

[Prev \(4 stashing.html\)](#) [Next \(4 tracking branches.html\)](#)

[Git Community Book \(index.html\)](#)

Git Treeishes

There are a number of ways to refer to a particular commit or tree other than spelling out the entire 40-character sha. In Git, these are referred to as a 'treeish'.

Partial Sha

If your commit sha is '980e3ccdaac54a0d4de358f3fe5d718027d96aae', git will recognize any of the following identically:

```
980e3ccdaac54a0d4de358f3fe5d718027d96aae
980e3ccdaac54a0d4
980e3cc
```

As long as the partial sha is unique - it can't be confused with another (which is incredibly unlikely if you use at least 5 characters), git will expand a partial sha for you.

Branch, Remote or Tag Name

You can always use a branch, remote or tag name instead of a sha, since they are simply pointers anyhow. If your master branch is on the 980e3 commit and you've pushed it to origin and have tagged it 'v1.0', then all of the following are equivalent:

```
980e3ccdaac54a0d4de358f3fe5d718027d96aae
origin/master
refs/remotes/origin/master
master
refs/heads/master
v1.0
refs/tags/v1.0
```

Which means the following will give you identical output:

```
$ git log master

$ git log refs/tags/v1.0
```

Date Spec

The Ref Log that git keeps will allow you to do some relative stuff locally, such as:

```
master@{yesterday}

master@{1 month ago}
```

Which is shorthand for 'where the master branch head was yesterday', etc. Note that this format can result in different shas on different computers, even if the master branch is currently pointing to the same place.

Ordinal Spec

This format will give you the Nth previous value of a particular reference. For example:

```
master@{5}
```

will give you the 5th prior value of the master head ref.

Carrot Parent

This will give you the Nth parent of a particular commit. This format is only useful on merge commits - commit objects that have more than one direct parent.

```
master^2
```

Tilde Spec

The tilde spec will give you the Nth grandparent of a commit object. For example,

```
master~2
```

will give us the first parent of the first parent of the commit that master points to. It is equivalent to:

```
master^^
```

You can keep doing this, too. The following specs will point to the same commit:

```
master^^^^^^
```

```
master~3^^2
```

```
master~6
```

Tree Pointer

This disambiguates a commit from the tree that it points to. If you want the sha that a commit points to, you can add the `'{tree}'` spec to the end of it.

```
master^{tree}
```

Blob Spec

If you want the sha of a particular blob, you can add the blob path at the end of the treeish, like so:

```
master:/path/to/file
```

Range

Finally, you can specify a range of commits with the range spec. This will give you all the commits between 7b593b5 and 51bea1 (where 51bea1 is most recent), excluding 7b593b5 but including 51bea1:

```
7b593b5..51bea1
```

This will include every commit *since* 7b593b:

```
7b593b..
```

This book is maintained by Scott Chacon, and hosting is donated by GitHub.

Please email me at [schacon@gmail.com \(mailto:schacon@gmail.com\)](mailto:schacon@gmail.com) with patches, suggestions and comments.

<http://github.com>

