

05. Tagging

Tags are refs that point to specific points in Git history. Tagging is generally used to capture a point in history that is used for a marked version release (i.e. v1.0.1). A tag is like a branch that doesn't change. Unlike branches, tags, after being created, have no further history of commits.

For a bit more in-depth discussion see <https://www.atlassian.com/git/tutorials/inspecting-a-repository/git-tag>.

- **Listing your tags —**

```
# to list all of your tags
$ git tag
```

```
# to search for a specific string in the tag list
$ git tag -l <search string>
```

- **Creating tags —**

```
# Annotated Tag: annotated tags store extra meta data
# such as: the tagger name, email, date, tagging message etc.
# they can also be signed and verified
$ git tag -a <tag> -m <message>
```

```
# Lightweight Tag
$ git tag <tag>
```

```
# Tagging at a previous commit
$ git tag -a <tag> <commit checksum (or part of it)>
```

- **Checking out tags —**

```
$ git checkout <tag name>
```

Note: This puts the repository in a detached `HEAD` state. This means any changes made will not update the tag. They will create a new detached commit. This new detached commit will not be part of any branch and will only be reachable directly by the commits' SHA hash. Therefore it is a best practice to create a new branch anytime you're making changes in a detached `HEAD` state.

- **Sharing/transferring tags to the remote —**

By default, the `git push` command doesn't transfer tags to remote servers. You will have to explicitly push tags to a shared server after you have created them. This process is just like sharing remote branches —

```
$ git push <remote name> <tag name>

# If you have a lot of tags that you want to push up at once use
$ git push <remote name> --tags.
```

- **Deleting tags —**

```
# From your local machine
$ git tag -d <tag name>
```

```
# From the remote repository
$ git push <remote name> --delete <tag name>
```