==> What is Git and Github ?

-- Git is distributed version control System

ال git بيهندل ال versions بتاعتك
لو انت بتشتغل في software house او في أي شركة و عندكم project معين شغالين عليه و ال project دا بيطور مع الوقت سواء بتحلوا فيه مشاكل معينة او بتضيفوا فيه features جديدة او بتعملوا enhance للكود ف المراحل اللي انت بتمر بيها دي كلها اسمهاversions

-- Git is free and open source

أي tool بتشتغل بيها لازم تعرف هي بفلوس ولا لأ عشان تقرر قبل ما تتكلم عنها او تشتغل بيها هل هي محتاجة منك cost ولا لأ
ال git عبارة عن free وopen source

-- Github is source for projects and sources [Gitlab , BitBucket]

Github عبارة عن website فيه source ال projects اللي انت بتشوفها
الشركات كلها بتحط فيه ال source بتاع ال projects بتاعتهم سواء الحاجات ال open sourceاو المكتبات اللي انت بتشتغل عليها هتلاقي ال project بتاعها موجود على github ممكن ت contribute عليه و تحط فيه تعديلات و كدا
في مواقع كتير زي gitlab و bitbucket بتأدي نفس وظيفةgithub

-- Github simplify using Git

ال github بيسهل استخدام ال git بطريقة بسيطة و سهلة و بيديلك شوية إمكانيات كويسة زي issue trackerال

-- you can use Git without Github

-- Git has GUI

ال git ليه واجهة جرافيكية يعني تقدر تستخدمه بال mouse عادي
حتى موقع Github ليه برنامج اسمه github desktop تقدر تعمل فيه كل القصة دي بال mouse عادي

==========

==> Why you must learn Git ?

-- Devs Contribute to the same project

مع ال Git كلنا بن contribute لبروجيكت واحد مفيش حد منفصل او شغال لوحده

-- you can revert changes

تقدر ت revert changes و تعرف مين عملها و تقدر تلغيها

-- you can collaborate to fix issues

-- you can collaborate to create new features

-- you can solve conflicts

لو واحد خد function معينة و عدل عليها و ال function دي فيها عندي اختلاف ف بسهولة شديدة بيجيب الاتنين و يجيب مقارنة مبينهم و نشوف ال conflict فين و تعدل و ترفع و هتلاقي الشخص التاني ياخد النسخة السليمة او العكس

-- you can organize features

==> Words you will hear

-- Repository

هو المستودع اللي بتحط فيه اكواد ال project بتاعك كاملة لو انت مثلا عندك two projects شغال عليهم في الشركة ف المفروض يكون عندك اتنينrepository
ال repository بيختصروها لrepo

-- Branch

يعني قسم او فرع
ال branch في عالم ال Git فرع من ال repo ممكن تاخد branch عشان تعمل تعديل معين بس في البروجيكت التعديل ممكن يكون إضافة صغيرة مطلوبة منك او ممكن تاخده تحل بيه bugطبعا دي آلية من شركة لشركة لكن دا الأساس

-- Local Repo

ال local هو الحاجة بتاعتك انت اللي شغال عليها دلوقتي على ال PC بتاعك

-- Remote Repo

ال remote هنا ممكن يكون موقع ال github او ال bitbucket او ال gitlab او السيرفر بتاع الشركة

-- Commit (snapshot or checkpoint in your local Repo)

لما انت تعمل تعديل معين او خلصت feature معينة او حليت ال bug بتعملcommit
ال commit دا بيعمل checkpoint للكود في ال local repo بتاعك

-- Clone [from local to remote]

Cloneيعني استنساخ بستنسخ ال repo اما من ال local او الremote
ال local ممكن تستنسخ منه بس في location تاني

-- push [upload local changes to remote]

ال push بترفع منه بت upload ال changes بتاعتك لل remote repo

-- pull [you pull changes from remote repo to your local]

ال pull بتسحب التعديلات من ال remote repo لل local repo بتاعتك

-- pull request [tell other about your changes to pull it from local to remote]

ال pull request انت عملت تعديل معين ف بتقول للي معاك في تعديل عايزك تاخده من ال localلل remote عندك

-- README

بنكتب فيه أي حاجة تخص المشروع أي notes عاوز تحطها في المشروع أي حاجة محتاج الشخص لما يجي يدخل على المشروع يعمل أي حاجة يشوف مثلا ال steps دي يعملها
ال README مهم جدا بيكون فيه الحاجات اللي ممكن تخلي الناس ميغلطوش في حاجة مثلا انت بتعملها
امتداده md و دي markdown بتكتب بيها زي ال text كدا بس بتنثقه

==> Important Notes

-- Create Repository for every project

-- Create a new branch for every feature or enhancement

-- No need to connect to remote repo when working

-- Anyone can push and pull depend on permissions

==> Getting Started

-- you can create a repository with either of the following commands :

- ❖ git clone < your repository URL >
  - ⇨ Copies a remote repository into your current directory
- ❖ git init
  - ⇨ Creates a new empty repo in your current directory

==> Add / Reset / Commit

-- show status :

- ❖ git status
  - ⇨ Lists changes in working directory and stages files

Git status بيوريني الحالة او ايه اللي ناقص او ايه اللي عندك في ال working directory يعني بيفهمني ايه اللي بيحصل

-- add files :

- ❖ git add < file1 > < file2 >
  - ⇨ Adds < file1 > and < file2 > to the staging area
- ❖ git add *.extention (git add * .py)
  - ⇨ Adds all python files in the current directory to the staging area
- ❖ git add *
  - ⇨ Adds all files to the staging area

-- unstage:

- ❖ git reset head < file >

بيرجع ال file من ال stage اللي مش عاوز اعمله track عاوز اسيبه

- ❖ git reset
  - ⇨ removes all files from staging area (opposite of git add)
- ❖ git reset < filename >
  - ⇨ removes < filename > from staging area

-- commit changes :

- ❖ git commit
  - ⇨ Records everything in the staging area to your repository. The default text editor will prompt you far a commit message
- ❖ git commit -m “ commit message “
  - ⇨ Records everything in the staging area to your repository with the commit message “ commit message “
- ❖ git commit –amend
  - ⇨ Modify last commit instead of creating a new one useful for fixing small mistakes

==> Push Local Changes To Remote Repository

-- show branches:

- ❖ git branch
  - ⇨ list branches
- ❖ git branch < branch-name >
  - ⇨ create new branch < branch-name >
- ❖ git remote -v
  - ⇨ git remote repository

-- push changes :

- ❖ git push < Repo > < Branch >
  - ⇨ incorporates changes from local repo into < repo > < branch >
- ❖ git push
  - ⇨ incorporates changes from local repo into ‘origin’

==> Pull Changes From Remote Repository

-- pull changes from remote repository :

- ❖ git pull
  - ⇨ incorporates changes from ‘origin’ into local repo
- ❖ git pull < repo > < branch >
  - ⇨ incorporates changes from < repo > < branch > into local repo

---

==> Everything about configuration

-- show all configurations :

- ❖ git help config

هيفتح ال manual بتاع ال git اللي فيه حاجات ممكن متكونش مكتوبة في ال file اللي عندي

- ❖ git config -l

هيجبلي ال configuration اللي عندي

-- show user email :

- ❖ git config --global user.email

-- change user email :

- ❖ git config –global user.email “ your github email here “

-- edit configuration :

- ❖ git config --global –edit
- ❖ git config --global credential -helper store

⇨ will store your login details next time they are entered, saving you from having to enter them again

---

==> generate and test github public key

-- Generate key :

- ❖ ssh -keygen -t rsa -b 4096 -c “ your github email here “

-- Git key to copy :

- ❖ cat ~/ .ssh/ id_rsa.pub

-- same command example in windows " change shorouk with your username " :

- cat c:\users\shorouk\.ssh\id_rsa.pub

-- test key and connection :

- ssh -T git@github.com

==> create repository from existing project

-- create new directory :

- mkdir " your directory name here "

-- initialize empty git repository :

- git init

-- create empty file :

- touch " your file name here "

-- show status :

- git status

-- add files :

- git add < file > < file >

-- commit file :

- git commit -m " your commit message here"

-- add repository :

- git remote add origin "ssh repository URL"

-- example :

- git remote add origin " git@github.com: shorouk/course.git"

-- push data :

- git push -u origin master

ال u- بتخليك تعتبر انك بتعمل pull و بعدين push الاتنين متتاليين

==> Everything about aliases

-- make alias for command " status " :

- git config --global alias.st status

-- show alias content :

- git config --global alias.st

-- test "status" command :

- git st

-- make alias for command " branch " :

- git config --global alias.br branch

-- show alias content :

- git config --global alias.br

-- test " branch " command :

- git br

-- make alias for command " commit -m " :

- git config --global alias.cm “ commit -m “

-- show alias content :

- git config --global alias.cm

-- test " branch " command :

- ❖ git cm

-- show all edits :

- ❖ git config --global –edit

==> braching and merging

-- show branches :

- ❖ git branch

-- switch to branch :

- ❖ git checkout “ branch name “

- ⇨ switch to editing branch < branch-name >

-- delete branch :

- ❖ git branch -d < branch name >

-d بيعملك حاجة اسمها save delete بيخلي ال git ي check هل ال branch بتاع كدا فيه تعديلات معينة متعملهاش merge و في الحالة دي مبيرضاش يحذف

-D بيحذفه حتى لو في تعديلات

-- move / rename branch :

- ❖ git branch -m “ new branch name “

-- merge branch with current branch :

- ❖ git merge “ Branch name you need to merge “

- ⇨ merge < branch name > into current branch

==> Mastering Stash

-- Create text file with " hello world " string inside it :

- ❖ echo “ hello world “ > about readme.txt

-- save work to stash :

- ❖ git stash

-- list items in stash :

- ❖ git stash list

-- Get work from stash :

- ❖ git stash pop

stash بتخزن فيه الملفات بتاعتك زي الصندوق اللي بحتفظ فيه حاجة لحد ما اجى استخدمها
Pop بيجيب الملفات و يعمل drop لل stash

-- save work to stash with description :

- ❖ git stash save “ description here “

-- get specific stash :

- ❖ git stash pop stash@{1}

-- delete stash :

- ❖ git stash drop stash@{2}

-- show latest added stash content :

- ❖ git stash show

-- show specific stash content :

- ❖ git stash show stash@{1}

-- empty the stash :

- ❖ git stash clear

-- get work from stash without delete it :

- ❖ git stash apply

apply بياخد التعديلات زي ما هي و يسيبلك ال stash في مكانه

==> Restore / clean

-- restore staged files :

- ❖ git restore --staged “ file name here “

لو محتاج اشيل فايل من الstage area

- ❖ git restore --staged *

-- show files that will be cleaned :

- ❖ git clean -n

-- remove un needed files :

- ❖ git clean -f

==> resetting the head

ال reset بيخليك تحذف ال commit اللي انت مش محتاجها
بيخليك تحذف مجموعة ال commits اللي انت بتكون عاوز تلغيها او فيها مشاكل ايا كان

-- show log file :

- ❖ git log

- ⇨ prints commit history of repo

- ❖ git log < file name >
  - ⇨ prints commit history of < file name >

-- reset head :

- ❖ git reset --hard " commit hash here "

-- push edits from local to remote with force flag :

- ❖ git push origin main –force

لما بخلي ال head عند commit معين وبعدها اعمل reset ف كل ال commits اللي بعده بتتمسح

==> ignoring files and directories

ازاي نتجاهل ملفات معينة عشان متترفعش للمشروع بتاعنا

-- create gitignore file :

- ❖ touch -gitignore

ملف ال gitignore هو الملف اللي هيحتوي على الملفات اللي عاوزين نتجاهلها و متترفعش مع ال project

-- push file that is ignored :

- ❖ git add -f < file name >
- ❖ git add --force < filename >

لما يبقى في file في ال gitignored و عاوز اعمله push

-- example for gitignored file content :

```
*.log
! vip.log
node_packs/
text.txt
```

==> tagging and releasing

ال tag هي حاجة بتخليك تعلم على جزء معين من المشروع بتاعك في حقبة زمانية معينة بيستخدموها دلوقتي للاصدارات في المشروع بتاعك
ميزة ال tag بتخليك تحمل الجزء الاول لوحده و الجزء التاني لوحده و هكذا

-- show all tags :

- git tag

-- add new lightweight tag :

- git tag " version name or tag name here "

ال git lightweight tag هي tag مرتبطة بال commit حتى المعلومات او ال message اللي انت كتبتها في ال commit

-- add new annotated tag :

- git tag -a " version name or tag name here " -m " description or message "

ال annotated tag ملهاش علاقة بال commit انت بنفسك اللي هتكتب ال description الخاص بيها

-- push tag to remote :

- git push origin " tag name here "

-- list all tags starting with v1 :

- git tag -1 "v1.*"

-- delete tag :

- git tag -d " version name or tag name here "

-- delete tag from remote :

- git push origin --delete "version name or tag name here"

Course link :

- https://www.youtube.com/watch?v=ACOiGZoqC8w&list=PLDoPjvoNmBAw4eOj58MZPakHjaO3frVMF

Documentation link ;

- https://docs.github.com/en/get-started/using-git/about-git

By ➔ Shorouk Eldeep