architrcure وكانت ايه المتطلبات اللي اتحققت في ال GIT!

- مع كل تعديله حتي لو حرف في file فال Git بتاخد Snapshot من نفس ال file كله مش بس ال line أو الحرف اللي اتغير.

- في ال Working Directory /Tree بيكون عندي نسخه واحده بس من أي file مهما كان الفايل ده كان فيه منه أكتر من عشروميت فيرجن في الأخر مش هيشوف غير نسخه واحده وبس.
- في ال Git repository أو زي ما بندلعها ال git repo في ده بيكون شايل كل ال versions من الفايل ده.
- ال Git بي Track Everything سواء كان في ال content محتويات أي فايل أو فولد وال Metadata علشان لو غيرت اسم فايل مثلا أكون عارف ان الفايل ده كان اسمه قبل كده ايه واتعدل امتي ومن قبل مين؟!
- ال Git بيكون OS Independant شغال مع أي OS مش فارق معاه علشان كده ال Git في الأخر عباره عن فولدر هيشتغل لو اتحط في أي حته (.git).
- أي object ال Git بيعمل له track لازم يكون له (SHA-1 hash)Unique IDعلشان لو غير اسم الفايل أو محتوياته أبقي ماسكه ب Id وأقارنه بال Idالجديد لو في اختلاف يبقي حصل تعديل لو مفيش اختلاف يبقي الفايل زي ما هو متعدلش.
- ال Git بي Track History بيعني أشوف ايه التعديلات اللي حصلت ومين عملها ولو حد رجع لفيرجن قديم أكون عارف أن الشخص ده رجع لفيرجن قديم (Log).
- ال Git شغال بال 3-tree Achitecture (Working Tree/Stagging Area"Index"/Git Repo)

سؤال : ايه ال Items اللي Git بيعمل لها Tracking ؟ (عندي 4 حاجات/Objects)

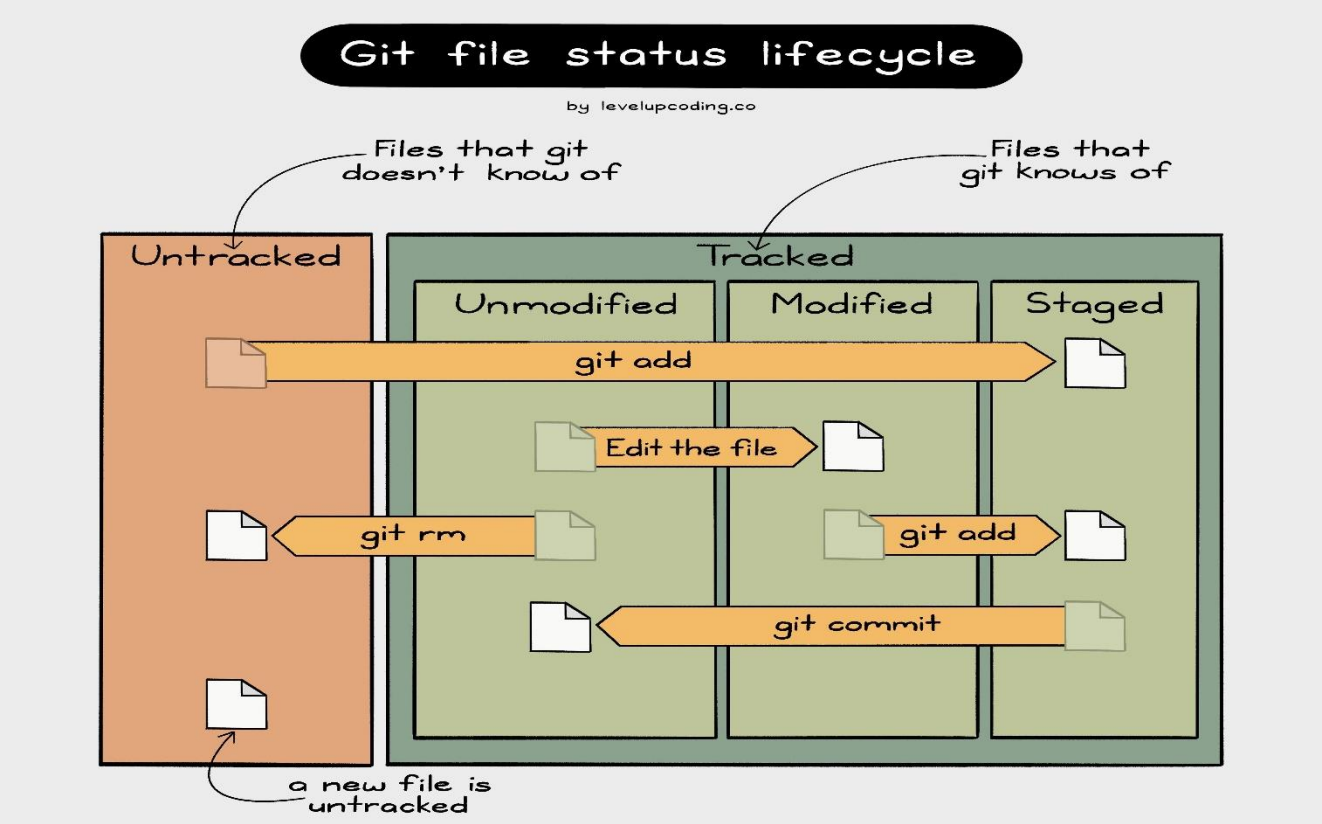
1- Blob = الFile and metadata
2- Tree= ال Folder and metadata
3- Commit مُجرد wrapper بيعرفك ال commit دي كان جواها أي فايل و فولدرات ومين اللي عملها واتعملت امتي وبدايه التعديل كانت منين tree "من الاخر بتعرفك أن شويه الحاجات دي اتعدلت في كومنااايه واحده مع بعض"
4- Tagged annotation

```
git add .
```

adds a change in the working directory to the staging area. ‘staging’ بيعمل حاجه اسمها
وبيجهزه لأول Commit

بتحول الفايل لو كان جديد من ال unTracking ل Tracking بس ده مش معناه أنه بقا عندي فيرجن 1 من الفايل ده لسه لحد ما تعمل commit.

طب لو الفايل مكنش جديد ف هي بتحوله ل Modified




```
git commit
```

Record changes to the repository. بتعتمد التغيرات بتاعت في الريبو

GIT INSTALLATION

To Install Git … → https://git-scm.com/downloads

التسطيب الخاص ب Git بسيط جدا وعباره عن Next…=>Next ولكن في شويه Configuration لازم تعمله وأقل Configurationبتتعمل هي انك تضيف إسمك والإيميل بتاعك.

طب ليه بيطلب الإسم و الإيميل بتاعي؟! 🤔 = علشان لما تيجي تعمل commitلحاجه مثلاً لازم يكون مين اللي عدل وعمل كده وامتي تم هذا التعديل بالتاريخ والوقت بالدقيقه والثانيه😂

NOTES

بعض ال Configuration تم شرحها إعلاه ولكن ممكن هنا هزود شويه تفاصيل أكتر لفهمها ولمعرفه ال Configurationالتي تم تطبيقها عندك بنستخدم الأمر ده:

git config --list

#لحد هنا أنت جاهز إنك تستخدم ال Git

- لو عندي فولدر البروجيكت بتاعي علي الجهاز وعاوز أحوله ل ريبو... عن طريق إني أقوله git init ولازم ال command ده أعمله Executeمن داخل الفولدر الخاص بالبروجيكت بتاعي.
- هلاحظ بعد تنفيذ ال command السابق أنه أنشاء فولد اسمه .git وبيكون hidden folder.
- **#لحد هنا أنا حولت البروجيكت بتاعي ل ريبووو**

- git status بيوريني حاله الفايلز اللي موجوده عندي من حيث هل هي Tracked أو Untracked هل هي modified ولا unmodified ... من الآخر بتعرفك ايه الوضع في ال Working Directory بتاعك.
- git ls-files Show information about files in the index and the working tree.

بشوف ايه الحاجات اللي ال git بيعمل لها Tracking

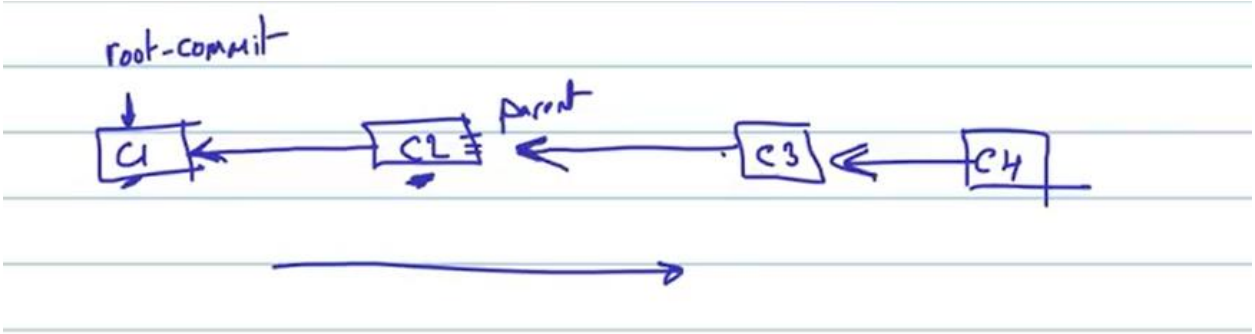
- كل مره بنعمل commit فأحنا بنعمل commit للفايل بال metadata بتاعته علشان لو كان في فولدر واتمسح أكون عارف المسار بتاعه.

```
- git log
```

- Shows the commit logs.
- بيعرض ال commands اللي حصلت ومين ال author والايميل بتاعه واتعدلت امتي بالضبط والمسج اللي اتبعتت من الكومنت.
- ال commit object هو اللي بيوصلي للحاجات اللي اتعدلت جواه.

```
git restore
```

- علشان لو عملت Stagging لفايل بالغلط وعاوز ارجعه تاني واعمله unstagging
- ملحوظه: لو كان عندي فايل واتعمل Commite عليه ورجعت عدلت فيه وعملت Commite ف ده هيتحفظ جواه ال Parent لأن كل commitبيشاور علي ال Parent بتاعه علشان يوريني ال History للتغيرات كانت أي ووصلت ل أي!! والكلام ده بيتخزن في Linked List ده بس علشان التغيرات متكونش سايحه علي بعضهاا.



- الشكل ال Linear ده في Git بنسميه Branch "عباره عن تسلسل لشويه Commits"

- A branch in Git is simply a lightweight movable pointer to one of these commits

The default branch name in Git is master .

```
git diff
```

- علشان يوريني الفرق ما بين ال 3 areas وهيظهر الفروقات باللون الأخضر +
- الفرق ما بين ال Working tree وال Index و once إني عملت stagging مش هلاقي فروقات.
- طب لو عاوز أظهر الفروقات ما بين اللي في ال stagging area والريبو هعمل أي؟؟!

Git diff – staged

- كنا اتكلمنا عن ال git commit –m "any msg" وقولنا اني أقد اكتب مسج معاها.. طب لو أنا عاوز اسرد في المسج دي مش مجرد جمله ف من ضمن ال Configuration الخاصه بال Git فهي بتتيح لينا انك لما تكتبك ال Command ده تفتح لي Editor اكتب فيه المسج براحتي.
- ال Editor ده بيكون ال Default Editor عندك.

git config --global core.editor "code --wait"

- بمجرد ما بكتب git commit بيفتح لي ال Editor علشان اكتب المسج بتاعتي وأسرد براحتي 😎.

UNDOING THINGS

- لو عاوزه امسح الريبو بتاعي كله بكتب rm -rf <repository_name>
- لو عاوزه اعمل undoing ل فايل من حاله ال Tracking لحاله ال untracking بكتب الأمر ده

```
git rm --cached filename
```

- لو كان عندي فايل وعملت له Commit وبعدين عدلت فيه وبقا Modified "يعني الفايل اللي في ال Working Tree مش هو اللي نفسه اللي في ال Stage area " طب أنا عاوز ال Discard changes دي أعمل أيه؟!

```
git restore <file>
```

معناها أنه هي Restore الفايل ده من اللي في ال Stagging area والفايل حالته هترجع Unmodified يعني اللي موجود في ال Working Tree هو نفسه اللي في ال Stagging area.

- لو عندي فايل وعملت له Commit والمره دي عاوزه ارجع التعديلات اللي حصلت دي اعمل أي؟!

1- هرجع الفايل من ال Stagging area ل Working Tree بإستخدام الأمر
git restore --staged <file_name> كده رجع ولكن التعديلات لسه فيه.

2- علشان Discard هذه التعديلات هكتب الأمر git restore <file>

- ملحوظه `git commit -am "any msg"` الأمر ده عباره عن اختصار بحيث يعمل stagging لكل التعديلات وكمان يضيف مسج وكل ده في سطر واحد.
- طب لو عاوز ارجع بحيث اعدل في آخر رساله Commit بكتب الامر ده

```
-  git commit --amend -m "an updated commit message"
```

is a convenient way to modify the most recent commit.

- طب لو عاوز أرجع فيرجن قديم عن اللي أنا واقف عنده مع العلم في حاجه اسمها ال Head وده بيشاور حالياً علي آخر فيرجن. يعني من الآخر هحرك ال head ده علشان ارجع لفيرجاانات قديمه أو اطلع لفيرجن حديث. git reset HEAD ~1 يعني إرجع خطوه لوراء والتعديل ده هيسمع في ال Stagging area بس ولكن لو عاوزاه يسمع في Working Tree كمان هكتب git reset –hard HEAD ~1
- طب لو عاوز أعمل Fastforward لل Commit اللي أنا رجعتها دي أعمل أي؟ 😭 الأول قبل ما ارجع لو عاوز اشوف ال Log بتاع الكلام ده بستخدم الأمر ده git reflog git reset HEAD @{1}

TAGS

ملحوظه: طول ما أنا شغال وقاعد أعمل Commit حتي لو كنت معدل في line واحد إنجازا بنقول ان ده version بس لا ... في الواقع أنا بقعد أعدل مجموعه من الحاجات لحد ما أوصل ل Commit مُعينه ببقا عاوز أعلمها أو أعمل لها bookmark علي أنها دي الVersion بتاعي لأنه مش شرط كل تعديل بضيفه للRepo يكون عباره عن Version جديد لا ده بيكون مُجرد Fix.

عندي نوعين من ال Tags وهما :-

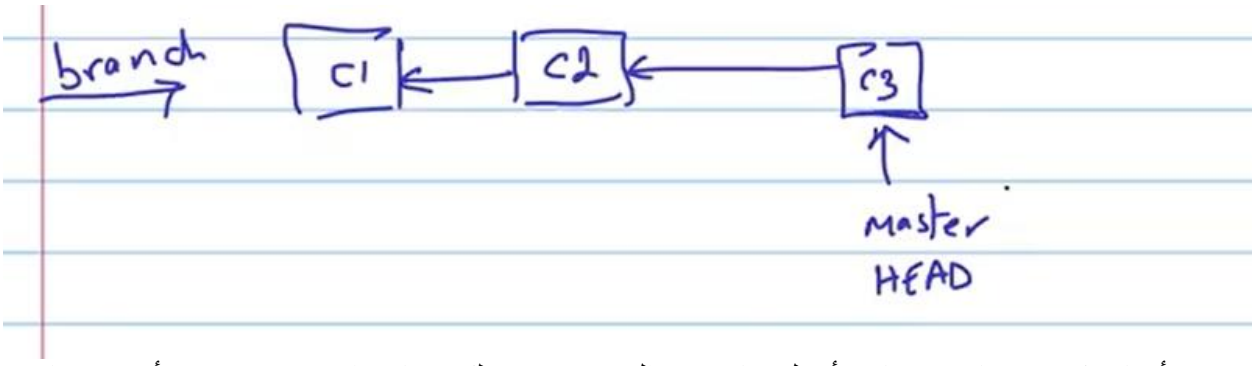
1- Lightweight Tags 2- Annotated Tags

```
git tag -a v1.4 -m "my version 1.4"
```

```
git Show v1.4 علشان انط علطول علي عرض الفيرجن اللي أنا عاوزه بدل ما اعرضهم بال Commits
```

GIT BRANCHING

ال Brach هو عباره عن سلسله من Commited object نتكون Connected ببعض عن طريق ال Parent



- أوقات كتيره ببقا مش عاوز أعدل حاجه في ال Master ولكن عاوز اترفع منه بحيث أجرب حاجه بعيداً عنه أو أحل Issue أو أتيست عليه وبعدين أقرر هل أنا محتاج الكلام ده بالفعل ولا لا وعاوز أرجع لل Master بتاعي.
- ممكن أعمل Branch بحيث يكون C3 هو ال Parent بتاعي وأعمل Commit هعمله هيبقي متعلم علي أنه تبع ال Branch الجديد.
- علشان أعمل Branch جديد git branch branch_Name
- علشان اعرض ال Branches اللي عندي هقوله git branch
- ال Branch الحالي هلاحظ أن لونه بالأخضر – طب أي المقصود بال Current Branch ؟ = هو ال Branch اللي لو عدلت في أي File وعملت Commit هيبقي امتداد لل Master Branch.
- علشان اخلي ال Branch الجديد اللي عملته هو ال Current هقوله

```
git switch <branch_name>
```

- لو التعديلات اللي عملتها في ال Branch الجديد تمام و أووكي معي وعاوز أنقلها لل Master Branch هعمل Merge بس لازم أكون واقف عند ال Branch اللي هعمل Merge فيه

```
git merge master
```

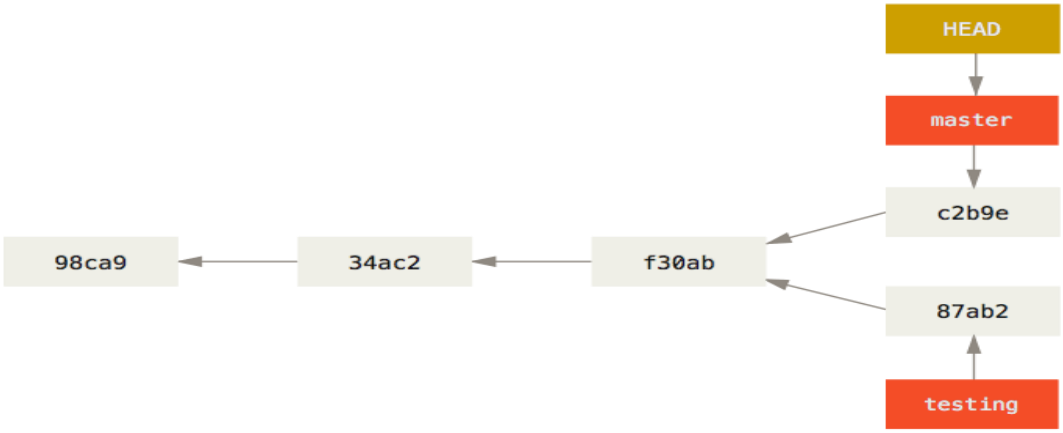
- لو عاوز أشوف أي هي ال Branches اللي اتعمل لها Merge في ال Master Branch بتاعي هقوله

```
git branch --merged
```

- طب لو عاوز امسح ال Branch اللي كنت لسه عامله إنشاء هقوله

```
git branch -d localBranchName
```

- لحد هنا والدنيا بسيطه ولذيذه ولكن مش دايما بتكون كده.. لا ده ممكن تعمل new Branch من ال Master وتروح تشتغل عليه وتعمل Commits وتعيش الحياه وفي نفس الوقت يكون الماستر اتعمل عليه شويه Commits حلوين وده اللي بيسموه ال Divergent History وده بيكون شكله



- وضع ال Merging هنا مُختلف `git merge testing_branch`

= هيعمل Commit جديده خالص بيكون فيها ال Merge بتاعت ال Master بالتغيرات اللي حصلت عليه بعد ما عملت ال Branch الجديد و التغيرات اللي علي ال Branch الجديد نفسه واقدر امسح ال Branch الجديد بعد ما ضفت تعديلاته لل Branch الرئيسي.

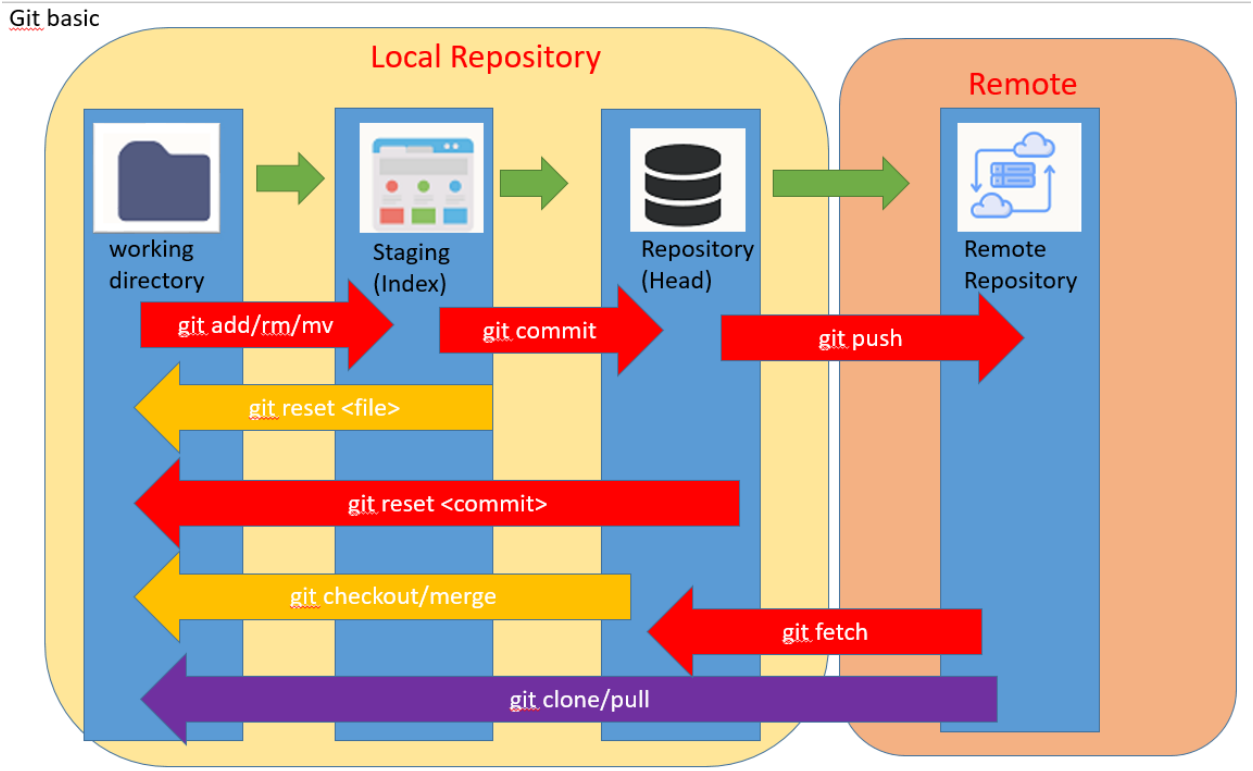
- هل بقا كده الدنيا لطيفه وحلوه ومفيش مشاكل..؟!

= لا طبعا 😒 ممكن يحصل Conflict أثناء ال Merging لو كنت مثلا معدل في نفس الفايل في ال 2 Branches اللي هتعمل لهم Merging 🤦‍♀️ ... طب أي الحل؟!

هو هيظهر لك الفايل ويقولك السطر ده في ال Master Branch شكله كده وفي ال Testing Branch شكله كده شوف بقا أنت عاوز انهي سطر فيهم وامسح التاني.

WORKING WITH REMOTES

- أول مره بتنزل البروجيكت بتعمل حاجه اسمها Clone وبيكون عند نسخه طبق الأصل من ال Remote Repo وقت ما حصل ال Clone ولو عدلت شويه حاجات أقدر أرفعها من ال Local Repo واعمل لها Push وأقدر كمان أعمل Pull بحيث أقدر اشوف كل ال Updates اللي حصلت علي البروجيكت من قبل باقي التيم اللي معي.



- لعرض جميع ال Remotes الموجوده عندنا بنستخدم الأمر

```
git remote                          # lists all the repos of your project
```

- لو قولت له الأمر ده ف معناه أنه هيظهر تفاصيل أكتر عن ال Remotes وكمان أي اللي تقدر تعمله عليه سواء fetch أو Pull

```
git remote  -v
```

- So, when you run git remote -v, Git lists all the remote repositories associated with your local repository along with their corresponding URLs. This information helps you understand where your local changes will be pushed when you use commands like git push, and from where you can pull changes using git pull. It's a way to inspect the configuration of your repository's connections to remote locations.

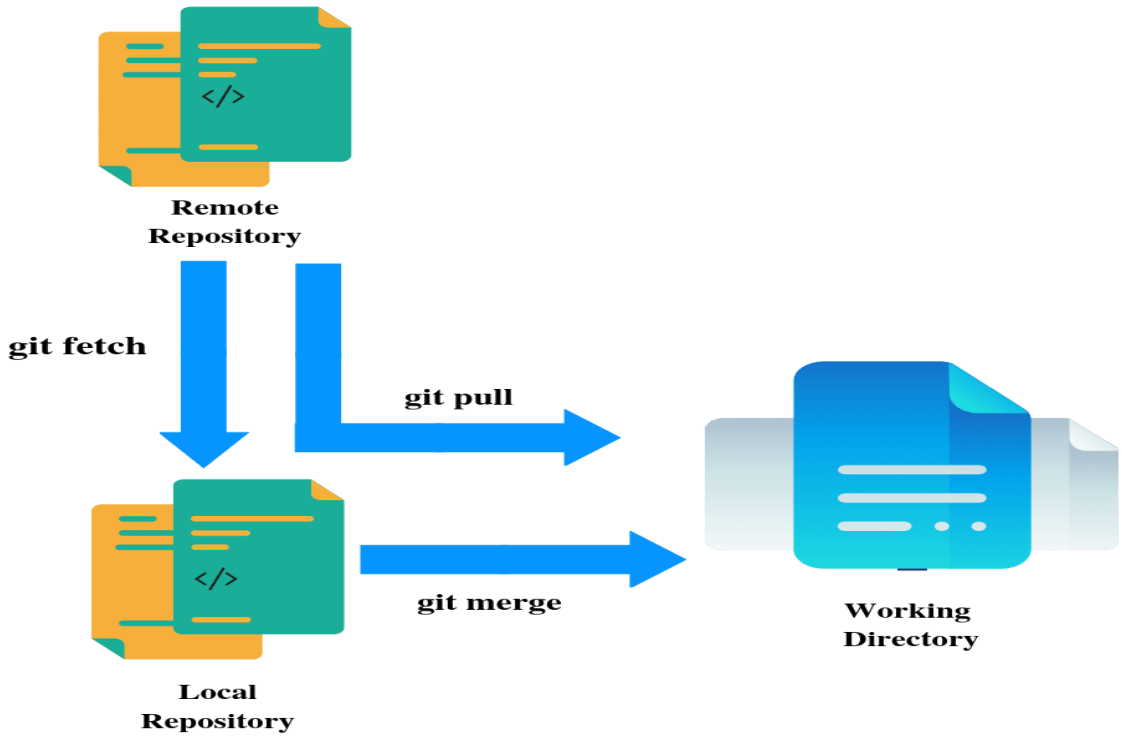
- If you want to fetch changes from a remote branch named from the origin remote, you can use the following command:

```
git fetch origin
```

- Git fetch remote branch is a command used to fetch changes from a remote repository without merging them into your local branch. This allows you to

see the changes made in the remote branch without affecting your local branch.

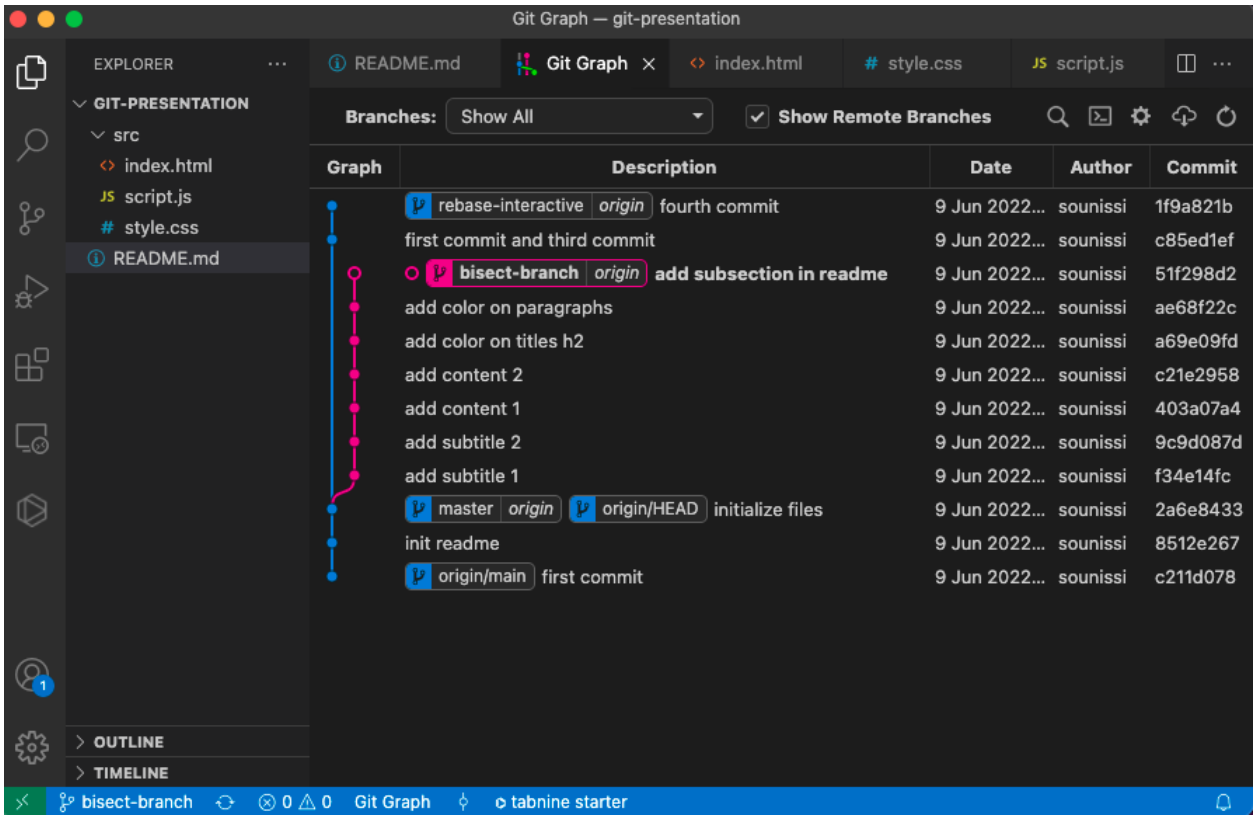
- `git merge origin` لو عاوز أوافق علي التغيرات دي واخليها تسمع عندي في اللوكل
- Note: The most significant difference between "git pull" and "git fetch" is that "git pull" automatically merges the fetched changes into the current branch, while "git fetch" does not. This makes "git pull" a more convenient command if you want to quickly update your local branch with changes from the remote repository.



- git push is a Git command used to send your local changes to a remote repository.
- `git push -u origin main` => u معناها `upstream`

GIT IN VSCODE

- Visual Studio Code (VSCode) provides robust support for Git integration. It includes a built-in Git extension that allows you to perform most Git operations directly from within the editor.
- **GitLens** Extension مهمه وبتعطيني شويه مميزات زي عرض جميع ال commits اللي عملتها وهي أثرت علي كام فايل وشكل الفايل قبل وبعد التعديل عليه – ممكن اعرض ال Branches اللي موجوده عندي ...
- **Git Graph** بيعرض لي بشكل جرافيكالي كده التعديلات وال Bracnching اللي حصل



What is the difference of Git and GitHub?

Git is a version control system that allows developers to track changes in their code. GitHub is a web-based hosting service for git repositories.

SUMMARY

- **git add .** add command adds a change in the working directory to the staging area.

- **git commit -m "any msg"** The git commit -m command is used in Git to create a new commit with a specified commit message.

- **git push -u origin Brance_Name** is used to push local changes to a remote repository. لازم أكون لي صلاحيات أني أرفع تعديلاتي

- **git pull origin branch_name** is used to receive data from remote server. It fetches and merges changes from the remote server to your working directory.

REFERENCES

1- What is Git In Arabic https://lnkd.in/d22AmKMc

2- git Issam Abdelnabi Playlist
https://www.youtube.com/playlist?list=PL4n1Qos4Tb6R4guGC4oX_PZVt8E8XpvqE

3- Git Tutorial From Javatpoint https://www.javatpoint.com/git
4- Eng Ahmed Sami https://youtu.be/Q6G-J54vgKc?si=YPHcDTaK4sFJRdJV
5- https://marklodato.github.io/visual-git-guide/index-en.html