Git Cheat Sheet

Git Configuration

command	*description*
`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

command	*description*
`git help` *verb*	full manpage for *verb*
`man git-`*verb*	full manpage for *verb*
`git` *verb* `-h`	concise help for *verb*

Creating Local Repository

command	*description*
`git init` *dirname*	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-rcs/newpkg.git`	clone remote repository

Exploring git repository

command	*description*
`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

command	*description*
`git add` *file1 file2*	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

command	*description*
`git show HEAD:`*filename*	view source of the file in the last commit
`git show 0721696:`*filename*	view source in a specific commit
`git annotate` *filename*	show who made changes to the file

Travelling in time

command	*description*
`git checkout 0721696`	change state to a specific commit
`git checkout master`	go back to the present time

Remote repository workflow

command	*description*
`git remote add origin` *url*	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`

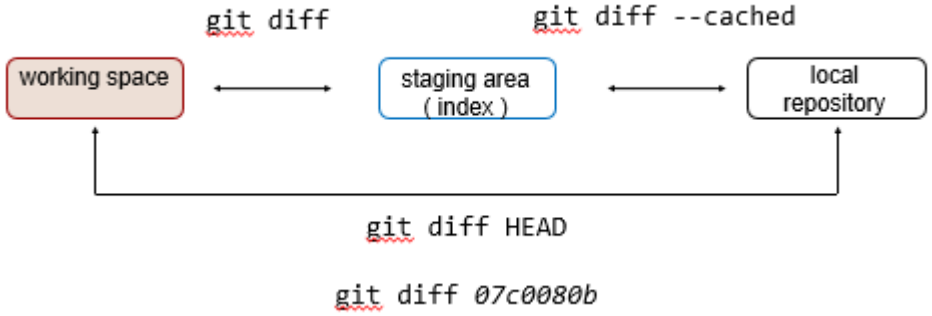
Branches

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

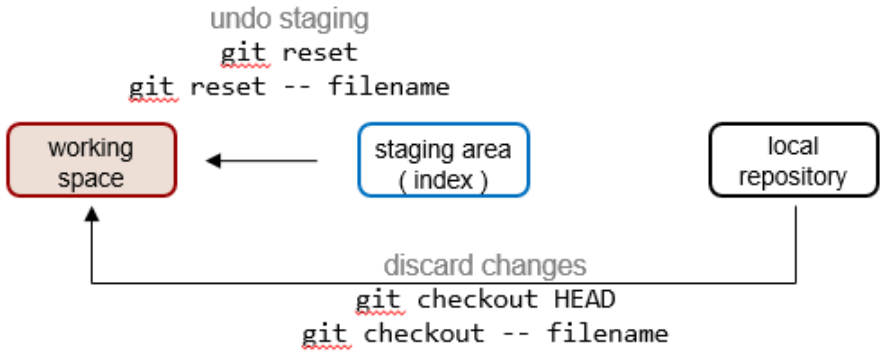
Stash

command	*description*
`git stash`	Temporary store modified tracked files
`git stash pop`	restores stashed files
`git stash list`	list all stashed changes
`git stash drop`	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`