git Quick Reference

Blake Bourque - 2015

Note: *italiized* words are parameters and *[italicized]* words in square brackets are optional parameters

Getting Help:

git help *command*
git *command* --help

Git Setup - Used to set author on commits

git config --global user.name "*Your Name*"
git config --global user.email "*you@ex.com*"

Repository creation:

git init	Create a repository in the current directory
git clone *url*	Clone a remote repository into a subdirectory

File operations:

git add *path*	Add file or files in directory recursively
-p	Add parts of files
git rm *path*	Remove file or directory from the working tree
-f	Force deletion of file(s) from disk
git mv *path destination*	Move file or directory to new location
-f	Overwrite existing destination files
git checkout *[rev] file*	Restore file from current branch or revision
-f	Overwrite uncommitted local changes

Working tree:

git status	Show status of the working tree
git diff *[path]*	Show diff of changes in the working tree
git diff HEAD *path*	Show diff of stages and unstaged changes
git add *path*	Stage file(s) for commit
git reset HEAD *path*	Unstage file for commit
git commit	Commit files that have been staged (via git-add)
-a	Automatically stage all modified files
git reset --soft HEAD^	Undo commit & keep changes in the working tree
git reset --hard HEAD^	Reset the working tree to the last commit
git clean	Clean unknown files from the working tree

Examining History:

git log *[path]*	View commit log, optionally for specific path
git log *[from[..to]]*	View commit log for a given revision range
--stat	List diffstat for each revision
-S'pattern'	Search history for changes matching pattern
git blame *[file]*	Show file annotated with line modifications

Remote repositories - *remotes*:

git fetch *[remote]*	Fetch changes from a remote repository
git pull *[remote]*	Fetch and **merge** changes from a remote repository
git push *[remote]*	Push changes to a remote repository
git remote -v	List remote repositories
git remote add *remote_name url*	Add remote to list of tracked repositories

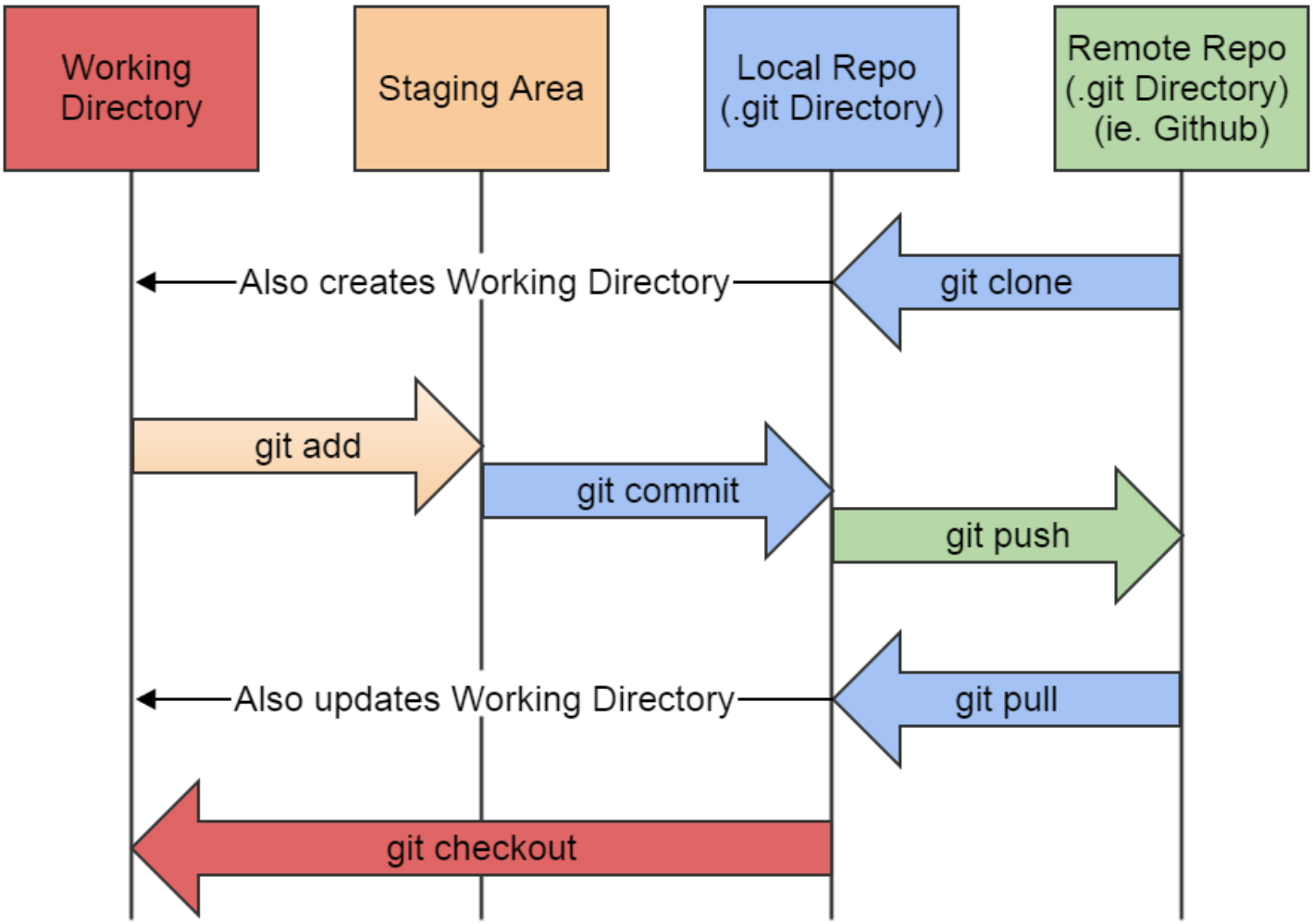
Branches:

git checkout *branch*	Switch working tree to branch
-b *branch*	Create branch before switching to it
git branch	List local branches
git merge *branch*	Merge changes from branch into current branch

Tags:

git tag *name [revision]*	Create tag for a given revision
-l *[pattern]*	List tags, optionally matching pattern

Visualizing git commands



Example Flows

Getting existing repository:

git clone *url*	Download and checkout a remote repository

Creating a new repository:

<start work>	Make some bugs, or create a feature
git init	Create a repository in current directory
git add -p *glob*	Stage file(s) for commit
git commit	Create a commit from staged files
git remote add *remote_name url*	Add remote to list of tracked repositories
git push -u origin master	Send the changes to remote repository

Working on a Repo:

git pull	Get the latest version with other changes
<make changes>	Make some bugs, or add a feature
git add -p *glob*	Stage file(s) for commit
git commit	Create a commit from staged files
git push	Send the changes back to original repository

Helpful Hints:

use git-number to save typing	https://github.com/holygeek/git-number
create aliases and use them	https://git.io/vCIBR
use ssh keys with passphrases	https://help.github.com/articles/generating-ssh-keys/
checkout tldr-manpages	https://github.com/tldr-pages/tldr
ignore binary and compiled files	add them to a .gitignore file https://git-scm.com/docs/gitignore
use cygwin on windows	get a linux terminal in windows https://cygwin.com/