

# Introducing Git

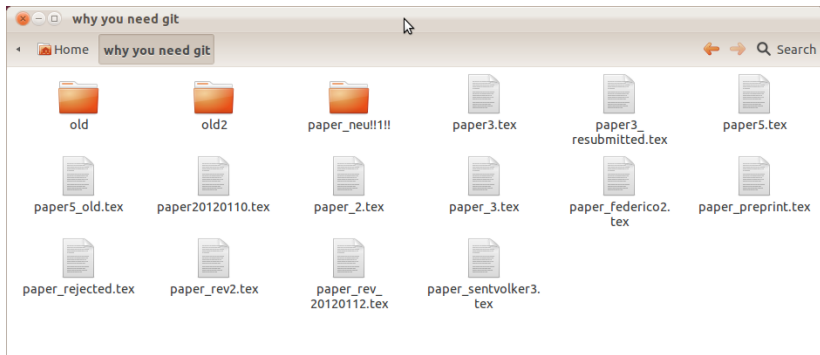
Christoph Conrads    Cornelia Gamst

Fachgebiet Numerische Mathematik  
Institut für Mathematik  
Technische Universität Berlin

Oct 20, 2014



# Today's Topic



Source: André Gaul, Jan Heiland, and Nico Schlömer

# Overview

- What is Git?
- Why use Git?
- Getting started with Git
- Hands-on exercise
- Hints

# What is Git?

## What is Git?

Git is a distributed version control software (VCS).

# What is Git?

## What is Git?

Git is a distributed version control software (VCS).

## What is Git?

Git is a software that allows you to manage different revisions of files, for example, your  $\text{\LaTeX}$  documents and your source code.

# Why use Git?

- Keep the history of a file
- Keep multiple versions of a file
- Coordinate multiple authors editing the same file
- Editing files is independent from the Git file management
- No central server needed

# Why do we use Git?

- Edit files without worrying about losing changes
- Track changes of files

# Git Terminology

## Commit

A “commit” is a set of changes saved by Git. If you save changes to files or if you want to restore old files, then you are looking for a certain “commit”. Commits are identified by their associated hash.



# Git Terminology

## Commit

A “commit” is a set of changes saved by Git. If you save changes to files or if you want to restore old files, then you are looking for a certain “commit”. Commits are identified by their associated hash.

## History

The “history” is the list of revisions saved in Git. The history consists of a list of commits.

# Getting Started

- Go to the directory with your files
- Type `git init`
- Add all files with `git add .`
- Save all files in Git with `git commit`
- Start working

# Workflow

- Edit files
- Review changes with `git diff`
- Discard changes with `git checkout <file>`
- Add new and edited files with `git add <file>`
- Save changes with `git commit`

# Hands-on Exercise

# Hints

## Git and Backups

A Git repository is not a backup.

# Hints

## Git and Backups

A Git repository is not a backup.

## Git and Binary Files

With text files, Git saves only the differences to the previous commit.

With binary files, Git saves the whole file.

Committing PDFs or executables will make your repository grow quickly.

# Hints

## Git and Backups

A Git repository is not a backup.

## Git and Binary Files

With text files, Git saves only the differences to the previous commit. With binary files, Git saves the whole file. Committing PDFs or executables will make your repository grow quickly.

## Linear History

Prefer to keep a linear history.

# Good Commit Messages

- High-level description of what you changed
- Maybe write why you changed something
- One logic change – one commit



# Formatting Commit Messages

These points ensure that your commit messages can be processed automatically.

- Summary with max. 50 characters in first line
- Blank line after summary
- Maximal 72 characters in all other lines
- Use present tense

# Getting Help

- Paste error messages into Google
- Read the Git book: <http://git-scm.com/book>
- Read a Git tutorial: <http://githowto.com>