How to use Github like Silicon Valley Developers

By Altotech

What is Git?

- Keeps track of changes
  - especially text changes
  - version1, version2, version3
- Version control system (VCS)
- Source code management (SCM)

Who should use Git?

- Anyone wanting to track edits
  - Review a history log of changes made
  - View differences between versions
  - Retrieve old versions
- Anyone needing to share changes with collaborators

Try Github!

Installing Git

- Download http://git-scm.com/download

- Install

Initialize repository

- Open Terminal
- Go inside the root project folder
- Type git init

Commit

1. Make changes

   - new file, edit file

2. Add the changes

   Type `git add .` (for all changed file)

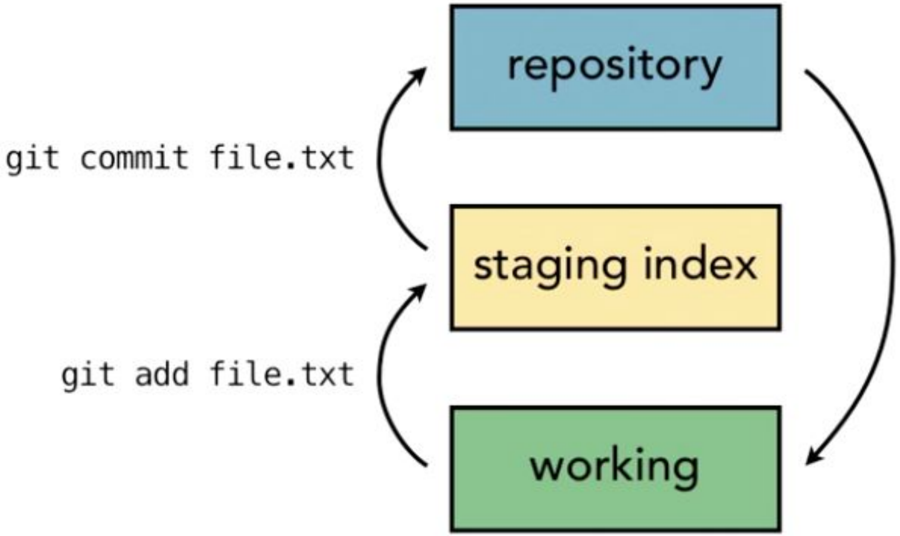
   `git add <filename>` (for any file)

3. Commit changes to the repository with a message

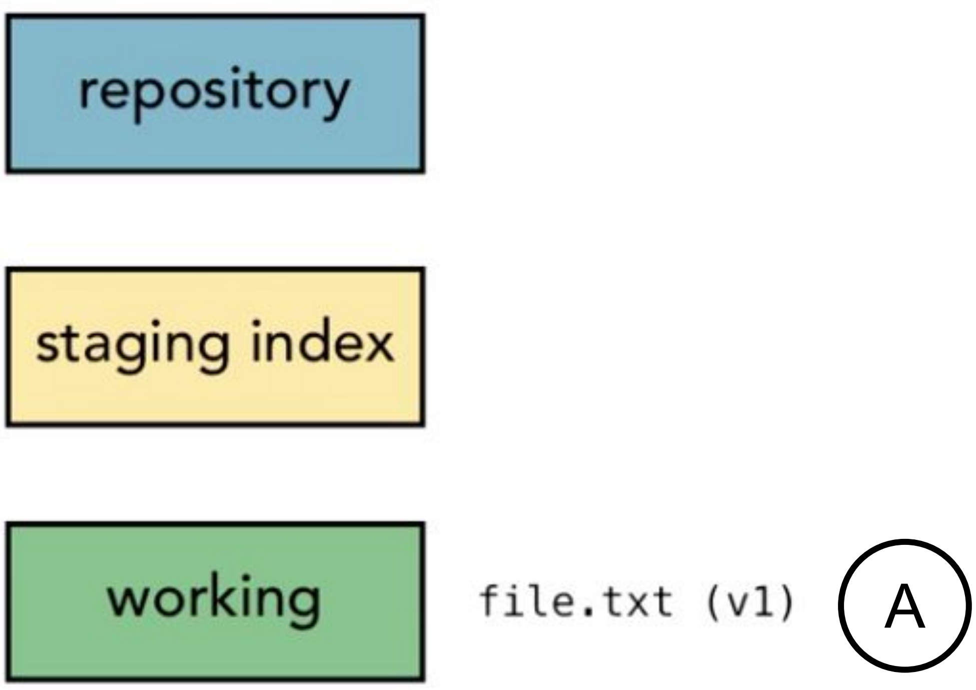
   Type `git commit -m “any descriptive message”`

Git concepts and architecture

Three-trees architecture



Git workflow



repository

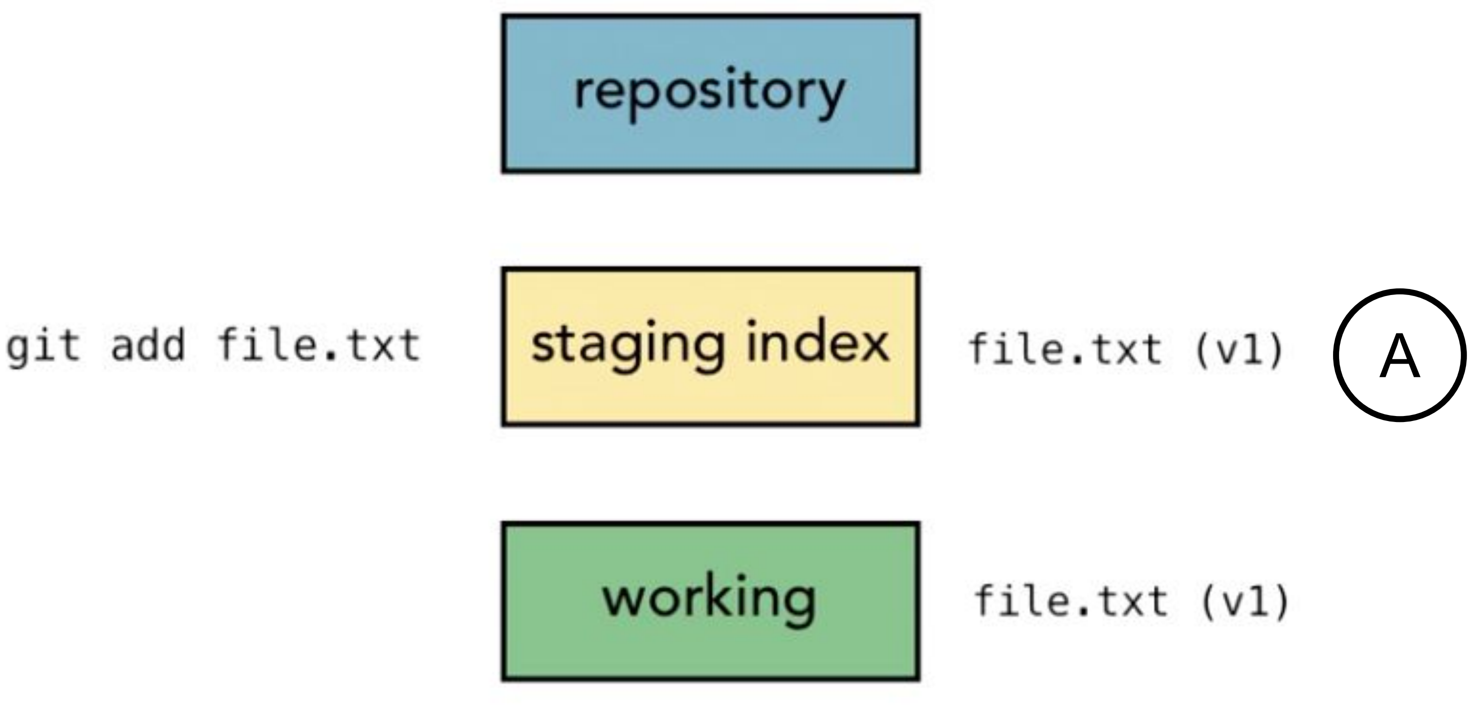
staging index

working

file.txt (v1)

A

Git workflow



repository

```
git add file.txt
```

staging index

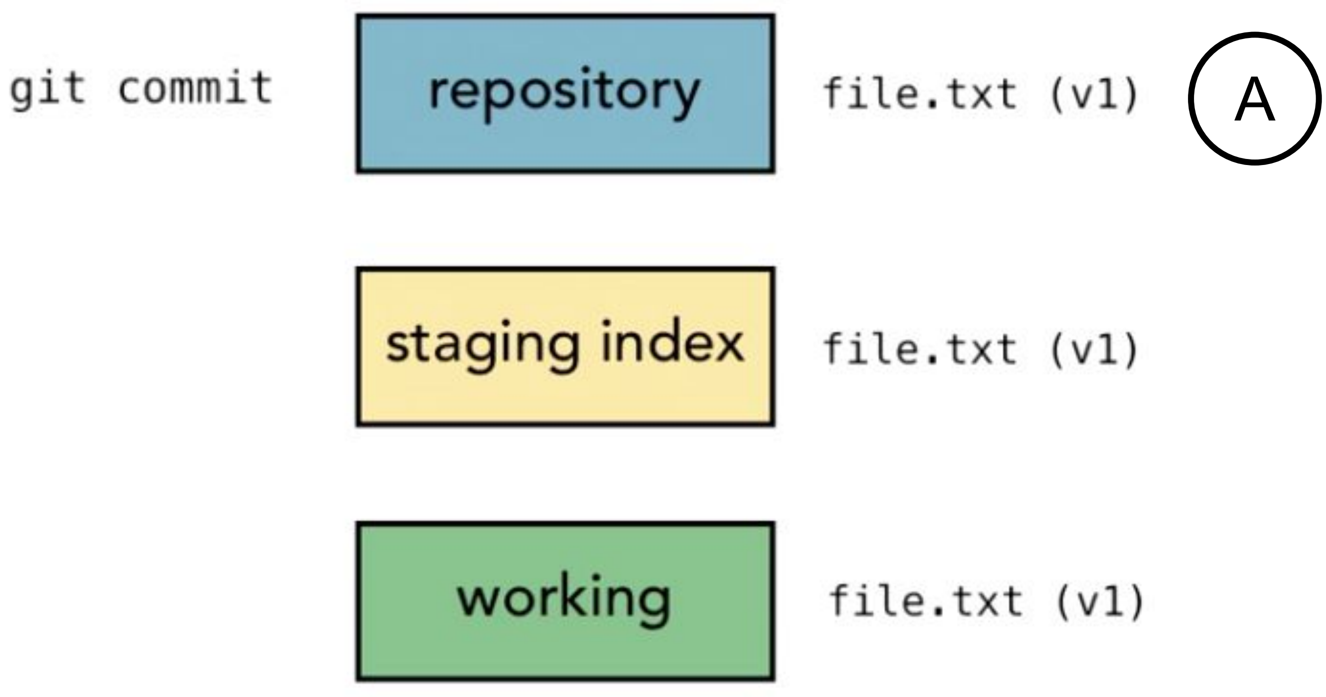
file.txt (v1)

A

working

file.txt (v1)

Git workflow



git commit

repository

file.txt (v1)

A

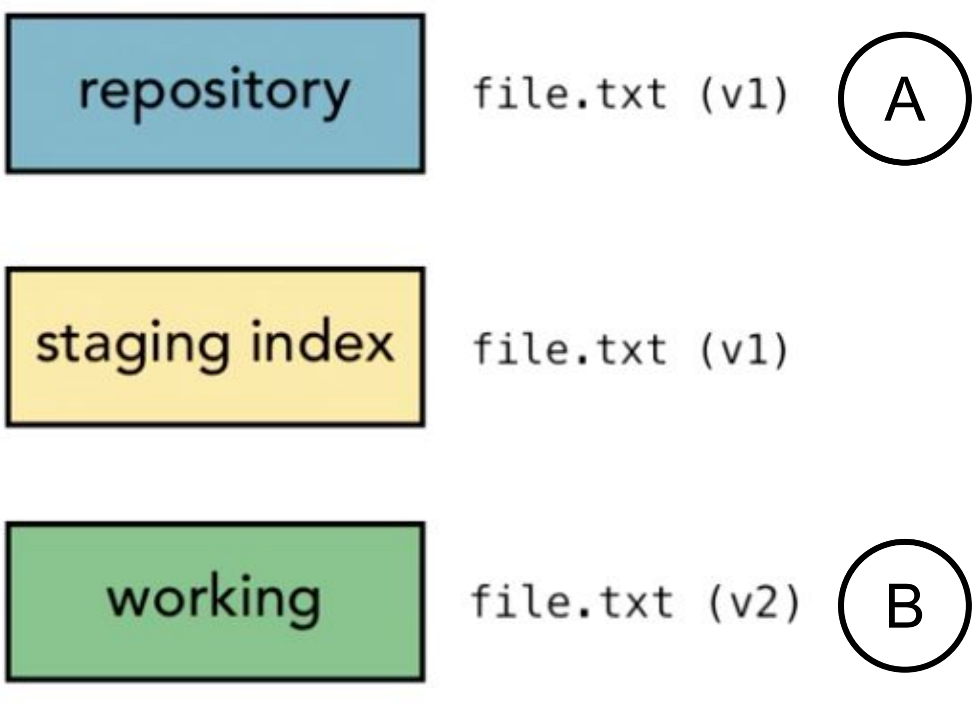
staging index

file.txt (v1)

working

file.txt (v1)

Git workflow

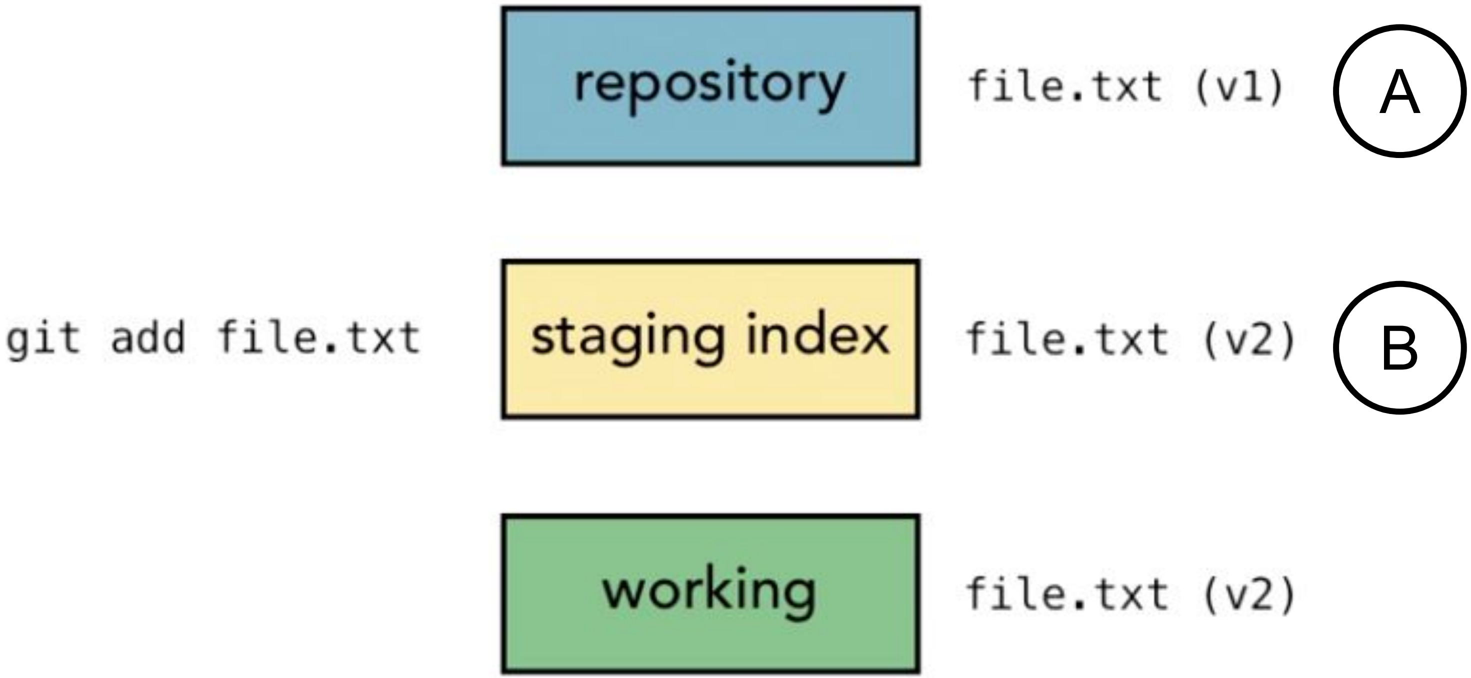


repository file.txt (v1) A

staging index file.txt (v1)

working file.txt (v2) B

Git workflow



repository

file.txt (v1)

A

git add file.txt

staging index

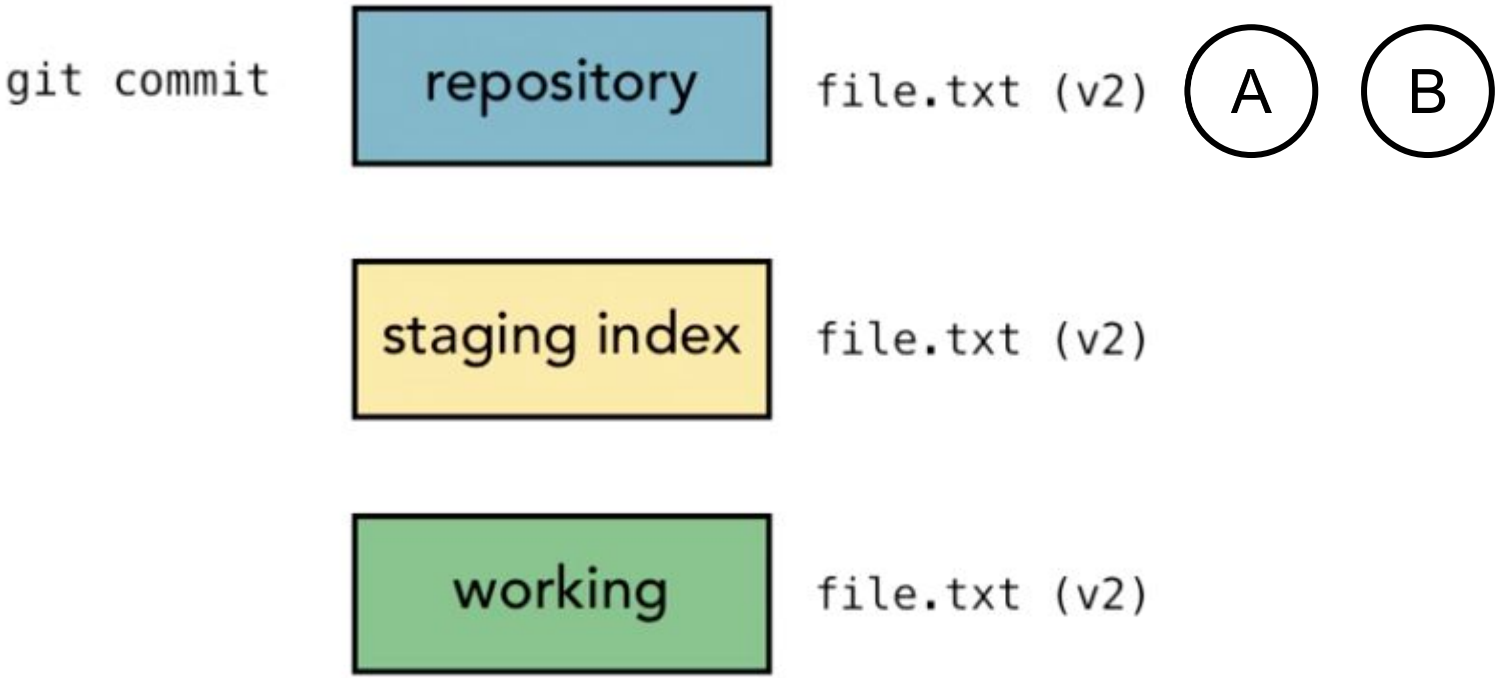
file.txt (v2)

B

working

file.txt (v2)

Git workflow



git commit

repository

file.txt (v2)

A

B

staging index

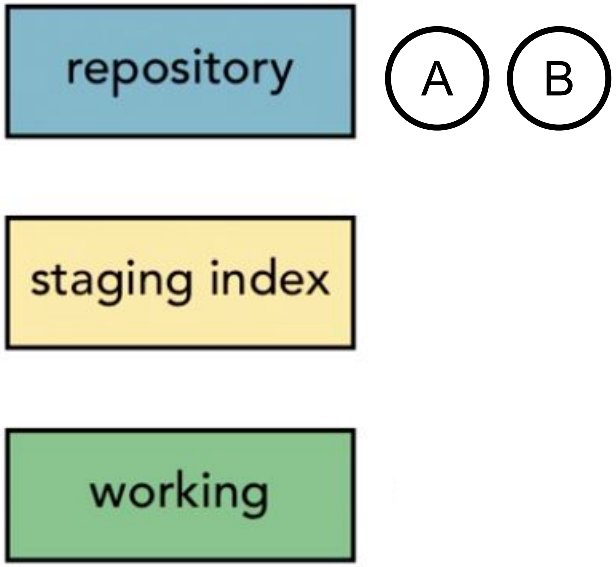
file.txt (v2)

working

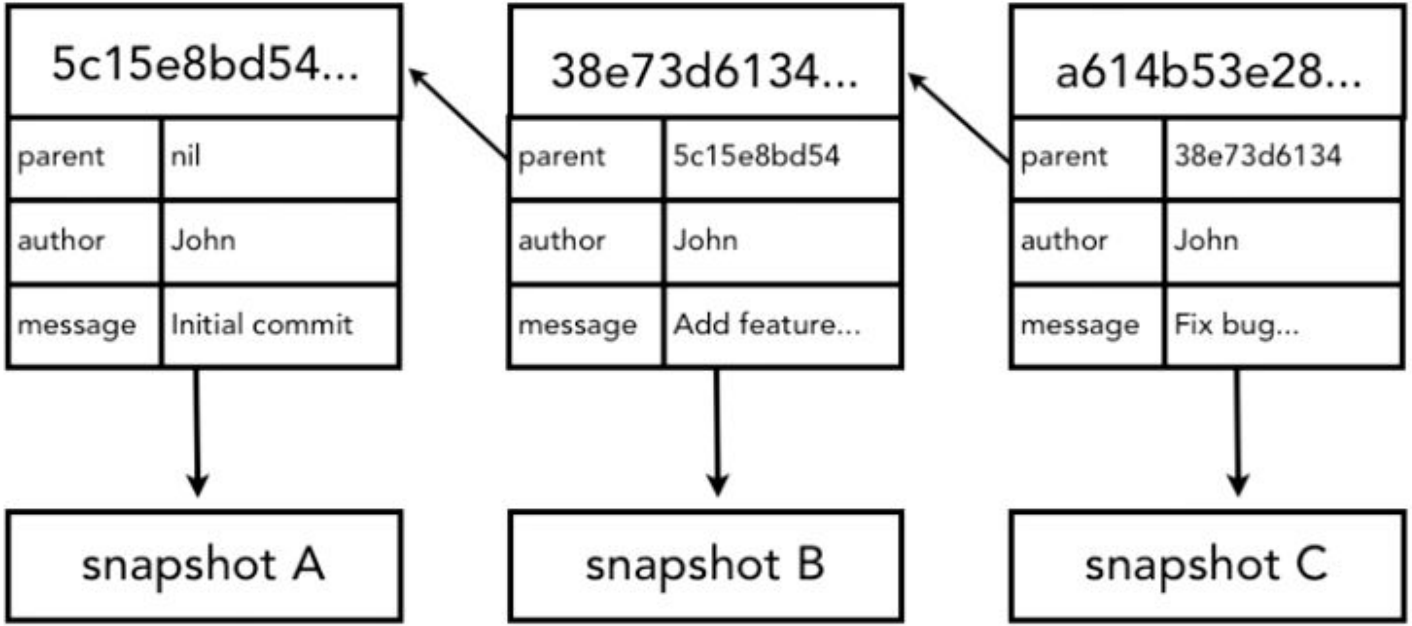
file.txt (v2)

Hash values

- Git generates a checksum for each change set
  - checksum algorithms convert data into a simple number
  - same data always equals same checksum
- Data integrity is fundamental
  - changing data would change checksum
- Git uses SHA-1 hash algorithm to create checksums
  - 40-character hexadecimal string (0-9, a-f)



Referring to commits



5c15e8bd54...	
parent	nil
author	John
message	Initial commit

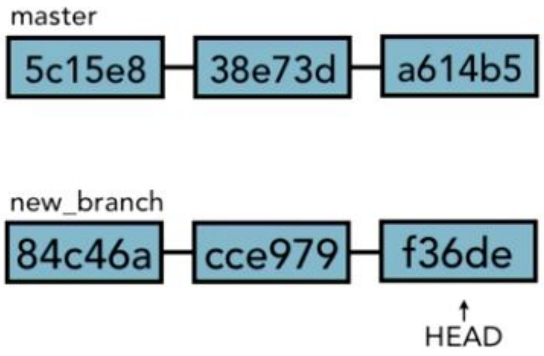
38e73d6134...	
parent	5c15e8bd54
author	John
message	Add feature...

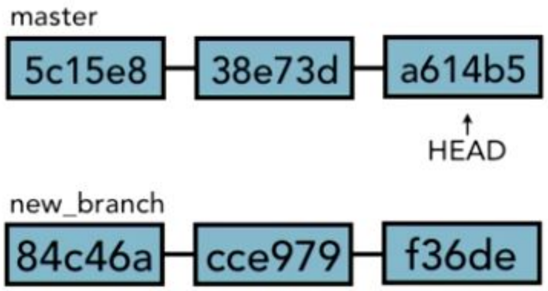
a614b53e28...	
parent	38e73d6134
author	John
message	Fix bug...

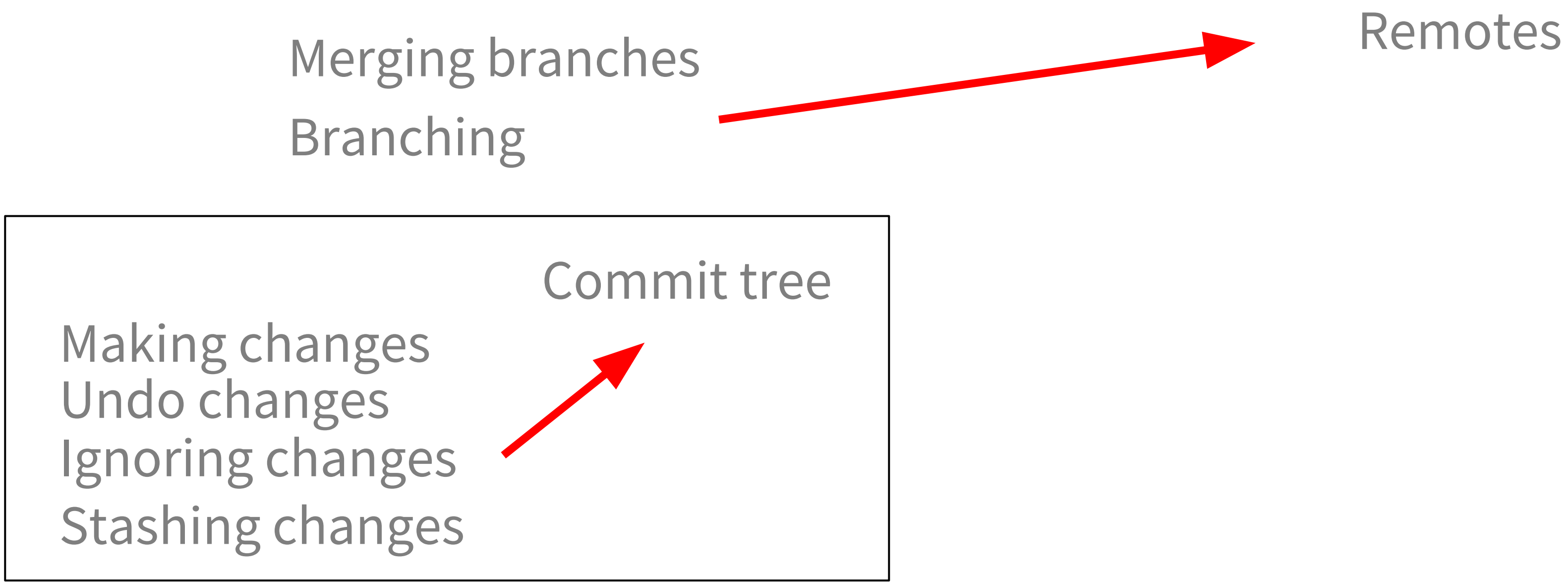
HEAD pointer

- Pointer to “tip” of current branch in repository
- Last state of repository, what was last checked out
- Points to parent of next commit

akes







Merging branches

Branching

Remotes

Commit tree

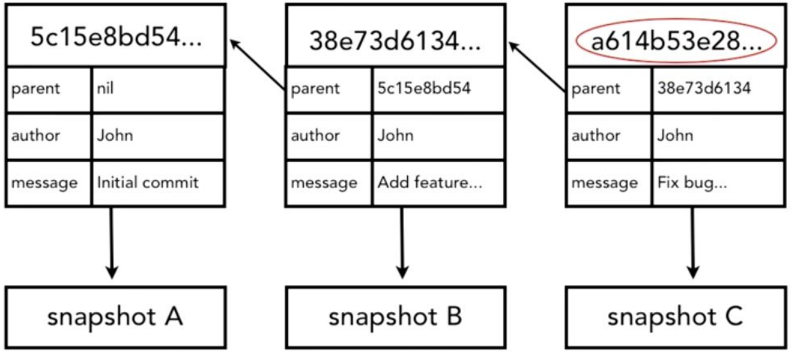
Making changes
Undo changes
Ignoring changes
Stashing changes

Undoing changes

git checkout -- <filename>	(for undoing any file)
git reset HEAD <filename>	(for unstaginging any file)

Amending commits

```
git commit --amend -m “Message”
```



5c15e8bd54...	
parent	nil
author	John
message	Initial commit

38e73d6134...	
parent	5c15e8bd54
author	John
message	Add feature...

a614b53e28...	
parent	38e73d6134
author	John
message	Fix bug...

Retrieving old versions

```
git checkout <SHA> -- <filename>
```

Undoing commits

```
git reset --hard <SHA>
```

--soft : does not change staging index or working directory

--mixed(default) : changes staging index to match repository and does not change working directory

--hard : changes staging index and working directory to match repository

Ignoring files

- project/.gitignore
- *.php, !index.php

Navigating the commits tree

- Parent commit
  - HEAD^, acf57504^, master^
  - HEAD~1, HEAD~
- Grandparent commit
  - HEAD^^, acf57504^^, master^^
  - HEAD~2

```
git ls-tree HEAD
```

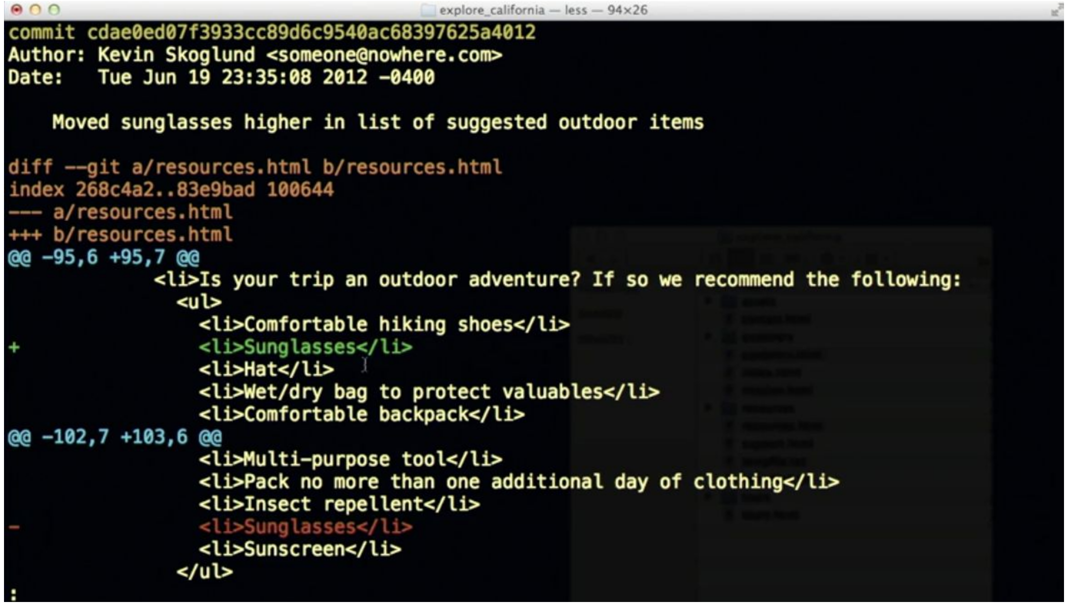
```
git ls-tree master^
```

The commit log

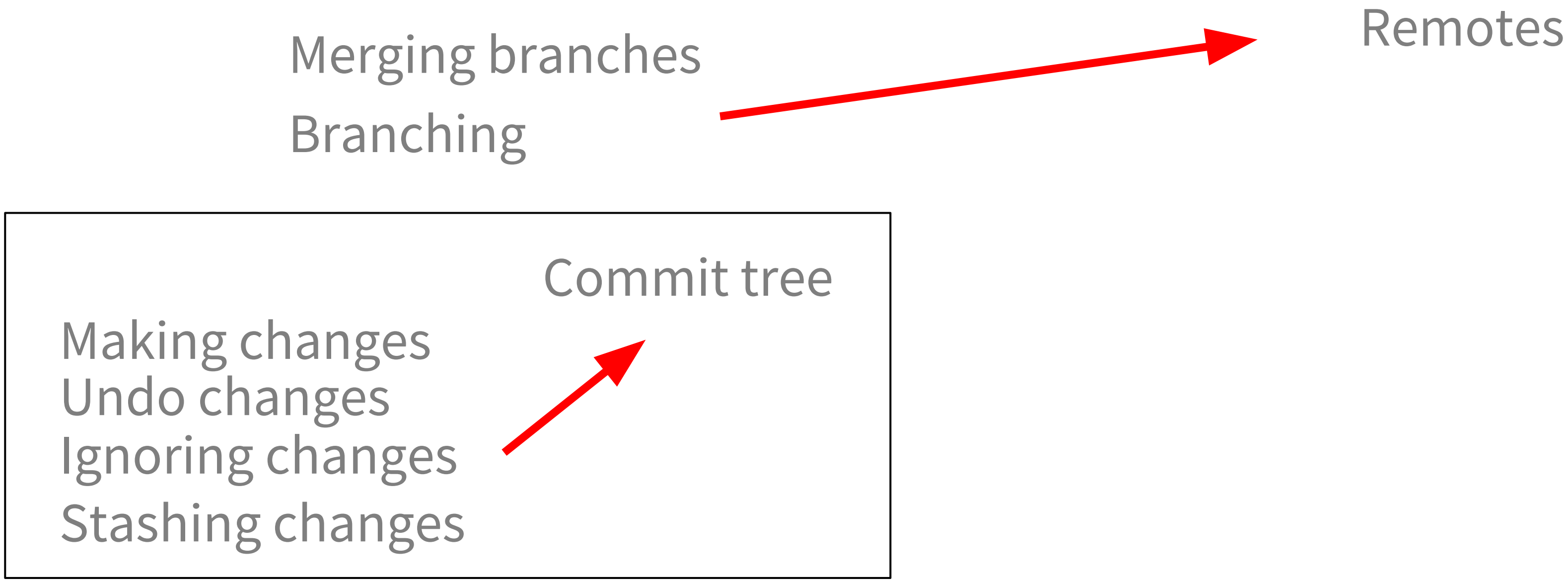
```
git log

git log --oneline

git log --since=“2012-06-25”

git log --until=“3 days ago”

git log --since=“2 weeks” --until=3.days

git log --author=”John”

git log -p
```

Viewing commits

git show <SHA>



```
commit cdae0ed07f3933cc89d6c9540ac68397625a4012
Author: Kevin Skoglund <someone@nowhere.com>
Date:   Tue Jun 19 23:35:08 2012 -0400

    Moved sunglasses higher in list of suggested outdoor items

diff --git a/resources.html b/resources.html
index 268c4a2..83e9bad 100644
--- a/resources.html
+++ b/resources.html
@@ -95,6 +95,7 @@
             <li>Is your trip an outdoor adventure? If so we recommend the following:
               <ul>
                 <li>Comfortable hiking shoes</li>
+                <li>Sunglasses</li>
                 <li>Hat</li>
                 <li>Wet/dry bag to protect valuables</li>
                 <li>Comfortable backpack</li>
@@ -102,7 +103,6 @@
                 <li>Multi-purpose tool</li>
                 <li>Pack no more than one additional day of clothing</li>
                 <li>Insect repellent</li>
-                <li>Sunglasses</li>
                 <li>Sunscreen</li>
               </ul>
:
```



Merging branches

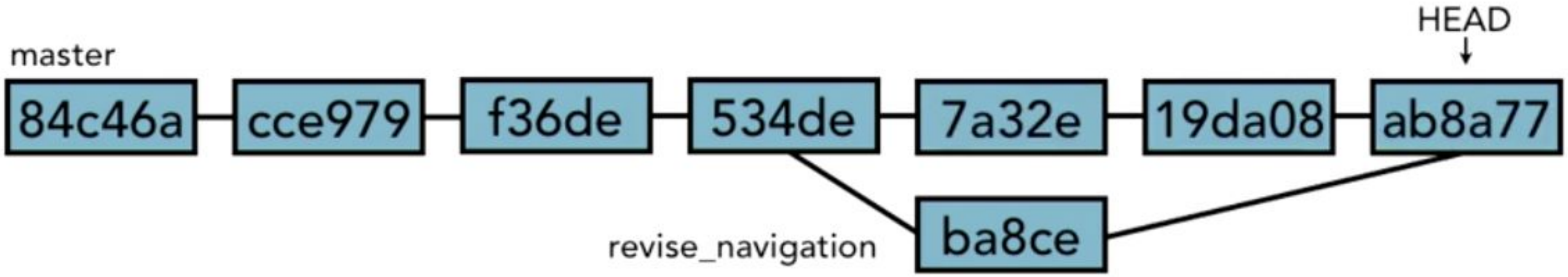
Branching

Remotes

Commit tree

Making changes
Undo changes
Ignoring changes
Stashing changes

Branching



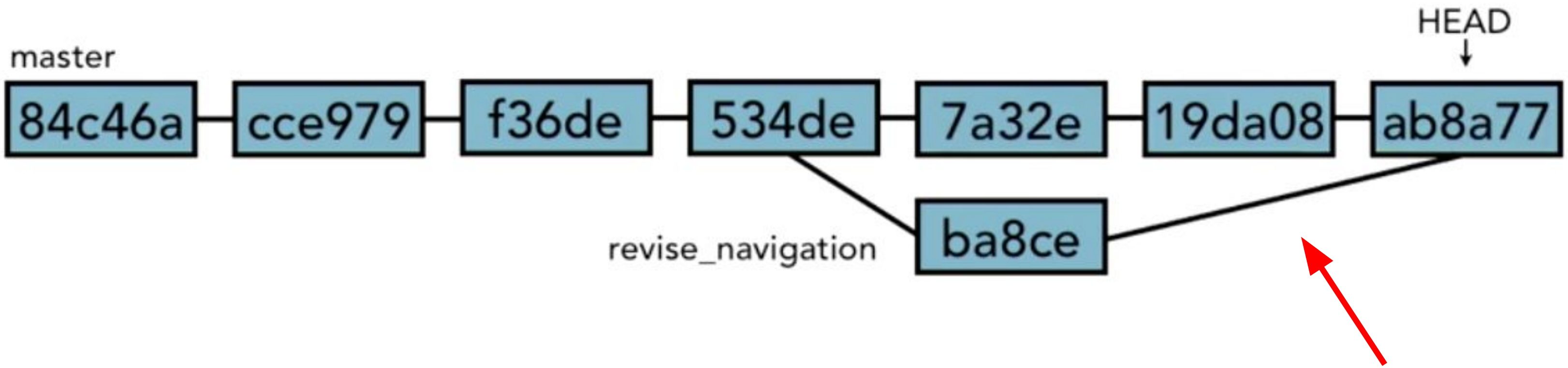
git branch	(show all branches)
git branch <a new branch name>	(create a new branch)
git checkout <branch name>	(switch branch)
git checkout -b <a new branch name>	(create and switch branch)

Branching

git diff <branch name> <branch name>	(show differences between 2 branches)
git branch -m <branch name> <new branch name>	(rename)
git branch -d <branch name>	(delete branch)

Merging branches

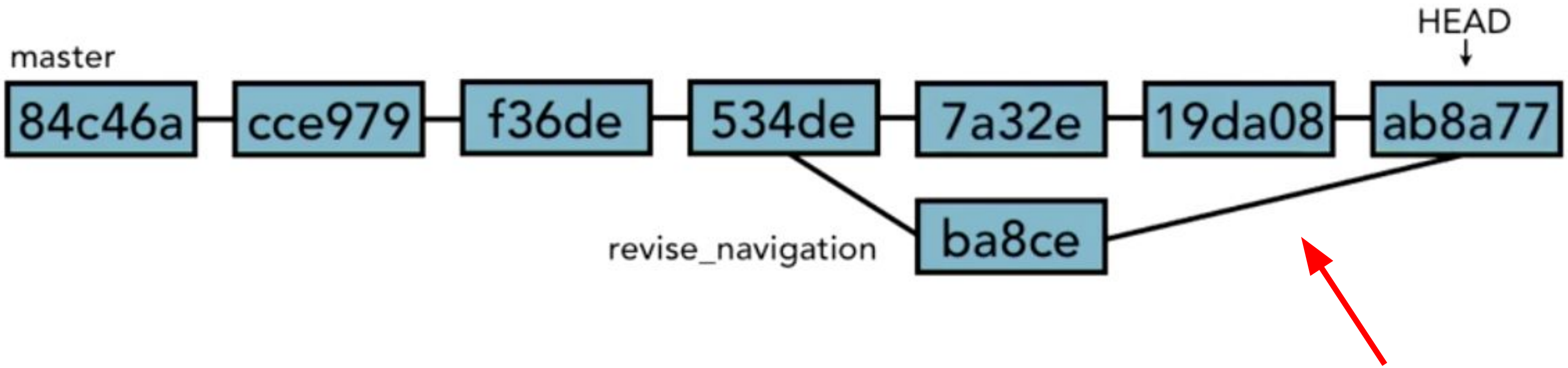
git merge <branch name>   (merge a branch into HEAD branch)



Merging branches

git merge <branch name>

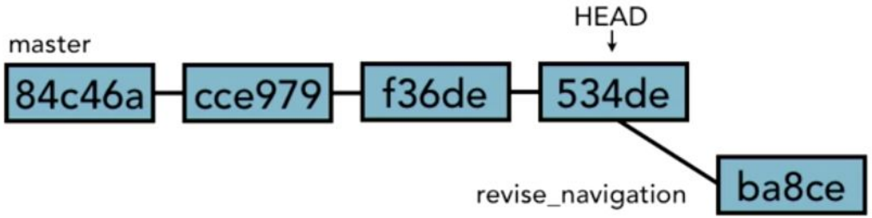
(merge a branch into HEAD branch)



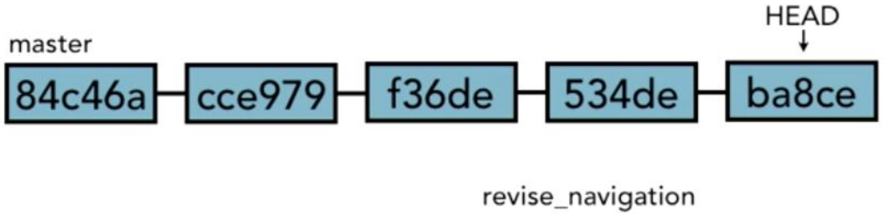
Using fast-forward merge vs. true merge

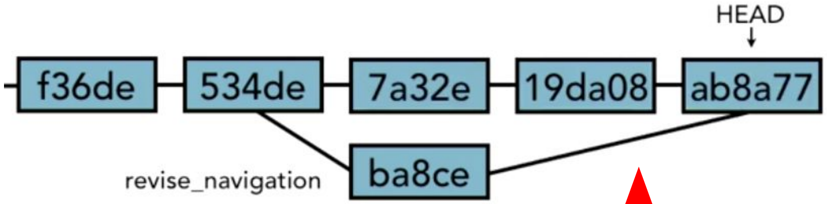
git merge --ff-only <branch name>

git merge --no-ff <branch name>



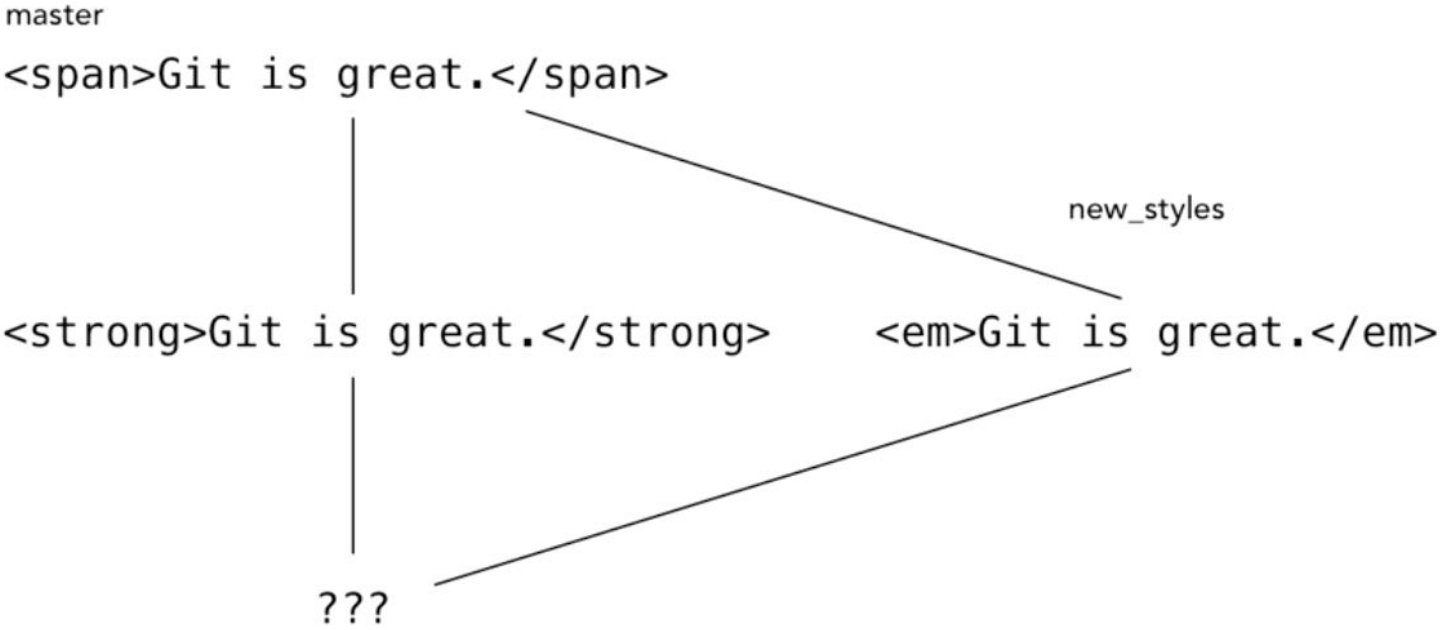
fast-forward merge





true merge

Merge conflicts



master

<span>Git is great.</span>

new_styles

<strong>Git is great.</strong>

<em>Git is great.</em>

???

Resolving merge conflicts

- Abort merge
    - git merge --abort
- Resolve the conflicts manually
    - git merge <filename>
    - Choose a needed version in the file
- Use a merge tool
    - git mergetool --tool= <toolname>

Stashing changes

Saving changes

- git stash
- git stash save ‘name’

Stashing changes

View stash list

- git stash list
- git stash show title’

Stashing changes

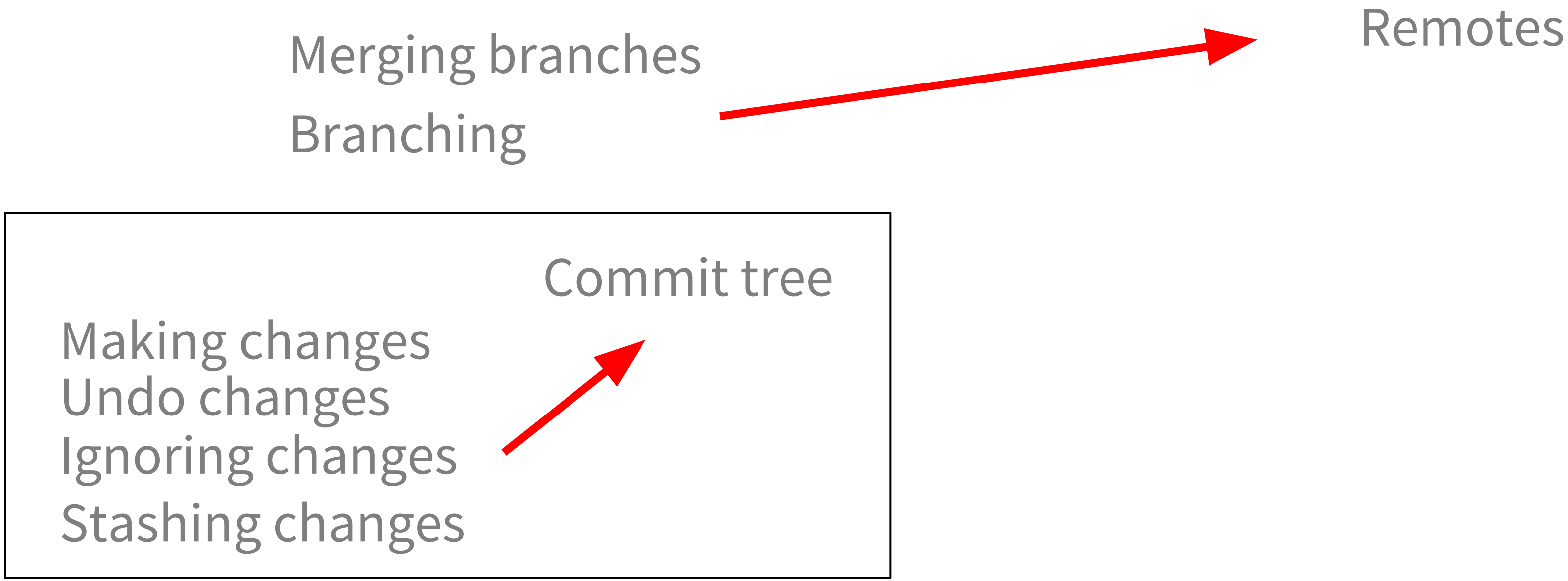
Retrieving stash

- git stash pop
- git stash apply

Stashing changes

Deleting stash

- git stash drop ‘name’
- git stash clear



Remotes

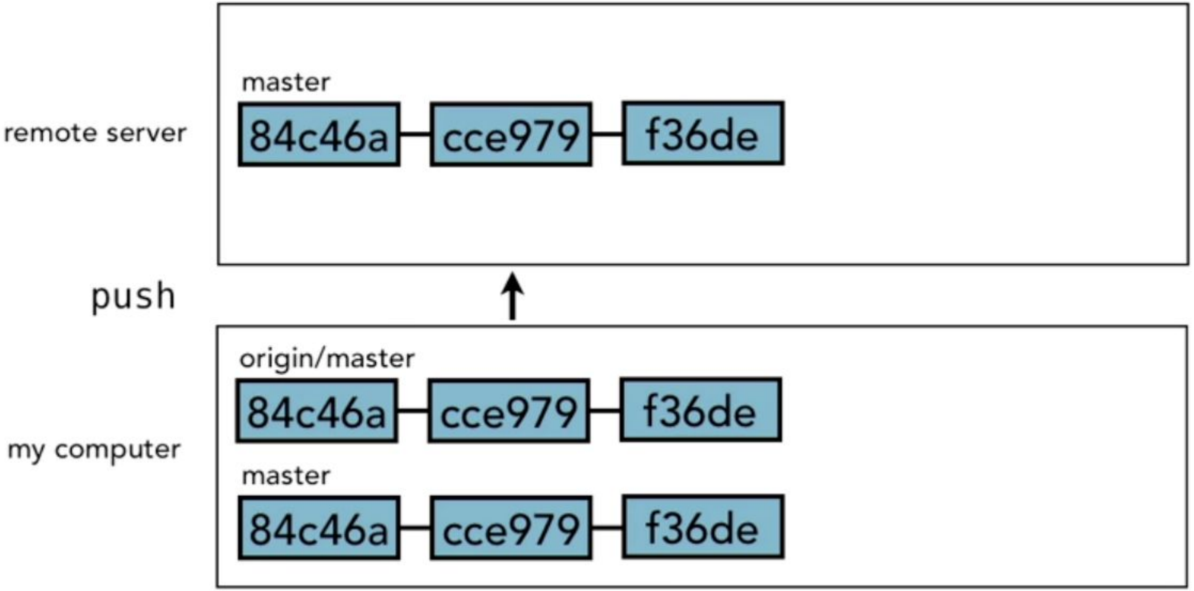
Merging branches

Branching

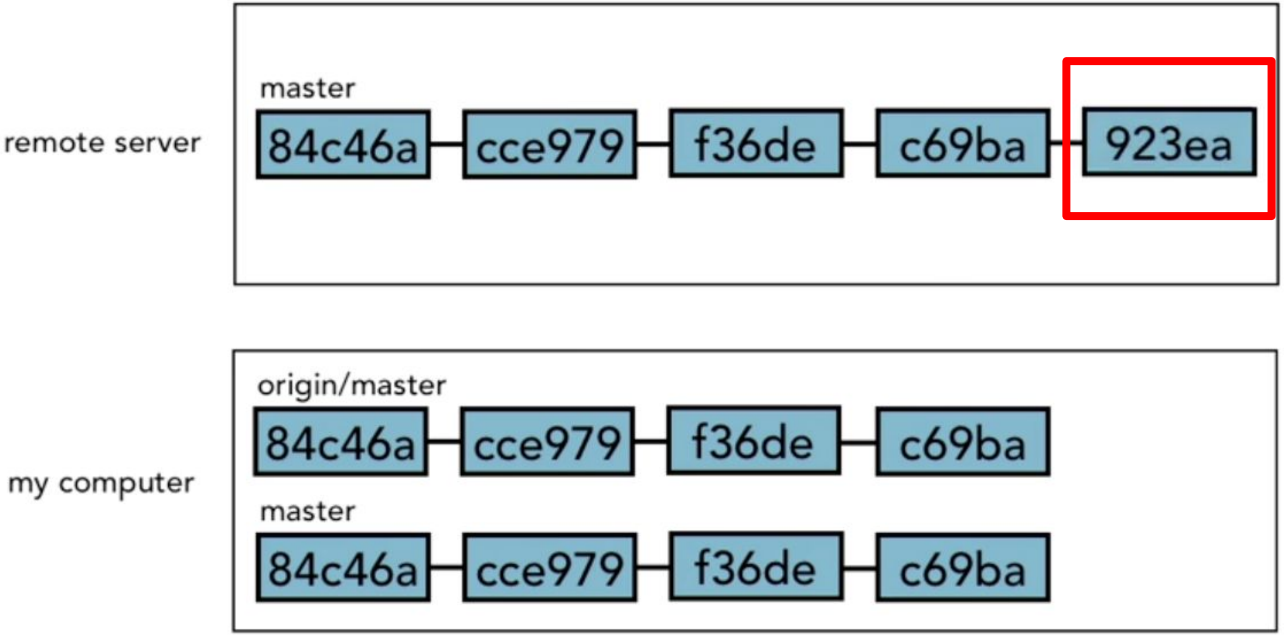
Commit tree

Making changes
Undo changes
Ignoring changes
Stashing changes

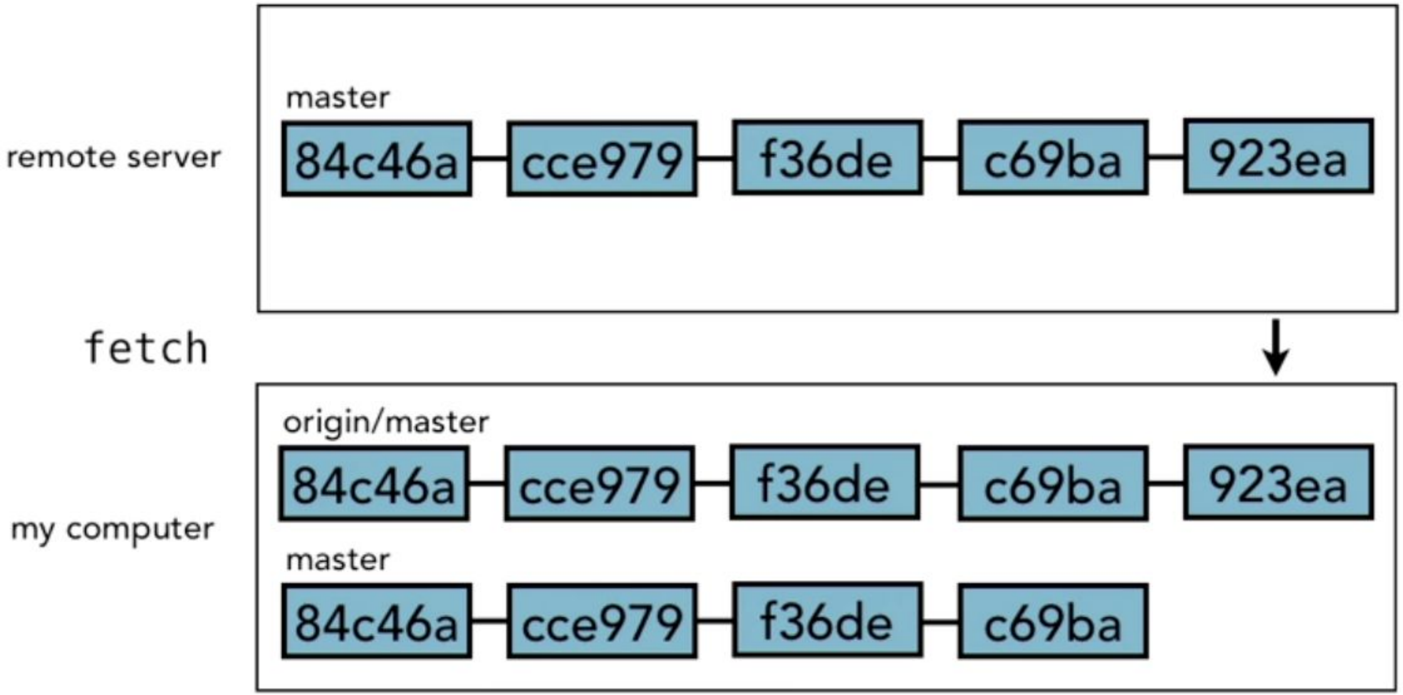
Remotes


remote server
master
84c46a
cce979
f36de
push
origin/master
84c46a
cce979
f36de
my computer
master
84c46a
cce979
f36de

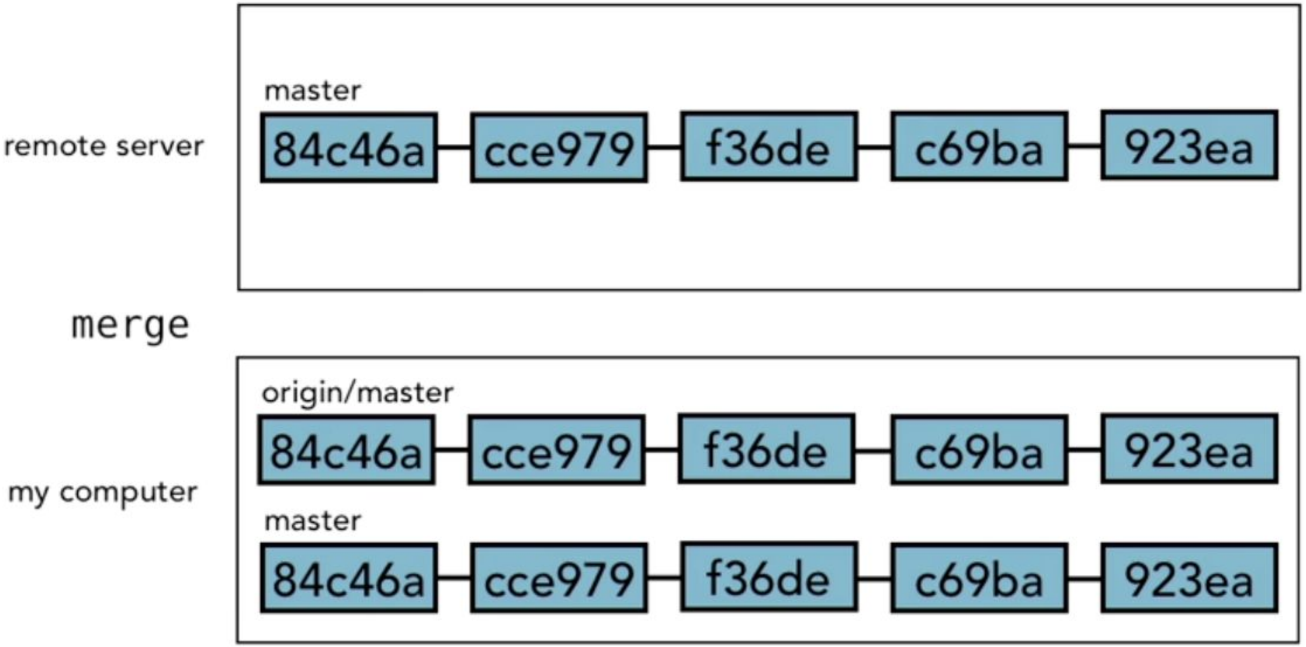
Remotes


remote server
master
84c46a
cce979
f36de
c69ba
923ea
my computer
origin/master
84c46a
cce979
f36de
c69ba
master
84c46a
cce979
f36de
c69ba

Remotes


remote server
master
84c46a
cce979
f36de
c69ba
923ea
fetch
my computer
origin/master
84c46a
cce979
f36de
c69ba
923ea
master
84c46a
cce979
f36de
c69ba

Remotes



remote server

master

84c46a — cce979 — f36de — c69ba — 923ea

merge

my computer

origin/master

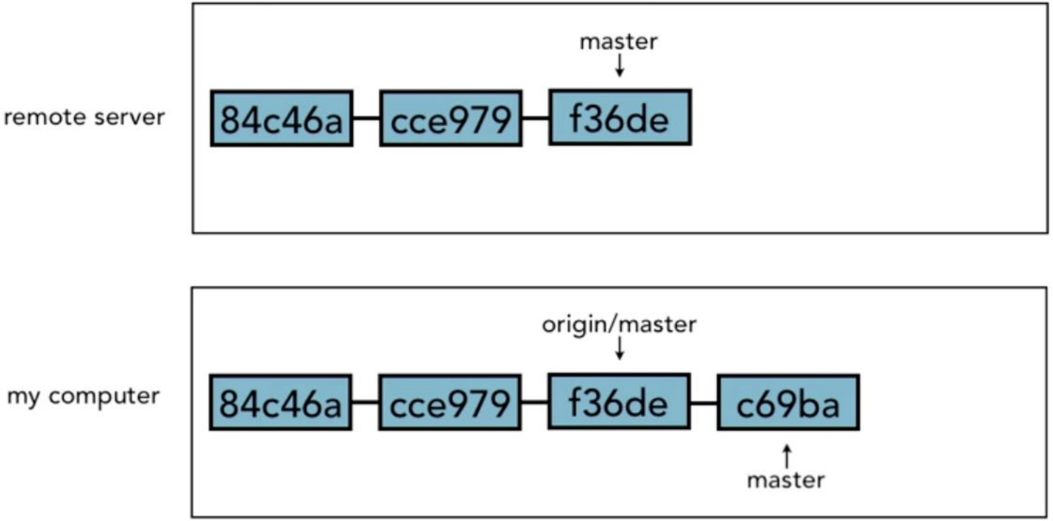
84c46a — cce979 — f36de — c69ba — 923ea

master

84c46a — cce979 — f36de — c69ba — 923ea

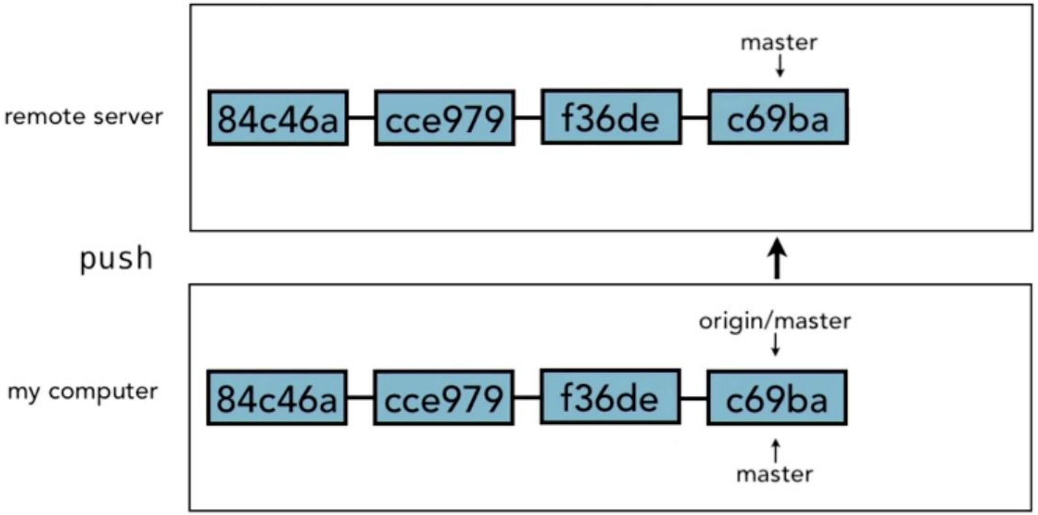
Remotes

If we make the change



Remotes

Push it



Remotes

Adding remote

- git remote add <remote_name> <url>

View

- git remote -v

Remove

- git remote rm <remote_name>

Remotes

Pushing the change

- git push -u <remote_name> <branch_name>

Show brach

- git branch (local)
- git branch -r (remote)
- git branch -a (both)

Remotes

Clone remote

- git clone <url>

Fetching

- Fetch before you work
- Fetch before you push
- Fetch often

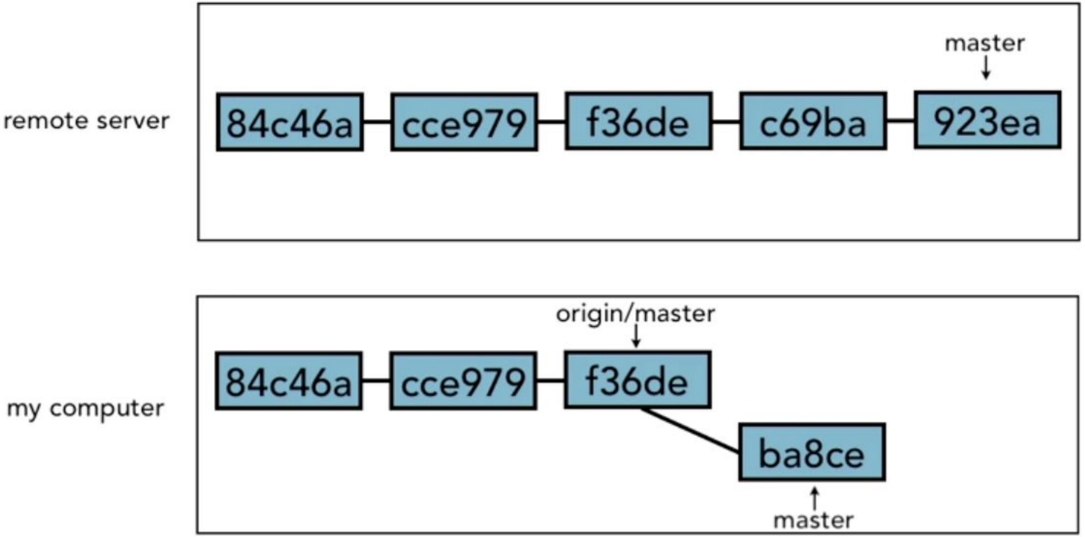
Remotes

Merging

- git merge <target_branch>

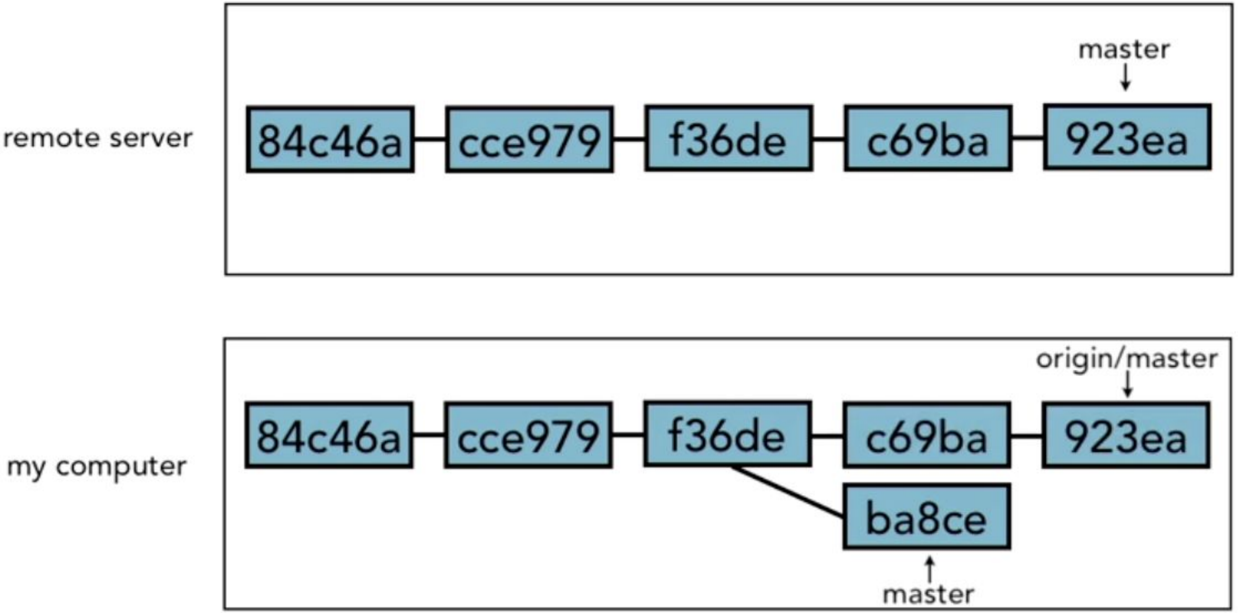
Remotes

Push to updated branch



Remotes

Update



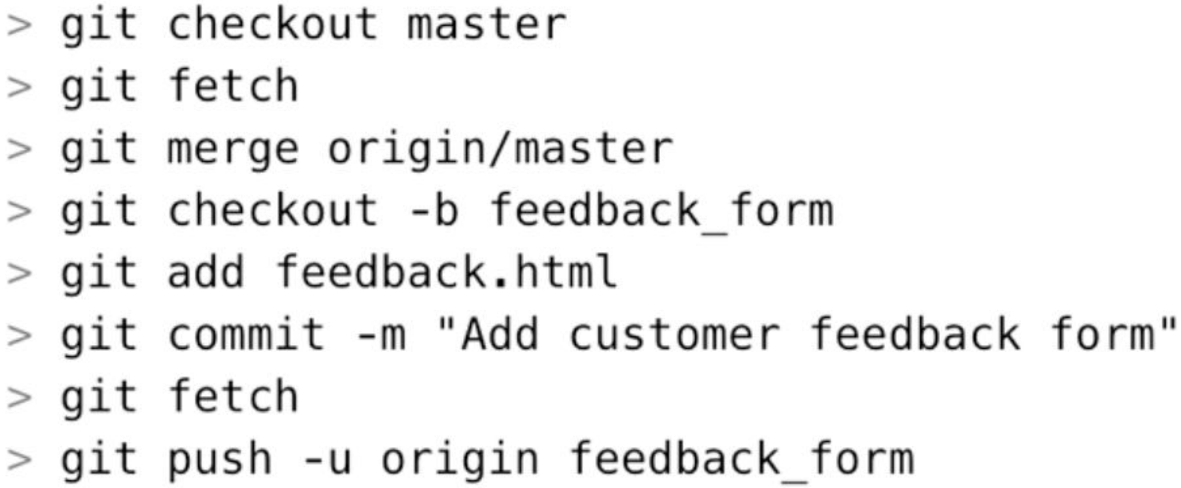
Remotes

Delete branch

- git branch -d <branch_name>

Remotes

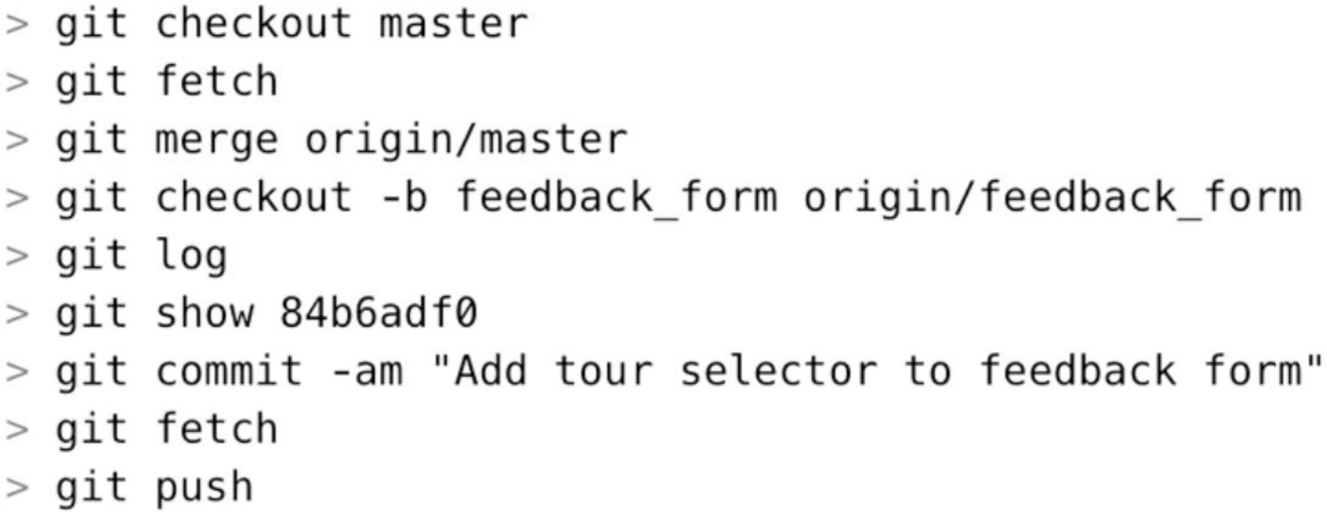
Workflow



```
> git checkout master
> git fetch
> git merge origin/master
> git checkout -b feedback_form
> git add feedback.html
> git commit -m "Add customer feedback form"
> git fetch
> git push -u origin feedback_form
```

Remotes

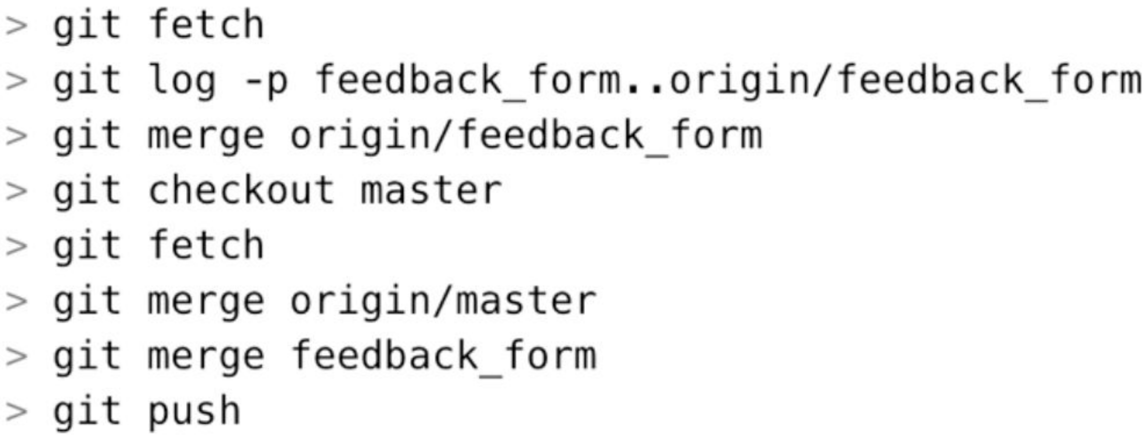
Someone do something



```
> git checkout master
> git fetch
> git merge origin/master
> git checkout -b feedback_form origin/feedback_form
> git log
> git show 84b6adf0
> git commit -am "Add tour selector to feedback form"
> git fetch
> git push
```

Remotes

Workflow



```
> git fetch
> git log -p feedback_form..origin/feedback_form
> git merge origin/feedback_form
> git checkout master
> git fetch
> git merge origin/master
> git merge feedback_form
> git push
```

Tools and next steps

Tools and next steps

Set up alias

- git config --global alias.st “status”
- git config --global alias.co “checkout”

SSH

Generate ssh

- ssh-keygen

Copy .pub and paste into git setting

https://medium.com/altotech/how-to-use-git-and-github-pea-hive-68b7af3e513b