

Certainly! Here's the list with the Git commands highlighted for easier identification:

1. Basic Git Commands

- **Setup & Configuration:**
 - **`git config --global user.name "Your Name"`**: Set global username.
 - **`git config --global user.email "your.email@example.com"`**: Set global email.
 - **`git config --global core.editor "code --wait"`**: Set default editor.
- **Repository Creation & Cloning:**
 - **`git init`**: Initialize a new Git repository.
 - **`git clone <repo-url>`**: Clone an existing repository.
- **File Tracking:**
 - **`git add <file>`**: Stage a specific file for commit.
 - **`git add .`**: Stage all changes in the directory.
- **Committing Changes:**
 - **`git commit -m "commit message"`**: Commit staged changes with a message.
 - **`git commit -am "message"`**: Commit all changes to tracked files with a message.
- **Viewing Changes:**
 - **`git status`**: Check the status of your working directory.
 - **`git diff`**: View changes between commits or working directory and index.
 - **`git log`**: View commit history.
 - **`git log --oneline`**: View a condensed commit history.

2. Branching & Merging

- **Branch Management:**
 - **`git branch`**: List all branches.
 - **`git branch <branch-name>`**: Create a new branch.
 - **`git checkout <branch-name>`**: Switch to a branch.
 - **`git checkout -b <branch-name>`**: Create and switch to a new branch.
- Note:** Check out can be replace by **switch**

- **Merging & Rebasing:**
 - ***git merge <branch-name>***: Merge a branch into the current branch.
 - ***git rebase <branch-name>***: Rebase the current branch onto another branch.
 - ***git rebase -i <commit>***: Interactive rebase for editing commits.
- **Branch Deletion:**
 - ***git branch -d <branch-name>***: Delete a local branch.
 - ***git push origin --delete <branch-name>***: Delete a remote branch.

3. Remote Repositories

- **Remote Management:**
 - ***git remote -v***: List remote repositories.
 - ***git remote add <name> <url>***: Add a new remote repository.
 - ***git remote remove <name>***: Remove a remote repository.
- **Pushing & Pulling Changes:**
 - ***git push <remote> <branch>***: Push changes to a remote repository.
 - ***git pull <remote> <branch>***: Fetch and merge changes from a remote repository.
 - ***git fetch <remote>***: Fetch changes from a remote repository without merging.

4. Conflict Resolution

- **Handling Merge Conflicts:**
 - After a conflict occurs during ***git merge*** or ***git pull***, manually resolve conflicts in the files.
 - ***git add <file>***: Stage resolved files.
 - ***git commit***: Complete the merge commit.
- **Abort Merge or Rebase:**
 - ***git merge --abort***: Abort the merge process.
 - ***git rebase --abort***: Abort the rebase process.

5. Tagging

- **Creating & Managing Tags:**

- ***git tag <tag-name>***: Create a lightweight tag.
- ***git tag -a <tag-name> -m "message"***: Create an annotated tag with a message.
- ***git tag -d <tag-name>***: Delete a local tag.
- ***git push origin <tag-name>***: Push a tag to the remote repository.
- ***git push origin --tags***: Push all tags to the remote repository.

6. Stashing

- **Stashing Changes:**
 - ***git stash***: Stash changes in a dirty working directory.
 - ***git stash pop***: Apply the most recent stash and remove it from the stash list.
 - ***git stash list***: List all stashes.
 - ***git stash apply <stash>***: Apply a specific stash without removing it.

7. Reverting & Resetting

- **Reverting Changes:**
 - ***git revert <commit>***: Create a new commit that undoes changes from a previous commit.
- **Resetting Changes:**
 - ***git reset <file>***: Unstage a file.
 - ***git reset --hard <commit>***: Reset the working directory to a specific commit, discarding all changes.

8. Advanced Concepts

- **Submodules:**
 - ***git submodule add <url> <path>***: Add a submodule.
 - ***git submodule update --init --recursive***: Initialize and update submodules.
- **Rewriting History:**
 - ***git filter-branch***: Rewrite commit history (use with caution).
- **Bisecting:**
 - ***git bisect start***: Begin bisecting to find a commit that introduced a bug.
 - ***git bisect good <commit>***: Mark a commit as good.
 - ***git bisect bad <commit>***: Mark a commit as bad.

- ***git bisect reset***: End the bisect session.

9. Hooks & Automation

- **Git Hooks:**

- ***pre-commit***: Hook executed before a commit.
- ***post-commit***: Hook executed after a commit.
- Located in the `.git/hooks` directory.

10. Git Configuration

- **Local vs. Global Configuration:**

- Use ***git config --local*** for repository-specific settings.
- Use ***git config --global*** for user-specific settings.

Feel free to ask if you need more details on any specific command or concept!

create a new repository on the command line

```
echo "# Test-Repo" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/G0809/Test-Repo.git
git push -u origin main
```

push an existing repository from the command line

```
git remote add origin https://github.com/G0809/Test-Repo.git
git branch -M main
git push -u origin main
```