Video Lecture # 07
`git` Part-II

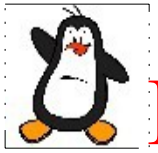
Course: SYSTEM PROGRAMMING

Instructor: Arif Butt

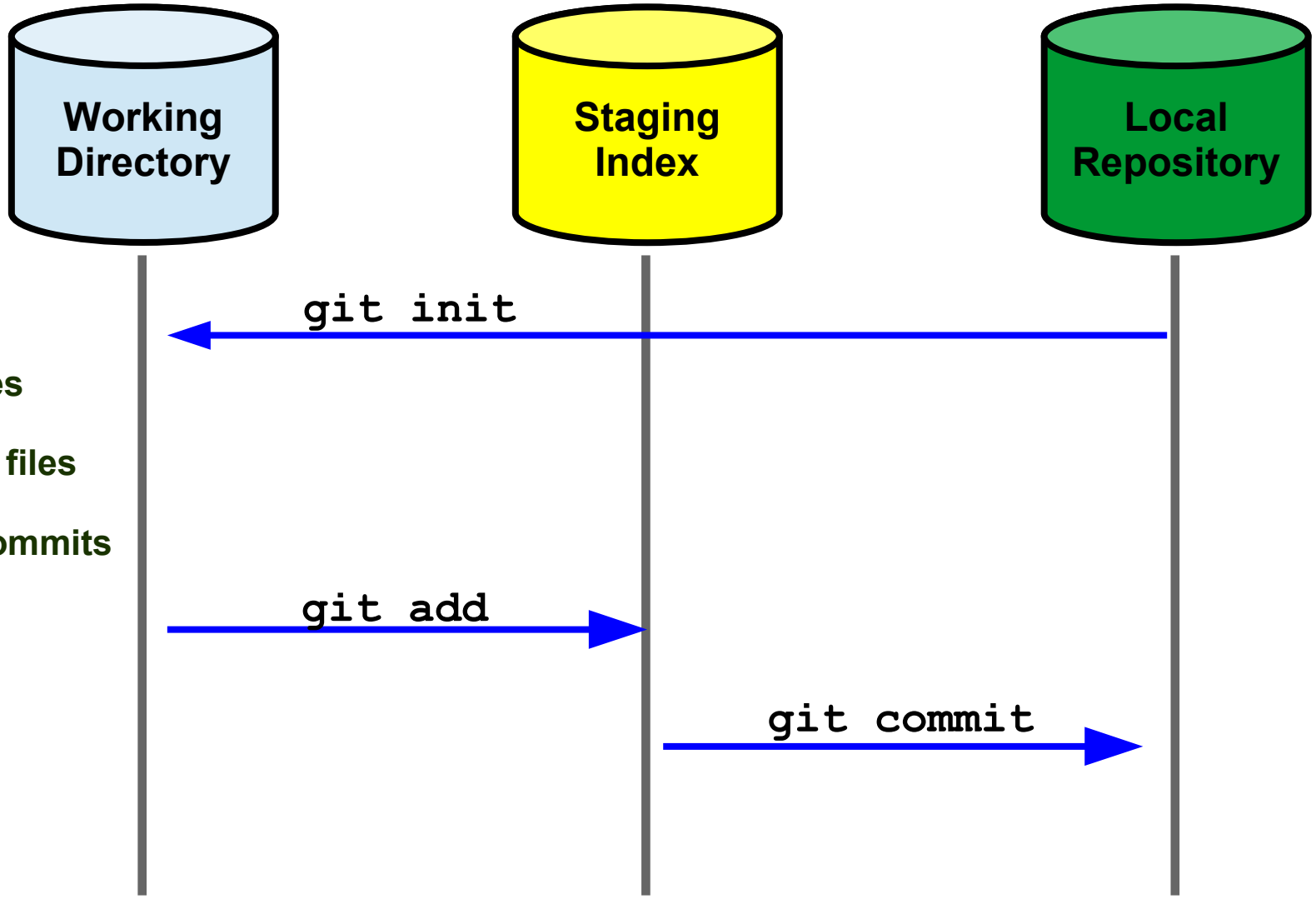
Punjab University College of Information Technology (PUCIT)
University of the Punjab

Agenda

- Review of previous session
- Working with branches in **`git`**
  - Creating a new branch
  - Switching branches and working on them
  - Comparing two branches
  - Renaming and deleting a branch
  - Merging branches and handling merge conflicts
- Working with Remote Repositories in **`git`**
  - What is remote repository
  - Git hosting services
  - Creating a remote repository on bitbucket.org
  - Uploading a local project repo on bitbucket
  - Downloading a remote project repo from bitbucket



Basic Workflow of `git` (Review)



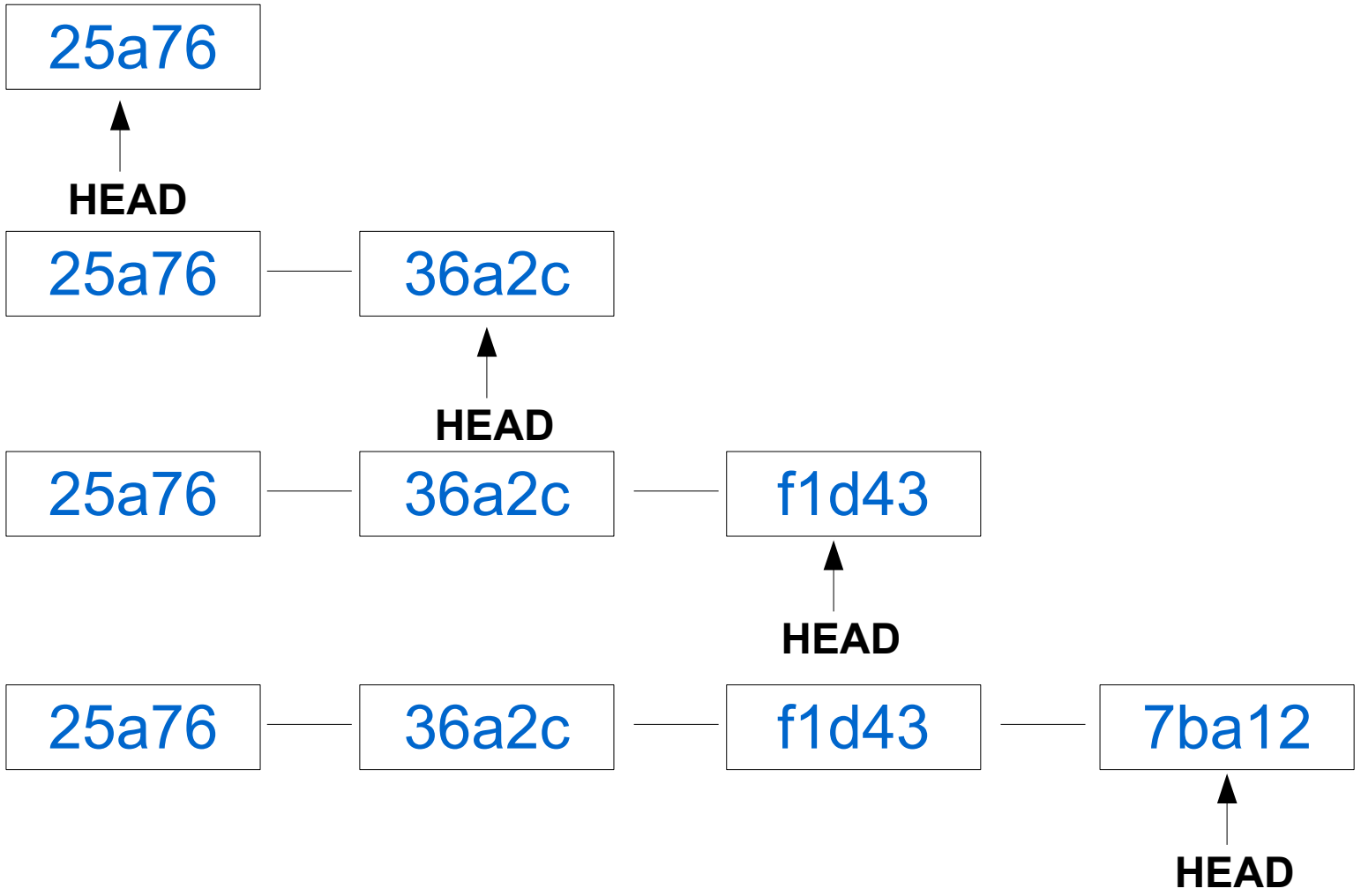
Tracked Files (Unmodified, Modified, Staged)
Untracked Files

git

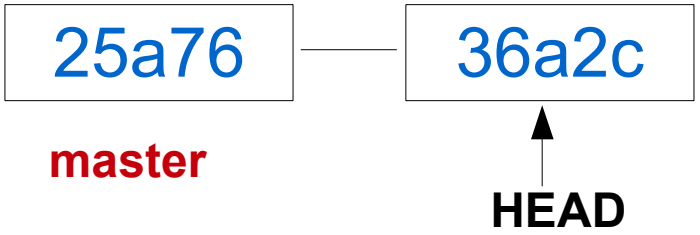
Branches

Overview of git Branches

A `git` branch represents an independent line of development. Every `git` repository has at least one branch called the master branch. An illustration of **master branch** is shown below:

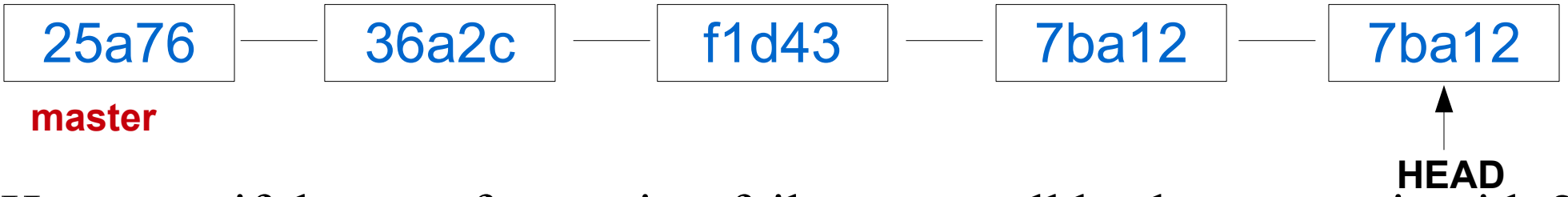


Overview of git Branches (cont...)



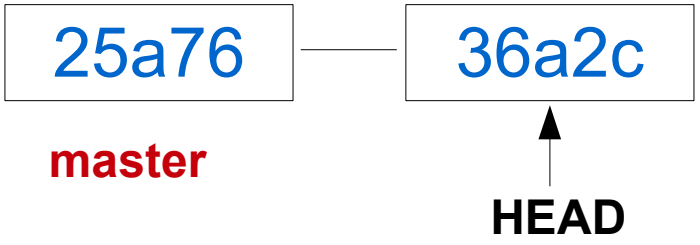
Suppose you are working on a project and have done some commits on the master branch which is the main line of your project development as shown above. You think of adding a new feature to your project but you are not sure whether it will work or not

- **OPTION 1:** You continue working on the same branch. If the new feature is a success, its GR8 and the development continues as shown below:

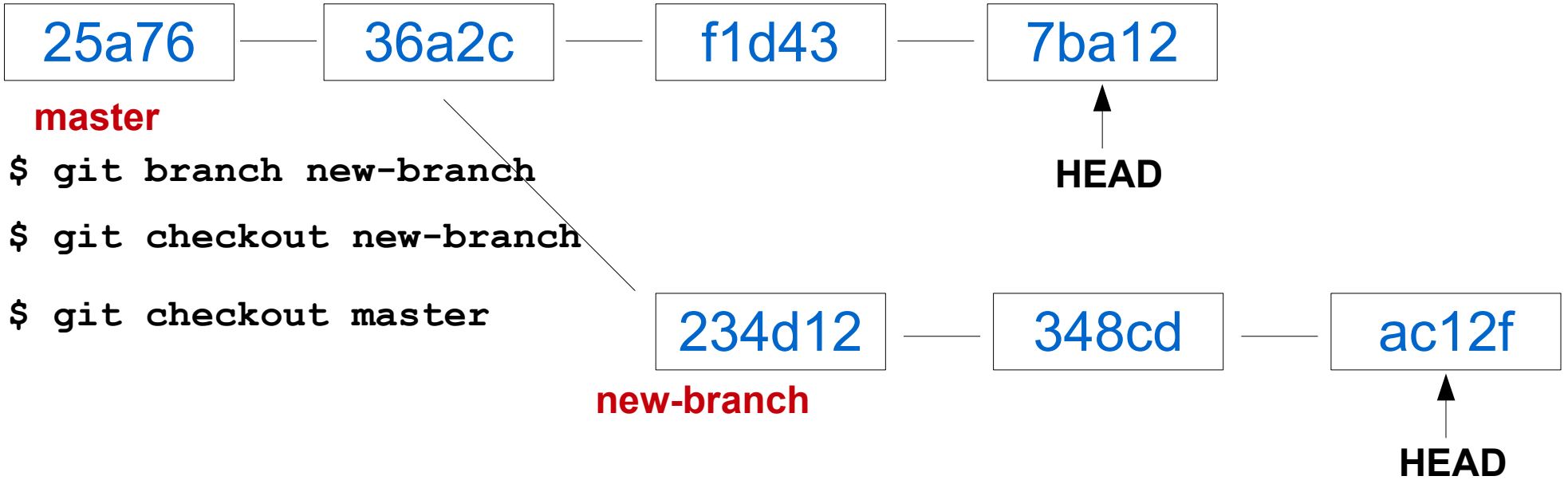


However, if the new feature is a failure you roll back to commit with SHA `36a2c` using a `git reset`, and your master branch again becomes similar to the one shown at the top

Overview of git Branches (cont...)



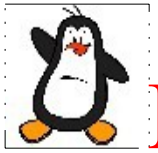
OPTION 2:



If the new-branch is a success, then you need to merge your new-branch with the master branch, otherwise, you can delete the new-branch and the master branch continue growing

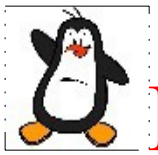
git

Merging the Branches

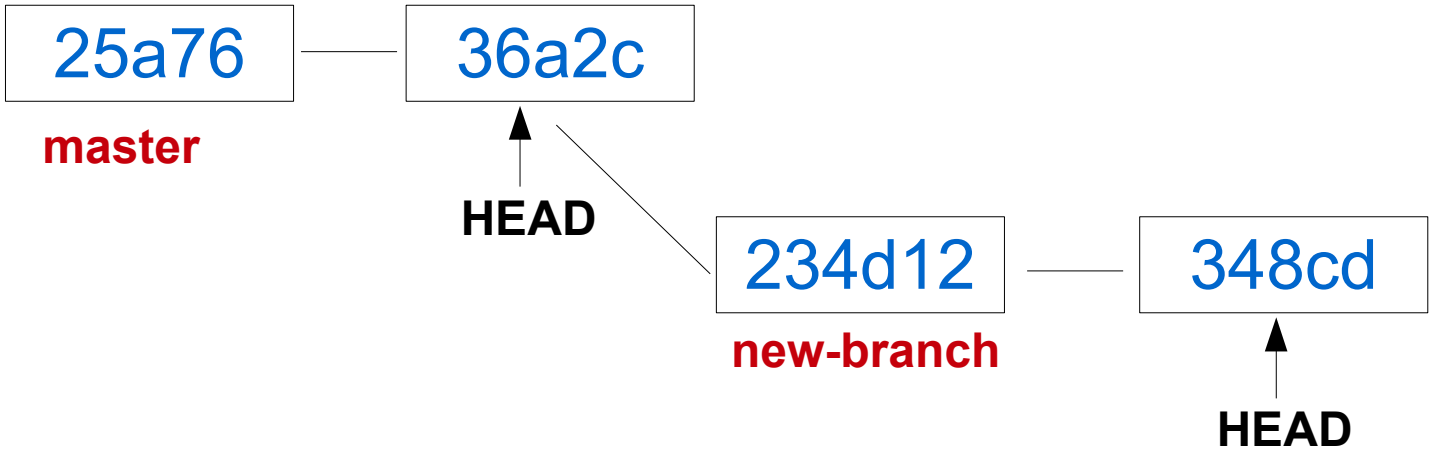


Merging Branches in git

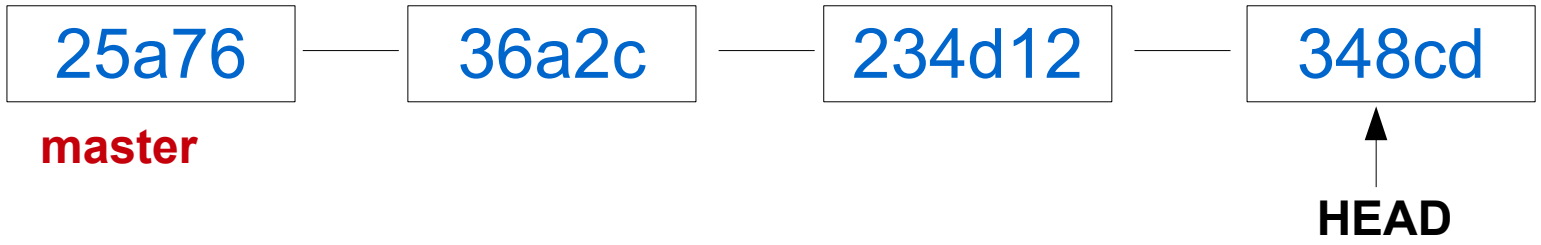
- Now we know how to create a new branch and how to perform development on that branch. After we are done developing and testing the new feature, it is time to bring those changes back to the master branch. For this we need to do a merge
- There can be two types of merges
  - Fast Forward Merge
  - Real Merge



Fast Forward Merge

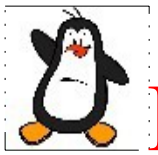


After fast forward merge:

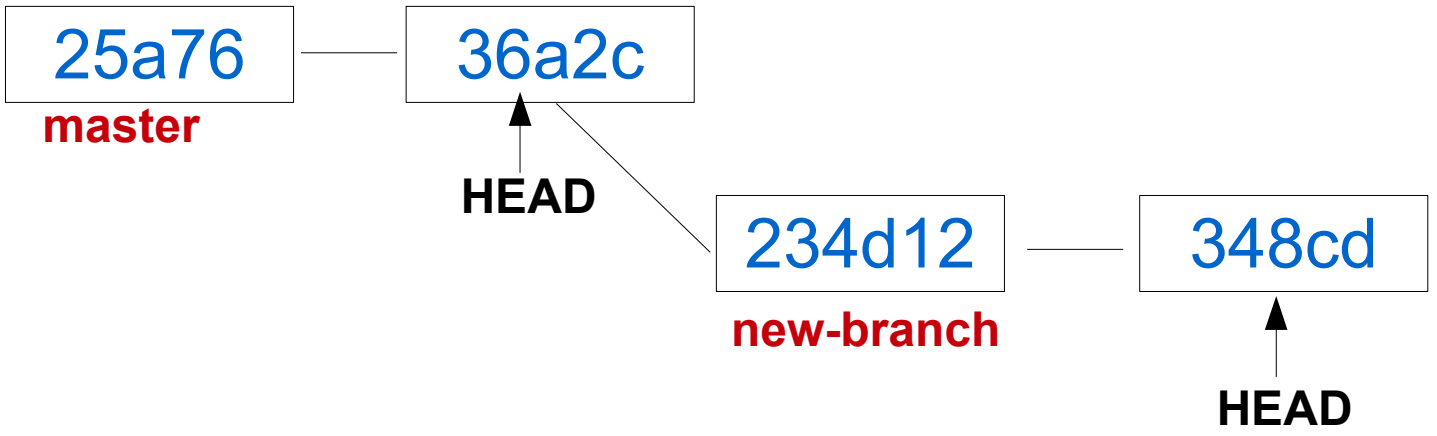


Before you give merge command, your current branch should be the receiving branch

```
$ git checkout master
$ git merge new-branch
```

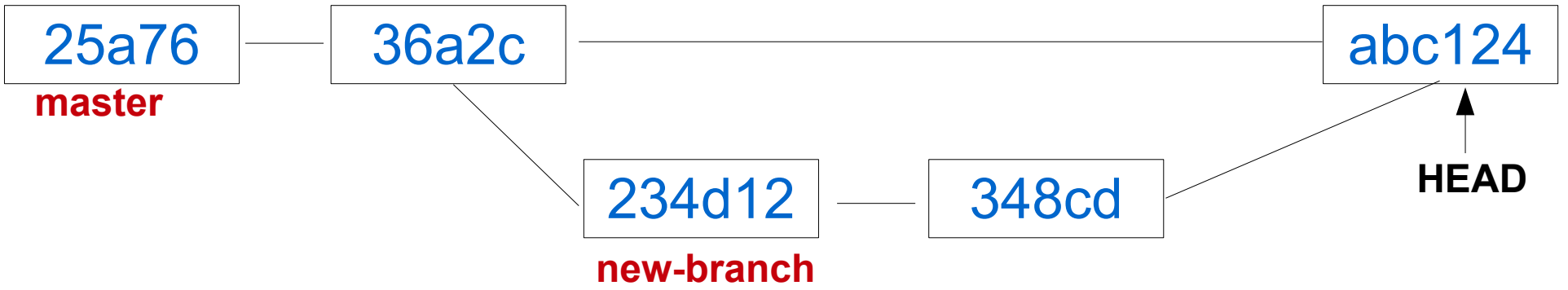


Real Merge

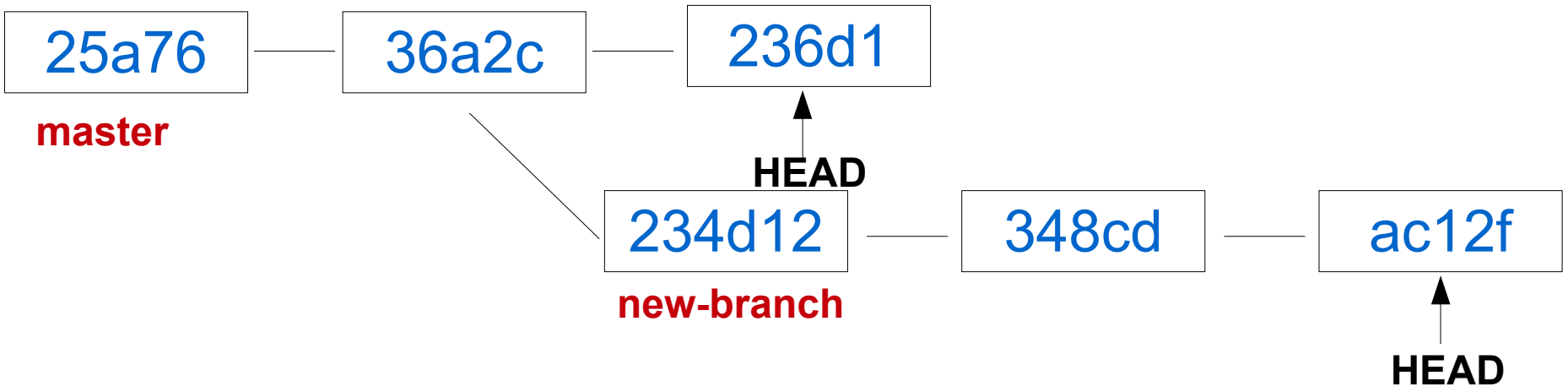


You can always force `git` not to do a fast forward merge, rather do an additional commit merge

```
$git checkout master
$git merge --no-ff newbranch
```

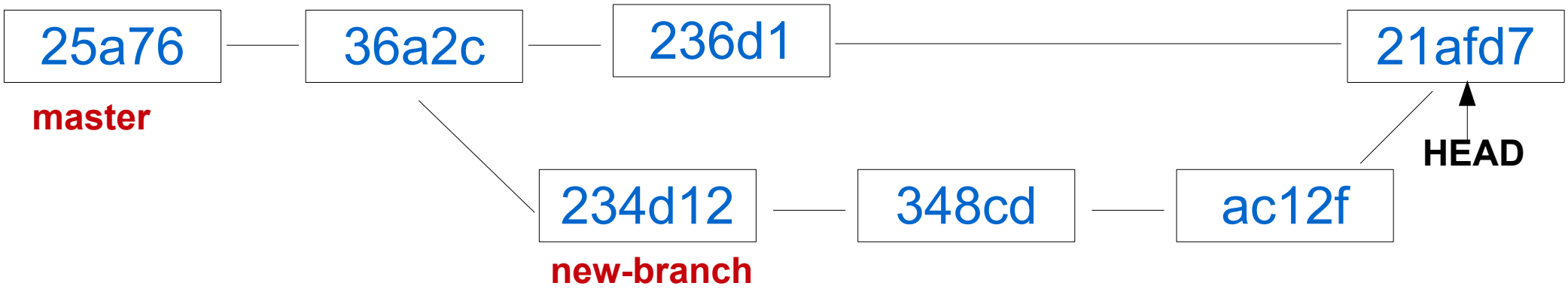


Real Merge



In this scenario a fast forward merge is not possible. So when you do a merge, git will perform a real merge

```
$git checkout master
$git merge newbranch
```

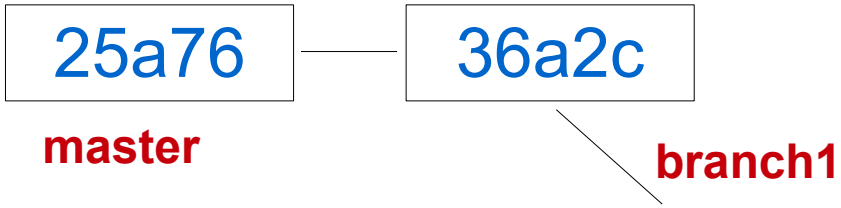


git

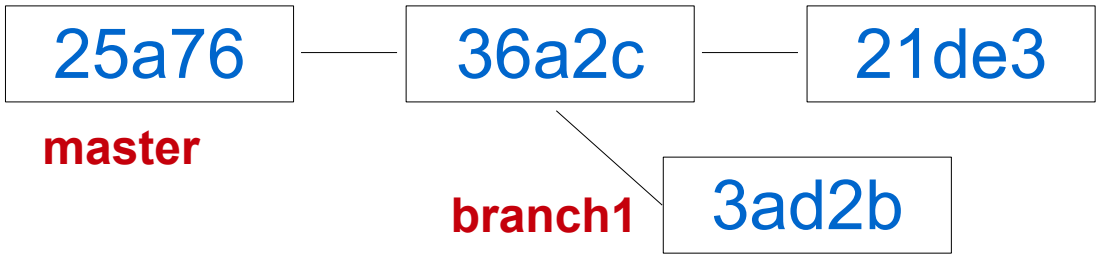
Handling Merge Conflicts

What is a Merge Conflict?

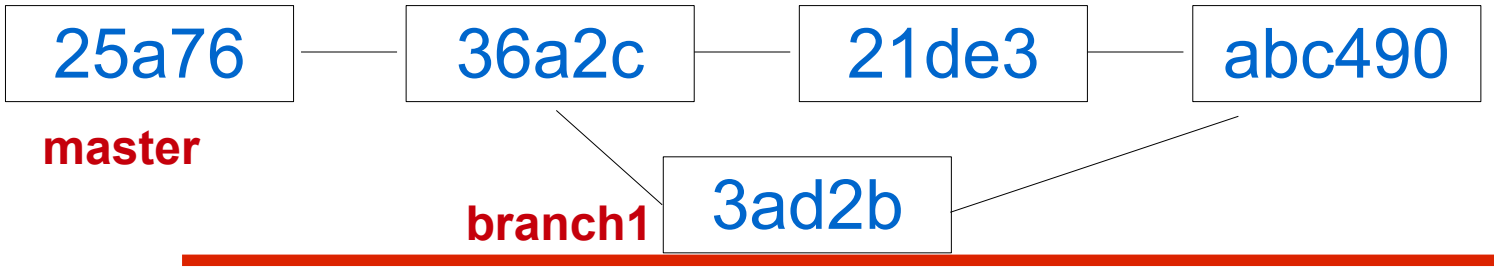
Suppose there are two branches **master** and **branch1**, as shown:

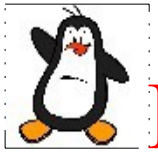


Both have a file suppose `file1.txt`, which is of course similar in both. A developer on **master** branch edit line#25 of `file1.txt` and do a commit. Another developer on **branch1** edit line#50 of `file1.txt` and do a commit



Now if you merge, it will be a success, because both have made changes to same file, but to different lines





Handling Merge Conflicts

However, if both the developers have made changes to same line or set of lines a conflict will occur, which `git` cannot handle and it will give a message that auto-merging failed. In case of a merge conflict we have three choices to resolve the conflict:

Abort merge:

```
$ git merge --abort
```

Resolve manually:

Open the file in some editor and perform the changes manually, add, commit, and finally perform merge

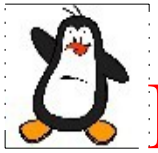
```
$ git merge <branchname>
```

Use merge tools:

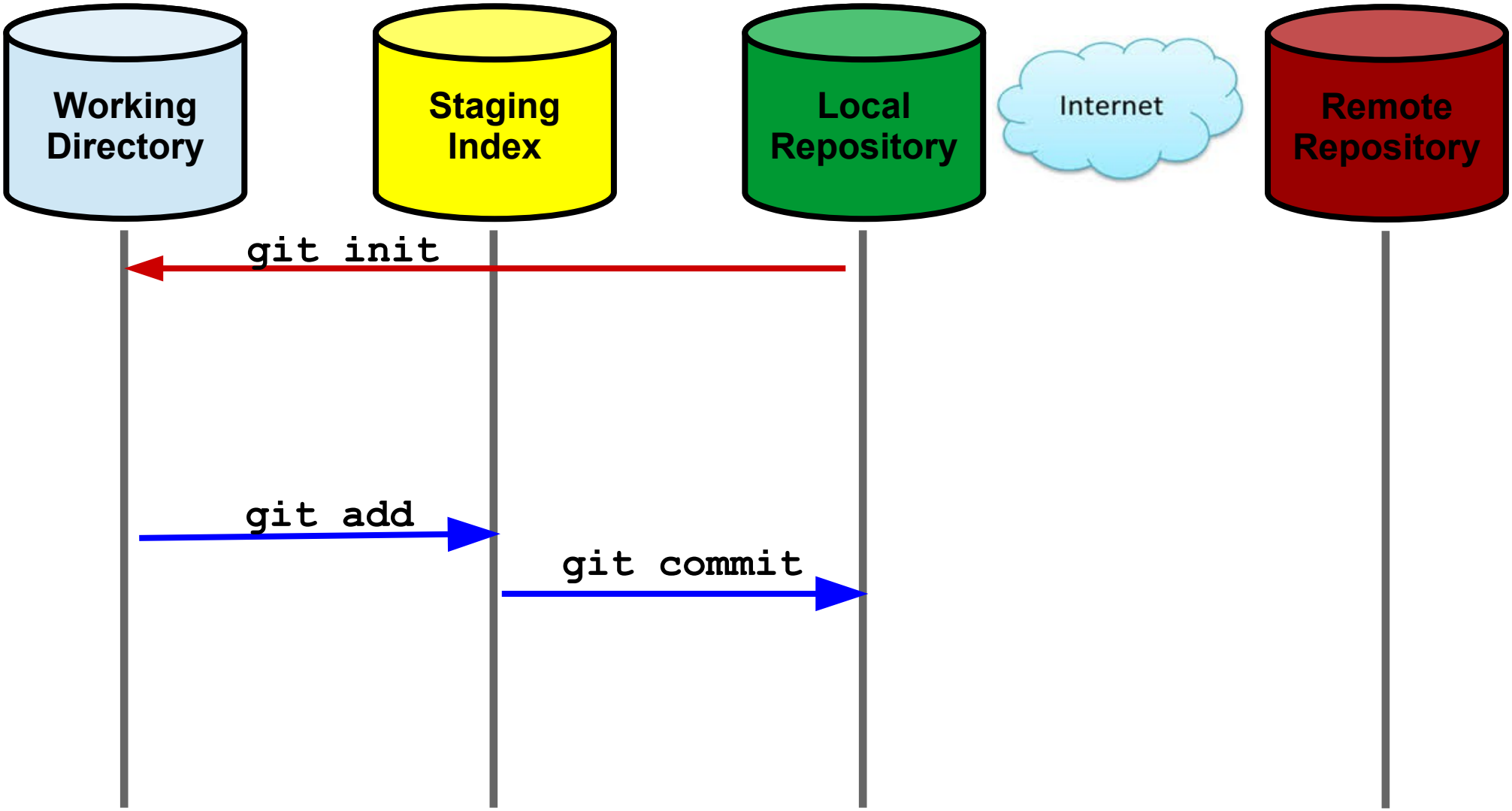
You can use different tools to automate this process like `araxis, diffuse, kdiff3, xxdiff, diffmerge`

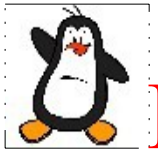
```
$ git mergetool  --tool=diffuse
```

Working of `git` Remote Repositories



Remote Repository

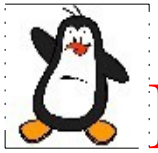




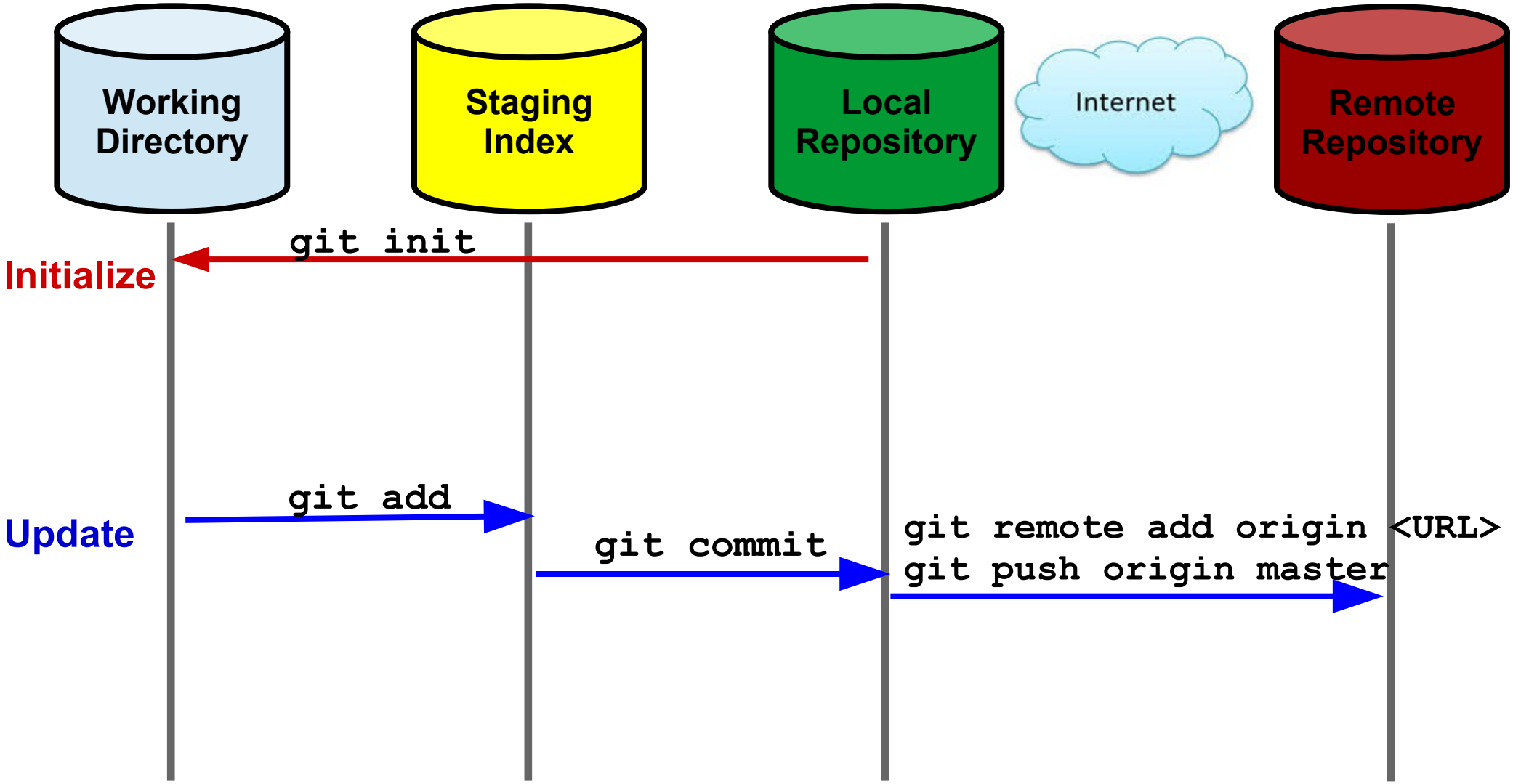
Hosting Services for Versioning Systems

Some of the most famous hosting services are:

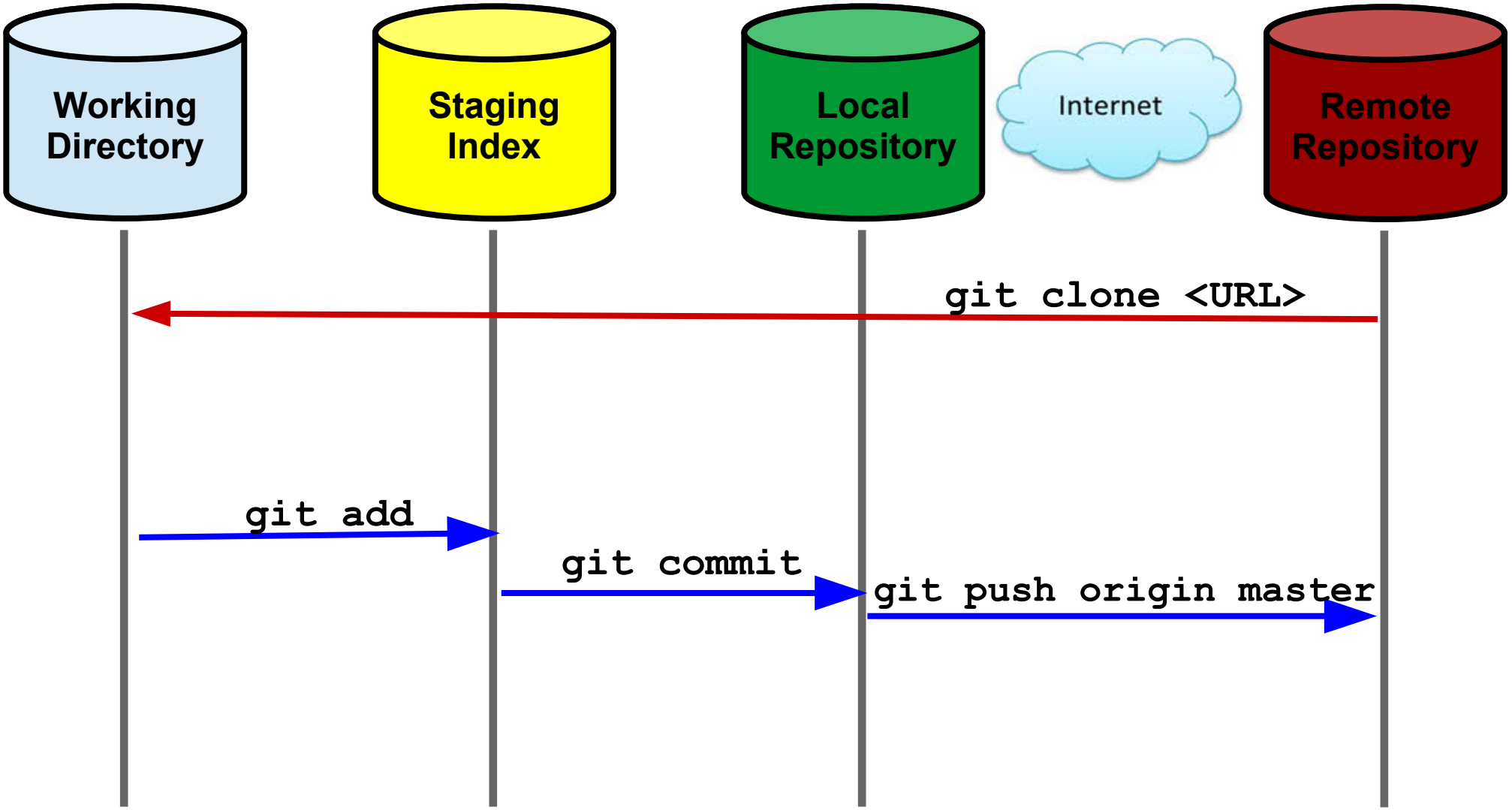
- https://bitbucket.org
- https://github.com
- https://gitlab.com



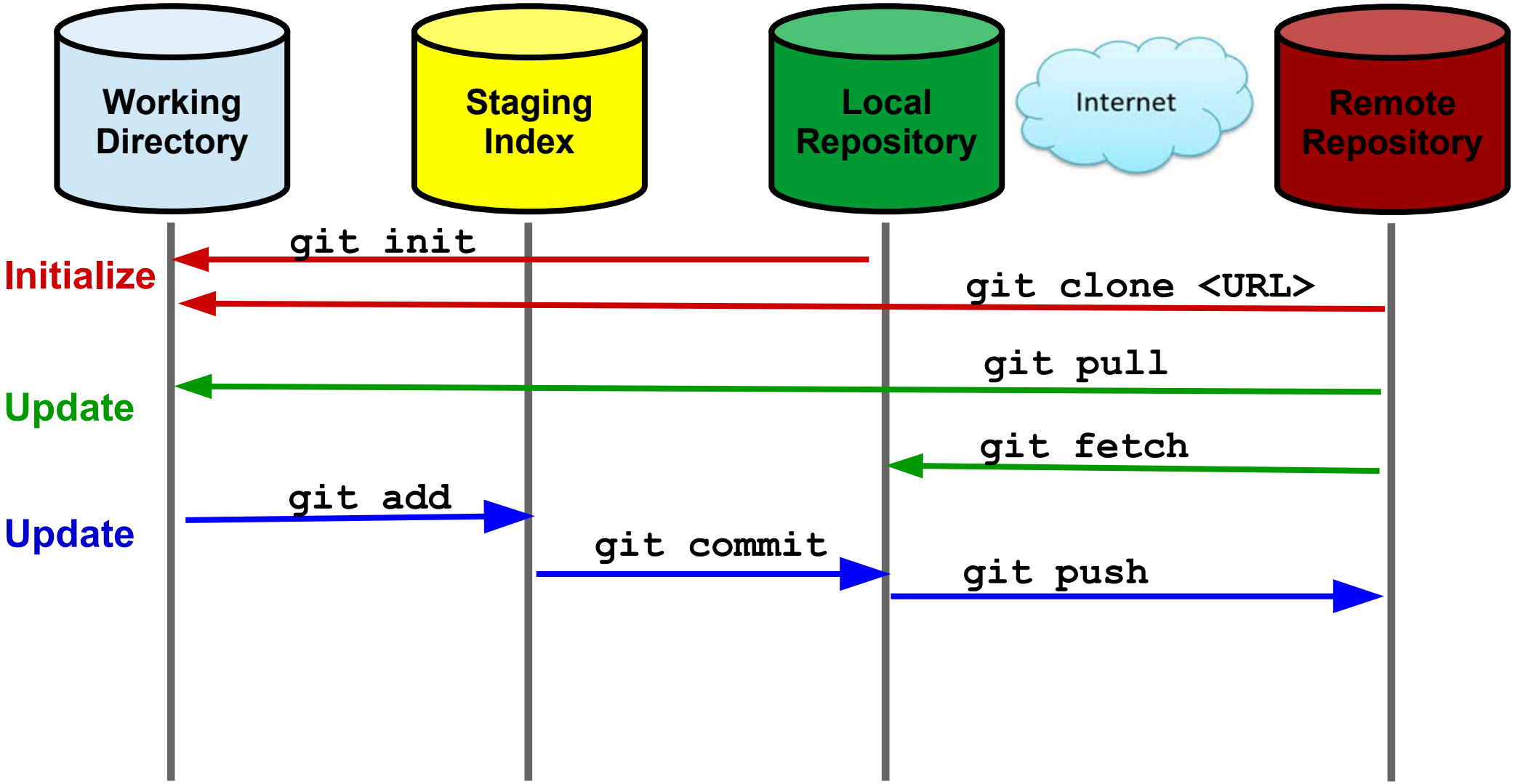
Pushing a Local Repo to Remote Repo



Cloning Remote Repo to Local



Workflow of `git` (Review)



Things To Do

If you have problems visit me in counseling hours. . . .