Git Setup

See Git Location: `which git`

Get Version of Git: `git --version`

Create Alias for Git Status: `git config --global alias.st status`

Add or Edit gitignore: `nano .gitignore`

Track Empty Directory: `touch dir/.gitkeep`

Help: `git help`

Crucial Git Commands

Initialize Git: `git init`

Set Everything to Commit: `git add .`

Set Single File to Commit: `git add index.php`

Commit Changes: `git commit -m "COMMITMESSAGE"`

Add and Commit: `git commit -am "COMMITMESSAGE"`

Remove File from Git: `git rm index.php`

Update All Changes: `git add -u`

Remove File and Do Not Track Anymore: `git rm --cached index.php`

Move or Rename Files: `git mv index.php dir/index-two.php`

Undo Modifications to Last Commit: `git checkout -- index.php`

Restore File From Specific Commit: `git checkout COMMITNAME -- index.php`

Reset

Go Back to Previous Commit: `git revert COMMITNAME`

Soft Reset (HEAD only): `git reset --soft COMMITNAME`

Undo Previous Commit: `git reset --soft HEAD~`

Mixed Reset (HEAD and staging): `git reset --mixed COMMITNAME`

Hard Reset (HEAD, staging, and dir): `git reset --hard COMMITNAME`

Update & Delete

Test the Delete of Untracked Files: `git clean -n`

Delete Untracked Files: `git clean -f`

Unstage: `git reset HEAD index.php`

Commit to Most Recent Commit: `git commit --amend -m "COMMITMESSAGE"`

Update Most Recent Commit Message: `git commit --amend -m"COMMITMESSAGE"`

Branch

List Branches: `git branch`

Create New Branch: `git branch BRANCHNAME`

Change Branch: `git checkout BRANCHNAME`

Create and Change to New Branch: `git checkout -b BRANCHNAME`

Rename Branch: `git branch -m BRANCHNAME BRANCHNAMETWO` or: `git branch --move BRANCHNAME BRANCHNAMETWO`

Show Branches Merged with Current Branch: `git branch --merged`

Delete Merged Branch: `git branch -d BRANCHNAME` or: `git branch --deleteBRANCHNAME`

Delete Unmerged Branch: `git branch -D BRANCHNAME`

Merge

Simle Merge: `git merge BRANCHNAME`

Merge to Master: `git merge --ff-only BRANCHNAME`

Stop Merge: `git merge --abort`

Stash

Move to Stash: `git stash save "COMMITMESSAGE"`

Show Stash: `git stash list`

List Stash Stats: `git stash show stash@{0}`

Show Stash Changes: `git stash show -p stash@{0}`

Use Custom Stash Item and Drop It: `git stash pop stash@{0}`

Use Custom Stash Item and Do Not Drop It: `git stash apply stash@{0}`

Delete Stash Item: `git stash drop stash@{0}`

Delete Stash: `git stash clear`

Compare

Compare Modified Files: `git diff`

Compare Modified Files and Highlight Changes: `git diff --color-words index.php`

Compare Modified Files Within Staging: `git diff --staged`

Compare Branches: `git diff master..BRANCHNAME`

Compare Branches Like Above: `git diff --color-words master..BRANCHNAME^`

Compare commits:

```
git diff COMMITNAME
```

```
git diff COMMITNAME..HEAD
```

```
git diff COMMITNAME..COMMITNAMETWO
```

Compare Commits of File: `git diff COMMITNAME index.php` `git diff COMMITNAME..COMMITNAMETWO index.php`

Compare Without Caring About Spaces: `git diff -b COMMITNAME..HEAD` or: `git diff --ignore-space-change COMMITNAME..HEAD`

Compare Without Caring About All Spaces: `git diff -w COMMITNAME..HEAD` or: `git diff --ignore-all-space COMMITNAME..HEAD`

Useful Comparison: `git diff --stat --summary COMMITNAME..HEAD`

Blame: `git blame -L10,+1 index.php`

Log

Show Commits: `git log`

Show Summary of Commits: `git log --oneline`

Show Summary of SHA-1 Commits: `git log --format=oneline`

Show Summary of Last Three Commits: `git log --oneline -3`

Show All Changes: `git log -p`

Show Every Commit Since Special Commit for Custom File: `git log COMMITNAME.. index.php`

Show Changes of Every Commit Since Special Commit for Custom File: `git log -p COMMITNAME.. index.php`

Show Stats and Summary of Commits: `git log --stat --summary`

Show History of Commits as Graph: `git log --graph`

Show Summary History of Commits as Graph: `git log --oneline --graph --all --decorate`

Collaborate

Show Remote: `git remote`

Show Remote Details: `git remote -v`

Add Remote Origin from GitHub Project: `git remote add origin https://github.com/user/project.git`

Add Remote Origin from Existing Empty Project: `git remote add origin ssh://root@123.123.123.123/path/to/repository/.git`

Remove Origin: `git remote rm origin`

Show Remote Branches: `git branch -r`

Show All Branches: `git branch -a`

Compare: `git diff origin/master..master`

Push: `git push -u origin master`

Push to Default: `git push origin master`

Fetch: `git fetch origin`

Pull: `git pull`

Pull Branch: `git pull origin BRANCHNAME`

Merge Fetched Commits: `git merge origin/master`

Clone to Local: `git clone https://github.com/user/project.git` or: `git clone ssh://user@domain.com/~/dir/.git`

Clone to Local Folder: `git clone https://github.com/user/project.git ~/dir/folder`

Clone Specific Branch to Local: `git clone -b BRANCHNAME \https://github.com/user/project.git\`

Delete Remote Branch: `git push origin :BRANCHNAME` or: `git push origin --delete BRANCHNAME`

Archive

Create Zip-Archive: `git archive --format zip --output filename.zip master`

Export/Write Custom Log to File: `git log --author=sven --all &gt; log.txt`

Releases & Version Tags

Show All Released Versions: `git tag`

Show All Released Versions with Comments: `git tag -l -n1`

Create Release Version: `git tag v1.0.0`

Create Release Version with Comment: `git tag -a v1.0.0 -m'COMMITMESSAGE'`

Checkout a Specific Release Version: `git checkout v1.0.0`