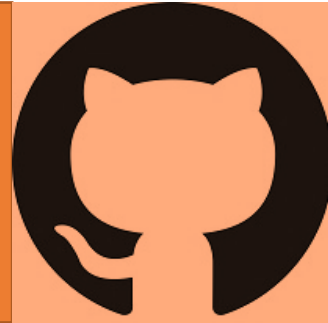




Git and Github



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

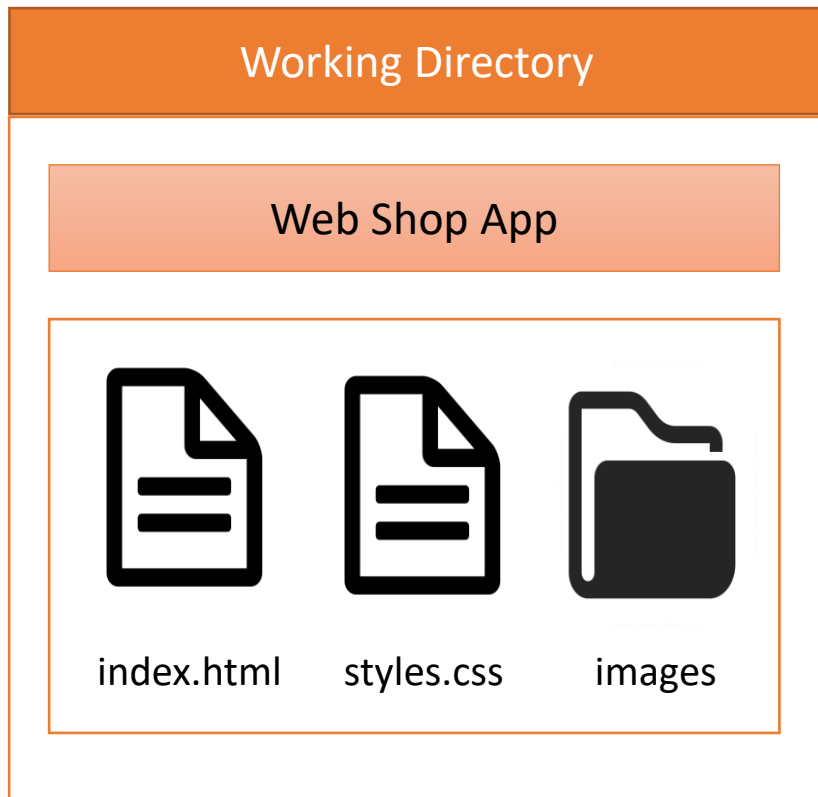


Largest Development Platform

Cloud Hoisting and Collaboration
Provider

Git Repository Hoisting

How does Git Works?



Commit 1

“snapshot 1”



Commit 2

“snapshot 2”

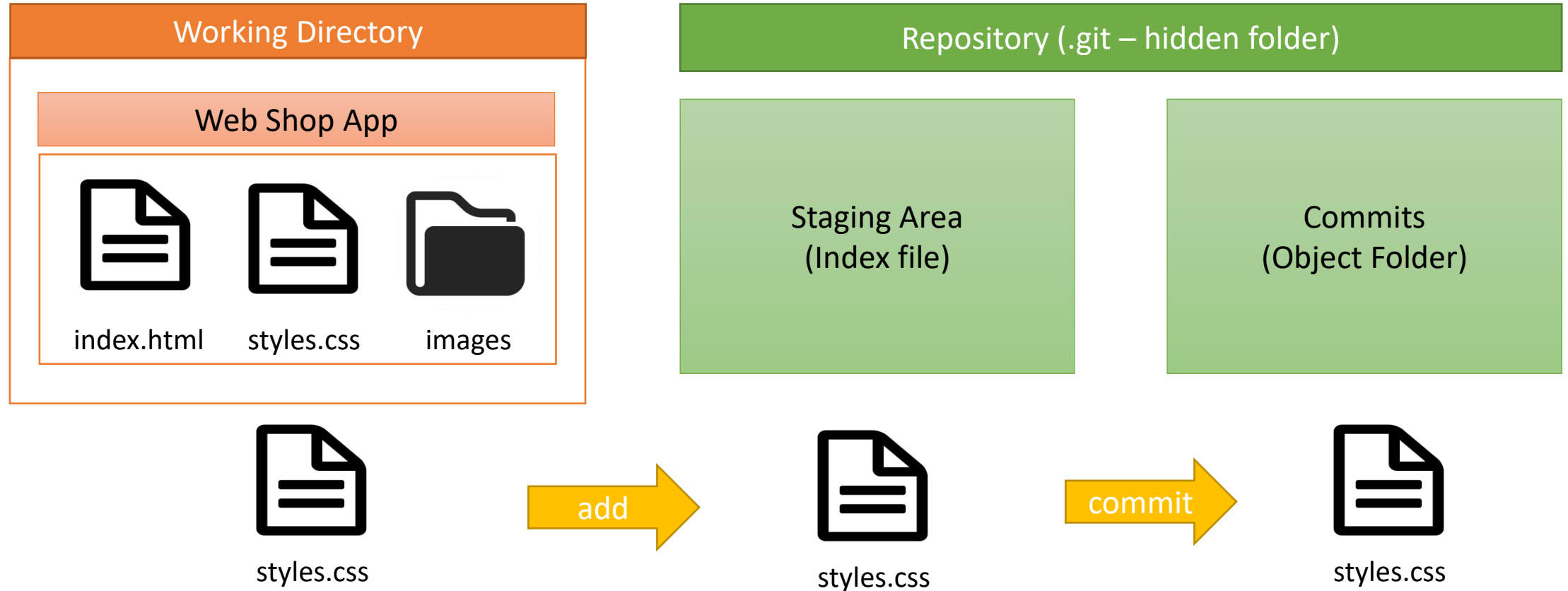


Commit 3

“snapshot 3”



Git under the hood

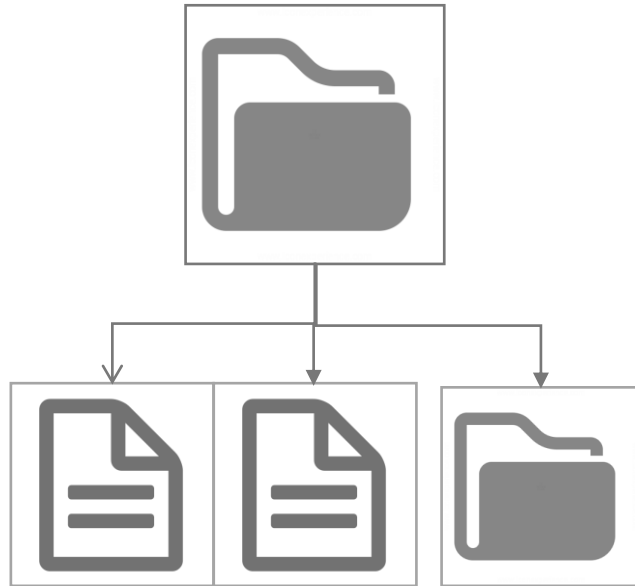


Git = tracking changes – NOT storing the files again and again

Branches and Commits



Working Directory / Tree



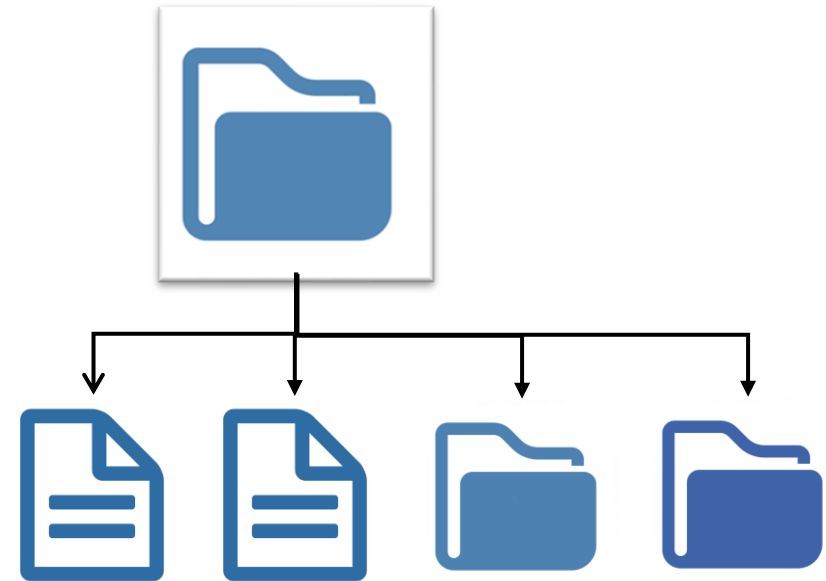
Master Branch

Commit 1

Commit 2

Commit 3

Working Directory / Tree



Development Branch

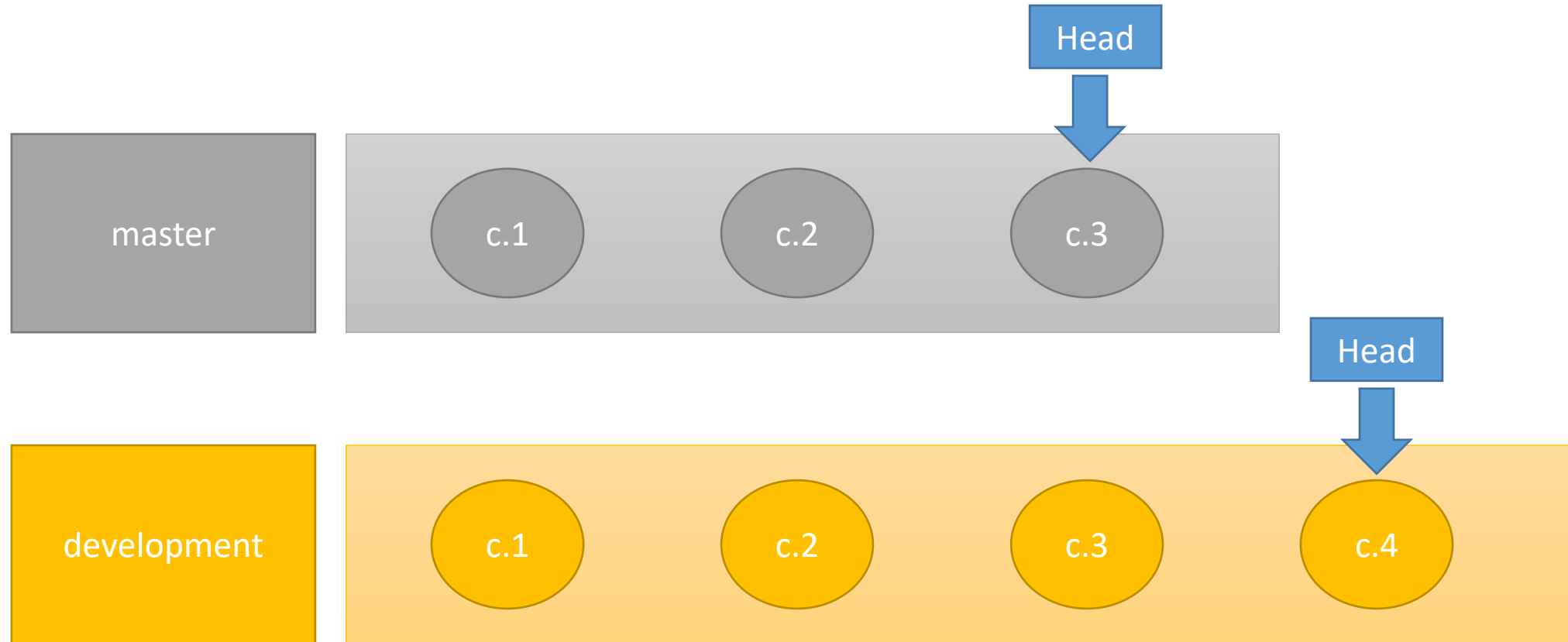
Commit 1

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**

Creates new branch

git checkout -b
branchName

git switch -c **branchName**

Creates new branch and switch into it

git checkout **branchName**
git switch **branchName**

Go to branch

git merge **otherBranch**

Bring other branch changes to current
branch

Commands Summary –Deleting Data



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

Git Assignment

- Create a new folder and initialize the repository
- Create 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