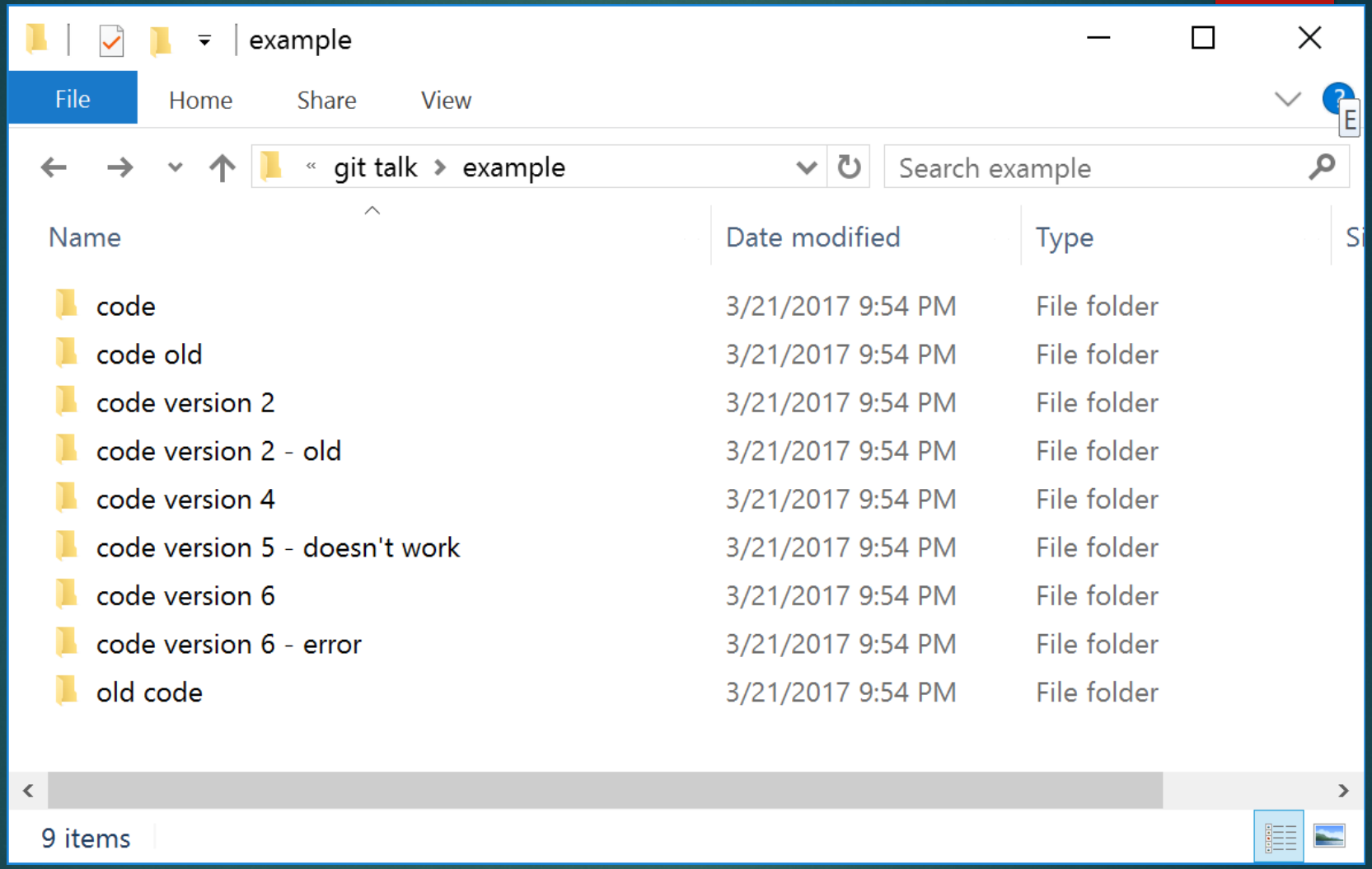
Version control

"GIT" WITH IT


example
File
Home
Share
View
git talk › example
Search example
Name
Date modified
Type
code
3/21/2017 9:54 PM
File folder
code old
3/21/2017 9:54 PM
File folder
code version 2
3/21/2017 9:54 PM
File folder
code version 2 - old
3/21/2017 9:54 PM
File folder
code version 4
3/21/2017 9:54 PM
File folder
code version 5 - doesn't work
3/21/2017 9:54 PM
File folder
code version 6
3/21/2017 9:54 PM
File folder
code version 6 - error
3/21/2017 9:54 PM
File folder
old code
3/21/2017 9:54 PM
File folder
9 items

Code breaks

- Eventually your code will break
- Time machine?

$$\frac{d^2 x^\mu}{dt^2} = -\Gamma^\mu{}_{\alpha\beta} \frac{dx^\alpha}{dt} \frac{dx^\beta}{dt} + \Gamma^0{}_{\alpha\beta} \frac{dx^\alpha}{dt} \frac{dx^\beta}{dt} \frac{dx^\mu}{dt}$$

- ... too hard

Version control

- A better way to manage code history
- Isn't undo one of your most used shortcuts?
  - What if instead of seconds it could go back years...

What is version control?

- A system that records snapshots (commits) of a set of files over time, so that you can switch between and compare different versions later

Benefits of version control

- Better development
  - Play around without breaking anything
  - Feel comfortable deleting old code
  - Have one known good and many test versions
- Tagged versions
- Synchronize multiple interdependent files
- Multiple people, multiple code versions, multiple places!

Installing git

- Download: git-scm.com
- Or demo in browser: try.github.io

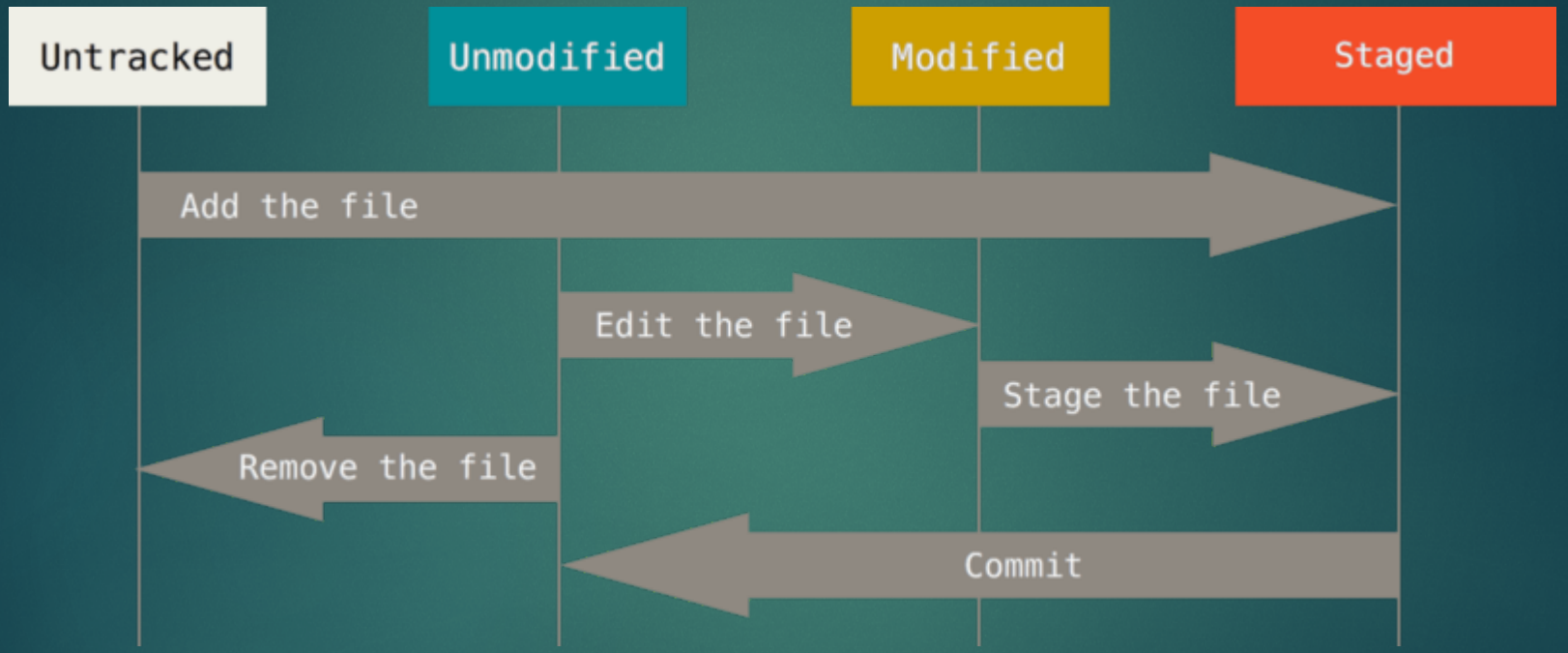
git init

- **git init <directory>**
  - Create a new repository in directory - if no directory given, uses current directory

git add

- Add files to the repository or “staging area”

Staging


Untracked
Unmodified
Modified
Staged
Add the file
Edit the file
Stage the file
Remove the file
Commit

How to get the status?

- git status
- Gives the status of all files

What's changed?

- git diff
- Status says which files changed, diff says what inside of the files is different

Committing

- Actually create a snapshot:
- git commit –m “message”
- Commit message should be descriptive

Now go play!

- Make a repo

  Open command line

  mkdir “test”

  cd “test”

  git init
- Create and add some files
- Commit

- Change some stuff
- Commit again

How does it work?

- .git directory
- Keeps track of differences

Can do

- Restore your files to any version that you *committed*
- Show you the differences between versions
- Multiple different versions seamlessly in a normal directory
- Merge together different versions of files

Cant do

- Restore your files to any version that you *didn't commit*
- Handle large and complex files well (ie illustrator, word)
  - Still useful
- Not a backup system
  - Make one (or more!) backup copies of your code, including the version control

Branches

- Why branch?
  - Development branch
  - Multiple coders

GUIs and IDEs

- Sourcetree
- Pycharm (VCS integration)

- And many others!