Git, Android Studio, and Beyond

How and open source tool integrates with an open source platform

A Brief History of Open Source

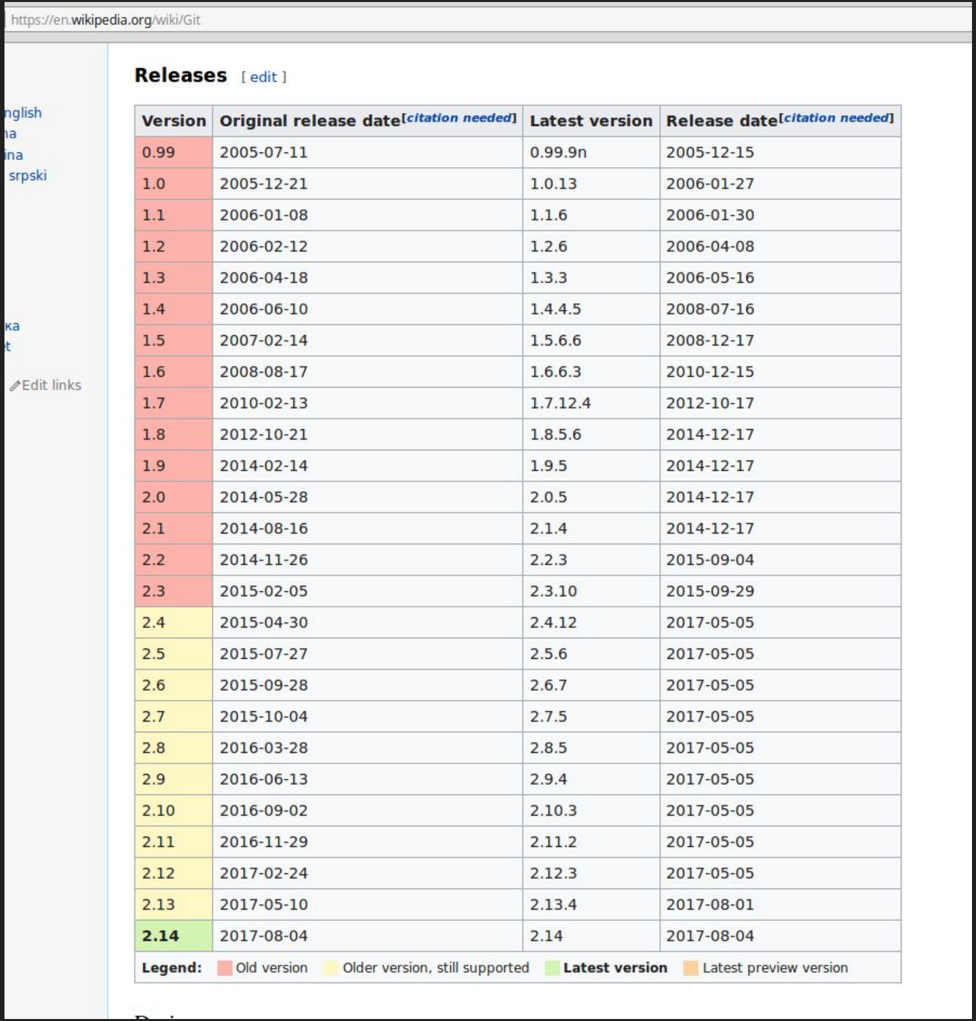
- Linux
- Git
- The Cathedral and The Bazaar
- The spread of open source
- Android sits on Linux

Where did Linux come from

- Unix
- Linus Torvalds
- Open Source development
- Spread
- Something else

The need for Git

- Former VCS Bitkeeper, proprietary system
- Need for a new VCS with better performance and right functionality
- Developed in 2005
- Many possible origins of the name Git
- About a month to really get up and running and performing the way intended



https://en.wikipedia.org/wiki/Git

nglish
na
ina
srpski
ка
t
Edit links

Releases [edit]

Version	Original release date[citation needed]	Latest version	Release date[citation needed]
0.99	2005-07-11	0.99.9n	2005-12-15
1.0	2005-12-21	1.0.13	2006-01-27
1.1	2006-01-08	1.1.6	2006-01-30
1.2	2006-02-12	1.2.6	2006-04-08
1.3	2006-04-18	1.3.3	2006-05-16
1.4	2006-06-10	1.4.4.5	2008-07-16
1.5	2007-02-14	1.5.6.6	2008-12-17
1.6	2008-08-17	1.6.6.3	2010-12-15
1.7	2010-02-13	1.7.12.4	2012-10-17
1.8	2012-10-21	1.8.5.6	2014-12-17
1.9	2014-02-14	1.9.5	2014-12-17
2.0	2014-05-28	2.0.5	2014-12-17
2.1	2014-08-16	2.1.4	2014-12-17
2.2	2014-11-26	2.2.3	2015-09-04
2.3	2015-02-05	2.3.10	2015-09-29
2.4	2015-04-30	2.4.12	2017-05-05
2.5	2015-07-27	2.5.6	2017-05-05
2.6	2015-09-28	2.6.7	2017-05-05
2.7	2015-10-04	2.7.5	2017-05-05
2.8	2016-03-28	2.8.5	2017-05-05
2.9	2016-06-13	2.9.4	2017-05-05
2.10	2016-09-02	2.10.3	2017-05-05
2.11	2016-11-29	2.11.2	2017-05-05
2.12	2017-02-24	2.12.3	2017-05-05
2.13	2017-05-10	2.13.4	2017-08-01
2.14	2017-08-04	2.14	2017-08-04

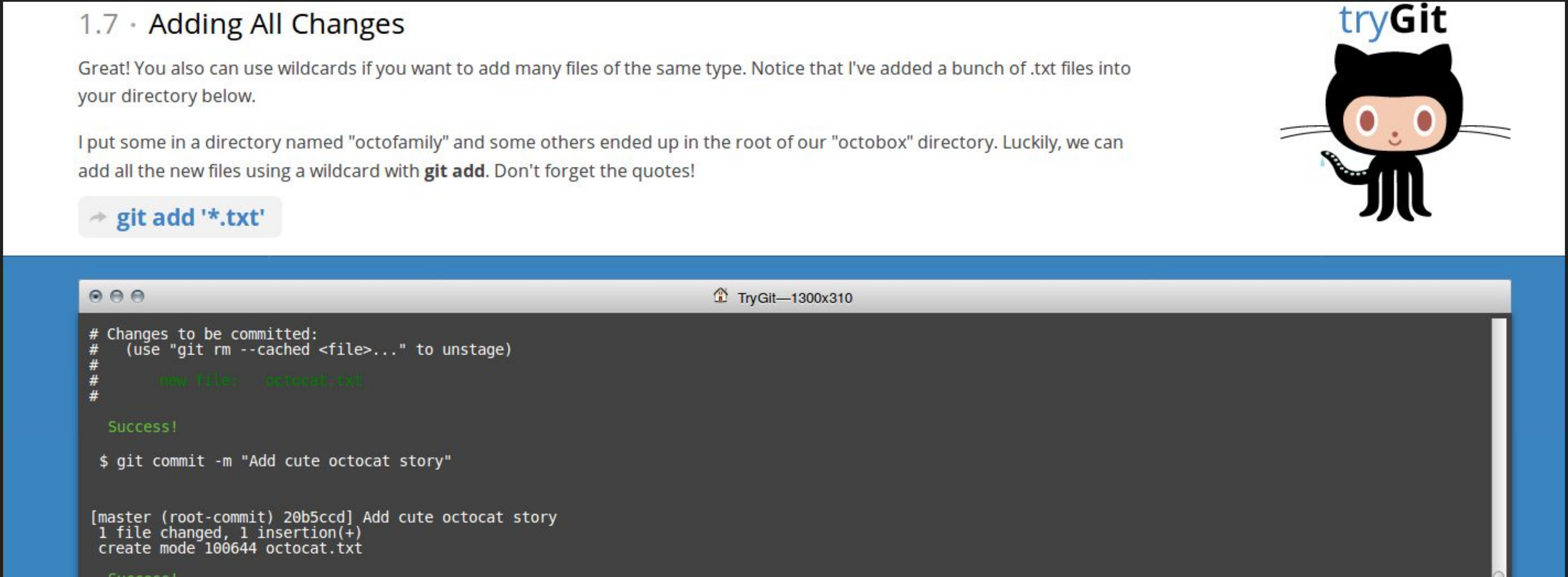
Legend: 🟥 Old version 🟨 Older version, still supported 🟩 **Latest version** 🟧 Latest preview version

Basic Git commands

- git add
- git commit
- git push
- git pull
- git log
- git status

Git web site and demo

- Home for git -- https://git-scm.com/
- Tutorial to try git and learn basics -- https://try.github.io/levels/1/challenges/5



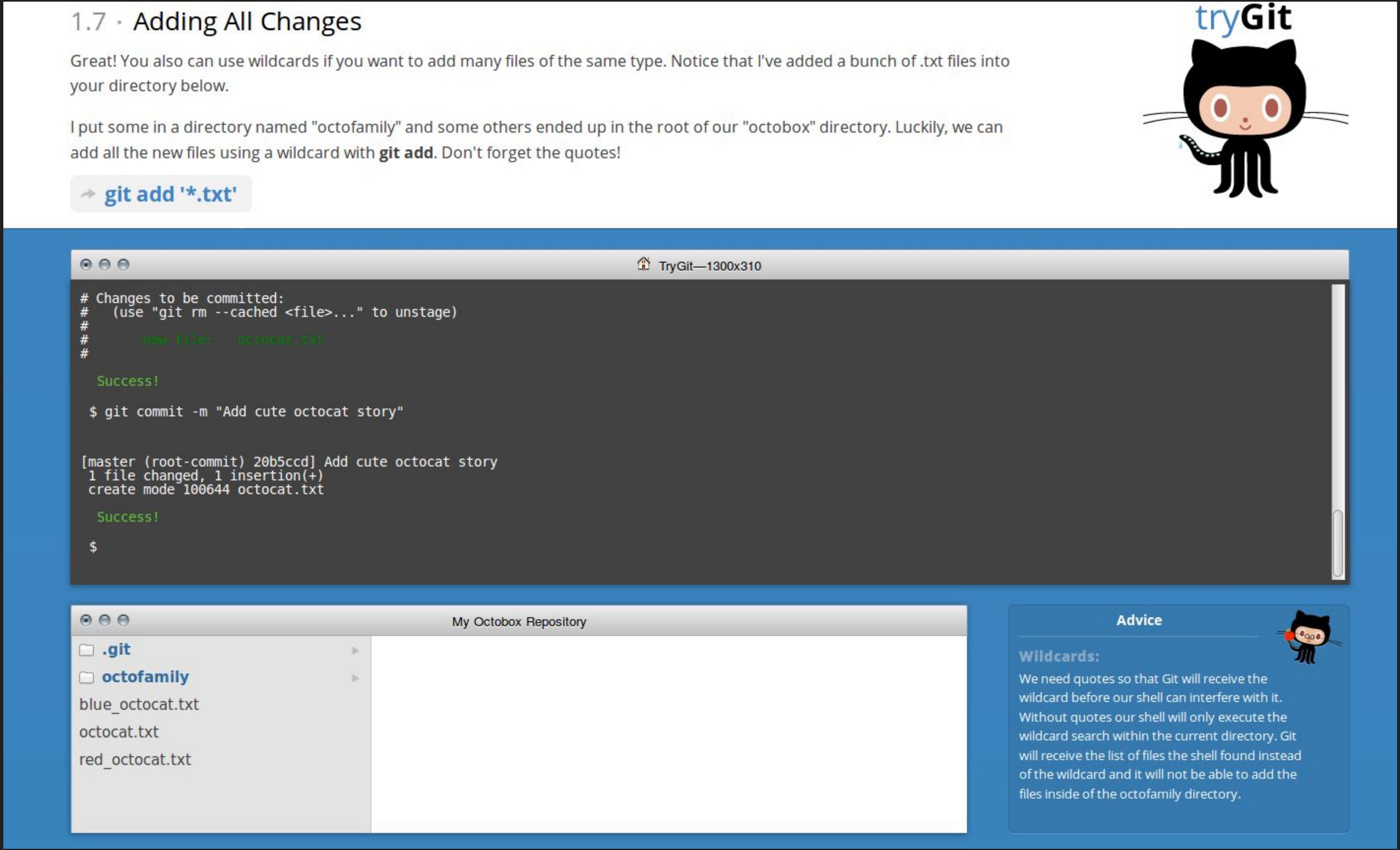
1.7 · Adding All Changes

Great! You also can use wildcards if you want to add many files of the same type. Notice that I've added a bunch of .txt files into your directory below.

I put some in a directory named "octofamily" and some others ended up in the root of our "octobox" directory. Luckily, we can add all the new files using a wildcard with **git add**. Don't forget the quotes!

```
git add '*.txt'
```

```
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file:   octocat.txt
#

  Success!

 $ git commit -m "Add cute octocat story"


[master (root-commit) 20b5ccd] Add cute octocat story
 1 file changed, 1 insertion(+)
 create mode 100644 octocat.txt

  Success!
```



tryGit

1.7 · Adding All Changes

Great! You also can use wildcards if you want to add many files of the same type. Notice that I've added a bunch of .txt files into your directory below.

I put some in a directory named "octofamily" and some others ended up in the root of our "octobox" directory. Luckily, we can add all the new files using a wildcard with **git add**. Don't forget the quotes!

```
git add '*.txt'
```

TryGit—1300x310

```
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file:   octocat.txt
#

  Success!

 $ git commit -m "Add cute octocat story"


[master (root-commit) 20b5ccd] Add cute octocat story
 1 file changed, 1 insertion(+)
 create mode 100644 octocat.txt

  Success!

 $
```

My Octobox Repository

- .git
- octofamily
- blue_octocat.txt
- octocat.txt
- red_octocat.txt

Advice

Wildcards:

We need quotes so that Git will receive the wildcard before our shell can interfere with it. Without quotes our shell will only execute the wildcard search within the current directory. Git will receive the list of files the shell found instead of the wildcard and it will not be able to add the files inside of the octofamily directory.

More git training resources

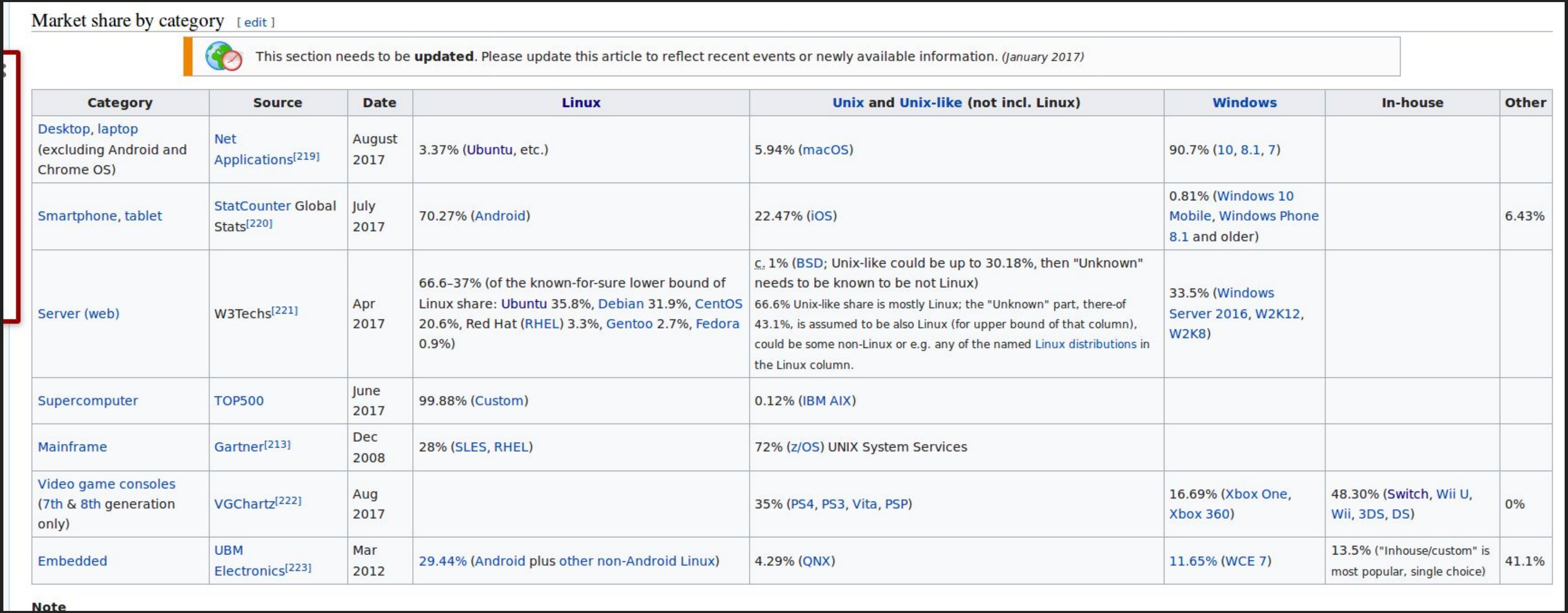
- https://try.github.io/wrap_up
- https://help.github.com/
- https://services.github.com/customized-training
- http://gitimmersion.com/
- http://rogerdudler.github.io/git-guide/
-

Linus Torvalds talks about Linux and Git

- Youtube TED talk -- https://www.youtube.com/watch?v=Vo9KPk-gqKk
- 20 minute talk where Linus Torvalds talks about his work, open source, Linux, git, and what makes him good at what he does

Some Linux numbers and facts

- 50 Core developers
- Roughly 1,000 contributors to each release
- Linux design tried initially to mimic Unix
- Git much more of a design from scratch
- Linux has a single kernel that runs on a multitude of processors -- from mobile devices to supercomputers

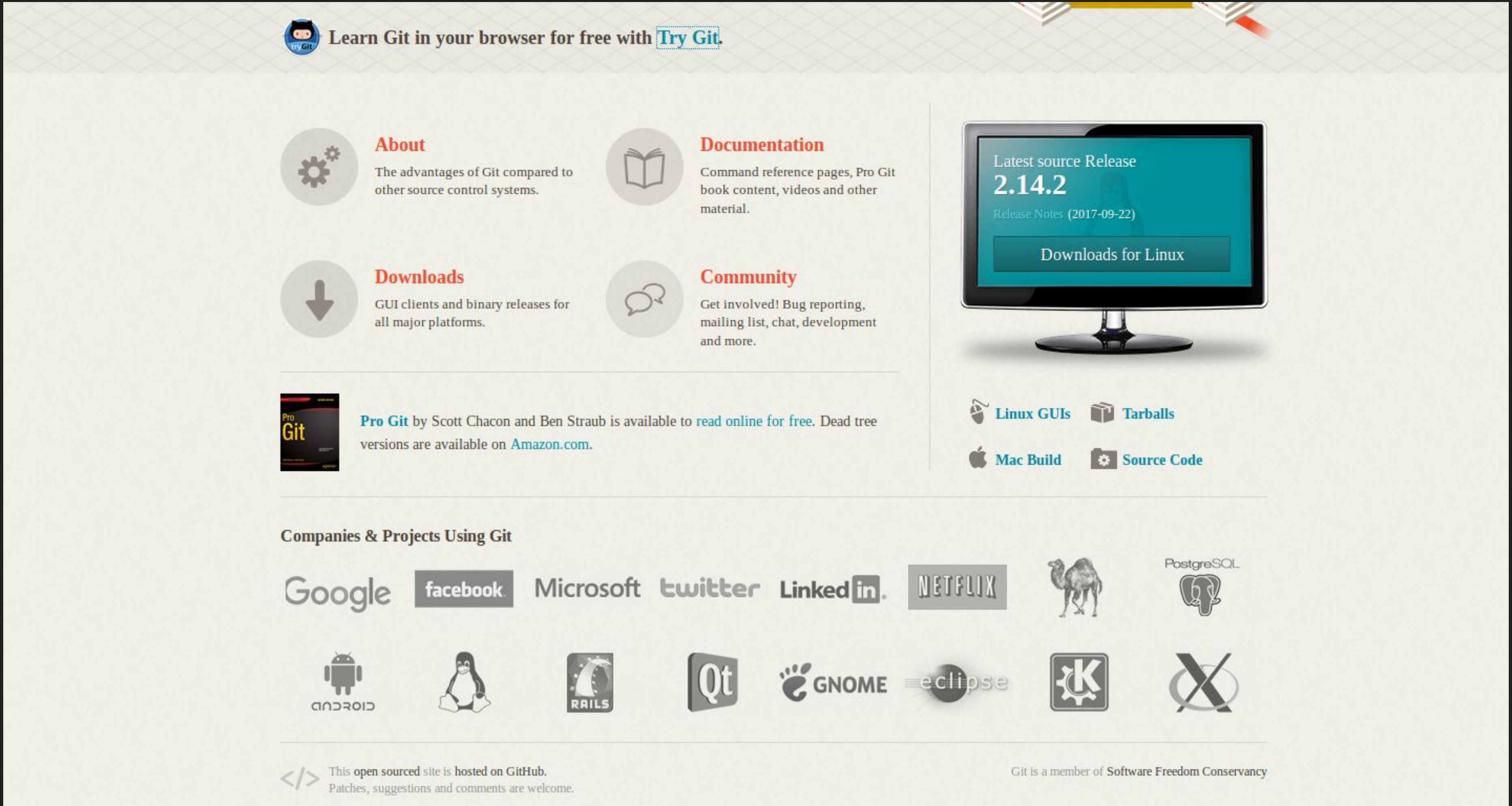


Market share by category [edit]

This section needs to be **updated**. Please update this article to reflect recent events or newly available information. *(January 2017)*

Category	Source	Date	Linux	Unix and Unix-like (not incl. Linux)	Windows	In-house	Other
Desktop, laptop (excluding Android and Chrome OS)	Net Applications[219]	August 2017	3.37% (Ubuntu, etc.)	5.94% (macOS)	90.7% (10, 8.1, 7)		
Smartphone, tablet	StatCounter Global Stats[220]	July 2017	70.27% (Android)	22.47% (iOS)	0.81% (Windows 10 Mobile, Windows Phone 8.1 and older)		6.43%
Server (web)	W3Techs[221]	Apr 2017	66.6–37% (of the known-for-sure lower bound of Linux share: Ubuntu 35.8%, Debian 31.9%, CentOS 20.6%, Red Hat (RHEL) 3.3%, Gentoo 2.7%, Fedora 0.9%)	c. 1% (BSD; Unix-like could be up to 30.18%, then "Unknown" needs to be known to be not Linux) 66.6% Unix-like share is mostly Linux; the "Unknown" part, there-of 43.1%, is assumed to be also Linux (for upper bound of that column), could be some non-Linux or e.g. any of the named Linux distributions in the Linux column.	33.5% (Windows Server 2016, W2K12, W2K8)		
Supercomputer	TOP500	June 2017	99.88% (Custom)	0.12% (IBM AIX)			
Mainframe	Gartner[213]	Dec 2008	28% (SLES, RHEL)	72% (z/OS) UNIX System Services			
Video game consoles (7th & 8th generation only)	VGChartz[222]	Aug 2017		35% (PS4, PS3, Vita, PSP)	16.69% (Xbox One, Xbox 360)	48.30% (Switch, Wii U, Wii, 3DS, DS)	0%
Embedded	UBM Electronics[223]	Mar 2012	29.44% (Android plus other non-Android Linux)	4.29% (QNX)	11.65% (WCE 7)	13.5% ("Inhouse/custom" is most popular, single choice)	41.1%

Note



Learn Git in your browser for free with Try Git.

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! Bug reporting, mailing list, chat, development and more.

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

Latest source Release

2.14.2

Release Notes (2017-09-22)

Downloads for Linux

Linux GUIs

Tarballs

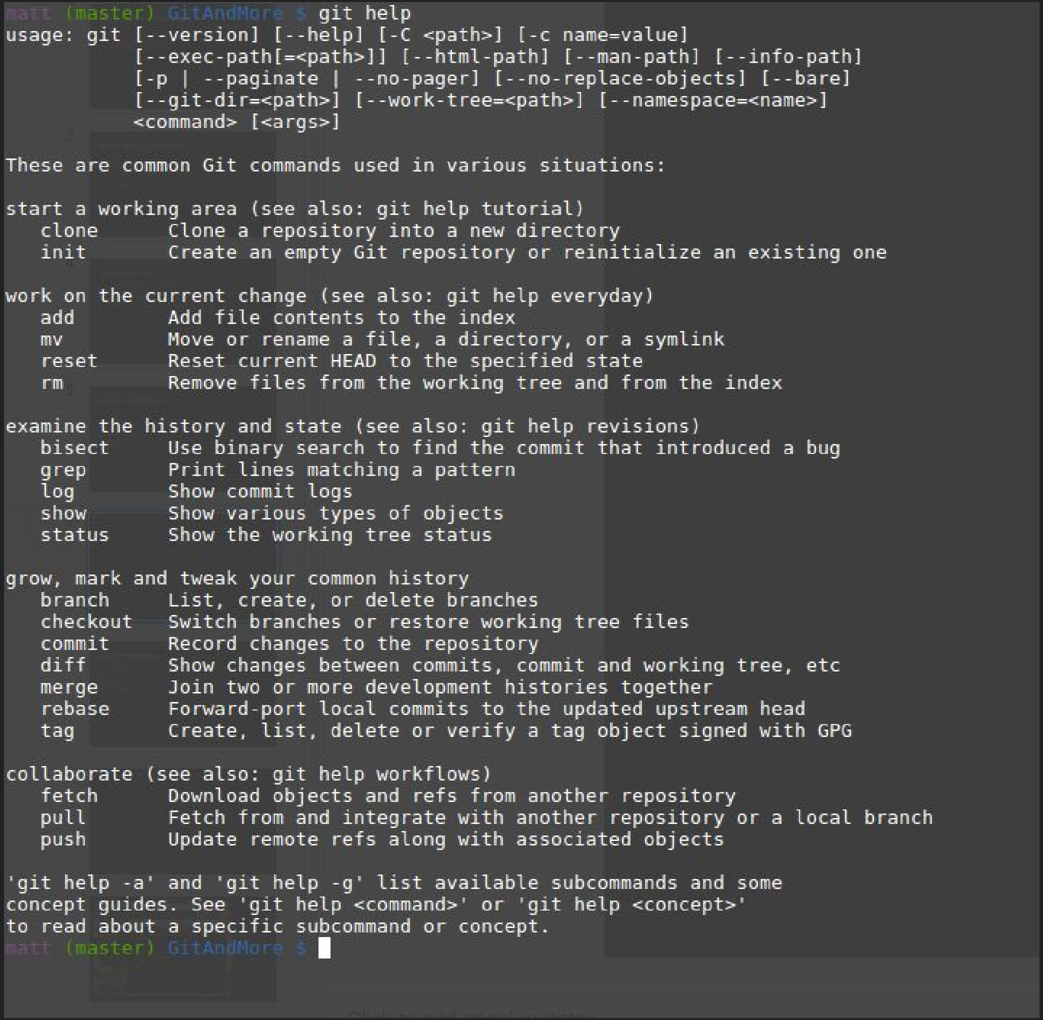
Mac Build

Source Code

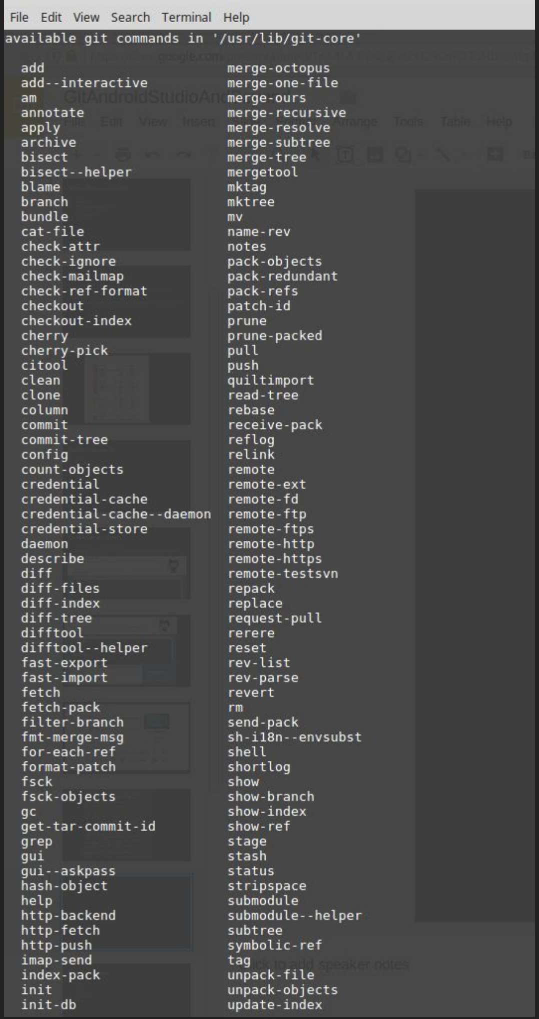
Companies & Projects Using Git

</> This open sourced site is hosted on GitHub. Patches, suggestions and comments are welcome.

Git is a member of Software Freedom Conservancy



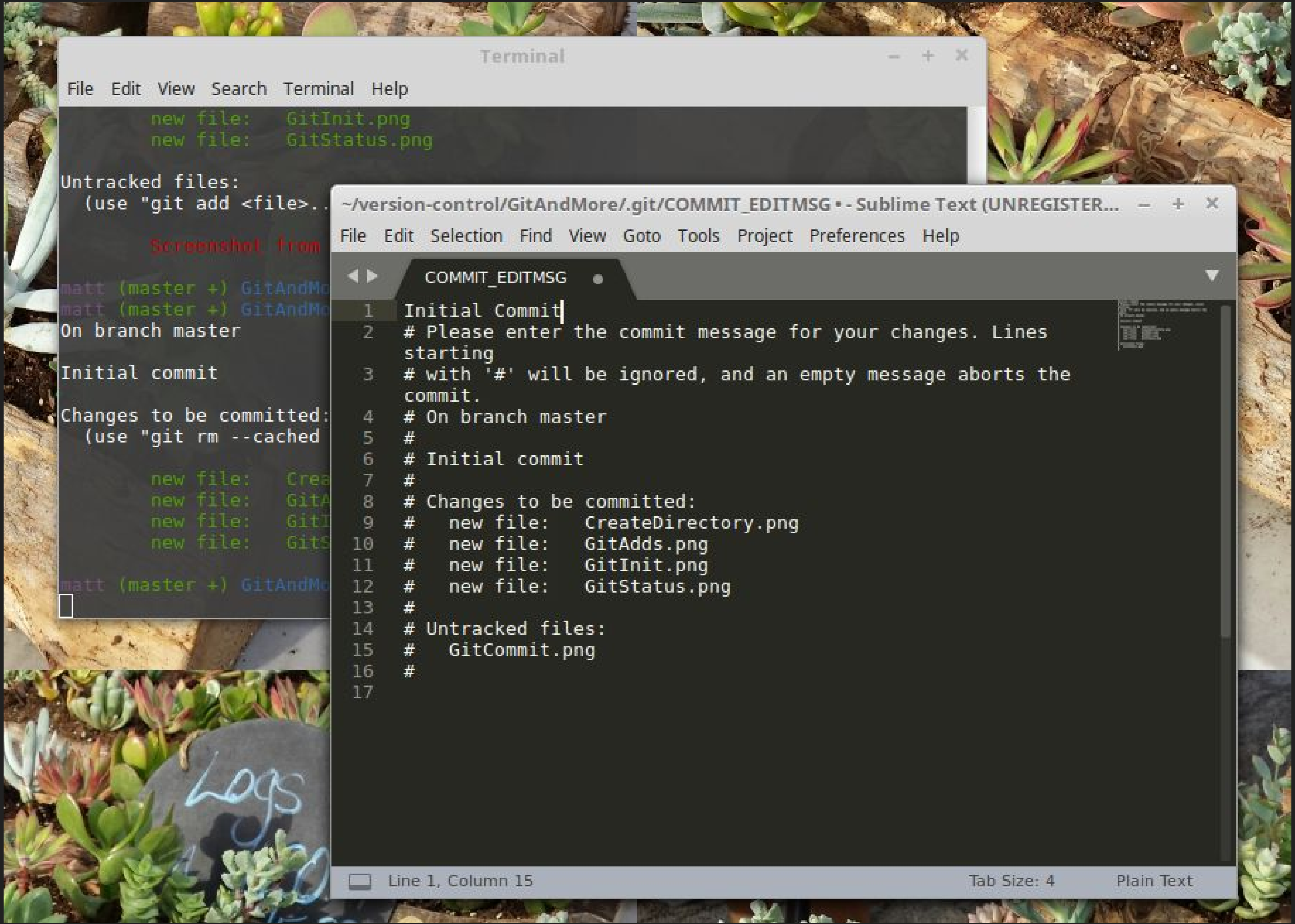
```
matt (master) GitAndMore $ git help
usage: git [--version] [--help] [-C <path>] [-c name=value]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
   clone      Clone a repository into a new directory
   init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
   add        Add file contents to the index
   mv         Move or rename a file, a directory, or a symlink
   reset      Reset current HEAD to the specified state
   rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
   bisect     Use binary search to find the commit that introduced a bug
   grep       Print lines matching a pattern
   log        Show commit logs
   show       Show various types of objects
   status     Show the working tree status

grow, mark and tweak your common history
   branch     List, create, or delete branches
   checkout   Switch branches or restore working tree files
   commit     Record changes to the repository
   diff       Show changes between commits, commit and working tree, etc
   merge      Join two or more development histories together
   rebase     Forward-port local commits to the updated upstream head
   tag        Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
   fetch      Download objects and refs from another repository
   pull       Fetch from and integrate with another repository or a local branch
   push       Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
matt (master) GitAndMore $ 
```



```
available git commands in '/usr/lib/git-core'

  add                       merge-octopus
  add--interactive          merge-one-file
  am                        merge-ours
  annotate                  merge-recursive
  apply                     merge-resolve
  archive                   merge-subtree
  bisect                    merge-tree
  bisect--helper            mergetool
  blame                     mktag
  branch                    mktree
  bundle                    mv
  cat-file                  name-rev
  check-attr                notes
  check-ignore              pack-objects
  check-mailmap             pack-redundant
  check-ref-format          pack-refs
  checkout                  patch-id
  checkout-index            prune
  cherry                    prune-packed
  cherry-pick               pull
  citool                    push
  clean                     quiltimport
  clone                     read-tree
  column                    rebase
  commit                    receive-pack
  commit-tree               reflog
  config                    relink
  count-objects             remote
  credential                remote-ext
  credential-cache          remote-fd
  credential-cache--daemon  remote-ftp
  credential-store          remote-ftps
  daemon                    remote-http
  describe                  remote-https
  diff                      remote-testsvn
  diff-files                repack
  diff-index                replace
  diff-tree                 request-pull
  difftool                  rerere
  difftool--helper          reset
  fast-export               rev-list
  fast-import               rev-parse
  fetch                     revert
  fetch-pack                rm
  filter-branch             send-pack
  fmt-merge-msg             sh-i18n--envsubst
  for-each-ref              shell
  format-patch              shortlog
  fsck                      show
  fsck-objects              show-branch
  gc                        show-index
  get-tar-commit-id         show-ref
  grep                      stage
  gui                       stash
  gui--askpass              status
  hash-object               stripspace
  help                      submodule
  http-backend              submodule--helper
  http-fetch                subtree
  http-push                 symbolic-ref
  imap-send                 tag
  index-pack                unpack-file
  init                      unpack-objects
  init-db                   update-index
```

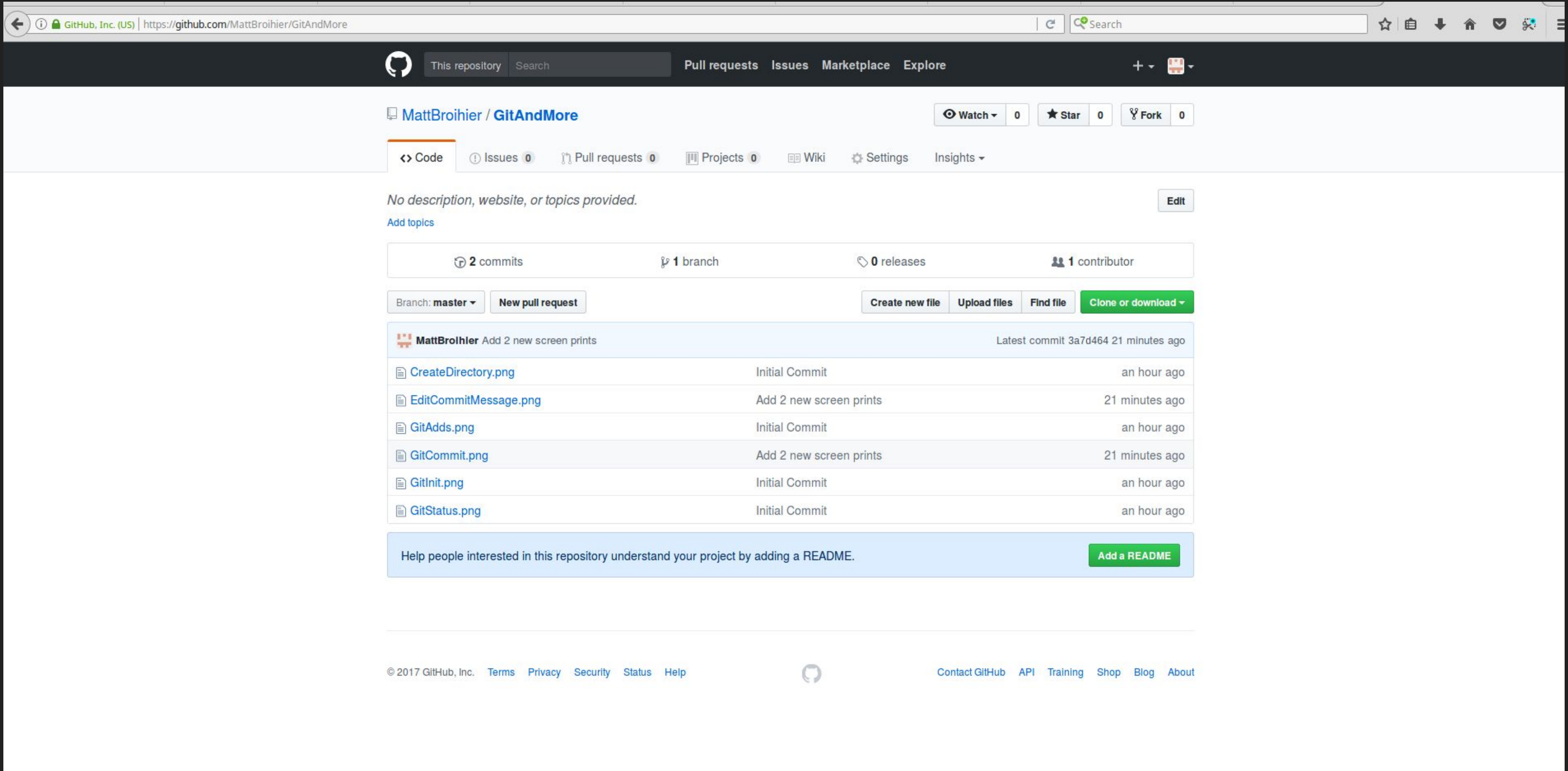
Basic Git concepts

- Repository
- Branch
- Fork
- Pull request
- Merge


Terminal
File Edit View Search Terminal Help
~/version-control/GitAndMore/.git/COMMIT_EDITMSG • - Sublime Text (UNREGISTER...
File Edit Selection Find View Goto Tools Project Preferences Help
COMMIT_EDITMSG
Line 1, Column 15
Tab Size: 4
Plain Text

```
        new file:   GitInit.png
        new file:   GitStatus.png

Untracked files:
  (use "git add <file>..

        Screenshot from

matt (master +) GitAndMo
matt (master +) GitAndMo
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached

        new file:   Crea
        new file:   GitA
        new file:   GitI
        new file:   GitS

matt (master +) GitAndMo
```

```
Initial Commit
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch master
#
# Initial commit
#
# Changes to be committed:
#	new file:   CreateDirectory.png
#	new file:   GitAdds.png
#	new file:   GitInit.png
#	new file:   GitStatus.png
#
# Untracked files:
#	GitCommit.png
#

```



MattBroihier / GitAndMore

No description, website, or topics provided.

Add topics

2 commits | 1 branch | 0 releases | 1 contributor

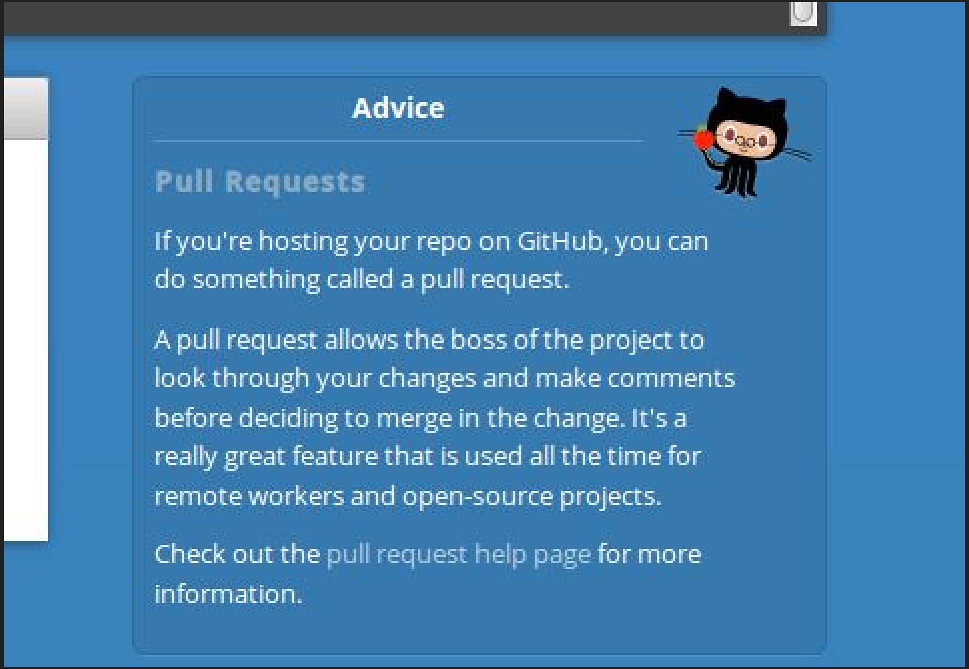
MattBroihier Add 2 new screen prints | Latest commit 3a7d464 21 minutes ago

CreateDirectory.png	Initial Commit	an hour ago
EditCommitMessage.png	Add 2 new screen prints	21 minutes ago
GitAdds.png	Initial Commit	an hour ago
GitCommit.png	Add 2 new screen prints	21 minutes ago
GitInit.png	Initial Commit	an hour ago
GitStatus.png	Initial Commit	an hour ago

Help people interested in this repository understand your project by adding a README.

© 2017 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub API Training Shop Blog About

What is a pull request?



Advice

Pull Requests

If you're hosting your repo on GitHub, you can do something called a pull request.

A pull request allows the boss of the project to look through your changes and make comments before deciding to merge in the change. It's a really great feature that is used all the time for remote workers and open-source projects.

Check out the pull request help page for more information.