

GIT IT ON

a hands on presentation by
@martinbing

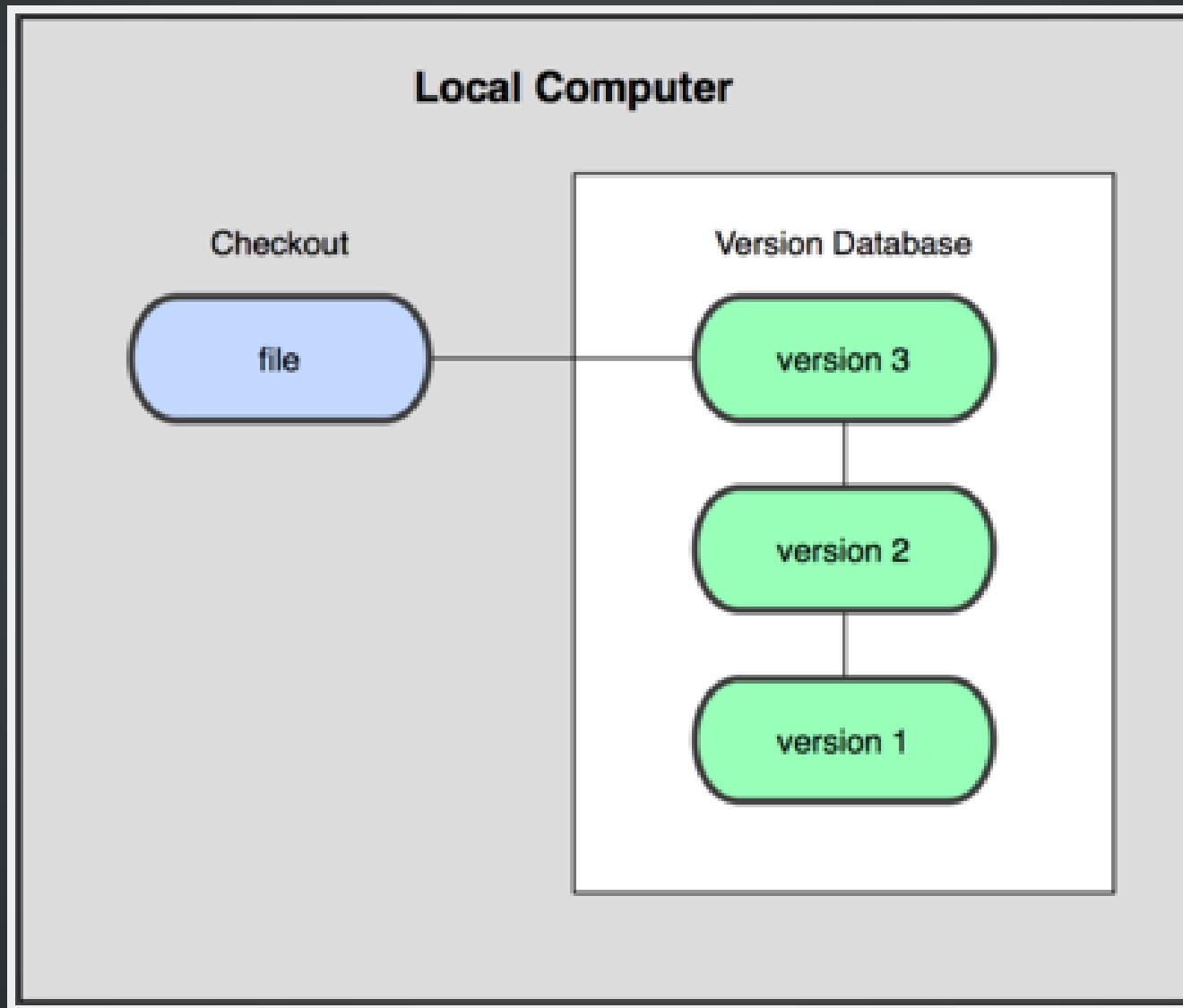
GIT WHAT?

- Open Source
- Distributed Version Control System (DVCS)
- (VCS) aka Source Code Management (SCM)
- Tool to keep track of changes (in files)
- Command Line (BASH)
- Supported by UNIX-like systems (Linux, Mac OS X, Solaris) and Windows

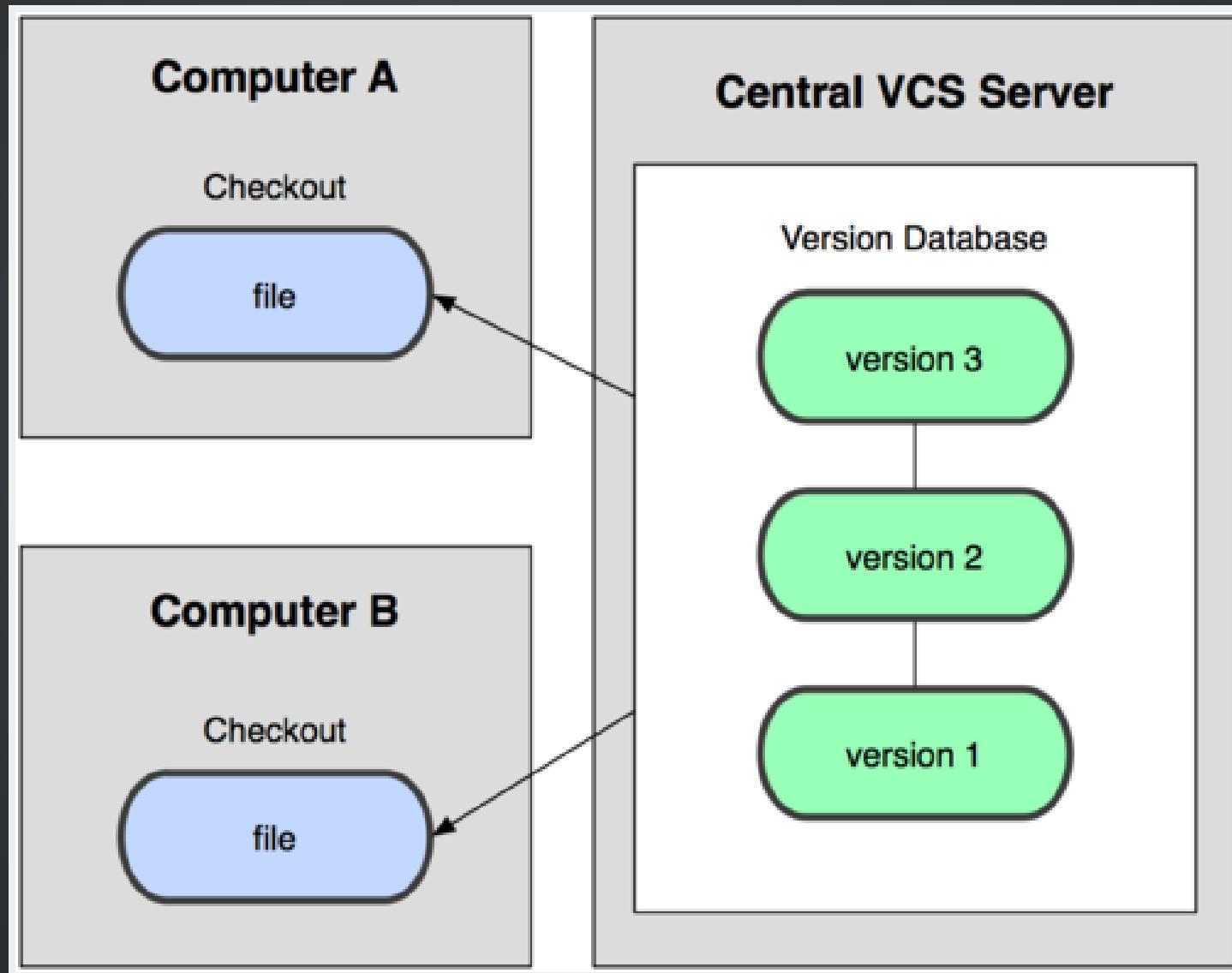
VERSION CONTROL SYSTEMS

- Local
- Centralized
- Distributed

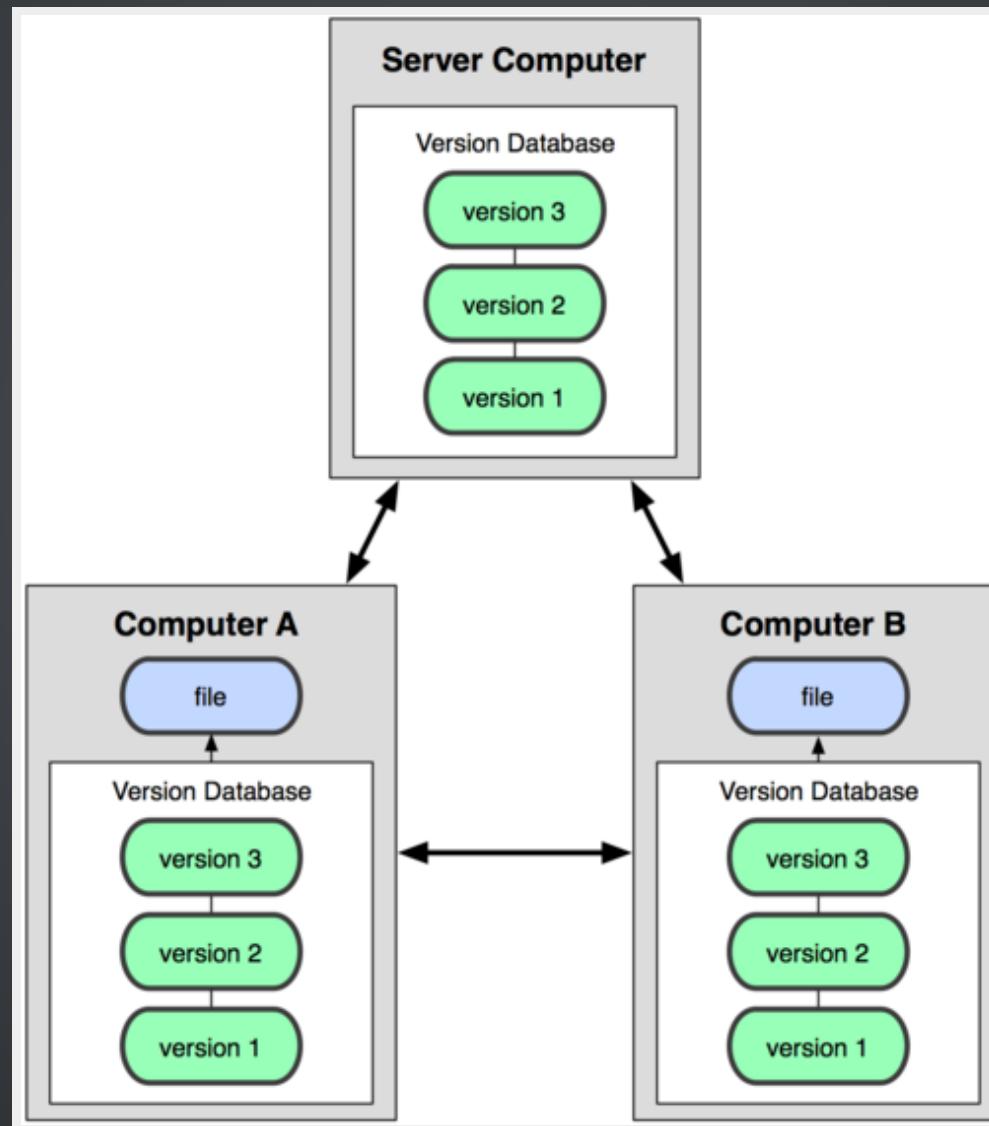
LOCAL VERSION CONTROL SYSTEM



CENTRALIZED VERSION CONTROL SYSTEM



DISTRIBUTED VERSION CONTROL SYSTEM



HISTORY

- GIT created in 2005 by Linus Torvalds
- Developed on a Linux (what else?)
- Made in 2 weeks
- Why pay if you can build it better (ref. BitKeeper)
- See the man in action: [Linus Torvalds on Git](#)



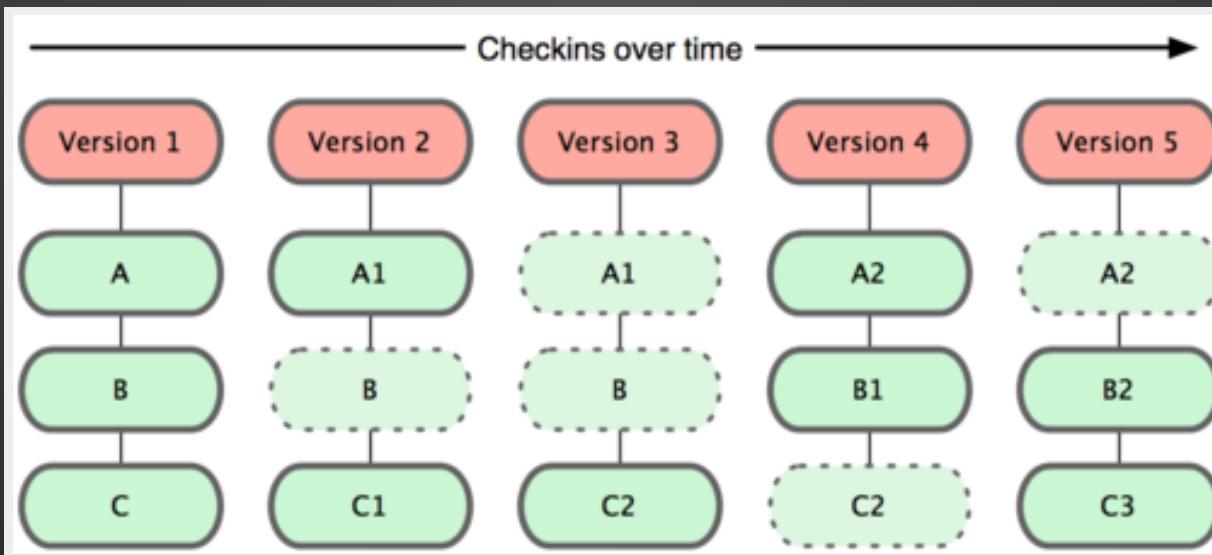
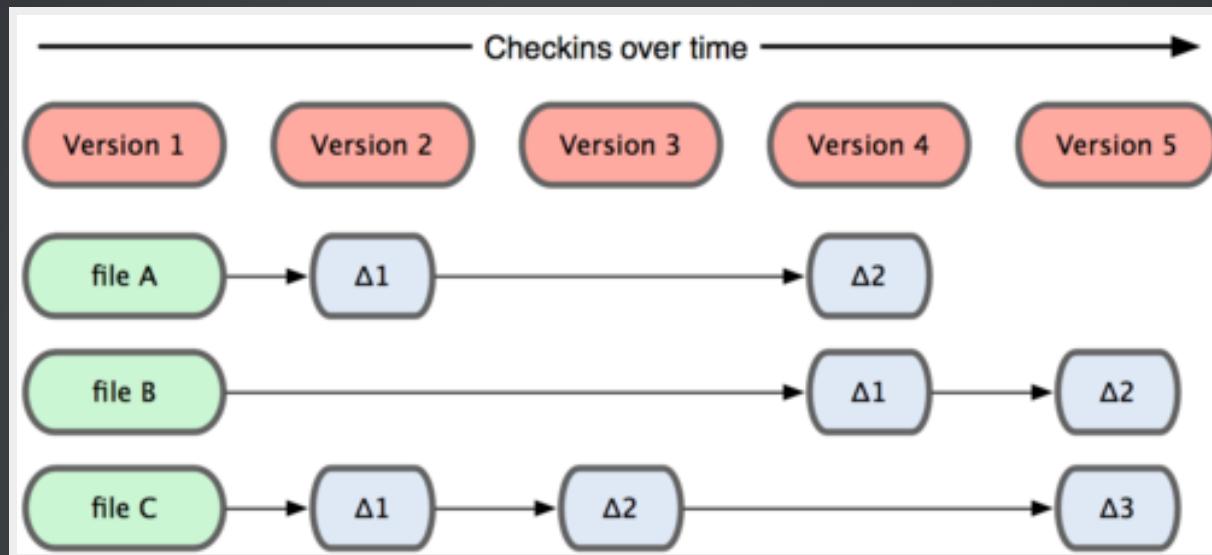
PREDECESSORS

- (SCCS) Source Code Control System
1972, closed source (FIFO)
- (RCS) Revision Control System
1982, open source (LIFO)
- (CVS) Concurrent Versions System
1986-1990, open source (+users on 1 file)
- (SVN) Apache Subversion
2000, open source (+non-txt-files)
- (SCM) BitKeeper
2000, closed source, proprietary
free 'community version' till april 2005
(used for linux kernel 2002-2005)

WHY USE GIT?

- To check edits (code)
 - review history log of changes
 - view diff between versions
 - retrieve old versions
- To share changes with collabs

GIT BASICS



!!! GIT CAN NOT TRACK !!!

- Non-text files
 - binary files (imgs etc..)
- Word Processing files
- Spreadsheets
- PDF's

GIT WHO?

- People NOT afraid of command-line tools
- Programmers & developers
- mainly used languages:
 - HTML
 - CSS
 - JavaScript
 - ...

HANDS ON..?



SET UP GIT

DOWNLOAD:

<http://git-scm.com/downloads>

INSTALL

See guidelines [here](#)
or following slides to view the set up process
(Windows)

 Git Setup

Welcome to the Git Setup Wizard

This will install Git version 1.7.9-preview20120201 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.



Next >

Cancel

Git Setup



Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change

<http://msysgit.googlecode.com/>

< Back

Next >

Cancel

Git Setup

Select Destination Location

Where should Git be installed?



Setup will install Git into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files (x86)\Git

[Browse...](#)

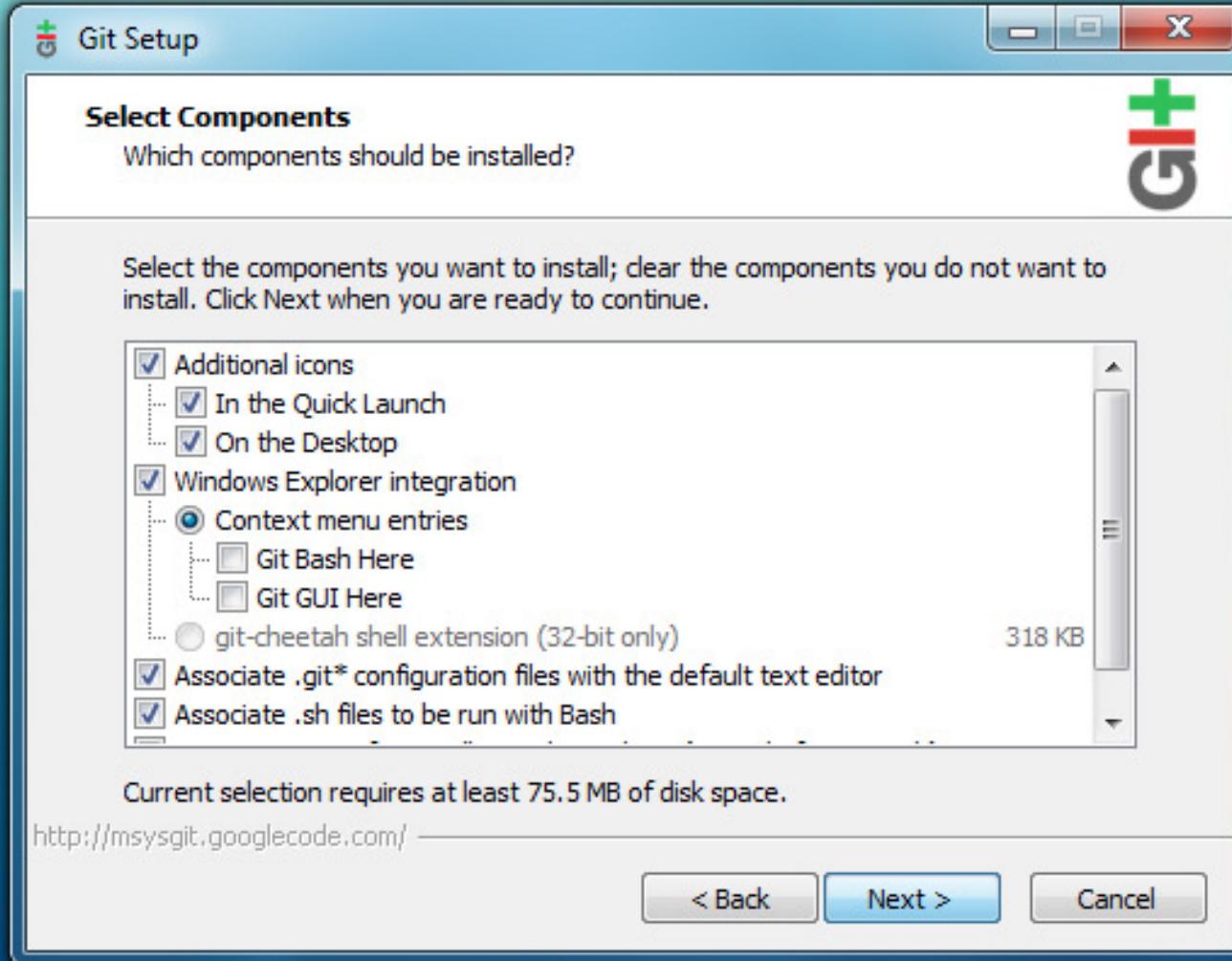
At least 75.5 MB of free disk space is required.

<http://msysgit.googlecode.com/>

< Back

[Next >](#)

Cancel



Git Setup

Select Start Menu Folder

Where should Setup place the program's shortcuts?



Setup will create the program's shortcuts in the following Start Menu folder.

Git

[Browse...](#)

To continue, click Next. If you would like to select a different folder, click Browse.

Git

[Browse...](#)



Don't create a Start Menu folder

<http://msysgit.googlecode.com/>

[< Back](#)

[Next >](#)

[Cancel](#)

Git Setup

Adjusting your PATH environment

How would you like to use Git from the command line?



Use Git Bash only

This is the most conservative choice if you are concerned about the stability of your system. Your PATH will not be modified.

Run Git from the Windows Command Prompt

This option is considered safe and no conflicts with other tools are known. Only Git will be added to your PATH. Use this option if you want to use Git from a Cygwin Prompt (make sure to not have Cygwin's Git installed).

Run Git and included Unix tools from the Windows Command Prompt

Both Git and its accompanying Unix tools will be added to your PATH.

Warning: This will override Windows tools like find.exe and sort.exe. Select this option only if you understand the implications.

<http://msysgit.googlecode.com/>

< Back

Next >

Cancel

Git Setup

Configuring the line ending conversions

How should Git treat line endings in text files?



Checkout Windows-style, commit Unix-style line endings

Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").

Checkout as-is, commit Unix-style line endings

Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform project: this is the recommended setting on Unix ("core.autocrlf" is set to "input").

Checkout as-is, commit as-is

Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").

<http://msysgit.googlecode.com/>

< Back

Next >

Cancel

Git Setup

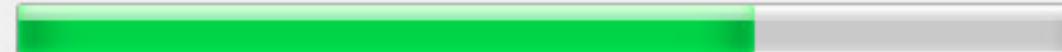
Installing

Please wait while Setup installs Git on your computer.



Extracting files...

C:\Program Files (x86)\Git\lib\tcl8.5\encoding\big5.enc



<http://msysgit.googlecode.com/>

[Cancel](#)

 Git Setup

Completing the Git Setup Wizard

Setup has finished installing Git on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

View ReleaseNotes.rtf



Finish

COME WITH US THEY SAID



ITS GONNA BE FUN THEY SAID

GIT UP SET

Create an account on: <https://github.com/signup/free>

username:

```
$ git config --global user.name "Your name here"  
# Sets the default name for git to use when you commit
```

email:

```
$ git config --global user.email "Your email here"  
# Sets the default email for git to use when you commit
```

GIT BASH

```
# some BASH commands you will use quite often from now on:  
$ ls #show contents of directory you are in  
$ cd Documents  
$ mkdir git_mb  
$ cd git_mb  
$ git init #initialise this working directory as a new repo  
$ touch README.md  
$ vi README.md  
# you can edit this file in any text based editor like notepad
```

GIT STATUS

```
Martin@MB-PC ~/Documents/git_mb (master)
$ git status
# On branch master
#
# Initial Commit
#
# Untracked files:
#       (use "git add <file>..." to include in what will be committed)
#
#       README.md

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

Martin@MB-PC ~/Documents/git_mb (master)
$
```

GIT ADD

```
Martin@MB-PC ~/Documents/git_mb (master)
$ git add README.md

Martin@MB-PC ~/Documents/git_mb (master)
$ git status
# On branch master
#
# Initial Commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file:README.md
#
Martin@MB-PC ~/Documents/git_mb (master)
$
```

GIT COMMIT

```
Martin@MB-PC ~/Documents/git_mb (master)
$ git commit -m 'Initialise repo and creation of readme file'
[master (root-commit) e881faa] initialise repo and creation
of readme file
1 file changed, 1 insertion(+)
create mode 100644 README.md
Martin@MB-PC ~/Documents/git_mb (master)
$ git status
# On branch master
nothing to commit, working directory clean
Martin@MB-PC ~/Documents/git_mb (master)
$
```

GIT LOG

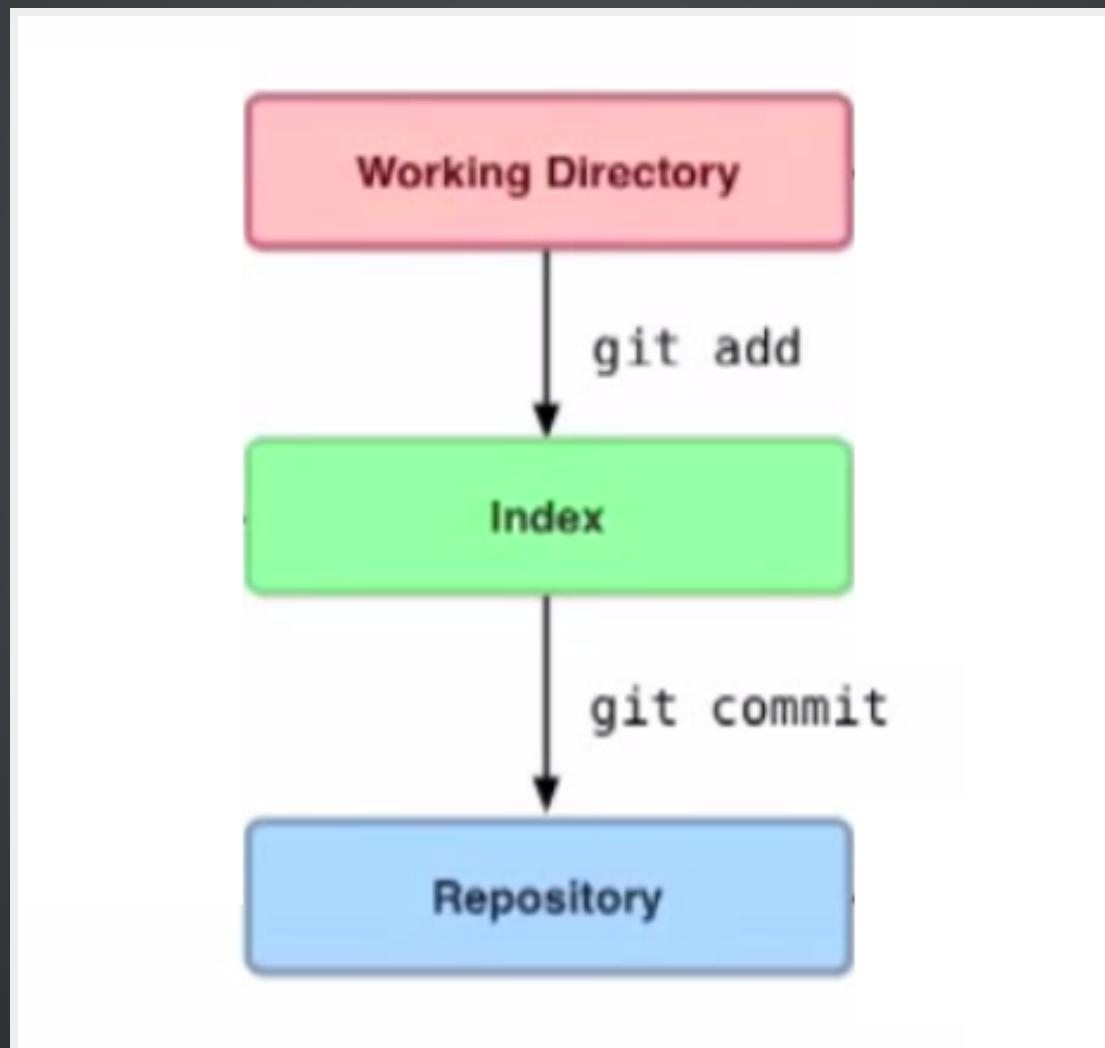
```
Martin@MB-PC ~/Documents/git_mb (master)
$ git log
commit e881faa051b1fb5e51074e05e88a744349cb97a6
Author: MBing <email@somedomain.com>
Date:   Mon Jun 3 22:38:32 2013 +0200
```

Initialise repo and creation of readme file

```
Martin@MB-PC ~/Documents/git_mb (master)
$
```

I GIT... EUH LOST?!?

Git Architecture simplified:



WHAT IS GITHUB?

- Web Based Hosting Service for software development
- Uses Git revision control system
- Free for open source repo's
- Paid plans available for private repo's
- easy to use GUI's available
- Makes the use of Git easier for most people



<https://help.github.com/>

GITHUB REPO

click [here](#) to see the online tutorial on GitHub

GITHUB - FORK A REPO

The screenshot shows a GitHub repository page for 'MBing / GIT-presentation'. The top navigation bar includes links for 'Explore', 'Gist', 'Blog', and 'Help'. On the right side of the header, there is a user profile for 'MBing' and several icons. Below the header, the repository name 'MBing / GIT-presentation' is displayed, along with a 'PUBLIC' badge. The main content area shows the repository's code, network, pull requests, issues, wiki, graphs, and settings. A prominent 'Code' tab is selected. The 'Fork' button, located in the top right of the main content area, is circled in red. A tooltip for this button reads 'Fork this repo'. Below the 'Code' tab, there are buttons for 'Clone in Windows', 'ZIP', 'HTTP', 'SSH', 'Git Read-Only', and a link to the repository's URL (<https://github.com/MBing/GIT-presentation.git>). Further down, there are buttons for 'branch: master', 'Files', 'Commits', 'Branches (1)', and 'Tags'. The 'Commits' section shows two commits, both authored by 'MBing' and dated '3 days ago'. The first commit is titled 'Update README.md'. The second commit is titled 'README.md'. The commit details show 'latest commit ad764dc573' and a link to view the commit. At the bottom of the page, there is a large preview of the 'README.md' file, which contains the text 'GIT-presentation'.

GITHUB CLONE

The screenshot shows a GitHub repository page for 'MBing / GIT-presentation'. The top navigation bar includes links for 'Explore', 'Gist', 'Blog', and 'Help'. The repository details show it's a 'PUBLIC' repo with 0 stars and 0 forks. A red oval highlights the cloning options at the bottom of the main content area, which include 'Clone in Windows', 'ZIP', 'HTTP', 'SSH', 'Git Read-Only', and a URL 'https://github.com/MBing/GIT-presentation.git'. Another red oval highlights the 'Read+Write access' button next to the cloning options. Below this, the repository structure shows a single file 'README.md' with a commit history. The commit history lists an update to 'README.md' by MBing 3 days ago, with the latest commit hash being 'ad764dc573'. The README content is displayed as 'GIT-presentation'.

This is your own folder, not the original repo!!
(look for your username in the git uri)

GIT CLONE

```
Martin@MB-PC ~/Documents/git_mb (master)
$ git clone https://github.com/yourusername/GIT-presentation.git
Cloning into 'GIT-presentation'...
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (6/6), done.

Martin@MB-PC ~/Documents/git_mb (master)
$ ls
GIT-presentation
$ cd GIT-presentation/
$ ls
README.md

Martin@MB-PC ~/Documents/git_mb (master)
$
```

GIT ADD NEW FILE

```
Martin@MB-PC ~/Documents/GIT-presentation (master)
$ touch yourname.html

Martin@MB-PC ~/Documents/GIT-presentation (master)
$ git status
# On branch master
# Untracked files:
#       (use "git add <file> ..." to include in what will be committed)
#
#           yourname.html
nothing added to commit but untracked files present (use git
"git add" to track)

Martin@MB-PC ~/Documents/GIT-presentation (master)
$
```

GIT ADD

```
Martin@MB-PC ~/Documents/GIT-presentation (master)
```

```
$ git add yourname.html
```

```
Martin@MB-PC ~/Documents/GIT-presentation (master)
```

```
$ git status
```

```
# On branch master
```

```
# Changes to be committed:
```

```
#       (use "git reset HEAD <file>..." to unstage)
```

```
#
```

```
#       new file:    yourname.html
```

```
#
```

```
Martin@MB-PC ~/Documents/GIT-presentation (master)
```

```
$
```

GIT COMMIT

```
Martin@MB-PC ~/Documents/GIT-presentation (master)
$ git commit -m 'adding an html file with my name'
[master 328e0cb] adding an html file with my name
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 yourname.html

Martin@MB-PC ~/Documents/GIT-presentation (master)
$ git status
# On branch master
# Your branch is ahead of 'origin/master' by 1 commit.
#       (use "git push" to publish your local commits)
#
nothing to commit, working directory clean
Martin@MB-PC ~/Documents/GIT-presentation (master)
$
```

GIT PUSH

```
Martin@MB-PC ~/Documents/GIT-presentation (master)
$ git push origin master
Counting objects: 4 done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 302 bytes, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/yourname/GIT-presentation.git
ad764dc..328e0cb master -> master
Martin@MB-PC ~/Documents/GIT-presentation (master)
$ git status
# On branch master
nothing to commit, working directory clean
Martin@MB-PC ~/Documents/GIT-presentation (master)
$
```

GO AND WATCH GITHUB

The screenshot shows a GitHub repository page for 'MBing / GIT-presentation'. The repository is public and contains 3 commits. A red circle highlights the commit for 'yourname.html'.

Repository Details:

- Name:** MBing / GIT-presentation
- Owner:** MBing
- Public:** PUBLIC
- Last Commit:** 9 minutes ago
- Commits:** 3 commits

Commit List:

| File | Date | Message |
|---------------|---------------|--|
| README.md | 3 days ago | Update README.md [MBing] |
| yourname.html | 9 minutes ago | adding an html file with my name [MBing] |

OTHER THINGS YOU CAN DO WITH GIT

- fetch
- merge
- checkout
- branch
- diff
- pull
- show
- reset
- ...

RESOURCES

The Pro Git book by Scott Chacun:
<http://git-scm.com/book>

Heroku Cheat Sheet :
[Cheat Sheet PDF file link](#)

YouTube:
[Introduction to Git with Scott Chacun](#)
[Power Your Workflow with Git \(Patrick Hogan\)](#)

GIT STARTED!

I don't always say
thank you..

