

Git

The Stupid Content Tracker

Different way of looking
at version control

How does `git` store data?

Storage on disk

Git objects

blob

tree

commit

tag

Git objects

Immutable!

Why is this awesome?

References (refs)

heads
remotes
tags

Directed Acyclic Graph

vs

Delta

How does this all work
in practice?

The first commit

The second commit

The third commit

A branch!

A commit to the branch

A commit to the master

Merging the two

Dealing with other
people (remotes)

How to refer to stuff
(the treeish)

Treeish

Full SHA-1

3d3d4eb2e5daf3b9fccb3c175d0d5db9acc887b0

Treeish

Partial SHA-1

3d3d4eb2e5daf3b9fccb3c175d0d5db9acc887b0

3d3d4eb2e5daf3b9f

3d3d4

Treeish

Branch, Remote, or Tag

v1.0

master

origin/master

Treeish

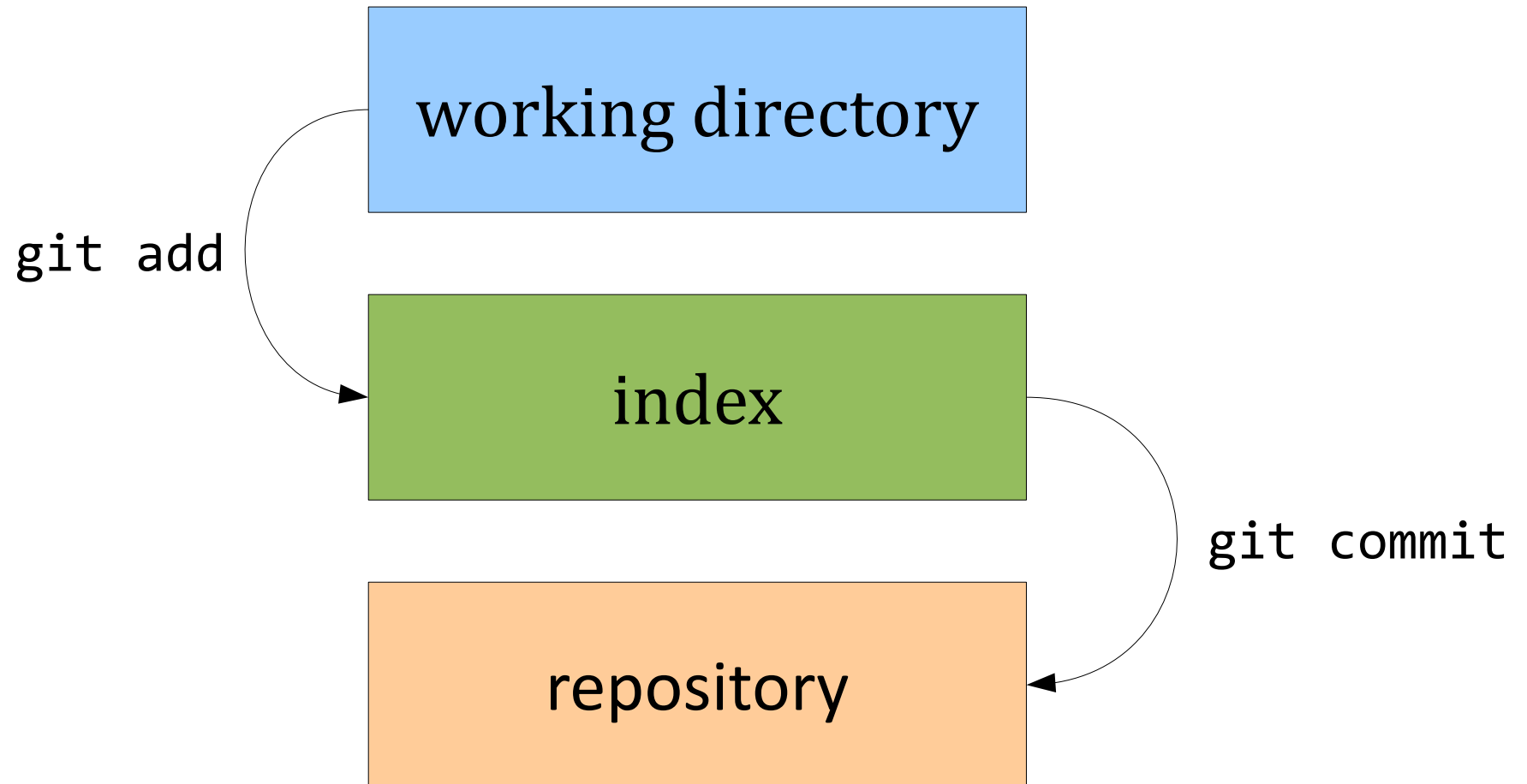
Caret Parent

master[^]

Treeish

Others (also has ranges)

How commits work



Ok, I know how it works.
How do I use it?

git commands

add	diff-files	lost-found	parse-remote	shell
add--interactive	diff-index	ls-files	patch-id	shortlog
am	diff-tree	ls-remote	peek-remote	show
annotate	fast-export	ls-tree	prune	show-branch
apply	fast-import	mailinfo	prune-packed	show-index
archive	fetch	-mailsplit	pull	show-ref
bisect	fetch--tool	merge	push	stash
blame	fetch-pack	merge-base	quiltimport	status
branch	filter-branch	merge-file	read-tree	strip-space
bundle	fmt-merge-msg	merge-index	rebase	submodule
cat-file	for-each-ref	merge-octopus	rebase--interactive	symbolic-ref
check-attr	format-patch	merge-one-file	receive-pack	tag
check-ref-format	fsck	merge-ours	reflog	tar-tree
checkout	fsck-objects	merge-recursive	relink	unpack-file
checkout-index	gc	merge-resolve	remote	unpack-objects
cherry	get-tar-commit-id	merge-stupid	repack	update-index
cherry-pick	grep	merge-subtree	repo-config	update-ref
clean	hash-object	merge-tree	request-pull	update-server-info
clone	http-fetch	mergetool	rerere	upload-archive
commit	http-push	mktag	reset	upload-pack
commit-tree	imap-send	mtree	rev-list	var
config	index-pack	mv	rev-parse	verify-pack
count-objects	init	name-rev	revert	verify-tag
daemon	init-db	pack-objects	rm	web--browse
describe	instaweb	pack-redundant	send-pack	whatchanged
diff	log	pack-refs	sh-setup	write-tree

There will be a quiz

Actually, you don't need
to know them all

Setting up git

```
git config -global username  
    "Hans Nielsen"
```

```
git config -global user.email  
    "hnielsen@umail.ucsb.edu"
```

Getting a repository

```
git clone  
ssh://hans@example.com/blah.git
```

Creating a repository

```
git init  
git add .  
git commit
```

Basic toolset

```
git add  
git status  
git commit  
git log  
git diff
```


Branching / Merging

```
git branch  
git checkout  
git merge
```

Working with others

```
git remote  
git fetch  
git merge  
git pull  
git push
```