Basic Git commands

Here is a list of some basic Git commands to get you going with Git.

For more detail, check out the **Atlassian Git Tutorials** for a visual introduction to Git commands and workflows, including examples.

Git task	Notes	Git commands
Tell Git who you are	Configure the author name and email address to be used with your commits. Note that Git strips some characters (for example trailing periods) from user.name.	git configglobal user.name "Sam Smith" git configglobal user.email sam@example.com
Create a new local repository		git init
Check out a repository	Create a working copy of a local repository:	git clone /path/to/repository
	For a remote server, use:	git clone username@host:/path/to/repository
Add files	Add one or more files to staging (index):	<pre>git add <filename> git add *</filename></pre>
Commit	Commit changes to head (but not yet to the remote repository):	git commit -m "Commit message"
	Commit any files you've added with git add, and also commit any files you've changed since then:	git commit -a
Push	Send changes to the master branch of your remote repository:	git push origin master
Status	List the files you've changed and those you still need to add or commit:	git status
Connect	If you haven't connected your local	git remote add origin <server></server>

repository Dist all currently configured remote repository:	4/14/2020	Basic Git	commands - Atlassian Documentation
List all currently configured remote repositories: Branches Create a new branch and switch to it: Switch from one branch to another: List all the branches in your repo, and also tell you what branch you're currently in: Delete the feature branch: Push the branch to your remote repository, so others can use it: Push all branches to your remote repository: Delete a branch on your remote repository: Delete a branch on your remote repository: Update from the remote server to your working directory: To merge a different branch into your active branch: View all the merge conflicts: View the conflicts against the base file: Preview changes, before merging: After you have manually resolved any conflicts, you mark the changed file: Tags You can use tagging to mark a significant changeset, such as a release: View all can use tagging to mark a significant changeset, such as a release:	remote		
it: Switch from one branch to another: List all the branches in your repo, and also tell you what branch you're currently in: Delete the feature branch: Push the branch to your remote repository, so others can use it: Push all branches to your remote repository: Delete a branch on your remote repository: Delete a branch on your remote repository: Update from the remote repository: To merge a different branch into your active branch: View all the merge conflicts: View the conflicts against the base file: Preview changes, before merging: After you have manually resolved any conflicts, you mark the changed file: Tags You can use tagging to mark a significant changeset, such as a release: git tag 1.0.0 <commitid></commitid>			git remote -v
List all the branches in your repo, and also tell you what branch you're currently in: Delete the feature branch: Push the branch to your remote repository, so others can use it: Push all branches to your remote repository: Delete a branch on your remote repository: Delete a branch on your remote repository: Fetch and merge changes on the remote server to your working directory: To merge a different branch into your active branch: View all the merge conflicts: View the conflicts against the base file: Preview changes, before merging: After you have manually resolved any conflicts, you mark the changed file: Tags You can use tagging to mark a significant changeset, such as a release: git branch git branch -d git push origin git push origin : prigin remote origin git push origin : git push origin : git push origin : git push origin : prigin remote origin : git push origin git push origin git push origin : git push origin : prigin remote origin : git push origin span changes git push origin git push origin span changes git push origin span changes span chang	Branches		git checkout -b branchname>
and also tell you what branch you're currently in: Delete the feature branch: Push the branch to your remote repository, so others can use it: Push all branches to your remote repository: Delete a branch on your remote repository: Update from the remote server to your working directory: To merge a different branch into your active branch: View all the merge conflicts: View the conflicts against the base file: Preview changes, before merging: After you have manually resolved any conflicts, you mark the changed file: Tags You can use tagging to mark a significant changeset, such as a release: But branch -d git push origin git push origin : prigin : spit push origin : git push origin : spit all origin remove all origin remove spit push origin : git push origin : git push origin : git push origin : spit push origin : spit push origin : git push origin : spit		Switch from one branch to another:	git checkout branchname>
Push the branch to your remote repository, so others can use it: Push all branches to your remote repository: Delete a branch on your remote repository: Update from the remote server to your working directory: To merge a different branch into your active branch: View all the merge conflicts: View the conflicts against the base file: Preview changes, before merging: After you have manually resolved any conflicts, you mark the changed file: Tags You can use tagging to mark a significant changeset, such as a release: git push origin git push origin : git push origin git push origin span call origin span call origin git push origin span call origin git push origin span call origin git push origin span call origin git push origin span call origin git diff span call origin git diff span call origin git diff span call origin git diff span call origin span call origin git diff span call origin span call origin s		and also tell you what branch you're	git branch
repository, so others can use it: Push all branches to your remote repository: Delete a branch on your remote repository: Delete a branch on your remote repository: Git push origin : git push origin : git pull Fetch and merge changes on the remote server to your working directory: To merge a different branch into your active branch: View all the merge conflicts: View the conflicts against the base file: Preview changes, before merging: After you have manually resolved any conflicts, you mark the changed file: Tags You can use tagging to mark a significant changeset, such as a release: git tag 1.0.0 <commitid></commitid>		Delete the feature branch:	git branch -d <branchname></branchname>
repository: Delete a branch on your remote repository: Update from the remote repository Fetch and merge changes on the remote server to your working directory: To merge a different branch into your active branch: View all the merge conflicts: View the conflicts against the base file: Preview changes, before merging: After you have manually resolved any conflicts, you mark the changed file: Tags You can use tagging to mark a significant changeset, such as a release: git pull git pull git diff git diff git diff sourcebranch> git diff <sourcebranch> <targetbranch> git add <filename> git add <filename> git tag 1.0.0 <commitid></commitid></filename></filename></targetbranch></sourcebranch>			git push origin branchname>
repository: Update from the remote server to your working directory: To merge a different branch into your active branch: View all the merge conflicts: View the conflicts against the base file: Preview changes, before merging: After you have manually resolved any conflicts, you mark the changed file: Tags You can use tagging to mark a significant changeset, such as a release: git pull git pull git merge git diff git diffbase <filename> git diff <sourcebranch> <targetbranch> git add <filename> git add <filename> git tag 1.0.0 <commitid></commitid></filename></filename></targetbranch></sourcebranch></filename>			git pushall origin
from the remote remote repository remote server to your working directory: git merge git diff View all the merge conflicts: View the conflicts against the base file: Preview changes, before merging: git diff git diffbase <filename> After you have manually resolved any conflicts, you mark the changed file: git add <filename> Tags You can use tagging to mark a significant changeset, such as a release: git tag 1.0.0 <commitid></commitid></filename></filename>			git push origin : branchname>
To merge a different branch into your active branch: View all the merge conflicts: View the conflicts against the base file: Preview changes, before merging: After you have manually resolved any conflicts, you mark the changed file: You can use tagging to mark a significant changeset, such as a release: git diff git diff git diffbase <filename> git add <filename> git add <filename> git tag 1.0.0 <commitid></commitid></filename></filename></filename>	from the remote	remote server to your working	git pull
View the conflicts against the base file: Preview changes, before merging: After you have manually resolved any conflicts, you mark the changed file: Tags You can use tagging to mark a significant changeset, such as a release: git diffbase <filename> git diffbase <filename> git diffbase <filename> git add <filename> git add <filename></filename></filename></filename></filename></filename>		_	git merge branchname>
file: Preview changes, before merging: After you have manually resolved any conflicts, you mark the changed file: Tags You can use tagging to mark a significant changeset, such as a release: git diffbase <filename> git diffbase <filename> git add <filename> git add <filename> git add <filename> git add <filename></filename></filename></filename></filename></filename></filename>		_	git diff
After you have manually resolved any conflicts, you mark the changed file: You can use tagging to mark a significant changeset, such as a release: You changeset, before merging: git add <filename> git add <filename></filename></filename>		_	git diffbase <filename></filename>
any conflicts, you mark the changed file: Tags You can use tagging to mark a significant changeset, such as a release: git tag 1.0.0 <commitid></commitid>		Preview changes, before merging:	git diff <sourcebranch> <targetbranch></targetbranch></sourcebranch>
significant changeset, such as a release:		any conflicts, you mark the changed	git add <filename></filename>
Committed is the leading characters git log	Tags	significant changeset, such as a	git tag 1.0.0 <commitid></commitid>
		CommitId is the leading characters	git log

4/14/2020	basic die communius - Adassian bocumentation		
	of the changeset ID, up to 10, but must be unique. Get the ID using:		
	Push all tags to remote repository:	git pushtags origin	
Undo local changes	If you mess up, you can replace the changes in your working tree with the last content in head: Changes already added to the index, as well as new files, will be kept.	git checkout <filename></filename>	
	Instead, to drop all your local changes and commits, fetch the latest history from the server and point your local master branch at it, do this:	git fetch origin git resethard origin/master	
Search	Search the working directory for foo():	git grep "foo()"	

Last modified on Sep 21, 2017