GitHub for Windows htps://windows.github.com

GitHub for Mac htps://mac.github.com

For Linux and Solaris platforms, the latest release is available on the official Git web site.

Git for All Platforms htp://git-scm.com

**SETUP Configuring user information used across all local repositories**

git config --global user.name “[first name last name]”

set a name that is identifiable for credit when review version history

git config --global user. email “[valid-email]”

set an email address that will be associated with each history marker

git config --global color.ui

auto set automatic command line coloring for Git for easy reviewing

**Getting & Creating Projects**

| **Command** | **Description** |
| --- | --- |
| git init | Initialize a local Git repository |
| git clone [url]  git clone ssh://git@github.com/[username]/[repository-name].git | Create a local copy of a remote repository |

**Basic Snapshotting**

| **Command** | **Description** |
| --- | --- |
| git status | Check status |
| git add [file-name.txt] | Add a file to the staging area |
| git add -A | Add all new and changed files to the staging area |
| git reset [file] | Un stage a file while retaining the changes in working directory |
| git diff | diff of what is changed but not staged |
| git diff --staged | diff of what is staged but not yet commited |
| git commit -m "[commit message]" | Commit changes |
| git rm -r [file-name.txt] | Remove a file (or folder) |

**Branching & Merging**

| **Command** | **Description** |
| --- | --- |
| git branch | List branches (the asterisk denotes the current branch) |
| git branch -a | List all branches (local and remote) |
| git branch [branch name] | Create a new branch |
| git branch -d [branch name] | Delete a branch |
| git push origin --delete [branchName] | Delete a remote branch |
| git checkout -b [branch name] | Create a new branch and switch to it |
| git checkout -b [branch name] origin/[branch name] | Clone a remote branch and switch to it |
| git checkout [branch name] | Switch to a branch |
| git checkout - | Switch to the branch last checked out |
| git checkout -- [file-name.txt] | Discard changes to a file |
| git merge [branch name] | Merge a branch into the active branch |
| git merge [source branch] [target branch] | Merge a branch into a target branch |
| git stash | Stash the changes in a dirty working directory away |
| git stash clear | Remove all the stashed states |
| git log | show all commits in the current branch’s history |

**Sharing & Updating Projects**

| **Command** | **Description** |
| --- | --- |
| git push origin [branch name] | Push a branch to your remote repository |
| git push -u origin [branch name] | Push changes to remote repository (and remember the branch) |
| git push | Push changes to remote repository (remembered branch) |
| git push origin --delete [branch name] | Delete a remote branch |
| git pull | Update local repository to the newest commit |
| git pull origin [branch name] | Pull changes from remote repository |
| git remote add origin ssh://git@github.com/[username]/[repository-name].git | Add a remote repository |
| git remote set-url origin ssh://git@github.com/[username]/[repository-name].git | Set a repository's origin branch to SSH |
| git fetch [alias] | fetch down all the branches from that Git remote |
| git merge [alias]/[branch] | merge a remote branch into your current branch to bring it up to date |

**Inspection & Comparison**

| **Command** | **Description** |
| --- | --- |
| git log | View changes |
| git log --summary | View changes (detailed) |
| git log branchB..branchA | show the commits on branchA that are not on branchB |
| git log --follow [file] | show the commits that changed file, even across renames |
| git diff branchB...branchA | show the diff of what is in branchA that is not in branchB |
| git diff [source branch] [target branch} | Preview changes before merging |
| git show [SHA] | show any object in Git in human-readable format |

**TRACKING PATH CHANGES**

Versioning file removes and path changes

git rm [file]

delete the file from project and stage the removal for commit

git mv [existing-path] [new-path]

change an existing file path and stage the move

git log --stat -M

show all commit logs with indication of any paths that moved

**REWRITE HISTORY**

Rewriting branches, updating commits and clearing history

git rebase [branch]

apply any commits of current branch ahead of specified one

git reset --hard [commit]

clear staging area, rewrite working tree from specified commit

IGNORING PATTERNS

Preventing unintentional staging or committing of files

logs/

\*.notes

pattern\*/

Save a file with desired paternsas .gitignore with either direct stringmatches or wildcard globs.

git config --global core.excludesfile [file]

system wide ignore patern for all local repositories

**TEMPORARY COMMITS**

Temporarily store modified, tracked files in order to change branches

git stash Save modified and staged changes

git stash list list stack-order of stashed file changes

git stash pop write working from top of stash stack

git stash drop discard the changes from top of stash stack