**Git Commands Guide**

**COMMITS & STAGING**

* **git add <file name> 📂  
  بتنقل الملف المحدد لمنطقة التمكين (staging area).**
* **git add . 📂  
  بتنقل كل الملفات لمنطقة التمكين.**
* **git commit -m "MSG" 💾  
  بتحفظ التغييرات مع رسالة توضيحية.**
* **git init 🆕  
  بتبدأ مستودع جديد في Git.**

**LOG & STATUS**

* **git status 📊  
  بتعرض الحالة الحالية للملفات في الدليل العامل.**
* **git log 📜  
  بتعرض سجل الـ commits للمستودع أو الفرع الحالي.**

**Removing Commits**

* **git reset (--hard || --soft) HEAD~N 🔄  
  بتحذف آخر N commits.** 
  + **--hard: بتتجاهل التغييرات في الدليل العامل.**
  + **--soft: بتحتفظ بالتغييرات في الدليل العامل.**
* **git reset <commit ID> ⏪  
  بترجع لcommit معين لو حذفت commit بالغلط.**
* **git reflog 🕒  
  بتعرض تاريخ كل العمليات، بما فيهم الـ commits اللي اتعملهم reset.**
* **git reset --hard <commit ID> 🔄  
  بترجع لcommit معين باستخدام الـ ID بتاعه.**

**Checkout**

* **git checkout <commit ID> ⏩  
  بتنقل لحالة المشروع عند الـ commit المحدد.**
* **git checkout <branch name> 🌿  
  بتتنقل للفرع المحدد.**

**Branching**

**أنواع الفروع**

* **Local branch: 🏠 موجود في المستودع المحلي.**
* **Remote branch: 🌐 موجود في المستودع البعيد.**
* **Remote tracking branch: 🔄 نسخة من الفرع البعيد بتستخدم للدمج.**
* **Local tracking branch: 📡 فروع محلية بتتبع الفروع البعيدة بسبب إعداد الـ -u.**

**أوامر الفروع**

* **git branch 🌿  
  بتعرض كل الفروع المحلية.**
* **git branch <branch name> 🌿  
  بتنشئ فرع جديد.**
* **git checkout <branch name> 🌿  
  بتتنقل للفرع المحدد.**
* **git checkout -b <branch name> 🌿  
  بتنشئ وتتنقل لفرع جديد في خطوة واحدة.**
* **git switch <branch name> 🔄  
  بتتنقل لفرع تاني.**
* **git switch -c <branch name> 🔄  
  بتنشئ وتتنقل لفرع جديد في خطوة واحدة.**

**عرض الفروع**

* **git branch 🌿  
  بتعرض الفروع المحلية بس.**
* **git branch -a 🌿  
  بتعرض كل الفروع (المحلية والبعيدة).**
* **git branch -r 🌿  
  بتعرض الفروع البعيدة بس.**

**نشر الفرع**

* **git push origin <branch name> 🌐  
  بترفع فرع للمستودع البعيد.**

**جلب التحديثات**

* **git fetch 🔄  
  بتجيب التحديثات من المستودع البعيد.**
* **git fetch origin 🔄  
  بتجيب التحديثات من المستودع البعيد اللي اسمه origin.**

**Delete Branch**

* **git branch -d <branch name> 🗑️  
  بتحاول تحذف فرع محدد (dry run).**
* **git branch -D <branch name> 🗑️  
  بتحذف فرع محدد بالقوة.**
* **git push origin --delete <branch name> 🗑️  
  بتحذف فرع من المستودع البعيد.**

**Restore Deleted Branch**

* **git checkout <commit ID> ⏩  
  بتنقل لآخر commit في الفرع المحذوف.**
* **git reflog 🕒  
  بتجيب الـ commit ID للفرع المحذوف.**
* **git switch -c <branch name> 🔄  
  بتنشئ فرع جديد من الـ commit ID.**

**Rename Branch**

* **git branch -M <new branch name> ✏️  
  بتعيد تسمية الفرع.**

**Remote Repository**

* **git remote add origin <repo link> 🌐  
  بتربط المستودع المحلي بمستودع بعيد.**
* **git push -u origin main 🌐  
  بترفع التغييرات للمستودع البعيد وبتعمل إعداد التتبع.**

**Rebase**

* **git switch <branch name> 🔄  
  بتتنقل للفرع اللي عاوز تعمله rebase.**
* **git rebase <source branch> 🔄  
  بتعمل rebase للفرع الحالي على الفرع المصدر.**

**Cherry-pick**

* **git cherry-pick <commit ID> 🍒  
  بتطبق التغييرات من commit معين على الفرع الحالي.**

**Miscellaneous**

* **git ls-files 📂  
  بتعرض الملفات المتعقبة اللي اتنقلت لمنطقة التمكين.**
* **git status 📊  
  بتعرض الحالة الحالية للملفات المتعقبة وغير المتعقبة.**

**Corrections**

1. **git reset <commit ID> ⏪** 
   * **تصحيح: الأمر ده بيستخدم لنقل الـ HEAD لcommit معين، مش لاستعادة commit محذوف. لاستعادة commit محذوف، استخدم git reflog عشان تجيب الـ commit ID وبعدها git checkout <commit ID>.**
2. **git branch -d <branch name> 🗑️** 
   * **تصحيح: الأمر ده بيحذف الفرع بس لو كان مدموج. لو الفرع فيه تغييرات غير مدموجة، استخدم git branch -D <branch name>.**
3. **git push origin --delete <branch name> 🗑️** 
   * **تصحيح: الأمر ده بيحذف الفرع من المستودع البعيد. تأكد إنك حذفته محليًا الأول باستخدام git branch -D <branch name>.**
4. **git rebase <source branch> 🔄** 
   * **تصحيح: تأكد إنك في الفرع اللي عاوز تعمله rebase قبل ما تشغل الأمر ده.**
5. **git cherry-pick <commit ID> 🍒** 
   * **تصحيح: الأمر ده بيطبق التغييرات من commit معين على الفرع الحالي. تأكد إنك في الفرع الصحيح قبل ما تشغل الأمر ده.**

**Merging**

**قبل أن تقوم بعملية الدمج، لازم تكون في الفرع اللي عاوز تدمج التغييرات فيه. يعني لو عاوز تدمج فرع في الفرع الرئيسي (main)، لازم تكون في الفرع الرئيسي وتكتب أمر الدمج هناك.**

* **git merge <source branch> 🔄  
  دمج التغييرات من الفرع المصدر للفرع 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 ⏩  
  بيحصل لما تدمج فرع في الفرع الرئيسي ومافيش أي تغييرات في الفرع الرئيسي من وقت إنشاء الفرع اللي بتدمجه.**

**Discarding File Changes**

* **git restore <file name> 🗑️  
  تجاهل التغييرات في الملف المحدد.**
* **git restore . 🗑️  
  تجاهل كل التغييرات في كل الملفات.**
* **git checkout <file name> ⏪  
  تجاهل التحديثات غير الملتزمة في الملف وإعادته لحالة آخر commit.**
* **git rm <file name> 🗑️  
  حذف الملف المحدد وبعدها تستخدم 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 بس، 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 كل 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 🔄  
  بيجمع 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.**