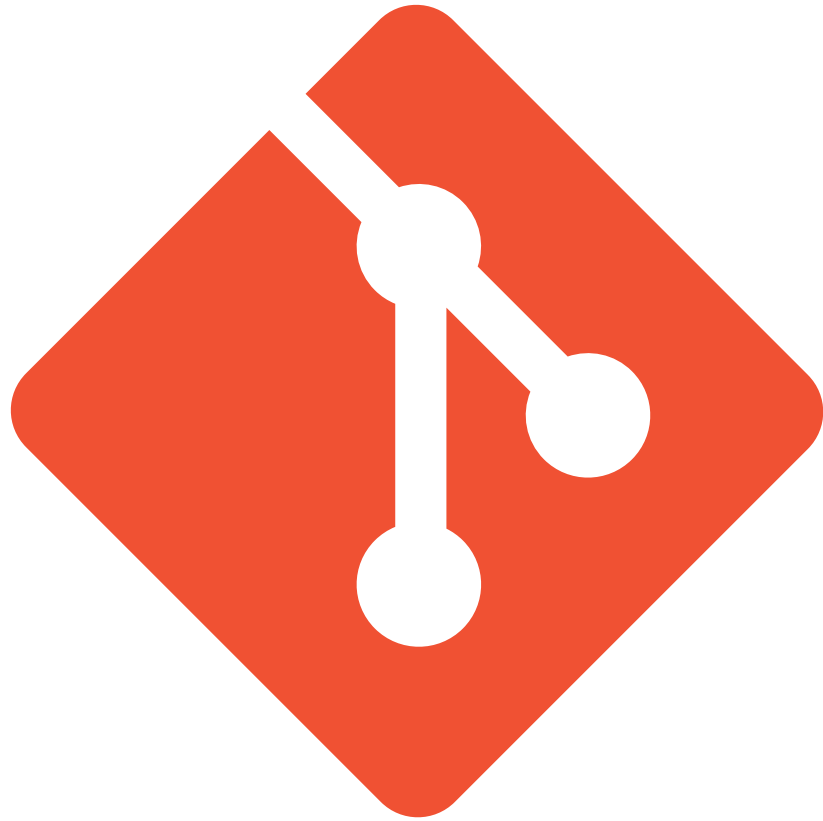


Git for beginners



git

Neil Smith @neilnjae

What is Git?

Git is a *distributed version control system*

version control system

- Keeps track of changes
- Fixing mistakes is easy

distributed

- Works with teams
- Everyone is a peer: no bosses

What is Github?

- Just a site to make sharing easy

Alternatives

Many distributed VCSs

- Mercurial (aka hg)
- Subversion (aka svn) (legacy only)

Many code sharing websites

- Gitlab
- Bitbucket

Part 1

Commit and changes

Concept: repository

Everything in the project

Entire history

Err towards including in repo

- Code
- Configuration
- Database schema
- Tests

Things not to include:

- Secrets (API keys)
- Logs
- Automatically created things (compiled or minified files, pulled-in dependencies, etc.) Look into `.gitignore`

Making a repo

Start `bash`

```
$ mkdir yourname-git-workshop  
$ cd yourname-git-workshop  
$ git init  
$ ls -a
```

Get some source data

Download your two favourite novels from Project Gutenberg

Make some files

- Create **two** text files, one for each novel
- Put the first couple of sentences of each novel in each file
- Add a title and heading *commit 1* at the top of the file
- Save the files in your Git repository directory
 - (Filenames with no spaces, all lowercase, end `.txt`)

```
# Frankenstein, by Mary Shelly
```

```
# Commit 1
```

```
You will rejoice to hear that no disaster has accompanied the  
commencement of an enterprise which you have regarded with such evil  
forebodings. I arrived here yesterday, and my first task is to assure  
my dear sister of my welfare and increasing confidence in the success  
of my undertaking.
```

Concept: commit

A **commit** is a snapshot of a project

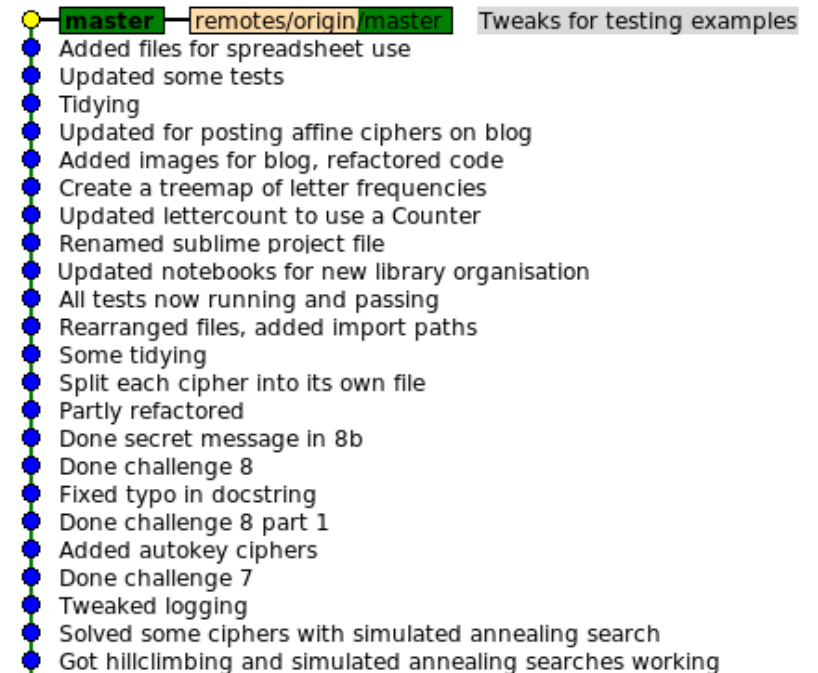
It contains all the files in all the the directories in the project

- Taken at a particular point in time
- Has a unique key (1dbb1a9)
- Exists forever in the history

Each commit knows its parent(s)

Commits are cheap (just changes from parent)

You can *always* refer to another commit, get files from it

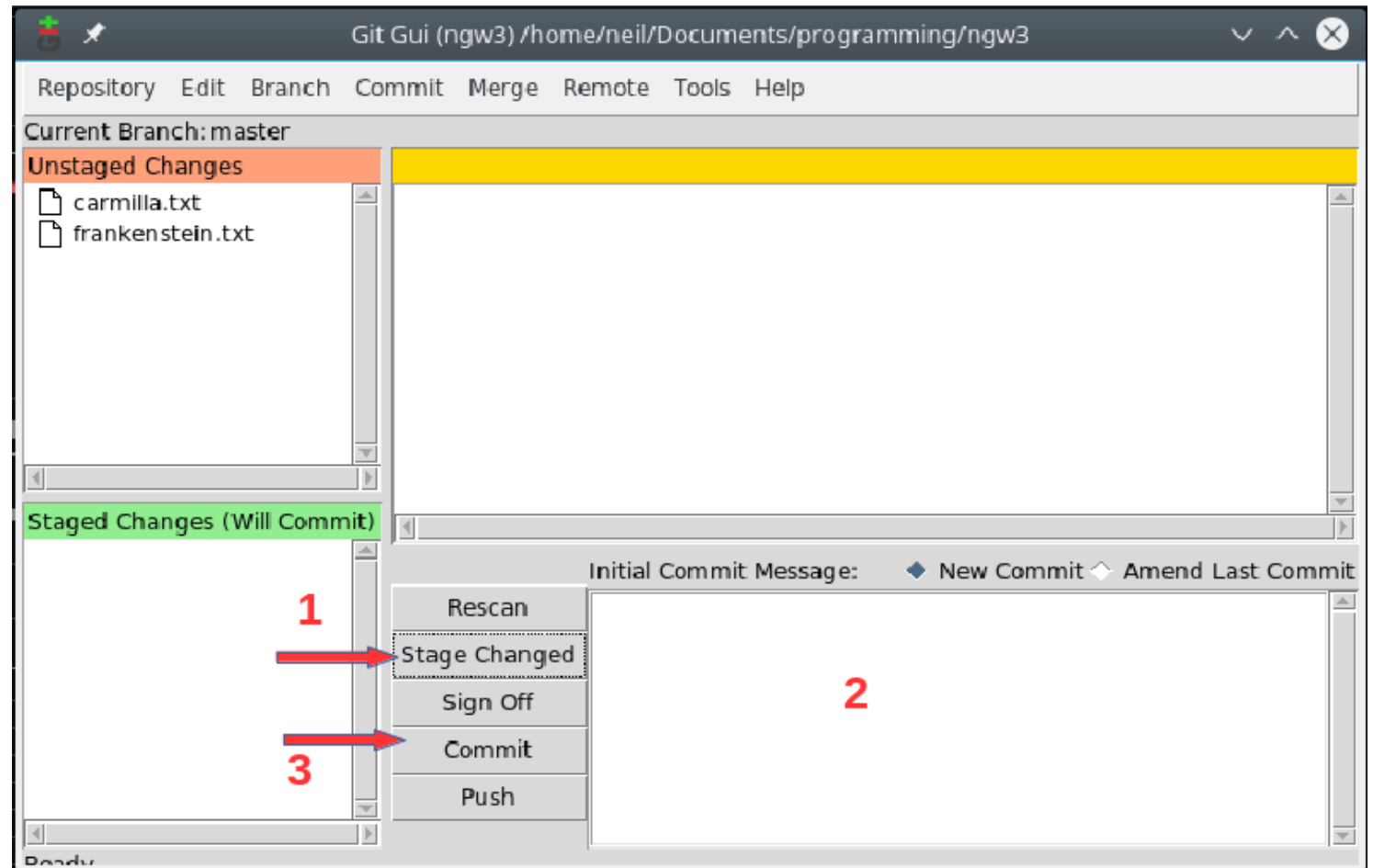


Making a first commit

Fire up `$ git gui &`

1. Stage changed
 - Confirm "stage untracked files"
2. Give a commit message, e.g. *First commit*
3. Commit

Fire up `$ gitk &`



Two more commits

- In each file, add another paragraph
- Head the paragraph with "commit *n*" heading
 - (You'll need these headings later)
- Commit it
- Add another paragraph
- Commit again



```
# Frankenstein, by Mary Shelly
```

```
# Commit 1
```

```
You will rejoice to hear that no disaster has accompanied ...
```

```
# Commit 2
```

```
I am already far north of London, and as I walk in the streets ...
```

```
# Commit 3
```

```
I try in vain to be persuaded that the pole is the seat of ...
```

Let's make a mistake!

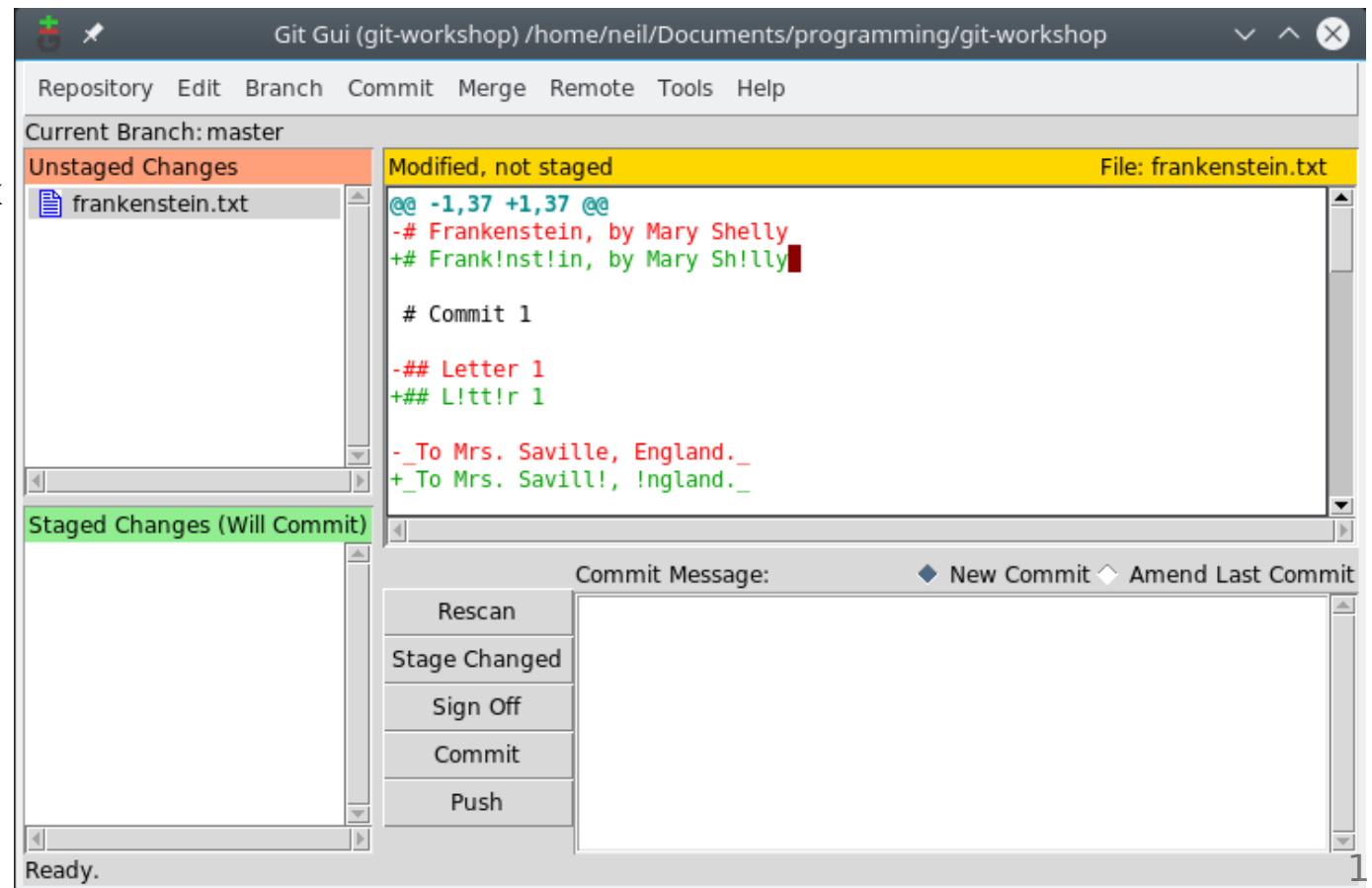
In one of your files, replace all the e with !

(This is wrong)

Save the file

Check the wrong file is on disk

- `$ cat frankenstein.txt`



Undoing a mistake

Rescan in `git-gui`

Select the file in `git-gui`

Commit -> Revert changes

(Accept the warning)

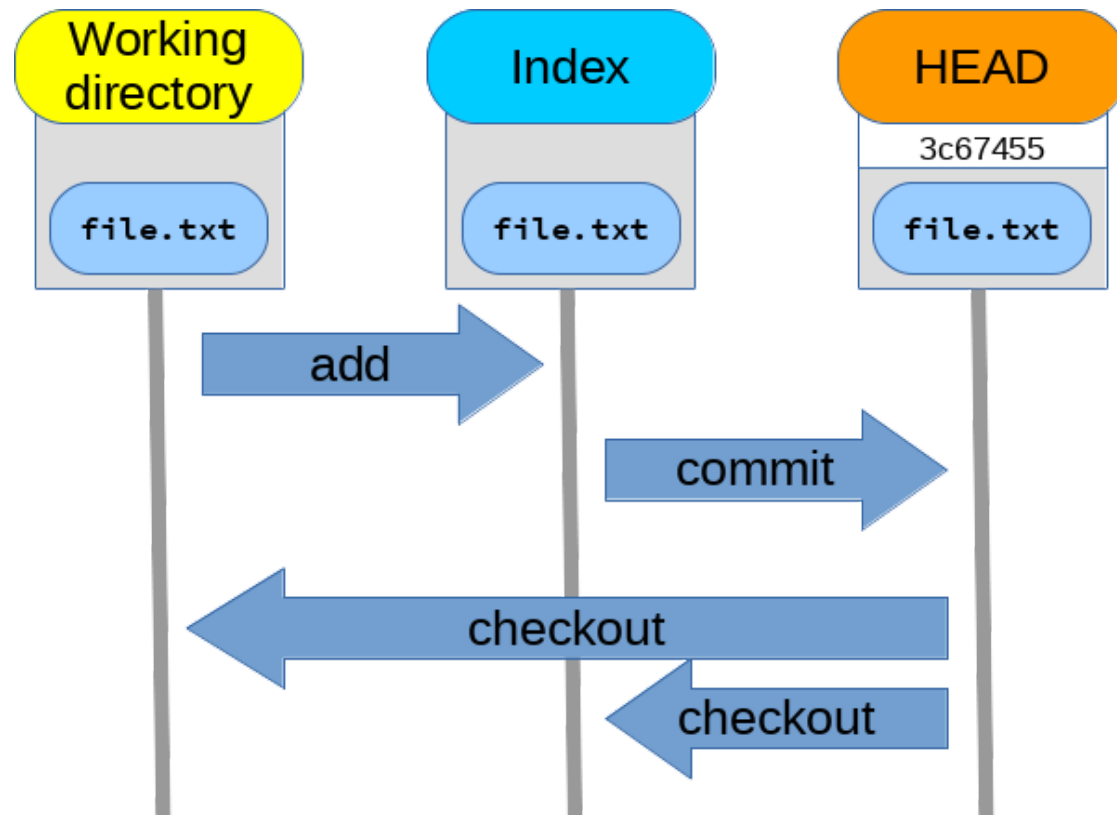
Refresh the file in the text editor

Feel happy

But what happened?

Concept: three trees

("Tree" means collection of files and folders; a "directory tree")



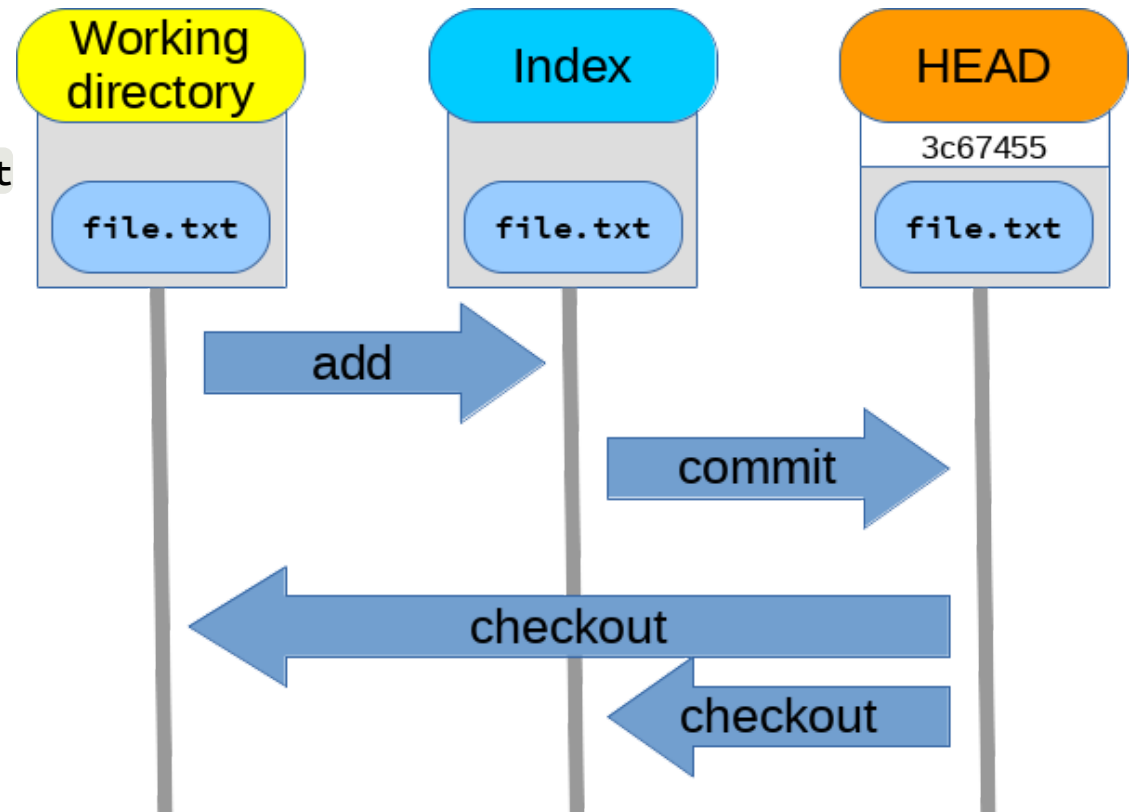
Command line fun!

1. `$ git status`
2. Replace all the e with !
3. `$ git status`
4. `$ git checkout -- frankenstein.txt`
5. `$ git status`
6. Check the file

Note: `git checkout` takes arguments

```
$ git checkout <commit> [--] <file>  
<file> ...
```

If you don't specify a commit, Git assumes you mean **HEAD**



Let's really mess things up!

1. `$ git status`
2. Replace all the e with !
3. `$ git status`
4. `$ git add frankenstein.txt`
5. `$ git status`
6. `$ git commit -m "No more vowels"`
7. `$ git status`

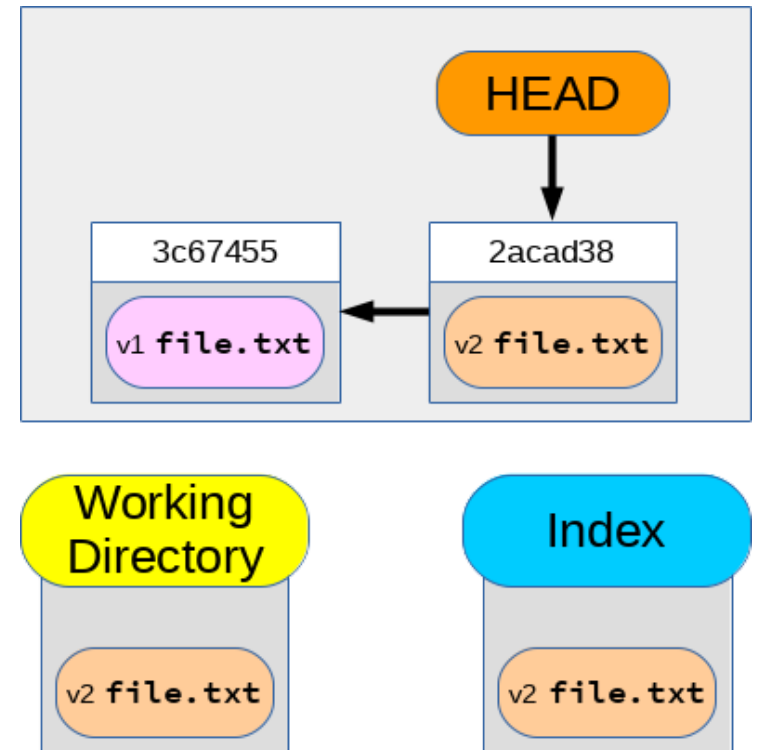
What have we done?

How do we get the good file back?

```
$ git checkout <commit> [--] <file> <file> ...
```

Hint: `HEAD~` means "parent of HEAD"

- Note difference between `~` (tilde) and `-` (dash)



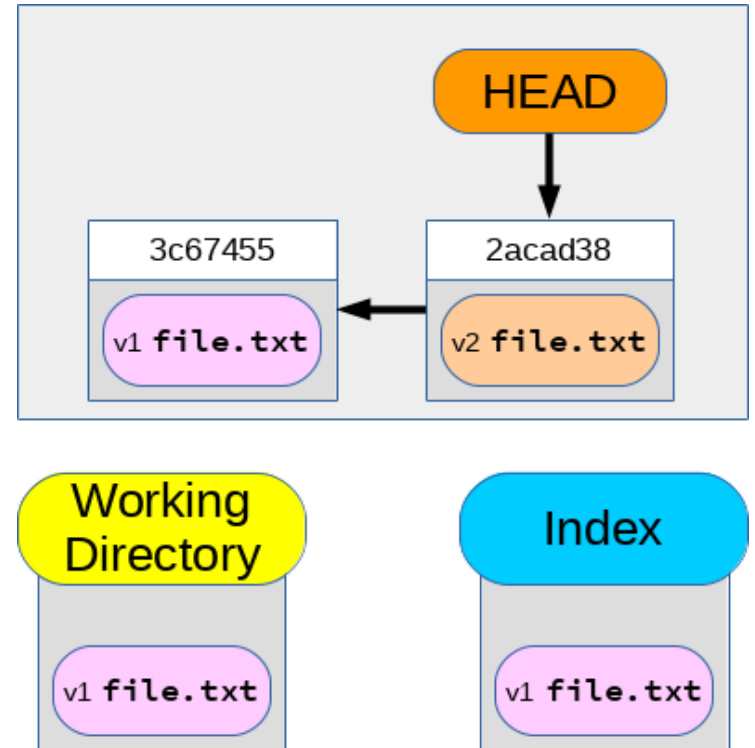
Let's really mess things up!

1. `$ git status`
2. Replace all the e with !
3. `$ git status`
4. `$ git add frankenstein.txt`
5. `$ git status`
6. `$ git commit -m "No more vowels"`
7. `$ git status`

Fixed with

```
$ git checkout HEAD~ -- frankenstein.txt
```

- `HEAD~` is "Parent of `HEAD`"
- `HEAD~~` or `HEAD~2` is "Grandparent of `HEAD`"
- `HEAD~~~` or `HEAD~3` is "Great grandparent of `HEAD`"



End of part 1

- Repositories
- Commits
- Three trees
- `add`, `commit`, `checkout`

Take a break, get some coffee

