



IA Frameworks

Introduction to Git





Version control

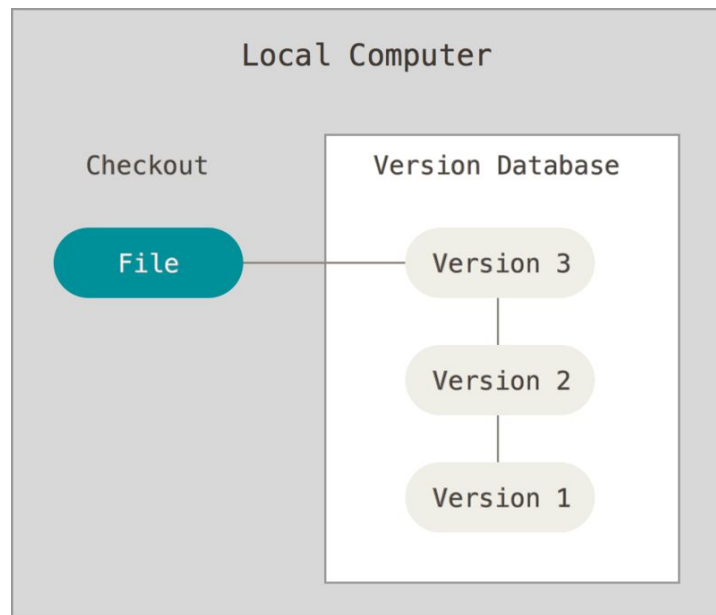


Version control

- System that records changes to a file or set of files over time
- Revert selected files back to a previous state
- Revert the entire project back to a previous state
- Compare changes over time
- Easy recovery.

Local Version Control Systems

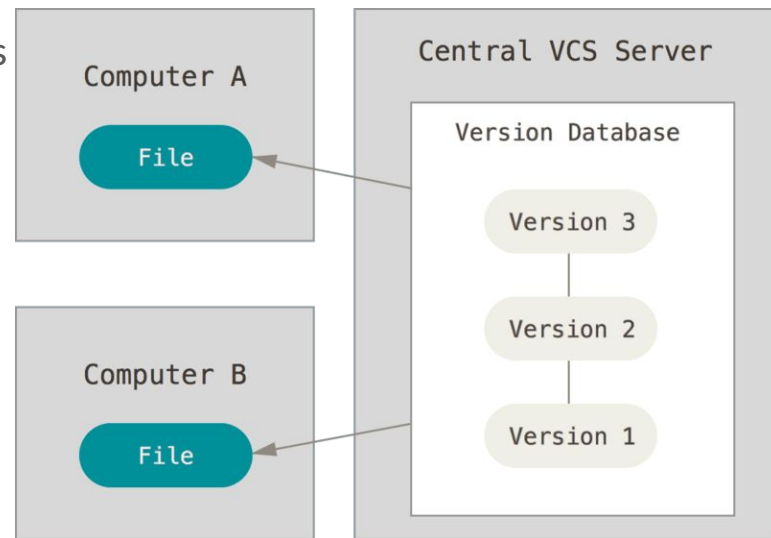
- The Revision Control System (RCS)



(source Git-smc)

Centralized Version Control Systems

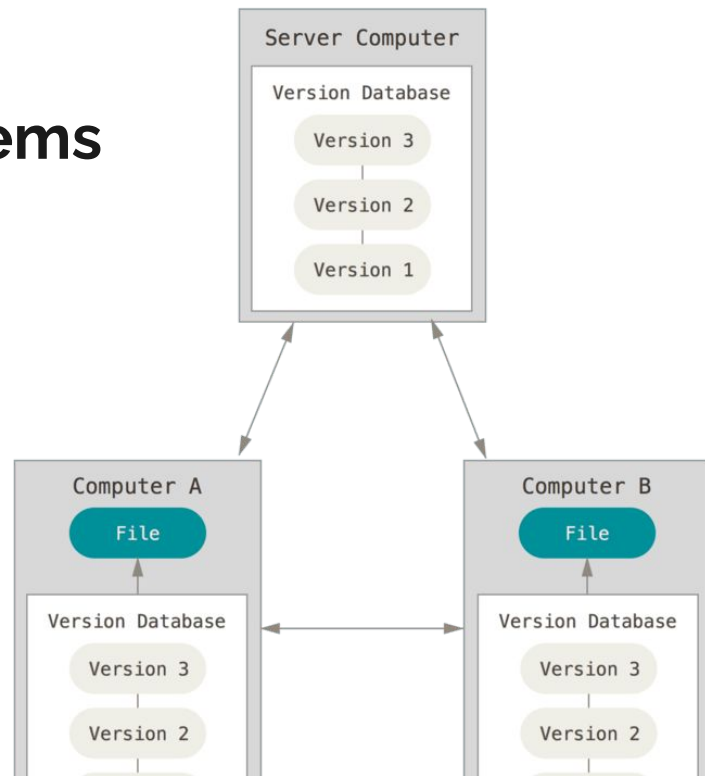
- single server that contains all the versioned files
- number of clients that check out files from that central place
- Easy to administrate
- single point of failure



(source Git-smc)

Distributed Version Control Systems

- clients fully mirror the repository
- every clone is a full backup
- collaborate with different groups of people in different ways simultaneously



(source Git-smc)



Git



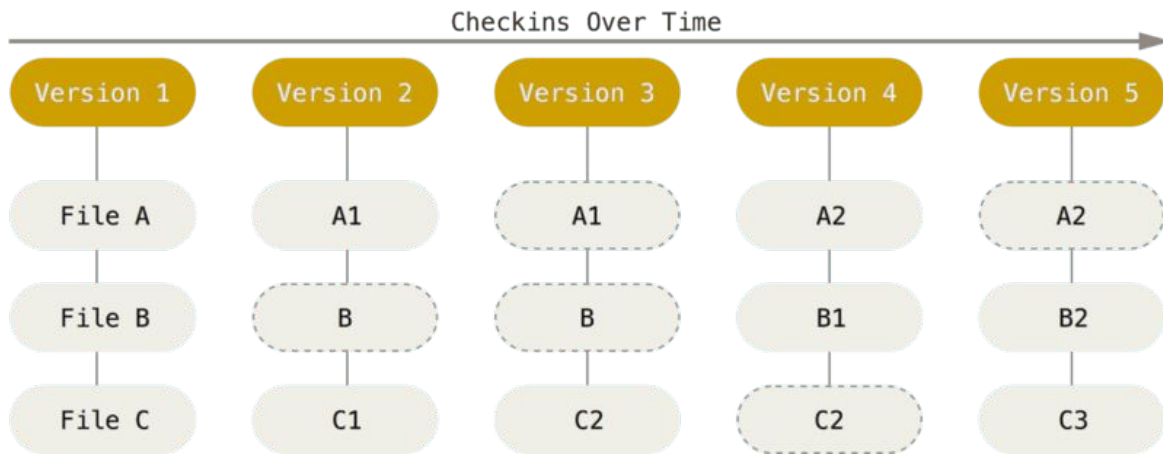
What is git?

- Distributed Version Control Systems
- Created in 2005 by Linus Torvalds
- Most used
- git ≠ Github



What is git?

- Series of snapshots
- Commit
- Store references to the commit
- Only stores changes
- Most operations are local



(source Git-smc)

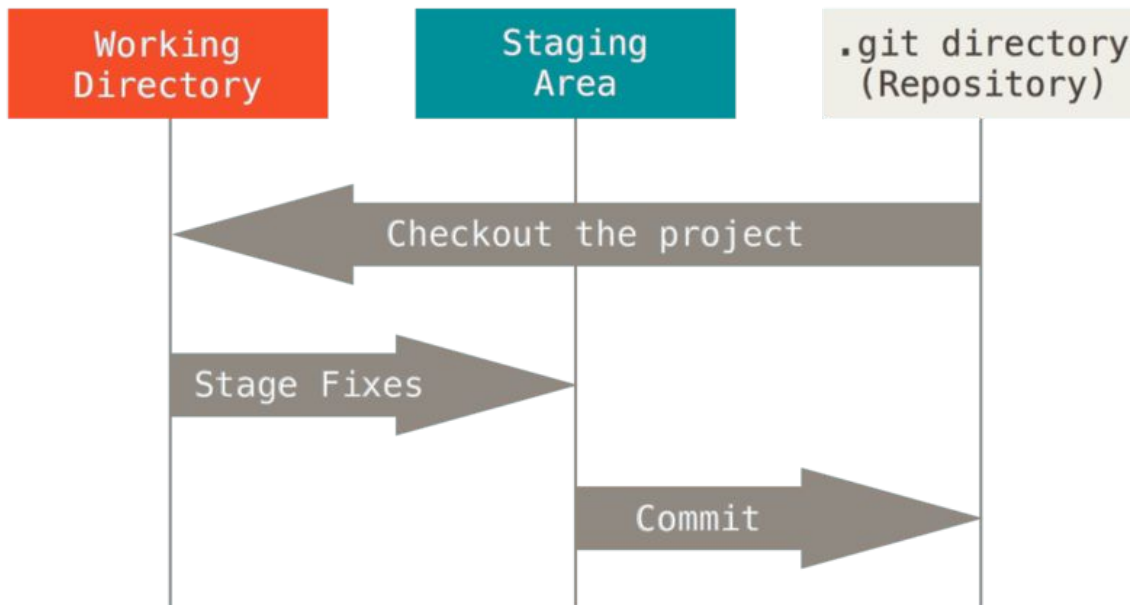


Git's advantages

- Nearly every operation is local
- Integrity
- Git generally only adds data

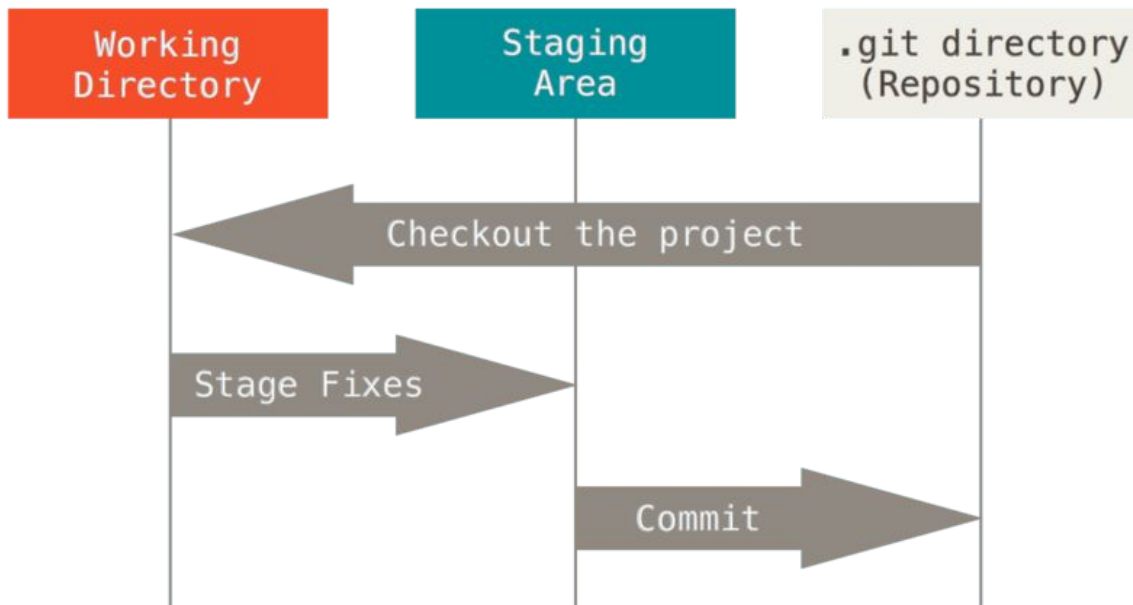
Working areas

- Working tree
- Staging area
- Git directory

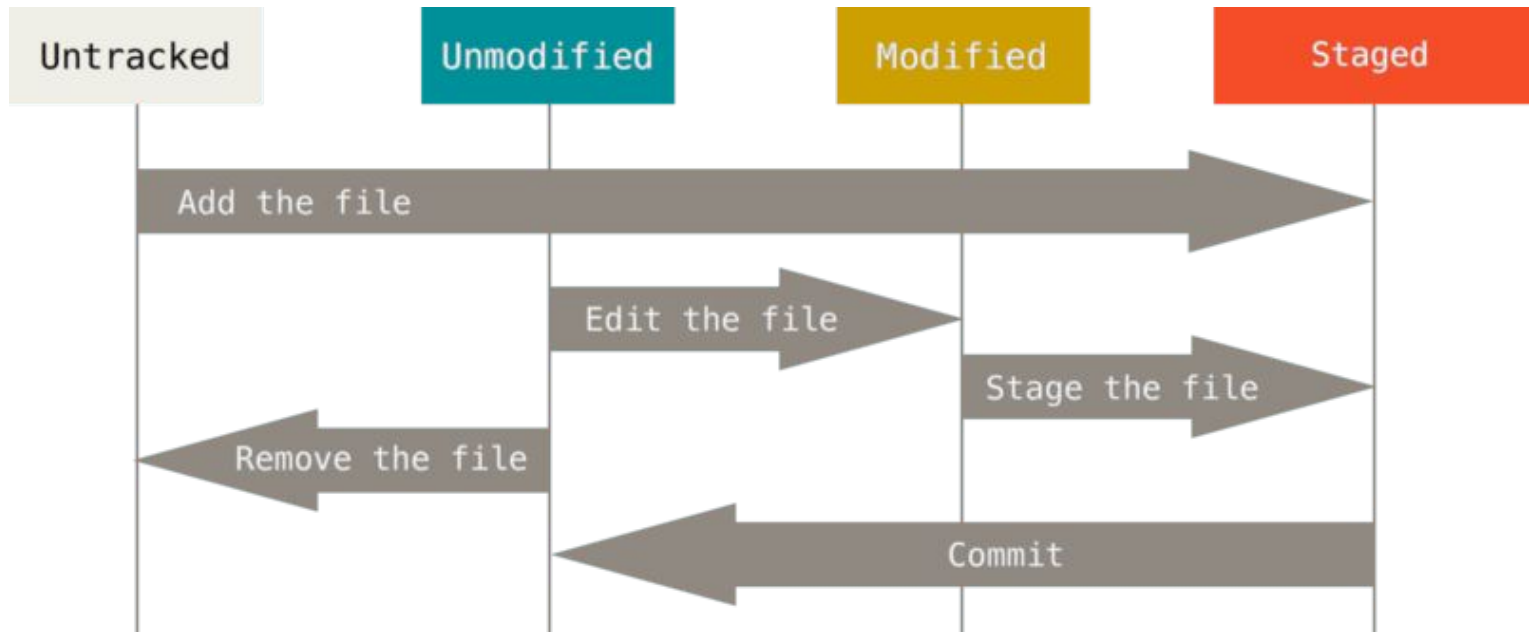


The four States

- Untracked
- Unmodified
- Modified
- Staged



Typical Workflow





Git in practice



Initialize a repository

Two ways:

- Initialize a repository in an existing directory
- Clone an existing repository



Initialize a repository from an existing directory:

```
$ git init
```



Initialize a repository from an existing directory:

```
$ git status
```



file1.txt



.git

Initialize a repository from an existing directory:

```
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        file1.txt

nothing added to commit but untracked files present (use "git add" to track)
```



file1.txt



.git

Initialize a repository from an existing directory:

```
$ git add file1.txt
```

```
$ git status
on branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   file1.txt
```



file1.txt



.git

Initialize a repository from an existing directory:

```
$ git add file1.txt
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   file1.txt

$ git commit -m 'first commit'
[master (root-commit) 43176a7] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file1.txt

$ git status
On branch master
nothing to commit, working tree clean
```



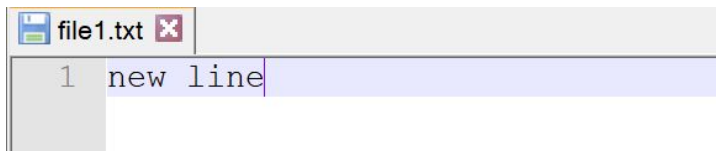
Cloning an existing repository:

```
$ git clone https://github.com/wikistat/Al-Frameworks.git
```



AI-Frameworks

Recording Changes to the Repository



```
$ git status
on branch master
changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   file1.txt

no changes added to commit (use "git add" and/or "git commit -a")
```



file1.txt



.git

Recording Changes to the Repository

```
$ git add file1.txt
```

```
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   file1.txt
```

```
$ git commit -m 'add new line'
[master 802715a] add new line
1 file changed, 1 insertion(+)
```

```
$ git commit -a -m 'add new line'
```



file1.txt



.git

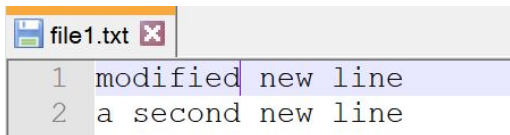
Recording Changes to the Repository

```
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

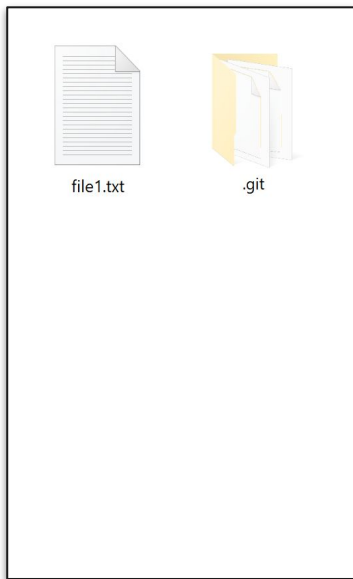
        modified:   file1.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

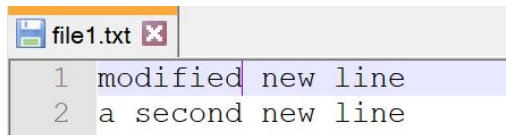
```
$ git diff
diff --git a/file1.txt b/file1.txt
index c6568ea..b9ca418 100644
--- a/file1.txt
+++ b/file1.txt
@@ -1,2 @@
-new line
\ No newline at end of file
+modified new line
+a second new line
\ No newline at end of file
```



```
file1.txt
1 modified new line
2 a second new line
```



Recording Changes to the Repository



```
1 modified new line
2 a second new line
```

```
$ git commit -a -m 'add second line'
[master 53c470a] add second line
1 file changed, 2 insertions(+), 1 deletion(-)
```



You forgot something?

```
file1.txt
1 modified new line
2 a second new line
3 a third new line
```

```
$ git add file1.txt
```

```
$ git commit --amend
```



Ignoring Files

```
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    Data/

nothing added to commit but untracked files present (use "git add" to track)
```

```
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

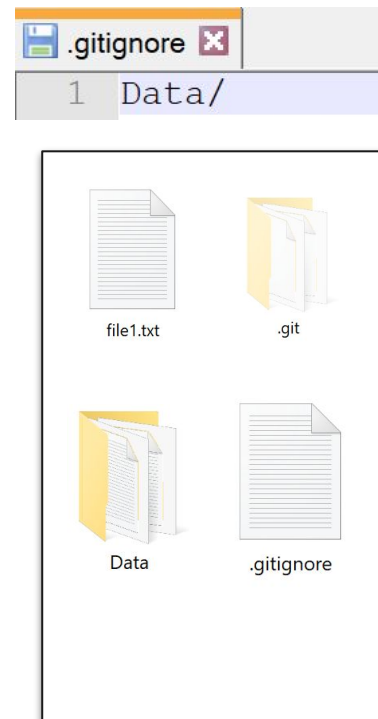
    .gitignore

nothing added to commit but untracked files present (use "git add" to track)

david.bertoin@B20181627 MINGW64 ~/workspace/Sandbox/INSA/git_intro (master)
$ git commit -a -m 'add gitignore'
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    .gitignore

nothing added to commit but untracked files present
```



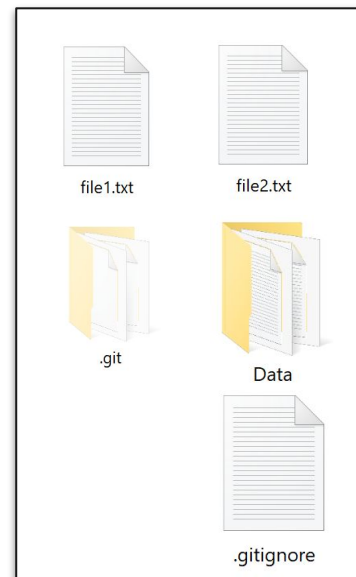
Delete Files

```
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
```

```
file2.txt
```

```
$ git add file2.txt
```

```
$ git commit -m 'add file2'
[master 1d6e521] add file2
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file2.txt
```



Delete Files

```
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)
```

```
    deleted:    file2.txt
```

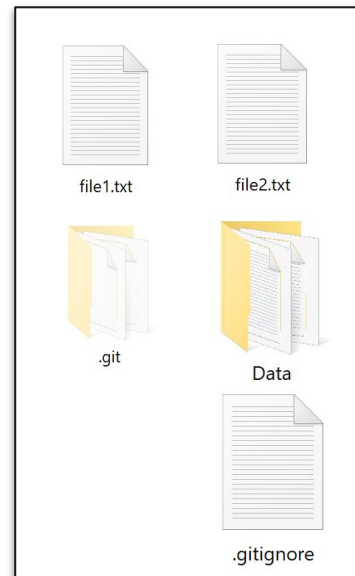
```
$ git rm file2.txt
rm 'file2.txt'
```

```
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)
```

```
    deleted:    file2.txt
```

```
$ git commit -m 'delete file 2'
[master 630bdc] delete file 2
1 file changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 file2.txt
```

```
$ git status
On branch master
nothing to commit, working tree clean
```



Delete Files

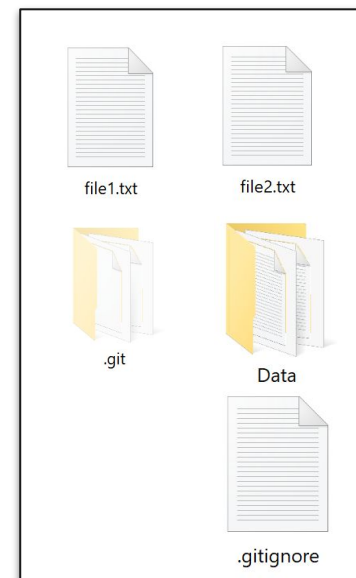
```
$ git rm file2.txt --cached
rm 'file2.txt'
```

```
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        deleted:    file2.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        file2.txt
```



Unstage Files

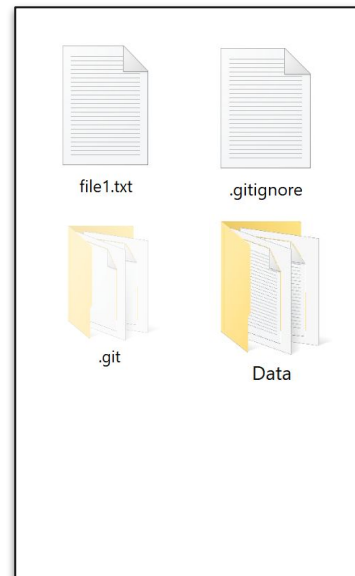
```
$ git add file1.txt
```

```
$ git status
on branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   file1.txt
```

file1.txt

```
1 modified new line
2 a second new line
3 a third new line
4 a fourth new line
```



Commit History

```
$ git log
commit 4e9240e61210332bf2ed7dc7416dac10ee532d19 (HEAD -> master)
Author: David <davideurofr1@gmail.com>
Date:   Wed Oct 20 14:54:37 2021 +0200

    delete file2

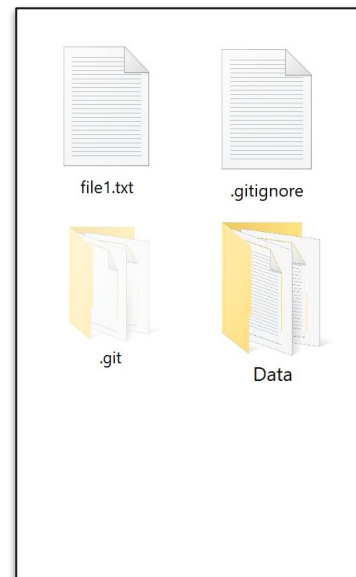
commit 8490684539ab79ac8d5a7e7017bd832d06d13381
Author: David <davideurofr1@gmail.com>
Date:   Wed Oct 20 14:16:48 2021 +0200

    add file2

commit 630bdcbase8363b90603acbc5ba208356fc084d6
Author: David <davideurofr1@gmail.com>
Date:   Wed Oct 20 14:12:55 2021 +0200

    delete file 2

commit 1d6e5219628c3babc3a3add647002d1131036306
Author: David <davideurofr1@gmail.com>
```



Commit History

```
$ git log -p -2
commit 4e9240e61210332bf2ed7dc7416dac10ee532d19 (HEAD -> master)
Author: David <davideurofr1@gmail.com>
Date:   Wed Oct 20 14:54:37 2021 +0200

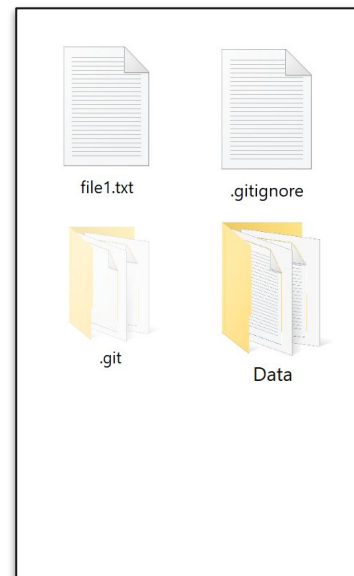
    delete file2

diff --git a/file2.txt b/file2.txt
deleted file mode 100644
index e69de29..0000000

commit 8490684539ab79ac8d5a7e7017bd832d06d13381
Author: David <davideurofr1@gmail.com>
Date:   Wed Oct 20 14:16:48 2021 +0200

    add file2

diff --git a/file2.txt b/file2.txt
new file mode 100644
index 0000000..e69de29
```



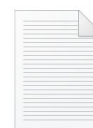
Commit History

```
$ git log --pretty=oneline
4e9240e61210332bf2ed7dc7416dac10ee532d19 (HEAD -> master) delete file2
8490684539ab79ac8d5a7e7017bd832d06d13381 add file2
630bdcbae8363b90603acbc5ba208356fc084d6 delete file 2
1d6e5219628c3bab3a3add647002d1131036306 add file2
30d73ae295a60b31147506141253bbbed29a88720 add gitignore
53c470a9c90ca4b9a2a858ac5c19608f8b0a9064 add second line
802715a9ca986627a220bcea6a98ed4ca7e25499 add new line
43176a7ebd79fa5aeb12e1f987a7766eacb64e1 first commit
```

```
$ git log --pretty=format:"%h - %an, %ar : %s"
4e9240e - David, 23 minutes ago : delete file2
8490684 - David, 61 minutes ago : add file2
630bdcbae - David, 65 minutes ago : delete file 2
1d6e521 - David, 71 minutes ago : add file2
30d73ae - David, 78 minutes ago : add gitignore
53c470a - David, 28 hours ago : add second line
802715a - David, 29 hours ago : add new line
43176a7 - David, 29 hours ago : first commit
```



file1.txt



.gitignore



.git

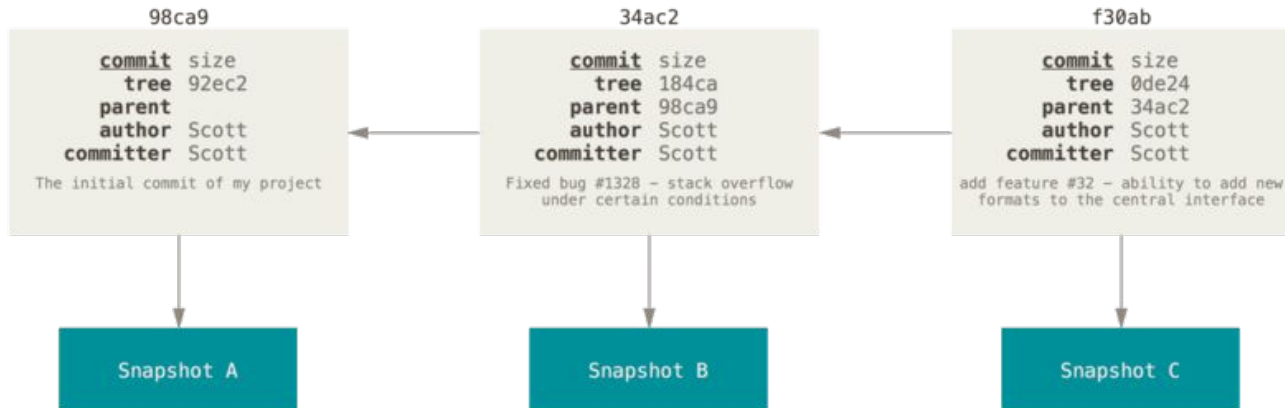


Data

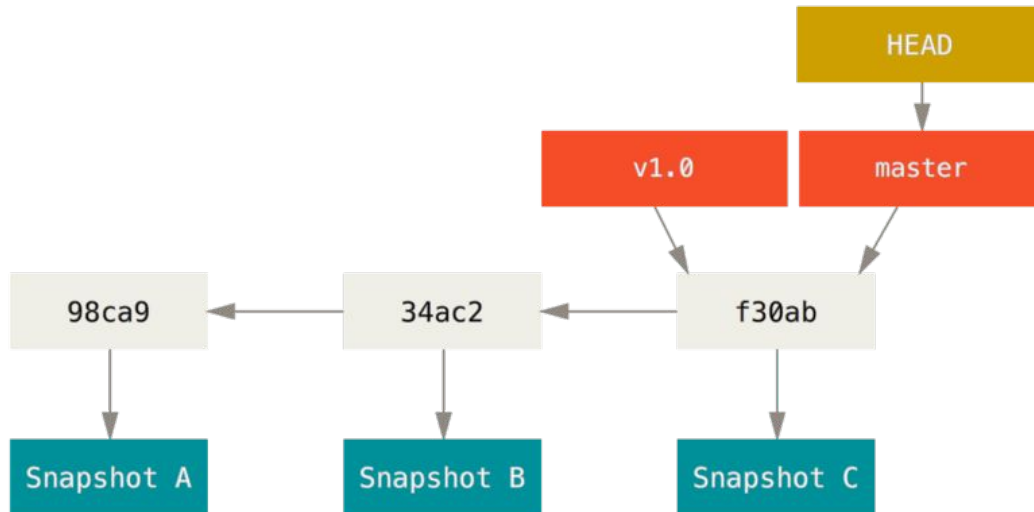


Branching

Git branching

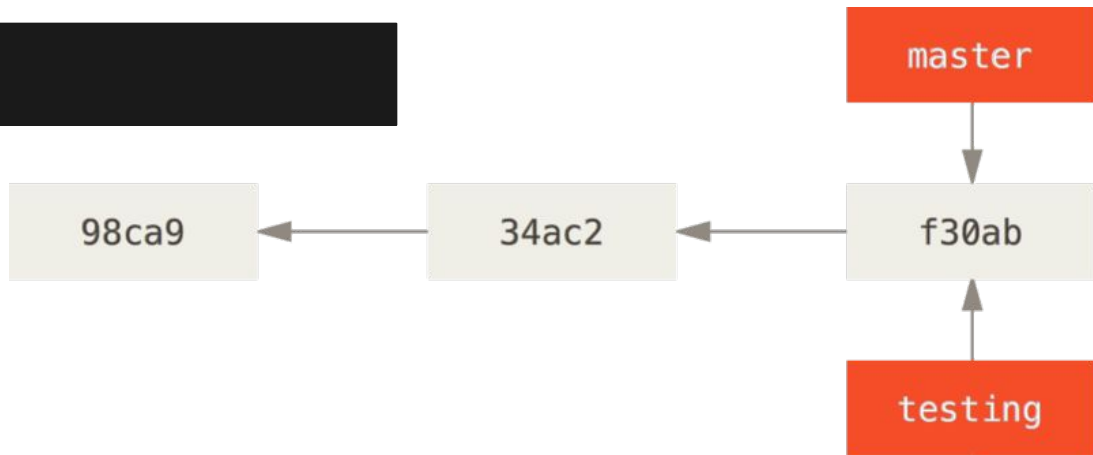


Git branching



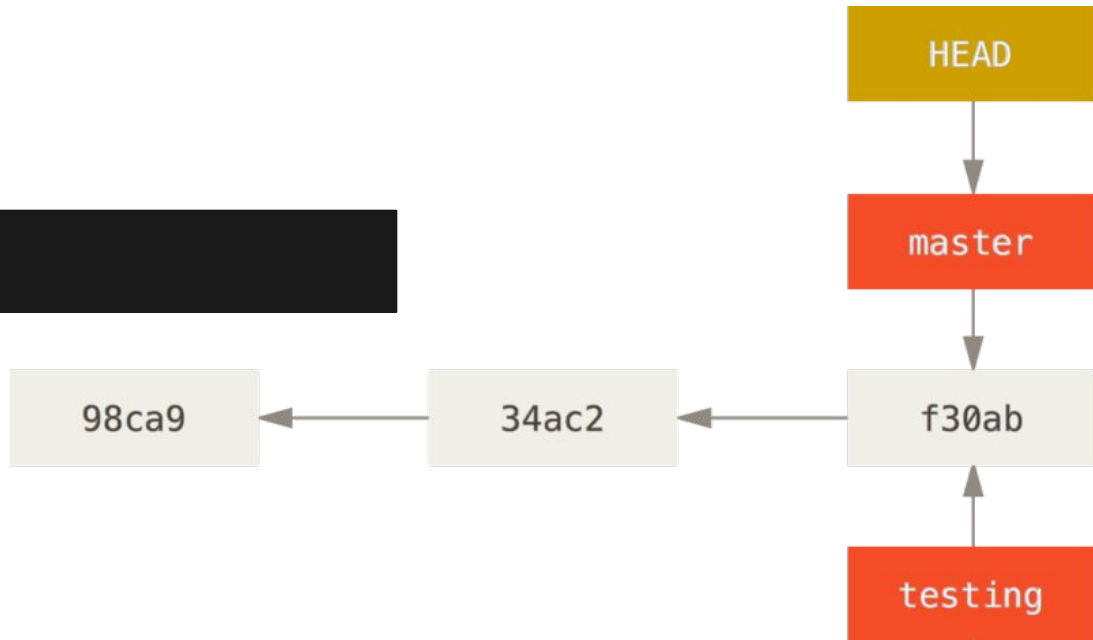
Git branching

```
$ git branch testing
```



Git branching

```
$ git branch testing
```



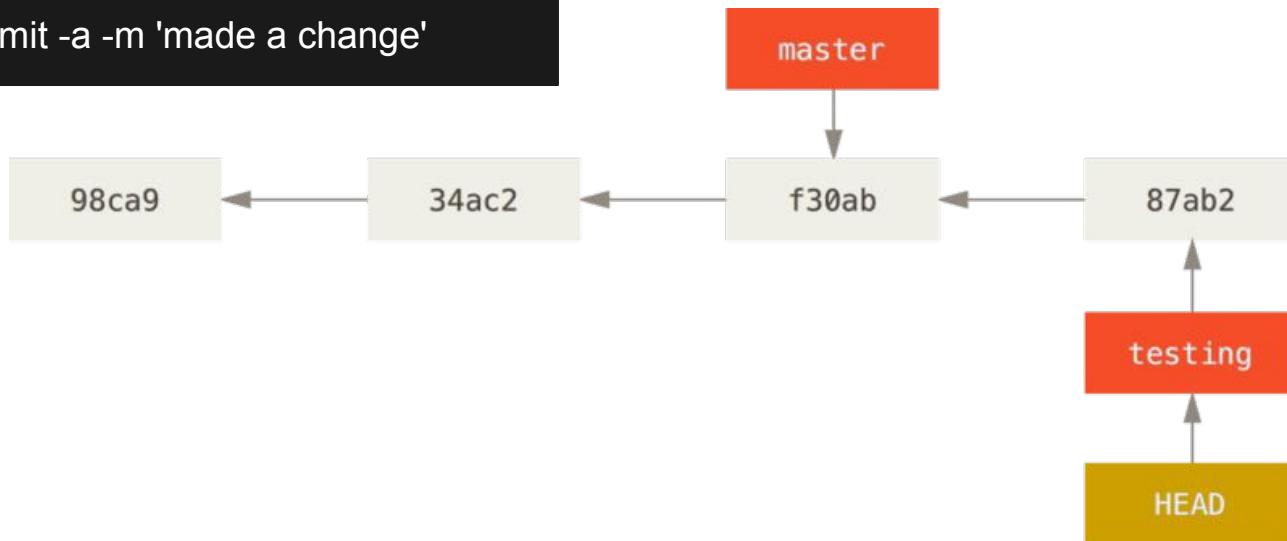
Git branching

```
$ git checkout testing
```

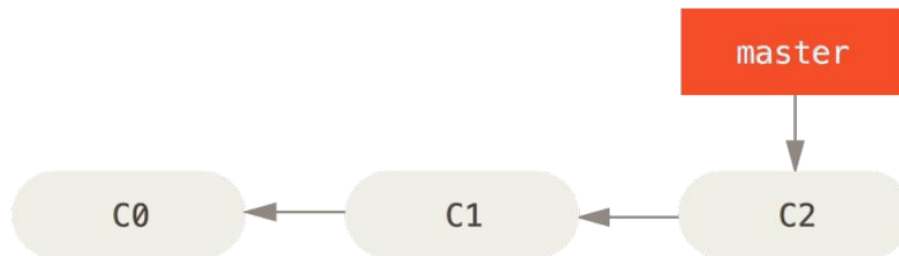


Git branching

```
$ git commit -a -m 'made a change'
```

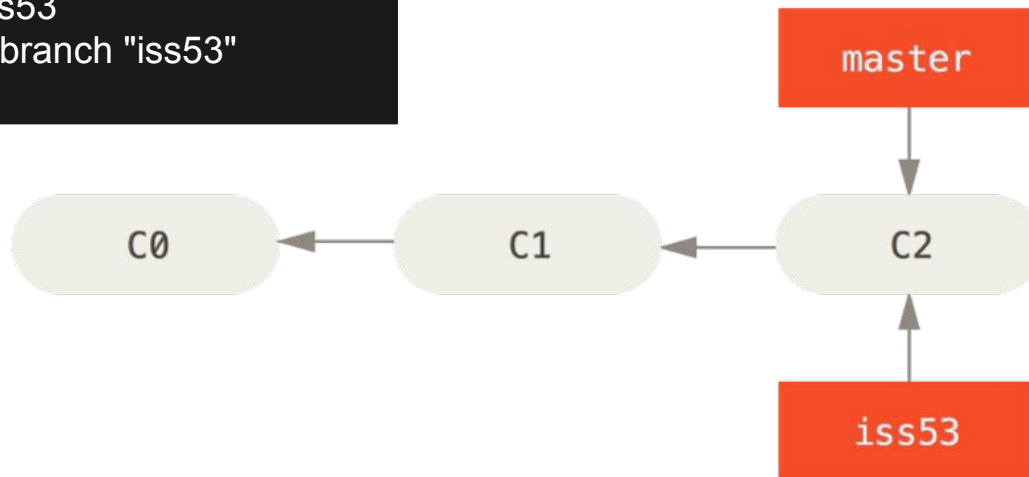


Merging branches



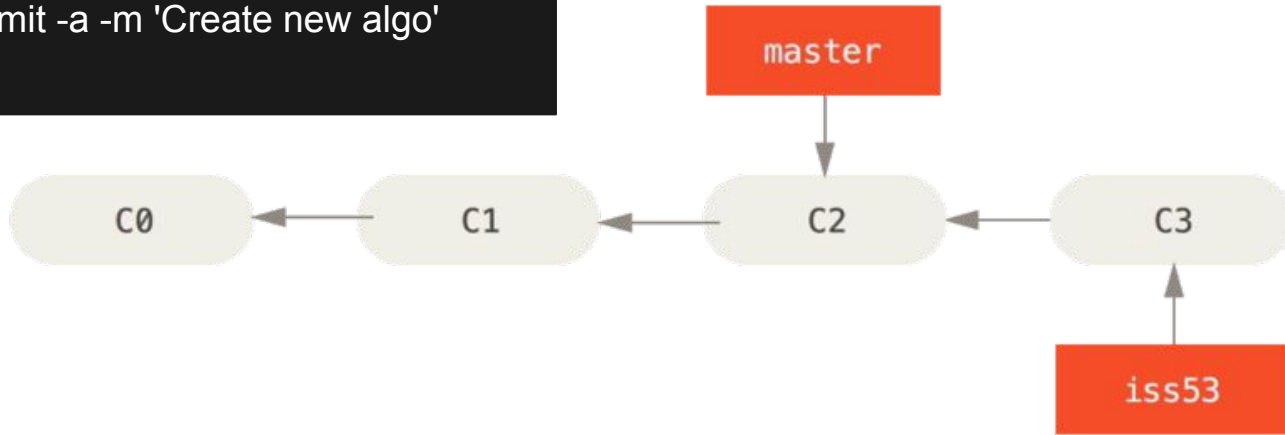
Merging branches

```
$ git checkout -b iss53
Switched to a new branch "iss53"
```



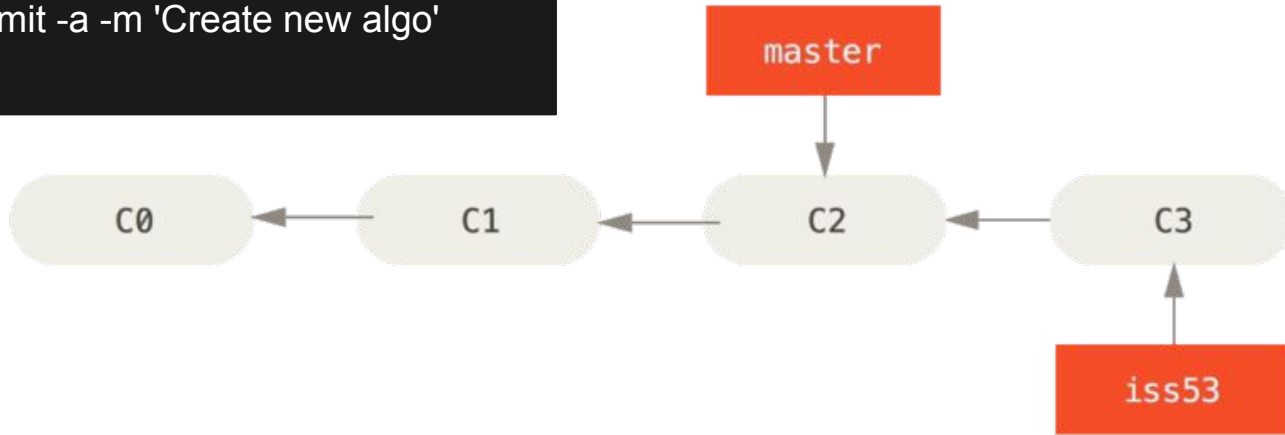
Merging branches

```
$ git commit -a -m 'Create new algo'
```



Merging branches

```
$ git commit -a -m 'Create new algo'
```

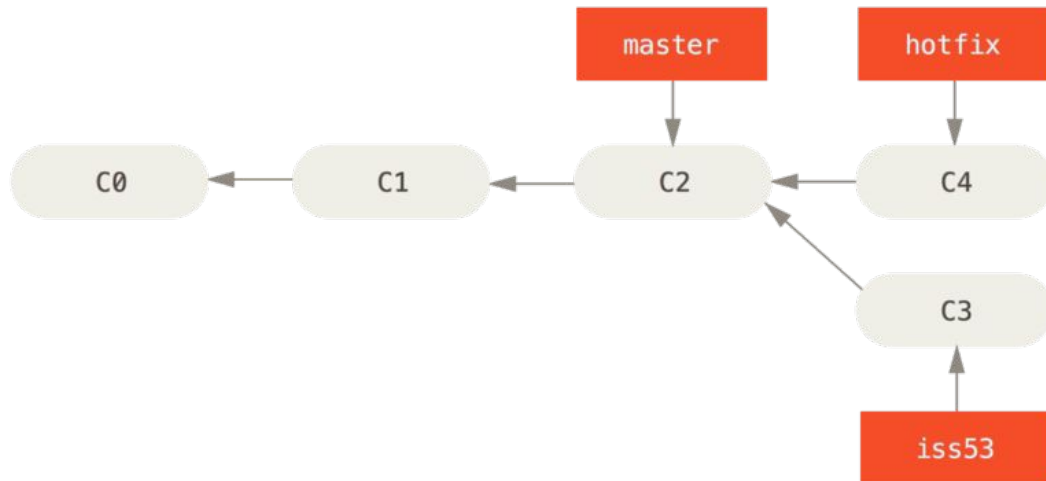


Merging branches

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

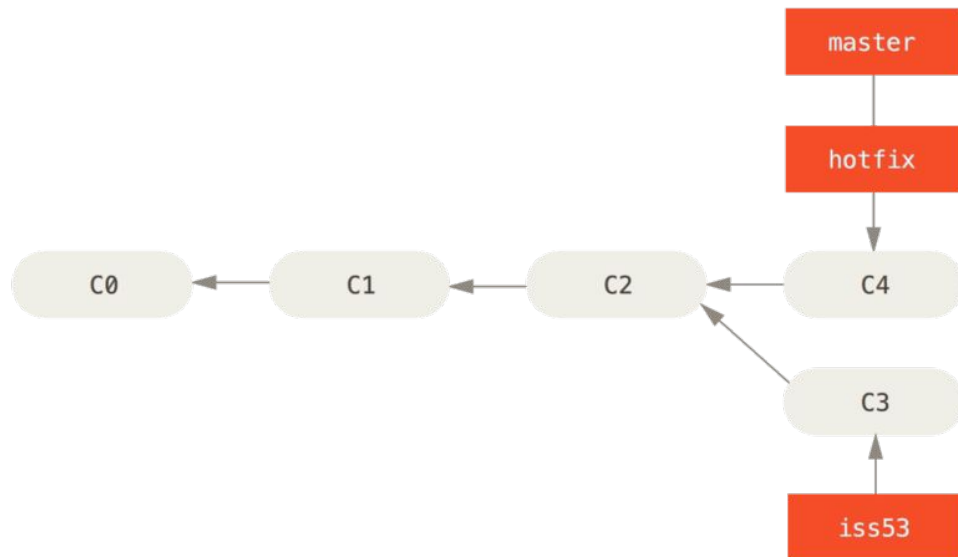
```
$ git checkout -b hotfix
Switched to a new branch 'hotfix'
```

```
$ git commit -a -m 'Fix bad learning rate'
[hotfix 1fb7853] Fix bad learning rate
1 file changed, 2 insertions(+)
```



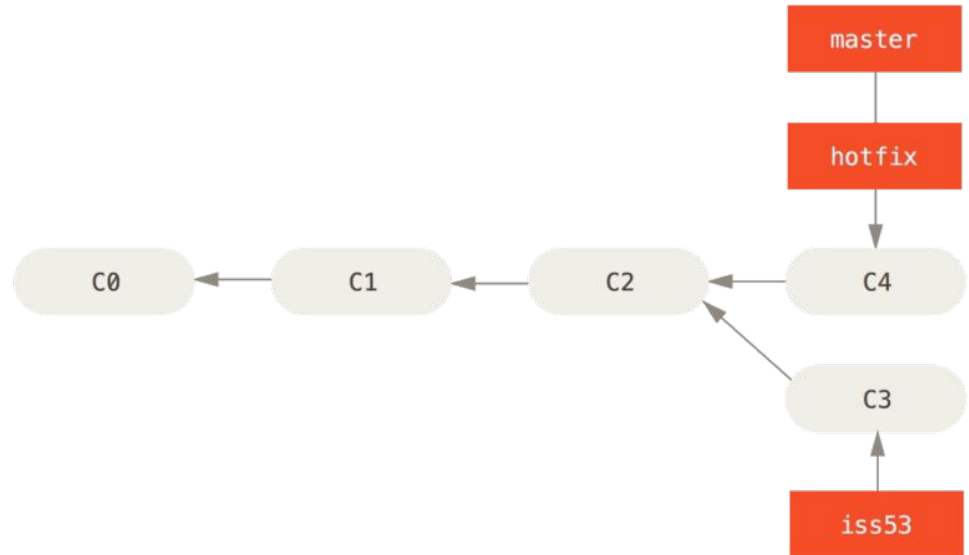
Merging branches

```
$ git checkout master
$ git merge hotfix
Updating f42c576..3a0874c
Fast-forward
 index.html | 2 ++
 1 file changed, 2 insertions(+)
```



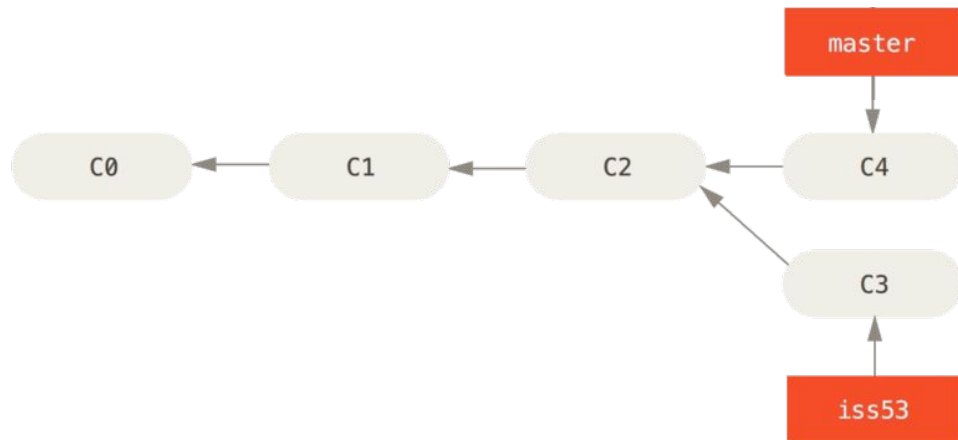
Merging branches

```
$ git checkout master
$ git merge hotfix
Updating f42c576..3a0874c
Fast-forward
 index.html | 2 ++
 1 file changed, 2 insertions(+)
```

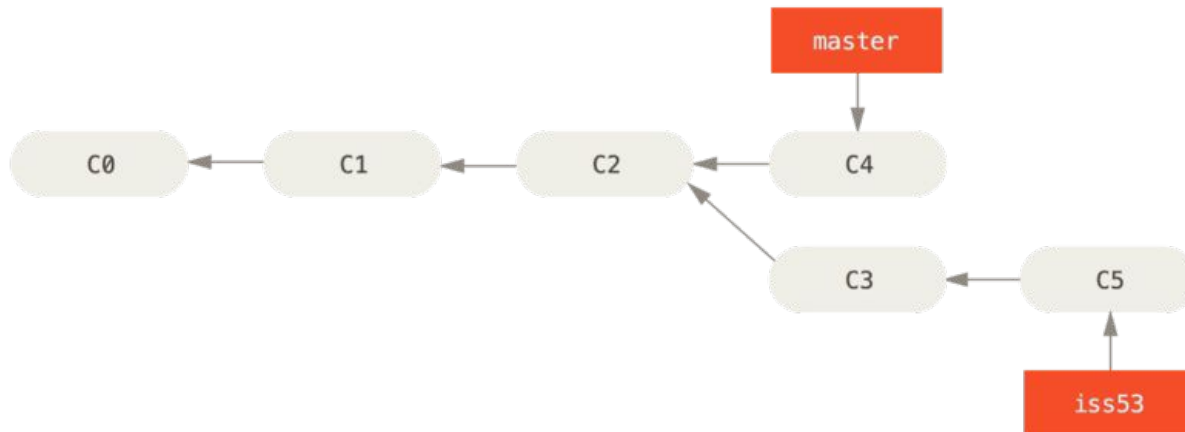


Merging branches

```
$ git branch -d hotfix
Deleted branch hotfix (3a0874c).
```

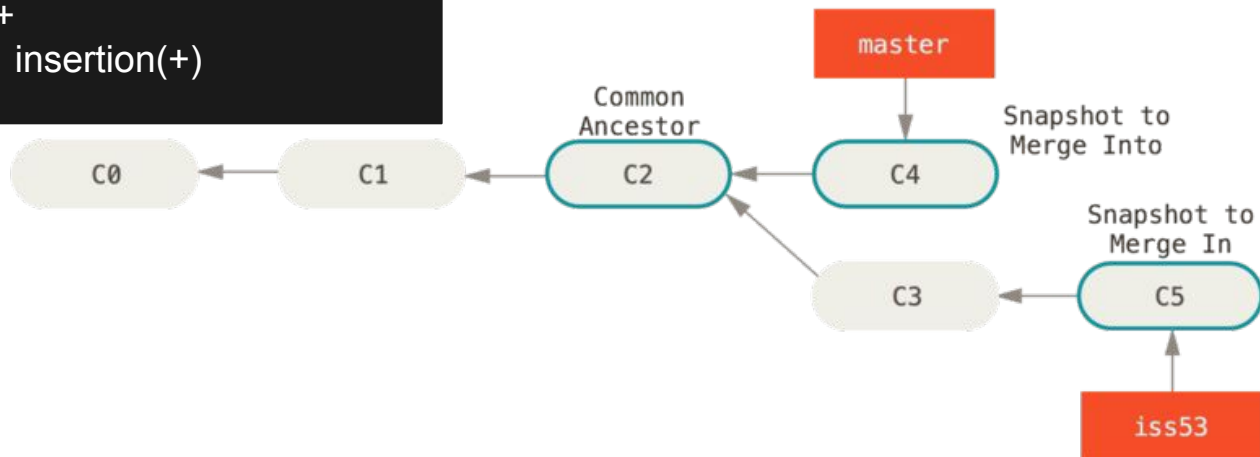


Merging branches



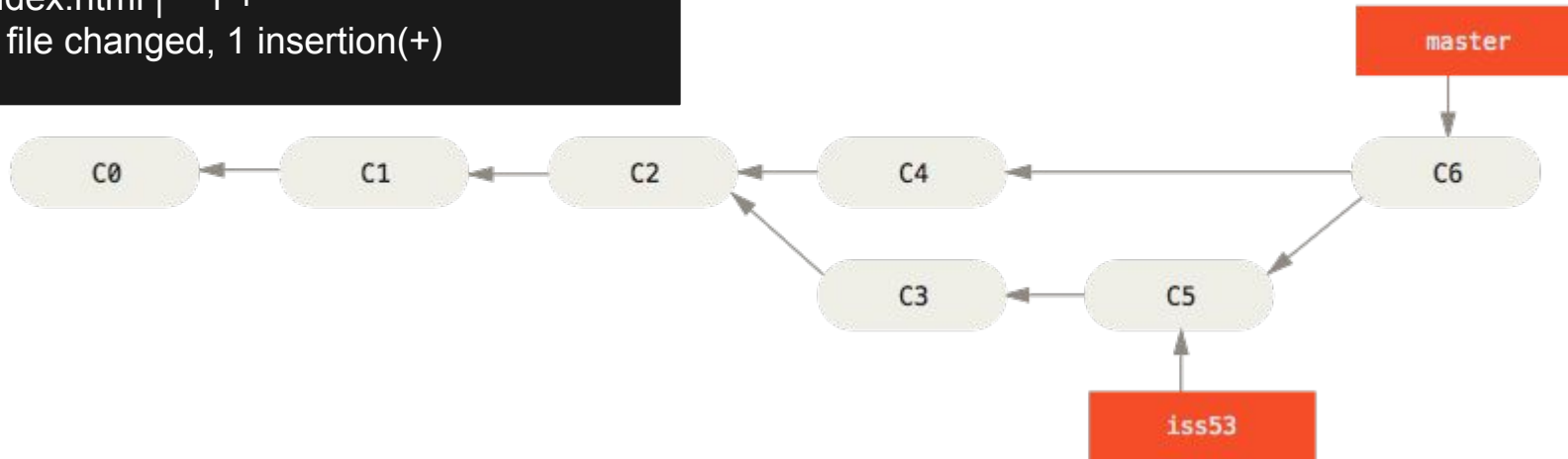
Merging branches

```
$ git checkout master
Switched to branch 'master'
$ git merge iss53
Merge made by the 'recursive' strategy.
index.html | 1 +
1 file changed, 1 insertion(+)
```



Merging branches

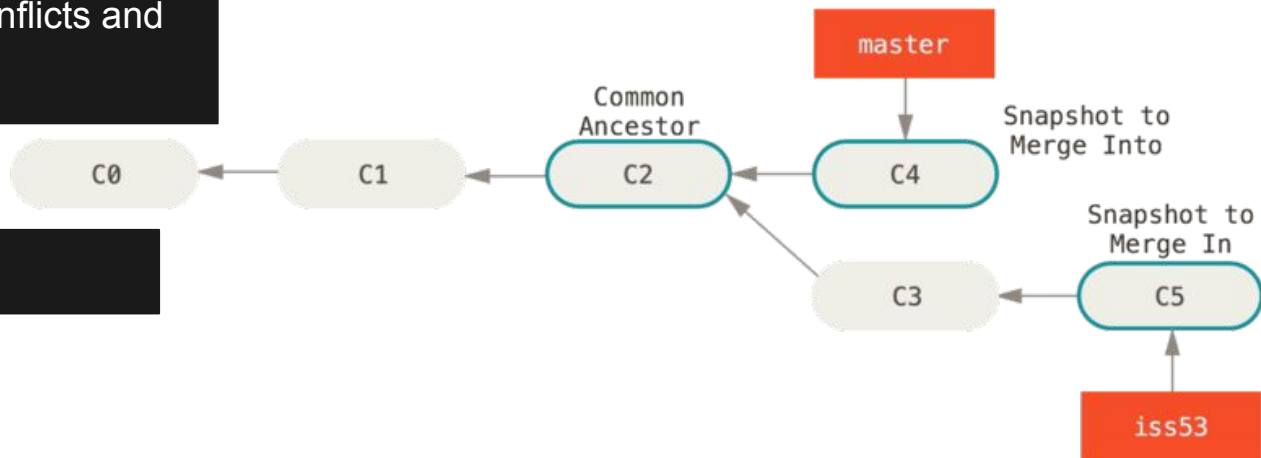
```
$ git checkout master
Switched to branch 'master'
$ git merge iss53
Merge made by the 'recursive' strategy.
index.html | 1 +
1 file changed, 1 insertion(+)
```



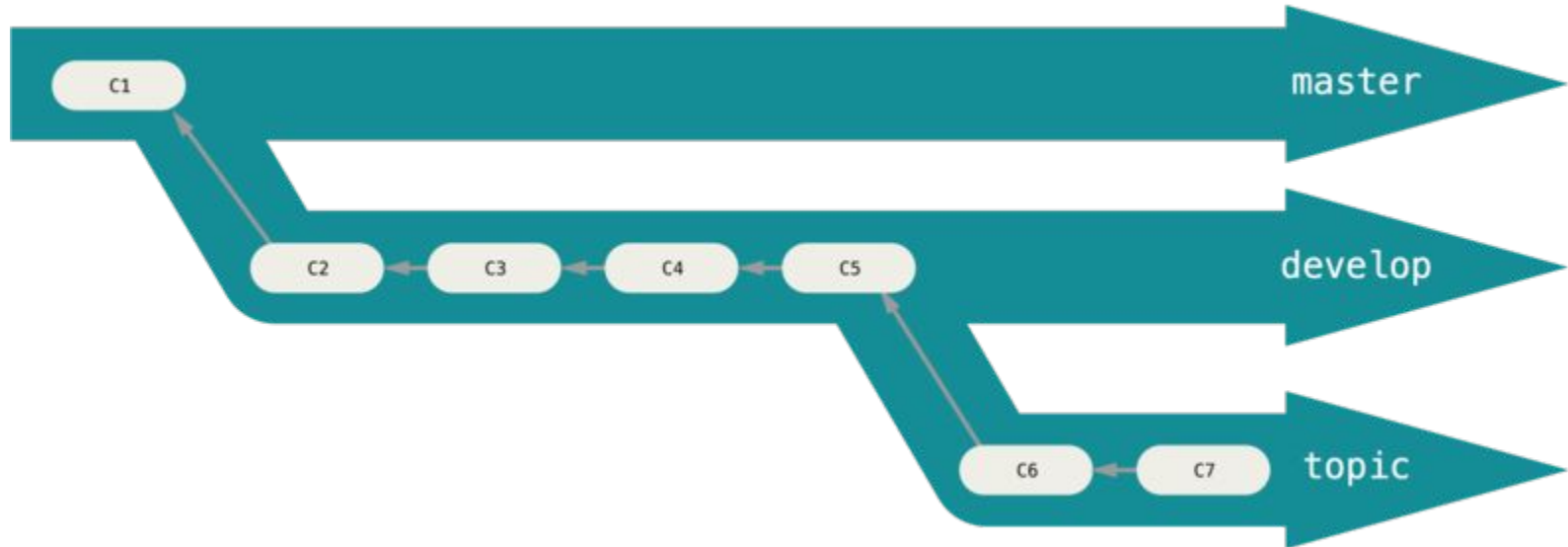
Merging branches

```
$ git merge iss53
Auto-merging index.html
CONFLICT (content): Merge conflict in
index.html
Automatic merge failed; fix conflicts and
then commit the result.
```

```
$ git mergetool
```



Merging branches

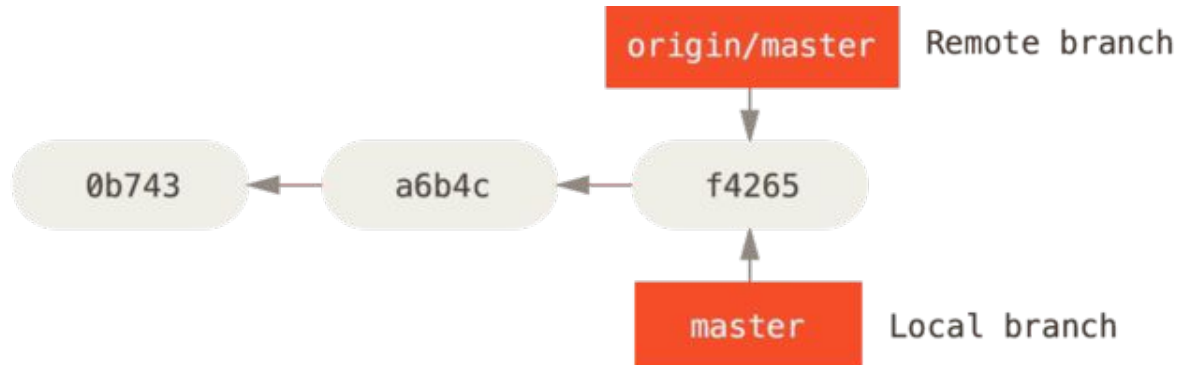




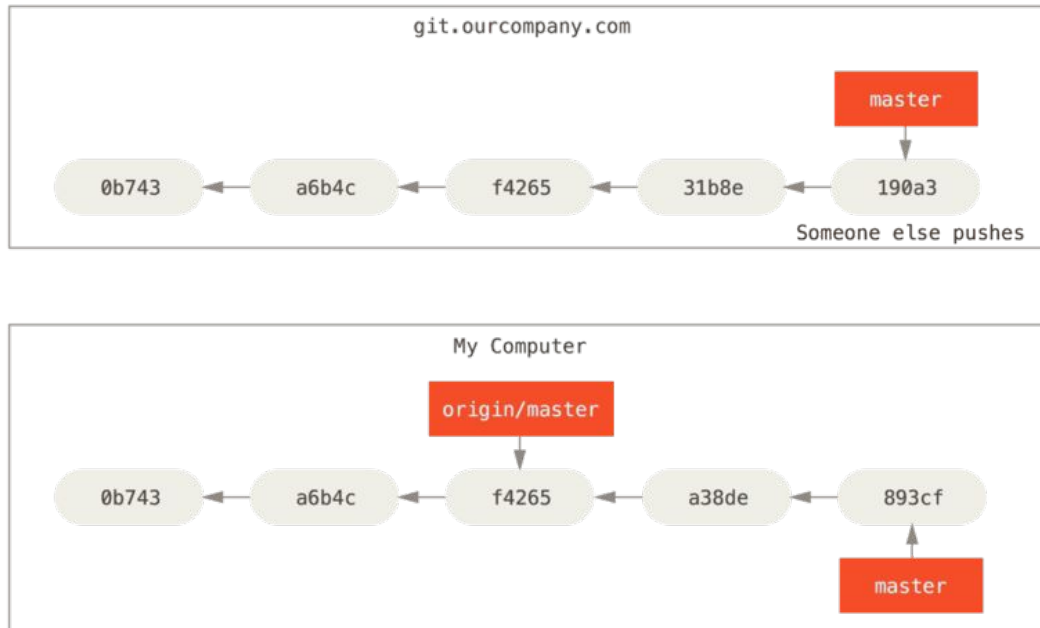
Remote Branching

Remote Branches

- branches in the remote repository
- `<remote>/<branch>`



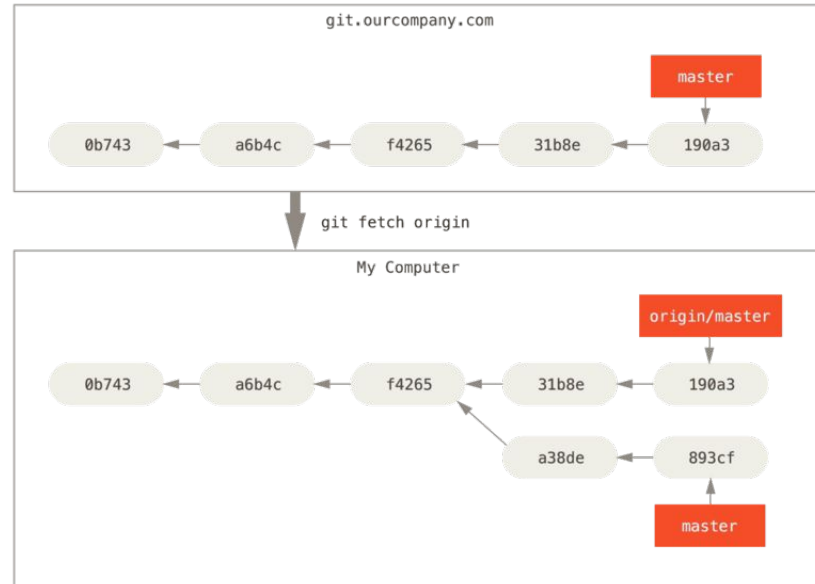
Remote Branches



Remote Branches

```
$ git fetch
```

```
$ git pull
```





Pushing

```
$ git push origin mybranch
Counting objects: 24, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (15/15), done.
Writing objects: 100% (24/24), 1.91 KiB | 0 bytes/s, done.
Total 24 (delta 2), reused 0 (delta 0)
To https://github.com/wikistat/Al-Frameworks
* [new branch]    mybranch -> mybranch
```



Git Cheat Sheet:

<https://training.github.com/downloads/github-git-cheat-sheet.pdf>

See you in class!