A short introduction to Git (not GIT)

Pieter Belmans

August 1 2017

Goal

1. explain what Git is
2. explain how it's useful for writing papers
3. explain how it's useful for the Stacks project

Goal

1. explain what Git is
2. explain how it's useful for writing papers
3. explain how it's useful for the Stacks project

From the website:

> *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.*

Goal

1. explain what Git is
2. explain how it's useful for writing papers
3. explain how it's useful for the Stacks project

From the website:

> *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.*

No time to give detailed instructions on how to use it, unfortunately: motivation, and quick demo.

What is Git?

Think of it as a *much* better form of Dropbox (but it's not that at all to be honest):

What is Git?

Think of it as a *much* better form of Dropbox (but it's not that at all to be honest):

1. `paper.tex`, `paperv2.tex`,
   `paperv3.tex`,
   `paperv3final.tex`,
   `paperv3submitted.tex`,
   ... (and all the temporary files)

What is Git?

Think of it as a *much* better form of Dropbox (but it's not that at all to be honest):

1. `paper.tex`, `paperv2.tex`,
   `paperv3.tex`,
   `paperv3final.tex`,
   `paperv3submitted.tex`,
   ... (and all the temporary files)
2. no description of changes

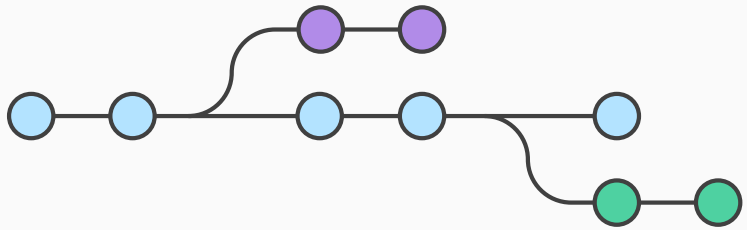
What is Git?

Think of it as a *much* better form of Dropbox (but it's not that at all to be honest):

1. `paper.tex`, `paperv2.tex`,
   `paperv3.tex`,
   `paperv3final.tex`,
   `paperv3submitted.tex`,
   ... (and all the temporary files)
2. no description of changes
3. no good handling of conflicts
4. ...

What is Git?

Think of it as a *much* better form of Dropbox (but it's not that at all to be honest):

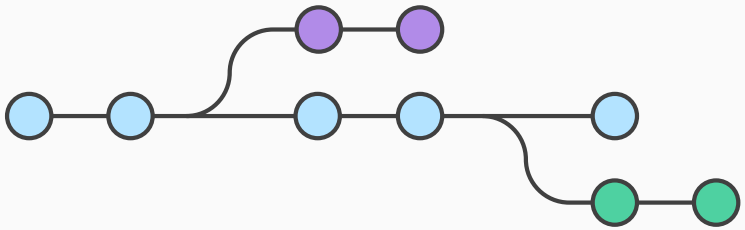


1. `paper.tex`, `paperv2.tex`,
   `paperv3.tex`,
   `paperv3final.tex`,
   `paperv3submitted.tex`,
   ... (and all the temporary files)
2. no description of changes
3. no good handling of conflicts
4. ...

What is Git?

Think of it as a *much* better form of Dropbox (but it's not that at all to be honest):

1. `paper.tex`, `paperv2.tex`, `paperv3.tex`, `paperv3final.tex`, `paperv3submitted.tex`, ... (and all the temporary files)
2. no description of changes
3. no good handling of conflicts
4. ...

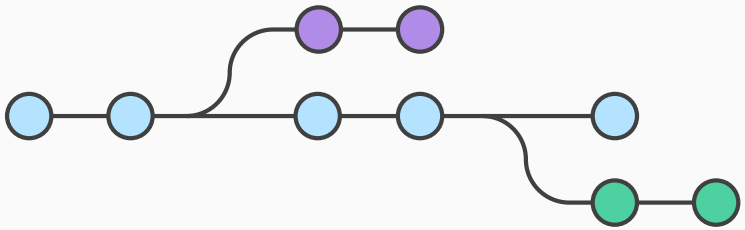


1. Dropbox is not about versioning, but Git is

What is Git?

Think of it as a *much* better form of Dropbox (but it's not that at all to be honest):

1. `paper.tex`, `paperv2.tex`, `paperv3.tex`, `paperv3final.tex`, `paperv3submitted.tex`, ... (and all the temporary files)
2. no description of changes
3. no good handling of conflicts
4. ...



1. Dropbox is not about versioning, but Git is
2. commits, branches, tags, ...
3. `git diff`

```
X$ is a regular
a squence of mo
a sequence of m
```

Using Git for writing papers

Git is a complicated beast. . .

Using Git for writing papers

Git is a complicated beast. . . Use tools!

Using Git for writing papers

Git is a complicated beast. . . Use tools!

GitHub a website for managing your projects (= repositories): collaboration, backup

GitHub Desktop a tool for interacting with your repositories

Using Git for writing papers

Git is a complicated beast... Use tools!

GitHub a website for managing your projects (= repositories): collaboration, backup

GitHub Desktop a tool for interacting with your repositories

Alternatives:

1. BitBucket (infinite private repositories!), GitLab, ...
2. SourceTree, SmartGit, ...

Using Git for writing papers

Git is a complicated beast. . . Use tools!

GitHub a website for managing your projects (= repositories): collaboration, backup

GitHub Desktop a tool for interacting with your repositories

Alternatives:

1. BitBucket (infinite private repositories!), GitLab, . . .
2. SourceTree, SmartGit, . . .

It's the best thing since sliced bread!

—Anonymous co-author of mine.

Learning Git

1. a 15 minute (interactive!) commandline tutorial:
   `https://try.github.io/`
2. a 10 minute read on using GitHub, including pull requests:
   `https://guides.github.com/activities/hello-world/`
3. ...
4. a (free!) 456 page book:
   `https://git-scm.com/book/en/v2`

An extremely short showcase

How to start collaborating on a new paper:

An extremely short showcase

How to start collaborating on a new paper: I need

- a volunteer with a GitHub account
- a title for the paper

An extremely short showcase

How to start collaborating on a new paper: I need

- a volunteer with a GitHub account
- a title for the paper

What I will do:

1. create the project
2. push to GitHub
3. add a collaborator
4. start editing

What the volunteer needs to do

1. clone the project
2. start editing

Interacting with the Stacks project

This is not required.

Interacting with the Stacks project

This is not required.

I repeat, this is not required. You can (depending on the type of change)

0. scan handwritten suggestions, and email these

Interacting with the Stacks project

This is not required.

I repeat, this is not required. You can (depending on the type of change)

0. scan handwritten suggestions, and email these
1. make comments on the website

Interacting with the Stacks project

This is not required.

I repeat, this is not required. You can (depending on the type of change)

0. scan handwritten suggestions, and email these
1. make comments on the website
2. download the TEX files, make changes, and email these to

Interacting with the Stacks project

This is not required.

I repeat, this is not required. You can (depending on the type of change)

0. scan handwritten suggestions, and email these
1. make comments on the website
2. download the TEX files, make changes, and email these to
3. use pull requests on GitHub

Interacting with the Stacks project (without Git)

For options 2 and 3:

`https://github.com/stacks/stacks-project`

Interacting with the Stacks project (without Git)

For options 2 and 3:

`https://github.com/stacks/stacks-project`

The Stacks project has big files, making some editors crash:

`http://sharelatex.com`

Interacting with the Stacks project (without Git)

For options 2 and 3:

`https://github.com/stacks/stacks-project`

The Stacks project has big files, making some editors crash:

`http://sharelatex.com`

The Stacks project uses a strict coding style:

`http://stacks.math.columbia.edu/download/coding.pdf`

Interacting with the Stacks project (with Git)

Via pull requests: see

`https://help.github.com/articles/about-pull-requests/`

Interacting with the Stacks project (with Git)

Via pull requests: see

`https://help.github.com/articles/about-pull-requests/`

fork make your own copy

clone use this copy on your computer

commit make edits, then commit

push make them public

Interacting with the Stacks project (with Git)

Via pull requests: see
`https://help.github.com/articles/about-pull-requests/`

fork make your own copy

clone use this copy on your computer

commit make edits, then commit

push make them public

pull request suggest that your changes are pulled into the Stacks project

`https://github.com/stacks/stacks-project/pulls`

Interacting with the Stacks project (with Git)

Via pull requests: see
`https://help.github.com/articles/about-pull-requests/`

fork make your own copy

clone use this copy on your computer

commit make edits, then commit

push make them public

pull request suggest that your changes are pulled into the Stacks project

`https://github.com/stacks/stacks-project/pulls`

word of advice: use small commits (allows cherry-picking)