

Git task	Notes
<a href="#">Tell Git who you are</a>	<p>Configure the author name and email address to be used with your commits.</p> <p><u>Note that Git strips some characters (for example trailing periods) from user.name.</u></p>
<a href="#">Create a new local repository</a>	
<a href="#">Check out a repository</a>	<p>Create a working copy of a local repository:</p> <p>For a remote server, use:</p>
<a href="#">Add files</a>	Add one or more files to staging (index):
<a href="#">Commit</a>	<p>Commit changes to head (but not yet to the remote repository):</p> <p>Commit any files you've added with <code>git add</code>, and also commit any files you've changed since then:</p>
<a href="#">Push</a>	Send changes to the master branch of your remote repository:
<a href="#">Status</a>	List the files you've changed and those you still need to add or commit:
<a href="#">Connect to a remote repository</a>	<p>If you haven't connected your local repository to a remote server, add the server to be able to push to it:</p> <p>List all currently configured remote repositories:</p>
<a href="#">Branches</a>	<p>Create a new branch and switch to it:</p> <p>Switch from one branch to another:</p> <p>List all the branches in your repo, and also tell you what branch you're currently in:</p> <p>Delete the feature branch:</p> <p>Push the branch to your remote repository, so others can use it:</p> <p>Push all branches to your remote repository:</p> <p>Delete a branch on your remote repository:</p>
<a href="#">Update from the remote repository</a>	<p>Fetch and merge changes on the remote server to your working directory:</p> <p>To merge a different branch into your active branch:</p> <p>View all the merge conflicts:</p> <p>View the conflicts against the base file:</p> <p>Preview changes, before merging:</p> <p>After you have manually resolved any conflicts, you mark the changed file:</p>

<b>Tags</b>	<p>You can use tagging to mark a significant changeset, such as a release:</p> <p>CommitId is the leading characters of the changeset ID, up to 10, but must be unique. Get the ID using:</p> <p>Push all tags to remote repository:</p>
<a href="#">Undo local changes</a>	<p>If you mess up, you can replace the changes in your working tree with the last content in head:</p> <p>Changes already added to the index, as well as new files, will be kept.</p> <p>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:</p>
<b>Search</b>	Search the working directory for <code>£○○ ( )</code> :

#### Git commands

```
git config --global user.name "Sam Smith"
```

```
git config --global user.email  
sam@example.com
```

```
git init
```

```
git clone /path/to/repository
```

```
git clone username@host:/path/to/repository
```

```
git add <filename>
```

```
git add *
```

```
git commit -m "Commit message"
```

```
git commit -a
```

```
git push origin master
```

```
git status
```

```
git remote add origin <server>
```

```
git remote -v
```

```
git checkout -b <branchname>
```

```
git checkout <branchname>
```

```
git branch
```

```
git branch -d <branchname>
```

```
git push origin <branchname>
```

```
git push --all origin
```

```
git push origin :<branchname>
```

```
git pull
```

```
git merge <branchname>
```

```
git diff
```

```
git diff --base <filename>
```

```
git diff <sourcebranch> <targetbranch>
```

```
git add <filename>
```

git tag 1.0.0 <commitID>
git log
git push --tags origin
git checkout -- <filename>
git fetch origin
git reset --hard origin/master
git grep "foo() "