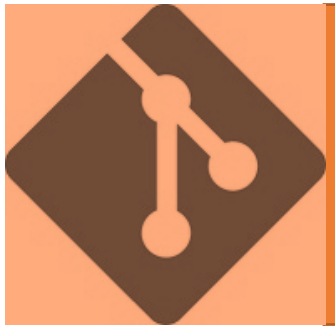
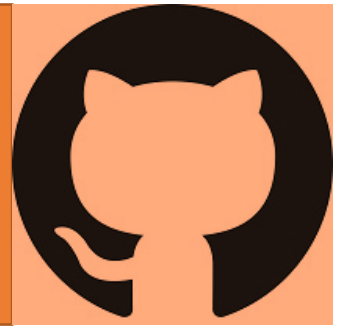


Git and Github



What is Version Management / Control?

Version control, also known as source control, is the practice of tracking and managing changes to software code.

Git and Github

Git

- Version Control System
- Manage Code History
- Track Changes

Github

- Largest Development Platform
- Cloud Hoisting and Collaboration Provider
- Git Repository Hoisting

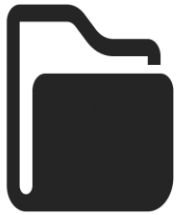
How does Git Works?

Working Directory

Web Shop App


index.html


styles.css


images


Commit 1

“snapshot 1”


Commit 2

“snapshot 2”


Commit 3

“snapshot 3”

Master Branch

Web Shop App


styles.css


script.js

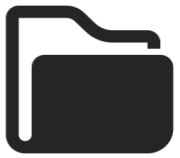
Git under the hood

Working Directory

Web Shop App


index.html


styles.css


images

Repository (.git – hidden folder)

Staging Area
(Index file)

Commits
(Object Folder)


styles.css

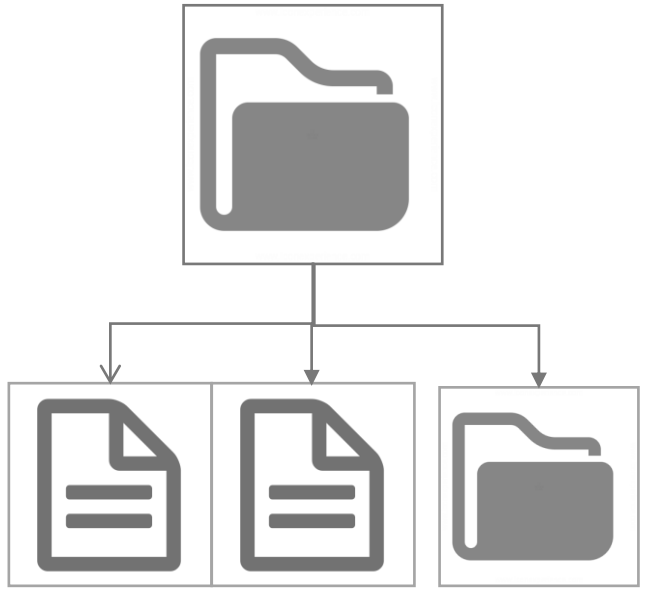

styles.css


styles.css

Git = tracking changes – NOT storing the files again and again

Branches and Commits

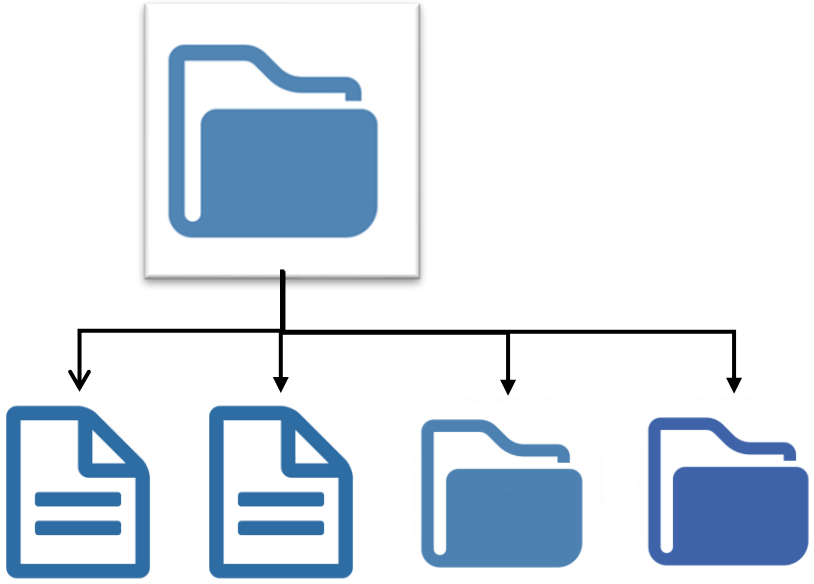
Working Directory / Tree



Master Branch

Commit 1 | Commit 2 | Commit 3

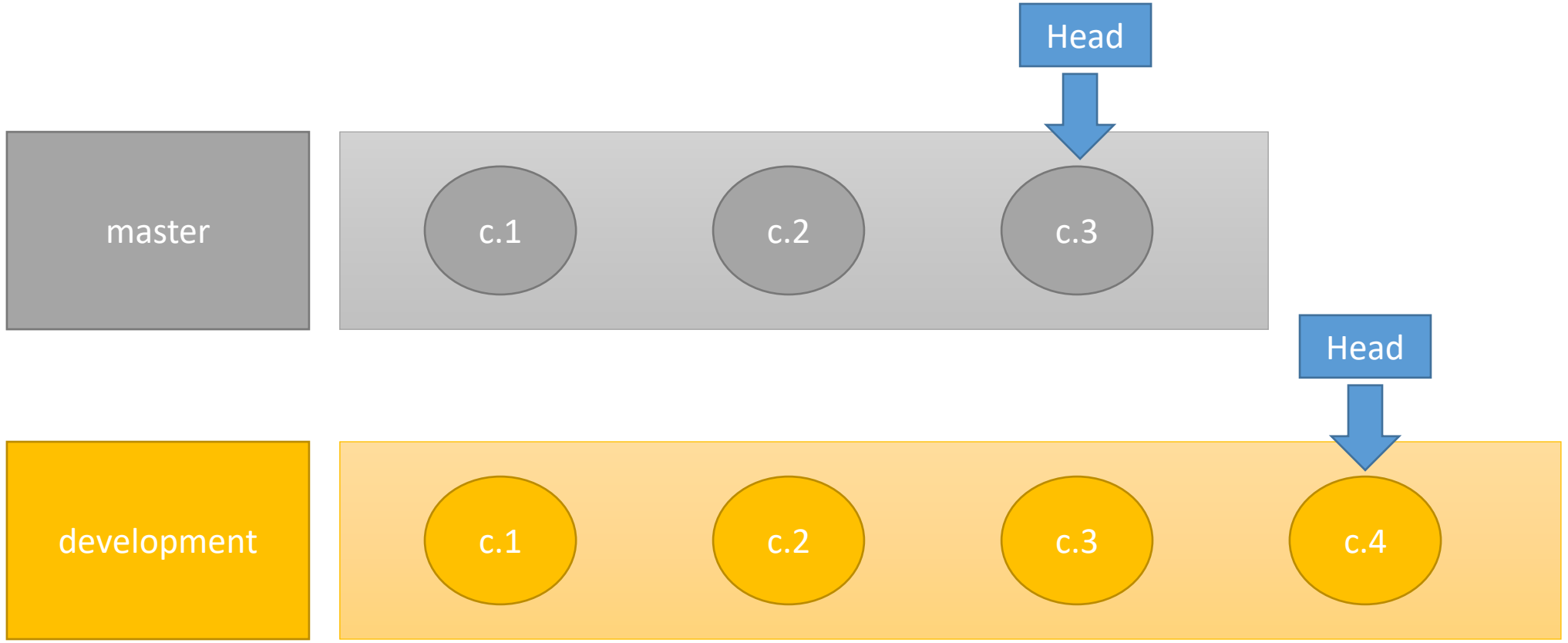
Working Directory / Tree



Development Branch

Commit 1 | Commit 2 | Commit 3 | Commit 4

What is HEAD?



Deleting Data

Working Directory Files
(Already part of previous commits)

Unstaged Changes

Staged Changes

Latest Commits

Branches

Commands Summary - General

git --version	Checks installed Git version
git init	Creates empty Git repository
git status	Check working directory & staging area status
git log	Display all commits of current branch
git ls-files	List tracked files

Commands Summary – Commit Creation and Access

git add filename git add .	Add single file or all WD files to staging area
git commit –m “message”	Creates new commit
git checkout commitID	Checkout commit (detached head)

Commands Summary –Branch Creation and Access

git branch branchName git switch branchName	Creates new branch
git checkout branchName	Go to branch
git checkout –b branchName	Creates and Access new Branch
git merge otherBranch	Bring other branch changes to current branch

Commands Summary –Deleting Data

WD File*	git rm filename git add filename	Run command after file was deleted from current directory
Unstaged Changes	git checkout . git restore filename or .	Revert changes in tracked file
	git clean -df	Delete untracked file
Staged Changes	git reset filename and git checkout – filename or git restore –staged filename .	Removes file from staging area
Latest Commits	git reset HEAD~1	Undo latest (~1) commit
Branches	git branch –D branchName	Delete branch

Git Assignment

- Create a new folder and initialize the repository
- Paste the "instructions.txt" file into this folder
- Add a .txt file named "file-1" containing any text of your choice to the working directory
- Create a second .txt file named "file-2"
- Add "file-1" and "file-2" to the staging area - don't add "instructions.txt"
- Change the initial text you added to "file-1"
- Now add all working directory files to the staging area
- Create the first commit
- Create a second branch named "feature" (two commands are possible)

Git Assignment (Contd…)

- Add a third .txt file ("file-3.txt") to this branch
- Create a new commit
- Add the following text to "file-3": "I will be deleted"
- Add the updated file to the staging area
- Undo the staged change
- Add the following text: "Please add me to the master/main branch"
- Commit this latest change
- Merge the "master" (or "main") branch with "feature"
- Delete the "feature" branch

Working with Stash

git stash	Record the current state of the working directory
git stash apply [index]	Restored the stashed item
git stash list	List down all stash
git stash push	Push new stash in the list (with tag)
git stash pop	Pop the stash item
git stash drop (index)	Drops single stash item

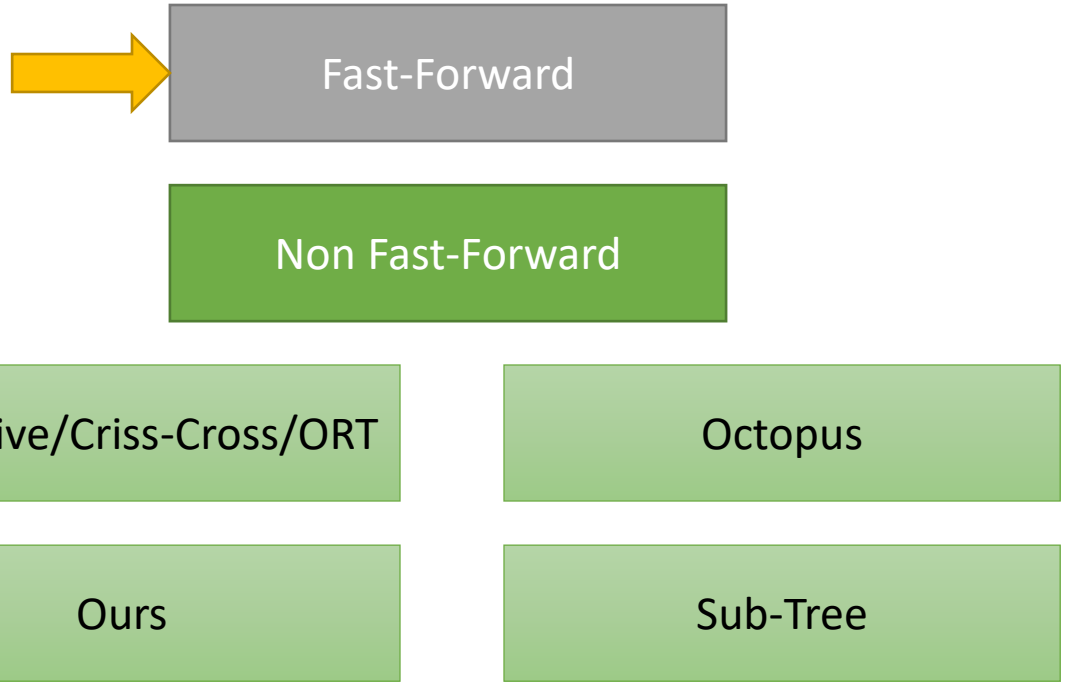
Reference Log

Git keeps track of updates to the tip of branches using a mechanism called reference logs,

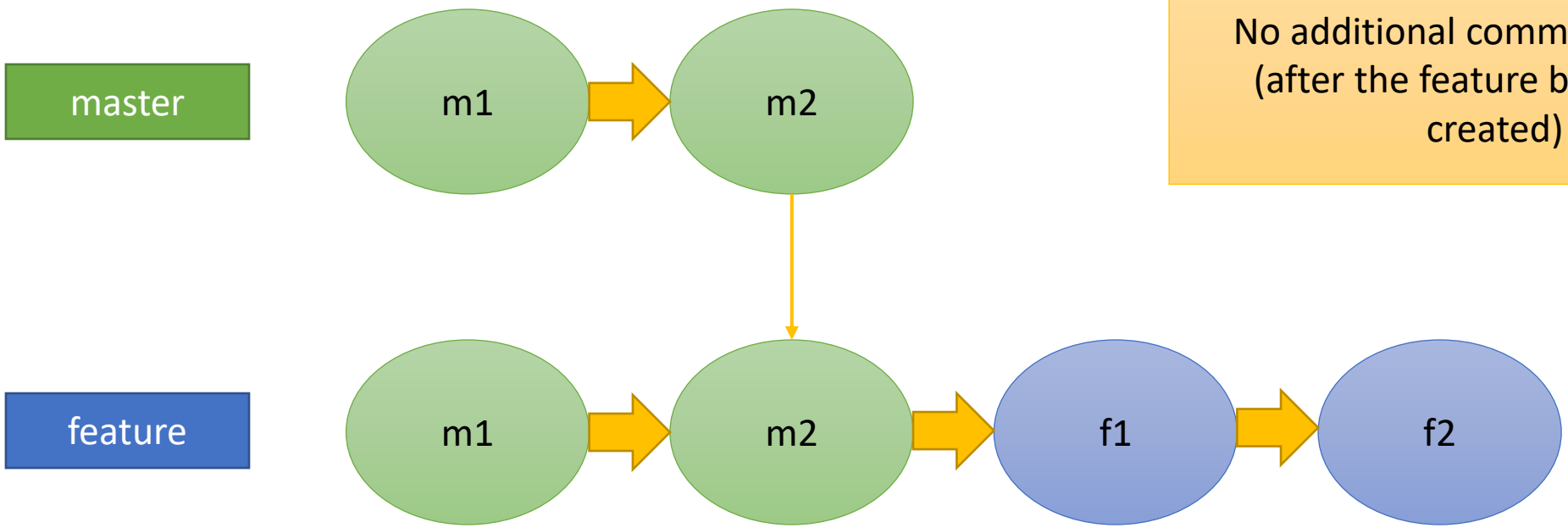
git reflog

Manage reflog information

Merge Types


Fast-Forward
Non Fast-Forward
Recursive/Criss-Cross/ORT
Octopus
Ours
Sub-Tree

Master and Feature – Merge (“fast-forward”)

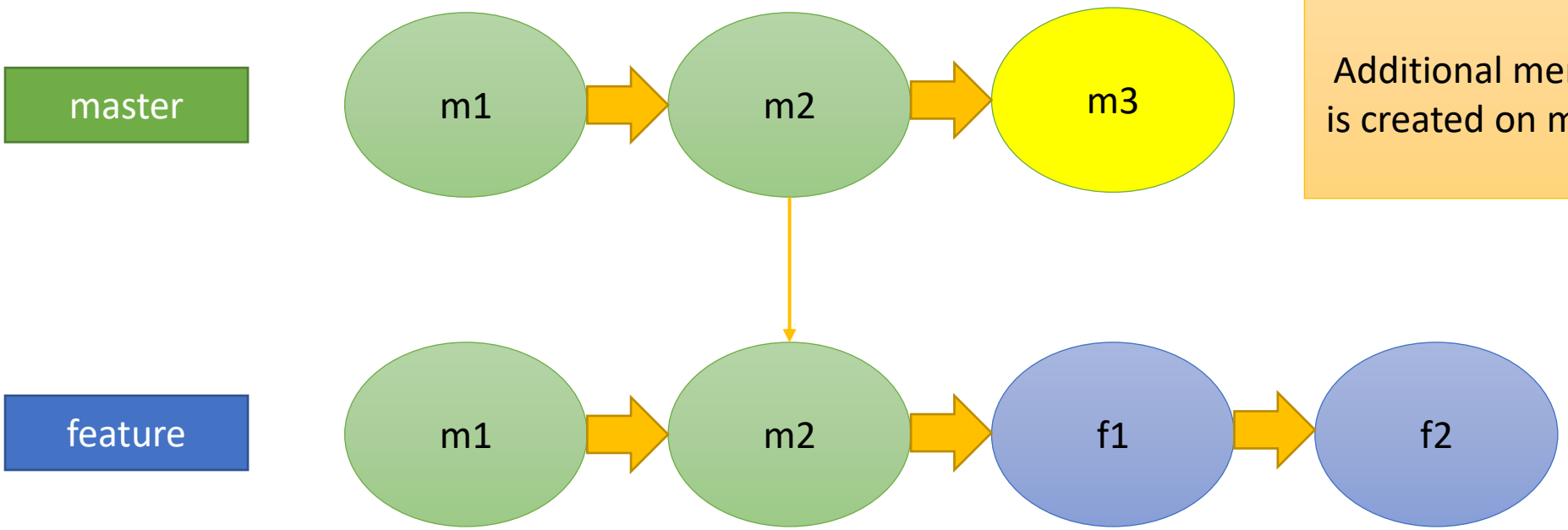


No additional commit in master (after the feature branch was created)

Master and Feature – Merge (“recursive”)

Additional commits in both master and feature branch after the feature branch was created

Additional merge (commit) is created on master branch

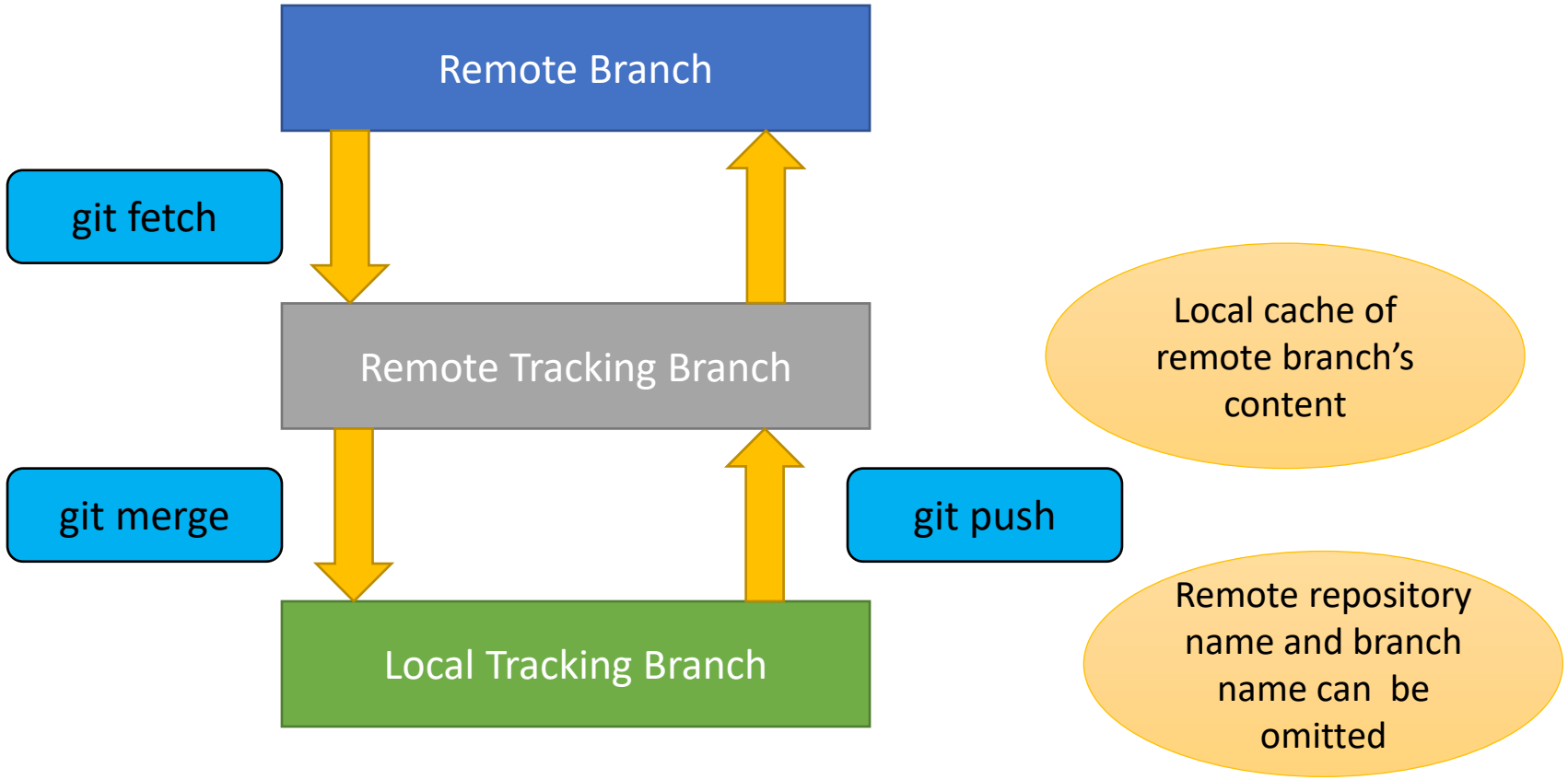


Branch Types

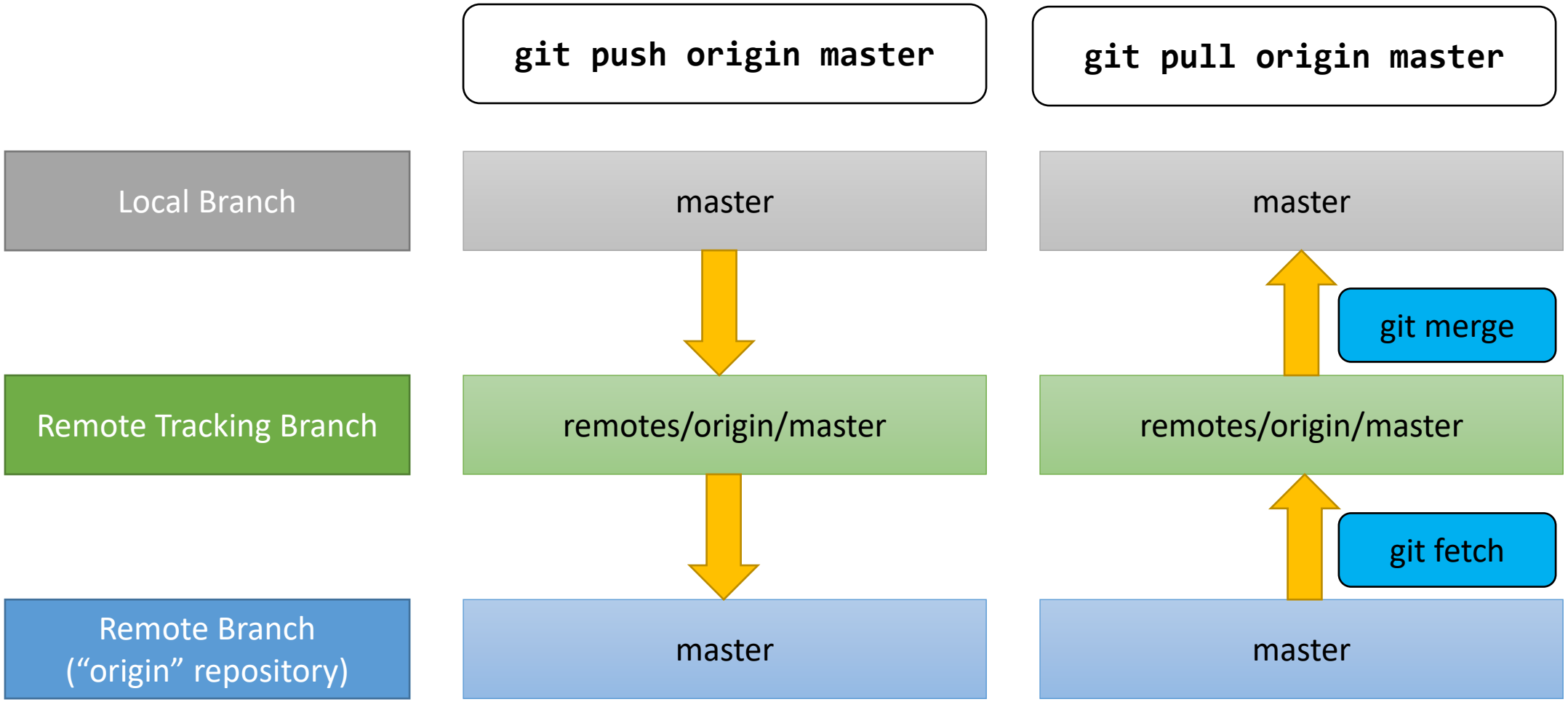
Local Branch	Branch on your machine only
Remote Branch	Branch on remote location

Remote Tracking Branch	Local copy of remote branch (not to be edited)
Local Tracking Branch	Local reference to remote tracking branch (to be edited)

Local & Remote Tracking Branches



More on Branches



Local & Remote Tracking Branches – General Commands

git remote	Show remote servers
git branch -a	List all branches
git branch -r	List all remote branches
git remote show origin	Show detailed configuration
git branch -vv	List local tracking branches and their remotes
git branch –track branchName origin/branchName	Creates local tracking branches

Local & Remote Tracking Branches – Deleting Branches

Branches	git branch –delete –remote origin/branchName	Delete remote reference branch from local machine
	git push origin –delete branchName	Delete the branch from remote server
Commit	git reset –hard HEAD~1	Delete last commit locally
	git push –force origin branchName	Delete last commit on remote server