

Twitter Thread by Pratham



Pratham

@PrasoonPratham



A git cheat sheet for all your needs!

<https://t.co/H7QdOrthwV>

Git Cheat Sheet

For more awesome cheat sheets visit rebellabs.org/

by ZERO TURNAROUND

<h3>Create a Repository</h3> <p>From scratch -- Create a new local repository</p> <pre>\$ git init [project name]</pre> <p>Download from an existing repository</p> <pre>\$ git clone my_url</pre> <h3>Observe your Repository</h3> <p>List new or modified files not yet committed</p> <pre>\$ git status</pre> <p>Show the changes to files not yet staged</p> <pre>\$ git diff</pre> <p>Show the changes to staged files</p> <pre>\$ git diff --cached</pre> <p>Show all staged and unstaged file changes</p> <pre>\$ git diff HEAD</pre> <p>Show the changes between two commit ids</p> <pre>\$ git diff commit1 commit2</pre> <p>List the change dates and authors for a file</p> <pre>\$ git blame [file]</pre> <p>Show the file changes for a commit id and/or file</p> <pre>\$ git show [commit]:[file]</pre> <p>Show full change history</p> <pre>\$ git log</pre> <p>Show change history for file/directory including diffs</p> <pre>\$ git log -p [file/directory]</pre>	<h3>Working with Branches</h3> <p>List all local branches</p> <pre>\$ git branch</pre> <p>List all branches, local and remote</p> <pre>\$ git branch -av</pre> <p>Switch to a branch, my_branch, and update working directory</p> <pre>\$ git checkout my_branch</pre> <p>Create a new branch called new_branch</p> <pre>\$ git branch new_branch</pre> <p>Delete the branch called my_branch</p> <pre>\$ git branch -d my_branch</pre> <p>Merge branch_a into branch_b</p> <pre>\$ git checkout branch_b</pre> <pre>\$ git merge branch_a</pre> <p>Tag the current commit</p> <pre>\$ git tag my_tag</pre>	<h3>Make a change</h3> <p>Stages the file, ready for commit</p> <pre>\$ git add [file]</pre> <p>Stage all changed files, ready for commit</p> <pre>\$ git add .</pre> <p>Commit all staged files to versioned history</p> <pre>\$ git commit -m "commit message"</pre> <p>Commit all your tracked files to versioned history</p> <pre>\$ git commit -am "commit message"</pre> <p>Unstages file, keeping the file changes</p> <pre>\$ git reset [file]</pre> <p>Revert everything to the last commit</p> <pre>\$ git reset --hard</pre>	<h3>Synchronize</h3> <p>Get the latest changes from origin (no merge)</p> <pre>\$ git fetch</pre> <p>Fetch the latest changes from origin and merge</p> <pre>\$ git pull</pre> <p>Fetch the latest changes from origin and rebase</p> <pre>\$ git pull --rebase</pre> <p>Push local changes to the origin</p> <pre>\$ git push</pre> <h3>Finally!</h3> <p>When in doubt, use git help</p> <pre>\$ git command --help <p>Or visit https://training.github.com/ for official GitHub training.</p> </pre>
---	---	---	---

```

graph LR
    WD[Working Directory] -- add --> S[Staging (index)]
    S -- commit --> LR[Local Repository]
    LR -- push --> RR[Remote Repository]
    RR -- fetch --> LR
    LR -- pull --> LR
    LR -- reset --> S
    LR -- "reset [commit]" --> S
  
```

BROUGHT TO YOU BY