

Git objects

Reuven M. Lerner, PhD

reuven@lerner.co.il • <http://lerner.co.il/>

Git objects

- Git stores four kinds of objects:
 - Blobs
 - Trees
 - Commits
 - Tags

Blobs

- Files are stored as “blobs” (binary large objects)
- Git stores each version of a file, not diffs
- Each file is uniquely identified via SHA-1
- Each version is stored only once
- So if a file doesn't change over multiple commits, all commits point to the same version

Blob metadata

- Git only cares about the file contents!
- The blob doesn't contain the filename
- This information is stored in the tree, i.e., the directory in which the blob is located

Trees

- Each tree represents one level of directory information
- Contains blob identifiers, pathnames, and some metadata
- May contain other trees, for directory hierarchies

Commits

- Each commit is its own object, with its own SHA-1
- Contains metadata (author, date, log)
- Points to a tree that captures the state of the repository
- Most commits have one parent
 - Except the root commit and merges

Tags

- A fourth Git object, the “tag,” lets you give a human-readable name to an object
- This is an “annotated tag,” meaning one that has a bit more information and heft than a regular tag (which is basically just a branch)

Object storage

- Objects are stored in `.git/objects`
- That directory contains all of the objects, divided into subdirectories by SHA1 name

Viewing objects

- To view an object's contents, use “git cat-file” with the -p option

```
$ git cat-file -p 3ad18  
print "Hello, world"
```

```
$ git cat-file -p 3ad19  
fatal: Not a valid object name 3ad19
```

Getting object types

- To view an object's type, use “git cat-file” with the -t option

```
$ git cat-file -t 3ad18  
blob
```

```
$ git cat-file -t 22d0  
commit
```

Mostly, think about commits

- Git has been described as a one-trick pony
- It knows how to create commits!
- (And it knows how to put names on some of those commits!)
- Everything in Git is about creating, moving, and tagging commits. Most commits have one parent. That's what you need to know about Git theory.