DevOps Bootcamp

Version Control with Git - Lecture #2

Dan Morgenstern

danm@sela.co.il

22/05/2022

(Please write “I am here (your name)” in the zoom chat to register your attendance)

Agenda

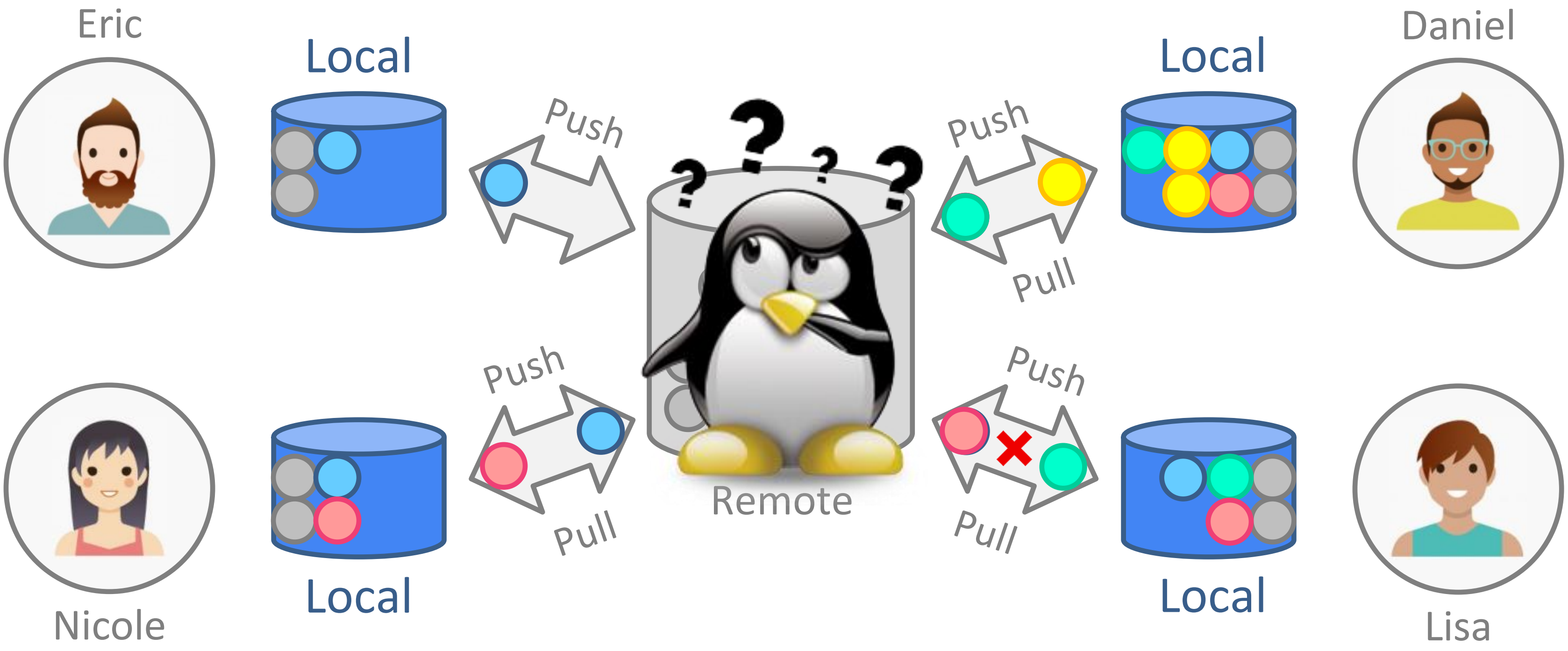
- Bootcamp News
- Git Introduction Recap
- Working Locally (Basics)
- Working with Branches
- Merge & Rebase
- Undoing Changes

Git Introduction Recap

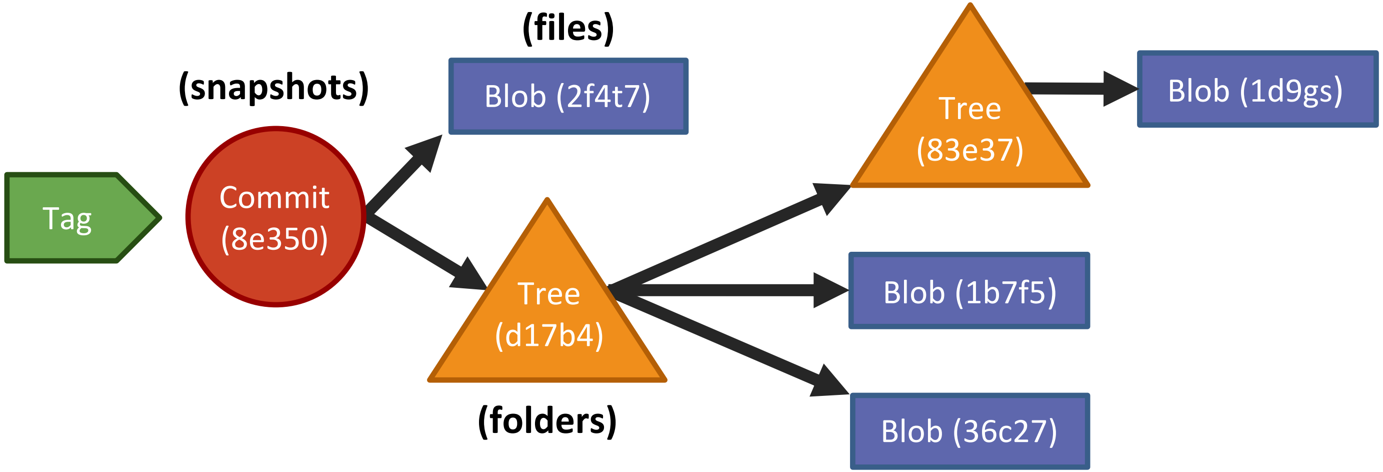
Introduction Summary

Git is a free and open source distributed version control system designed with performance, security and flexibility in mind

Git – Distributed but Centralized

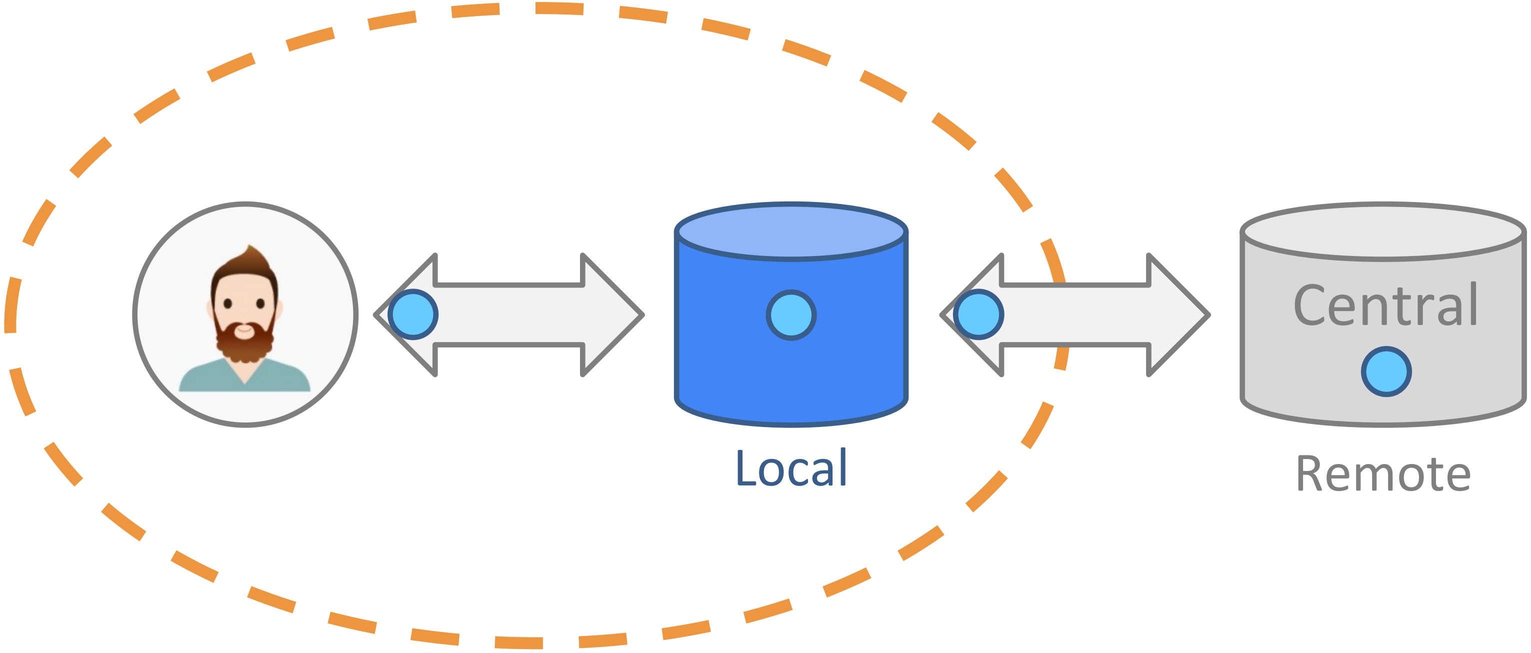


Git Structure - Objects



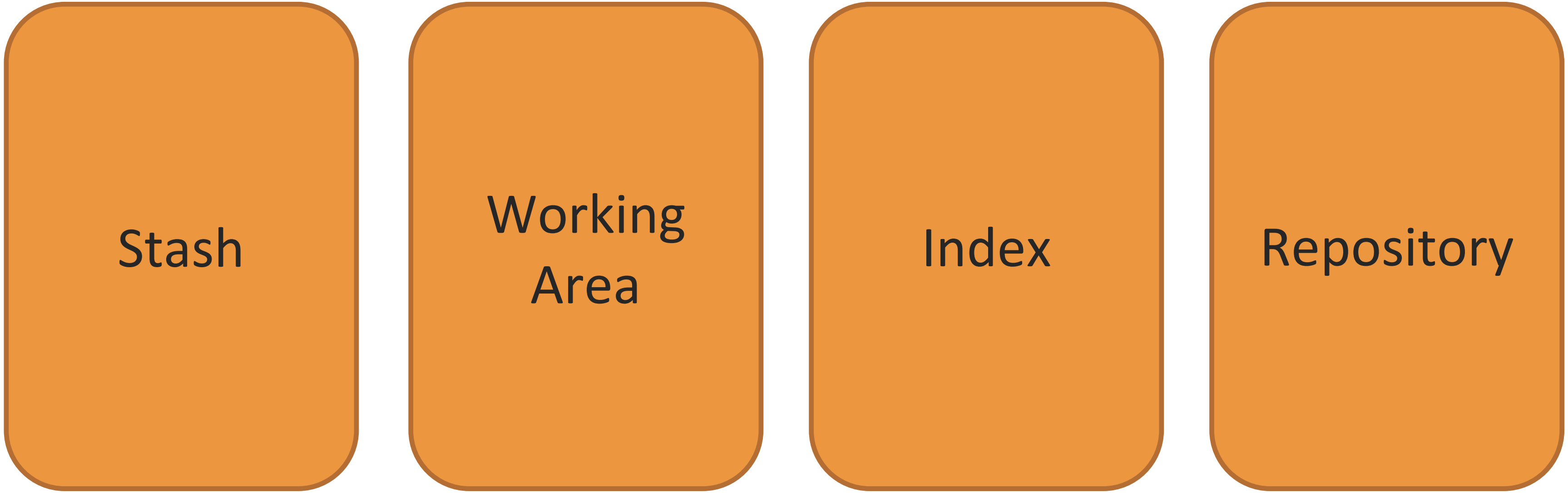
Commit = Snapshot at some point in time | **Tag** = Reference to a commit

Summary



Working Locally (Basics)

The Four Areas



Stash

Working Area

Index

Repository

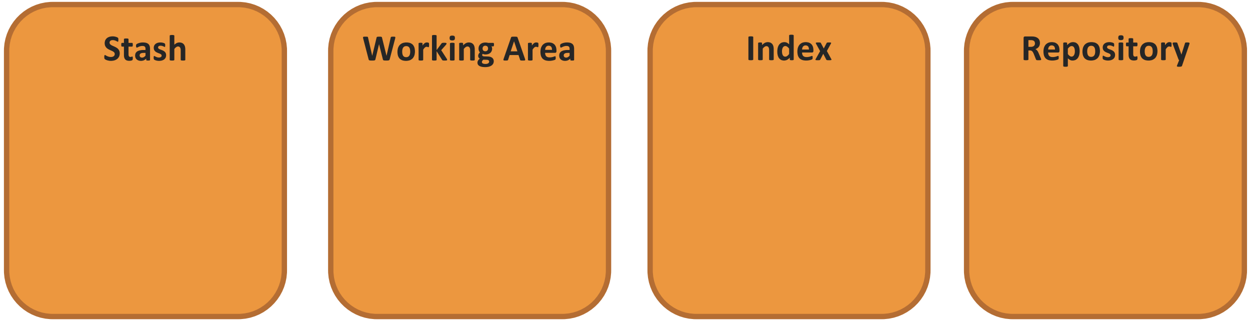
Working Locally (Commands)

Knowing a command means understanding how it affects the 4 areas

The best way to learn the commands is by using the command line

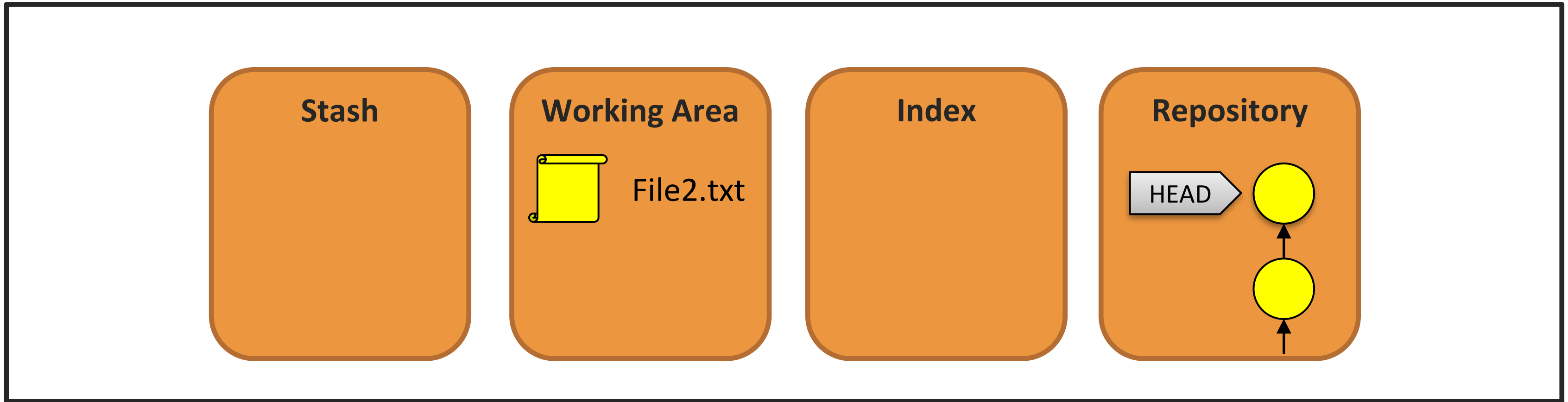
Git clients are good as long as you understand what they are doing

Working Locally (`git init`)



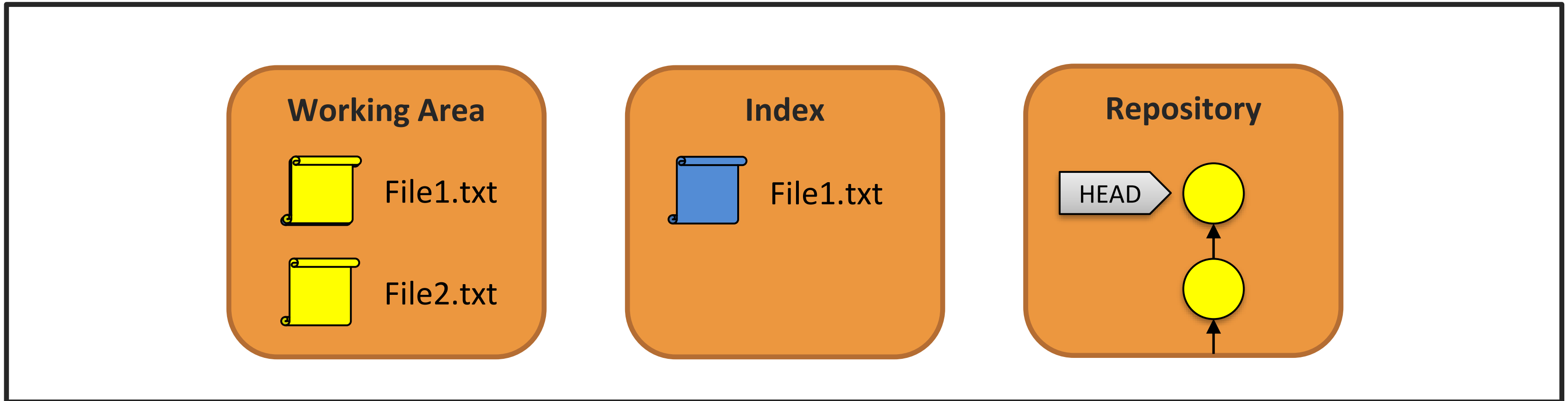
```
$ git init

Initialized empty Git repository in
C:/Users/leonj/Desktop/MyRepo/.git/
```

Working Locally (git clone)



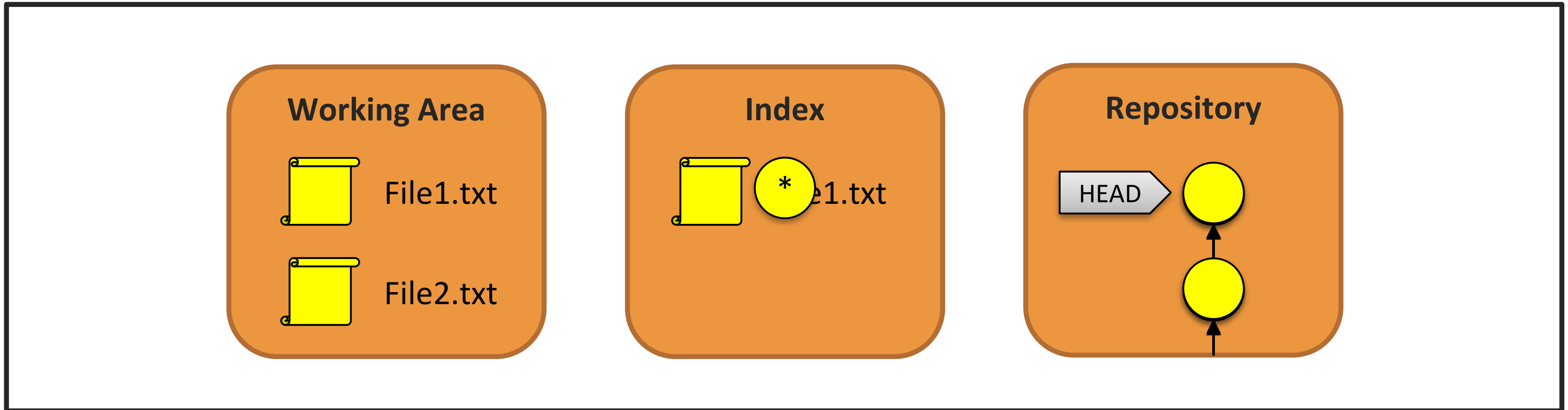
```
$ git clone <RepoUrl> “Folder”

  Cloning into ‘Folder’...

  Unpacking objects: 100% (19/19), done.
```

Working Locally (git add)



```
$ git add “File1.txt”

$ git status
  Changes to be committed:
        modified:   File1.txt
```

Working Locally (git commit)



Working Area

File1.txt

File2.txt

Index

*

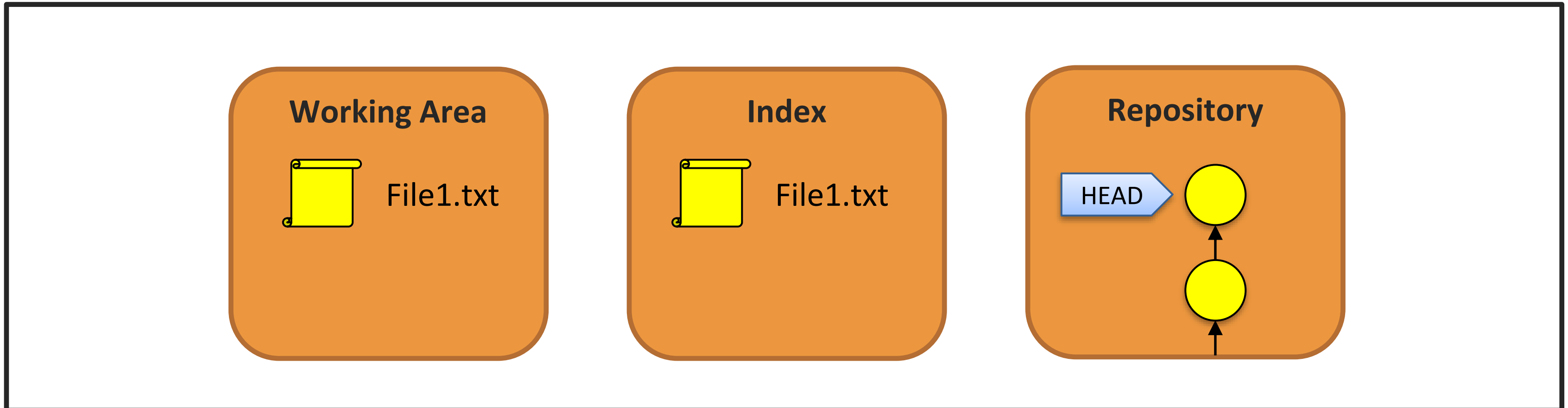
e1.txt

Repository

HEAD

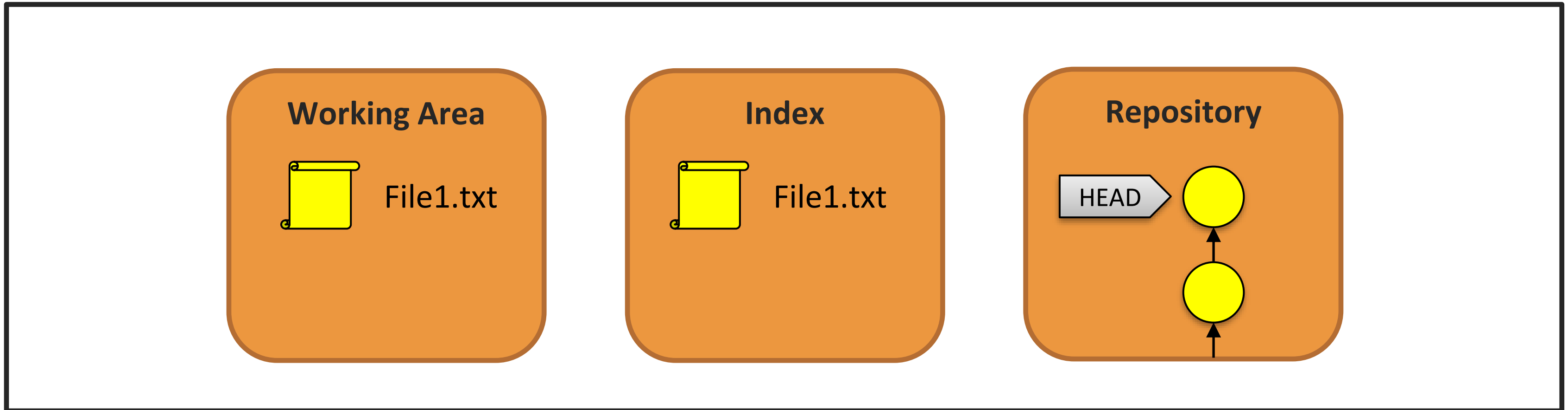
```
$ git commit –m “message”

   [master (root-commit) 22f1a52] message
    1 file changed, 0 insertions(+), 0 deletions(-)
    create mode 100644 MyFile.txt
```

Working Locally (git rm --cached)



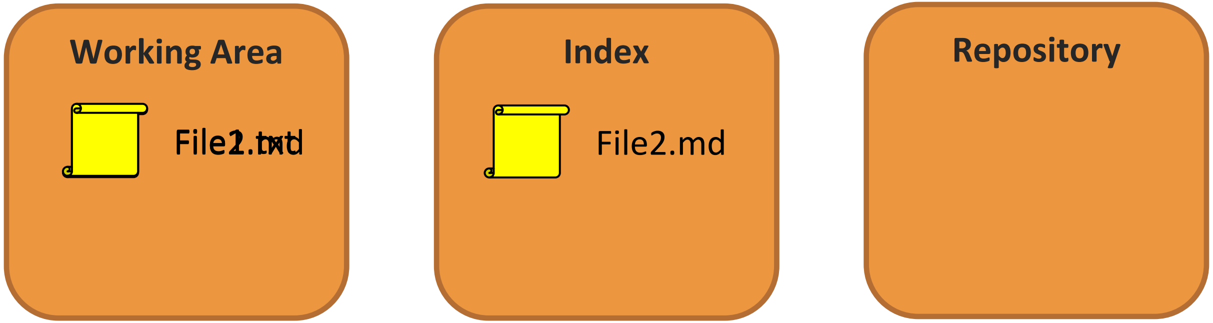
```
$ git rm “File1.txt”
  error: the following file has changes staged in the index
  (use --cached to keep the file, or -f to force removal)
$ git rm --cached “File1.txt”
```

Working Locally (git rm -f)



```
$ git rm “File1.txt”
  error: the following file has changes staged in the index
  (use --cached to keep the file, or –f to force removal)
$ git rm -f “File1.txt”
```

Working Locally (git mv)



```
$ git mv “File1.txt” “File2.md”

$ git status
  Changes to be committed:
        renamed:    File1.txt-> File2.md
```

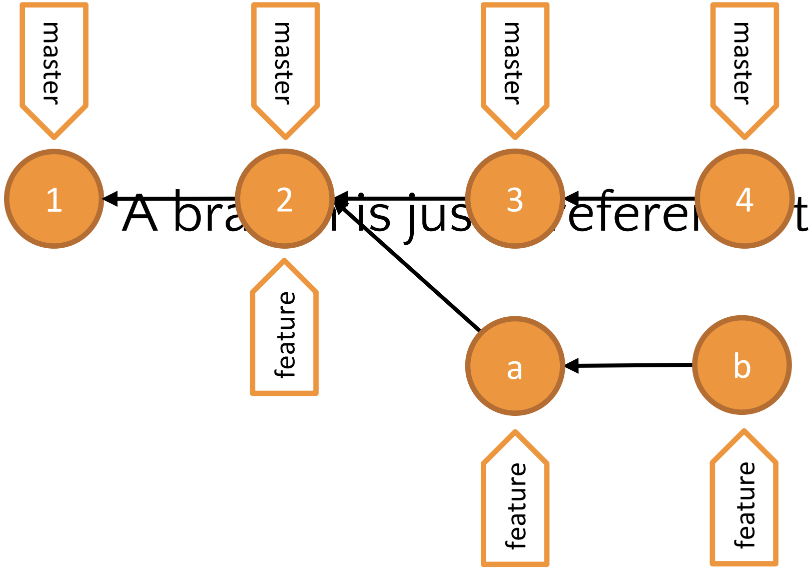
Working Locally (Basics)

Demo

Working with Branches

Working Locally – What Branches really are?

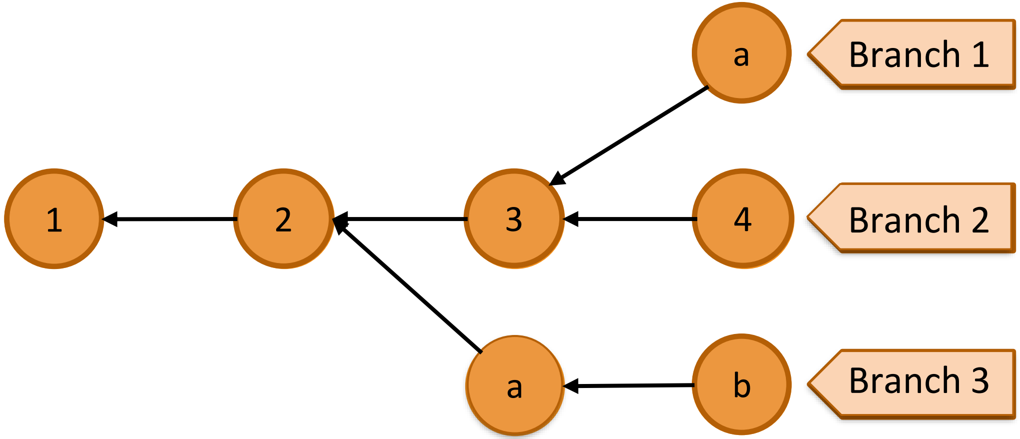
A bra r is jus eferel to a commit



Create "feature" Branch

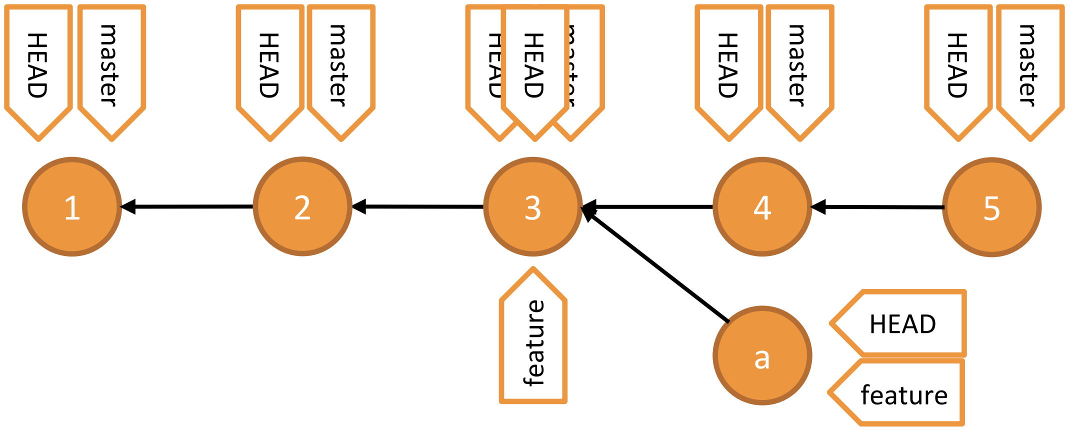
Working Locally – What Branches really are?

Or rather, the entrance to the sequence…



Working Locally – Losing the Head

HEAD is just a reference to the current commit



Checkout commit #3

Create "feature" Branch

Working Locally – Losing the Head

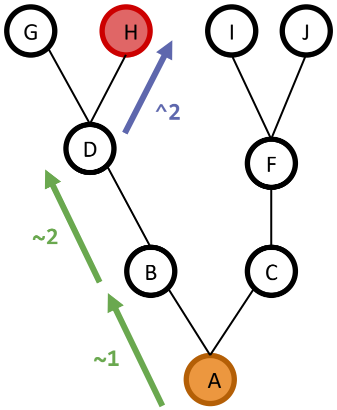
- HEAD is a reference to the current commit
- Happens when the HEAD is not referencing the branch tip
- Commits **can** be created directly from a detached head
- To preserve new commits you must create them under a branch

Working Locally – Relative References

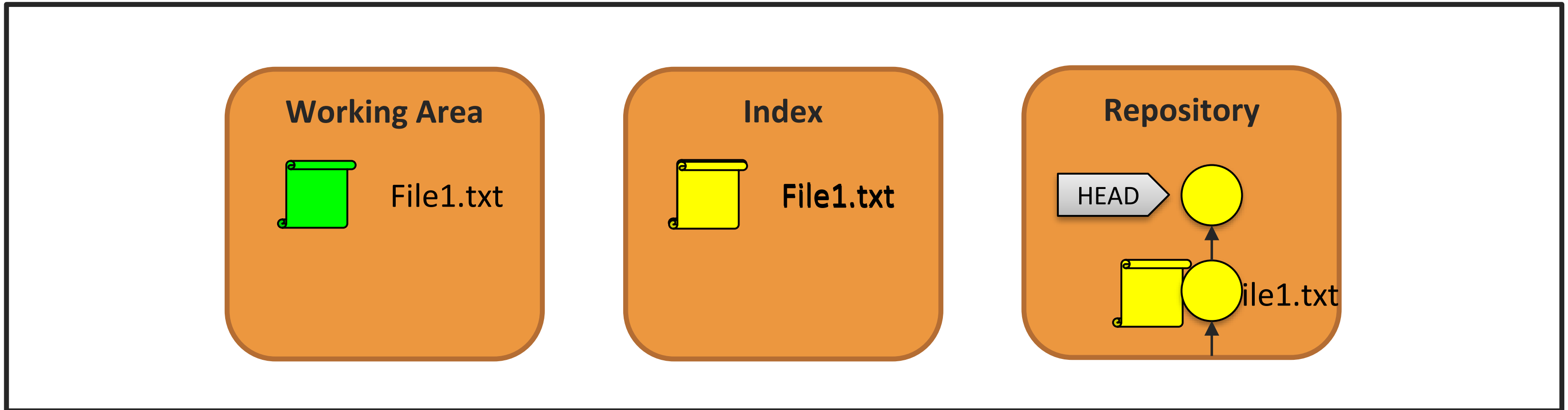
Using x~n
(n commits before x)

Using xˆn
(n parent of x)

H = A ~2 ^2

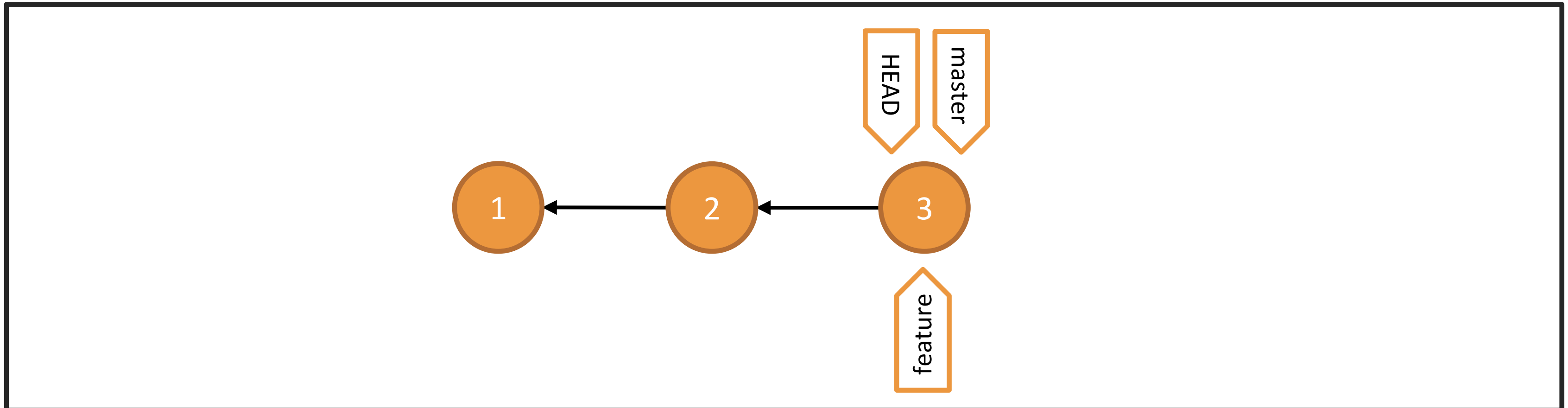


Working Locally (git checkout)



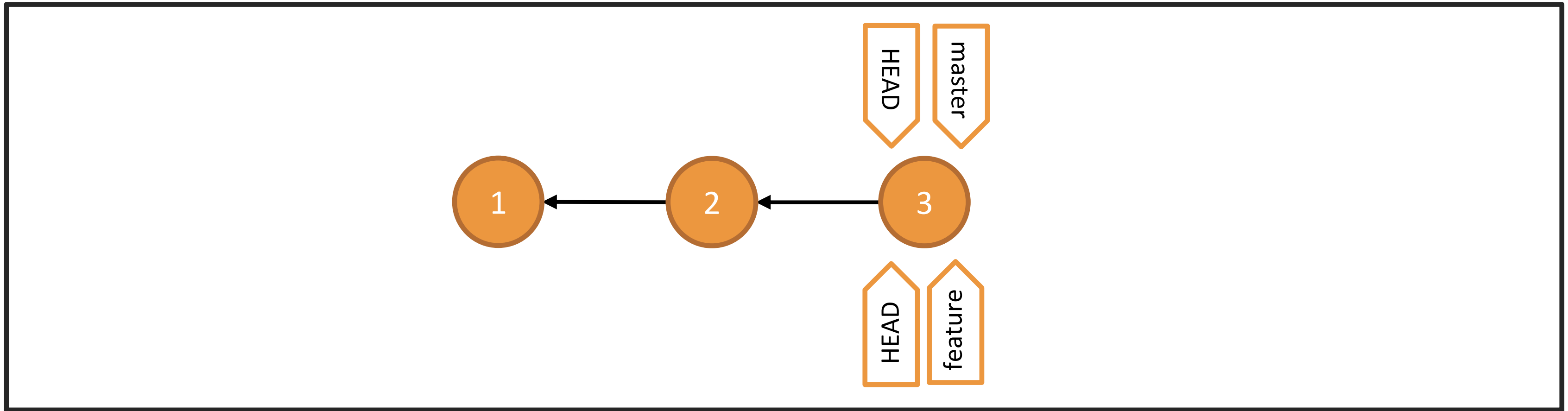
```
$ git checkout HEAD~1

  Switched to branch ‘master’

  Your branch is up-to-date with 'origin/master'
```

Working Locally (git branch)



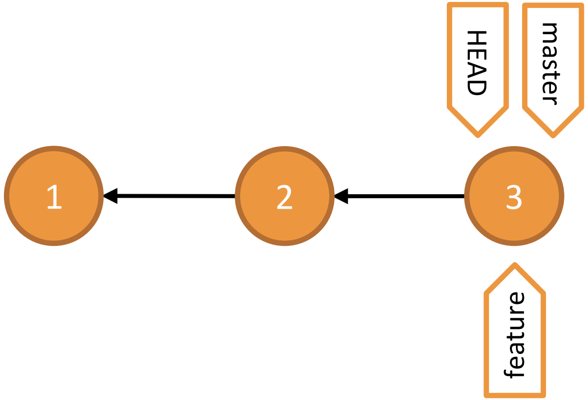
```
$ git branch “feature”

$ git branch

   feature
 * master
```

Working Locally (git checkout -b)



```
$ git checkout -b “feature”
   Switched to a new branch 'feature'
$ git branch
 * feature
   master
```

Working Locally (git branch -d)



```
$ git branch -d “feature”
   Deleted branch feature (was 41f1e7a).
```

Working with Branches

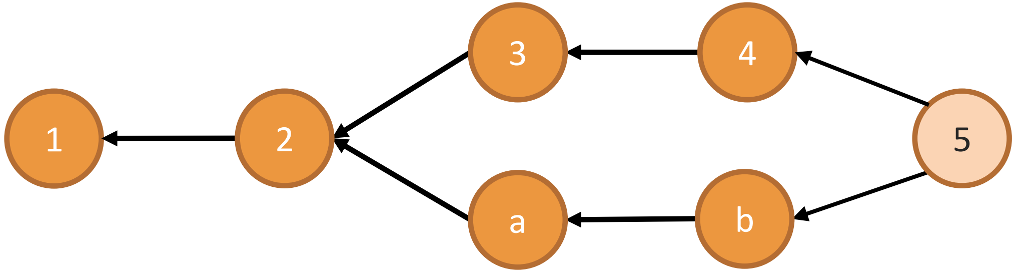
Demo

Merge & Rebase

GIT MERGE

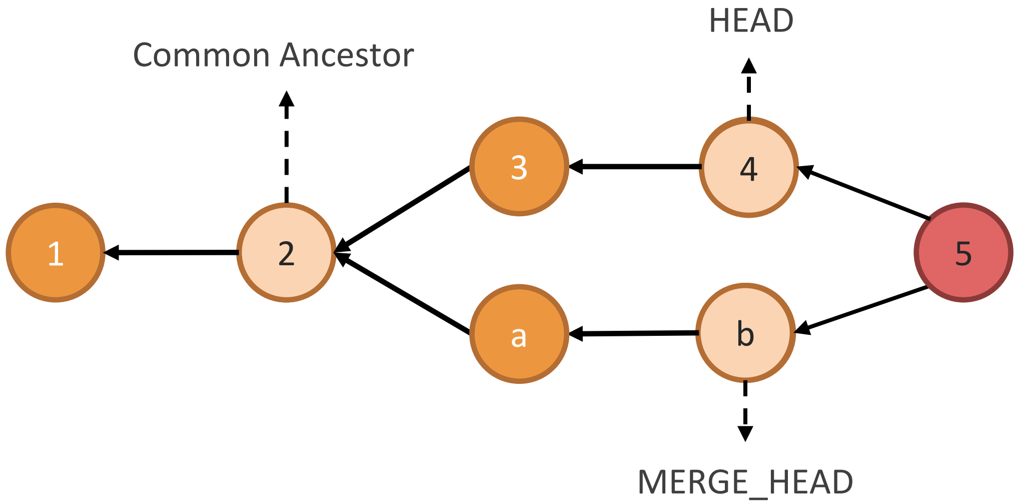
Working Locally - Merging Simplified

- Merge is just a commit with two parents



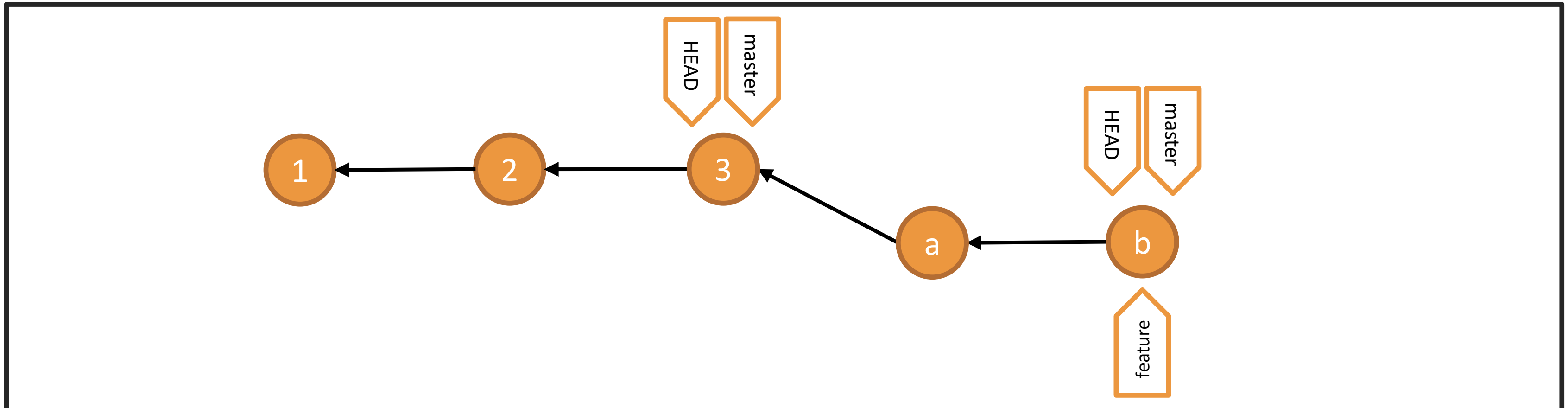
- Merge command says: “merge the branch X into the current branch”

Working Locally – 3 Way Merge Algorithm



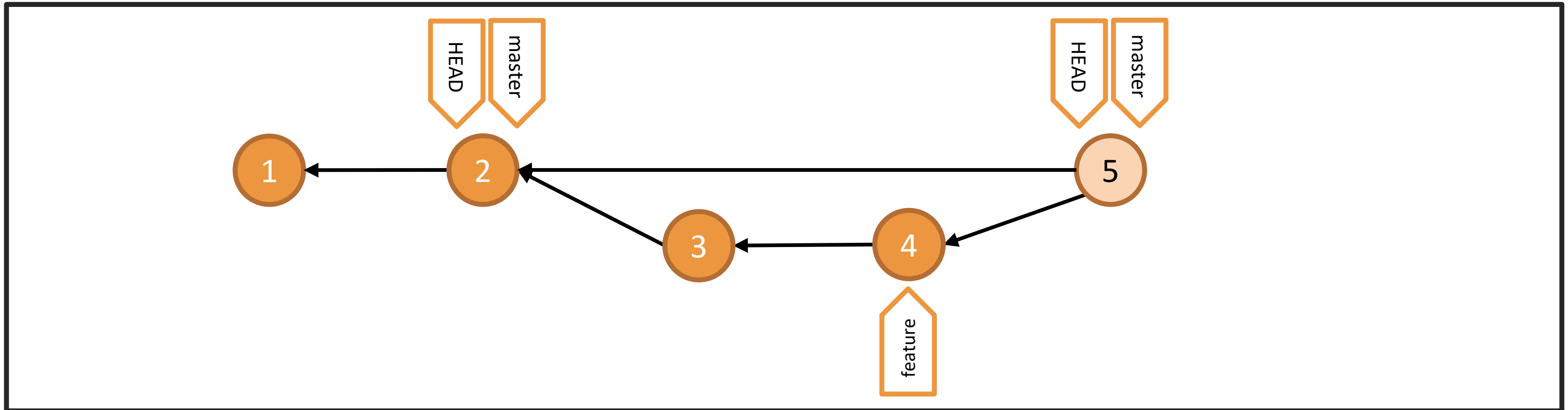
- Compare both commits to merge with it common ancestor

Working Locally (git merge|Fast-Forward)



```
$ git checkout master

$ git merge feature
  Updating b01bdbe..18ddbab
  Fast-forward
```

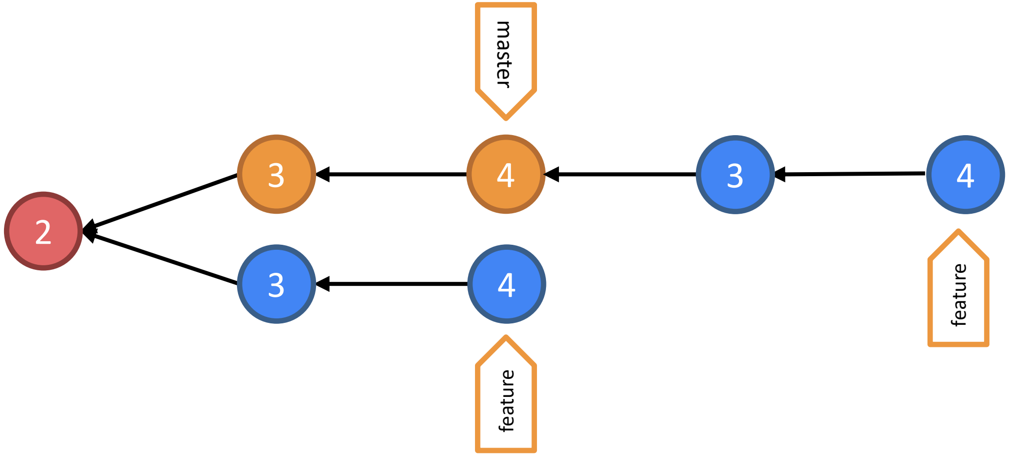
Working Locally (git merge|--no-ff)



```
$ git checkout master
   Switched to branch 'master'

$ git merge feature --no-ff
   Merge made by the 'recursive' strategy.
```

Working Locally – Rebasing Explained

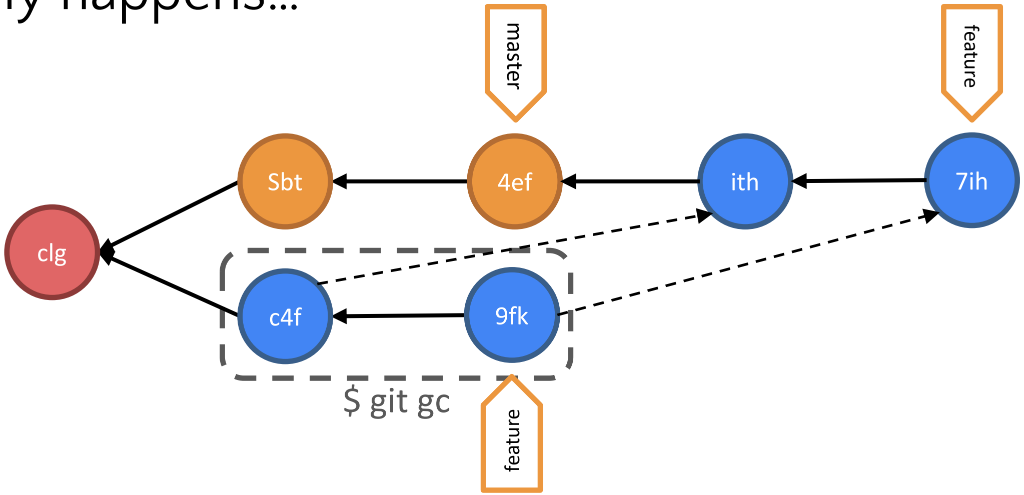
- How it Looks…



- Commits 3 and 4 are moved and references are updated

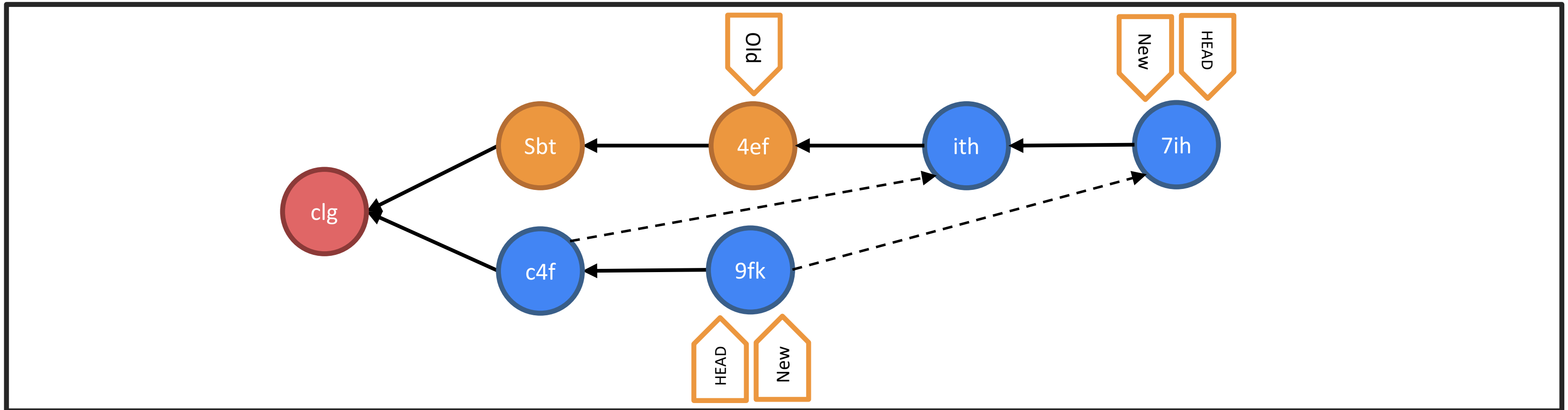
Working Locally – Rebasing Explained

- What really happens…



- Commits 3-4 are deleted and identical commits are created instead

Working Locally (git rebase)



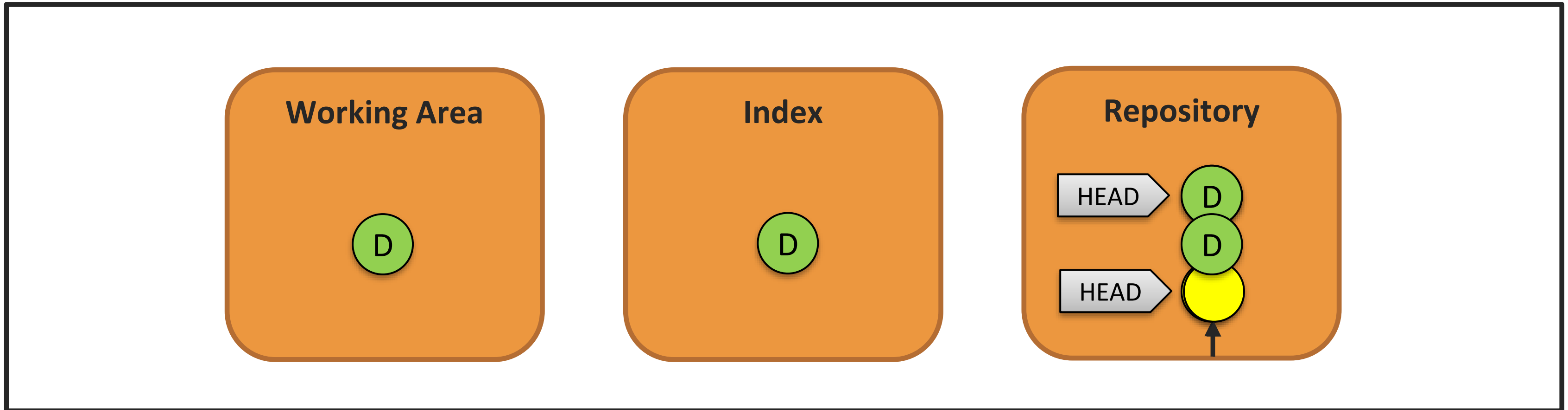
```
$ git checkout New

$ git rebase Old

   First, rewinding head to replay your work on top of it...
   Applying: ...
```

Merge & Rebase

Demo

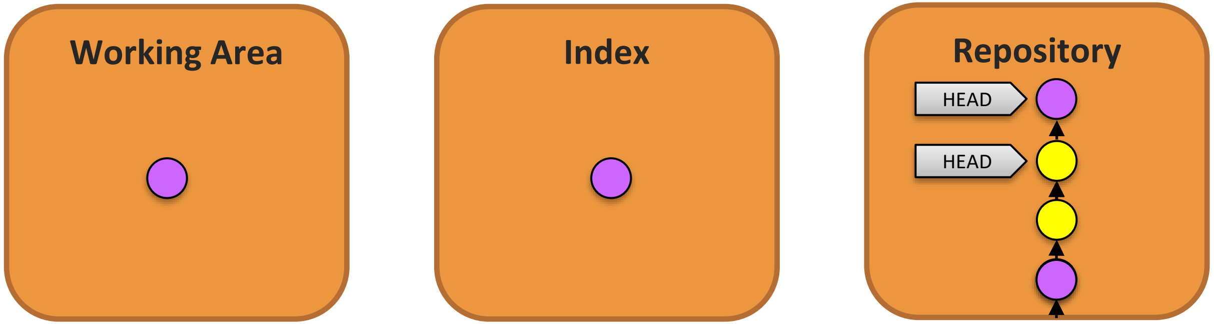
Undoing Changes

Working Locally (git revert)



```
$ git revert HEAD

  Revert "commitMessage"
  New Commit Message
  This reverts commit
  6b32e600dff05e6b27e0e42eeb829ee735134336
```

Working Locally (git cherry-pick)



```
$ git cherry-pick HEAD~2

   [master c85d13a] commit
    Date: Sat Mar 11 18:38:24 2017 +0200
    1 file changed, 1 insertion(+)
```

Working Locally (`git reset`)

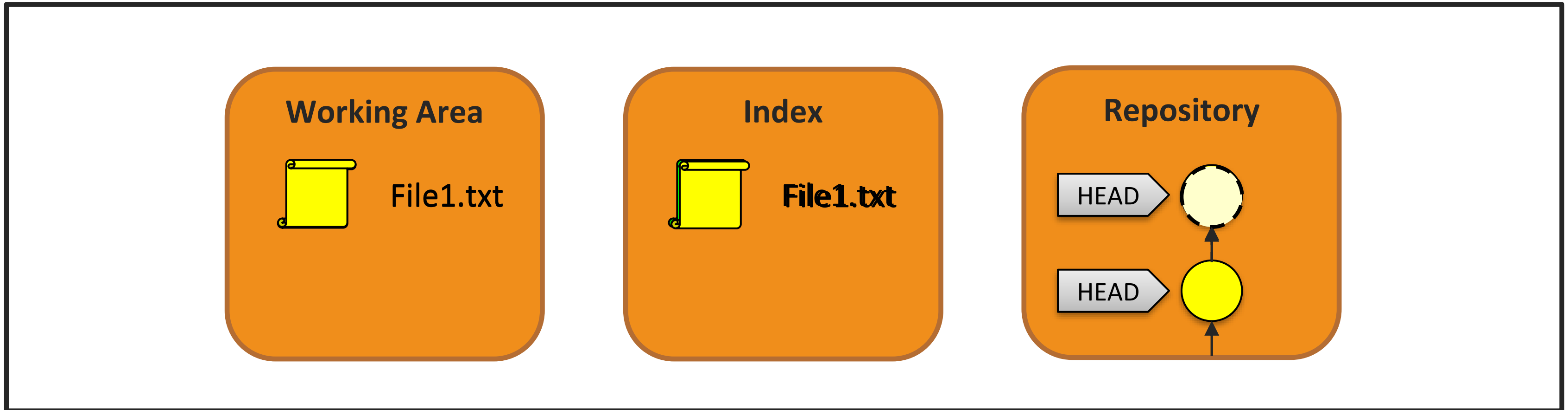
$git reset moves the current branch, and optionally copies data from the Repository to the other areas

`--soft`

`--mixed`

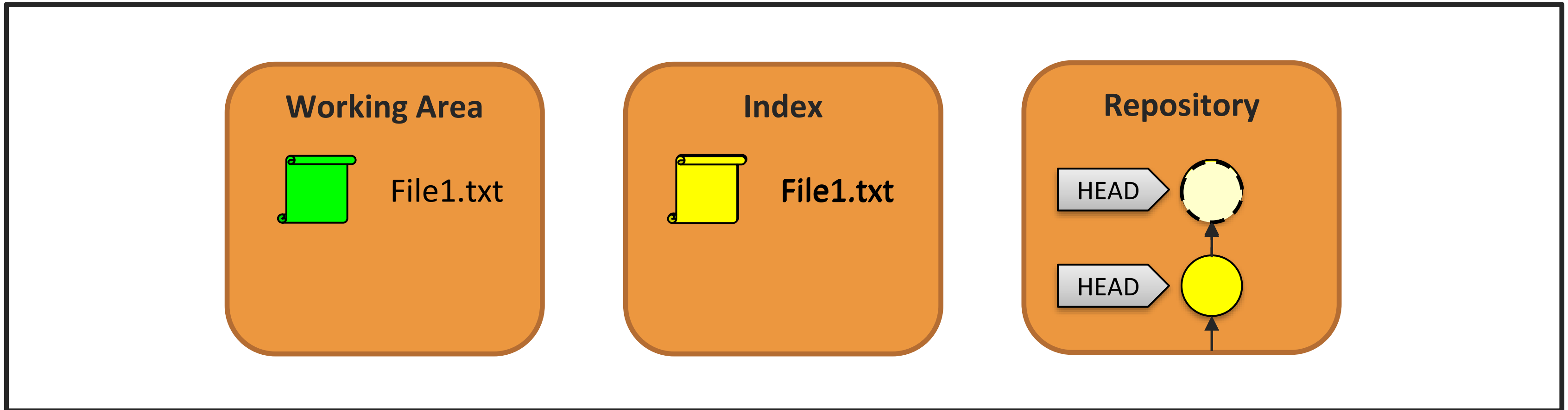
`--hard`

Working Locally (`git reset --hard`)



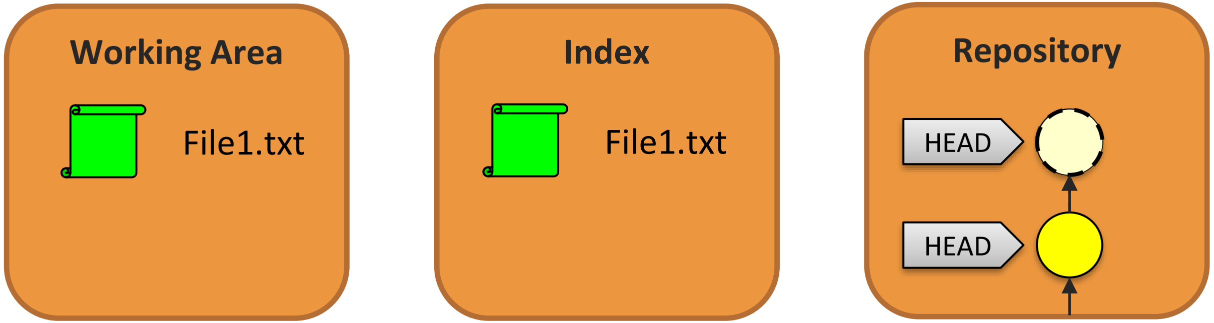
```
$ git reset --hard HEAD~1

  HEAD is now at ec288f2 CommitMessage
```

Working Locally (`git reset --mixed`)



```
$ git reset --mixed HEAD~1

  HEAD is now at ec288f2 CommitMessage
```

Working Locally (git reset --soft)



```
$ git reset --soft HEAD~1

  HEAD is now at ec288f2 CommitMessage
```

Questions

DevOps Bootcamp

Version Control with Git – Lecture #1

Natali Cutic
natalic@sela.co.il

Noam Amrani
noama@sela.co.il

Pushing YOU forward