

## Module 3: Git and Jenkins

---

Demo Document – 3

edureka!

**edureka!**

© Brain4ce Education Solutions Pvt. Ltd.

## Demo – 3: Stashing, Rebasing, Reverting and Resetting

- To Create a stash of your current working directory  
Syntax: git stash save 'message'
- To list all the saved stashes  
Syntax: git stash list
- Stash list uses a stack structure to save the list

```
edureka@kmaster:~/demo$ git stash list
stash@{0}: On feat1: Saving changes made to RepoFile
edureka@kmaster:~/demo$
```

- Saved stashes can be applied at any time on any branch  
Syntax: git stash apply <stash id>
- After applying a stash, it can still be accessed elsewhere because it remains in the stash

```
edureka@kmaster:~/demo$ git stash apply stash@{0}
On branch feat1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   RepoFile

no changes added to commit (use "git add" and/or "git commit -a")
edureka@kmaster:~/demo$
```

- Pop command can be used to apply the most recent stash and removing it from the stash stack  
Syntax: git stash pop

```
edureka@kmaster:~/demo$ git stash list
stash@{0}: On master: Updated new stash
stash@{1}: On feat1: Saving changes made to the RepoFile
edureka@kmaster:~/demo$ git stash pop
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   RepoFile

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (d6cbe48f8d07d6238654badd893399535e3bade3)
edureka@kmaster:~/demo$ git stash list
stash@{0}: On feat1: Saving changes made to the RepoFile
edureka@kmaster:~/demo$
```

- Stashes can be deleted from the stash list  
Syntax: git stash drop <stack id>

```
edureka@kmaster:~/demo$ git stash drop stash@{0}
Dropped stash@{0} (4efd84ff40abda6ee6d25b618f6a25f1781384cb)
edureka@kmaster:~/demo$
```

- The entire stack can also be deleted using one command  
Syntax: git stash clear

```
edureka@kmaster:~/demo$ git stash clear
edureka@kmaster:~/demo$
```

Rebasing, reverting, and resetting:

- To Rebase a branch  
Syntax: git rebase <rebase branch>

```
edureka@kmaster:~/demo$ git rebase master
First, rewinding head to replay your work on top of it...
Fast-forwarded feat1 to master.
edureka@kmaster:~/demo$
```

- Revert or **undo** the changes made in the previous commit

- New commit is created without the changes made in the other commit  
Syntax: `git revert <commit id>`
- Old commit still resides in the history

```
edureka@kmaster:~/demo$ git revert HEAD
[feat1 e8581e0] Revert "revert edit"
 1 file changed, 2 deletions(-)
edureka@kmaster:~/demo$
```

- Reset command can be used to undo changes at different levels
- Modifiers like `--hard`, `--soft` and `--mixed` can be used to decide the degree to which to reset

Syntax: `git reset <modifier> <commit id>`

```
edureka@kmaster:~/demo$ git reset v0.3
Unstaged changes after reset:
D      Demo.java
edureka@kmaster:~/demo$
```

edureka!