**Git Commands Guide**

**COMMITS & STAGING**

* **git add <file name>**  
  Move the specified file to the staging area.
* **git add .**  
  Move all files to the staging area.
* **git commit -m "MSG"**  
  Save changes with a commit message.
* **git init**  
  Initialize a new Git repository.

**LOG & STATUS**

* **git status**  
  Show the current status of the files in the working directory.
* **git log**  
  Show the commit log of the current repository or branch.

**Removing Commits**

* **git reset (--hard || --soft) HEAD~N**  
  Remove the last N commits.
  + --hard: Discard changes in the working directory.
  + --soft: Keep changes in the working directory.
* **git reset <commit ID>**  
  Revert to a specific commit if you deleted a commit by mistake.
* **git reflog**  
  Show the history of all actions, including commits that were reset.
* **git reset --hard <commit ID>**  
  Reset to a specific commit using its ID.

**Checkout**

* **git checkout <commit ID>**  
  Move to the state of the project at the specified commit.
* **git checkout <branch name>**  
  Switch to the specified branch.

**Branching**

**Types of Branches**

* **Local branch**: Exists in your local repository.
* **Remote branch**: Exists in the remote repository.
* **Remote tracking branch**: A copy of the remote branch used for merging.
* **Local tracking branch**: Local branches that track remote branches because of the -u setup stream.

**Branch Commands**

* **git branch**  
  Show all local branches.
* **git branch <branch name>**  
  Create a new branch.
* **git checkout <branch name>**  
  Switch to the specified branch.
* **git checkout -b <branch name>**  
  Create and switch to a new branch in one step.
* **git switch <branch name>**  
  Switch to another branch.
* **git switch -c <branch name>**  
  Create and switch to a new branch in one step.

**Show Branches**

* **git branch**  
  Show local branches only.
* **git branch -a**  
  Show all branches (local and remote).
* **git branch -r**  
  Show remote branches only.

**Publish Branch**

* **git push origin <branch name>**  
  Push a branch to the remote repository.

**Fetch Updates**

* **git fetch**  
  Fetch updates from the remote repository.
* **git fetch origin**  
  Fetch updates from the remote repository named origin.

**Delete Branch**

* **git branch -d <branch name>**  
  Try to delete a specified branch (dry run).
* **git branch -D <branch name>**  
  Force delete a specified branch.
* **git push origin --delete <branch name>**  
  Delete a branch from the remote repository.

**Restore Deleted Branch**

* **git checkout <commit ID>**  
  Move to the last commit of the deleted branch.
* **git reflog**  
  Get the commit ID of the deleted branch.
* **git switch -c <branch name>**  
  Create a new branch from the commit ID.

**Rename Branch**

* **git branch -M <new branch name>**  
  Rename a branch.

**Remote Repository**

* **git remote add origin <repo link>**  
  Link the local repository to a remote repository.
* **git push -u origin main**  
  Push changes to the remote repository and set up tracking.

**Rebase**

* **git switch <branch name>**  
  Switch to the branch you want to rebase.
* **git rebase <source branch>**  
  Rebase the current branch onto the source branch.

**Cherry-pick**

* **git cherry-pick <commit ID>**  
  Apply changes from a specific commit to the current branch.

**Miscellaneous**

* **git ls-files**  
  Show tracked files that have been moved to the staging area.
* **git status**  
  Show the current state of tracked and untracked files.

**Corrections**

1. **git reset <commit ID>**
   * **Correction**: This command is used to move the HEAD to a specific commit, not to restore a deleted commit. To restore a deleted commit, use git reflog to find the commit ID and then git checkout <commit ID>.
2. **git branch -d <branch name>**
   * **Correction**: This command deletes a branch only if it has been merged. If the branch has unmerged changes, use git branch -D <branch name>.
3. **git push origin --delete <branch name>**
   * **Correction**: This command deletes a branch from the remote repository. Ensure you have deleted it locally first using git branch -D <branch name>.
4. **git rebase <source branch>**
   * **Correction**: Ensure you are on the branch you want to rebase before running this command.
5. **git cherry-pick <commit ID>**
   * **Correction**: This command applies the changes from a specific commit to the current branch. Ensure you are on the correct branch before running this command.

**Merging**

قبل أن تقوم بعملية الدمج، يجب أن تكون في الفرع الذي تريد دمج التغييرات فيه. على سبيل المثال، إذا كنت تريد دمج فرع في الفرع الرئيسي (main)، يجب أن تكون في الفرع الرئيسي وتكتب أمر الدمج هناك.

* **git merge <source branch>**  
  دمج التغييرات من الفرع المصدر إلى الفرع الحالي.

**Removing Untracked Files**

* **git clean -dn**  
  تجربة حذف الملفات أو المجلدات غير المتعقبة (dry run).
* **git clean -df**  
  حذف الملفات أو المجلدات غير المتعقبة بشكل فعلي.

**Types of Merge**

* **Fast-forward merge**  
  يحدث عندما تقوم بدمج فرع في الفرع الرئيسي ولم يكن هناك أي تغييرات في الفرع الرئيسي منذ إنشاء الفرع الذي تقوم بدمجه.

**Discarding File Changes**

* **git restore <file name>**  
  تجاهل التغييرات في الملف المحدد.
* **git restore .**  
  تجاهل جميع التغييرات في جميع الملفات.
* **git checkout <file name>**  
  تجاهل التحديثات غير الملتزمة في الملف وإعادته إلى حالة آخر commit.
* **git rm <file name>**  
  حذف الملف المحدد ثم استخدام git commit لحفظ التغييرات.
* **git restore --staged <file name>**  
  إزالة الملف من منطقة التمكين (staging area).
* **git reset <file name>**  
  إزالة الملف من منطقة التمكين (staging area).

**Resetting Commits**

* **git reset HEAD~N**  
  إزالة N عدد من الـ commits من الأعلى.
* **git reset --soft HEAD~N**  
  إزالة الـ commits من منطقة العمل فقط، مع الاحتفاظ بالتغييرات في منطقة التمكين.
* **git reset --hard HEAD~N**  
  إزالة الـ commits من منطقة العمل ومنطقة التمكين، وتجاهل التغييرات المعدلة.

**Stashing**

* **git stash**  
  حفظ التغييرات مؤقتًا.
* **git stash apply**  
  استعادة التغييرات المحفوظة.
* **git stash list**  
  عرض قائمة بالتغييرات المحفوظة.
* **git stash apply <stash ID>**  
  استعادة تغيير محدد باستخدام معرفه.
* **git stash drop <stash ID>**  
  حذف تغيير محدد من القائمة.
* **git stash clear**  
  حذف جميع التغييرات المحفوظة.
* **git stash push -m "message"**  
  حفظ التغييرات مع رسالة توضيحية.
* **git stash pop**  
  استعادة التغييرات المحفوظة وحذفها من القائمة.

**Upstream**

* **git push --set-upstream origin main**  
  إعداد قناة بين المستودع المحلي والمستودع البعيد.
* **git push -u origin main**  
  إعداد قناة بين المستودع المحلي والمستودع البعيد بشكل مختصر.

**Pulling Changes**

* **git pull**  
  يجمع بين git fetch و git merge لجلب التحديثات من المستودع البعيد ودمجها في المستودع المحلي.

**GitHub**

* **git clone <repo link>** 📥  
  نسخ مستودع من المستودع البعيد.