

# Demystifying Git

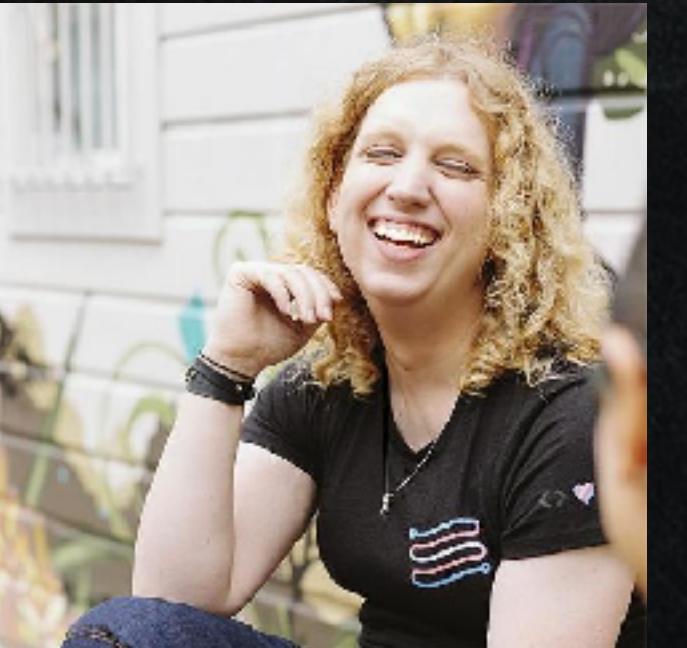
A look under the hood of  
your .git folder



Neha Batra  
[github.com/nerdneha](https://github.com/nerdneha)



Katrina Uychaco  
[github.com/kuychaco](https://github.com/kuychaco)



Michelle Tilley  
[github.com/BinaryMuse](https://github.com/BinaryMuse)

@nerdneha



## The new native

Extend your GitHub workflow beyond your browser with GitHub Desktop, completely redesigned with Electron. Get a unified cross-platform experience that's completely open source and ready to customize.

[Download for macOS](#)[Download for Windows](#)

By downloading, you agree to the [Terms and Conditions](#).

A screenshot of the GitHub Desktop application window. The top bar shows the current repository as "desktop", the branch as "esc-pr", and the status of "Fetch origin". The main area displays a pull request titled "Add event handler to dropdown component" by "iAmWillShepherd and Markus Olsson". The "History" tab is selected, showing a commit message: "Appease linter" by "IAmWillShepherd committed a day ago". Below this, the code editor shows a diff for the file "app/src/ui/t.../dropdown.tsx". The diff highlights changes in lines 145, 146, 147, 148, 149, 150, 151, and 152. Lines 145, 146, 147, and 148 are in red, while lines 149, 150, 151, and 152 are in green. The code snippet includes imports from "react", "prop-types", and "classnames", and defines a class "ToolbarDropdown" extending "React.Component".

```
@@ -145,6 +145,10 @@ export class ToolbarDropdown extends React.Component<
    this.state = { clientRect: null }
  }

  146 146    }

  147 147    148 +  private get isOpen() {
  149 +    return this.props.dropdownState === 'open'
  150 +  }

  151 +
  152 +  private dropdownToggel(state: DropdownState): OcticonSym
```

# WRITE / SPEAK / CODE



# Git Under The Hood

## ...of your .git folder

# Today you will learn

1. basic terminology - blobs, trees, commits, refs
2. .git/objects folder + blobs
3. commits, trees, and blobs + their pointers
4. how to understand any object
5. refs, branches, tags, and HEAD
6. reflog

# Git is like your project's DB

- **What** – Git's commands are like the SQL to **query** the DB:
  - **Porcelain commands** - outside-facing - what we deal with
  - **Plumbing commands** - inside-facing - lower-level commands (don't worry too much about this)
- **How** – Stores **entire files/folders + its history (not diffs)**

# Blob



**Blob: file1.txt**

**Blob: file2.txt**

# Trees

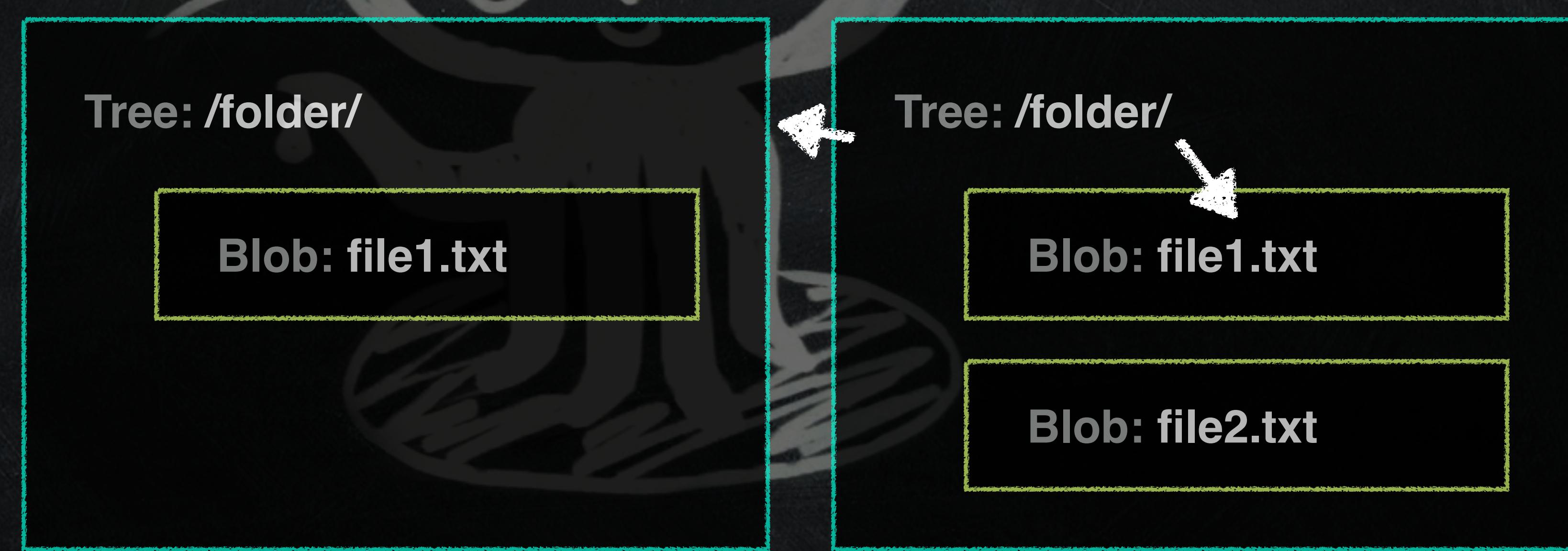


Tree: /folder/

Blob: file1.txt

Blob: file2.txt

# Trees



# Commits

Commit: `acb0ae9225e75f8083b0e`  
Neha Batra November 17, 2017  
Parent: `fa7d1d66875421372bcceb`

Add file2.txt

Tree: /folder/

Blob: file1.txt

Blob: file2.txt

# Commits

Commit: fa7d1d66875421372bcceb  
Neha Batra November 17, 2017

Initial commit

Tree: /folder/

Blob: file1.txt

Commit: acb0ae9225e75f8083b0e  
Neha Batra November 17, 2017  
Parent: fa7d1d66875421372bcceb

Add file2.txt

Tree: /folder/

Blob: file1.txt

Blob: file2.txt

# Refs

# Commit: fa7d1d66875421372bcceb Neha Batra      November 17, 2017

# Initial commit

# Tree: /folder/

## Blob: file1.txt

**Commit: acb0ae9225e75f8083b0e**  
**Neha Batra November 17, 2017**  
**Parent: fa7d1d66875421372bccceb**

# Add file2.txt

# Tree: /folder/

# Blob: file1.txt

# Blob: file2.txt

@nerdneha

Ref: HEAD

Ref: master

Commit: fa7d1d66875421372bcceb  
Neha Batra November 17, 2017

Initial commit

Tree: /folder/

Blob: file1.txt

@nerdneha

Commit: acb0ae9225e75f8083b0e  
Neha Batra November 17, 2017  
Parent: fa7d1d66875421372bcceb

Add file2.txt

Tree: /folder/

Blob: file1.txt

Blob: file2.txt



# Let's look at: .git folder

```
$ echo "Hello world" > hello.txt
$ git add hello.txt
$ cat .git/objects/80/2992c4220de19a90767f3000a79a31b98d0df7
xK??R04b?H??W(?)?I?B?i$
$ alias deflate="perl -MCompress::Zlib -e 'undef $/; print unpack(uncompress(<>))'"
$ deflate .git/objects/80/2992c4220de19a90767f3000a79a31b98d0df7
blob 12Hello world
$ printf "blob 12\000Hello world\n" | shasum
802992c4220de19a90767f3000a79a31b98d0df7 -
```

### git plumbing commands:

- git cat-file -t —> show type
- git cat-file -p —> show content

```
$ echo "Hello world" > hello.txt
$ git add hello.txt
$ cat .git/objects/80/2992c4220de19a90767f3000a79a31b98d0df7
xK??R04b?H??W(?)?I?B?i$"
$ alias deflate="perl -MCompress::Zlib -e 'undef $/; print uncompress(<>)'"
$ deflate .git/objects/80/2992c4220de19a90767f3000a79a31b98d0df7
blob 12Hello world
$ printf "blob 12\000Hello world\n" | shasum
802992c4220de19a90767f3000a79a31b98d0df7 -
```

### git plumbing commands:

git cat-file -t → show type  
git cat-file -p → show content

```
$ git cat-file -t 802992c4220de19a90767f3000a79a31b98d0df7
blob
$ git cat-file -p 802992c4220de19a90767f3000a79a31b98d0df7
Hello world
```

# How: Blob → .git files?

- Git creates objects for everything it needs to track
- when we `git add` ed the file to the index
  - git created a blob
- When we make our commit
  - git will add a **commit object** with a **tree object** with that blob

A black and white photograph of a hand holding a piece of paper. The paper features a simple line drawing of a car with two large, prominent headlights. The background is dark and textured.

# Let's make a commit

@nerdneha

```
$ git cat-file -t 802992c4220de19a90767f3000a79a31b98d0df7  
blob  
$ git cat-file -p 802992c4220de19a90767f3000a79a31b98d0df7  
Hello world  
$ git ls-files -s  
100644 802992c4220de19a90767f3000a79a31b98d0df7 0          hello.txt  
$ git commit -m "First commit"  
[master (root-commit) ef866f3] First commit      commit  
 1 file changed, 1 insertion(+)  
  create mode 100644 hello.txt  
$ git cat-file -t ef866f30a61b24b38a650c22223bf4a62fb9a6fb  
commit  
$ git cat-file -t a50b30eb6b223aef893c367a0b93e9a5b21f155f  
tree  
$ git cat-file -p ef866f30a61b24b38a650c22223bf4a62fb9a6fb  
tree a50b30eb6b223aef893c367a0b93e9a5b21f155f  
author Neha Batra <nerdneha@github.com> 1520390359 +0100  
committer Neha Batra <nerdneha@github.com> 1520390359 +0100
```

First commit

```
@nerdneha$ git cat-file -p a50b30eb6b223aef893c367a0b93e9a5b21f155f  
100644 blob 802992c4220de19a90767f3000a79a31b98d0df7      hello.txt
```

```
$ git cat-file -t 802992c4220de19a90767f3000a79a31b98d0df7  
blob  
$ git cat-file -p 802992c4220de19a90767f3000a79a31b98d0df7  
Hello world  
$ git ls-files -s  
100644 802992c4220de19a90767f3000a79a31b98d0df7 0 hello.txt  
$ git commit -m "First commit"  
[master (root-commit) ef866f3] First commit commit -> tree  
1 file changed, 1 insertion(+)  
create mode 100644 hello.txt  
$ git cat-file -t ef866f30a61b24b38a650c22223bf4a62fb9a6fb  
commit  
$ git cat-file -t a50b30eb6b223aef893c367a0b93e9a5b21f155f  
tree  
$ git cat-file -p ef866f30a61b24b38a650c22223bf4a62fb9a6fb  
tree a50b30eb6b223aef893c367a0b93e9a5b21f155f  
author Neha Batra <nerdneha@github.com> 1520390359 +0100  
committer Neha Batra <nerdneha@github.com> 1520390359 +0100
```

First commit

```
@nerdneha$ git cat-file -p a50b30eb6b223aef893c367a0b93e9a5b21f155f  
100644 blob 802992c4220de19a90767f3000a79a31b98d0df7 hello.txt
```

```
$ git cat-file -t 802992c4220de19a90767f3000a79a31b98d0df7  
blob  
$ git cat-file -p 802992c4220de19a90767f3000a79a31b98d0df7  
Hello world  
$ git ls-files -s  
100644 802992c4220de19a90767f3000a79a31b98d0df7 0 hello.txt  
$ git commit -m "First commit"  
[master (root-commit) ef866f3] First commit tree -> blob  
1 file changed, 1 insertion(+)  
create mode 100644 hello.txt  
$ git cat-file -t ef866f30a61b24b38a650c22223bf4a62fb9a6fb  
commit  
$ git cat-file -t a50b30eb6b223aef893c367a0b93e9a5b21f155f  
tree  
$ git cat-file -p ef866f30a61b24b38a650c22223bf4a62fb9a6fb  
tree a50b30eb6b223aef893c367a0b93e9a5b21f155f  
author Neha Batra <nerdneha@github.com> 1520390359 +0100  
committer Neha Batra <nerdneha@github.com> 1520390359 +0100
```

First commit

```
@nerdneha$ git cat-file -p a50b30eb6b223aef893c367a0b93e9a5b21f155f  
100644 blob 802992c4220de19a90767f3000a79a31b98d0df7 hello.txt
```

```
$ git cat-file -t 802992c4220de19a90767f3000a79a31b98d0df7  
blob  
$ git cat-file -p 802992c4220de19a90767f3000a79a31b98d0df7  
Hello world  
$ git ls-files -s  
100644 802992c4220de19a90767f3000a79a31b98d0df7 0 hello.txt  
$ git commit -m "First commit"  
[master (root-commit) ef866f3] First commit blob  
1 file changed, 1 insertion(+)  
create mode 100644 hello.txt  
$ git cat-file -t ef866f30a61b24b38a650c22223bf4a62fb9a6fb  
commit  
$ git cat-file -t a50b30eb6b223aef893c367a0b93e9a5b21f155f  
tree  
$ git cat-file -p ef866f30a61b24b38a650c22223bf4a62fb9a6fb  
tree a50b30eb6b223aef893c367a0b93e9a5b21f155f  
author Neha Batra <nerdneha@github.com> 1520390359 +0100  
committer Neha Batra <nerdneha@github.com> 1520390359 +0100
```

First commit

```
@nerdneha$ git cat-file -p a50b30eb6b223aef893c367a0b93e9a5b21f155f  
100644 blob 802992c4220de19a90767f3000a79a31b98d0df7 hello.txt
```

# Checkpoint

1. basic terminology - blobs, trees, commits, refs ✓
2. .git/objects folder + blobs ✓
3. commits, trees, and blobs + their pointers ✓
4. how to understand any object
5. refs, branches, tags, and HEAD
6. reflog

# Playing with commits

- Changing a committed file's content
- Making a second commit
- Committing a second file
- Committing a subfolder



# Let's make more commits!

@nerdneha

# Checkpoint

1. basic terminology - blobs, trees, commits, refs ✓
2. .git/objects folder + blobs ✓
3. commits, trees, and blobs + their pointers ✓
4. how to understand any object ✓
5. refs, branches, tags, and HEAD
6. reflog

# Refs, Branches, Tags, and HEAD

- **refs/heads** folder holds branches (ex: master)
  - master is a branch that git creates by default
- **refs/tags** folder holds tags = labels
  - lightweight tags point to commits just like branches do
  - annotated tags are tags+objects that have additional info



# Let's look at refs!

Ref: HEAD



@nerdneha

Ref: HEAD

→  
Ref: master



Ref: HEAD

Ref: master

Commit: fa7d1d66875421372bcceb  
Neha Batra November 17, 2017

Initial commit

Tree: /folder/

Blob: file1.txt

@nerdneha

Commit: acb0ae9225e75f8083b0e  
Neha Batra November 17, 2017  
Parent: fa7d1d66875421372bcceb

Add file2.txt

Ref: HEAD

Ref: master

Commit: fa7d1d66875421372bcceb  
Neha Batra November 17, 2017

Initial commit

Tree: /folder/

Blob: file1.txt

Commit: acb0ae9225e75f8083b0e  
Neha Batra November 17, 2017  
Parent: fa7d1d66875421372bcceb

Add file2.txt

Tree: /folder/

Ref: HEAD

Ref: master

Commit: fa7d1d66875421372bcceb  
Neha Batra November 17, 2017

Initial commit

Tree: /folder/

Blob: file1.txt

@nerdneha

Commit: acb0ae9225e75f8083b0e  
Neha Batra November 17, 2017  
Parent: fa7d1d66875421372bcceb

Add file2.txt

Tree: /folder/

Blob: file1.txt

Blob: file2.txt

# Checkpoint

1. basic terminology - blobs, trees, commits, refs ✓
2. .git/objects folder + blobs ✓
3. commits, trees, and blobs + their pointers ✓
4. how to understand any object ✓
5. refs, branches, tags, and HEAD ✓
6. reflog ✓

# CONGRATS



You are a git master

# YOU ARE FREE...TO EAT LUNCH



**There's food spread throughout the venue.  
If you see a line forming just continue back towards the front entrance  
for additional stations.**

@nerdneha