



# Introduction to Git

Jeffrey Leek  
Johns Hopkins Bloomberg School of Public Health

# What is Version Control?

“Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.”

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

# What is Version Control?

- Many of us constantly create something, save it, change it, then save it again
- Version (or revision) control is a means of managing this process in a reliable and efficient way
- Especially important when collaborating with others

[http://en.wikipedia.org/wiki/Revision\\_control](http://en.wikipedia.org/wiki/Revision_control)

# What is Git?

“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.”

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

# What is Git?

- Created by the same people who developed Linux
- The most popular implementation of version control today
- Everything is stored in local repositories on your computer
- Operated from the command line

<http://git-scm.com/book/en/Getting-Started-A-Short-History-of-Git>

# Download Git

- Go to the following website and click on the download link for your operating system (Mac, Windows, Linux, etc):

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

The screenshot shows the Git website homepage. At the top, the Git logo is followed by the tagline "--everything-is-local". A search bar is located in the top right corner. The main content area features a description of Git as a free and open source distributed version control system, highlighting its speed and efficiency. It also mentions that Git is easy to learn and has a tiny footprint with lightning fast performance. A diagram on the right side of the main content area illustrates a commit history with branches and merges. Below the main content, there are four sections: "About" (describing the advantages of Git), "Documentation" (providing links to command reference pages, Pro Git book content, videos, and other material), "Downloads" (listing GUI clients and binary releases for all major platforms), and "Community" (providing links to mailing lists, chat, development, and more). A section for "Pro Git" by Scott Chacon is also visible, noting it is available to read online for free and that dead tree versions are available on Amazon.com. The "Latest stable release" section displays "1.8.4.1" and "Released Nov 2013 (60-66)", with a "Download for Windows" button. Below this, there are links for "Windows GUIs", "Turballs", "Mac Build", and "Source Code". At the bottom, a section titled "Companies & Projects Using Git" displays logos for various organizations and projects, including Google, Facebook, Microsoft, Twitter, LinkedIn, Netflix, Amazon, and others.

# Install Git

- Once the file is done downloading, open it up to begin the Git installation



# Install Git

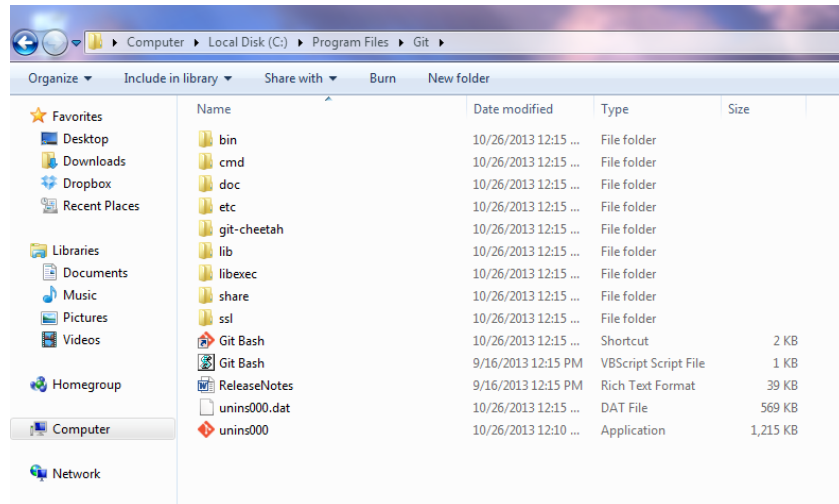
- Unless you really know what you are doing, just go with the default options at each step of the installation
- Once the install is complete, hit the "Finish" button (you may want to uncheck the box next to "Review ReleaseNotes.rtf")





# Open Git Bash

- Find a program called Git Bash, which is the command line environment for interacting with Git
- It should be located in the directory into which Git was installed (or, for Windows users, in the Start Menu)



# Open Git Bash

- Once Git Bash opens, you'll see a short welcome message followed by the name of your computer and a dollar sign on the next line
- The dollar sign means that it's your turn to type a command

```
Welcome to Git (version 1.8.4-preview20130916)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Nick@NICK-PC ~
$
```

# Configure Username and Email

- Each commit to a Git repository will be "tagged" with the username of the person who made the commit
- Enter the following commands in Git Bash, one at a time, to set your username and email:

```
$ git config --global user.name "Your Name Here"  
$ git config --global user.email "your_email@example.com"
```

- You'll only have to do this once, but you can always change these down the road using the same commands

# Configure Username and Email

- Now type the following to confirm your changes (they may be listed toward the bottom):

```
$ git config --list
```

```
Nick@NICK-PC ~  
$ git config --global user.name "John Doe"  
  
Nick@NICK-PC ~  
$ git config --global user.email "john@gmail.com"  
  
Nick@NICK-PC ~  
$ git config --list  
core.symlinks=false  
core.autocrlf=true  
color.diff=auto  
color.status=auto  
color.branch=auto  
color.interactive=true  
pack.packsizelimit=2g  
help.format=html  
http.sslcainfo=/bin/curl-ca-bundle.crt  
sendemail.smtpserver=/bin/msmtp.exe  
diff.astextplain.textconv=astextplain  
rebase.autosquash=true  
user.name=John Doe  
user.email=john@gmail.com  
  
Nick@NICK-PC ~  
$ _
```

# What's Next?

- Go ahead and close Git Bash with following command:

```
$ exit
```

- Now that Git is set up on your computer, we're ready to move on to GitHub, which is a web-based platform that lets you do some pretty cool stuff
- Once GitHub is up and running, we'll show you how to start using these tools to your benefit