1.What is Git?

Git is an open-source programming tool that allows users to effortlessly track the changes made during the early stages of software development. It allows individual programs to keep a record of the changes that they have made to easily restore or back-up earlier versions of their code and it allows teams of developers to record the changes that individual members make to a file or program. Designed to support distributed non-linear workflow, Git allows programmers to create non-linear histories and branched records of how a program has been developed over its time in Git.

2. What Can You Do with Git?

Git can be used to ensure that you have a detailed record of all the changes being made to a file of code. Individual users can take advantage of this documentation to quickly restore or reconstruct prior versions of code or to see where specific changes were made if those changes came with unintended consequences. Groups of developers can use Git to collaborate more effectively on a shared file as the system will keep track of all the changes made independently, providing a stable record of how each developer impacted the file over time.

- Learning Git will significantly increase your efficiency and productivity as a software developer. The program has become an indispensable tool for archiving and documenting source code.

- Will also help improve your efficiency as a personal programmer and it will ensure that you have a well-documented and protected record of your work and data.

3. Basic Git Terms

- **Repository (Repo)**: A storage space for project files and version history.
- **Branch**: A separate version of the project.
- **Commit**: A **snapshot** or **checkpoint** of your work in the local repo.
- **Remote Repository**: A repository hosted on GitHub or another server.
- **Clone**: A copy of a repository from a remote or local source.
- **Pull Request**: A request to merge changes from a branch into another.
- **Push** : upload modifications
- **Pull**: Fetch and merge changes from a remote repo to your local repo.

4. Important Git Commands

Show Branches

git branch
git remote -v

Push Changes

git push origin main

Pull Changes from a Remote Repository

git pull origin main

Merge Updates

git merge <branch_name>
تستخدم عند دمج تغييرات من فرع معين إلى الفرع الرئيسي.

Fetch Updates (without merging)

git fetch جلب التحديثات دون دمجها

5. Git Configurations

Set Up User Information

used when executing commits

git config --global user.name "Your Name"
git config --global user.email "youremail@gamil.com"

Secure Shell (SSH) Key for Authentication

ssh-keygen -t rsa -b 4096 -C "youremail@gmail.com"

- **SSH keys** allow secure authentication for GitHub and remote repositories. دون الحاجة لإدخال كلمة السر كل مره

---

6. Git Aliases

- **Aliases help shorten common Git commands.**

git config --global alias.st status

- Ex: Instead of typing git status, you can simply type git st