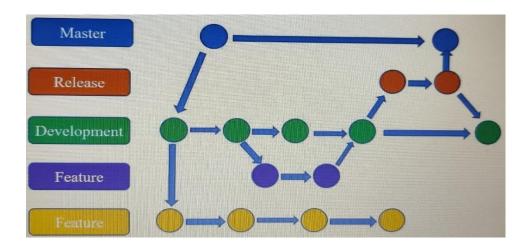


#### **GIT BRANCHING:**

- Branching is a feature available in most modern version control systems.
- In Git, a branch is a new/separate version of the main repository.
- branches are a part of your everyday development process.
- A Git project can have more than one branch. These branches are a pointer to a snapshot of your changes.
- A branch in Git is simply a lightweight movable pointer to one of these commits.
- The default branch name in Git is master.

#### > REAL WORLD BRANCHING SCENARIO:



**MASTER BRANCH:** Production ready copy

**RELEASE BRANCH:** This branch is created from the development branch to make it ready for the release and it is used for bug tracking and documentation purpose.

**DEVELOPMENT:** It is a developer branch where continuous work will be done.

**FEATURE:** Whenever the developers working on the new features, they use the feature branches and commit the code to that branch.

#### To list all branches:

\$git branch

#### To list remote branches:

\$git branch -r

### To list Local and remote Branches:

\$git branch -a

#### To create a new branch:

\$git branch development

\$git branch

### To create and switch a branch at a time:

\$git checkout -b branchname

### To switch a development branch:

\$git switch / checkout development

\$git branch

### Now create a new branch feature:

\$git branch feature

\$git switch feature

\$git branch

### **Create some files in feature branch:**

script1 script2 script3

\$git add.

\$git commit -m "New files are added"

### To rename a current branch:

\$git branch -m branch\_name

\$git branch -a

### **Creating a Branch from Another Branch:**

\$git checkout -b development release

### Now push the commits into GitHub:

\$git push origin feature

Now we can verify the feature branch and files in the GitHub Repository.

### Now switch to development branch:

\$git switch development

\$git branch

#### Here we can create some file.

dev1 dev2 dev3

\$git add.

\$git commit -m "New development files are added"

\$git status

### Now push the commits into GitHub:

\$git push origin development

Now we can verify the development branch and files in the GitHub Repo.

**GIT DIFF:** Show changes between commits, commit and working tree, etc

\$git diff

\$git diff development feature

### **MERGE BRANCHES:**

- In Git, merging is the process of combining changes from multiple branches into a single branch.
- Merging is Git's way of putting a forked history back together again.
- Before performing a merge there are a couple of preparation steps to take to ensure the merge goes smoothly.

# First confirm the receiving branch:

\$git checkout master

#### **Fetch the latest remote commits:**

Make sure the receiving branch and the merging branch are up-to-date with the latest remote changes.

\$git status

\$git fetch

### Now merging a branch:

\$git merge feature

**NOTE:** A fast-forward merge can occur when there is a linear path from the current branch tip to the target branch.

# Merge development to master:

\$git checkout master

\$git merge development

\$git status

\$1s

#### **Push the commits into GitHub:**

\$git push origin master

Delete a local branch (if not required)

\$git branch -d development

\$git branch

#### **Delete a remote branch:**

\$git push origin :refs/heads/feature-1 [even not available local repository]

# To delete a specified branch:

\$git branch -d branch\_name

\$git branch -D branch\_name