

Switch Branches

<code>git checkout <branch></code>	switches to branch
<code>git checkout -b <branch></code>	create a new branch that forks off of current branch
<code>git branch -a</code>	see all branches
<code>git stash</code>	store unstaged changes
<code>git stash pop</code>	restore unstaged changes

Commit Changes

<code>git status</code>	see current status of documents within git repo
<code>git log</code>	see past commits
<code>git diff <file></code>	see changes made to specified file
<code>git add (-u <files>)</code>	stage changed files (-u adds all tracked files)
<code>git commit -m "<message>"</code>	commit files
<code>git push origin <branch name></code>	push commits to remote repo

Undo Git Commands

<code>git reset --hard</code>	revert changes back to previous commit
<code>git reset --hard HEAD~3</code>	go back 3 commits
<code>git commit --amend</code>	change most recent commit message
<code>git push origin <branch name> --force</code>	change most recent commit message after a push

Rebase

1	<code>git checkout master</code>
2	<code>git pull</code> to get the most recent version of master
3	<code>git checkout <branch></code>
4	<code>git rebase master</code>
Fix any merge conflicts	
	<code>git add <files></code>
	<code>git rebase --continue</code>
5	<code>git push origin <branch> (--force, if there were merge conflicts)</code>

Update Git for a Project

1	<code>cd</code> to the folder that contains <code>bDocs-Master</code>
2	<code>git pull</code> (if already on correct branch) most likely will need to do <code>git stash</code> to store changes to <code>config/app.r</code>
3	<code>git pull</code>
4	<code>git stash pop</code>

