Git Commands

Command	Description
cd <path>	Change the current directory (folder) to the specified path.
git init	Create a new repository in the current directory.

Configure git:

git config --global user.name <username>	Set the username for every repository that will be created.
git config --global user.name	Check the current global username.
git config user.name <username>	Set the username for the current repository (replace the global username).
git config user.name	Check the username for the current repository.
git config --global user.email <email>	Set the email for every repository that will be created.
git config --global user.email	Check the current global email.
git config user.email <email>	Set the email for the current repository (replace the global email).
git config user.email	Check the email for the current repository.

Git Commands

Command	Description
Workflow:	
git status	Display a brief summary of the status of the repository, the files that have been modified and the changes that will be committed.
git add .	Add the current changes to the staging area to be committed with the next commit.
git commit –m “<commit_message>”	Create a commit and assign the message as the commit message.
git reset <file_name>	Remove the changes in the file from the staging area. The name of the file must include the extension.
git diff	Check the differences between files in the staging area and their current state in the working directory.
Commit History:	
git log	Display the commit history.
git log –p -<number>	Display the commit history but only include this number of recent commits.
git log --pretty=oneline	Display the commit history but only show the commit ID and message.
git log –relative-date	Display the commit history showing relative dates for the commits (e.g 2 hours ago).

Git Commands

Command	Description
Commits:	
git show <commit_id>	Show data for this specific commit.
git log --until=“date”	Show commits made until this date.
git log --after=“date”	Show commits made after this date.
git log --before=“date”	Show commits made before this date.
git log --since=“date”	Show commits made since this date.
git commit --amend	Edit the message of the last commit.
git rebase –i HEAD~3	Display the last three commits made to squash them or modify them.
git reset --soft HEAD~1	Remove the last commit but keep the changes in the working directory.
git reset --hard HEAD~1	Remove the last commit and also remove the changes from the working directory.
git reset --hard <commit_id>	Remove commits made after this commit and discard the changes.

Branches:

Command	Description
git branch <branch_name>	Create a branch and assign this name.
git branch	List all existing branches in the repository.
git checkout <branch_name>	Change the current branch to this branch.
git log --graph	Display the commit history as a graph.
git diff <branch1_name> <branch2_name>	Display the differences between these two branches.
git branch -D <branch_name>	Remove the branch even if it hasn’t been merged with another branch.
git branch -d <branch_name>	Remove the branch if it has been merged with another branch so no work is lost.
git log --graph --all	Show the commit history for all branches as a graph.
git merge <branch_to_merge>	Merge the specified branch INTO the current branch.
git cherry-pick <commit_id>	Choose a specific commit to add to the branch. (Use with caution!)

Command	Description
git merge --abort	Abort a merge if it runs into conflicts or if you need to revert it **and you haven't created a commit for the merge**.
git revert -m 1 <merge_commit_id>	Abort a merge **after the merge commit has been created**. This will undo the merge commit.

Files:

ls	List all the files in the current directory.
git rm <file_name>	Remove the file from the directory. You must include the file extension.
git mv <initial_name> <new_name>	Rename file. You must include the file extension for the current and new names.

Stash:

git stash	Stash current changes.
git stash list	Display all currently saved stashes.
git stash apply	Restore the most recently stashed changes.
git stash <stash_id>	Restore this specific stash.

Command	Description
git stash drop <stash_id>	Remove the specified stash from the list.

“Never memorize something that you can look up.”
— **Albert Einstein**