git tag will give a label to the commit

```
fms@git_practice:~/dev_flowers/sample-project$ git tag -a "creating tag after creating candy.sh"
fatal: 'creating tag after creating candy.sh' is not a valid tag name.
fms@git_practice:~/dev_flowers/sample-project$ git tag -a v1.0 -m "creating tag after creating candy.sh"
fms@git_practice:~/dev_flowers/sample-project$
```

To list the tags, use git tag

```
fms@git_practice:~/dev_flowers/sample-project$ git tag
v1.0
fms@git_practice:~/dev_flowers/sample-project$

fms@git_practice:~/dev_flowers/sample-project$ git show v1.0
tag v1.0
Tagger: Dev-Flowers <dev_flowers@FLOWERS.com>
Date:   Tue Nov 6 16:10:48 2018 +0530

creating tag after creating candy.sh

commit b52fa475a63ec7138d999039370941483bbcaa66 (HEAD -> branch_one, tag: v1.0, origin/branch_one)
Author: Dev-Flowers <dev_flowers@FLOWERS.com>
Date:   Tue Nov 6 15:41:46 2018 +0530

    new file candy.sh

diff --git a/candy.sh b/candy.sh
new file mode 100644
index 0000000..02d6e6d
--- /dev/null
+++ b/candy.sh
@@ -0,0 +1 @@
+chocolates
fms@git_practice:~/dev_flowers/sample-project$
```

To delete a tag

```
fms@git_practice:~/dev_flowers/sample-project$ git tag -d v1.0
Deleted tag 'v1.0' (was 1efc66e)
fms@git_practice:~/dev_flowers/sample-project$
fms@git_practice:~/dev_flowers/sample-project$ git tag

fms@git_practice:~/dev_flowers/sample-project$ git show v1.0
fatal: ambiguous argument 'v1.0': unknown revision or path not in the
working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'
fms@git_practice:~/dev_flowers/sample-project$
```

The tag we created got deleted.

```
fms@git_practice:~/dev_flowers/sample-project$ ls -al
total 32
drwxr-xr-x 3 fms fms 4096 Nov  6 15:44 .
drwxr-xr-x 3 fms fms 4096 Nov  6 13:21 ..
-rw-r--r-- 1 fms fms  414 Nov  6 15:43 0001-new-file-candy.sh.patch
-rw-r--r-- 1 fms fms   19 Nov  6 15:37 buds.cpp
-rw-r--r-- 1 fms fms   11 Nov  6 15:44 candy.sh
drwxr-xr-x 8 fms fms 4096 Nov  6 16:12 .git
-rw-r--r-- 1 fms fms    7 Nov  6 15:37 .gitignore
-rw-r--r-- 1 fms fms   12 Nov  6 15:37 sweets.py
```

To remove untracked files from the working tree use git clean

```
fms@git_practice:~/dev_flowers/sample-project$ git clean
fatal: clean.requireForce defaults to true and neither -i, -n, nor -f given;
refusing to clean
fms@git_practice:~/dev_flowers/sample-project$ git clean -i
Would remove the following item:
  0001-new-file-candy.sh.patch
*** Commands ***
    1: clean                2: filter by pattern    3: select by numbers    4: ask
each            5: quit                 6: help
What now> 1
Removing 0001-new-file-candy.sh.patch
fms@git_practice:~/dev_flowers/sample-project$

fms@git_practice:~/dev_flowers/sample-project$ ls -al
total 28
drwxr-xr-x 3 fms fms 4096 Nov  6 16:14 .
drwxr-xr-x 3 fms fms 4096 Nov  6 13:21 ..
```

```
-rw-r--r-- 1 fms fms   19 Nov  6 15:37 buds.cpp
-rw-r--r-- 1 fms fms   11 Nov  6 15:44 candy.sh
drwxr-xr-x 8 fms fms 4096 Nov  6 16:12 .git
-rw-r--r-- 1 fms fms    7 Nov  6 15:37 .gitignore
-rw-r--r-- 1 fms fms   12 Nov  6 15:37 sweets.py
fms@git_practice:~/dev_flowers/sample-project$
```

To see what has been changed after a certain time, use git whatchanged.

```
fms@git_practice:~/dev_flowers/sample-project$ git whatchanged
--since="2 hour ago" -- candy.sh
commit b52fa475a63ec7138d999039370941483bbcaa66 (HEAD ->
branch_one, origin/branch_one)
Author: Dev-Flowers <dev_flowers@FLOWERS.com>
Date:   Tue Nov 6 15:41:46 2018 +0530

    new file candy.sh

:000000 100644 0000000 02d6e6d A        candy.sh
fms@git_practice:~/dev_flowers/sample-project$

fms@git_practice:~/dev_flowers/sample-project$ git whatchanged
--since="4 hour ago" -- buds.cpp
commit 803fdf4b2bd9eb7b2f5e870e4b7a456268b29492
Author: Dev-Colors <dev_colors@COLORS.com>
Date:   Tue Nov 6 15:16:17 2018 +0530

    renamed flowers.c to buds.c

:000000 100644 0000000 3360a33 A        buds.cpp
fms@git_practice:~/dev_flowers/sample-project$

fms@git_practice:~/dev_flowers/sample-project$ git whatchanged
--since="2 hour ago" -- sweets.py
fms@git_practice:~/dev_flowers/sample-project$
```

We will understand git stash command.

```
fms@git_practice:~/dev_flowers/sample-project$ ls
buds.cpp  candy.sh  sweets.py
fms@git_practice:~/dev_flowers/sample-project$ vi emptyfile.txt
fms@git_practice:~/dev_flowers/sample-project$ cat  emptyfile.txt
empty file
```

```
fms@git_practice:~/dev_flowers/sample-project$

fms@git_practice:~/dev_flowers/sample-project$ git status
On branch branch_one
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    emptyfile.txt

nothing added to commit but untracked files present (use "git add" to track)
fms@git_practice:~/dev_flowers/sample-project$

fms@git_practice:~/dev_flowers/sample-project$ git add emptyfile.txt
fms@git_practice:~/dev_flowers/sample-project$ git status
On branch branch_one
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   emptyfile.txt

fms@git_practice:~/dev_flowers/sample-project$
```

We have this new file in staging area. If we do not want to commit this change now, we can record the changes in this file using git stash. This will save the changes made and reset the workspace to point to HEAD.

```
fms@git_practice:~/dev_flowers/sample-project$ git stash
Saved working directory and index state WIP on branch_one: b52fa47 new
file candy.sh
fms@git_practice:~/dev_flowers/sample-project$

fms@git_practice:~/dev_flowers/sample-project$ git status
On branch branch_one
nothing to commit, working tree clean
fms@git_practice:~/dev_flowers/sample-project$

fms@git_practice:~/dev_flowers/sample-project$ git stash list
stash@{0}: WIP on branch_one: b52fa47 new file candy.sh
fms@git_practice:~/dev_flowers/sample-project$

fms@git_practice:~/dev_flowers/sample-project$ git stash show
 emptyfile.txt | 1 +
 1 file changed, 1 insertion(+)
fms@git_practice:~/dev_flowers/sample-project$
```

```
fms@git_practice:~/dev_flowers/sample-project$ git stash drop
Dropped refs/stash@{0} (2acd91b3736e9a91d23be8bab30f2b974854c0cd)
fms@git_practice:~/dev_flowers/sample-project$
fms@git_practice:~/dev_flowers/sample-project$ git stash list
fms@git_practice:~/dev_flowers/sample-project$
fms@git_practice:~/dev_flowers/sample-project$ git status
On branch branch_one
nothing to commit, working tree clean
fms@git_practice:~/dev_flowers/sample-project$
```

Git commands covered so far:

24. git tag
25. git clean
26. git whatchanged
27. git stash