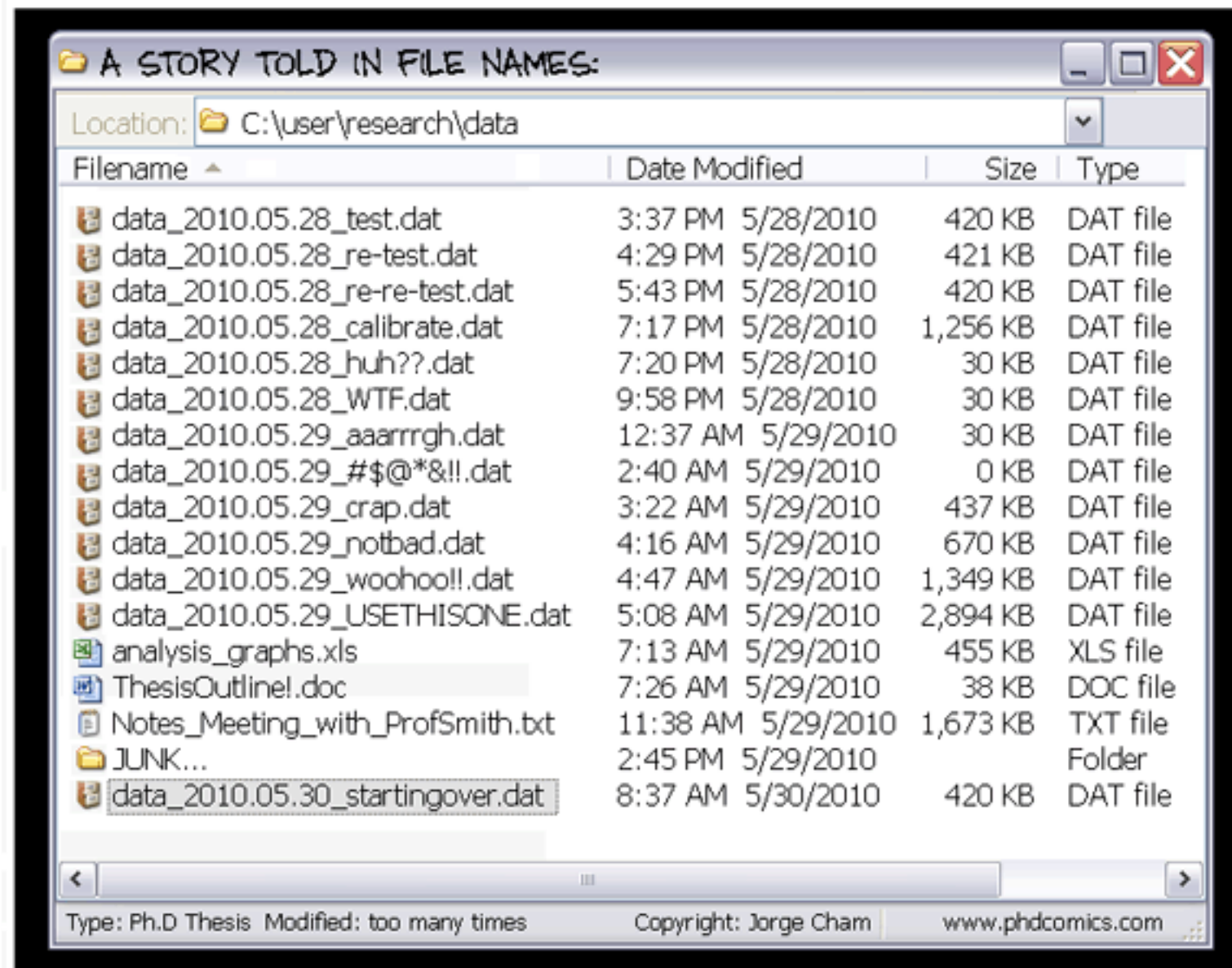


# Collaborating with **git**/**github**

**Rike-Benjamin Schuppner**

**@debilski // <http://debilski.de>**

# Motivation: A story told in filenames



<http://www.phdcomics.com/comics/archive.php?comicid=1323>

# **Why version control?**

- **Keeps all historical versions for easy tracking and reference**
- **Allows you to quickly checkpoint new ideas (and dump them if they fail)**
- **Benefits team collaboration**
- **Improves our efficiency**
- **Can be used as 'data centre' to make build automation possible and testing easy**

# Creating a project

- **Initializing**
  - **\$ git init .**  
**Initialized empty Git repository in /home/myself/Project01/.git/**
- **Cloning an existing project**
  - **\$ git clone <https://github.com/ASPP/GeoSim-Fall-School>**

# Configuration

- **Your identity:**
  - **\$ git config --global user.name "Your Name Comes Here"**  
**\$ git config --global user.email you@yourdomain.com**
- **Setting up your editor:**
  - **\$ git config --global core.editor vim**
- **Checking your settings:**
  - **\$ git config --list**

# **.gitignore**

- **Each repository (rather: each folder) can contain a .gitignore file**
- **Files and file patterns you do not want to track belong there**
  - **.pyc**  
**build/**  
**.o**  
**.DS\_Store**
- **Git also excludes them from `git status` automatically**

# Git concepts

- **Working directory**
  - **All the files you created in the directory**
- **Staging area (also called 'index')**
  - **Intermediate place to put files before committing them**
- **Master (commit), HEAD**
  - **Immutable snapshot with complete history, which can be shared with other people**

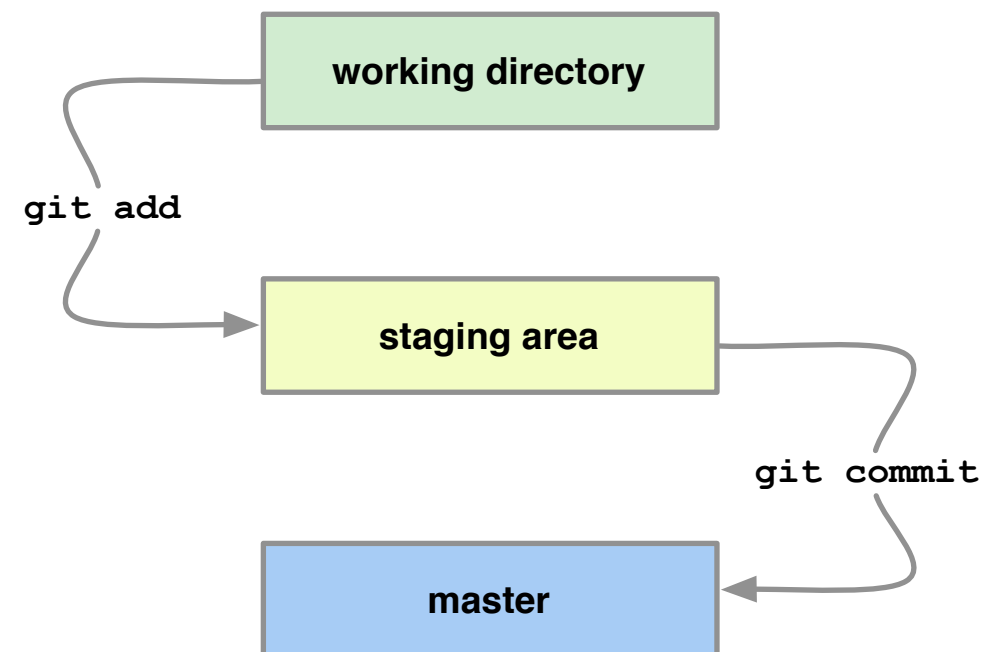
**working  
directory**

**staging  
area**

**master**

# Using the staging area

- Add files to staging area with
  - `$ git add <filename>`
- Commit the *whole* staging area using
  - `$ git commit`
  - `$ git commit -m "I committed those files"`





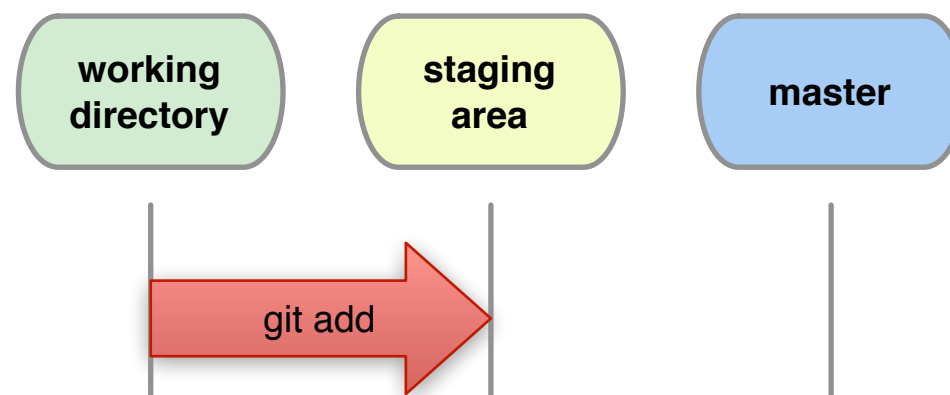
# Git commit

- Committing a version generates a hash calculated from:
  - Name of committer, current date
  - SHA of (one or more) parent commits
  - SHA of data tree
  - eg: `2a9a3c876b344cb74807edd6340b7bcd930014a8`
- This means that everybody who has a commit with the same hash knows:
  - It is from the same person
  - It has the same history
  - It contains the same data
- Sharing data is simply checking that the most recent hash is the same

# git add <filename>

adds a file to the index

- For adding *a//*tracked (and changed) files (and removing deleted ones):
  - `git add -u`
- For adding interactively:
  - `git add -p`



# git rm <filename>

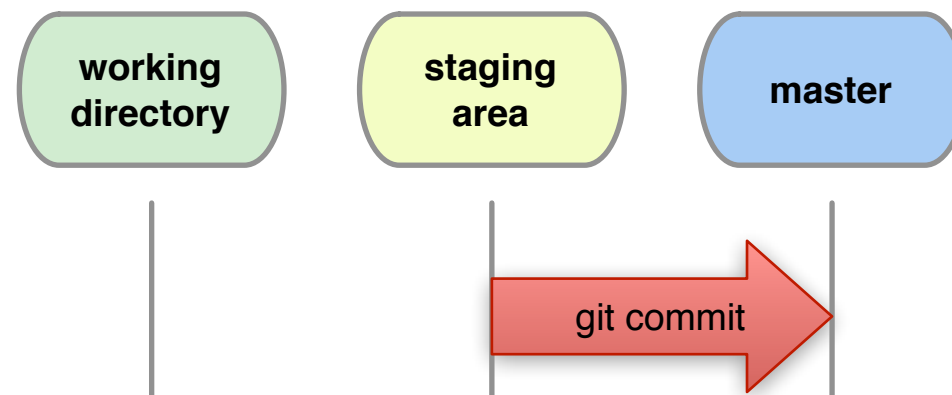
removes a file from the index

- For moving/renaming files, there is also
  - `$ git mv oldname newname`
  - (Internally it is the same as doing `git rm + git add`)

# git commit [-m "Commit message"]

adds a file to the index

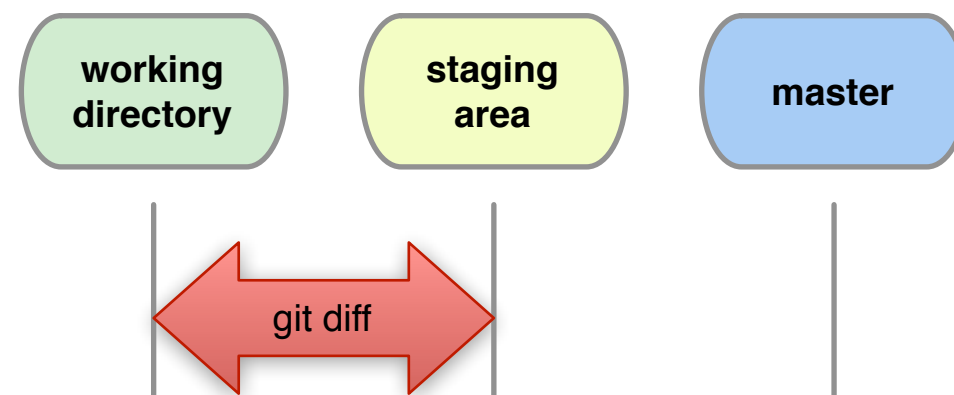
- Creates a new commit with the current index
- Moves the current HEAD branch to the new commit



# git diff

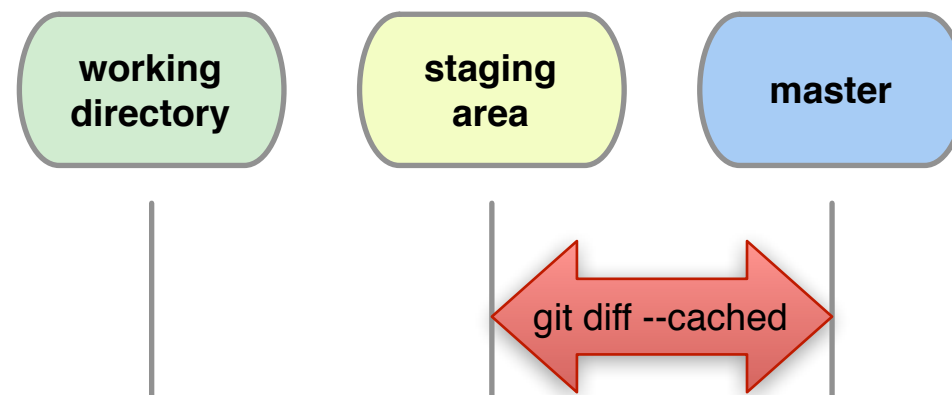
Shows what has been changed in working (and not yet added to the index)

- `diff --git a/Monday/Introduction-to-Plotting.ipynb b/Monday/Introduction-to-Plotting.ipynb`  
index 14fd585..d4b4fa7 100644  
--- a/Monday/Introduction-to-Plotting.ipynb  
+++ b/Monday/Introduction-to-Plotting.ipynb  
@@ -1,7 +1,7 @@  
{  
 "metadata": {  
 "name": "",  
- "signature": "sha256:61bfab1750c869858224ad5cb43995bf3327bcaa6cf0890a542bbf6623a26806"  
+ "signature": "sha256:00e69344cf858d41422342f6ec2179efe0482194a2dc8dec8bc7d3ec42d488d2"  
 }, "nbformat": 3,



# git diff --cached

Shows what is to be committed to HEAD (eg. shows the index)



# git log

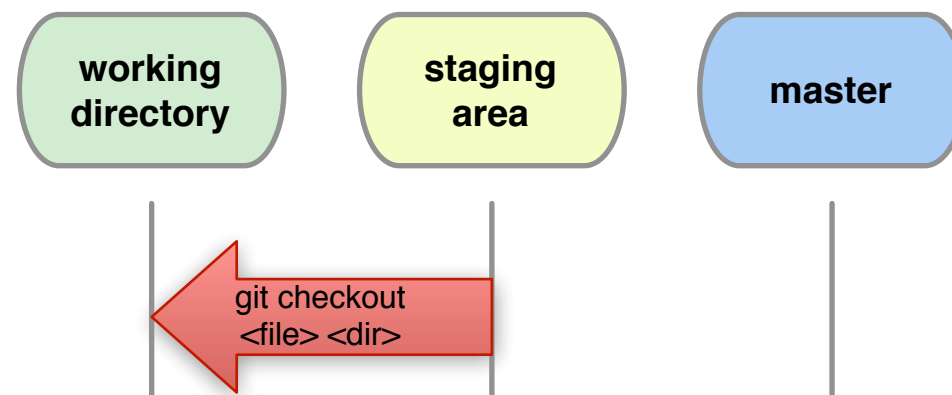
shows the history

- **commit 2a9a3c876b344cb74807edd6340b7bcd930014a8**  
**Author: Rike-Benjamin Schuppner <rikebs@debilski.de>**  
**Date: Mon Oct 6 09:07:07 2014 +0200**  
**ENH: Updates.**
- **commit d3cd330f259617aa7a9b59bb07f2bf15129bd4a1**  
**Author: Rike-Benjamin Schuppner <rikebs@debilski.de>**  
**Date: Fri Oct 3 14:47:44 2014 +0200**  
**NF: Introduction to Plotting.**
- **\$ git log --oneline --graph**

# git checkout -- <filename>

resets a file to the version in index

- Can also retrieve a file from a specific commit
- **\$ git checkout COMMIT -- <filename>**

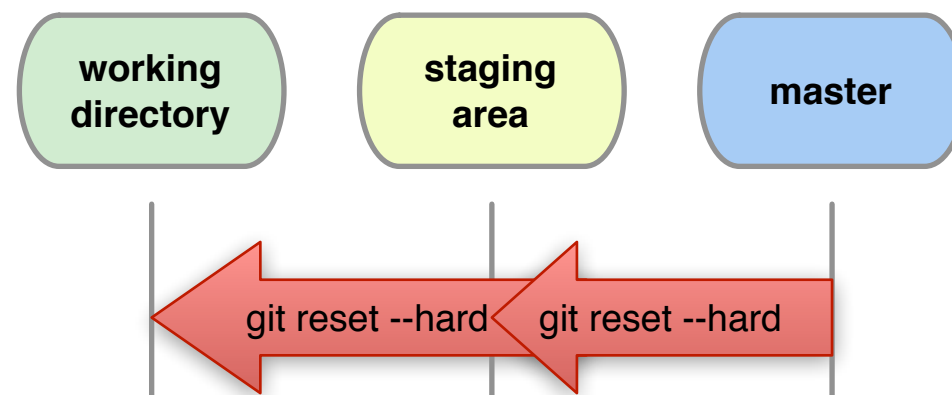




# git reset

reset current HEAD

- You can discard all changes in the working directory and index with
  - `$ git reset --hard`
- Always check git status before doing so
- git reset can also be used to move around the HEAD branch (see help)



# git clean

remove untracked files from working tree

- **Always make a dry-run before cleaning**
  - **\$ git clean -n**
- **-x option also removes files from .gitignore**
- **-X *only* removes files listed in .gitignore**

# Useful tricks

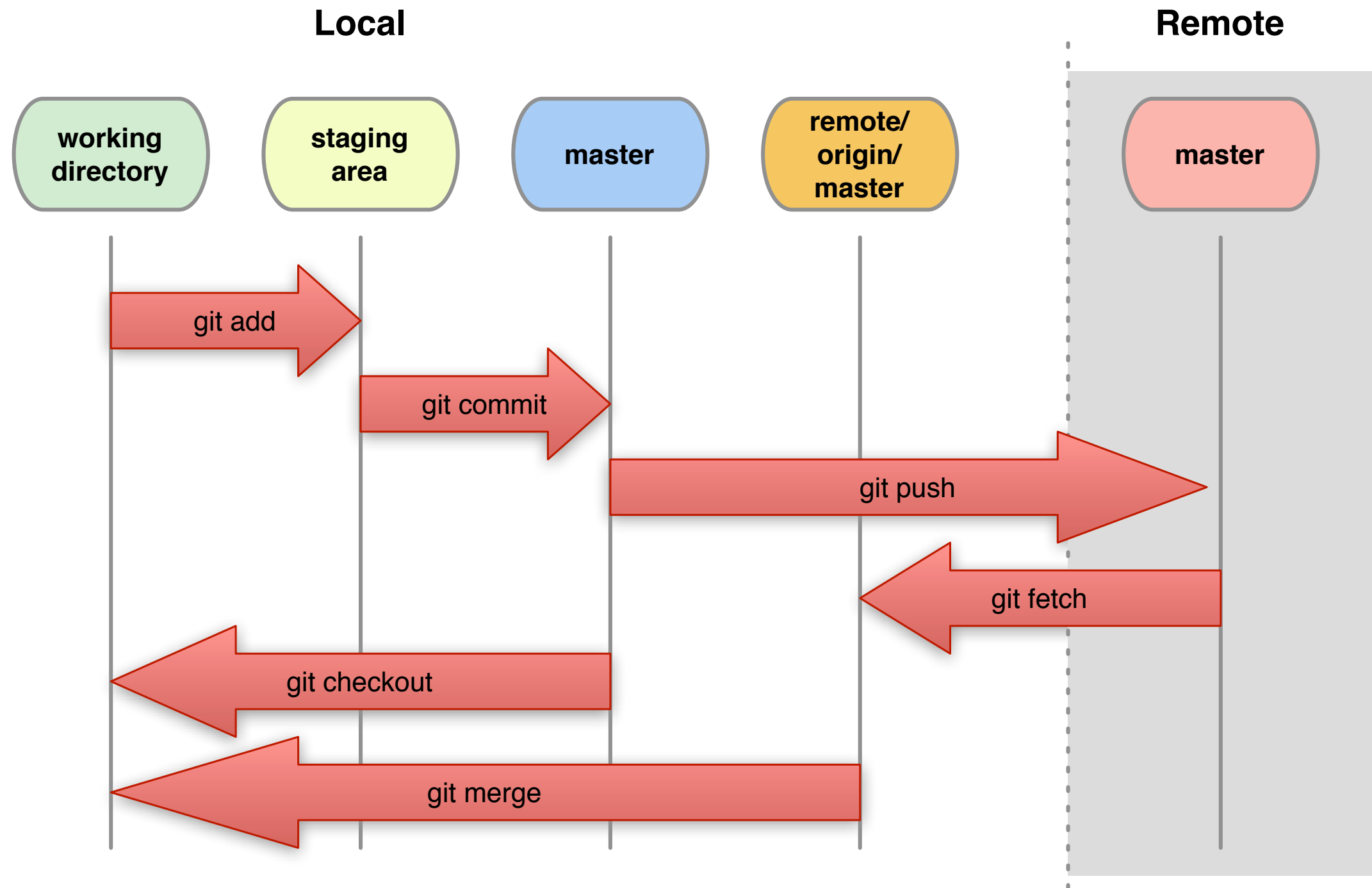
## examine your repository

- Search for a string in all tracked files:
  - `$ git grep PATTERN`
- Search the history for string addition/deletion:
  - `$ git log -Spattern`
- Find out who did something stupid/awesome
  - `$ git blame`

# Branches and remotes

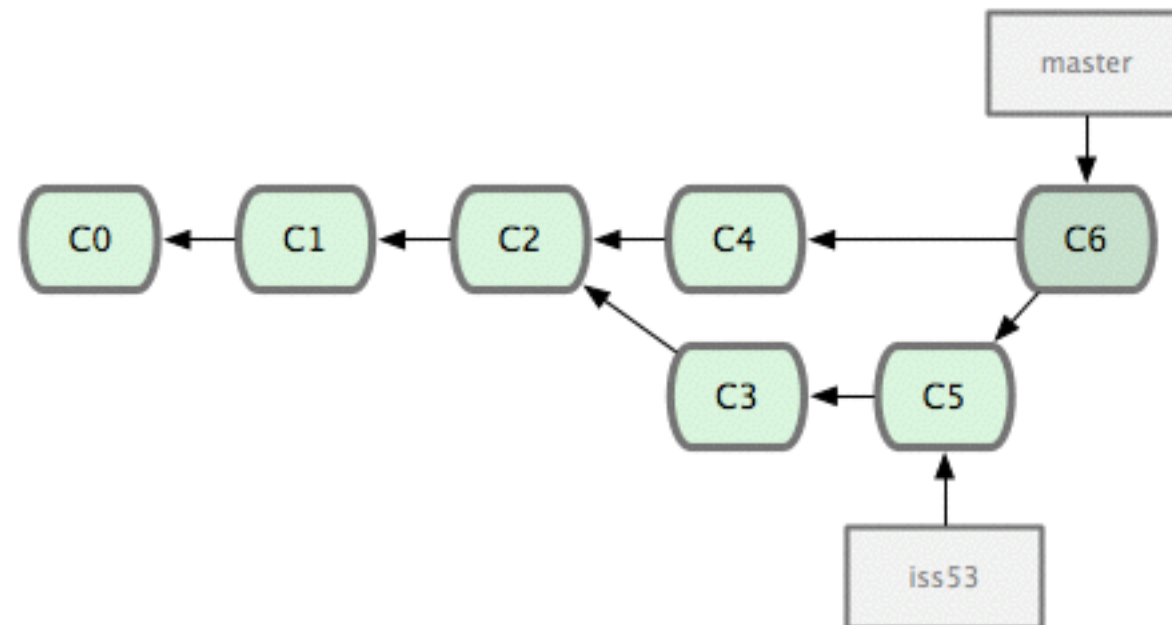
working non-linear and collaborating with others

# Command overview



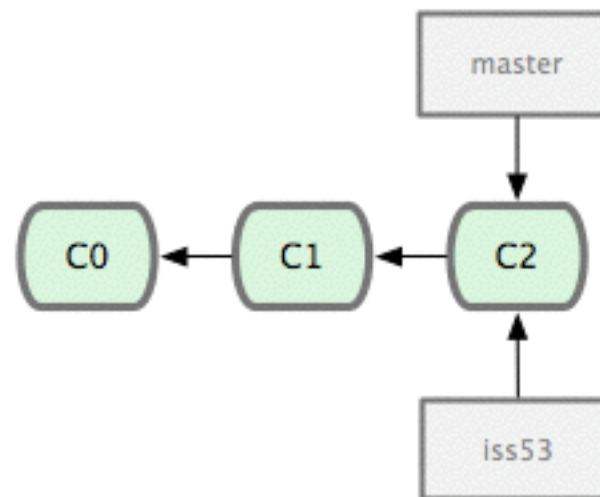
# Branches

History is no linear process



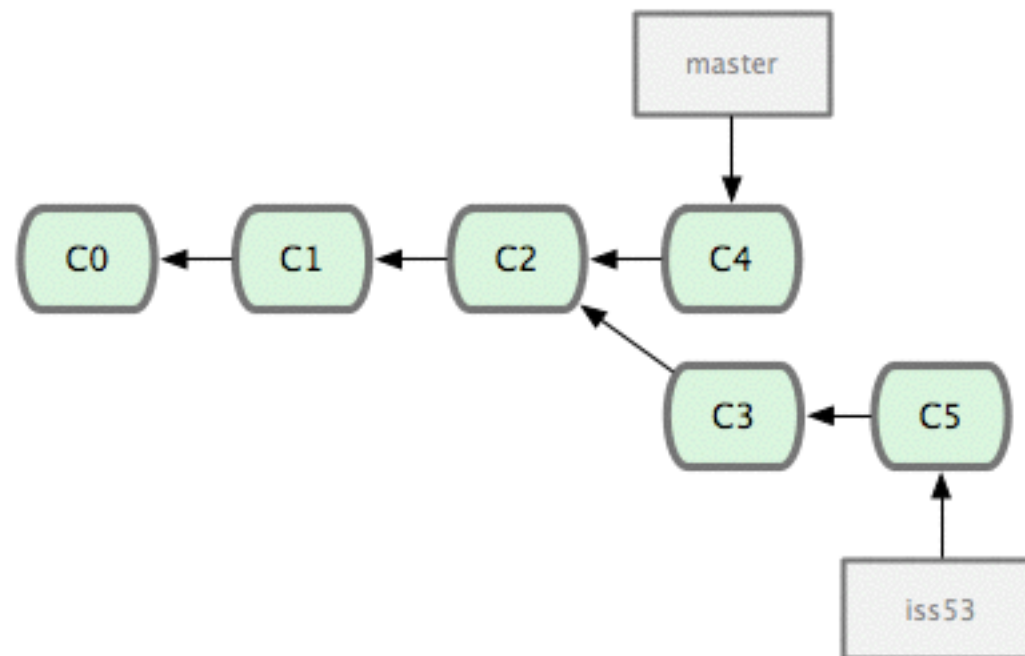
# git branch <branchname>

Creates a new branch



# git checkout <branchname>

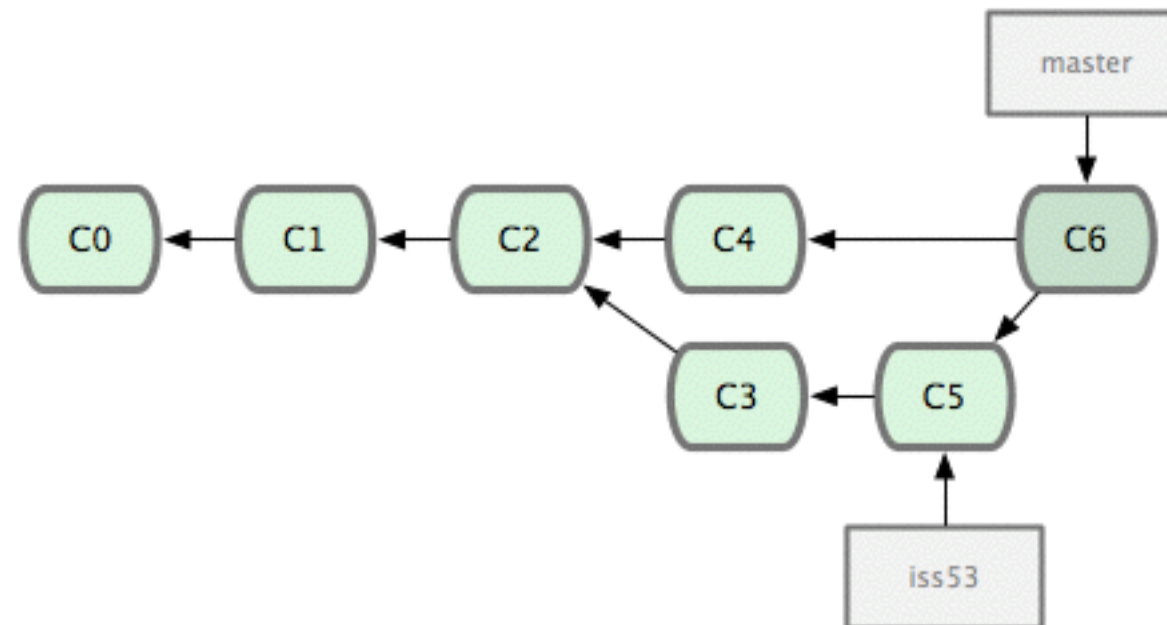
Switches to another branch





# git merge <branchname>

Merges the changes of another branch



# resolving merge conflicts (1)

When branches do not mix

- `$ git merge iss53`

Auto-merging index.html

**CONFLICT** (content): Merge conflict in index.html

Automatic merge failed; fix conflicts and then commit the result.

# resolving merge conflicts (2)

When branches do not mix

- `$ git status`  
index.html: needs merge  
# On branch master  
# Changed but not updated:  
# (use "git add <file>..." to update what will be committed)  
# (use "git checkout -- <file>..." to discard changes in working dir)  
#  
# **unmerged:** index.html  
#

# resolving merge conflicts (3)

When branches do not mix

- `$ cat index.html`

```
<<<<<<< HEAD:index.html
```

```
<div id="footer">contact : email.support@github.com</div>
```

```
=====
```

```
<div id="footer">
```

```
please contact us at support@github.com
```

```
</div>
```

```
>>>>>>> iss53:index.html
```

# resolving merge conflicts (4)

When branches do not mix

- **TODO list:**
  - **Resolve conflicts**
  - **Add file to index**
  - **Commit**
- **If you prefer one specific version:**
  - **\$ git checkout COMMIT -- filename**

# git remote -v

shows/manages the remote repositories

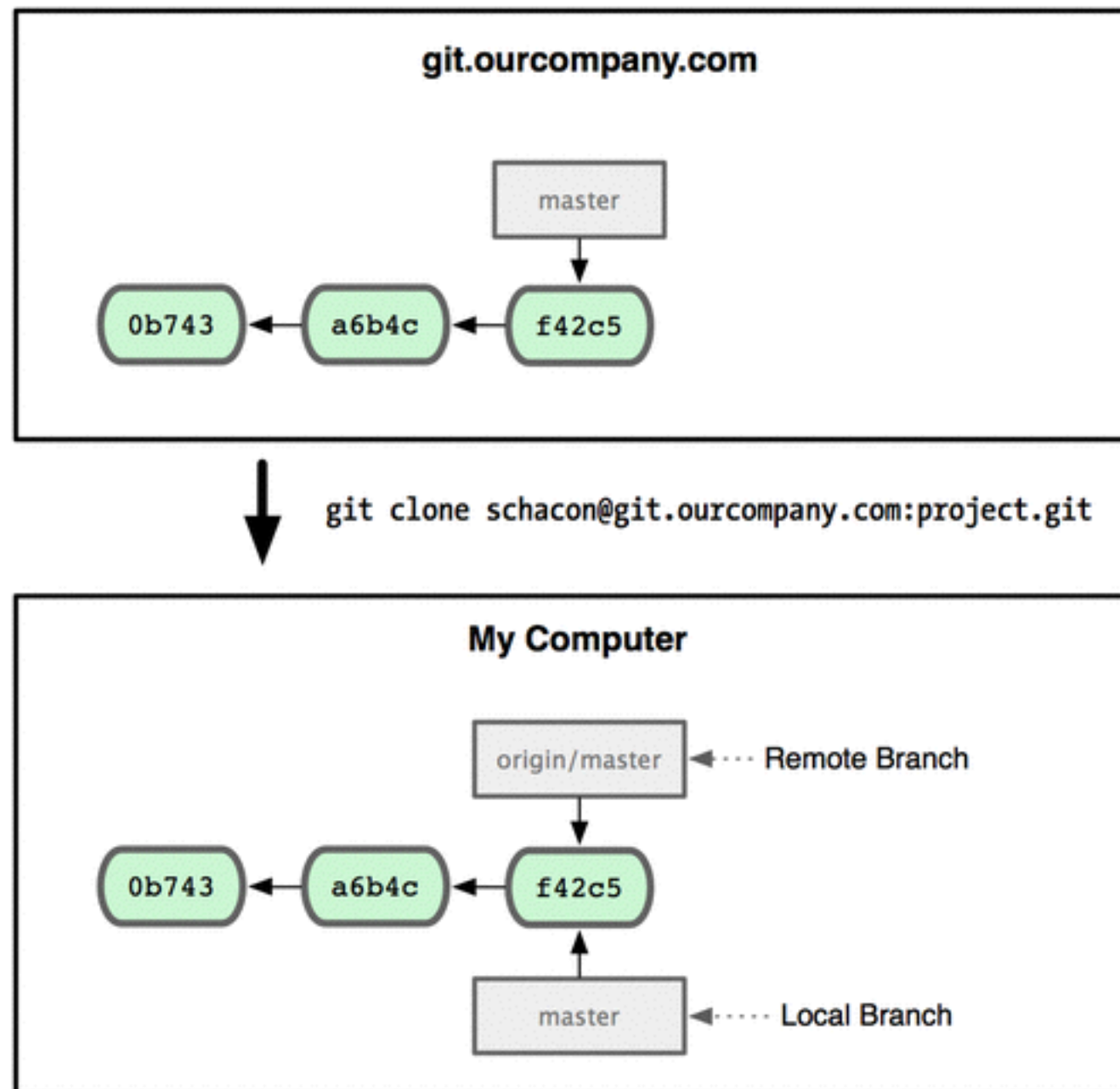
- You can have more than one remote repository! Usually **origin** refers to the remote (but for git it's just a name).
- **\$ git remote -v**  
**origin git@github.com:ASPP/GeoSim-Fall-School.git**
- For adding and removing remotes:
  - **\$ git remote add NAME REPO**
  - **\$ git remote rm NAME REPO**

# Remote repositories

- Set up empty repository on remote server
  - **remote** \$ git init --bare Project01.git
- Add remote to local repo (here named origin)
  - **local** \$ git remote add origin myself@remote:/Project01.git
- Push everything
  - **local** \$ git push origin master

# Working with others

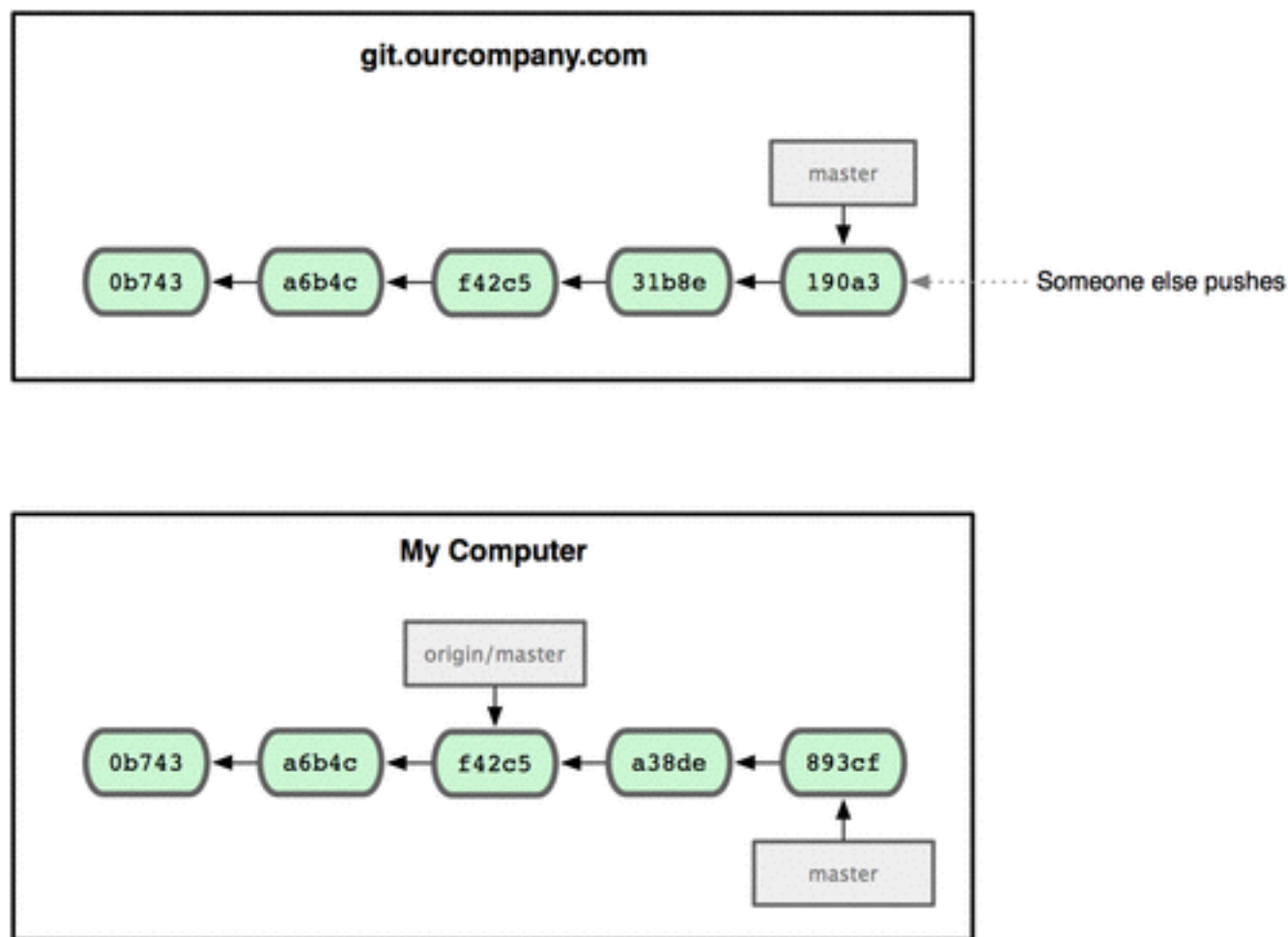
## Cloning another repository





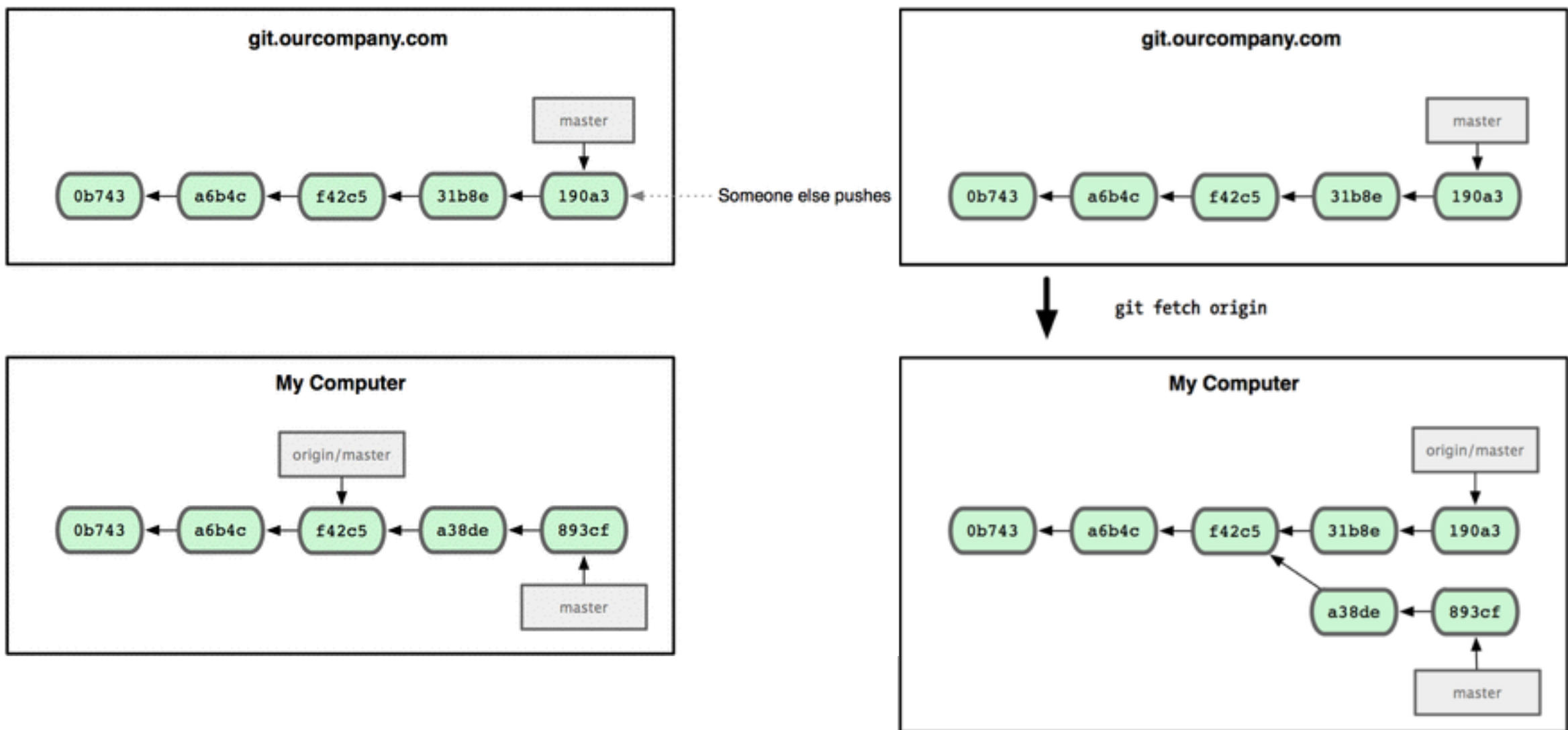
# git fetch origin

Fetch commits from a remote repository



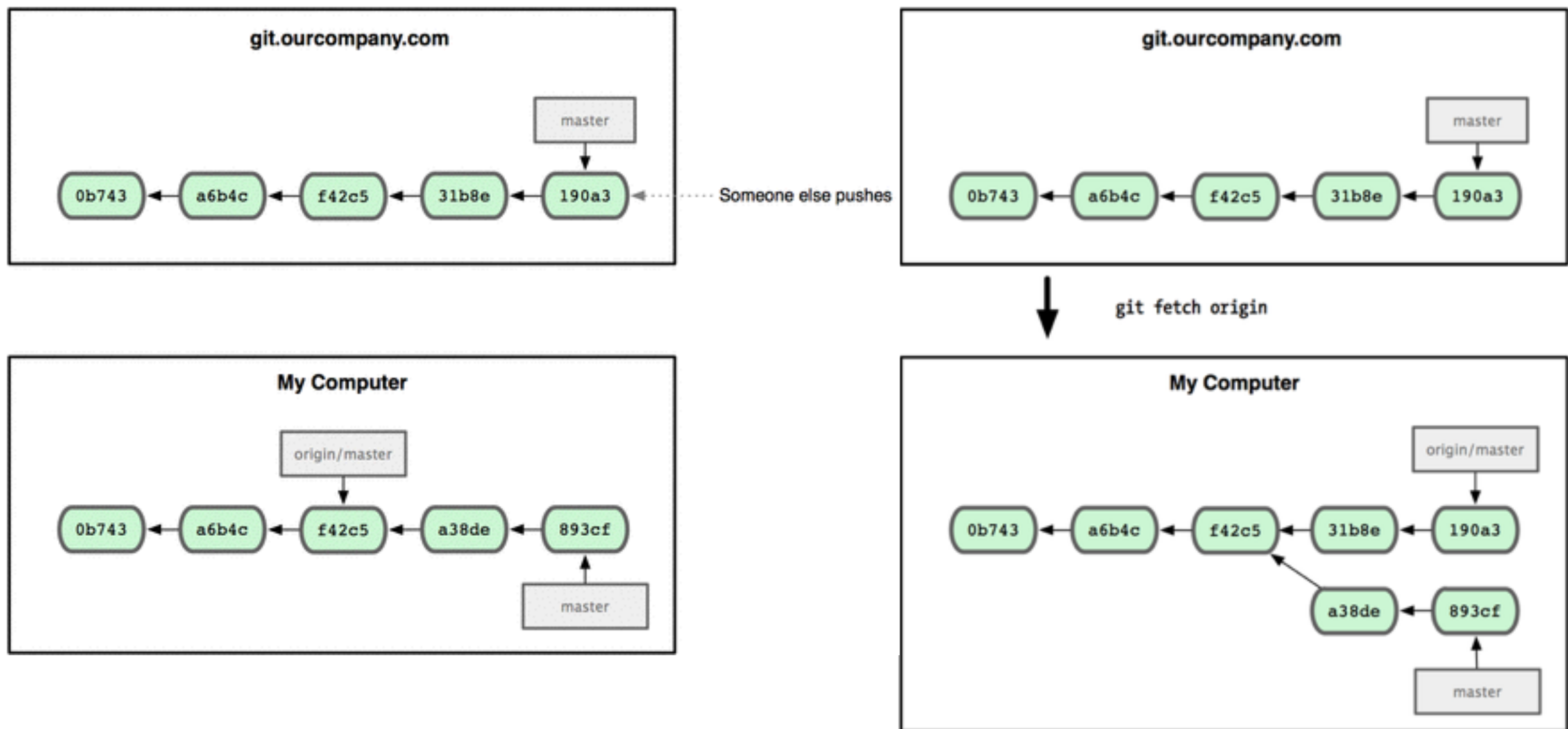
# git fetch origin

Fetch commits from a remote repository



# git fetch origin

Fetch commits from a remote repository



- Does not change local commits. Need to merge manually.

# git pull

= git fetch && git merge

- **Be careful with this one**
- **Fine to use when you want to update your code to upstream and do not have local changes**
- **If you do have local changes, you might prefer merging manually (sometimes)**

# git push

move your changes to a remote repository

- **\$ git push REMOTE BRANCH**
- **If the associated branch of the remote has been updated, your push will be rejected**
- **In this case, do a git pull (or fetch + merge) first and then do a new push**
- **Remote branches will only be moved forward, unless you force them to a specific commit (this makes it harder to rewrite history in a remote repo)**

# UI Helpers

Sometimes there's a better way than using the command line

- In general it is very useful to understand the basic command line commands and also a bit of git internals
- A GUI may never catch all corner cases of what can happen
- Some GUIs look like they need the same amount of learning time than the CLI
- However, GUIs are very useful in two cases:
  - Staging chunks of code (à la `git add -p`)
  - Inspecting history

# GUIs

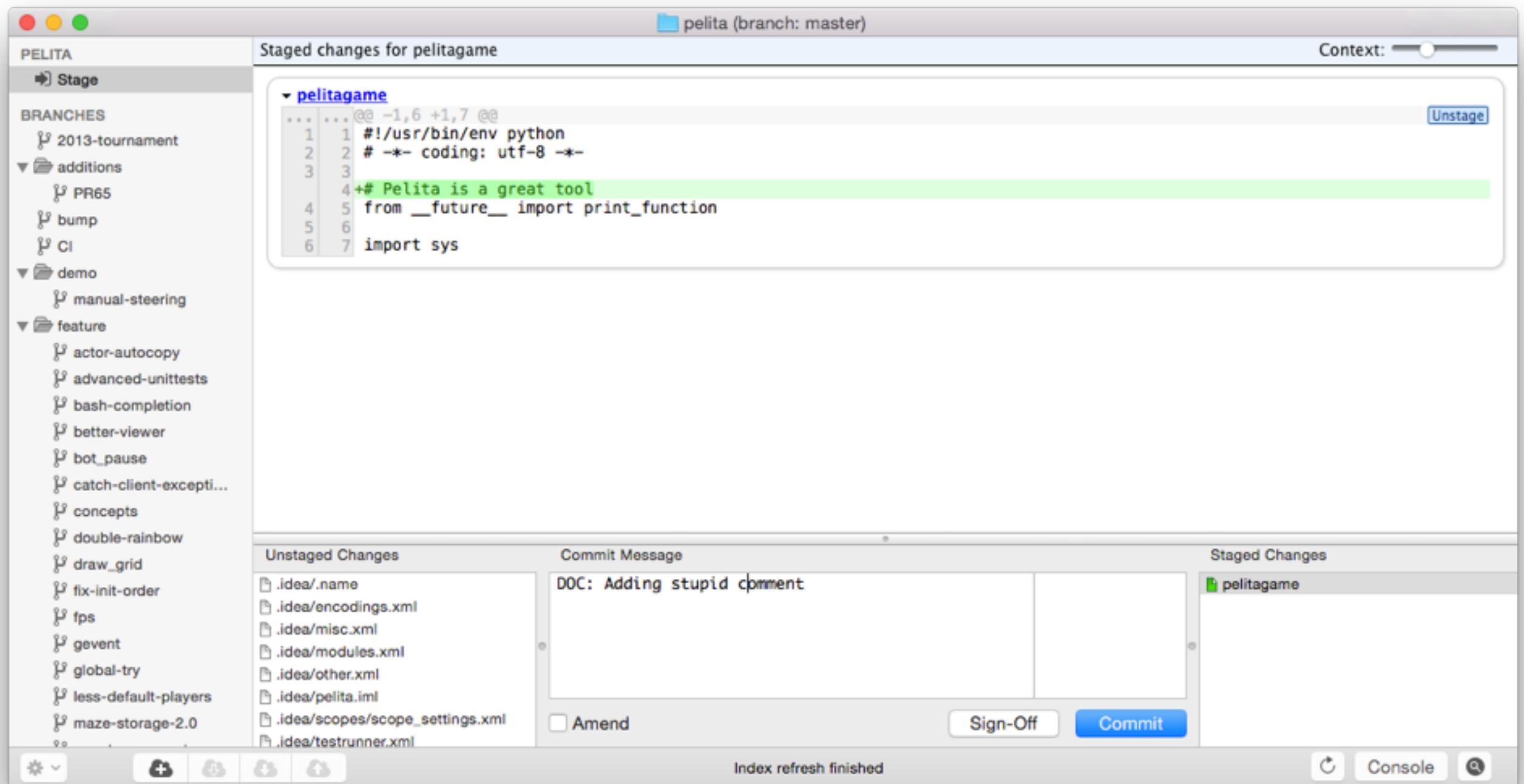
The screenshot displays the Sparkle (Git) GUI interface. The top toolbar includes icons for View, Commit, Checkout, Reset, Stash, Add, Remove, Add/Remove, Fetch, Pull, Push, Branch, Merge, Tag, Show in Finder, and Git Flow. The left sidebar shows the FILE STATUS (Working... 12), BRANCHES (master), TAGS (sparkle-1.5b1 to sparkle-1.5b6), REMOTES, STASHES, SUBMODULES, and SUBTREES. The main area shows a commit history table with columns for Graph, Commit, Description, Author, and Date. The bottom right pane shows a diff view for SUHost.m, highlighting changes in the defaultsDomain variable.

Graph	Commit	Description	Author	Date
	f341d0c	Corrected minor spelling errors	Christian Zachariasen <c...>	Feb 22, 2012, 11:35 AM
	e27828d	Merge pull request #144 from mattstevens/custom-defaults...	Andy Matuschak <andy@...>	Feb 14, 2012, 5:52 PM
	44cf0d0	Merge pull request #145 from mattstevens/install-on-quit-...	Andy Matuschak <andy@...>	Feb 14, 2012, 5:48 PM
	6be5568	Clean up after install on quit updates	Matt Stevens <matt@allo...>	Feb 13, 2012, 3:02 AM
	29b7321	Support a custom user defaults domain	Matt Stevens <matt@allo...>	Feb 11, 2012, 10:31 PM
	449b00e	Merge pull request #139 from mattstevens/pathToRelaunch	Andy Matuschak <andy@...>	Feb 9, 2012, 6:37 AM
	4c25737	Merge pull request #142 from mzch/master	Andy Matuschak <andy@...>	Feb 9, 2012, 6:14 AM
	88d4b34	Improved Japanese translations	Koichi MATSUMOTO <mz...>	Feb 8, 2012, 6:37 AM

Commit: e27828d2485a9f8db8cf21c6b1617dcec22f423c [e27828d]  
Parents: 44cf0d0e51, 29b7321094

```
25 25      SULog(@"Sparkle Error: the bundle being updated at %@",
26 26      +
27 27      +      defaultsDomain = [[bundle objectForKey:@"SUDefault
28 28      +      if (!defaultsDomain)
29 29      +      defaultsDomain = [[bundle bundleIdentifier] retain];
```

# GitX



<http://rowanj.github.io/gitx/>



# GitX

The screenshot shows the GitX application window. On the left is a sidebar with a file tree for a project named 'PELITA'. The main area displays a commit history table with columns for Short SHA, Subject, Author, and Date. The commit '8f7a235' is selected, showing the subject 'NF: Double rainbow patch.' and the author 'Rike-Benjamin Schuppner'. Below the table, the commit details are shown, including the SHA, parent SHA, and a list of files changed. The diff for 'pelita/ui/tk\_canvas.py' is displayed, showing a change in the 'UiCanvas' class constructor.

Short SHA	Subject	Author	Date
63d93e6	Merge pull request #270 from Debilski/feature/fix-timeout-print	Rike-Benjamin Sch...	16. September 2014 12:25
0154d16	BF: Need to reset the bot_error status before each step.	Rike-Benjamin Sch...	16. September 2014 12:23
2aaa27c	BF: Do not make the default mode...	Rike-Benjamin Sch...	13. September 2014 17:12
9904e92	ENH: Adding a mysterious flag.	Rike-Benjamin Sch...	13. September 2014 17:12
8f7a235	NF: Double rainbow patch.	Rike-Benjamin Sch...	13. September 2014 17:12
01e5e96	BF: fix --speak mode	Zbigniew Jędrzeje...	13. September 2014 16:59
409ceee	Merge pull request #268 from ASPP/python3-again	Rike-Benjamin Sch...	13. September 2014 14:43
fe93ec7	BF: do not treat non-empty stderr as error	Zbigniew Jędrzeje...	13. September 2014 13:01
f1ff791	UT: simplify a bit more	Zbigniew Jędrzeje...	13. September 2014 12:49
4c281ca	ENH: use better subprocess invocation	Zbigniew Jędrzeje...	13. September 2014 12:47
0f7c305	UT: clean up argparse formatting	Zbigniew Jędrzeje...	13. September 2014 12:16
a86088f	ENH: create a new directory for each game	Zbigniew Jędrzeje...	13. September 2014 12:16
49a89b8	BF: fix argument passing	Zbigniew Jędrzeje...	13. September 2014 12:16
a61a105	UT: Small cleanup	Zbigniew Jędrzeje...	13. September 2014 10:11
c7cab81	ENH: add --interactive option	Zbigniew Jędrzeje...	12. September 2014 19:12
c150b78	BF: decode Popen output as utf-8	Zbigniew Jędrzeje...	12. September 2014 19:11
d9faca8	tournament: update location	Zbigniew Jędrzeje...	12. September 2014 18:41
ecfcfd5	Revert "ENH: Upgrading for Split. Downgrading for Python 2.7"	Zbigniew Jędrzeje...	12. September 2014 18:41
617de29	Merge pull request #267 from Debilski/feature/tournament-fixes	Rike-Benjamin Sch...	12. September 2014 17:57

Subject: **NF: Double rainbow patch.**  
Author: Rike-Benjamin Schuppner <rikebs@debilski.de>  
Date: Mon May 19 2014 15:30:45 GMT+0200 (CEST)

SHA: 8f7a2355cbe9e20221cc5032021034c1aa16a2ff  
Parent: 01e5e962dd742860c676fc1afd32eddb78dd7ff5

NF: Double rainbow patch.

- pelita/ui/tk\_canvas.py
- pelita/ui/tk\_sprites.py

```
... @@ -259,6 +259,12 @@ class UiCanvas(object):
259 259     self.current_universe = universe
260 260
261 261     if round is not None and turn is not None:
262 +         import random
263 +         r = random.randint(0, 15)
264 +         g = random.randint(0, 15)
265 +         b = random.randint(0, 15)
266 +         self.canvas.configure(background=col(r*16, g*16, b*16))
267 +
```

<http://rowanj.github.io/gitx/>

# Collaborating with Github

- In principle, no external service is needed for using git
- Teams can have a 'blessed' repo on a server and everybody with ssh/unix access can write to it
- As simple as it gets. But eg. notification hooks must be made manually
- Services such as Github/Bitbucket (\*) provide additional features for collaboration
  - (\* or self-hosted instances à la GitLab/Gitbucket)

# GitHub

- Individuals or teams own a repository on the server
- Collaborators *fork* the repository and submit a *Pull Request* (PR) when they have made changes
- In principle everybody with a GitHub account can suggest changes
- PRs can be discussed prior merging
- PRs will be merged with their complete history
- PR model is useful even if all members of a team have direct write access to main repo

# GitHub



This repository Search

Explore Gist Blog Help



Debilski



ASPP / pelita

Unwatch

6

Unstar

23

Fork

19

Issues

Pull requests

Labels

Milestones

Filters

is:open is:issue

New Issue

56 Open 86 Closed

Author

Labels

Milestones

Assignee

Sort

Crash on timeout  
#269 opened 23 days ago by khrapovs

3

Tk window does not update properly when using anaconda on OS X  
#265 opened on Aug 16 by Debilski

3

Upgrade to Bootstrap 3 and show more code on front page  
#264 opened on Jul 29 by Debilski

0

First I thought it was fun. But then I liked the coloured version so much it was incredible. OMG.  
#262 opened on May 22 by Debilski

1

Rewrite core in Elixir for the MMO version  
#257 opened on Oct 16, 2013 by Debilski

0

Revise tournament setup in documentation  
#254 opened on Sep 26, 2013 by Debilski

1

# Advanced GitHub

- GitHub provides integration with <https://travis-ci.org> for automated testing/compiling of projects
- This can make sure that your code also runs in a 'standard environment'
- You'll get notified when a build breaks the test suite
- In principle doable with plain git hooks – but usually no-one bothers

**Debilski/pelita** 

Actor-based Toolkit for Interactive Language Education in Python

build passing



# Help me, please

- Homepage: <http://git-scm.com>
- Reference to common commands: <http://gitref.org>
- Pro Git: <http://git-scm.com/book>  
(especially, if you're interested in the internals of git)
- `$ git help`

# Excuses for not using git

- **The other people in my group surely are too stupid to understand git, why should I bother with it, then?**
- **My collaborators always send me whole files by email, so git does not work with me.**

# **No** excuses for not using git

- **The other people in my group surely are too stupid to understand git, why should I bother with it, then?**
- **My collaborators always send me whole files by email, so git does not work with me.**
- **There is nothing stopping you from using git yourself and let other people do their work manually.**