

# GIT CHEAT SHEET

## GIT

- Git is a distributed revision control and source code management system with an emphasis on speed.
- it is repository which is used to manage projects, set of files as they changes over the time.
- Using `git` every code change or commit you get latest development code for the project.

## GIT OPERATIONS & COMMANDS

### Git Configurations

- Initial config of username, email and code highlighting (optional) is to be performed.
- `$git config --global user.name "firstname lastname"`
- `$git config --global user.email "abc123@abc.com"`
- `$git config --global color.ui true` (enables code highlights)
- `$git config --list`

### Initialize

- You have to initialize by using 'init'
- To know the status run the 'status' command
- `$git init`
- `$git status`

### Create/Add files

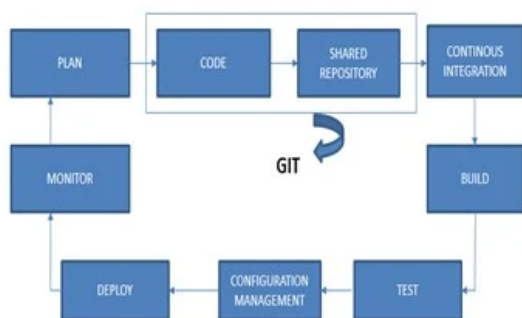
- To add a file: `$git add <filename>`
- To add multiple files: `$git add <filename> <2nd filename>`
- To add all updated files: `$git add -all` (use -A instead of -all too)
- To remove files: `$git rm -r <filename>`

### Commit changes

- To pass a message, use 'commit' and '-m': `$git commit -m "body_of_message"`
- Amend lets you amend the last commit or the last message: `$git commit --amend -m "new_message"`

### Push and Pull

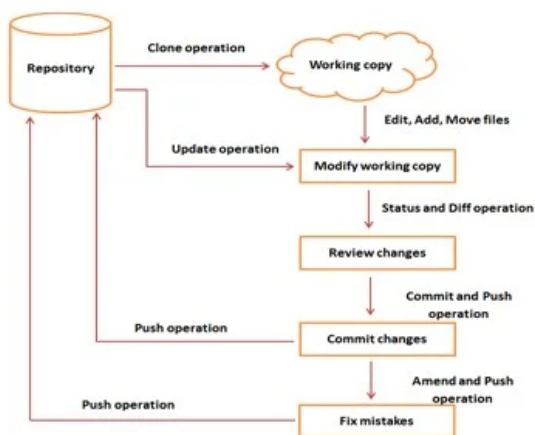
- A remote repository typically represents a remote server or a git server: Create a remote repository via github  
"`https://github.com/YourUsername/appname.git`"
- To add a link: `$git remote add origin <link>`
- Pushing files: `$git push -u origin master`
- To clone file: `$git clone <clone>`



## Version Control

- It is the management of changes to the code, documents, programs, large sites and other info.
- The changes are termed as versions.
- Version control system is used (VCS)
- The functions are:
  - Allows developers to work simultaneously.
  - Does not allow overwriting each other's changes.
  - Maintains a history of every version.

2 types of VCS - centralized and distributed. Git is distributed



## GIT & GITHUB

It is a VCS that supports distributed nonlinear workflows by providing data assurance for developing quality software.

### Features:

- Distributed**- distributed development of code
- compatible**- with existing systems and protocols
- Non-linear**- non linear development of code
- Branching**- easy to create and merge branches
- Lightweight**- lossless compression
- Reliable**- not viable to loss of data upon crashes
- Secure**- SHA1 and checksum are used
- Economical**- free

## Branching and Merging

Command	Description
<code>git branch</code>	List branches
<code>git branch -a</code>	List all branches
<code>git branch [branch name]</code>	Create a new branch
<code>git branch -d [branch name]</code>	Delete branch
<code>git push origin --delete [branchName]</code>	Delete a remote branch
<code>git checkout -b [branch name]</code>	Create a new branch and switch to it
<code>git checkout -b [branch name] origin/[branch name]</code>	Clone a remote branch and switch to it
<code>git checkout [branch name]</code>	Switch to a branch
<code>git checkout -</code>	Switch to the branch last checked out
<code>git checkout -- [file-name.txt]</code>	Discard changes to a file
<code>git merge [branch name]</code>	Merge a branch into the active branch
<code>git stash</code>	Stash changes in a dirty working directory
<code>git stash clear</code>	Remove all stashed entries

## Updating Projects

Commands	Description
<code>git push origin [branch name]</code>	Push a branch to your remote repository
<code>git push -u origin [branch name]</code>	Push changes to remote repository (-u remembers the branch for next use)
<code>git push origin --delete [branch name]</code>	Delete a remote branch
<code>git pull</code>	Update local repository to the newest commit
<code>git pull origin [branch name]</code>	Pull changes from remote repository
<code>git remote add origin ssh://git@github.com:[username]@[repository-name].git</code>	Add a remote repository
<code>git remote set-url origin ssh://git@github.com:[username]@[repository-name].git</code>	Set a repository's origin branch to SSH

## Inspection and Comparison

Command	Comparison
<code>git log</code>	View changes
View changes	View changes (detailed)
<code>git diff [source branch] [target branch]</code>	Preview changes before merging