Here’s a quick guide to the essential Git commands, laid out in sequence, so you can learn the basics in about 30 minutes:

**1. Setup Your Git Environment (Only once)**

* Set your identity (name and email):

bash

Copy code

git config --global user.name "Your Name"

git config --global user.email "your.email@example.com"

**2. Starting a New Repository**

* Initialize a new Git repository:

bash

Copy code

git init

* Clone an existing repository:

bash

Copy code

git clone <repository\_url>

**3. Basic Workflow: Add, Commit, Push**

* **Check Repository Status** (What’s changed?):

bash

Copy code

git status

* **Add Changes to Staging** (To stage specific file changes):

bash

Copy code

git add <file\_name> # Adds a specific file

git add . # Adds all changes in the current directory

* **Commit the Staged Changes** (Take a snapshot):

bash

Copy code

git commit -m "Commit message"

* **Push to Remote Repository** (Sync with online repo):

bash

Copy code

git push origin <branch\_name> # Typically `main` or `master`

**4. Working with Branches**

* **Create a New Branch**:

bash

Copy code

git branch <new\_branch\_name>

* **Switch to a Different Branch**:

bash

Copy code

git checkout <branch\_name>

* **Create and Switch to a New Branch**:

bash

Copy code

git checkout -b <new\_branch\_name>

* **Merge Branches** (Incorporate changes from one branch into another):

bash

Copy code

git checkout <branch\_you\_want\_to\_merge\_into>

git merge <branch\_name>

**5. Sync Changes**

* **Pull Changes from Remote Repository** (Fetch and merge updates from others):

bash

Copy code

git pull

**6. Inspecting the Repository**

* **See Commit History**:

bash

Copy code

git log

* **Show Changes Between Commits or Staged/Unstaged Files**:

bash

Copy code

git diff

git diff <file\_name> # For a specific file

**7. Undoing Changes**

* **Unstage Changes** (Move changes from staging back to working directory):

bash

Copy code

git reset <file\_name>

* **Discard Local Changes** (Revert to the last commit):

bash

Copy code

git checkout -- <file\_name>

**8. Handling Merge Conflicts**

* Git will notify you of conflicts when merging. You will need to manually resolve conflicts in the affected files, then commit the resolved files:

bash

Copy code

git add <resolved\_file>

git commit

**9. Deleting Branches**

* **Delete a local branch**:

bash

Copy code

git branch -d <branch\_name>

* **Delete a remote branch**:

bash

Copy code

git push origin --delete <branch\_name>

**10. Other Useful Commands**

* **Create a .gitignore File** (To ignore specific files):
  + Add a .gitignore file and specify patterns of files you don’t want Git to track.
* **Check Remote Repositories** (List remote URLs):

bash

Copy code

git remote -v