

# Git Basics

Taken from a complete n00b

Steve Offutt & Joshua Loscar

OSSEM

September 9, 2016



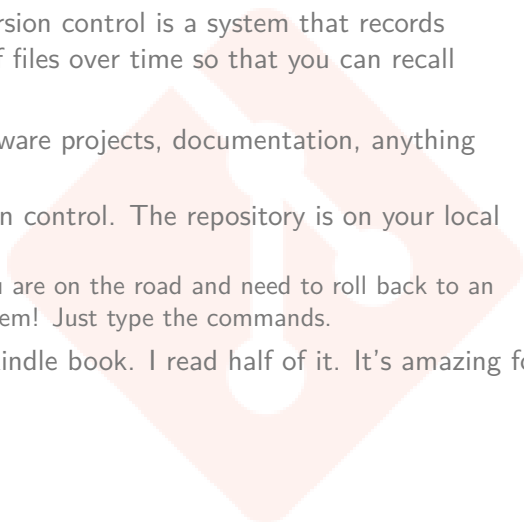
# Agenda

- ➊ What is git?
- ➋ Installing Git
  - Windows
  - Linux
  - OSX
- ➌ GUI vs CLI
- ➍ Staging?
- ➎ Basic Commands
  - init, clone, status, add
  - commit
  - branch
  - merge and push
- ➏ Who cares?
- ➐ References



# What is git?

- From Git's Website: "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."
- Keep track of stuff! Software projects, documentation, anything really.
- git is decentralized version control. The repository is on your local machine, always.
  - ▶ This means that if you are on the road and need to roll back to an older version, no problem! Just type the commands.
- "Ry's Git Tutorial" free kindle book. I read half of it. It's amazing for \$0.



# Installing Git

- Take it away Josh!



# How to use git? GUI or CLI? Choose wisely...

- The most universal way of using git is via the command line
- A lot of git GUIs are very nice. A visual really helps with branches.
- GitKraken is pretty cool looking! Cross platform and free (I think?).
- giggle and gitg both look pretty nifty as well.
- There are tons. To get you started check out <https://git-scm.com/downloads/guis>
- What makes sense for your project?



# WTF does staging in git mean?

- In git you must "stage" or "unstage" items prior to commits.
- Think of this as "setting the stage" for your commit.
  - ▶ All my stuff is set, now I'm ready to do my thing!
- This is very important when using git as to make sure everything you do or don't want added to your project is or is not added.



# Basic Commands

- `git init`
  - ▶ This is what you would use to typically start a NEW git repository.
  - ▶ Suppose you wanted to start a project called CoolProject in `/home/OSSEM/Projects/`
  - ▶ You would use the command `git init CoolProject`
  - ▶ Just like that the CoolProject directory is made along with all of the repository files/folders.
- `git clone`
  - ▶ Used to clone a repository.
  - ▶ Lets clone the DC Darknet Github defcon24 repo with `git clone https://github.com/thedarknet/defcon24.git`
  - ▶ The ENTIRE contents, including branches and commit histories, will be cloned to your working directory. How cool!
- `git status`
  - ▶ Display the state of the current branch.
  - ▶ Also displays the current branch. Just in case you forgot.
  - ▶ Files that have been added or deleted from the repo.
- `git add`
  - ▶ Simply stage files or directories.

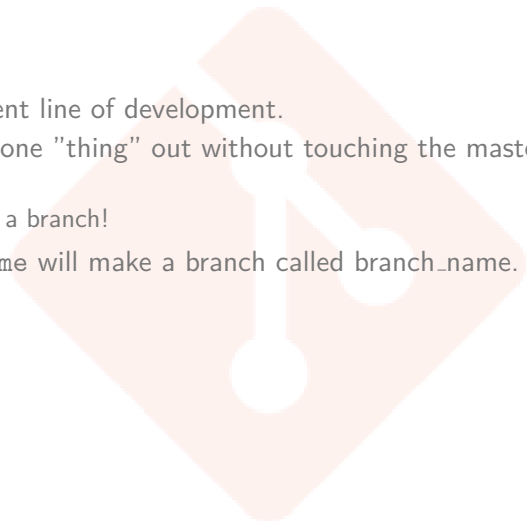
# git commit

- You must stage your items in your repo prior to committing.
- All unstaged items will not be brought into the commit.
- This command will take your current staged items and commit your changes!
- Git forces you to say something about the commit.
- If only the command `git commit` is typed, git will launch the editor of your choice defined by the `$EDITOR` environment variable
- After entering your message into the editor, git will continue with the commit.
- It is also recommended that you run `git status` prior to committing to ensure that everything is there.
- Your name and email typically should be set before submitting a commit.



# What is a branch?

- A branch is an independent line of development.
- Say you want to try this one "thing" out without touching the master branch.
  - ▶ Cool story, bro. Make a branch!
- `git branch branch_name` will make a branch called `branch_name`.



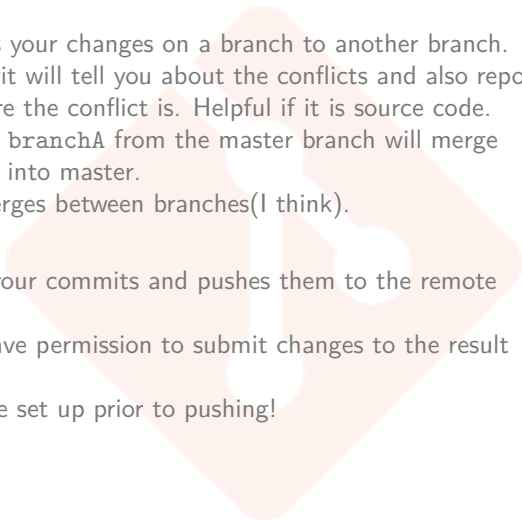
# git merge and git push

- `git merge`

- ▶ This command merges your changes on a branch to another branch.
- ▶ If there are conflicts, git will tell you about the conflicts and also report in the file exactly where the conflict is. Helpful if it is source code.
- ▶ Executing `git merge branchA` from the master branch will merge changes from branchA into master.
- ▶ This also works for merges between branches(I think).

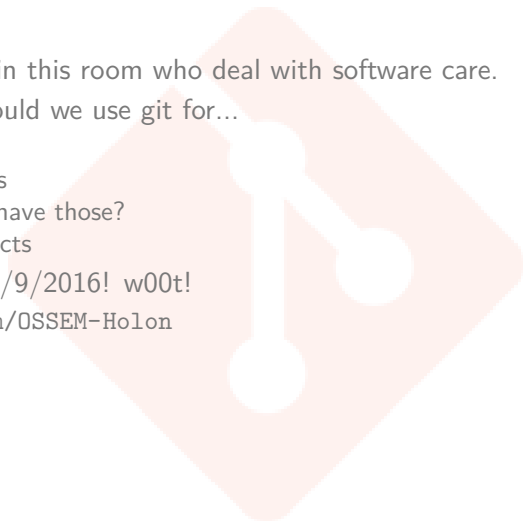
- `git push`

- ▶ This command takes your commits and pushes them to the remote repository.
- ▶ Of course you must have permission to submit changes to the result repo...
- ▶ Your config file must be set up prior to pushing!



# Who cares?

- Without a doubt people in this room who deal with software care.
- What about OSSEM? Could we use git for...
  - ▶ Badge reader
  - ▶ Presentation templates
  - ▶ By-laws. Do we even have those?
  - ▶ Keeping track of projects
- OSSEM Github started 9/9/2016! w00t!
  - ▶ <https://github.com/OSSEM-Holon>



# References. Take a look, it's in a book...

- Git's website. They have a (free)book!
  - ▶ <https://git-scm.com/book/en/v2>
- "Ry's git Tutorial" by Ryan Hudson on Amazon.com
  - ▶ <https://www.amazon.com/Rys-Git-Tutorial-Ryan-Hodson-ebook/dp/B00QFIA50C/>

