

Git - An Introduction

Kurtis Nusbaum

College of St. Benedict, St. John's University

December 9, 2013

Traditional Version Control

Traditional Version Control

- ▶ One central repository

Traditional Version Control

- ▶ One central repository
- ▶ All commits are remote

Traditional Version Control

- ▶ One central repository
- ▶ All commits are remote
- ▶ Branching and merging are generally painful

Traditional Version Control

- ▶ One central repository
- ▶ All commits are remote
- ▶ Branching and merging are generally painful
- ▶ Examples:
 - ▶ CVS
 - ▶ SVN

Distributed Version Control

Distributed Version Control

- ▶ Everyone has their own repository

Distributed Version Control

- ▶ Everyone has their own repository
- ▶ All commits are local
 - ▶ You can work offline

Distributed Version Control

- ▶ Everyone has their own repository
- ▶ All commits are local
 - ▶ You can work offline
- ▶ Branching and merging are strongly encouraged!

Distributed Version Control

- ▶ Everyone has their own repository
- ▶ All commits are local
 - ▶ You can work offline
- ▶ Branching and merging are strongly encouraged!
- ▶ New concepts of push and pull

Distributed Version Control

- ▶ Everyone has their own repository
- ▶ All commits are local
 - ▶ You can work offline
- ▶ Branching and merging are strongly encouraged!
- ▶ New concepts of push and pull
- ▶ Examples:
 - ▶ Git
 - ▶ Mercurial
 - ▶ Bazaar

Basic Git commands

Basic Git commands

- ▶ Creating Git repository
 - ▶ kurtis% cd projectdir
 - ▶ kurtis% git init

Basic Git commands

- ▶ Creating Git repository
 - ▶ `kurtis% cd projectdir`
 - ▶ `kurtis% git init`
- ▶ Adding/Removing Files
 - ▶ `kurtis% git add filename`
 - ▶ When adding a directory, all files in directory are also added
 - ▶ `kurtis% git rm filename`

Basic Git commands

- ▶ Creating Git repository
 - ▶ `kurtis% cd projectdir`
 - ▶ `kurtis% git init`
- ▶ Adding/Removing Files
 - ▶ `kurtis% git add filename`
 - ▶ When adding a directory, all files in directory are also added
 - ▶ `kurtis% git rm filename`
- ▶ Committing
 - ▶ `kurtis% git commit -a`
 - ▶ Without the `-a` no changes will be committed!

Branching and Merging

Branching and Merging

- ▶ By default you are on the "master" branch

Branching and Merging

- ▶ By default you are on the "master" branch
- ▶ To create a new branch and switch to it:
 - ▶ `kurtis% git branch branchName`
 - ▶ `kurtis% git checkout branchName`

Branching and Merging

- ▶ By default you are on the "master" branch
- ▶ To create a new branch and switch to it:
 - ▶ `kurtis% git branch branchName`
 - ▶ `kurtis% git checkout branchName`
- ▶ Merge changes back to master branch

Branching and Merging

- ▶ By default you are on the "master" branch
- ▶ To create a new branch and switch to it:
 - ▶ `kurtis% git branch branchName`
 - ▶ `kurtis% git checkout branchName`
- ▶ Merge changes back to master branch
 - ▶ `kurtis% git checkout master`

Branching and Merging

- ▶ By default you are on the "master" branch
- ▶ To create a new branch and switch to it:
 - ▶ `kurtis% git branch branchName`
 - ▶ `kurtis% git checkout branchName`
- ▶ Merge changes back to master branch
 - ▶ `kurtis% git checkout master`
 - ▶ `kurtis% git merge branchName`

Branching and Merging

- ▶ By default you are on the "master" branch
- ▶ To create a new branch and switch to it:
 - ▶ `kurtis% git branch branchName`
 - ▶ `kurtis% git checkout branchName`
- ▶ Merge changes back to master branch
 - ▶ `kurtis% git checkout master`
 - ▶ `kurtis% git merge branchName`
 - ▶ To delete branch
 - ▶ `kurtis% git branch -d branchName`

My Typical Git Workflow

My Typical Git Workflow

- ▶ Master branch is always clean and working

My Typical Git Workflow

- ▶ Master branch is always clean and working
- ▶ When I want to try a new idea, create a branch

My Typical Git Workflow

- ▶ Master branch is always clean and working
- ▶ When I want to try a new idea, create a branch
- ▶ Work and commit to that branch until code is clean and working

My Typical Git Workflow

- ▶ Master branch is always clean and working
- ▶ When I want to try a new idea, create a branch
- ▶ Work and commit to that branch until code is clean and working
- ▶ Switch back to master and merge branch into master

My Typical Git Workflow

- ▶ Master branch is always clean and working
- ▶ When I want to try a new idea, create a branch
- ▶ Work and commit to that branch until code is clean and working
- ▶ Switch back to master and merge branch into master
- ▶ Delete branch

Sharing Your Repository

Sharing Your Repository

- ▶ Get someone's repository
 - ▶ `steve% git clone steve@computer:/path/to/git/repo`
 - ▶ Other persons git repository must have read access

Sharing Your Repository

- ▶ Get someone's repository
 - ▶ `steve% git clone steve@computer:/path/to/git/repo`
 - ▶ Other persons git repository must have read access
- ▶ Get someone's changes

Sharing Your Repository

- ▶ Get someone's repository
 - ▶ `steve% git clone steve@computer:/path/to/git/repo`
 - ▶ Other persons git repository must have read access
- ▶ Get someone's changes
 - ▶ `steve% git pull`
 - ▶ Pulls changes and merges them into the current branch.
 - ▶ The command above pulls from master branch in "origin" repository.
 - ▶ To pull an arbitrary branch from some repository:
 - ▶ `git pull repository branchName`
 - ▶ `git pull bob@othercomputer:/other/repo/path crazyBranch`

Sharing Your Repository

- ▶ Get someone's repository
 - ▶ `steve% git clone steve@computer:/path/to/git/repo`
 - ▶ Other persons git repository must have read access
- ▶ Get someone's changes
 - ▶ `steve% git pull`
 - ▶ Pulls changes and merges them into the current branch.
 - ▶ The command above pulls from master branch in "origin" repository.
 - ▶ To pull an arbitrary branch from some repository:
 - ▶ `git pull repository branchName`
 - ▶ `git pull bob@othercomputer:/other/repo/path crazyBranch`
 - ▶ If you just want to get the branch, but not merge it, use `git fetch`

Sharing Your Repository

- ▶ Get someone's repository
 - ▶ `steve% git clone steve@computer:/path/to/git/repo`
 - ▶ Other persons git repository must have read access
- ▶ Get someone's changes
 - ▶ `steve% git pull`
 - ▶ Pulls changes and merges them into the current branch.
 - ▶ The command above pulls from master branch in "origin" repository.
 - ▶ To pull an arbitrary branch from some repository:
 - ▶ `git pull repository branchName`
 - ▶ `git pull bob@othercomputer:/other/repo/path crazyBranch`
 - ▶ If you just want to get the branch, but not merge it, use `git fetch`
 - ▶ `steve% git fetch repository branchName`

Sharing Your Repository

- ▶ Get someone's repository
 - ▶ `steve% git clone steve@computer:/path/to/git/repo`
 - ▶ Other persons git repository must have read access
- ▶ Get someone's changes
 - ▶ `steve% git pull`
 - ▶ Pulls changes and merges them into the current branch.
 - ▶ The command above pulls from master branch in "origin" repository.
 - ▶ To pull an arbitrary branch from some repository:
 - ▶ `git pull repository branchName`
 - ▶ `git pull bob@othercomputer:/other/repo/path crazyBranch`
 - ▶ If you just want to get the branch, but not merge it, use `git fetch`
 - ▶ `steve% git fetch repository branchName`
 - ▶ Other person could also push changes to you.

Things to Keep in Mind

Things to Keep in Mind

- ▶ Any branch can be merged to any other branch
 - ▶ Careful, this might get confusing

Things to Keep in Mind

- ▶ Any branch can be merged to any other branch
 - ▶ Careful, this might get confusing
- ▶ Must commit before switching branches

Things to Keep in Mind

- ▶ Any branch can be merged to any other branch
 - ▶ Careful, this might get confusing
- ▶ Must commit before switching branches
- ▶ Unless you add it, a file will not be tracked.

Things to Keep in Mind

- ▶ Any branch can be merged to any other branch
 - ▶ Careful, this might get confusing
- ▶ Must commit before switching branches
- ▶ Unless you add it, a file will not be tracked.
- ▶ If you want to ignore a file, add it to the `.git/info/exclude` file.

Things to Keep in Mind

- ▶ Any branch can be merged to any other branch
 - ▶ Careful, this might get confusing
- ▶ Must commit before switching branches
- ▶ Unless you add it, a file will not be tracked.
- ▶ If you want to ignore a file, add it to the `.git/info/exclude` file.
- ▶ To revert a file back to it's state at the last commit:
 - ▶ `git checkout fileName`

Things to Keep in Mind

- ▶ Any branch can be merged to any other branch
 - ▶ Careful, this might get confusing
- ▶ Must commit before switching branches
- ▶ Unless you add it, a file will not be tracked.
- ▶ If you want to ignore a file, add it to the `.git/info/exclude` file.
- ▶ To revert a file back to it's state at the last commit:
 - ▶ `git checkout fileName`
- ▶ To revert the entire repo to it's state at the last commit:
 - ▶ `git reset --hard HEAD`
 - ▶ CAREFULL! You'll loose all your changes since your last commit!

Things to Keep in Mind

- ▶ Any branch can be merged to any other branch
 - ▶ Careful, this might get confusing
- ▶ Must commit before switching branches
- ▶ Unless you add it, a file will not be tracked.
- ▶ If you want to ignore a file, add it to the `.git/info/exclude` file.
- ▶ To revert a file back to it's state at the last commit:
 - ▶ `git checkout fileName`
- ▶ To revert the entire repo to it's state at the last commit:
 - ▶ `git reset --hard HEAD`
 - ▶ CAREFULL! You'll loose all your changes since your last commit!
- ▶ Sick of typing out long repository names? Try the remote add command!

Things to Keep in Mind

- ▶ Any branch can be merged to any other branch
 - ▶ Careful, this might get confusing
- ▶ Must commit before switching branches
- ▶ Unless you add it, a file will not be tracked.
- ▶ If you want to ignore a file, add it to the `.git/info/exclude` file.
- ▶ To revert a file back to it's state at the last commit:
 - ▶ `git checkout fileName`
- ▶ To revert the entire repo to it's state at the last commit:
 - ▶ `git reset --hard HEAD`
 - ▶ CAREFULL! You'll loose all your changes since your last commit!
- ▶ Sick of typing out long repository names? Try the remote add command!
 - ▶ `kurtis% git remote add steve
steve@somecomputer:/path/to/repo`

Things to Keep in Mind

- ▶ Any branch can be merged to any other branch
 - ▶ Careful, this might get confusing
- ▶ Must commit before switching branches
- ▶ Unless you add it, a file will not be tracked.
- ▶ If you want to ignore a file, add it to the `.git/info/exclude` file.
- ▶ To revert a file back to it's state at the last commit:
 - ▶ `git checkout fileName`
- ▶ To revert the entire repo to it's state at the last commit:
 - ▶ `git reset --hard HEAD`
 - ▶ CAREFULL! You'll loose all your changes since your last commit!
- ▶ Sick of typing out long repository names? Try the remote add command!
 - ▶ `kurtis% git remote add steve steve@somecomputer:/path/to/repo`
 - ▶ `kurtis% git pull master steve`

Things to Keep in Mind

- ▶ Any branch can be merged to any other branch
 - ▶ Careful, this might get confusing
- ▶ Must commit before switching branches
- ▶ Unless you add it, a file will not be tracked.
- ▶ If you want to ignore a file, add it to the `.git/info/exclude` file.
- ▶ To revert a file back to it's state at the last commit:
 - ▶ `git checkout fileName`
- ▶ To revert the entire repo to it's state at the last commit:
 - ▶ `git reset --hard HEAD`
 - ▶ CAREFULL! You'll loose all your changes since your last commit!
- ▶ Sick of typing out long repository names? Try the remote add command!
 - ▶ `kurtis% git remote add steve steve@somecomputer:/path/to/repo`
 - ▶ `kurtis% git pull master steve`
- ▶ `http://git-scm.com/`

Things to Keep in Mind

- ▶ Any branch can be merged to any other branch
 - ▶ Careful, this might get confusing
- ▶ Must commit before switching branches
- ▶ Unless you add it, a file will not be tracked.
- ▶ If you want to ignore a file, add it to the `.git/info/exclude` file.
- ▶ To revert a file back to it's state at the last commit:
 - ▶ `git checkout fileName`
- ▶ To revert the entire repo to it's state at the last commit:
 - ▶ `git reset --hard HEAD`
 - ▶ CAREFULL! You'll loose all your changes since your last commit!
- ▶ Sick of typing out long repository names? Try the remote add command!
 - ▶ `kurtis% git remote add steve steve@somecomputer:/path/to/repo`
 - ▶ `kurtis% git pull master steve`
- ▶ <http://git-scm.com/>
- ▶ Google is your friend