

Git Cheat Sheet

Git Configuration

<i>command</i>	<i>description</i>
git config --global user.name "Katia Oleinik"	
git config --global user.email "koleinik@bu.edu"	
git config --global cor.editor "nano"	
git config --list [--global / --local]	list current settings
git config --list --show-origin	display path to config files

Getting help

<i>command</i>	<i>description</i>
git help <i>verb</i>	full manpage for <i>verb</i>
man git- <i>verb</i>	full manpage for <i>verb</i>
git <i>verb</i> -h	concise help for <i>verb</i>

Creating Local Repository

<i>command</i>	<i>description</i>
git init <i>dirname</i>	create a new empty repository
git init	a new git repo in an existing folder
git clone /project/scv/dirname	clone local repository
git clone https://github.com/bu-rcls/newpkg.git	clone remote repository

Exploring git repository

<i>command</i>	<i>description</i>
tree .git	list content of .git directory
git ls-tree master .	list content of a git tree object
git status	check status of your repository
git status -sb	show status concisely
git log	show commit log
git log --3	show last 3 commits
git log --graph	draw branch graph on the left
git log --oneline	compact log
git log -- file1	list commits with specific file modified
git log --author="Katia Oleinik"	filter by author
git log --after="2019-1-30"	filter by date
git log --grep="fix"	search commit messages
git config --global alias.lg "log --oneline"	create an alias
git show ae12375	show info about specific commit

Git Commit Workflow

<i>command</i>	<i>description</i>
git add <i>file1 file2</i>	add files to staging area
git add .	add all changes in directory (no deleted files)
git add -A	add all new, modified and delete files to staging area
git commit -m "Initial commit"	create commit with short message
git commit	create a commit (editor will open)
git tag -a v2.15 ae12375	Add a tag to a specific commit

View file source in a commit

<i>command</i>	<i>description</i>
git show HEAD: <i>filename</i>	view source of the file in the last commit
git show 0721696: <i>filename</i>	view source in a specific commit
git annotate <i>filename</i>	show who made changes to the file

Travelling in time

<i>command</i>	<i>description</i>
git checkout 0721696	change state to a specific commit
git checkout master	go back to the present time

Remote repository workflow

<i>command</i>	<i>description</i>
git remote add origin <i>url</i>	connect local and remote repositories
git push origin master	push changes to remote repository
git pull origin master	pull changes to remote repository

Resolving conflicts

1. Make commit
2. git push origin master (results in error)
3. git pull origin master
4. If editor is opened edit merge commit message. Otherwise edit a file, add and commit it
5. git push origin master

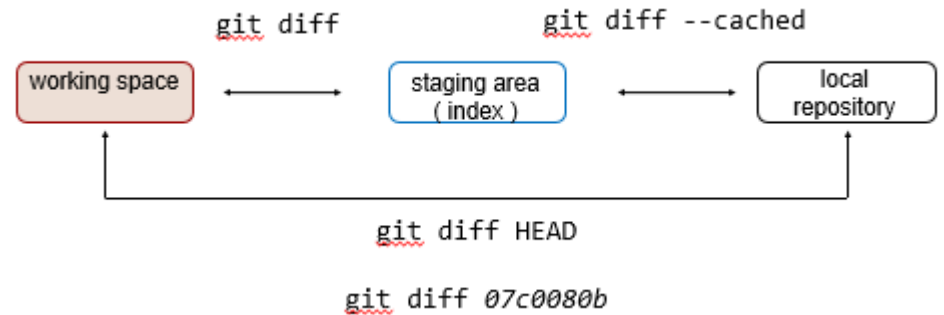
Branches

<i>command</i>	<i>description</i>
<code>git branch</code>	list all existing branches
<code>git branch --list</code>	list all branch names
<code>git branch -a</code>	list remote and local branches
<code>git branch -v</code>	verbose output
<code>git branch <i>branchName</i></code>	create a new branch
<code>git branch -d <i>branchName</i></code>	delete branch (only if it's merged)
<code>git branch -D <i>branchName</i></code>	delete branch (even if not merged)
<code>git checkout <i>branchName</i></code>	create a new branch
<code>git checkout -b <i>branchName</i></code>	create and checkout a new branch
<code>git merge <i>branchName</i></code>	merge <i>branchName</i> into current branch

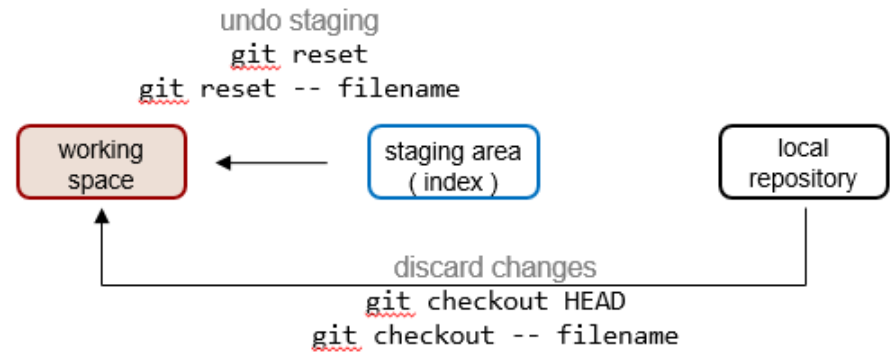
Stash

<i>command</i>	<i>description</i>
<code>git stash</code>	Temporary store modified tracked files
<code>git stash pop</code>	restores stashed files
<code>git stash list</code>	list all stashed changes
<code>git stash drop</code>	discard most recent shash changes

Exploring differences/changes



Discard changes



remove file from staging area
`git reset HEAD -- /path/to/file`

unstage all files
`git reset`