Aalto University
Media Factory

Digital_Fabrication_Studio.02
Version Control Systems – Git and GitHub

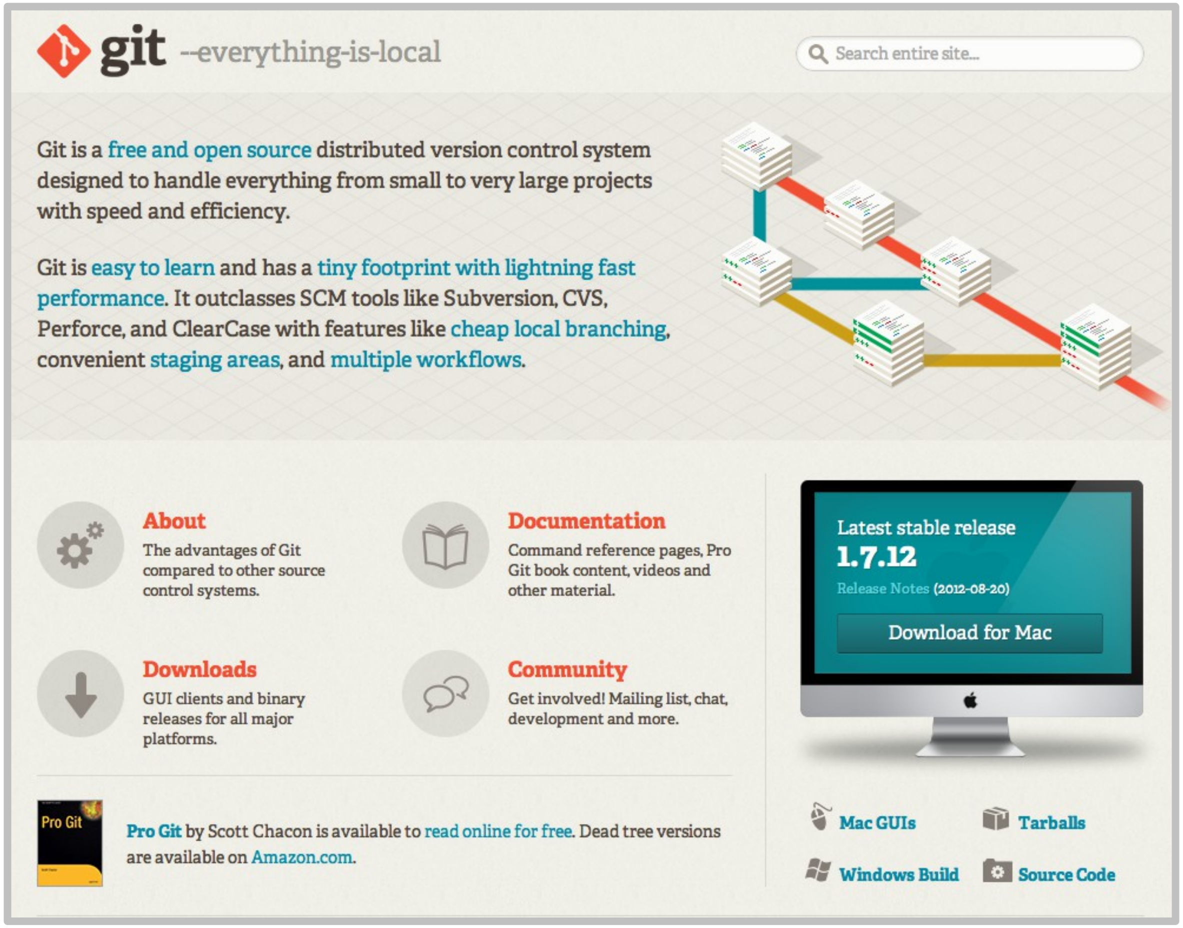
Massimo Menichinelli
massimo.menichinelli@aalto.fi
@openp2pdesign
http://www.slideshare.net/openp2pdesign

Today:

* Git
* GitHub
* Exercise

01.
Git – the distributed architectural engine

Git: a distributed version control system



Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

About
The advantages of Git compared to other source control systems.

Documentation
Command reference pages, Pro Git book content, videos and other material.

Downloads
GUI clients and binary releases for all major platforms.

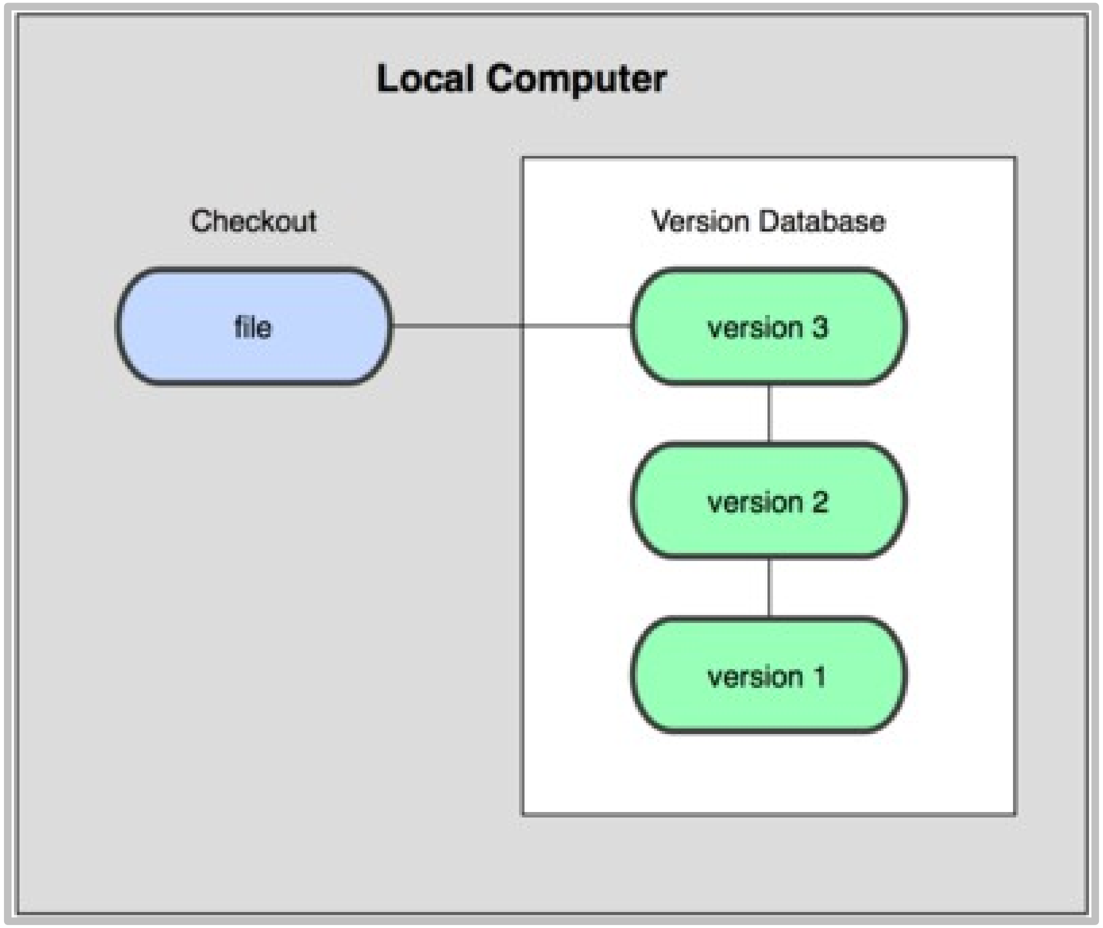
Community
Get involved! Mailing list, chat, development and more.

Latest stable release
1.7.12
Release Notes (2012-08-20)

Pro Git by Scott Chacon is available to read online for free. Dead tree versions are available on Amazon.com.

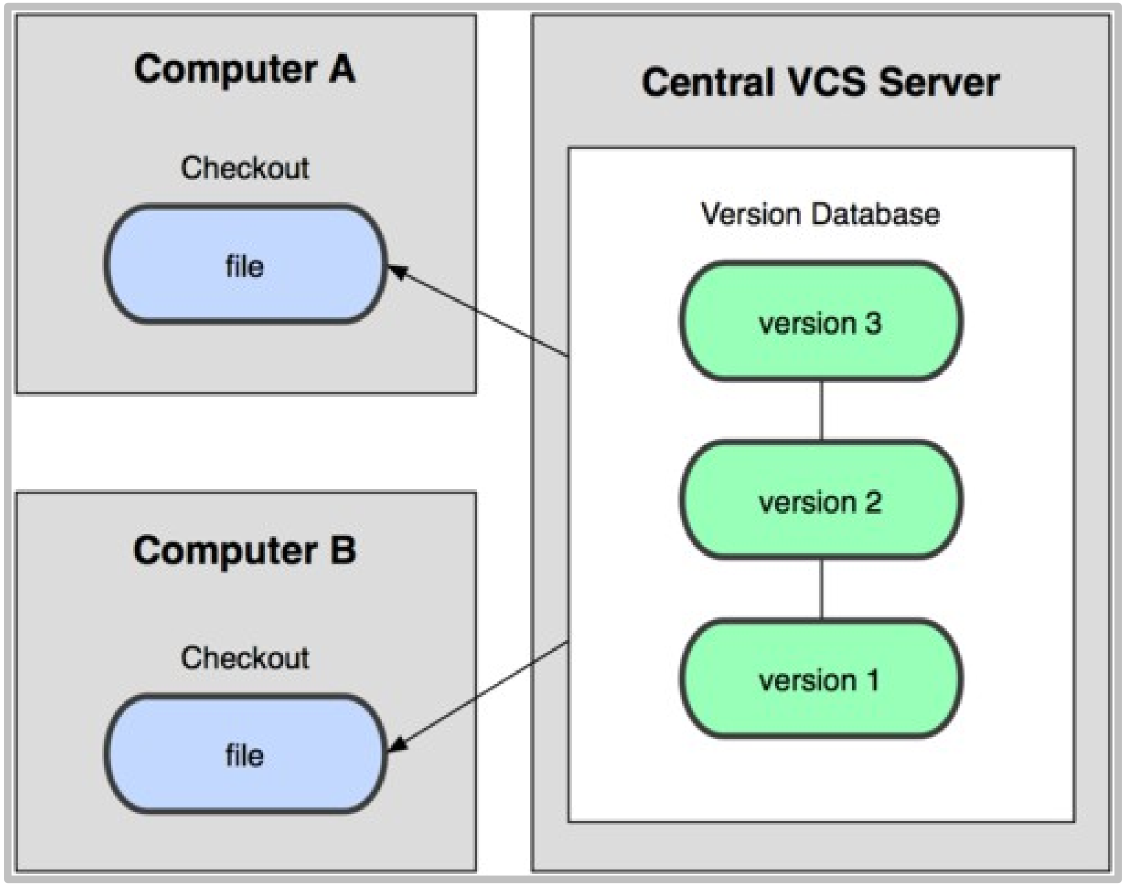
Source: http://git-scm.com/

A local version control system



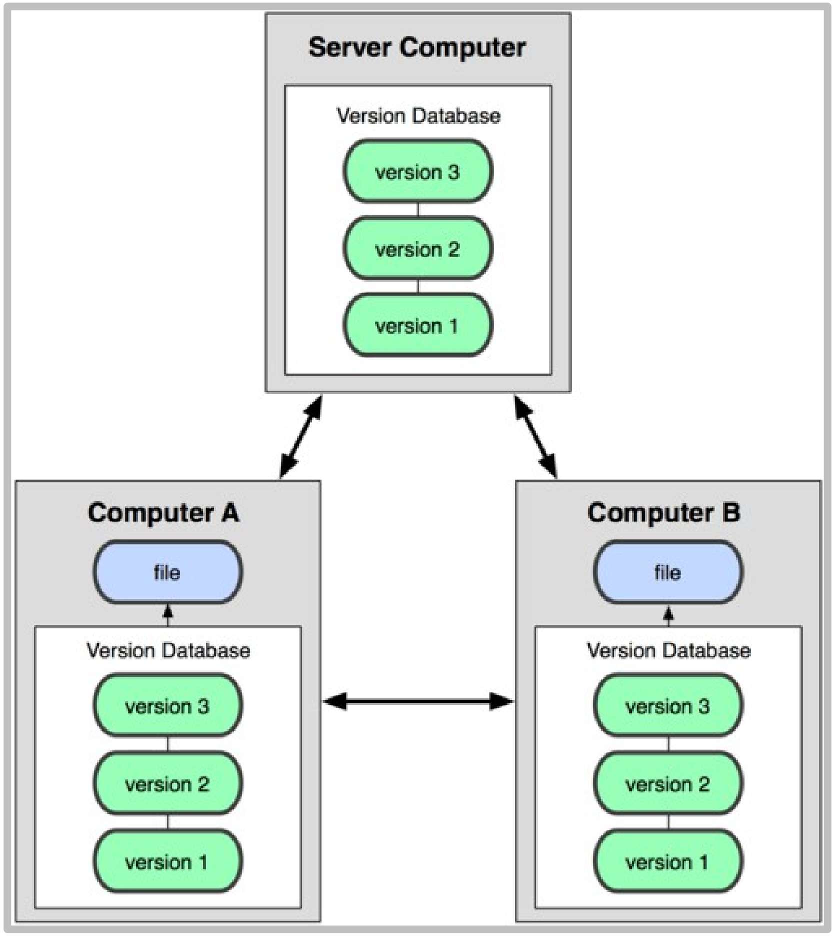
Source: http://git-scm.com/book/en/Getting-Started-About-Version-Control

A centralized version control system



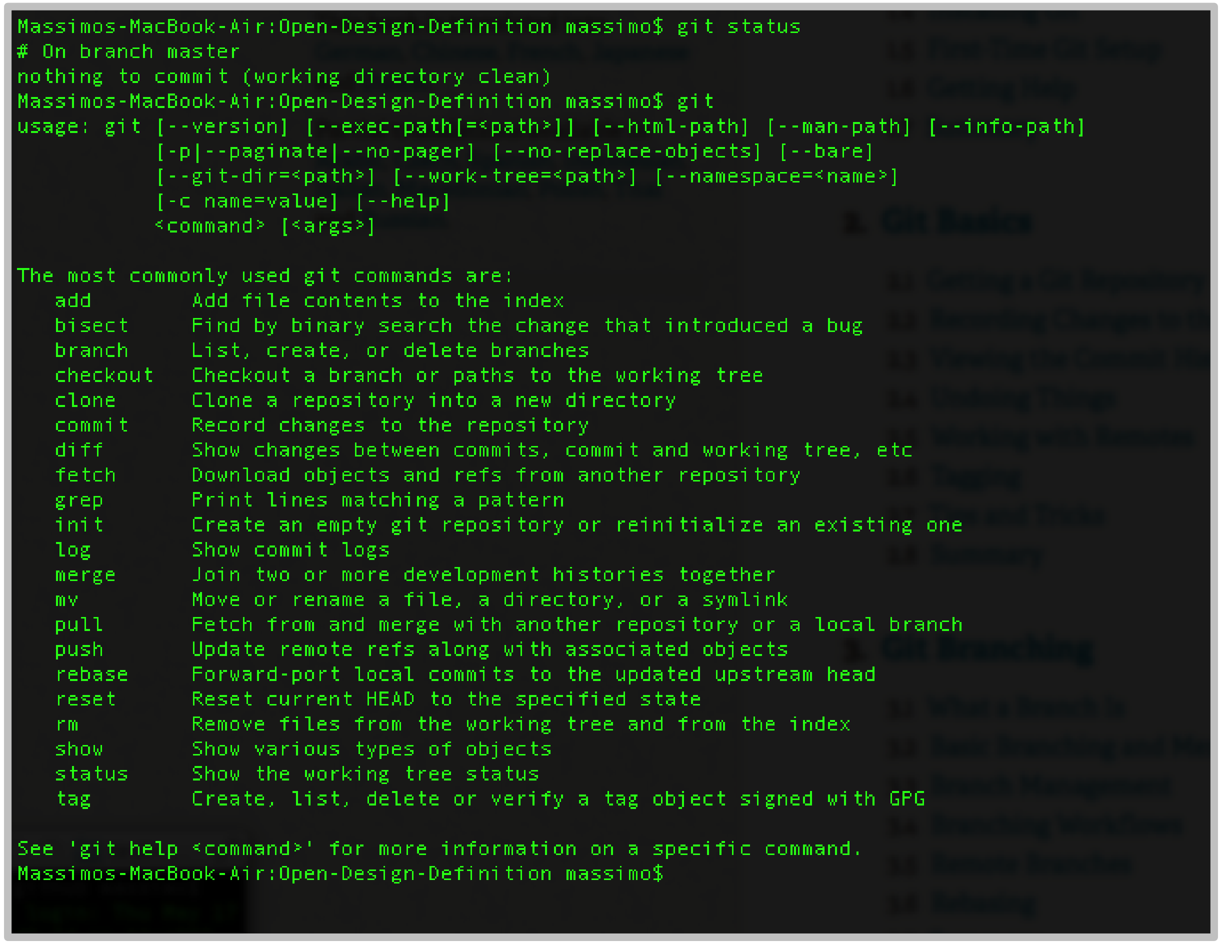
Source: http://git-scm.com/book/en/Getting-Started-About-Version-Control

Git: a distributed version control system



Source: http://git-scm.com/book/en/Getting-Started-About-Version-Control

Git: a command line tool

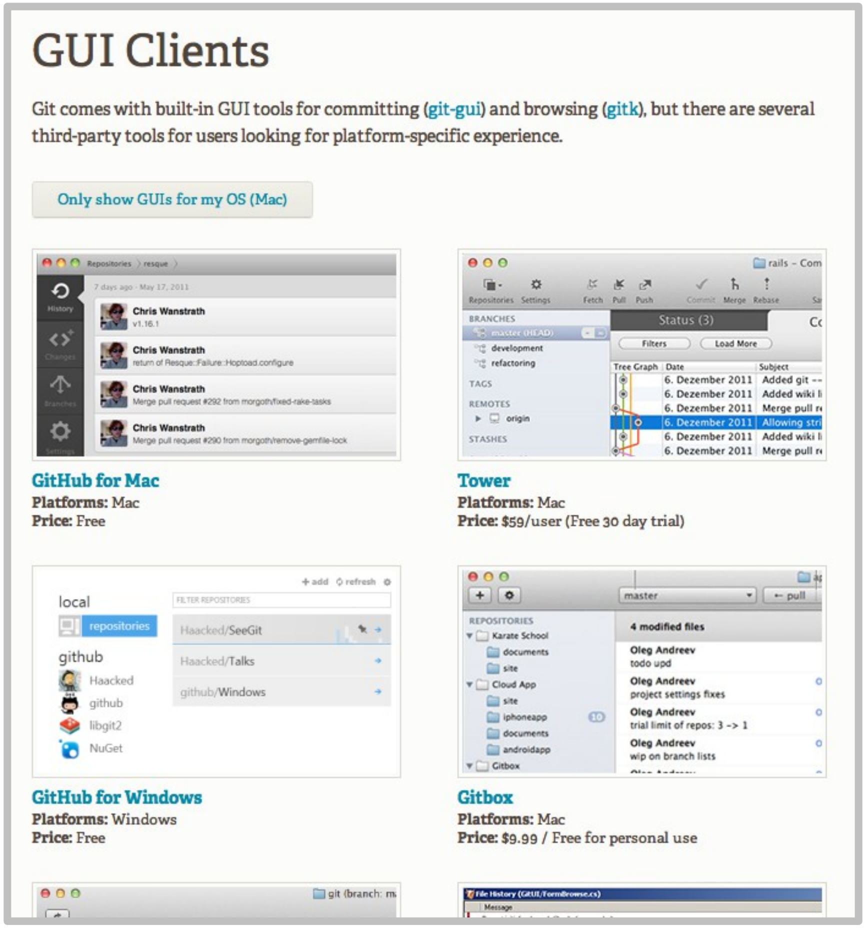


```
Massimos-MacBook-Air:Open-Design-Definition massimo$ git status
# On branch master
nothing to commit (working directory clean)
Massimos-MacBook-Air:Open-Design-Definition massimo$ git
usage: git [--version] [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p|--paginate|--no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           [-c name=value] [--help]
           <command> [<args>]

The most commonly used git commands are:
   add        Add file contents to the index
   bisect     Find by binary search the change that introduced a bug
   branch     List, create, or delete branches
   checkout   Checkout a branch or paths to the working tree
   clone      Clone a repository into a new directory
   commit     Record changes to the repository
   diff       Show changes between commits, commit and working tree, etc
   fetch      Download objects and refs from another repository
   grep       Print lines matching a pattern
   init       Create an empty git repository or reinitialize an existing one
   log        Show commit logs
   merge      Join two or more development histories together
   mv         Move or rename a file, a directory, or a symlink
   pull       Fetch from and merge with another repository or a local branch
   push       Update remote refs along with associated objects
   rebase     Forward-port local commits to the updated upstream head
   reset      Reset current HEAD to the specified state
   rm         Remove files from the working tree and from the index
   show       Show various types of objects
   status     Show the working tree status
   tag        Create, list, delete or verify a tag object signed with GPG

See 'git help <command>' for more information on a specific command.
Massimos-MacBook-Air:Open-Design-Definition massimo$
```

Git: GUIs also available



GUI Clients

Git comes with built-in GUI tools for committing (git-gui) and browsing (gitk), but there are several third-party tools for users looking for platform-specific experience.

Only show GUIs for my OS (Mac)

GitHub for Mac
Platforms: Mac
Price: Free

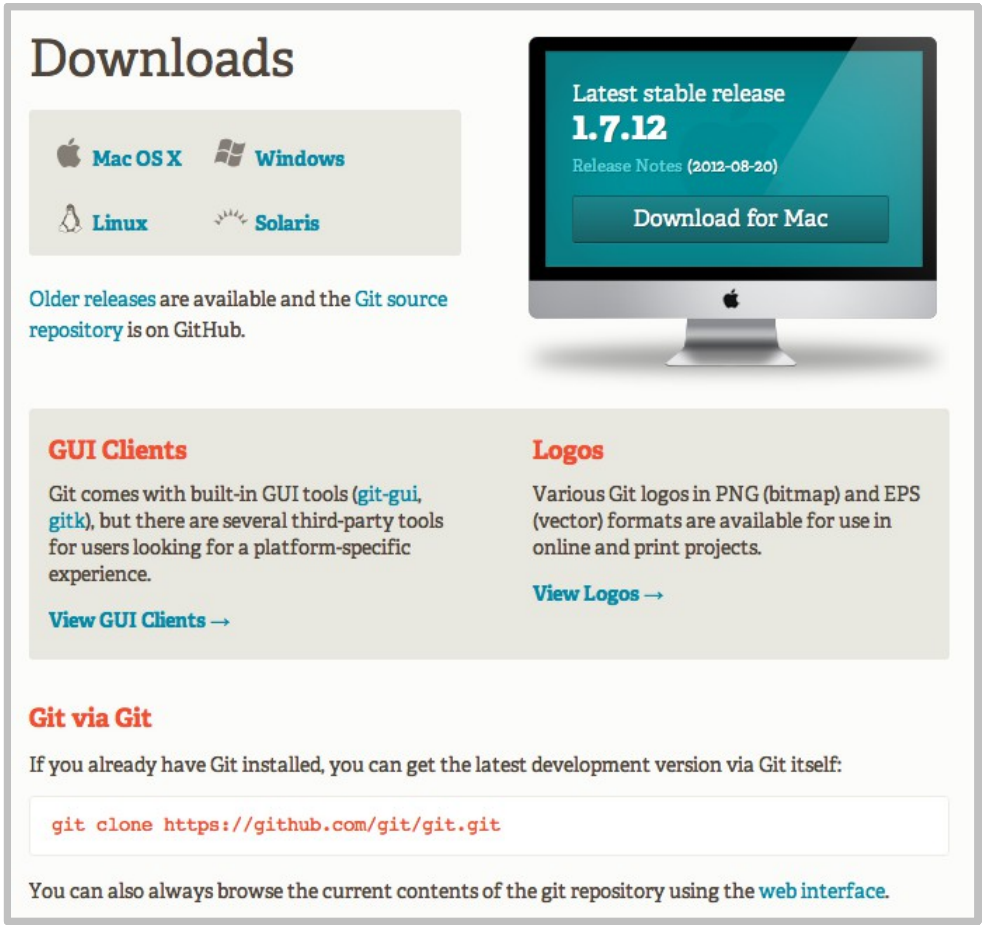
Tower
Platforms: Mac
Price: $59/user (Free 30 day trial)

GitHub for Windows
Platforms: Windows
Price: Free

Gitbox
Platforms: Mac
Price: $9.99 / Free for personal use

Source: http://git-scm.com/downloads/guis

01: Download and install Git



Downloads

Older releases are available and the Git source repository is on GitHub.

Latest stable release
1.7.12
Release Notes (2012-08-20)

GUI Clients

Git comes with built-in GUI tools (git-gui, gitk), but there are several third-party tools for users looking for a platform-specific experience.

View GUI Clients →

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

View Logos →

Git via Git

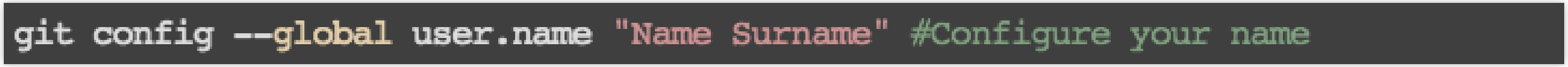
If you already have Git installed, you can get the latest development version via Git itself:

```
git clone https://github.com/git/git.git
```

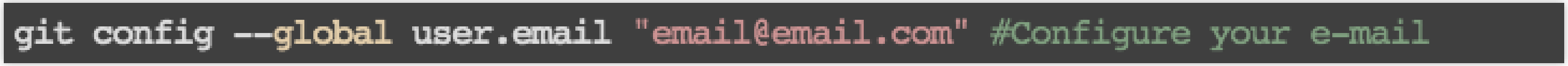
You can also always browse the current contents of the git repository using the web interface.

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

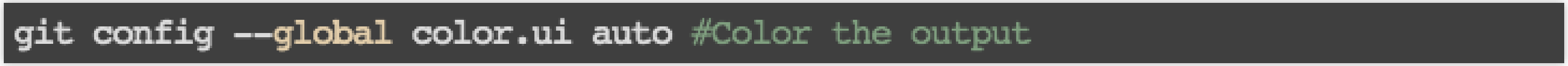
01: Install Git



```
git config --global user.name "Name Surname" #Configure your name
```

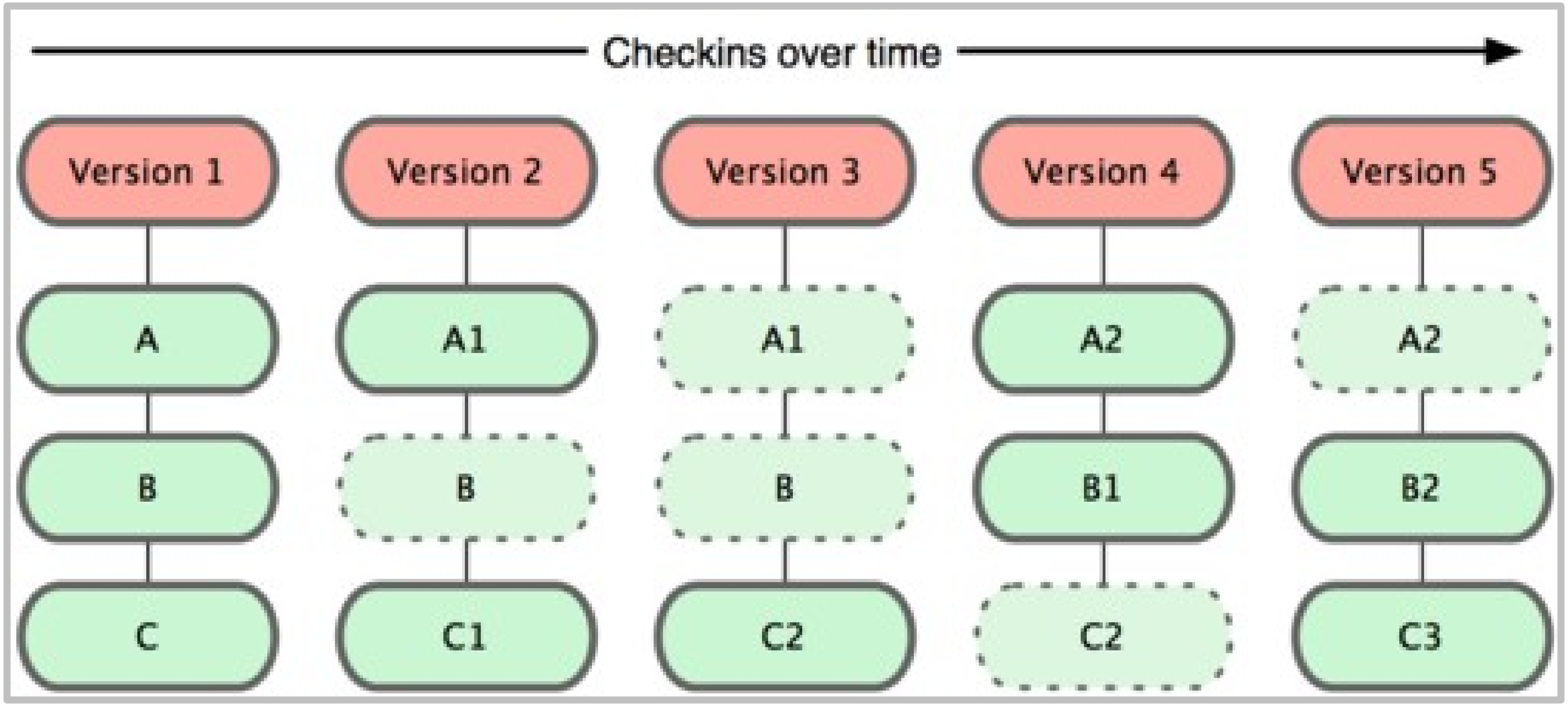


```
git config --global user.email "email@email.com" #Configure your e-mail
```



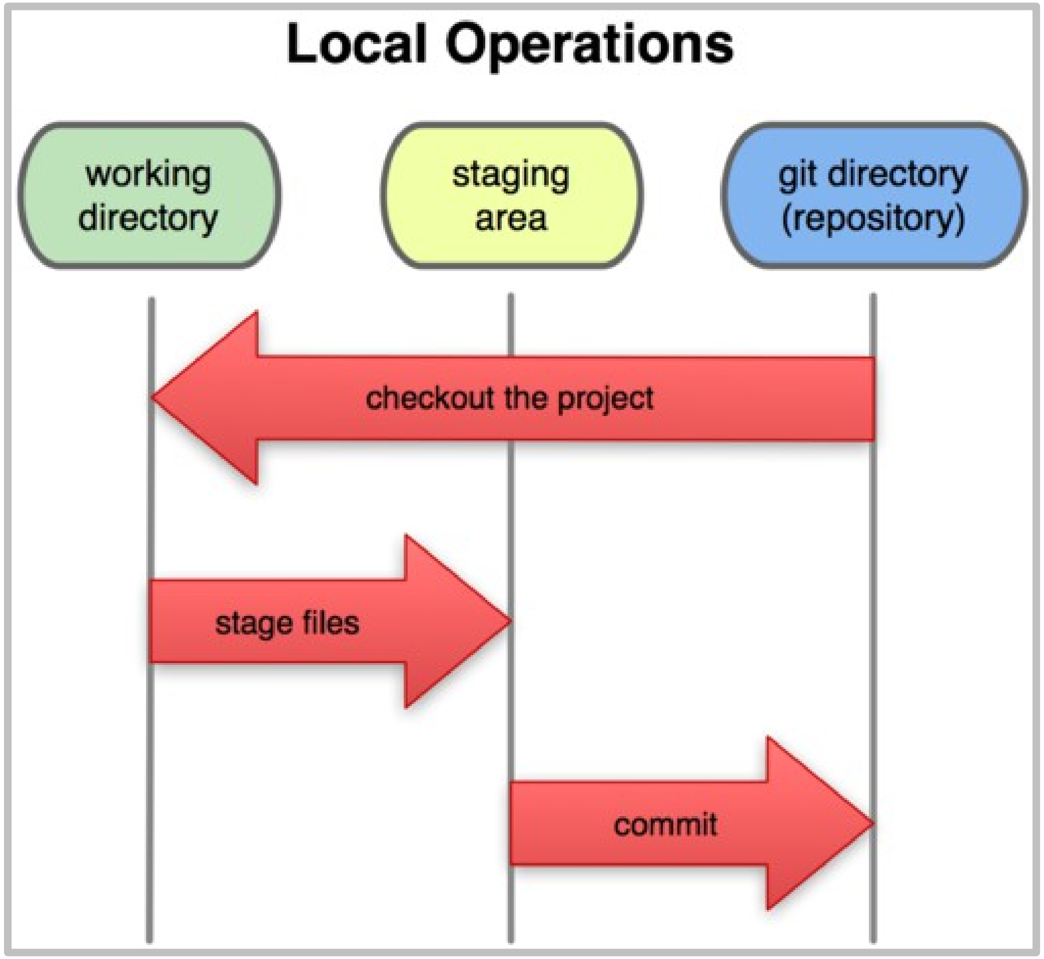
```
git config --global color.ui auto #Color the output
```

Git: history of a file system



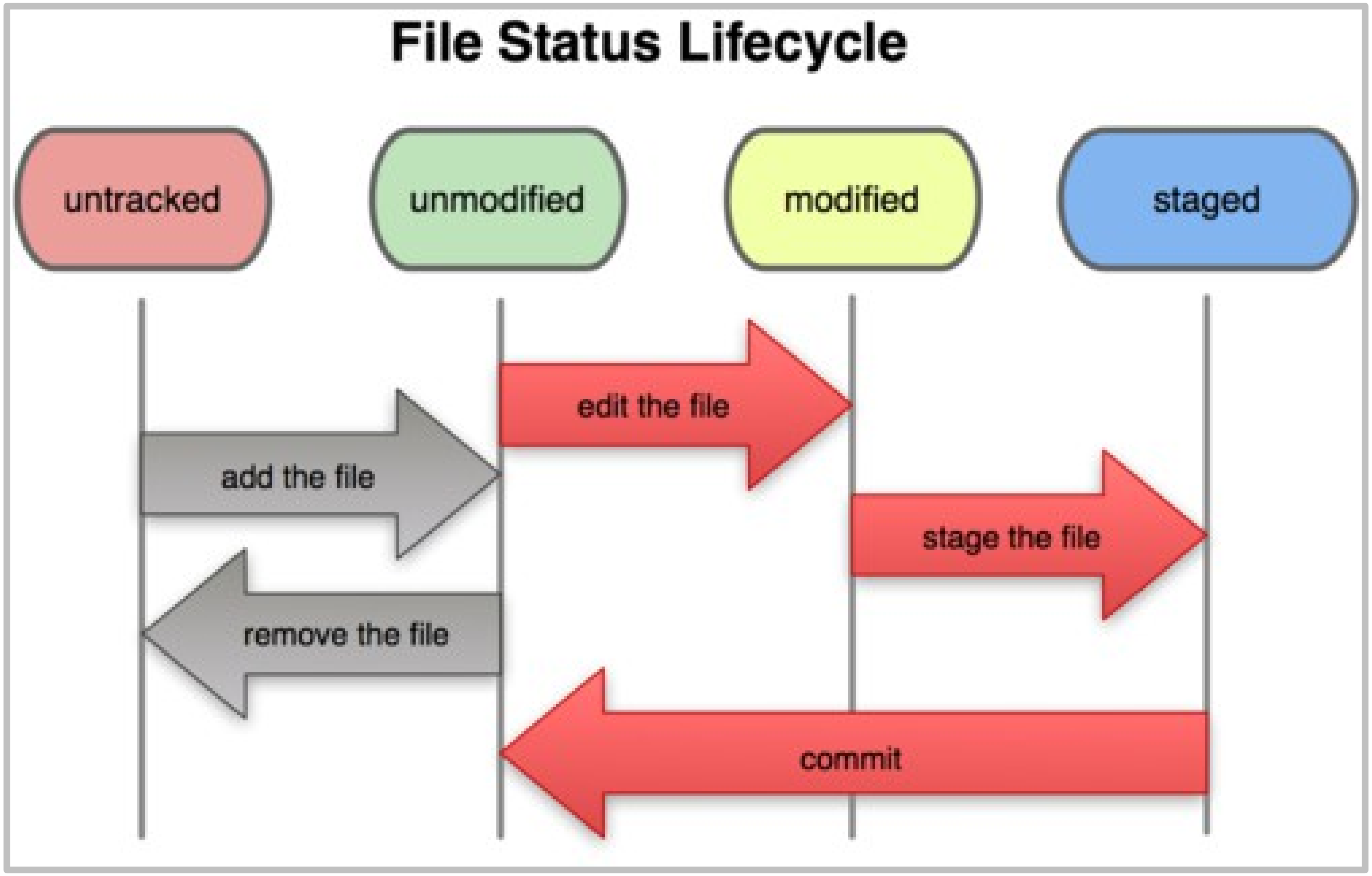
Source: http://git-scm.com/book/en/Getting-Started-Git-Basics

Git: architecture



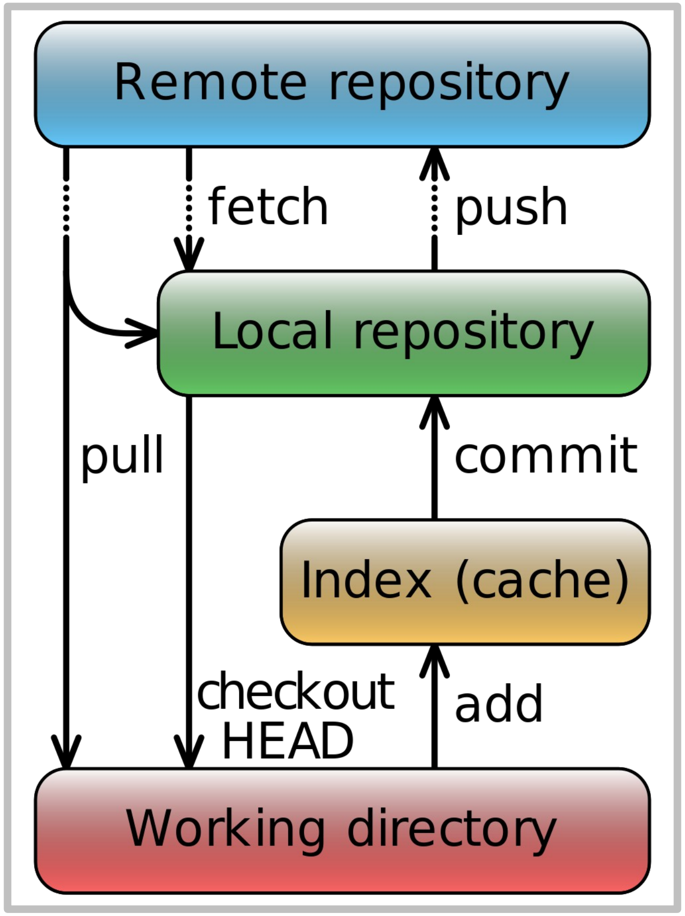
Source: http://git-scm.com/book/en/Getting-Started-Git-Basics

Git: history of a file



Source: http://git-scm.com/book/en/Git-Basics-Recording-Changes-to-the-Repository

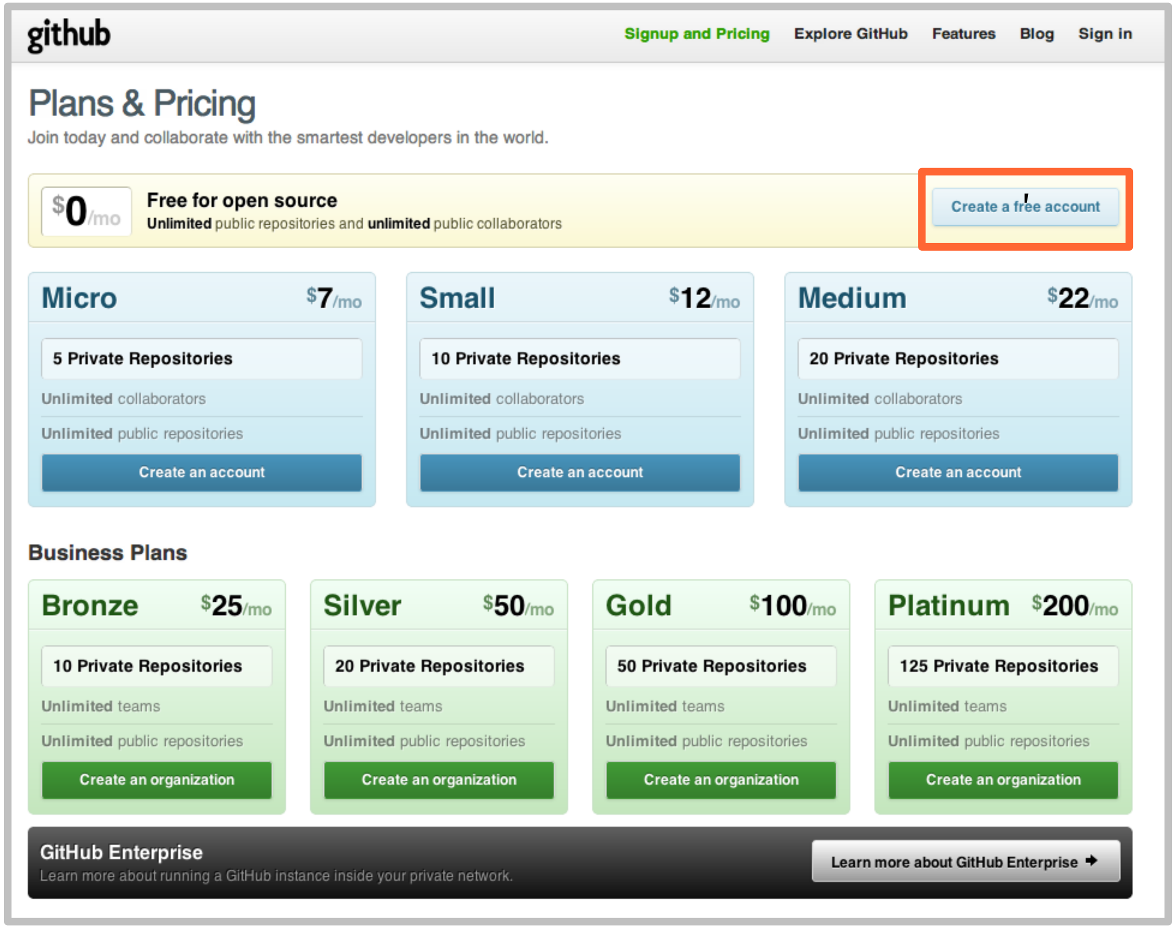
Git: commands and architecture



Source: http://en.wikipedia.org/wiki/Git_%28software%29

02.
GitHub – the easy online interface to a repository

02: Create a user on GitHub (free plan / student plan)



github

Signup and Pricing | Explore GitHub | Features | Blog | Sign in

Plans & Pricing

Join today and collaborate with the smartest developers in the world.

$0/mo **Free for open source**
Unlimited public repositories and **unlimited** public collaborators

Create a free account

Micro $7/mo
5 Private Repositories
Unlimited collaborators
Unlimited public repositories
Create an account

Small $12/mo
10 Private Repositories
Unlimited collaborators
Unlimited public repositories
Create an account

Medium $22/mo
20 Private Repositories
Unlimited collaborators
Unlimited public repositories
Create an account

Business Plans

Bronze $25/mo
10 Private Repositories
Unlimited teams
Unlimited public repositories
Create an organization

Silver $50/mo
20 Private Repositories
Unlimited teams
Unlimited public repositories
Create an organization

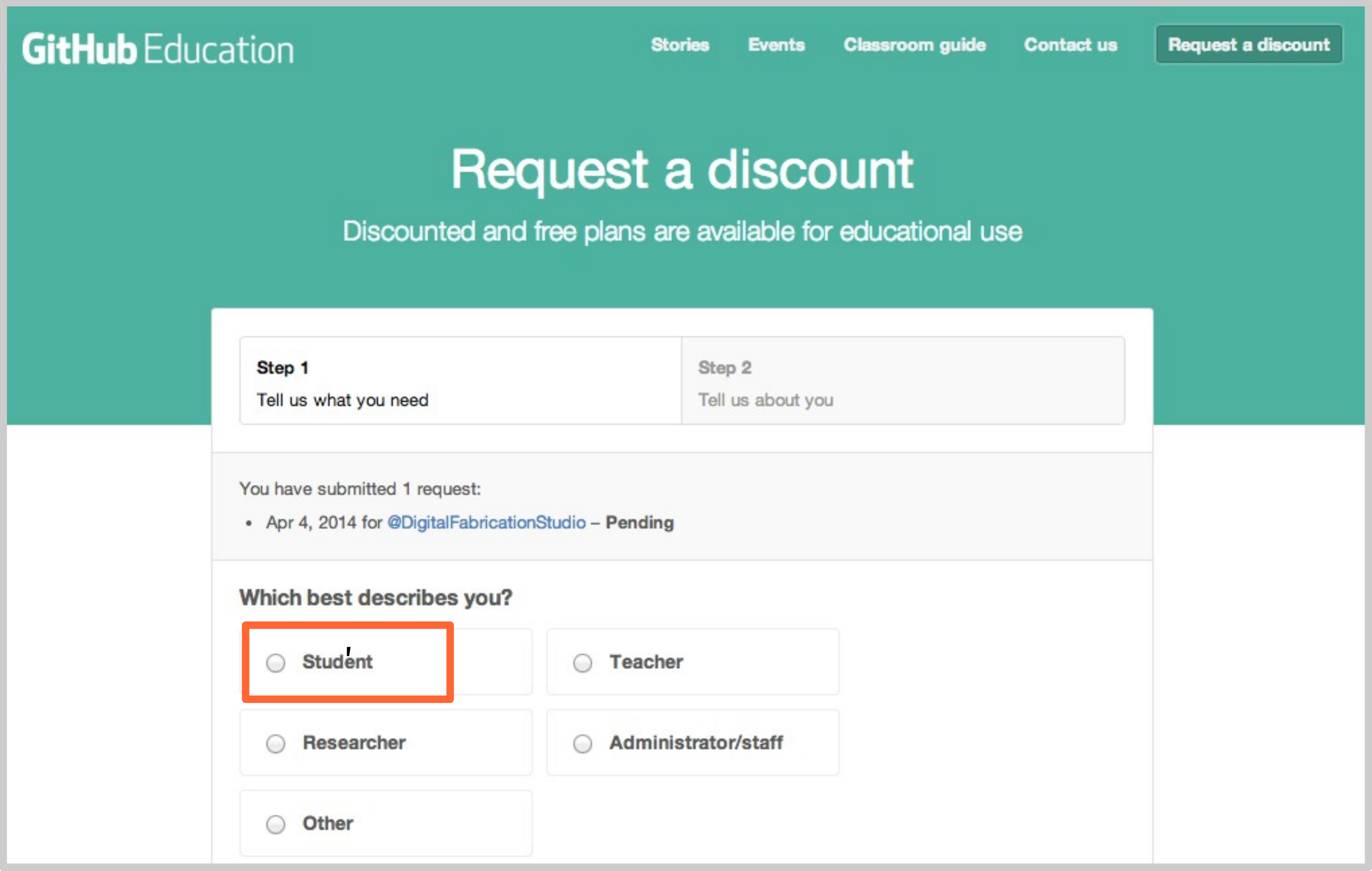
Gold $100/mo
50 Private Repositories
Unlimited teams
Unlimited public repositories
Create an organization

Platinum $200/mo
125 Private Repositories
Unlimited teams
Unlimited public repositories
Create an organization

GitHub Enterprise
Learn more about running a GitHub instance inside your private network.
Learn more about GitHub Enterprise ➔

Source: https://github.com/plans

02: Create a user on GitHub (free plan / student plan)



Request a discount

Discounted and free plans are available for educational use

Step 1
Tell us what you need

Step 2
Tell us about you

You have submitted 1 request:

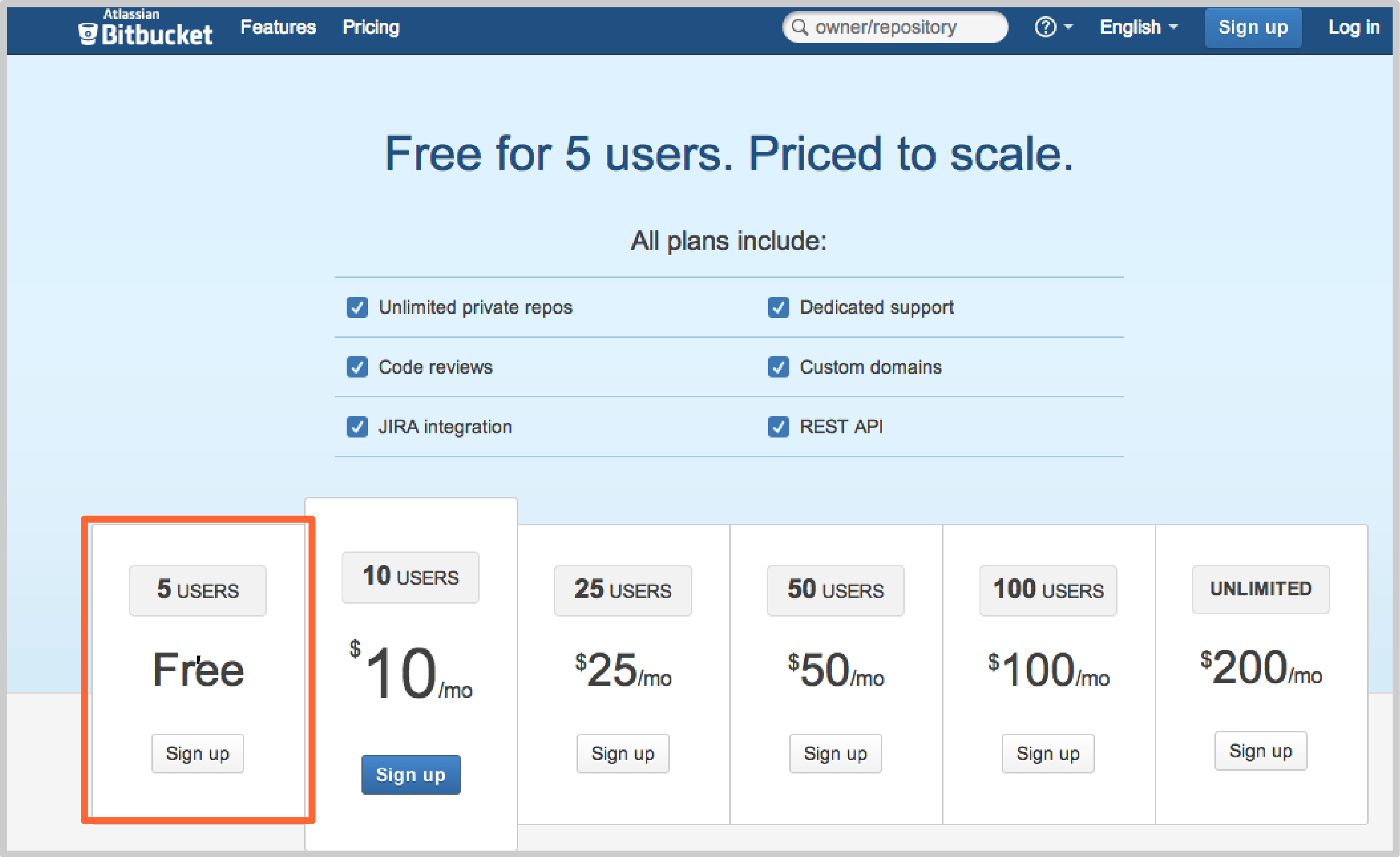
- Apr 4, 2014 for @DigitalFabricationStudio – **Pending**

Which best describes you?

○ Student
○ Teacher
○ Researcher
○ Administrator/staff
○ Other

Source: https://education.github.com/discount_requests/new

02: or Create a user on BitBucket (free plan)



Free for 5 users. Priced to scale.

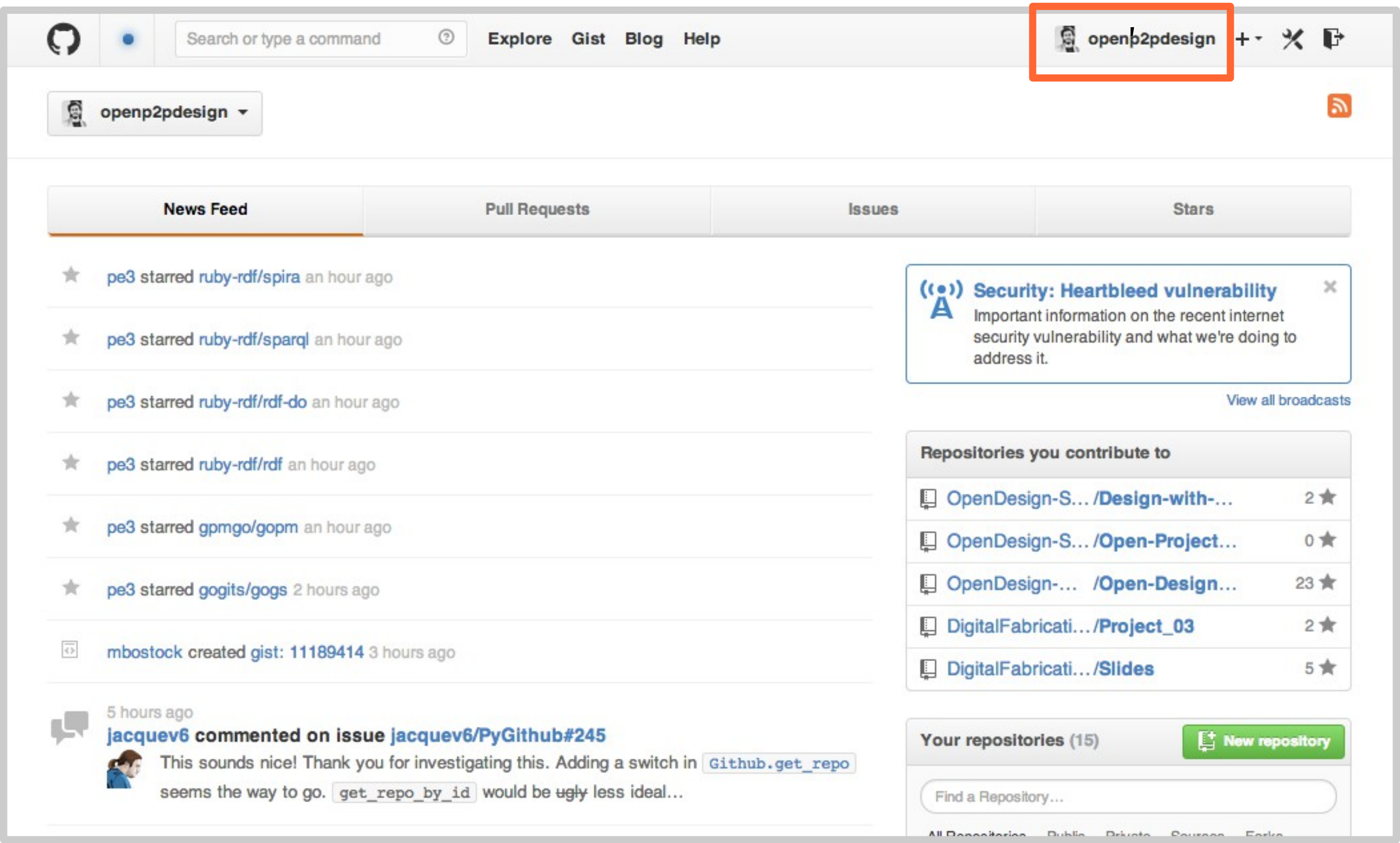
All plans include:

- ☑ Unlimited private repos
- ☑ Code reviews
- ☑ JIRA integration
- ☑ Dedicated support
- ☑ Custom domains
- ☑ REST API

5 USERS	10 USERS	25 USERS	50 USERS	100 USERS	UNLIMITED
Free	$10/mo	$25/mo	$50/mo	$100/mo	$200/mo
Sign up	Sign up	Sign up	Sign up	Sign up	Sign up

Source: https://bitbucket.org/plans

03: Send your username to massimo.menichinelli@aalto.fi



pe3 starred ruby-rdf/spira an hour ago

pe3 starred ruby-rdf/sparql an hour ago

pe3 starred ruby-rdf/rdf-do an hour ago

pe3 starred ruby-rdf/rdf an hour ago

pe3 starred gpmgo/gopm an hour ago

pe3 starred gogits/gogs 2 hours ago

mbostock created gist: 11189414 3 hours ago

5 hours ago
jacquev6 commented on issue jacquev6/PyGithub#245
This sounds nice! Thank you for investigating this. Adding a switch in `Github.get_repo` seems the way to go. `get_repo_by_id` would be ~~ugly~~ less ideal…

Security: Heartbleed vulnerability
Important information on the recent internet security vulnerability and what we're doing to address it.

View all broadcasts

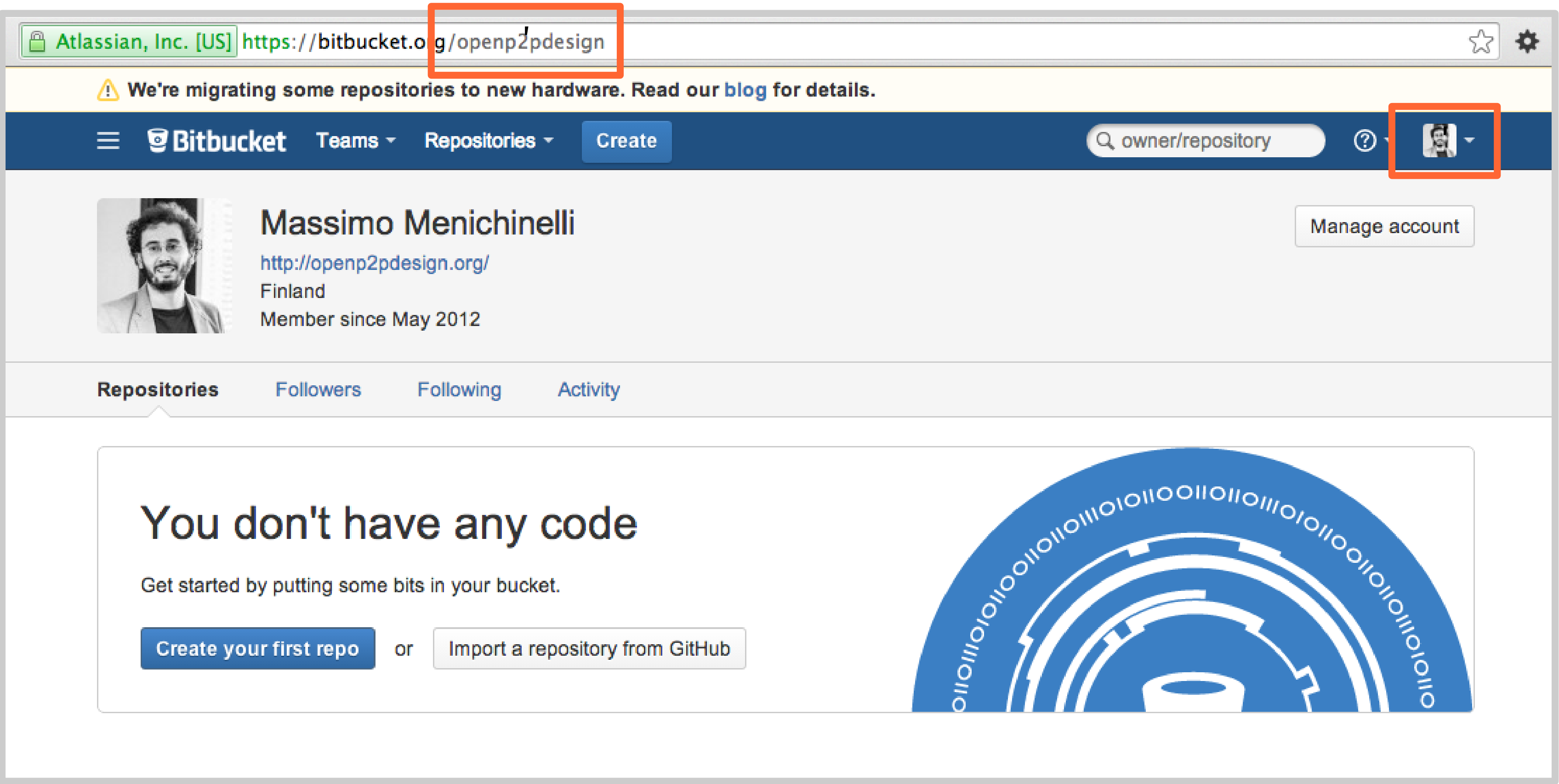
Repositories you contribute to

OpenDesign-S… /**Design-with-…**	2 ★
OpenDesign-S… /**Open-Project…**	0 ★
OpenDesign-… /**Open-Design…**	23 ★
DigitalFabricati… /**Project_03**	2 ★
DigitalFabricati… /**Slides**	5 ★

Your repositories (15)

Source: http://www.github.com

03: Send your username to massimo.menichinelli@aalto.fi



Atlassian, Inc. [US] https://bitbucket.org/openp2pdesign

We're migrating some repositories to new hardware. Read our blog for details.

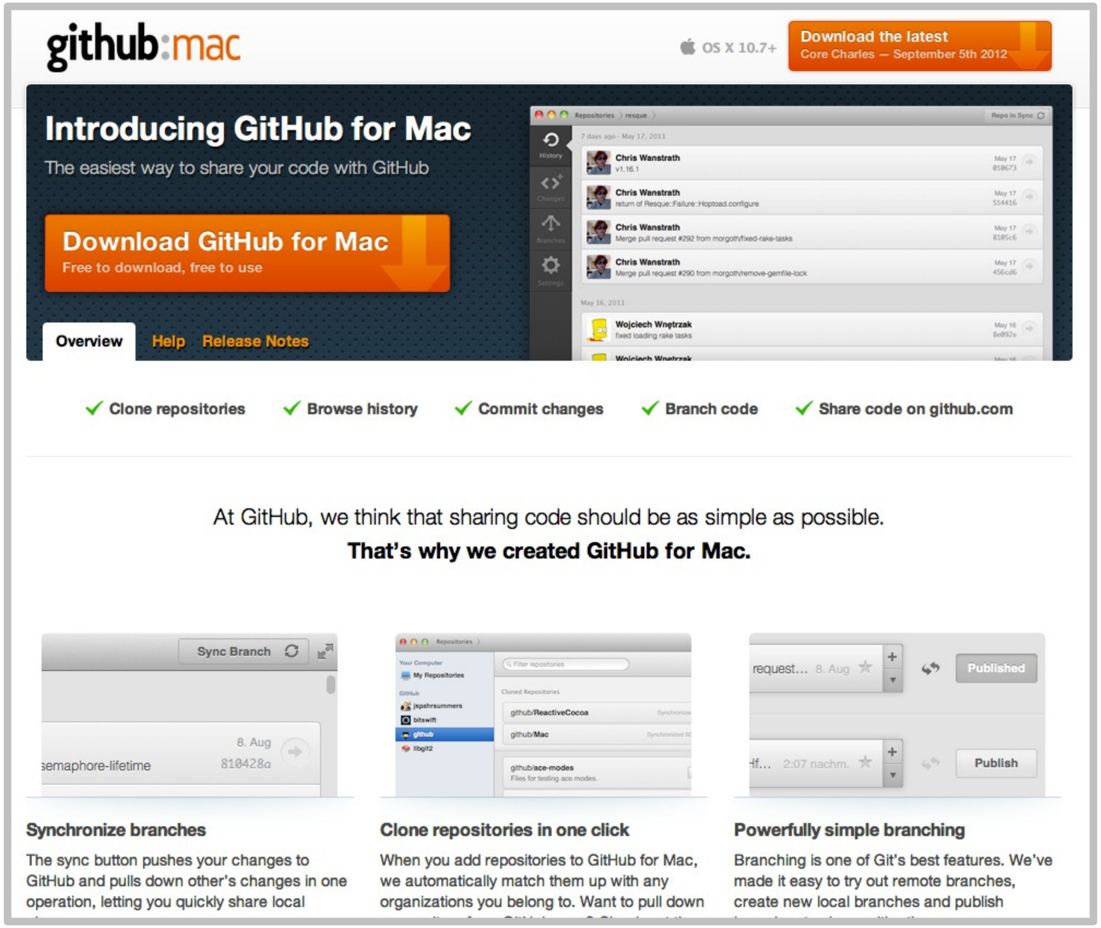
Massimo Menichinelli
http://openp2pdesign.org/
Finland
Member since May 2012

You don't have any code

Get started by putting some bits in your bucket.

Source: https://bitbucket.org/

04. Download a client: an easy client for Mac



github:mac

OS X 10.7+

Download the latest
Core Charles — September 5th 2012

Introducing GitHub for Mac

The easiest way to share your code with GitHub

Download GitHub for Mac
Free to download, free to use

Overview Help Release Notes

✓ Clone repositories ✓ Browse history ✓ Commit changes ✓ Branch code ✓ Share code on github.com

At GitHub, we think that sharing code should be as simple as possible.
That's why we created GitHub for Mac.

Synchronize branches

The sync button pushes your changes to GitHub and pulls down other's changes in one operation, letting you quickly share local

Clone repositories in one click

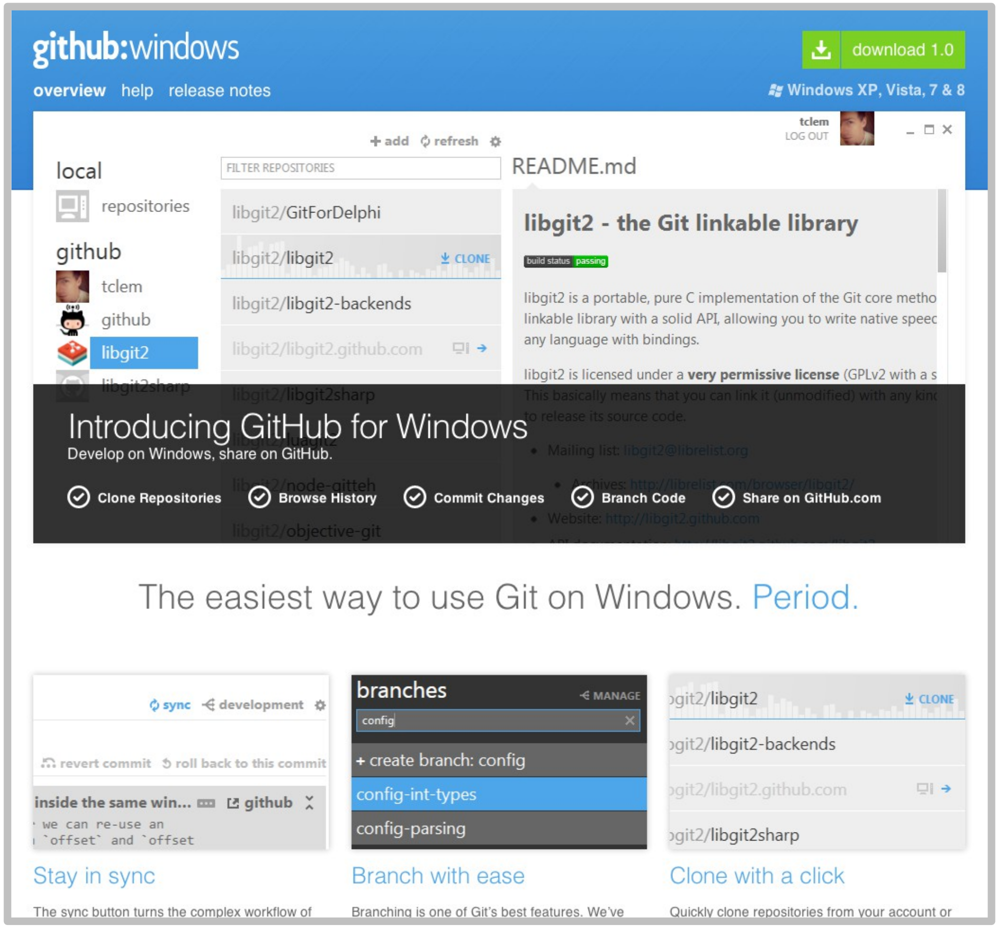
When you add repositories to GitHub for Mac, we automatically match them up with any organizations you belong to. Want to pull down

Powerfully simple branching

Branching is one of Git's best features. We've made it easy to try out remote branches, create new local branches and publish

Source: http://mac.github.com/

04. Download a client: an easy client for Windows



libgit2 - the Git linkable library

libgit2 is a portable, pure C implementation of the Git core metho linkable library with a solid API, allowing you to write native speec any language with bindings.

libgit2 is licensed under a **very permissive license** (GPLv2 with a s This basically means that you can link it (unmodified) with any kind to release its source code.

- Mailing list: libgit2@librelist.org
  - Archives: http://librelist.com/browser/libgit2/
- Website: http://libgit2.github.com

Introducing GitHub for Windows

Develop on Windows, share on GitHub.

The easiest way to use Git on Windows. Period.

```
we can re-use an
`offset` and `offset
```

Stay in sync

The sync button turns the complex workflow of

Branch with ease

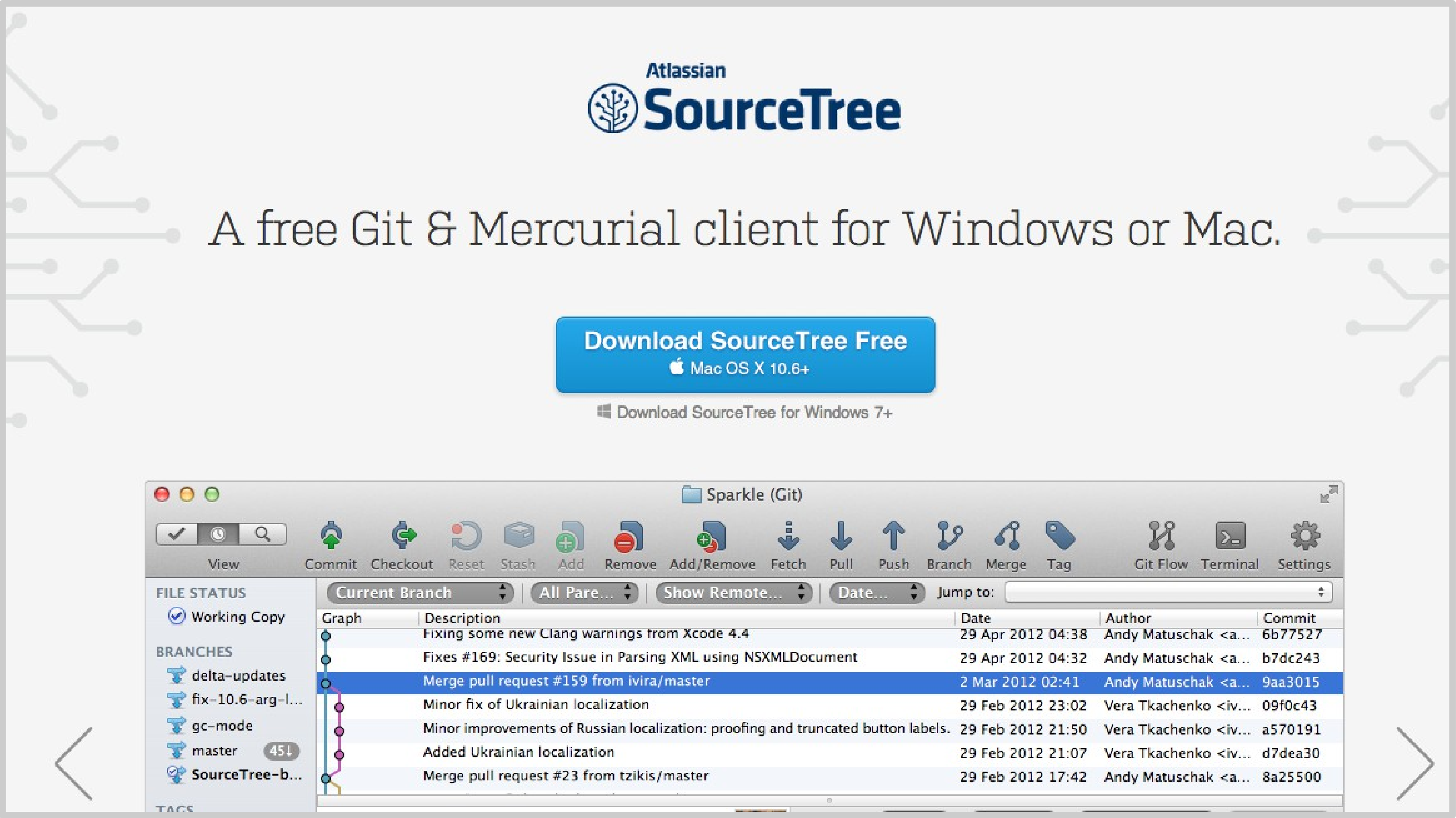
Branching is one of Git's best features. We've

Clone with a click

Quickly clone repositories from your account or

Source: http://windows.github.com/

04. Download a client: SourceTree



Atlassian SourceTree

A free Git & Mercurial client for Windows or Mac.

Download SourceTree Free
Mac OS X 10.6+

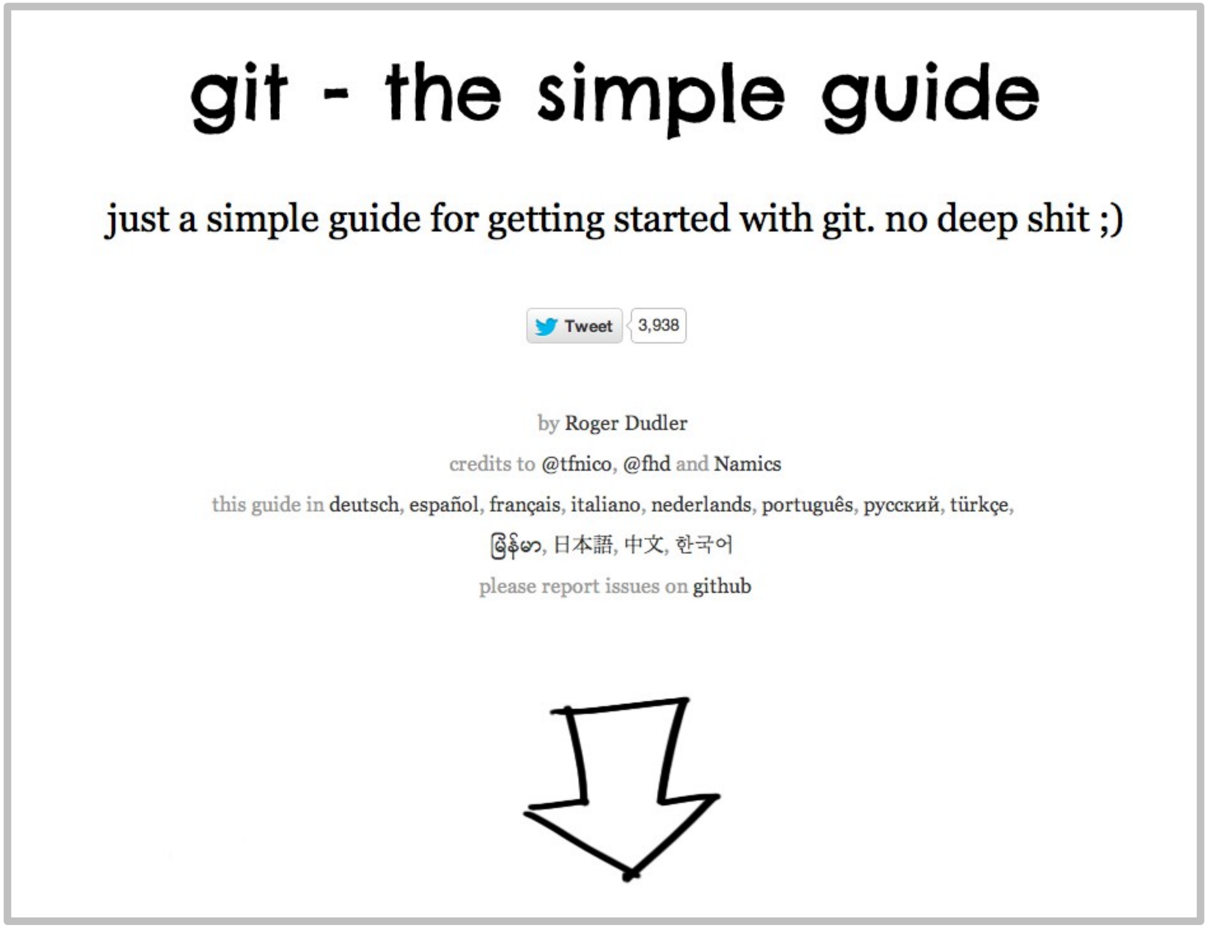
Download SourceTree for Windows 7+

Sparkle (Git)

Graph	Description	Date	Author	Commit
	Fixing some new Clang warnings from Xcode 4.4	29 Apr 2012 04:38	Andy Matuschak <a...	6b77527
	Fixes #169: Security Issue in Parsing XML using NSXMLDocument	29 Apr 2012 04:32	Andy Matuschak <a...	b7dc243
	Merge pull request #159 from ivira/master	2 Mar 2012 02:41	Andy Matuschak <a...	9aa3015
	Minor fix of Ukrainian localization	29 Feb 2012 23:02	Vera Tkachenko <iv...	09f0c43
	Minor improvements of Russian localization: proofing and truncated button labels.	29 Feb 2012 21:50	Vera Tkachenko <iv...	a570191
	Added Ukrainian localization	29 Feb 2012 21:07	Vera Tkachenko <iv...	d7dea30
	Merge pull request #23 from tzikis/master	29 Feb 2012 17:42	Andy Matuschak <a...	8a25500

Source: http://www.sourcetreeapp.com/

Git, a simple guide…



git - the simple guide

just a simple guide for getting started with git. no deep shit ;)

by Roger Dudler

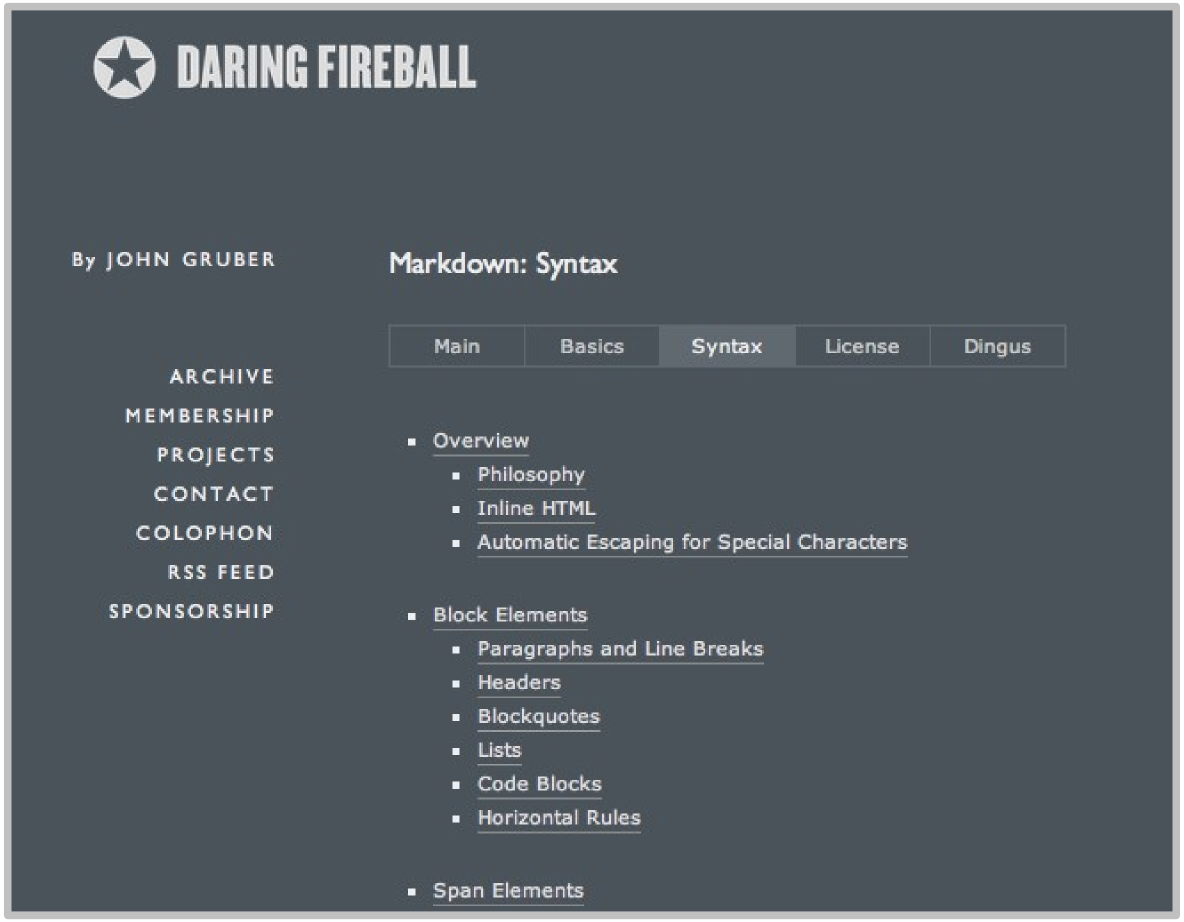
credits to @tfnico, @fhd and Namics

this guide in deutsch, español, français, italiano, nederlands, português, русский, türkçe,
မြန်မာ, 日本語, 中文, 한국어

please report issues on github

Source: http://rogerdudler.github.com/git-guide/

05. Learn Markdown



DARING FIREBALL

By JOHN GRUBER

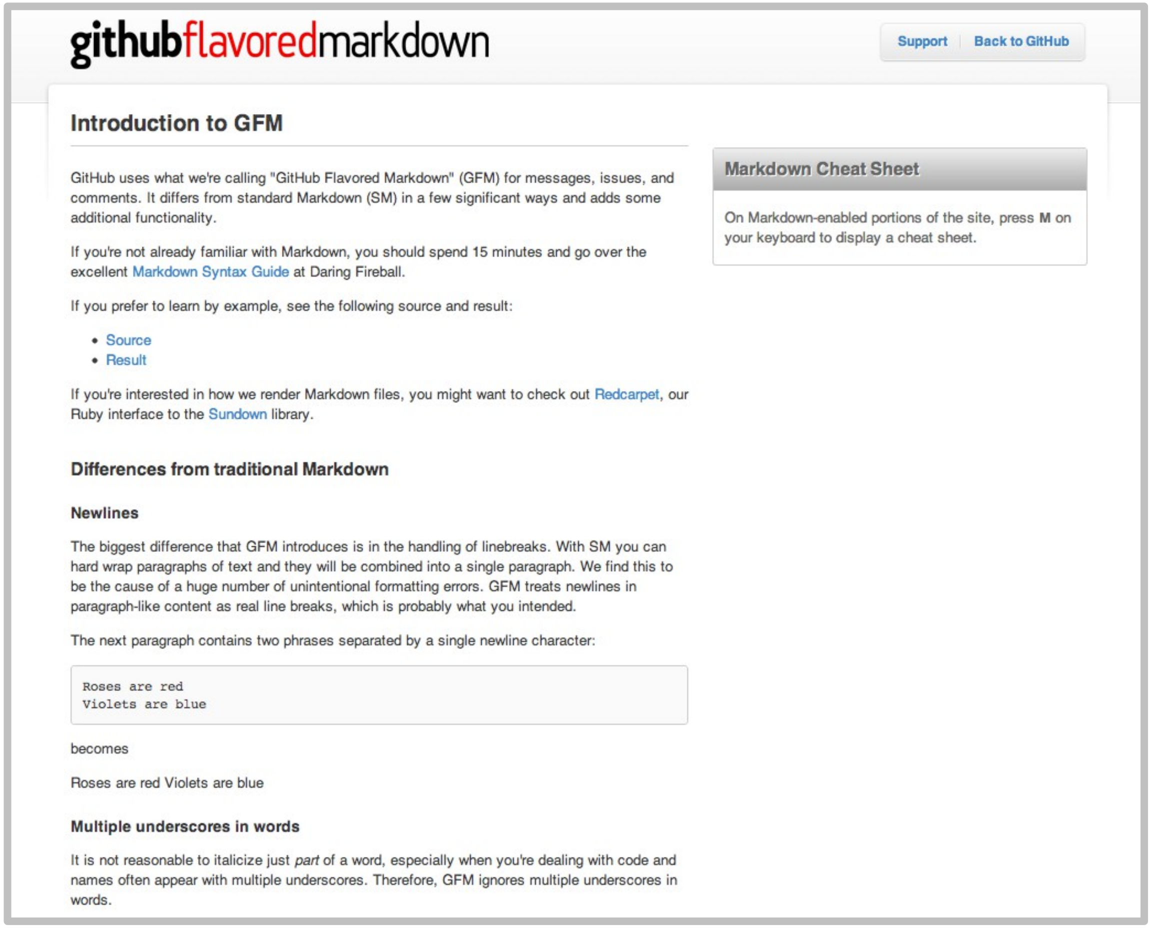
ARCHIVE
MEMBERSHIP
PROJECTS
CONTACT
COLOPHON
RSS FEED
SPONSORSHIP

Markdown: Syntax

- Overview
  - Philosophy
  - Inline HTML
  - Automatic Escaping for Special Characters
- Block Elements
  - Paragraphs and Line Breaks
  - Headers
  - Blockquotes
  - Lists
  - Code Blocks
  - Horizontal Rules
- Span Elements

Source: http://daringfireball.net/projects/markdown/syntax
http://en.wikipedia.org/wiki/Markdown

05. Learn GitHub Markdown



Introduction to GFM

GitHub uses what we're calling "GitHub Flavored Markdown" (GFM) for messages, issues, and comments. It differs from standard Markdown (SM) in a few significant ways and adds some additional functionality.

If you're not already familiar with Markdown, you should spend 15 minutes and go over the excellent Markdown Syntax Guide at Daring Fireball.

If you prefer to learn by example, see the following source and result:

- Source
- Result

If you're interested in how we render Markdown files, you might want to check out Redcarpet, our Ruby interface to the Sundown library.

Differences from traditional Markdown

Newlines

The biggest difference that GFM introduces is in the handling of linebreaks. With SM you can hard wrap paragraphs of text and they will be combined into a single paragraph. We find this to be the cause of a huge number of unintentional formatting errors. GFM treats newlines in paragraph-like content as real line breaks, which is probably what you intended.

The next paragraph contains two phrases separated by a single newline character:

```
Roses are red
Violets are blue
```

becomes

Roses are red Violets are blue

Multiple underscores in words

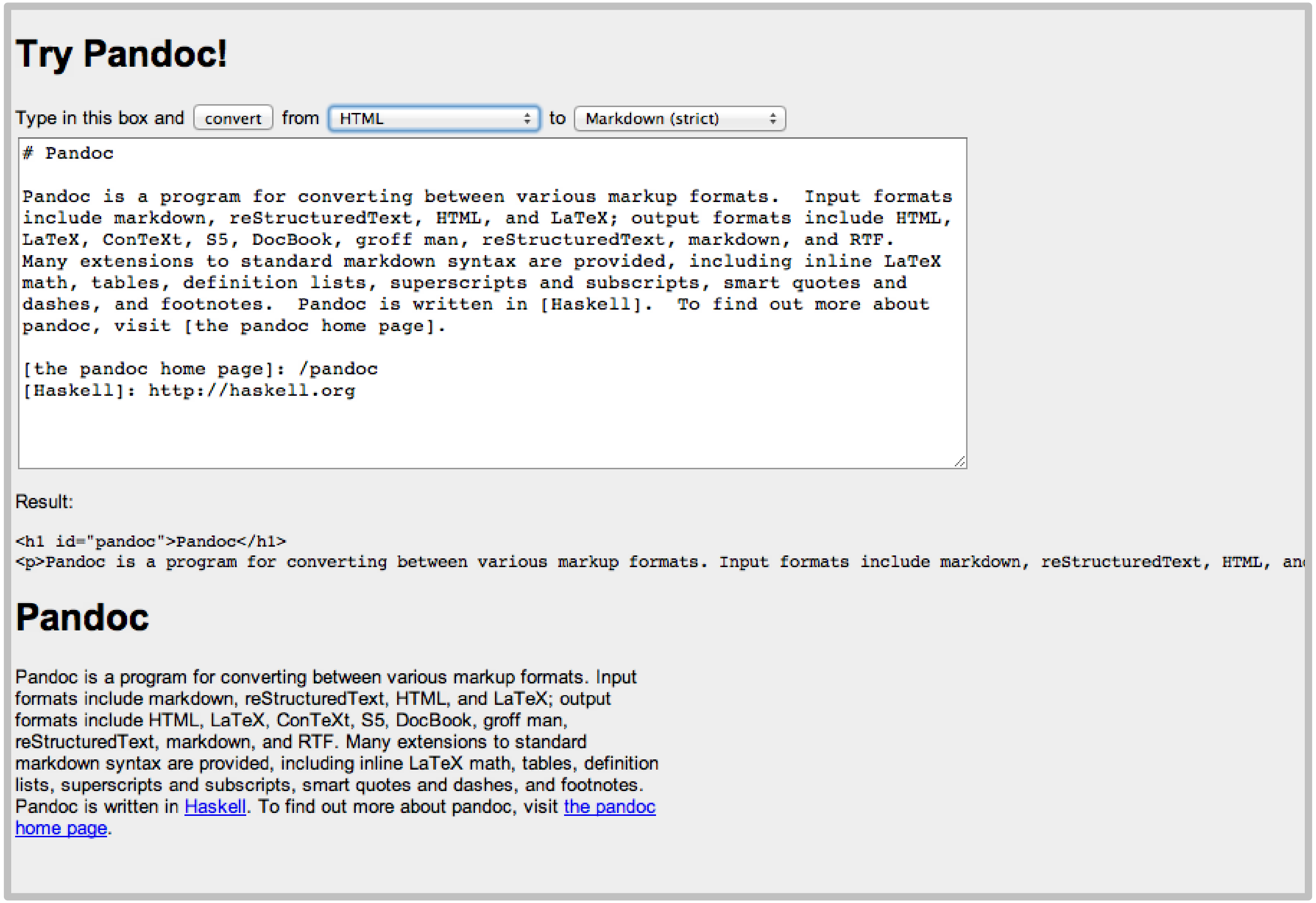
It is not reasonable to italicize just *part* of a word, especially when you're dealing with code and names often appear with multiple underscores. Therefore, GFM ignores multiple underscores in words.

Markdown Cheat Sheet

On Markdown-enabled portions of the site, press **M** on your keyboard to display a cheat sheet.

Source: http://github.github.com/github-flavored-markdown/

05a. Convert text to Markdown



Try Pandoc!

Type in this box and from to

```
# Pandoc

Pandoc is a program for converting between various markup formats.  Input formats
include markdown, reStructuredText, HTML, and LaTeX; output formats include HTML,
LaTeX, ConTeXt, S5, DocBook, groff man, reStructuredText, markdown, and RTF.
Many extensions to standard markdown syntax are provided, including inline LaTeX
math, tables, definition lists, superscripts and subscripts, smart quotes and
dashes, and footnotes.  Pandoc is written in [Haskell].  To find out more about
pandoc, visit [the pandoc home page].

[the pandoc home page]: /pandoc
[Haskell]: http://haskell.org
```

Result:

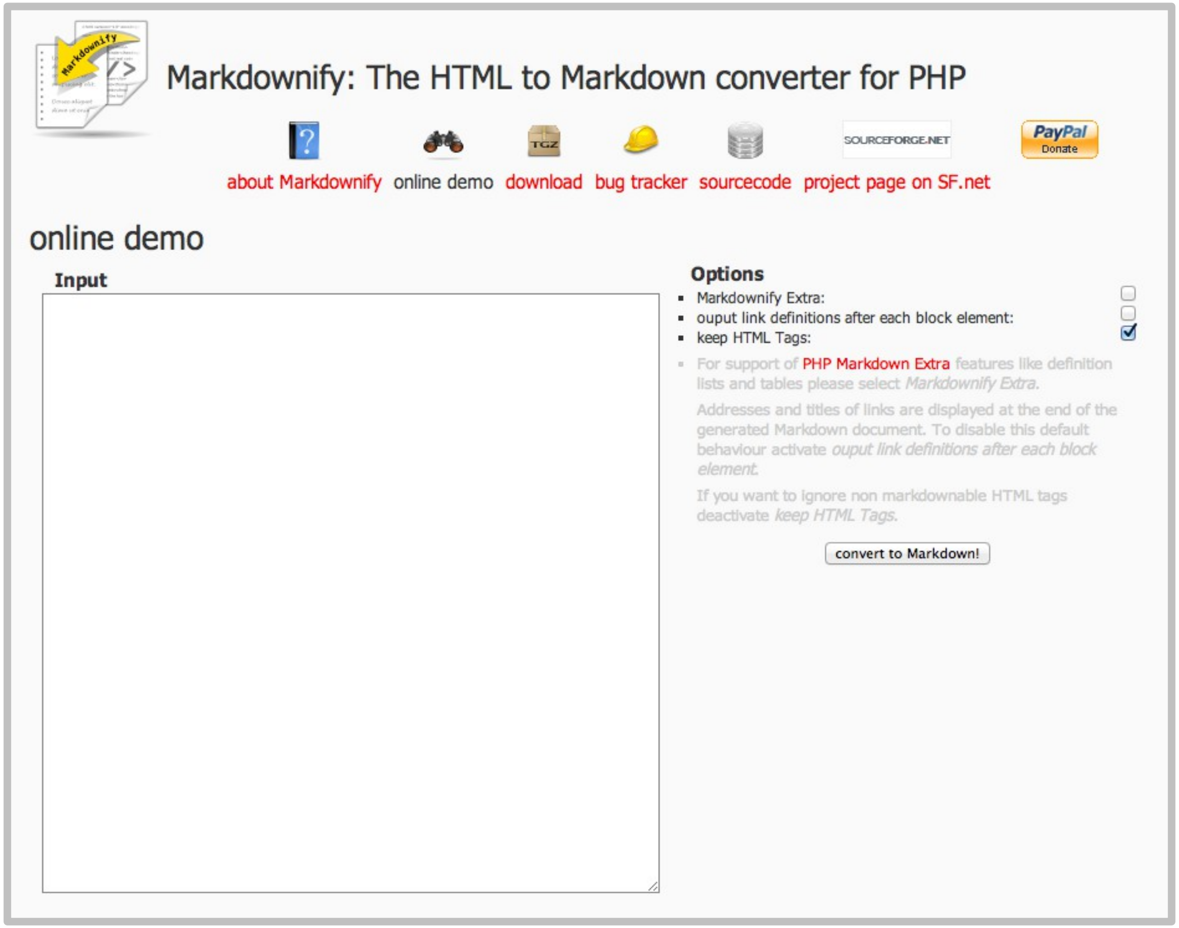
```
<h1 id="pandoc">Pandoc</h1>
<p>Pandoc is a program for converting between various markup formats. Input formats include markdown, reStructuredText, HTML, an
```

Pandoc

Pandoc is a program for converting between various markup formats. Input formats include markdown, reStructuredText, HTML, and LaTeX; output formats include HTML, LaTeX, ConTeXt, S5, DocBook, groff man, reStructuredText, markdown, and RTF. Many extensions to standard markdown syntax are provided, including inline LaTeX math, tables, definition lists, superscripts and subscripts, smart quotes and dashes, and footnotes. Pandoc is written in Haskell. To find out more about pandoc, visit the pandoc home page.

Source: http://johnmacfarlane.net/pandoc/try

05a. Convert text to Markdown



Markdownify: The HTML to Markdown converter for PHP

online demo

Input

Options

- Markdownify Extra: □
- ouput link definitions after each block element: □
- keep HTML Tags: ☑
- For support of PHP Markdown Extra features like definition lists and tables please select *Markdownify Extra.*

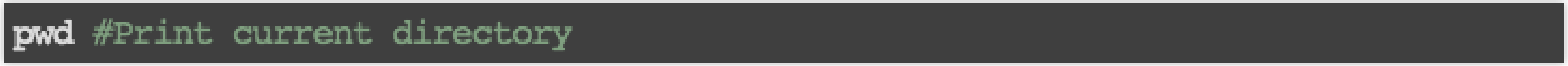
  Addresses and titles of links are displayed at the end of the generated Markdown document. To disable this default behaviour activate *ouput link definitions after each block element.*

  If you want to ignore non markdownable HTML tags deactivate *keep HTML Tags.*

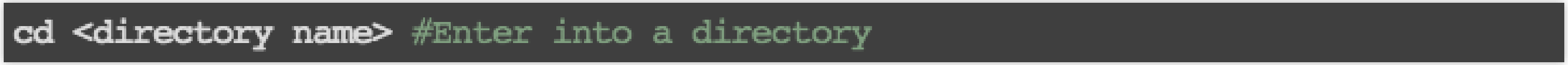
Source: http://milianw.de/projects/markdownify/demo.php

03.
Git Basics

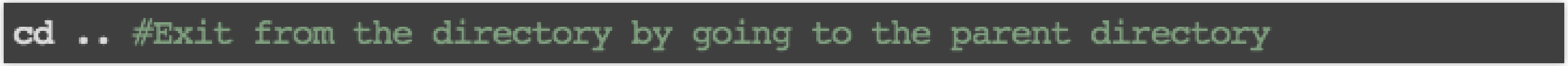
06: Terminal Basics



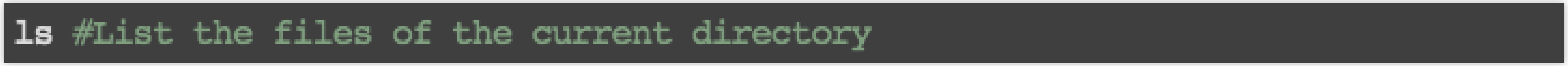
```
pwd #Print current directory
```



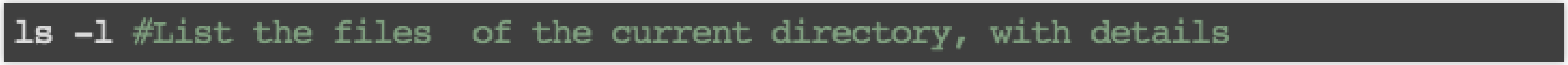
```
cd <directory name> #Enter into a directory
```



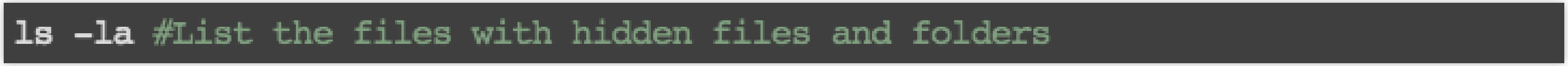
```
cd .. #Exit from the directory by going to the parent directory
```



```
ls #List the files of the current directory
```

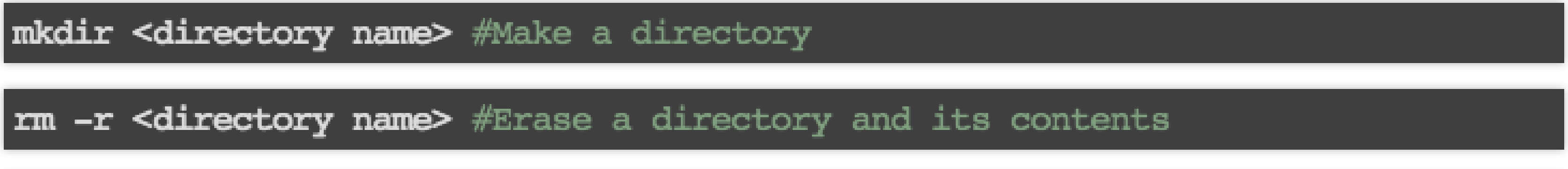


```
ls -l #List the files  of the current directory, with details
```



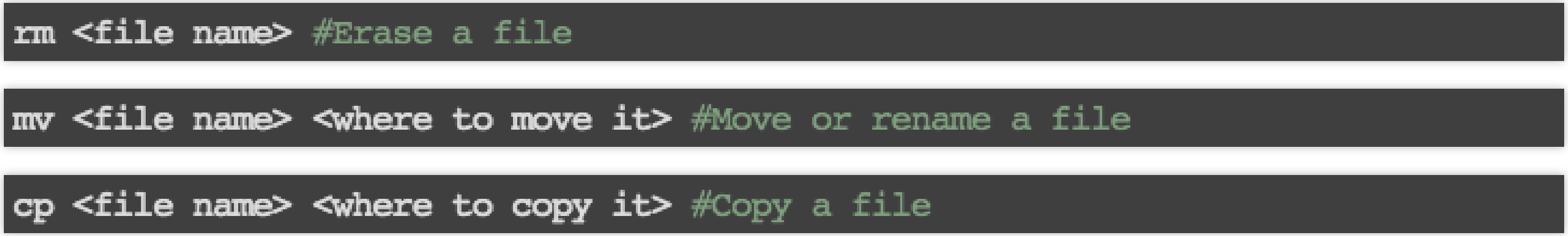
```
ls -la #List the files with hidden files and folders
```

06: Terminal Basics



```
mkdir <directory name> #Make a directory
```

```
rm -r <directory name> #Erase a directory and its contents
```

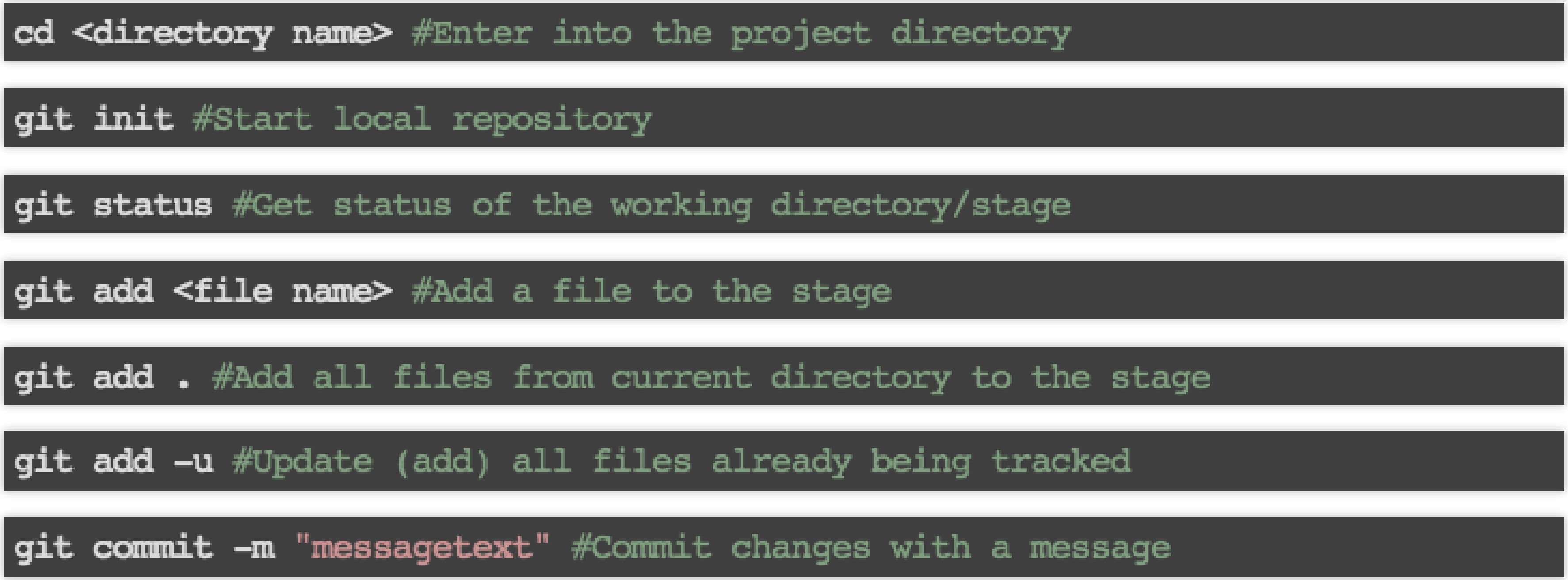


```
rm <file name> #Erase a file
```

```
mv <file name> <where to move it> #Move or rename a file
```

```
cp <file name> <where to copy it> #Copy a file
```

07: Git Basics



```
cd <directory name> #Enter into the project directory
```

```
git init #Start local repository
```

```
git status #Get status of the working directory/stage
```

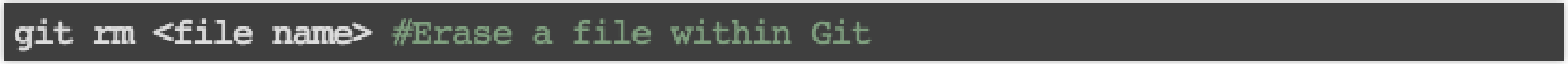
```
git add <file name> #Add a file to the stage
```

```
git add . #Add all files from current directory to the stage
```

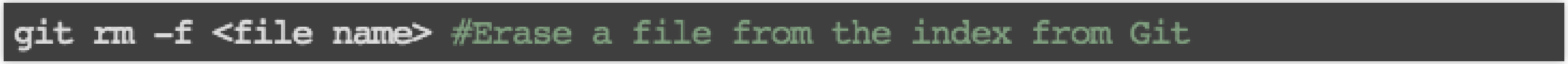
```
git add -u #Update (add) all files already being tracked
```

```
git commit -m "messagetext" #Commit changes with a message
```

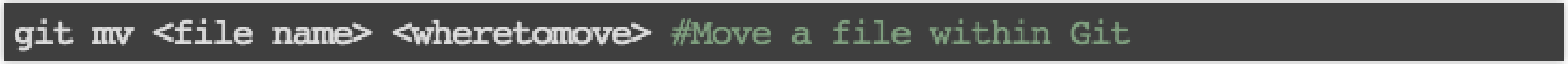
07: Git Basics



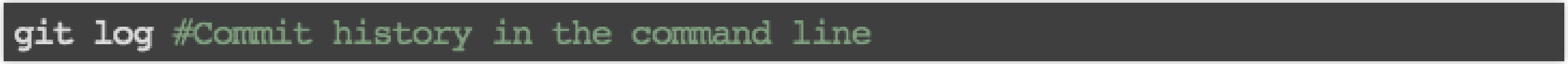
```
git rm <file name> #Erase a file within Git
```



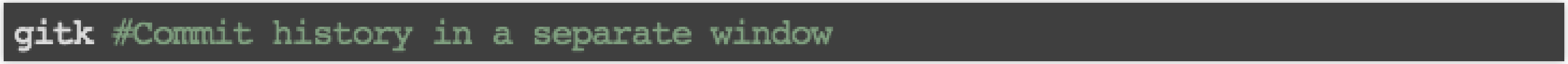
```
git rm -f <file name> #Erase a file from the index from Git
```



```
git mv <file name> <wheretomove> #Move a file within Git
```

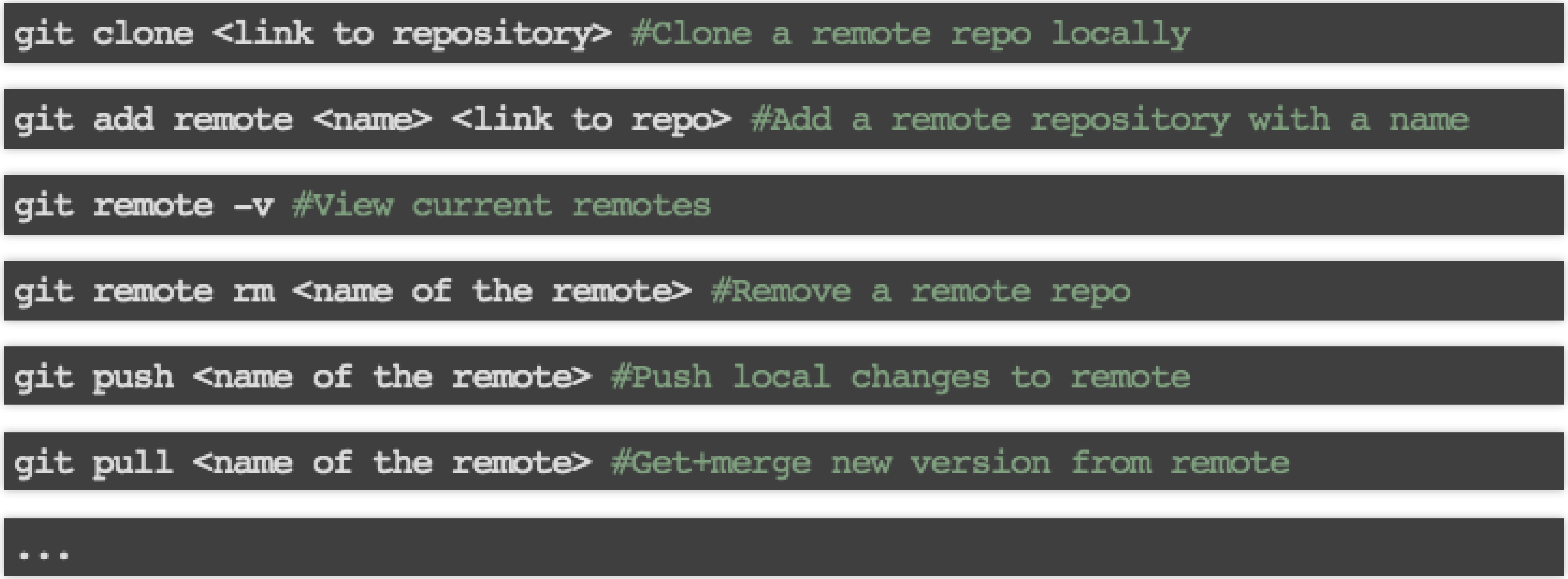


```
git log #Commit history in the command line
```



```
gitk #Commit history in a separate window
```

08: Git Remote



```
git clone <link to repository> #Clone a remote repo locally
```

```
git add remote <name> <link to repo> #Add a remote repository with a name
```

```
git remote -v #View current remotes
```

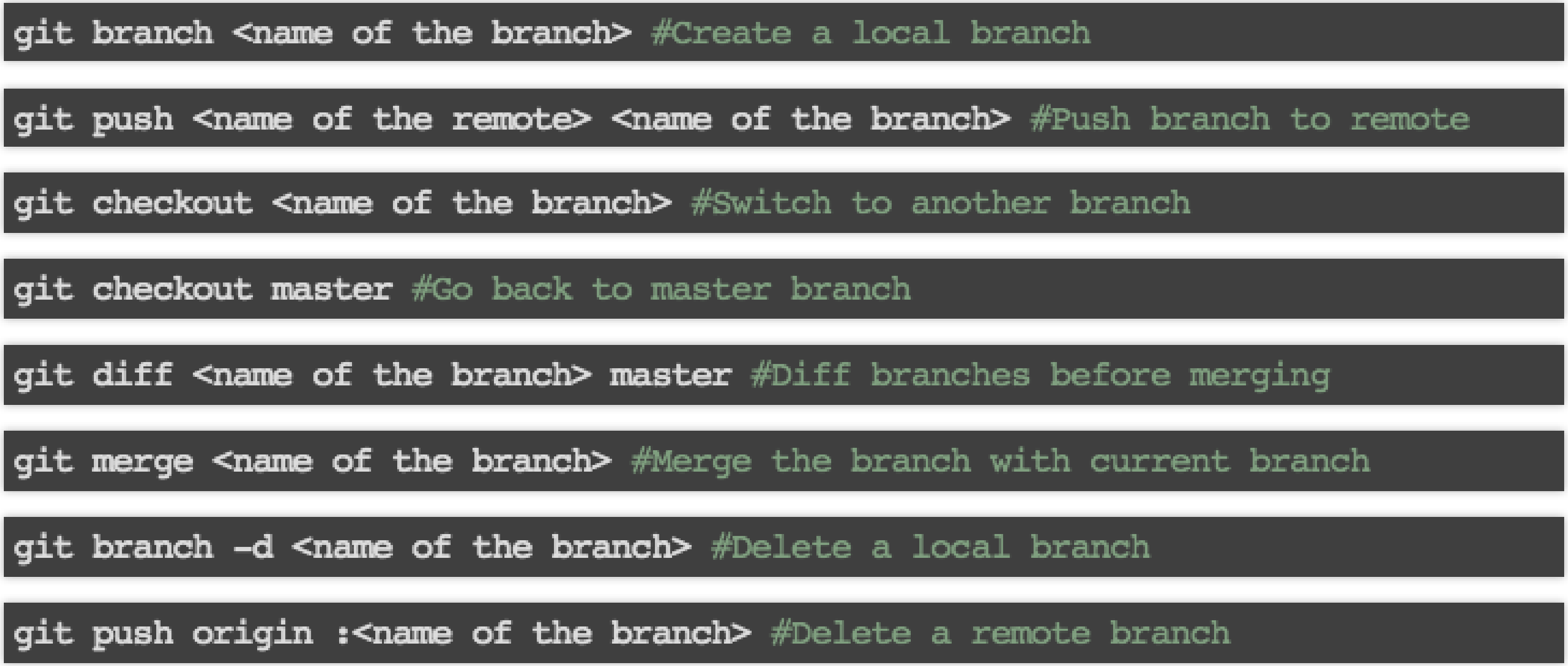
```
git remote rm <name of the remote> #Remove a remote repo
```

```
git push <name of the remote> #Push local changes to remote
```

```
git pull <name of the remote> #Get+merge new version from remote
```

```
...
```

09: Git Branches



```
git branch <name of the branch> #Create a local branch
```

```
git push <name of the remote> <name of the branch> #Push branch to remote
```

```
git checkout <name of the branch> #Switch to another branch
```

```
git checkout master #Go back to master branch
```

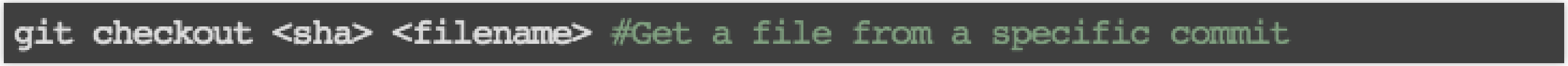
```
git diff <name of the branch> master #Diff branches before merging
```

```
git merge <name of the branch> #Merge the branch with current branch
```

```
git branch -d <name of the branch> #Delete a local branch
```

```
git push origin :<name of the branch> #Delete a remote branch
```

10: Git Back in Time



```
git checkout <sha> <filename> #Get a file from a specific commit
```

or

```
git checkout <sha> #Go back in time at a specific commit
```

You will get a Detached Head, so...

```
git branch <name of the branch> #But then create e new branch!
```

```
git checkout <name of the branch> #Go to the new branch!
```

```
... #Do your modifications
```

```
git checkout master  #Go to master
```

```
git merge <name of the branch> #Update master with the branch you created
```

```
git branch -d <name of the branch> #Delete that branch if not needed
```

04.
Exercise:
Send your GitHub / BitBucket username to massimo.menichinelli@aalto.fi

05.
Exercise:
Create a local repository, create your bio with your picture inside it. Try it with Git.

06.
Exercise:
Create a remote repository, create your bio with your picture inside it. Try it with Git and/or with the GitHub client.

Thank you!!

Massimo Menichinelli
Aalto Media Factory
massimo.menichinelli@aalto.fi
@openp2pdesign
http://www.slideshare.net/openp2pdesign