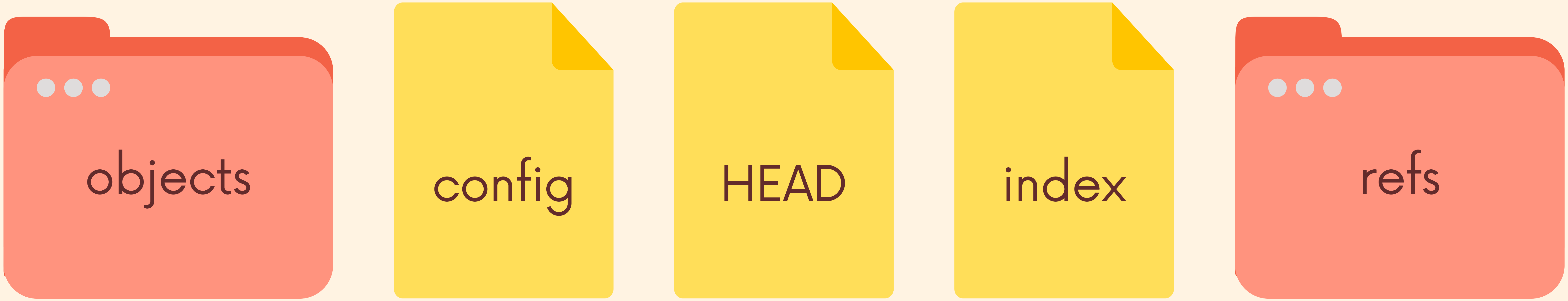
Git Behind The Scenes

What is in **.git**??



objects

config

HEAD

index

refs

There's more, but this is the juicy stuff

Config

The config file is for...configuration. We've seen how to configure global settings like our name and email across all Git repos, but we can also configure things on a per-repo basis.

Refs Folder

Inside of refs, you'll find a heads directory. **refs/heads** contains one file per branch in a repository. Each file is named after a branch and contains the hash of the commit at the tip of the branch.

For example **refs/heads/master** contains the commit hash of the last commit on the master branch.

Refs also contains a **refs/tags** folder which contains one file for each tag in the repo.

HEAD

HEAD is just a text file that keeps track of where HEAD points.

If it contains refs/heads/master, this means that HEAD is pointing to the master branch.

In detached HEAD, the HEAD file contains a commit hash instead of a branch reference

Index

The index file is a binary file that contains a list of the files the repository is tracking. It stores the file names as well as some metadata for each file.

Not that the index does NOT store the actual contents of files. It only contains references to files.

Objects Folder

The objects directory contains all the repo files. This is where Git stores the backups of files, the commits in a repo, and more.

The files are all compressed and encrypted, so they won't look like much!

4 Types of Git Objects

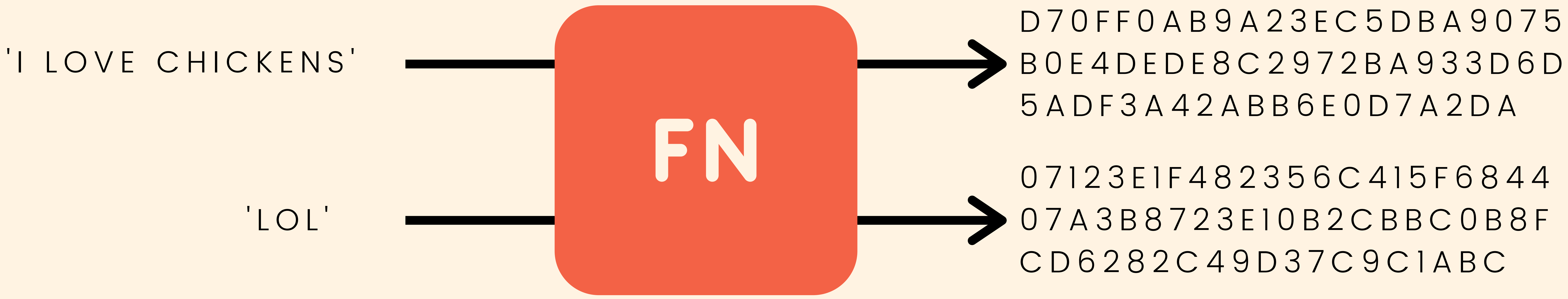
- commit
- tree
- blob
- annotated tag

TIME OUT!

We need to talk about hashing

HASHING FUNCTIONS

Hashing functions are functions that map input data of some arbitrary size to fixed-size output values.



'I LOVE CHICKENS'

'LOL'

FN

D70FF0AB9A23EC5DBA9075
B0E4DEDE8C2972BA933D6D
5ADF3A42ABB6E0D7A2DA

07123E1F482356C415F6844
07A3B8723E10B2CBBC0B8F
CD6282C49D37C9C1ABC

CRYPTOGRAPHIC HASH FUNCTIONS

1. One-way function which is infeasible to invert
2. Small change in input yields large change in the output
3. Deterministic - same input yields same output
4. Unlikely to find 2 outputs with same value

SHA-1

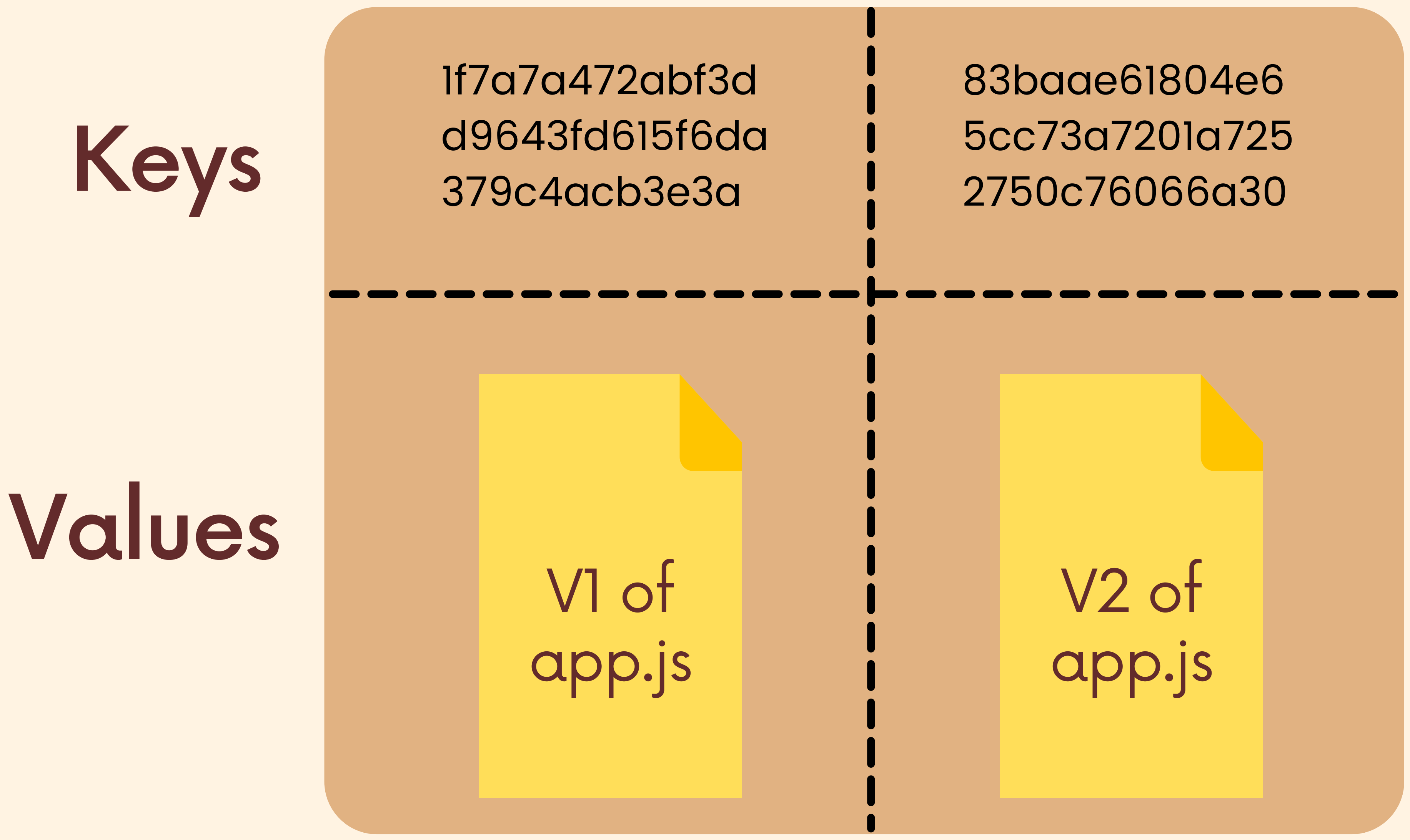
Git uses a hashing function called SHA-1 (though this is set to change eventually).

- SHA-1 always generates 40-digit hexadecimal numbers
- The commit hashes we've seen a million times are the output of SHA-1

Git Database

Git is a **key-value data store.** We can insert any kind of content into a Git repository, and Git will hand us back a unique key we can later use to retrieve that content.

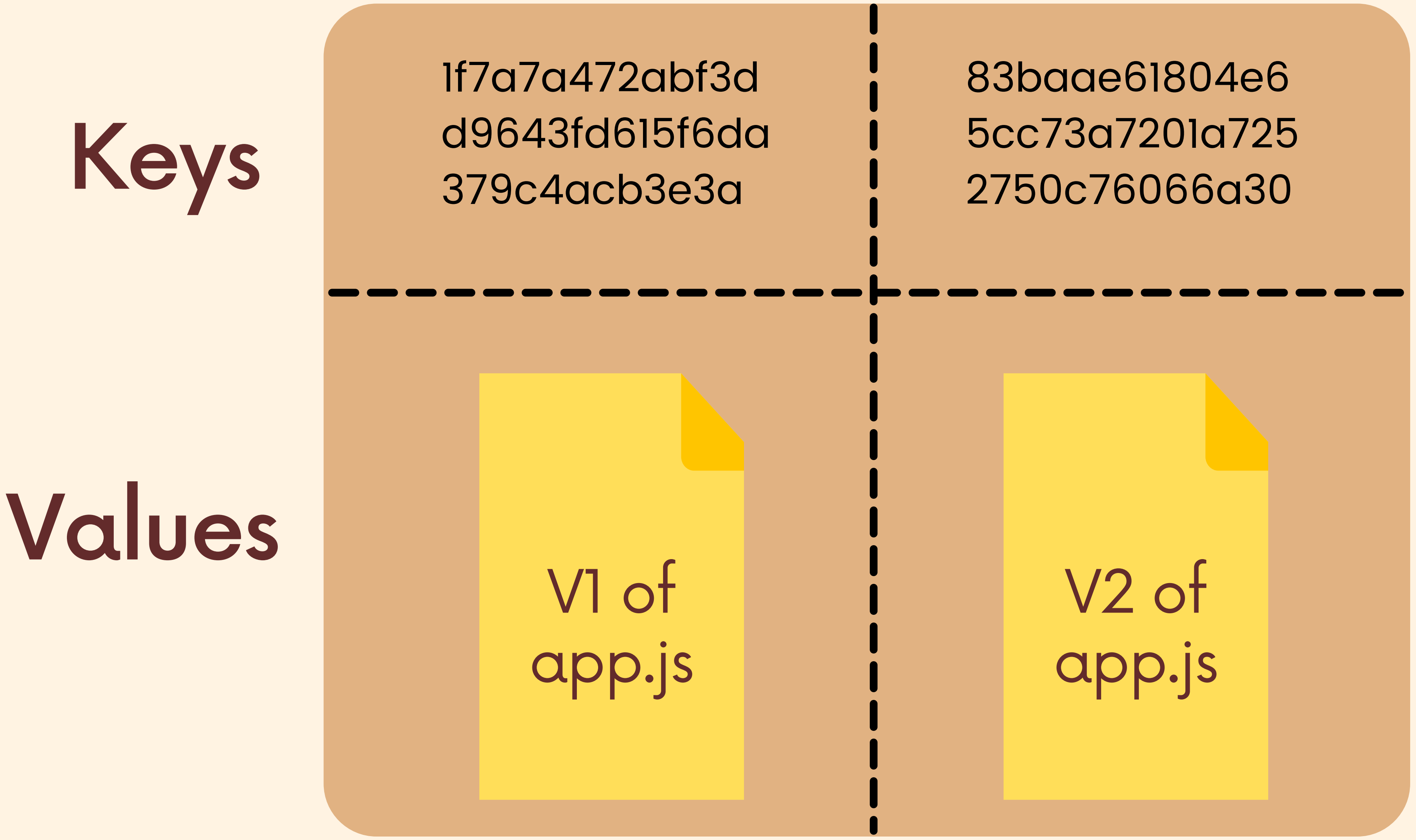
These keys that we get back are SHA-1 checksums.

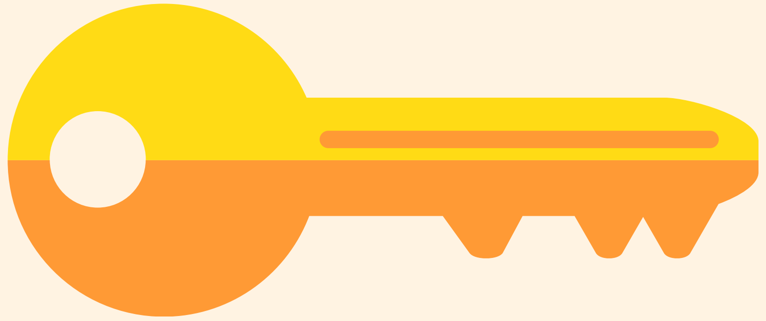


Keys	1f7a7a472abf3d d9643fd615f6da 379c4acb3e3a	83baae61804e6 5cc73a7201a725 2750c76066a30
Values	V1 of app.js	V2 of app.js

Please give me the content for this key:

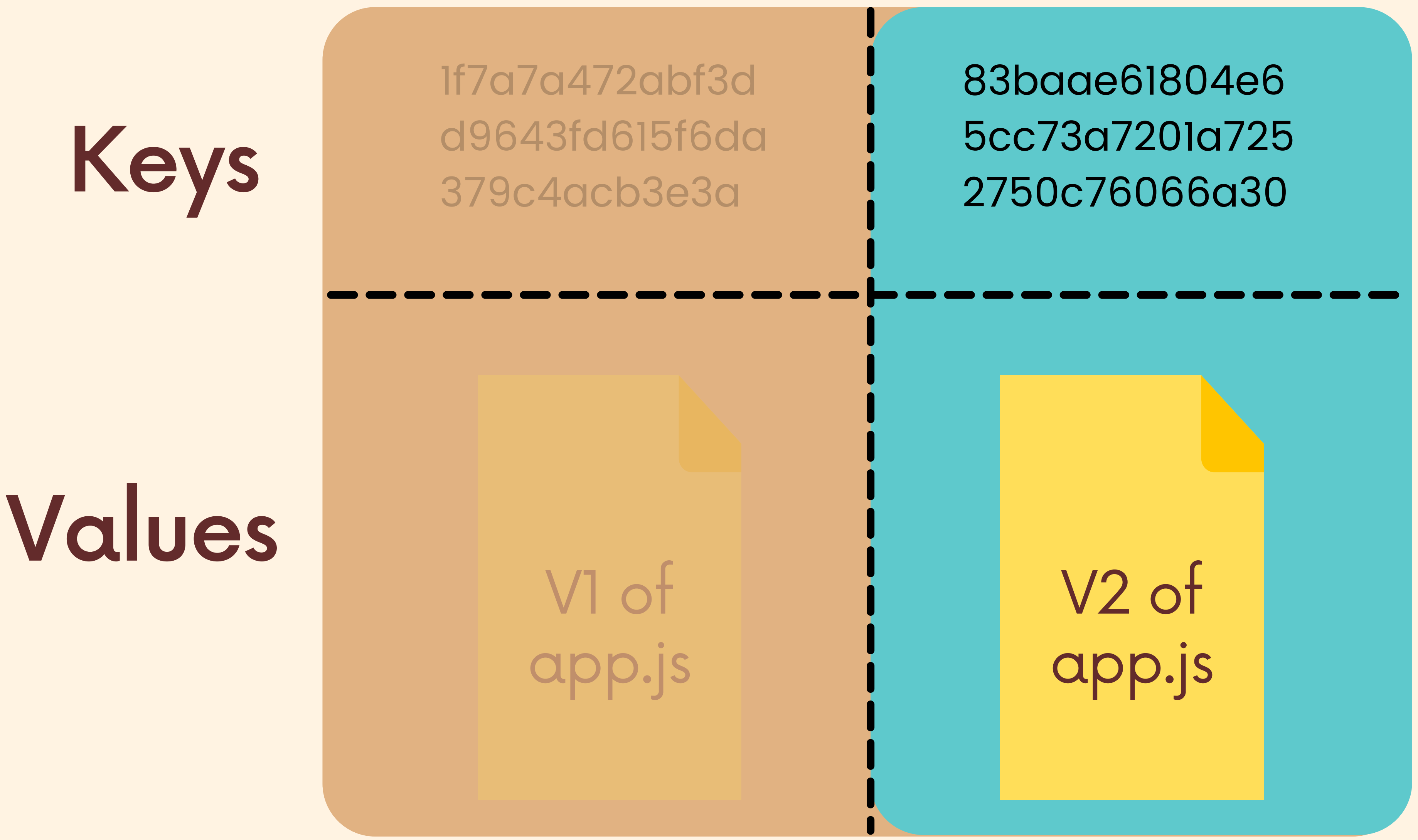
83BAAE61804E65CC73A720
1A7252750C76066A30





Please give me the content for this key:

83BAAE61804E65CC73A720
1A7252750C76066A30



Keys

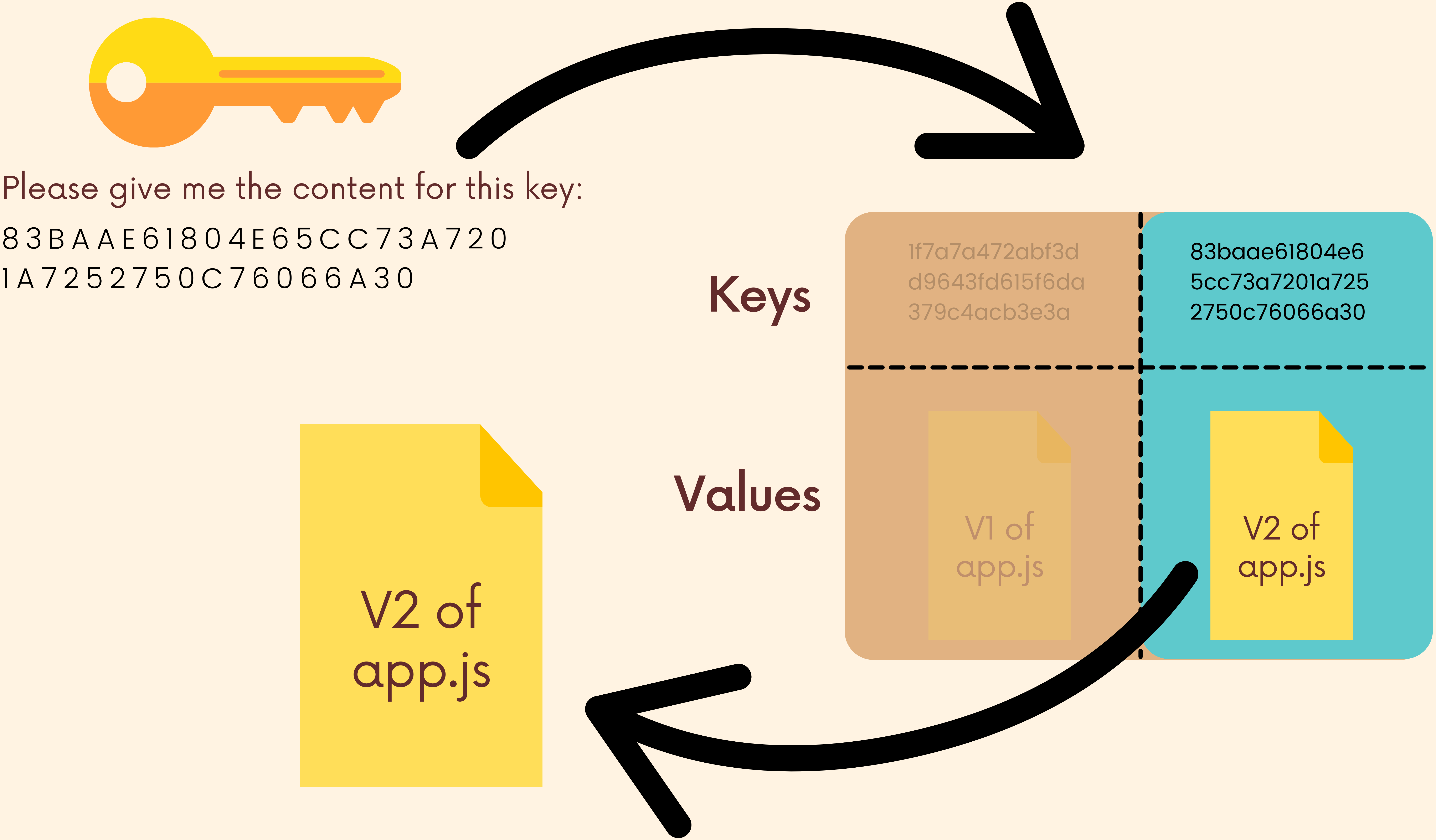
Values

1f7a7a472abf3d
d9643fd615f6da
379c4acb3e3a

83baae61804e6
5cc73a7201a725
2750c76066a30

V1 of app.js

V2 of app.js



Please give me the content for this key:

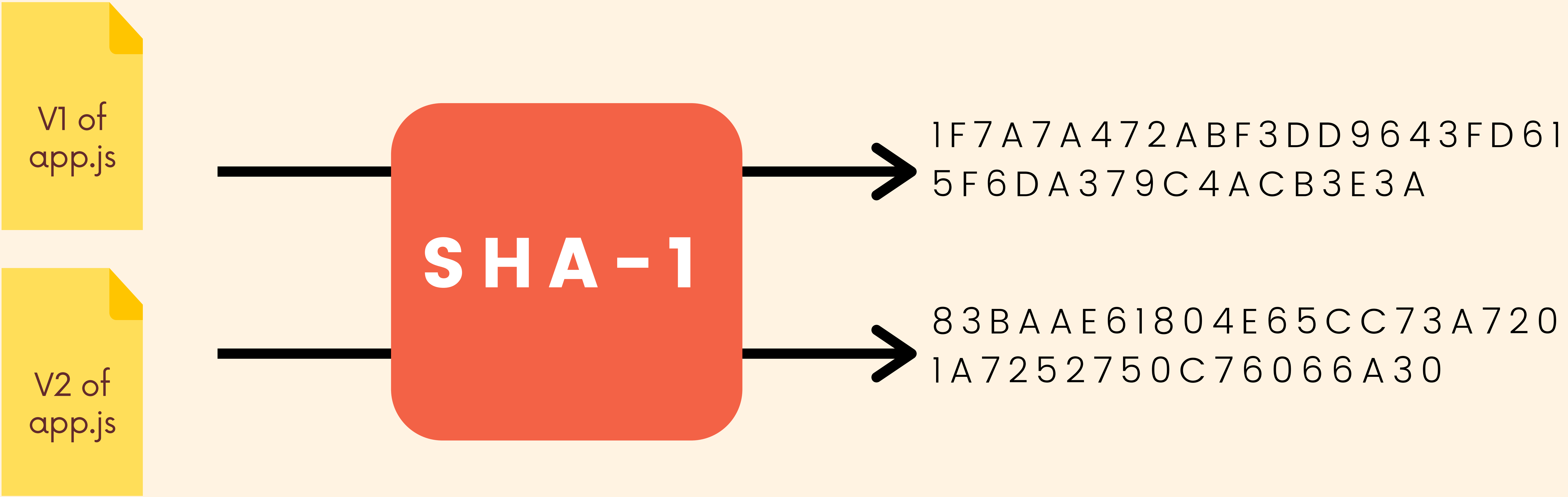
83BAAE61804E65CC73A720
1A7252750C76066A30

Keys	1f7a7a472abf3d d9643fd615f6da 379c4acb3e3a	83baae61804e6 5cc73a7201a725 2750c76066a30
Values	V1 of app.js	V2 of app.js

V2 of app.js

HASHING FUNCTIONS

Git uses SHA-1 to hash our files, directories, and commits.



V1 of app.js

V2 of app.js

SHA-1

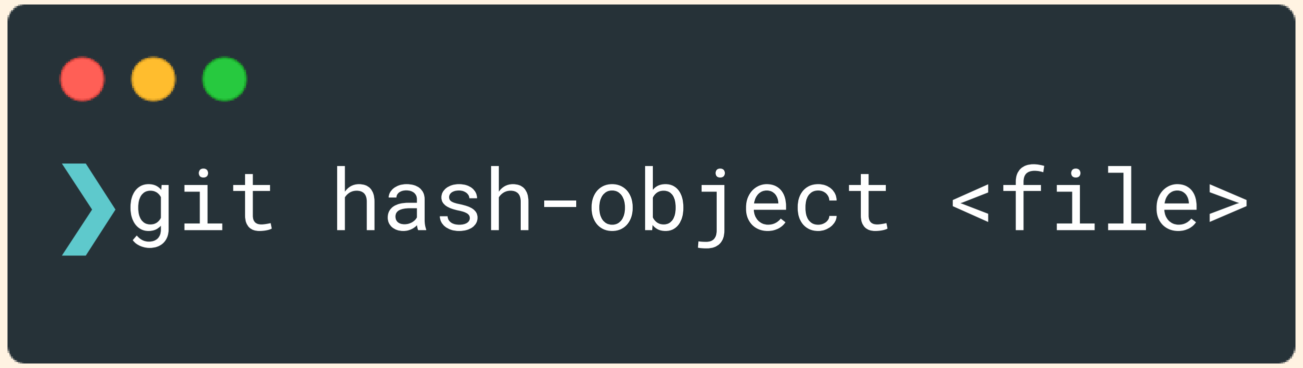
1F7A7A472ABF3DD9643FD61
5F6DA379C4ACB3E3A

83BAAE61804E65CC73A720
1A7252750C76066A30

Let's Try Hashing

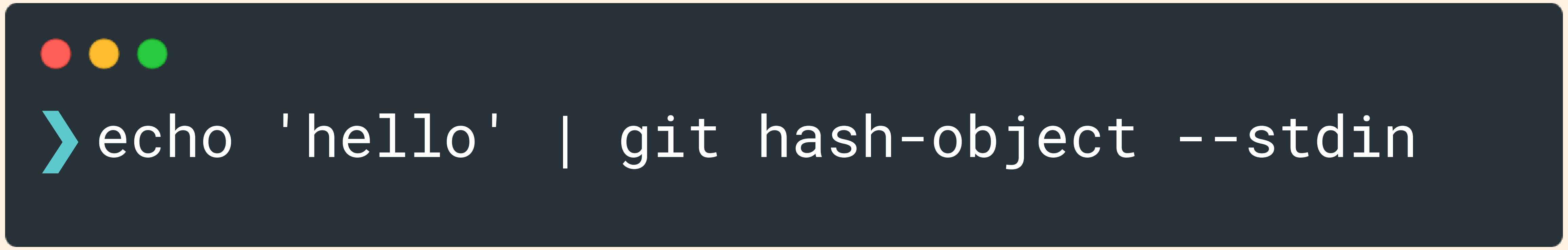
The **git hash-object** command takes some data, stores in in our .git/objects directory and gives us back the unique SHA-1 hash that refers to that data object.

In the simplest form (shown on the right), Git simply takes some content and returns the unique key that WOULD be used to store our object. But it does not actually store anything



```
❯git hash-object <file>
```

Let's Try Hashing

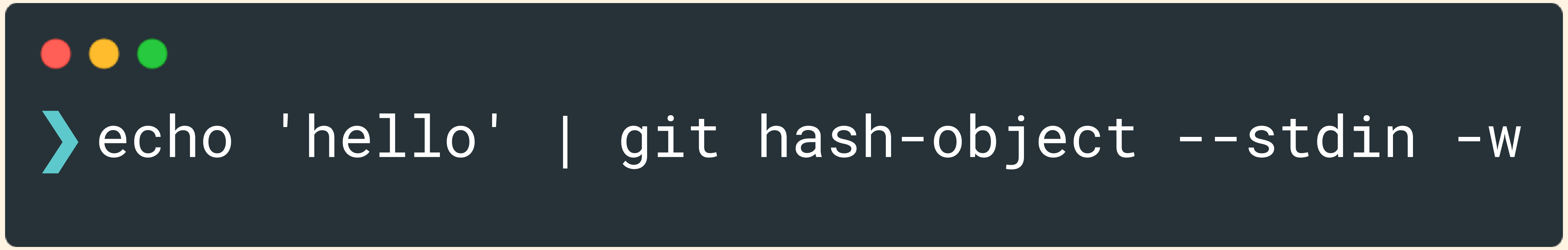


```
❯ echo 'hello' | git hash-object --stdin
```

The **--stdin** option tells git hash-object to use the content from stdin rather than a file. In our example, it will hash the word 'hello'

The echo command simply repeats whatever we tell it to repeat to the terminal. We pipe the output of echo to **git hash-object.**

Let's Try Hashing

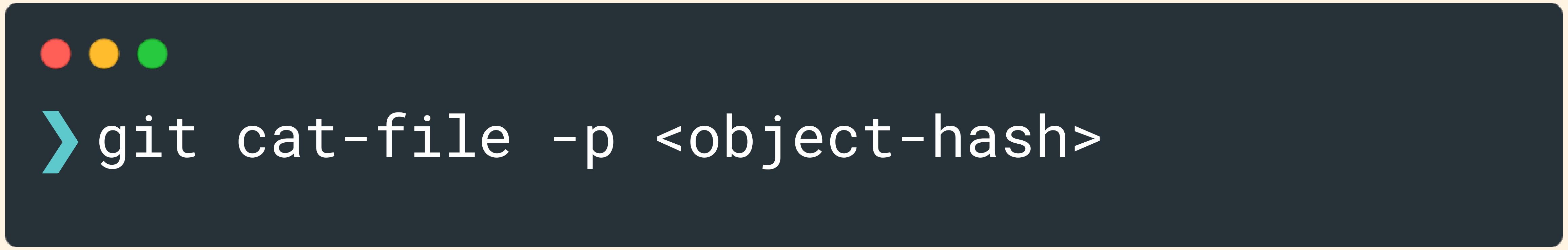


```
❯ echo 'hello' | git hash-object --stdin -w
```

Rather than simply outputting the key that git would store our object under, we can use the **-w** option to tell git to actually write the object to the database.

After running this command, check out the contents of **.git/objects**

Let's Try Hashing

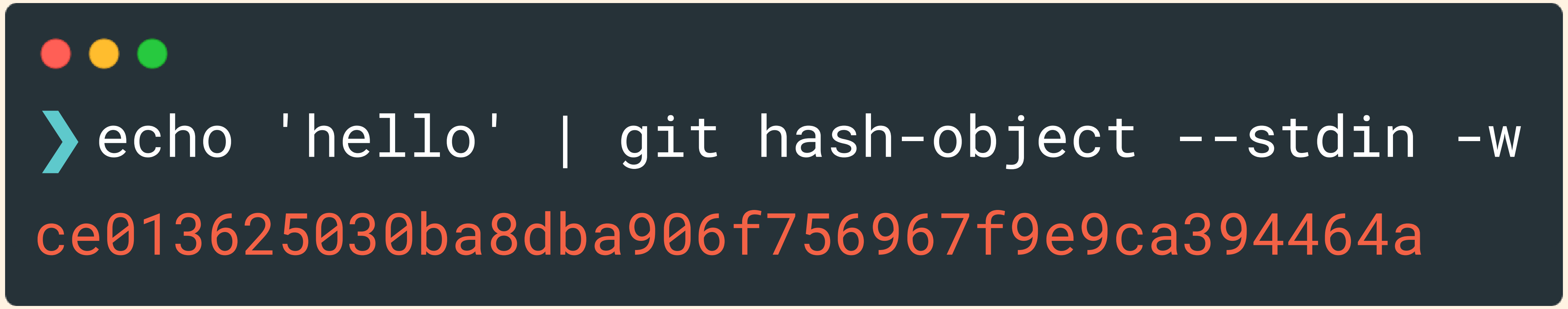


```
❯ git cat-file -p <object-hash>
```

Now that we have data stored in our Git object database, we can try retrieving it using the git cat-file command.

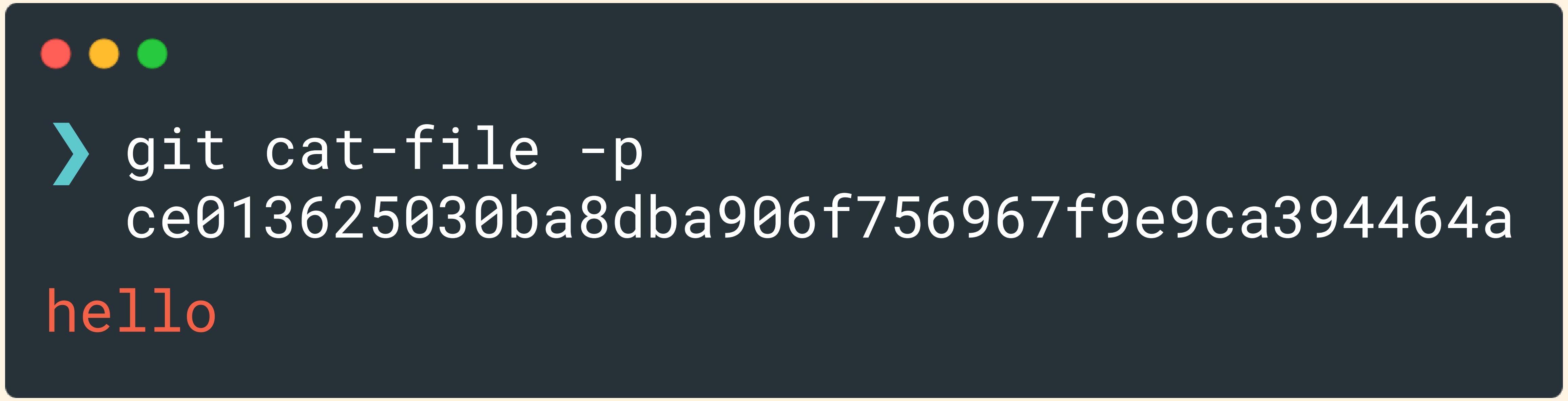
The -p option tells Git to pretty print the contents of the object based on its type.

1.Tell Git to store "hello". Note the hash that we get back.



```
❯ echo 'hello' | git hash-object --stdin -w
ce013625030ba8dba906f756967f9e9ca394464a
```

2. Pass the hash from above to git cat-file. Git retrieves the corresponding data "hello" that it had stored under that key



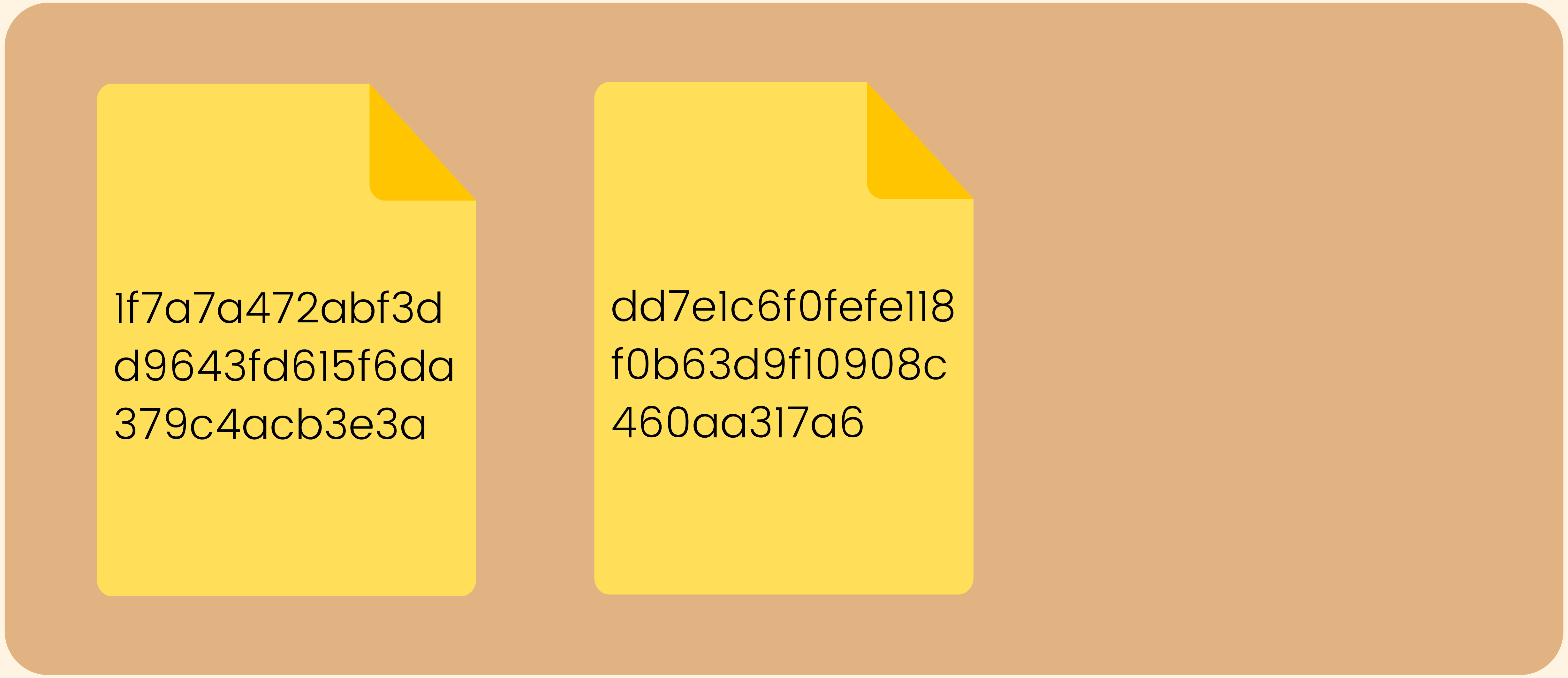
```
❯ git cat-file -p
  ce013625030ba8dba906f756967f9e9ca394464a
hello
```

'hello' → 1f7a7a472abf3dd9643fd615f6da379c4acb3e3a

1f7a7a472abf3dd9643fd615f6da379c4acb3e3a

'hello' → 1f7a7a472abf3dd9643fd615f6da379c4acb3e3a

'goodbye' → dd7e1c6f0fefe118f0b63d9f10908c460aa317a6

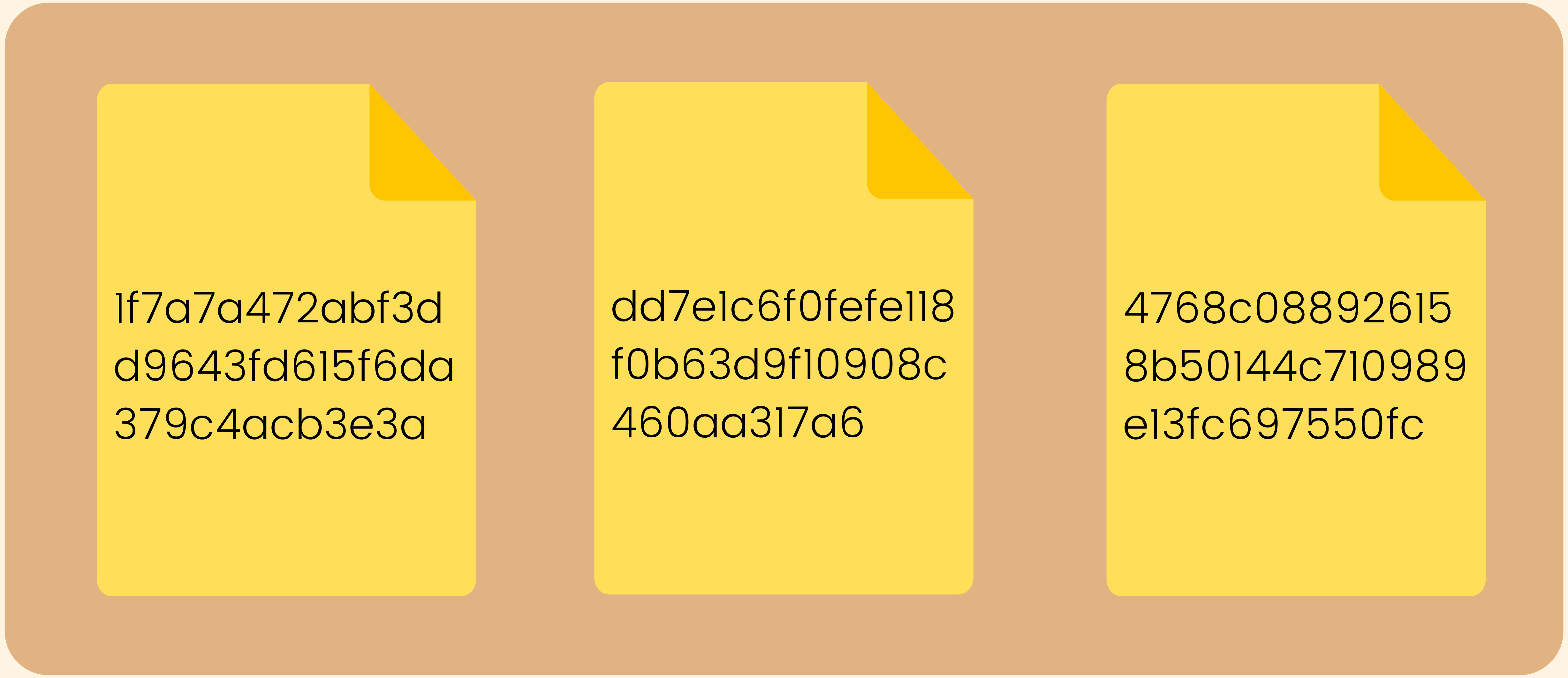


1f7a7a472abf3dd9643fd615f6da379c4acb3e3a
dd7e1c6f0fefe118f0b63d9f10908c460aa317a6

'hello' → 1f7a7a472abf3dd9643fd615f6da379c4acb3e3a

'goodbye' → dd7e1c6f0fefe118f0b63d9f10908c460aa317a6

'chicken' → 4768c088926158b50144c710989e13fc697550fc



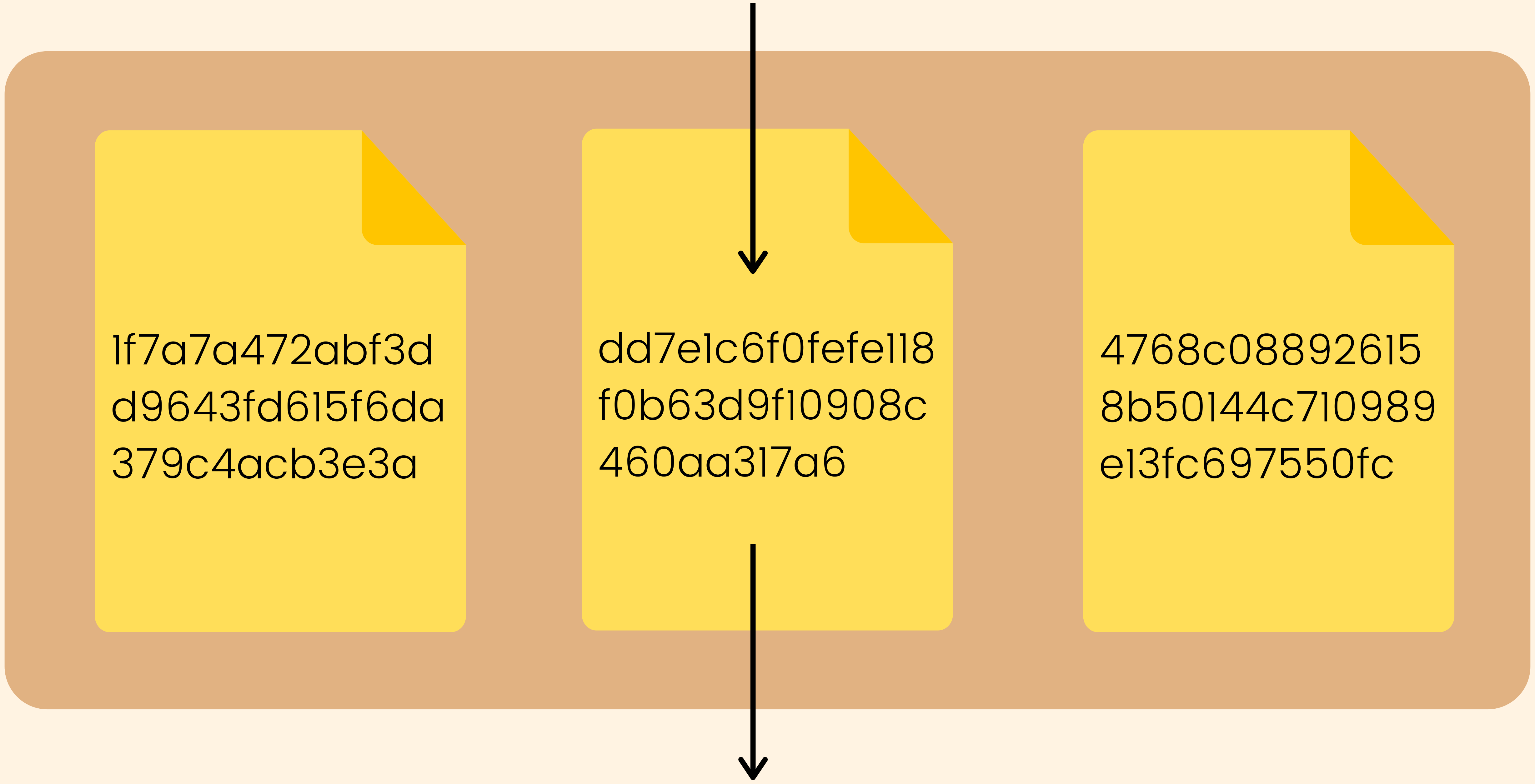
1f7a7a472abf3d
d9643fd615f6da
379c4acb3e3a

dd7e1c6f0fefe118
f0b63d9f10908c
460aa317a6

4768c08892615
8b50144c710989
e13fc697550fc

Hello Git, I would like to know what you have stored under the key:

dd7e1c6f0fefe118f0b63
d9f10908c460aa317a6



1f7a7a472abf3d
d9643fd615f6da
379c4acb3e3a

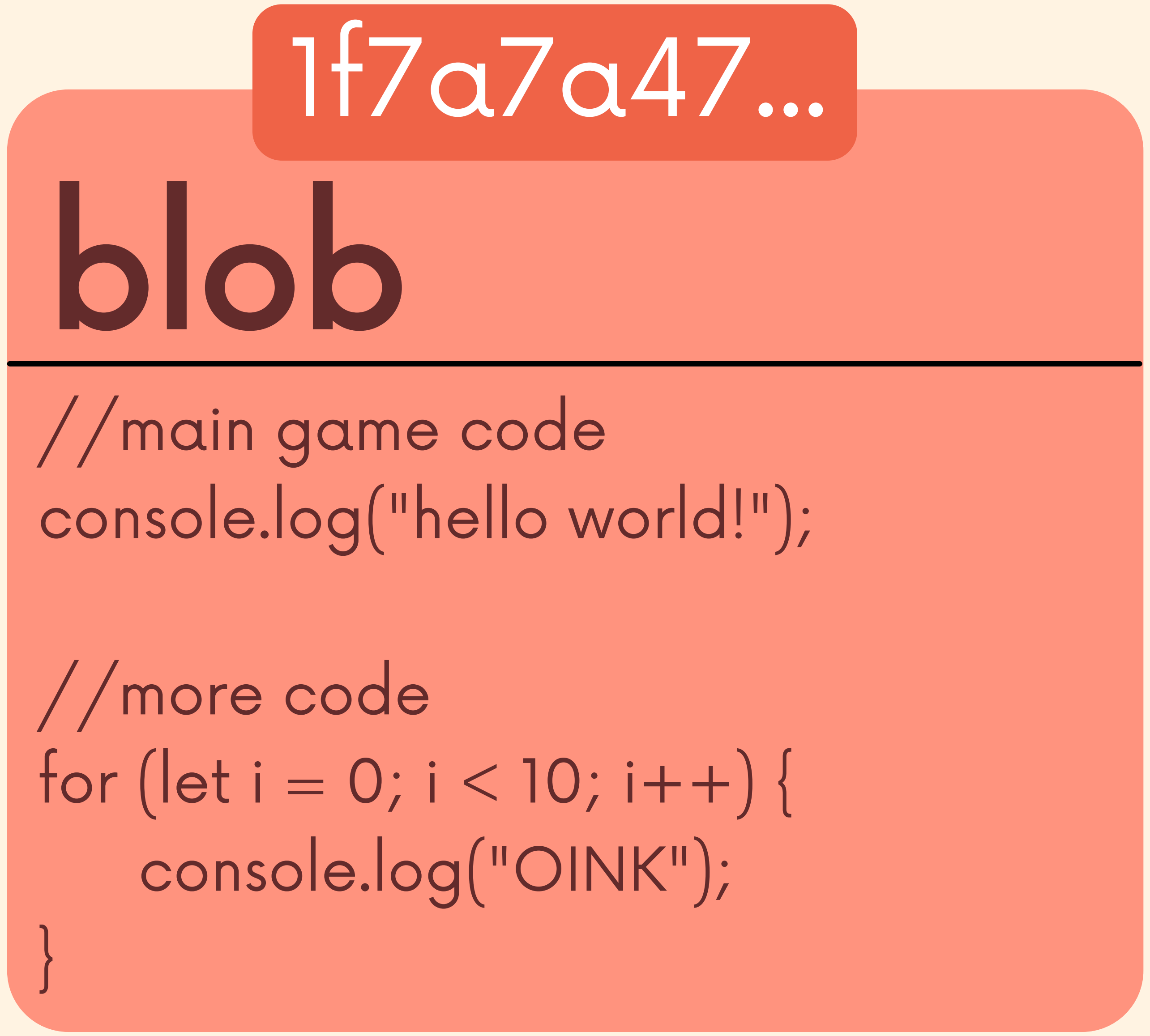
dd7e1c6f0fefe118
f0b63d9f10908c
460aa317a6

4768c08892615
8b50144c710989
e13fc697550fc

'goodbye'

Blobs

Git blobs (binary large object) are the object type Git uses to store the **contents of files** in a given repository. Blobs don't even include the filenames of each file or any other data. They just store the contents of a file!



1f7a7a47...

blob

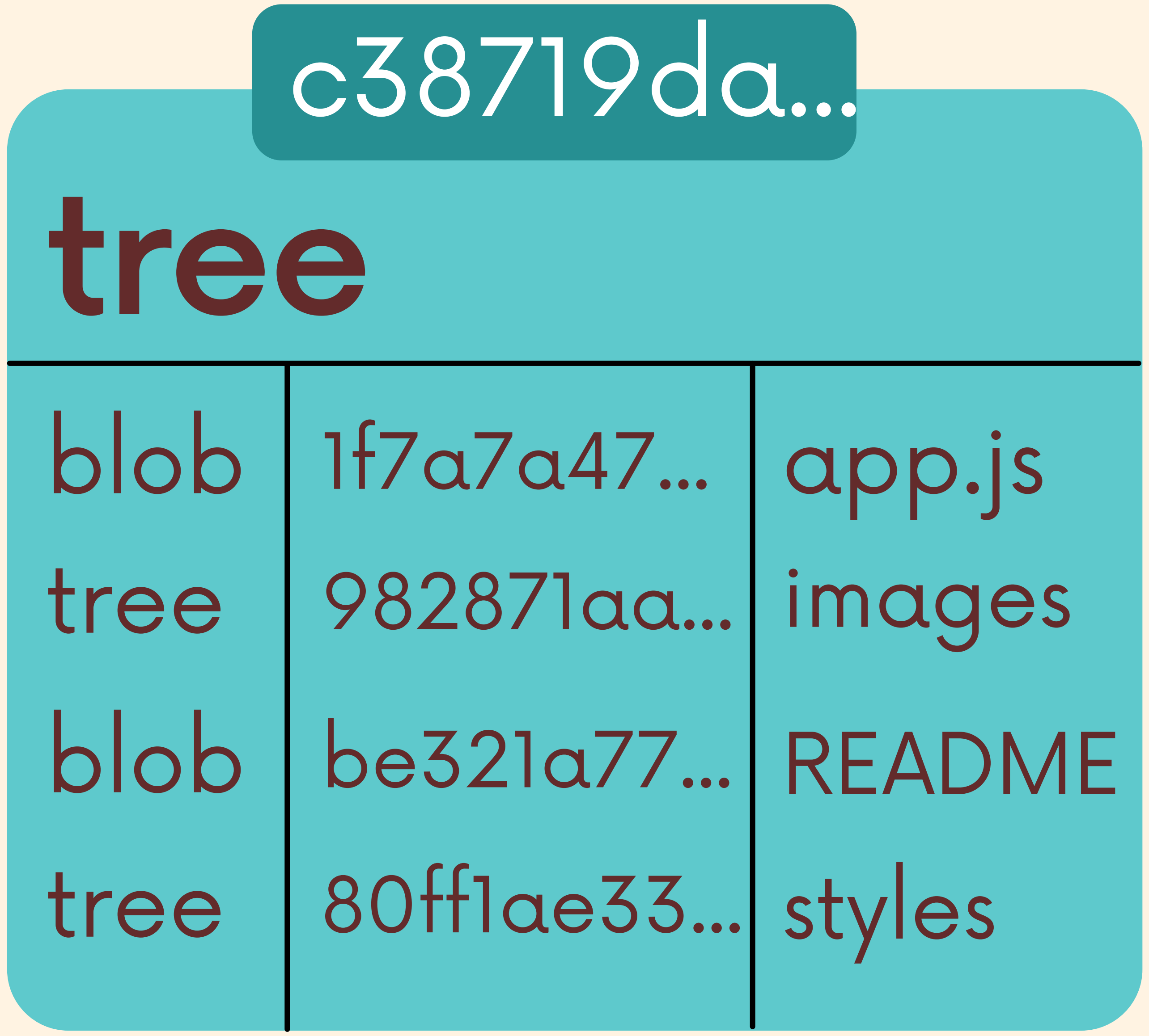
```
//main game code
console.log("hello world!");

//more code
for (let i = 0; i < 10; i++) {
    console.log("OINK");
}
```

Trees

Trees are Git objects used to store the contents of a directory. Each tree contains pointers that can refer to blobs and to other trees.

Each entry in a tree contains the SHA-1 hash of a blob or tree, as well as the mode, type, and filename

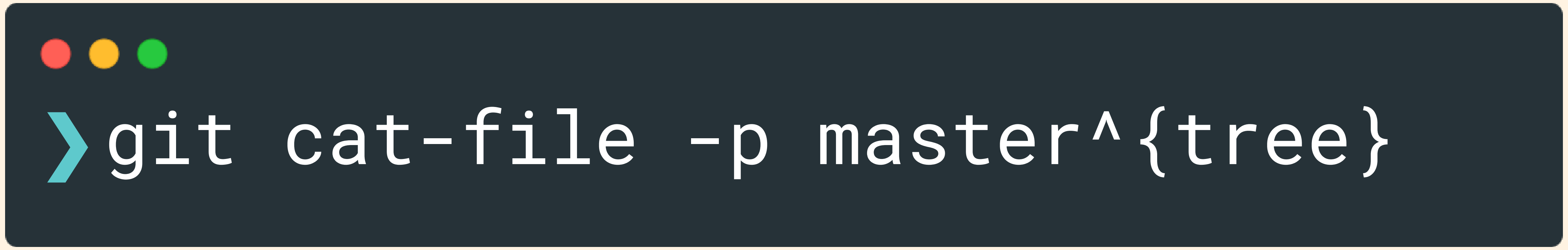


c38719da...

tree

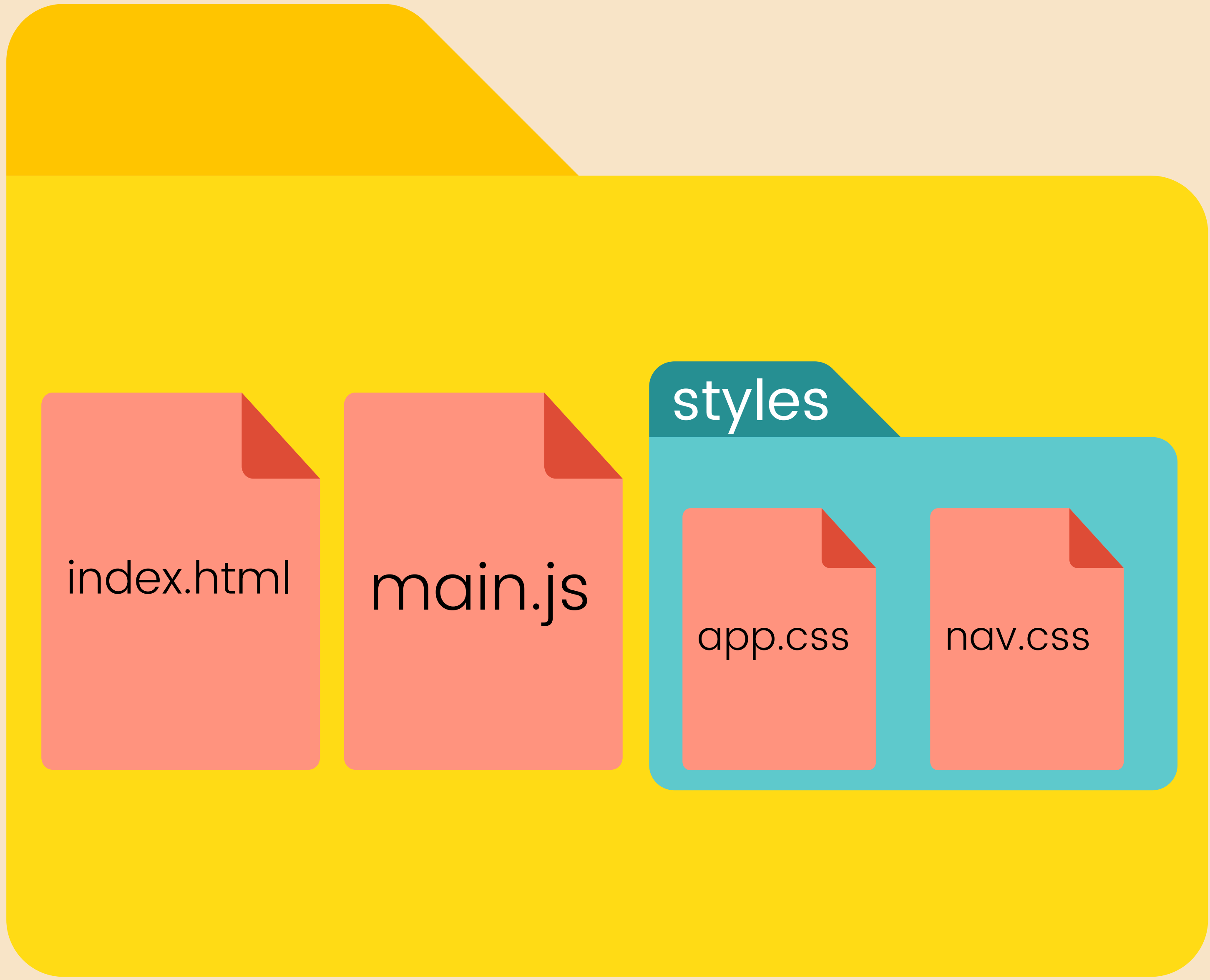
blob	1f7a7a47...	app.js
tree	982871aa...	images
blob	be321a77...	README
tree	80ff1ae33...	styles

Viewing Trees



```
❯git cat-file -p master^{tree}
```

Remember that **git cat-file** prints out Git objects. In this example, the **master^{tree}** syntax specifies the tree object that is pointed to by the tip of our master branch.



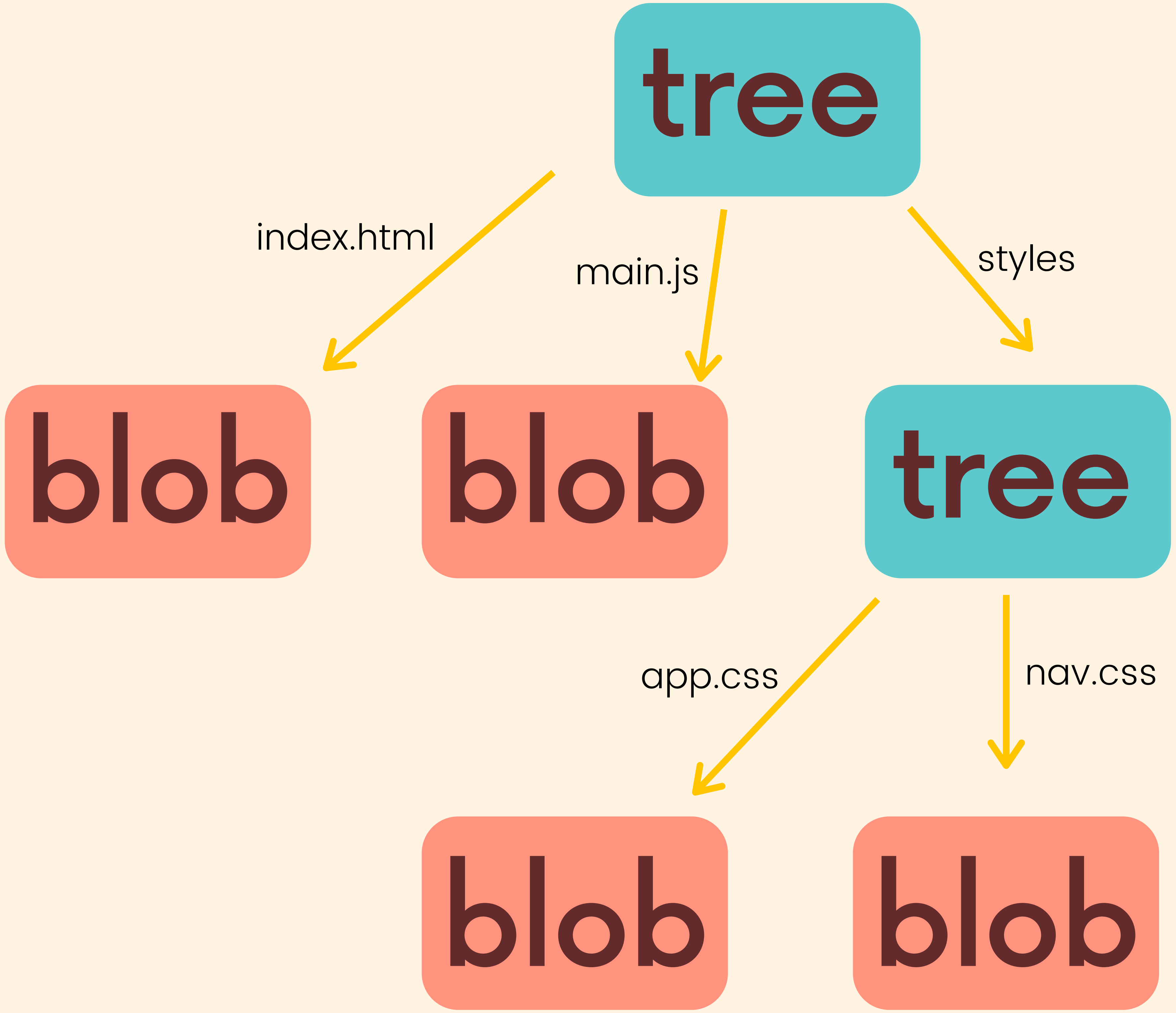
index.html

main.js

styles

app.css

nav.css



tree

index.html

main.js

styles

blob

blob

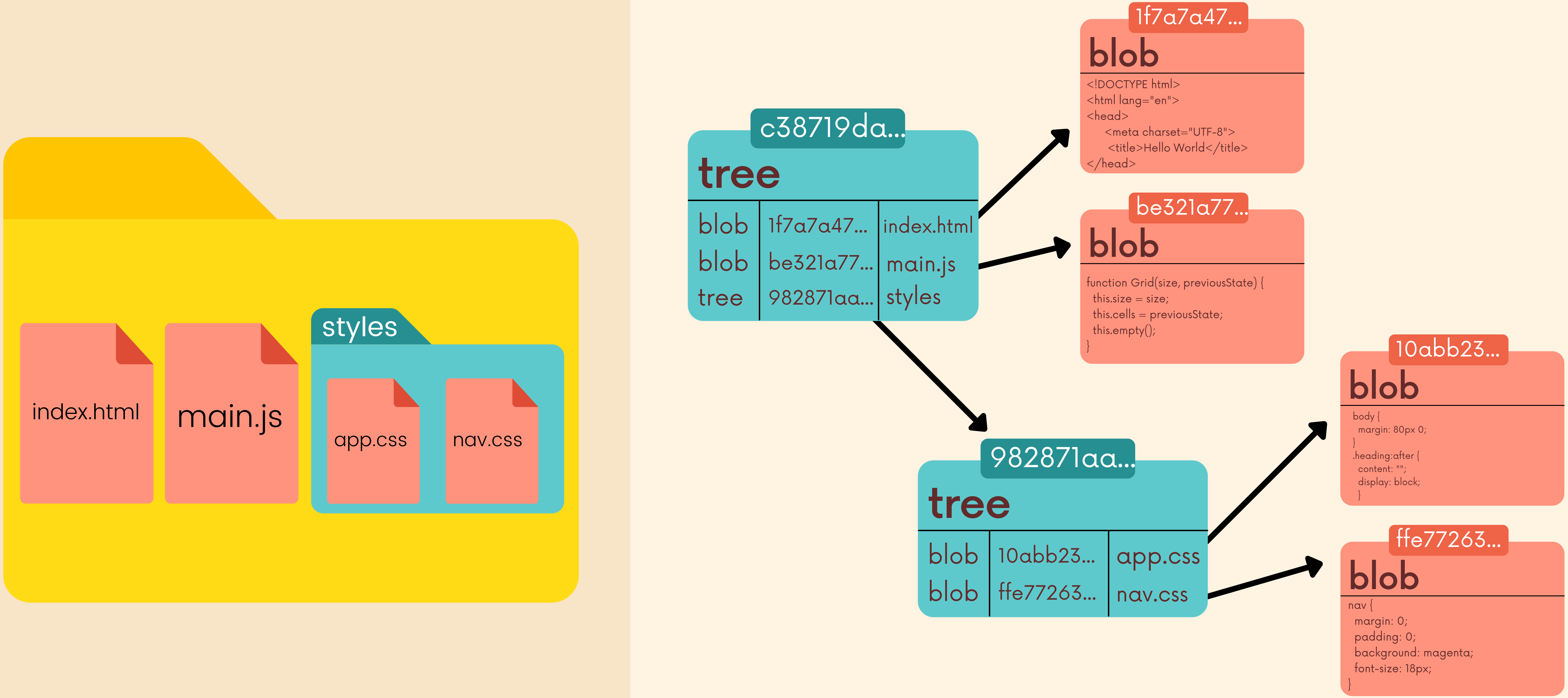
tree

app.css

nav.css

blob

blob



c38719da...

tree

blob	1f7a7a47...	index.html
blob	be321a77...	main.js
tree	982871aa...	styles

1f7a7a47...

blob

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
     <title>Hello World</title>
</head>
```

be321a77...

blob

```
function Grid(size, previousState) {
  this.size = size;
  this.cells = previousState;
  this.empty();
}
```

982871aa...

tree

blob	10abb23...	app.css
blob	ffe77263...	nav.css

10abb23...

blob

```
body {
  margin: 80px 0;
}
.heading:after {
  content: "";
  display: block;
  }
```

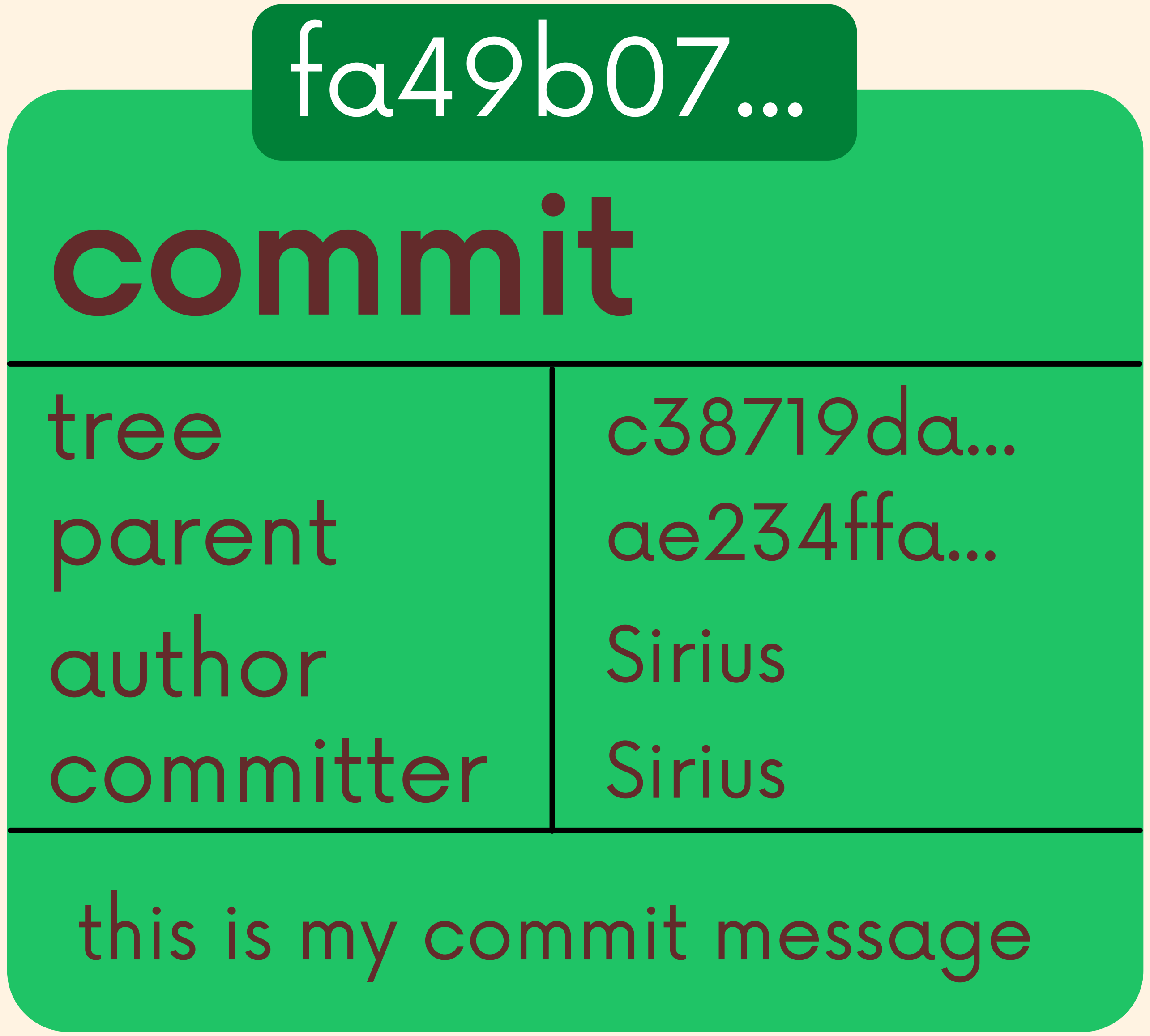
ffe77263...

blob

```
nav {
  margin: 0;
  padding: 0;
  background: magenta;
  font-size: 18px;
}
```

Commits

Commit objects combine a tree object along with information about the context that led to the current tree. Commits store a reference to parent commit(s), the author, the commiter, and of course the commit message!

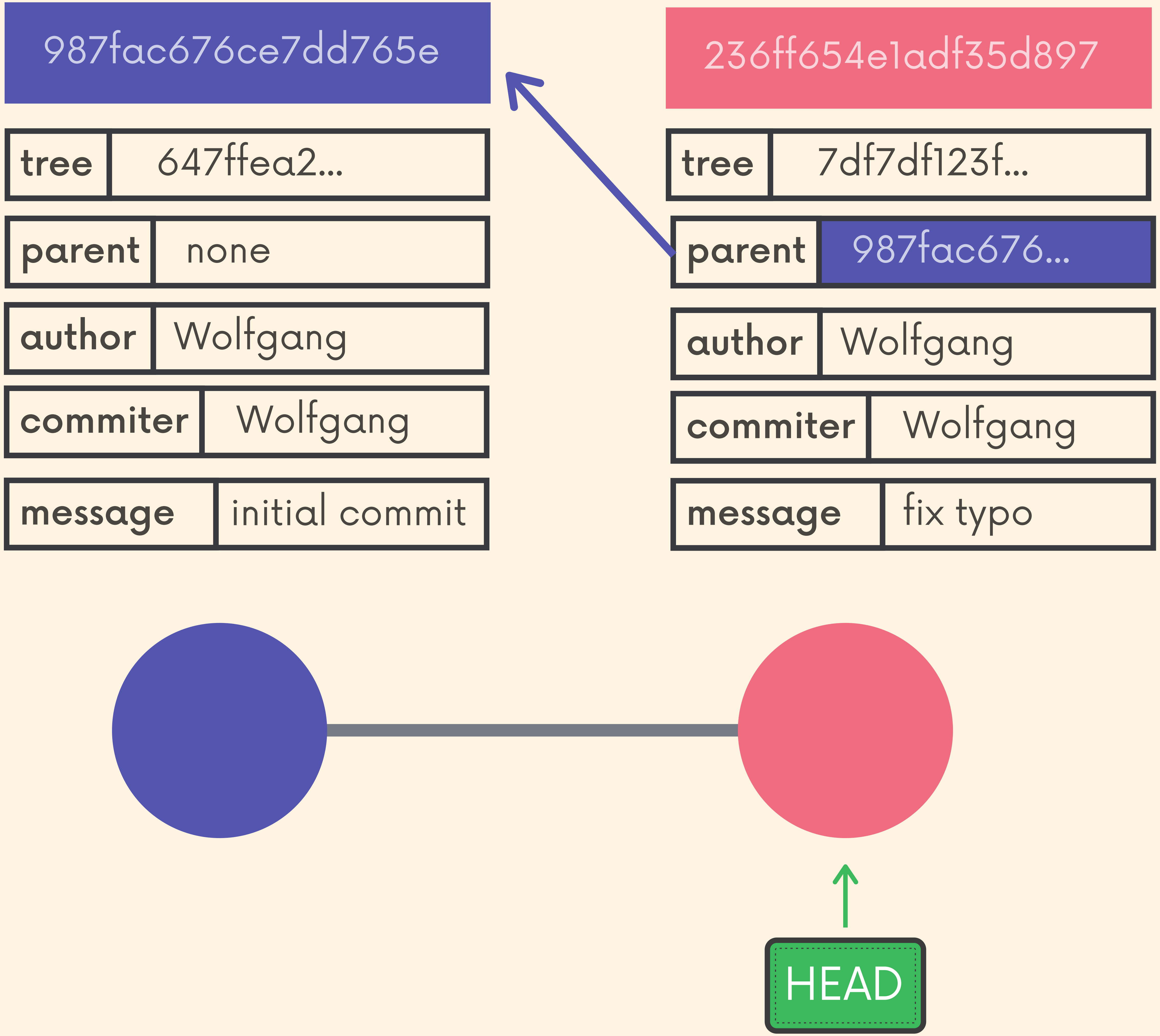


fa49b07...

commit

tree	c38719da...
parent	ae234ffa...
author	Sirius
committer	Sirius

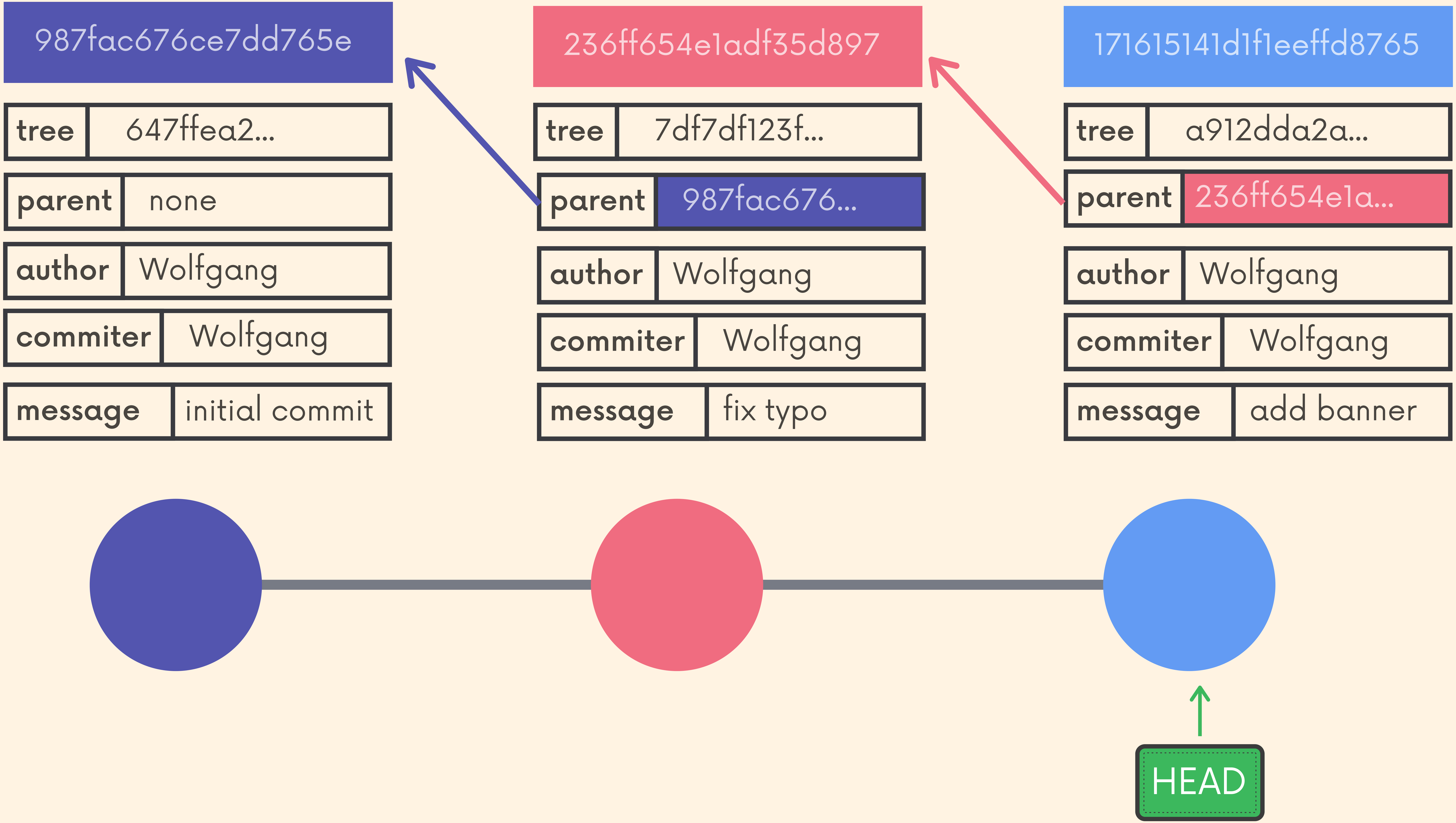
this is my commit message



987fac676ce7dd765e	
tree	647ffea2...
parent	none
author	Wolfgang
commiter	Wolfgang
message	initial commit

236ff654e1adf35d897	
tree	7df7df123f...
parent	987fac676...
author	Wolfgang
commiter	Wolfgang
message	fix typo

When we run git commit, Git creates a new commit object whose parent is the **current HEAD commit** and whose tree is the current content of the index.



987fac676ce7dd765e

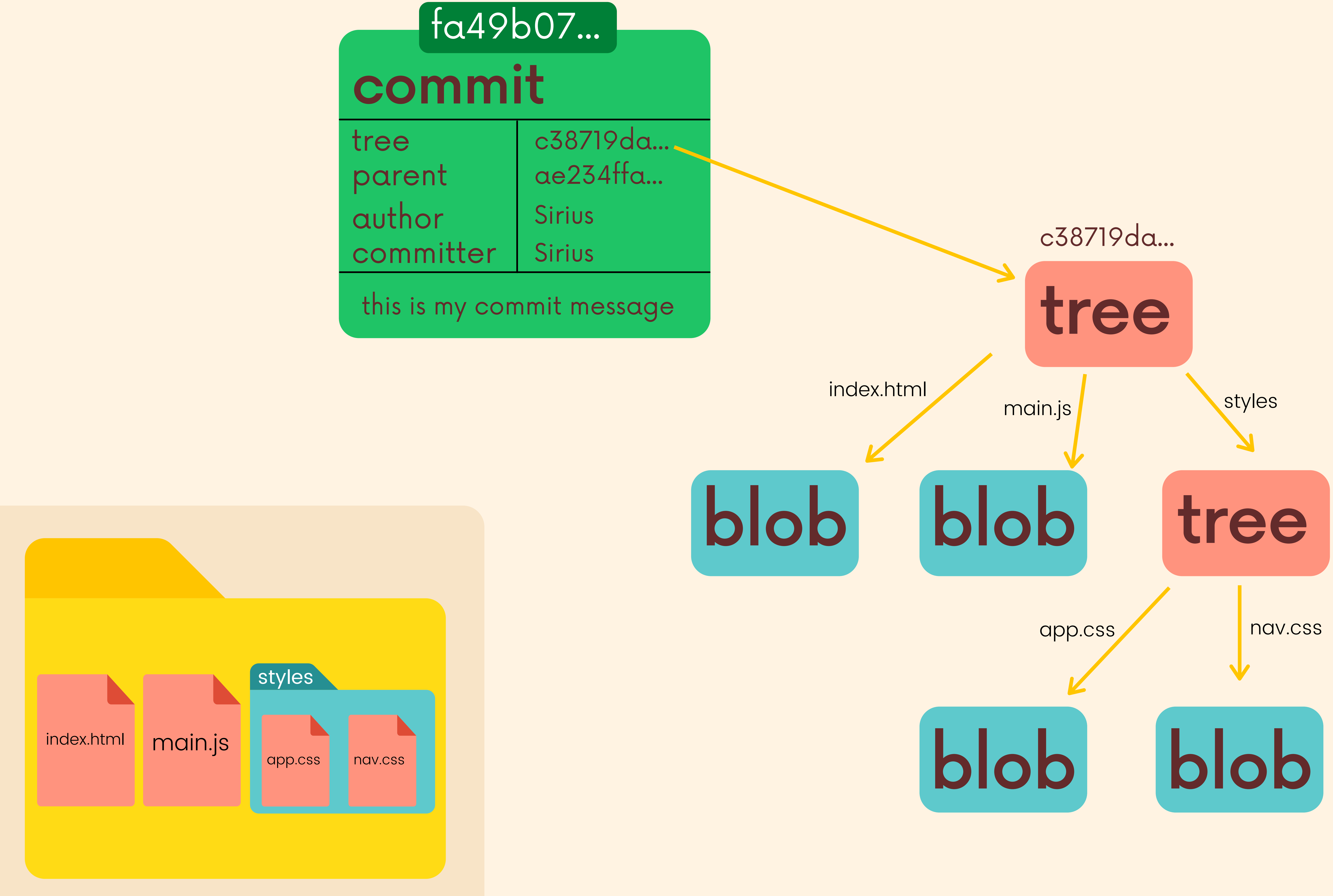
tree	647ffea2...
parent	none
author	Wolfgang
commiter	Wolfgang
message	initial commit

236ff654e1adf35d897

tree	7df7df123f...
parent	987fac676...
author	Wolfgang
commiter	Wolfgang
message	fix typo

171615141d1f1eeffd8765

tree	a912dda2a...
parent	236ff654e1a...
author	Wolfgang
commiter	Wolfgang
message	add banner

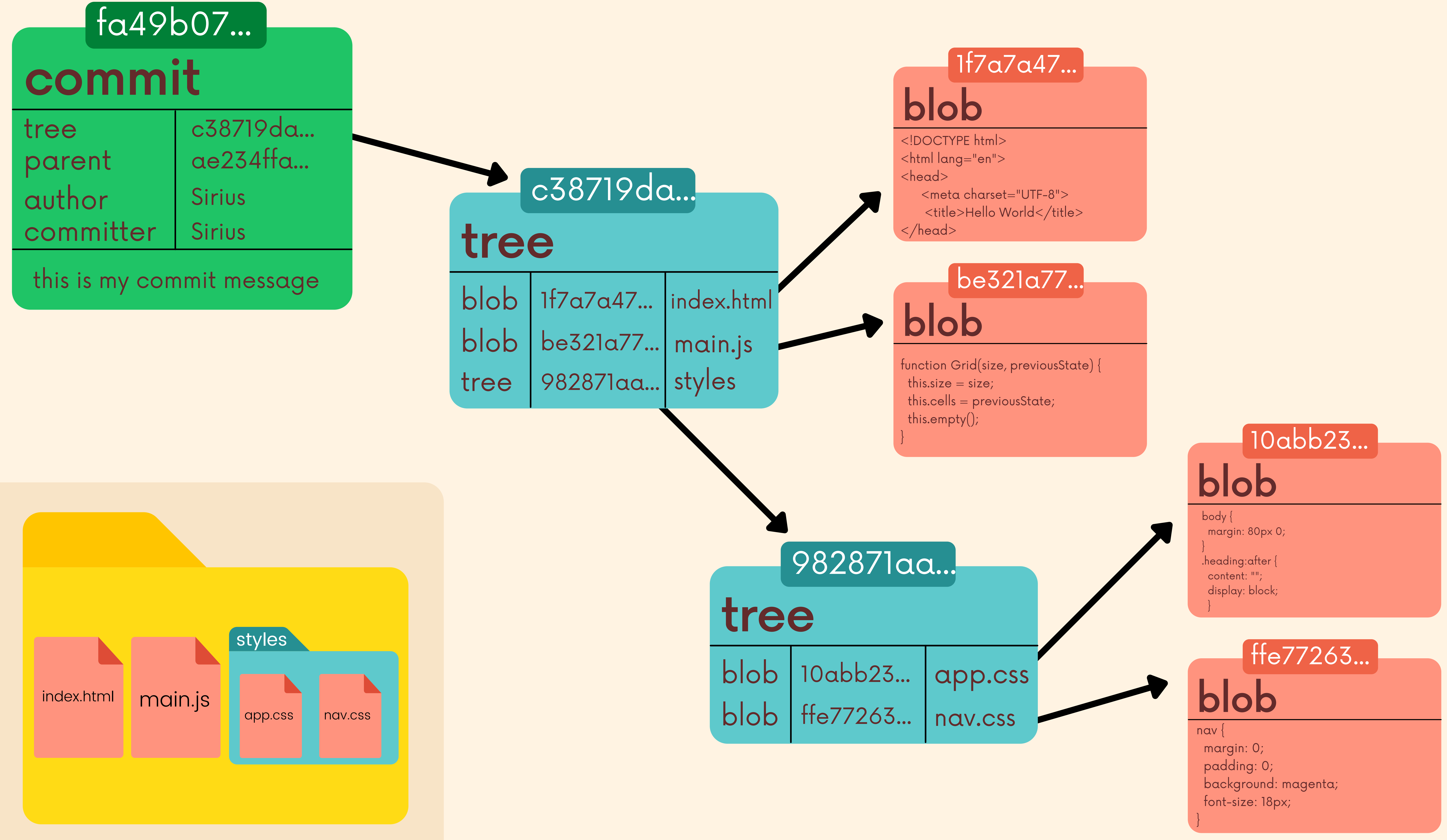


fa49b07...

commit

tree	c38719da...
parent	ae234ffa...
author	Sirius
committer	Sirius

this is my commit message



fa49b07...

commit

tree	c38719da...
parent	ae234ffa...
author	Sirius
committer	Sirius

this is my commit message

c38719da...

tree

blob	1f7a7a47...	index.html
blob	be321a77...	main.js
tree	982871aa...	styles

982871aa...

tree

blob	10abb23...	app.css
blob	ffe77263...	nav.css

1f7a7a47...

blob

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
     <title>Hello World</title>
</head>
```

be321a77...

blob

```
function Grid(size, previousState) {
  this.size = size;
  this.cells = previousState;
  this.empty();
}
```

10abb23...

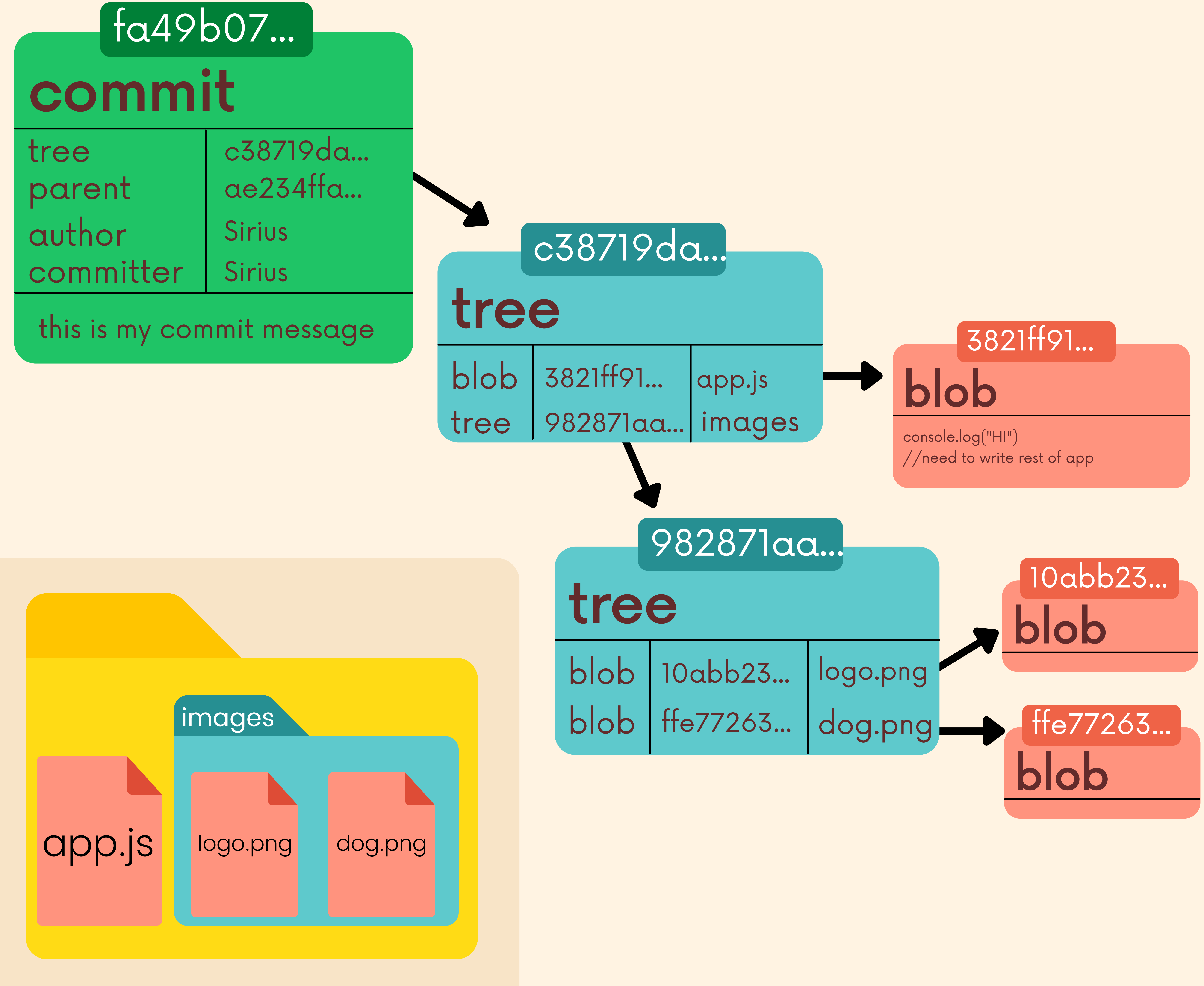
blob

```
body {
  margin: 80px 0;
}
.heading:after {
  content: "";
  display: block;
  }
```

ffe77263...

blob

```
nav {
  margin: 0;
  padding: 0;
  background: magenta;
  font-size: 18px;
}
```



fa49b07...

commit

tree	c38719da...
parent	ae234ffa...
author	Sirius
committer	Sirius

this is my commit message

c38719da...

tree

blob	3821ff91...	app.js
tree	982871aa...	images

3821ff91...

blob

```
console.log("HI")
//need to write rest of app
```

982871aa...

tree

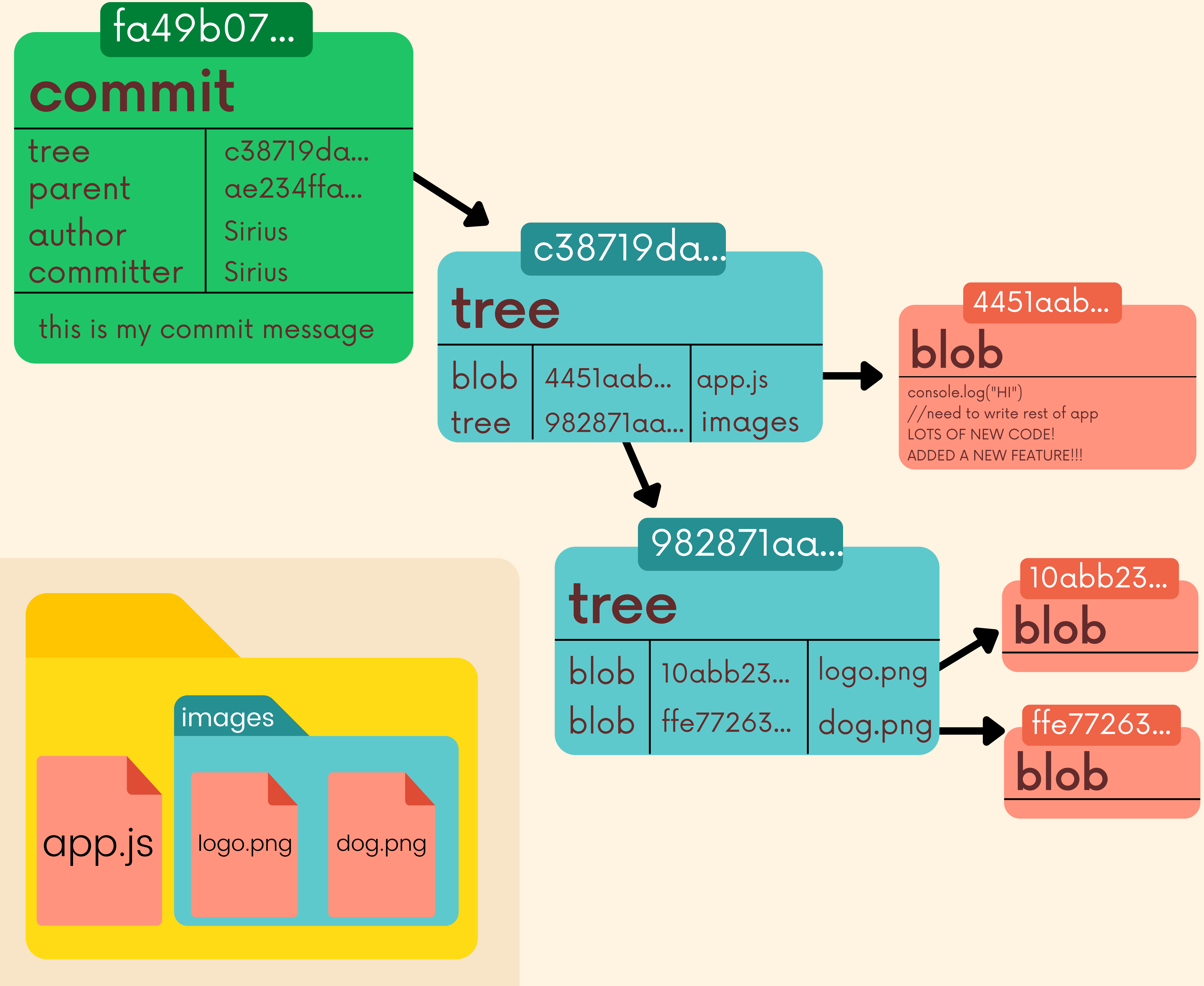
blob	10abb23...	logo.png
blob	ffe77263...	dog.png

10abb23...

blob

ffe77263...

blob



fa49b07...

commit

tree	c38719da...
parent	ae234ffa...
author	Sirius
committer	Sirius

this is my commit message

c38719da...

tree

blob	4451aab...	app.js
tree	982871aa...	images

4451aab...

blob

console.log("HI")
//need to write rest of app
LOTS OF NEW CODE!
ADDED A NEW FEATURE!!!

982871aa...

tree

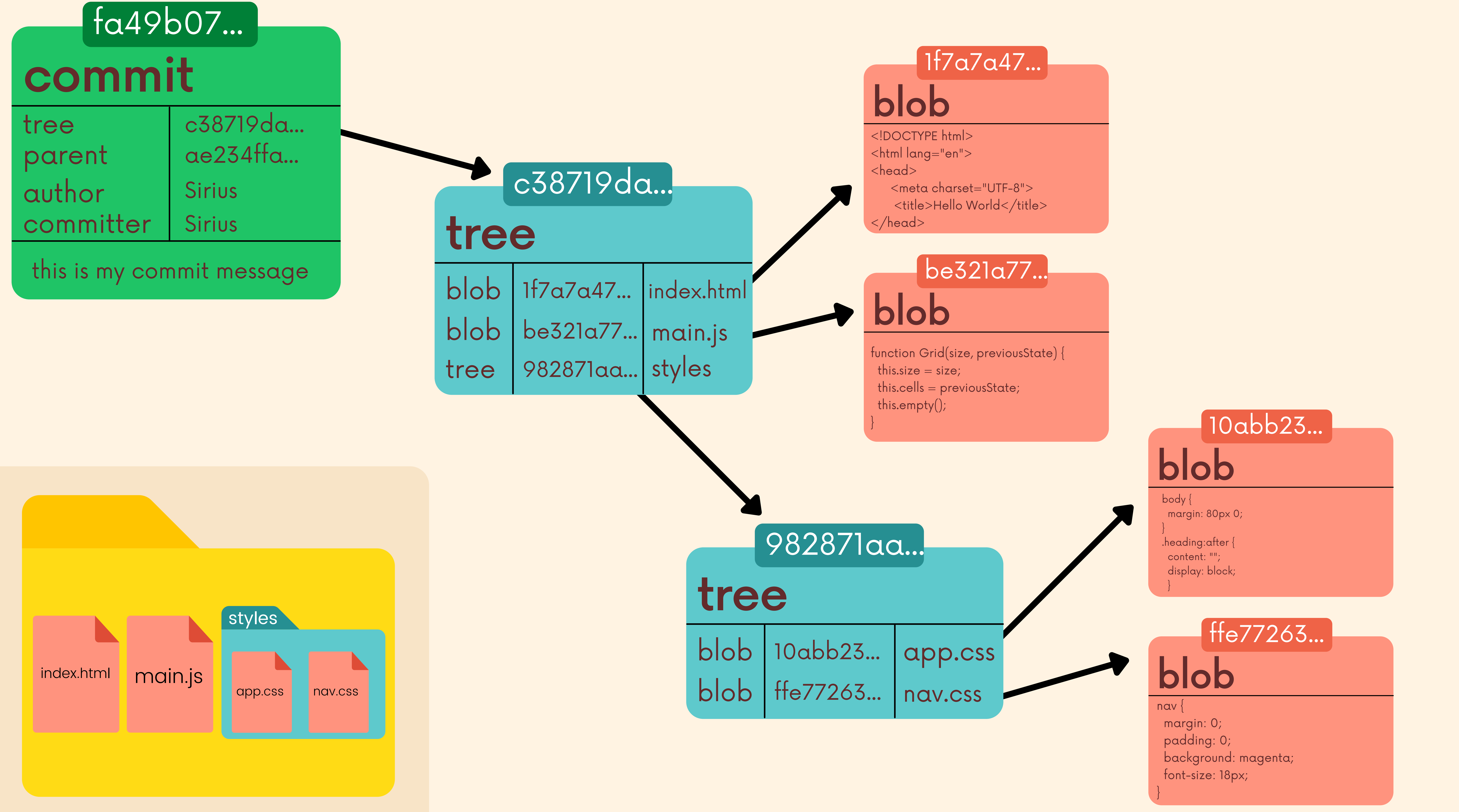
blob	10abb23...	logo.png
blob	ffe77263...	dog.png

10abb23...

blob

ffe77263...

blob



fa49b07...

commit

tree	c38719da...
parent	ae234ffa...
author	Sirius
committer	Sirius

this is my commit message

c38719da...

tree

blob	1f7a7a47...	index.html
blob	be321a77...	main.js
tree	982871aa...	styles

982871aa...

tree

blob	10abb23...	app.css
blob	ffe77263...	nav.css

1f7a7a47...

blob

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
     <title>Hello World</title>
</head>
```

be321a77...

blob

```
function Grid(size, previousState) {
  this.size = size;
  this.cells = previousState;
  this.empty();
}
```

10abb23...

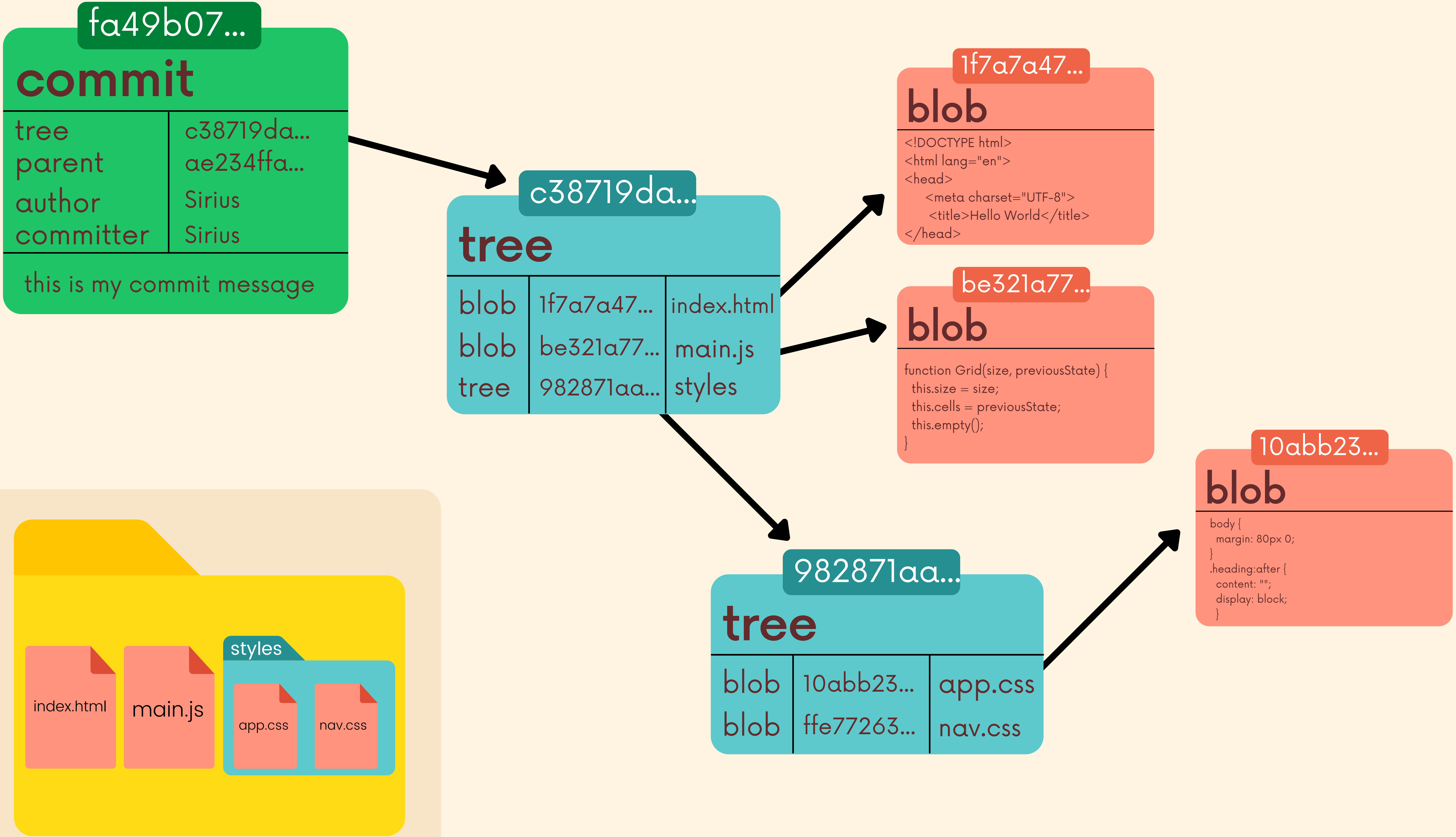
blob

```
body {
  margin: 80px 0;
}
.heading:after {
  content: "";
  display: block;
}
```

ffe77263...

blob

```
nav {
  margin: 0;
  padding: 0;
  background: magenta;
  font-size: 18px;
}
```



fa49b07...

commit

tree	c38719da...
parent	ae234ffa...
author	Sirius
committer	Sirius

this is my commit message

c38719da...

tree

blob	1f7a7a47...	index.html
blob	be321a77...	main.js
tree	982871aa...	styles

982871aa...

tree

blob	10abb23...	app.css
blob	ffe77263...	nav.css

1f7a7a47...

blob

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
     <title>Hello World</title>
</head>
```

be321a77...

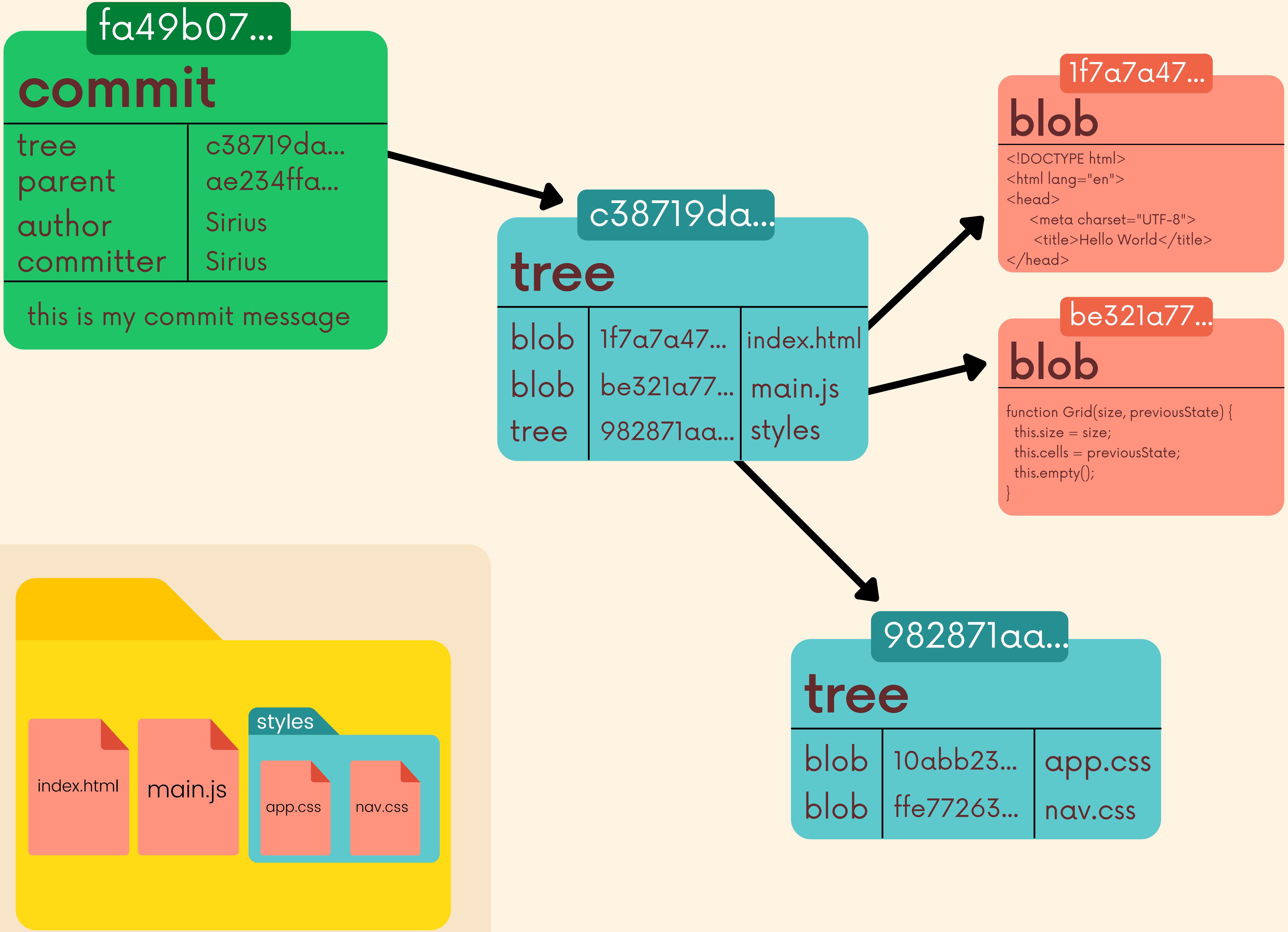
blob

```
function Grid(size, previousState) {
  this.size = size;
  this.cells = previousState;
  this.empty();
}
```

10abb23...

blob

```
body {
  margin: 80px 0;
}
.heading:after {
  content: "";
  display: block;
}
```



fa49b07...

commit

tree	c38719da...
parent	ae234ffa...
author	Sirius
committer	Sirius

this is my commit message

c38719da...

tree

blob	1f7a7a47...	index.html
blob	be321a77...	main.js
tree	982871aa...	styles

1f7a7a47...

blob

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
     <title>Hello World</title>
</head>
```

be321a77...

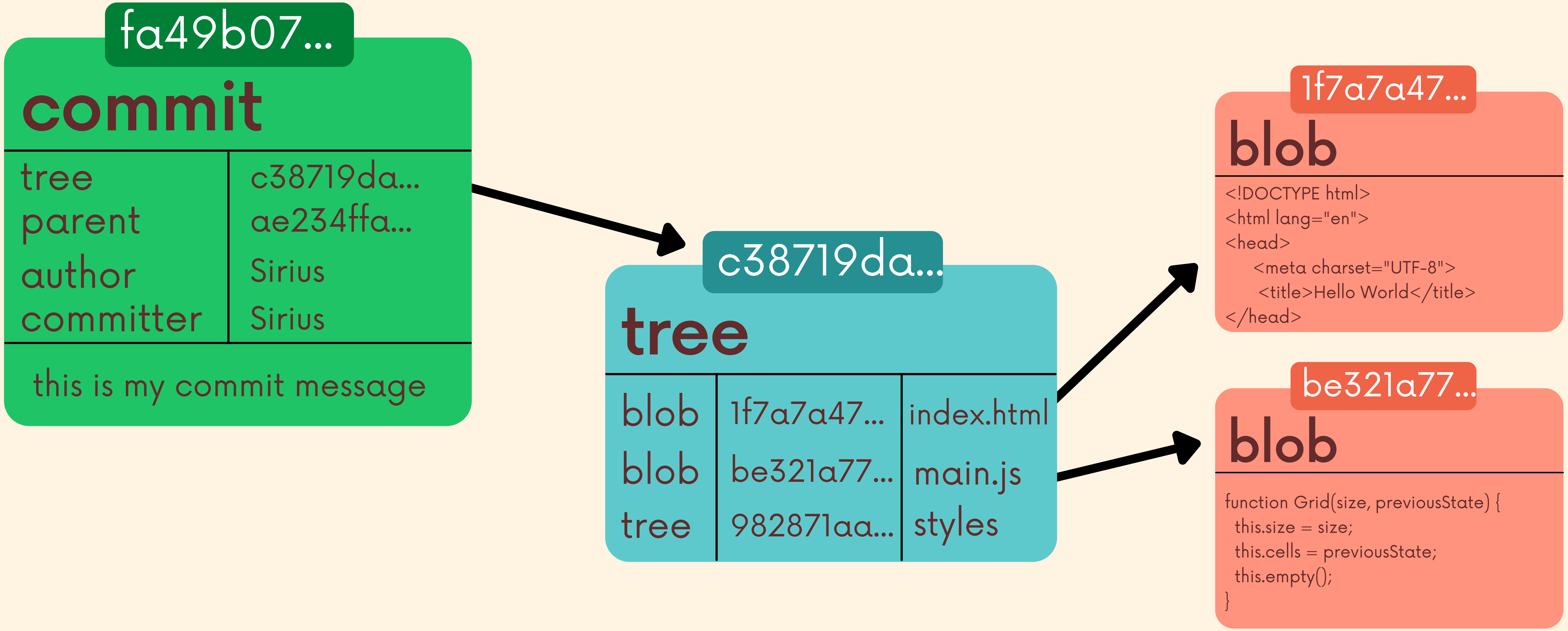
blob

```
function Grid(size, previousState) {
  this.size = size;
  this.cells = previousState;
  this.empty();
}
```

982871aa...

tree

blob	10abb23...	app.css
blob	ffe77263...	nav.css



fa49b07...

commit

tree	c38719da...
parent	ae234ffa...
author	Sirius
committer	Sirius

this is my commit message

c38719da...

tree

blob	1f7a7a47...	index.html
blob	be321a77...	main.js
tree	982871aa...	styles

1f7a7a47...

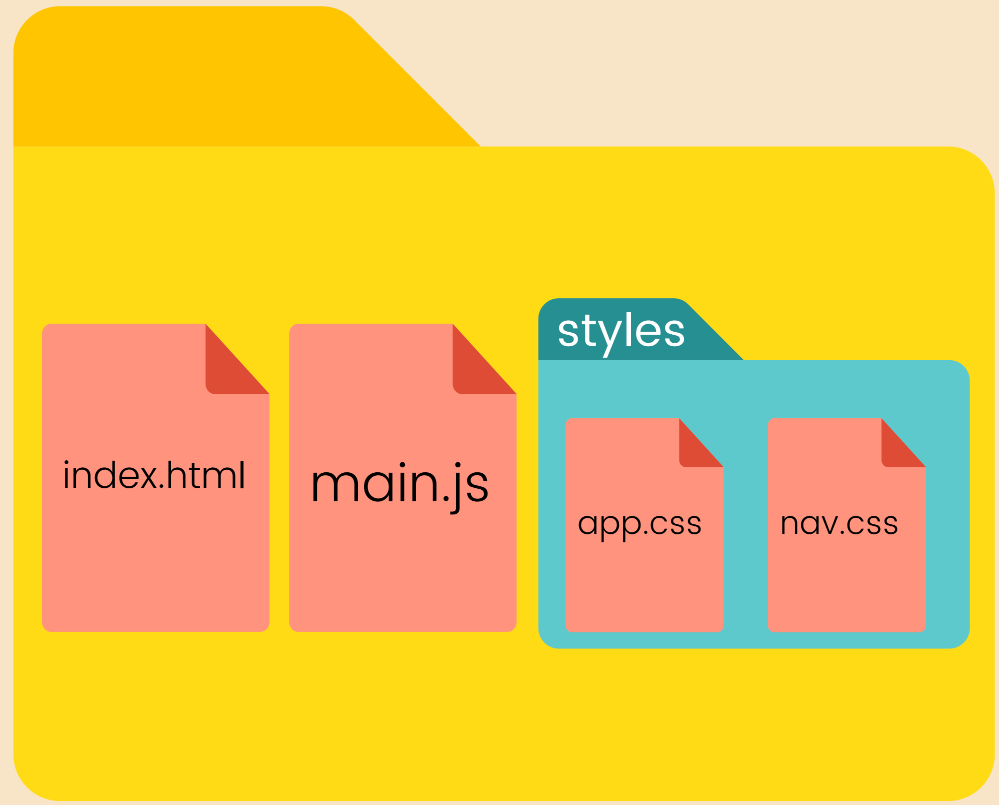
blob

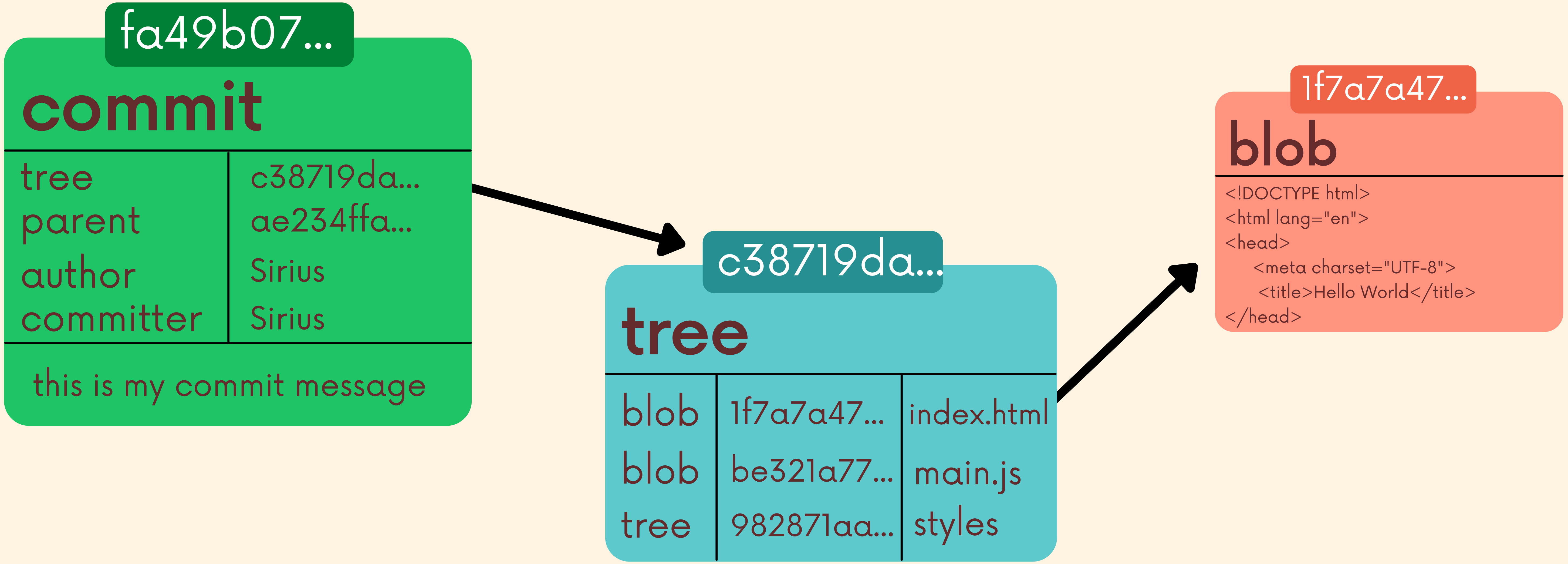
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
     <title>Hello World</title>
</head>
```

be321a77...

blob

```
function Grid(size, previousState) {
  this.size = size;
  this.cells = previousState;
  this.empty();
}
```





fa49b07...

commit

tree	c38719da...
parent	ae234ffa...
author	Sirius
committer	Sirius

this is my commit message

c38719da...

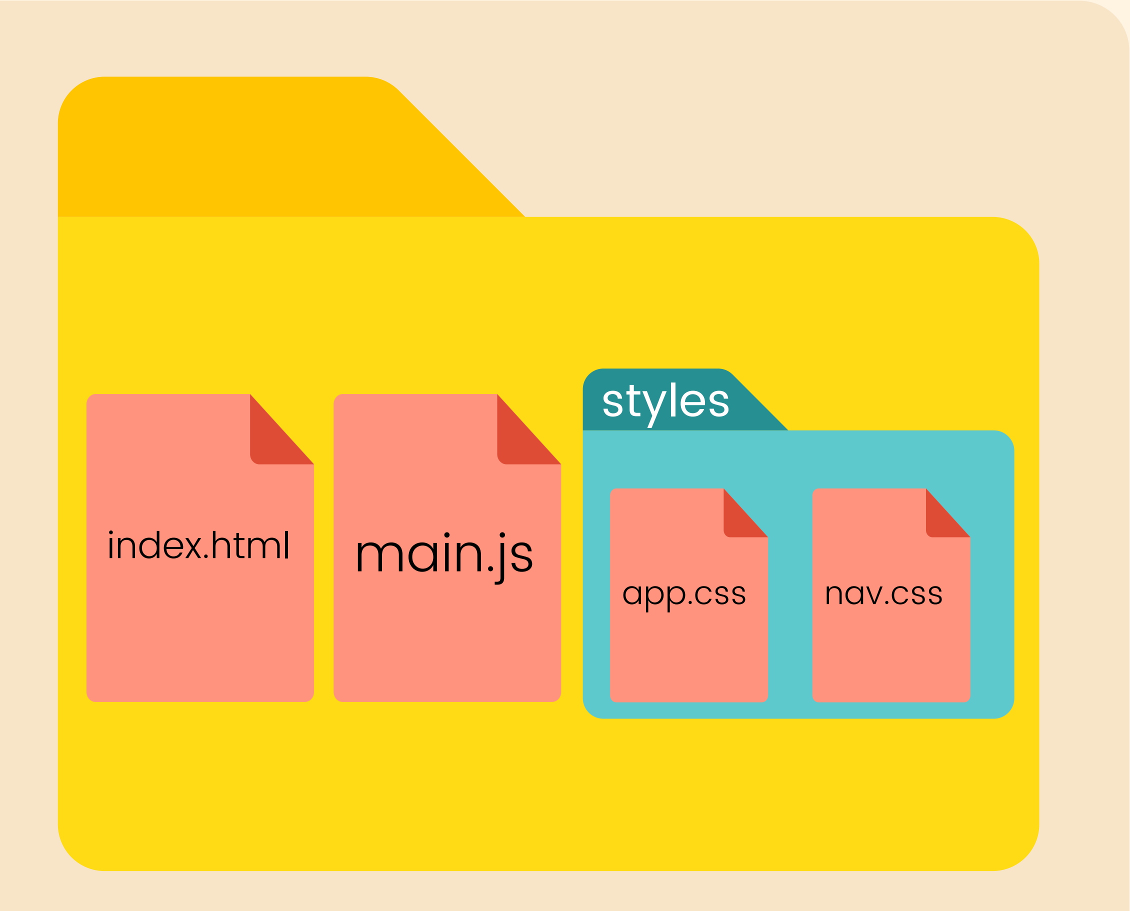
tree

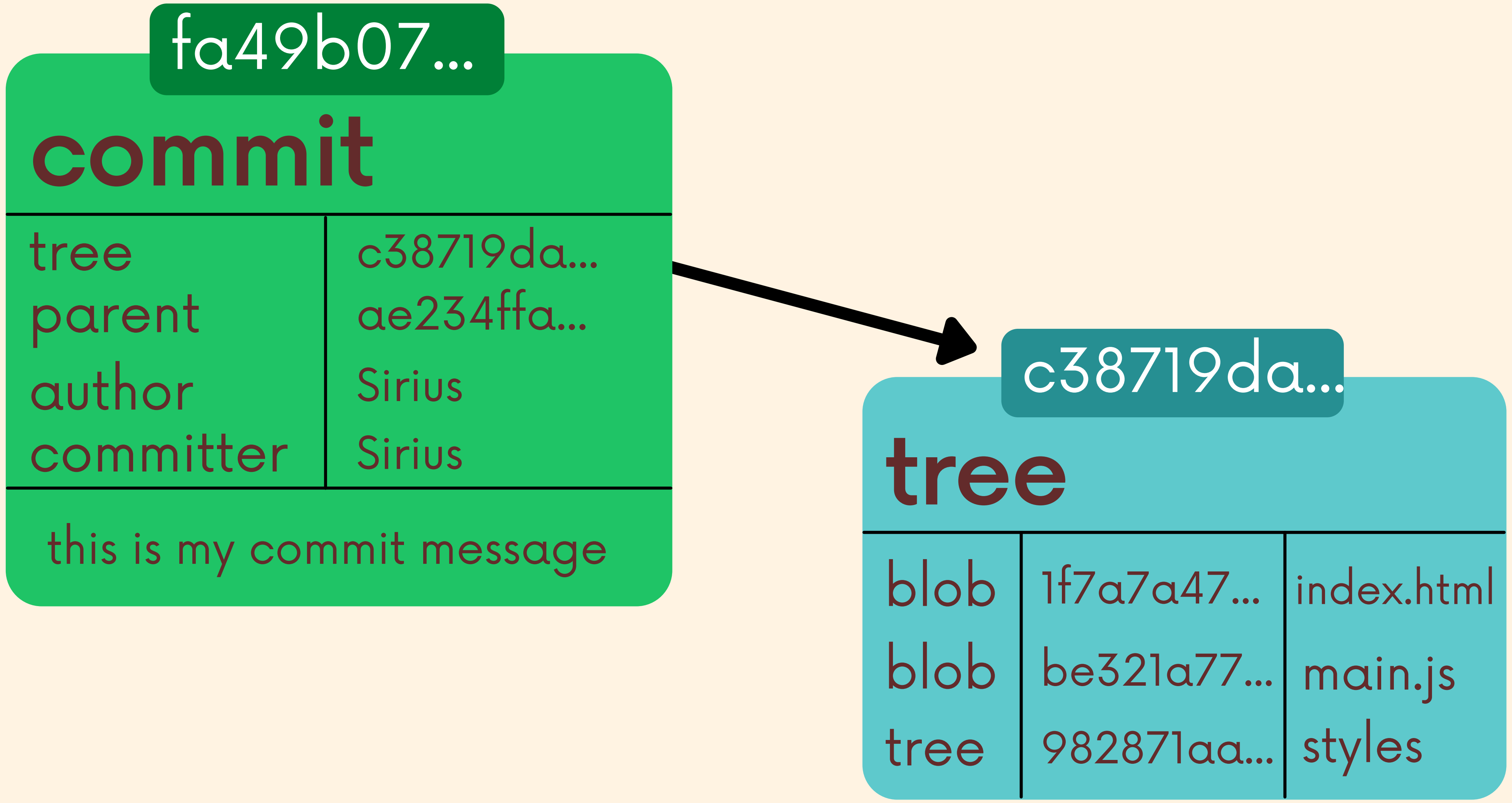
blob	1f7a7a47...	index.html
blob	be321a77...	main.js
tree	982871aa...	styles

1f7a7a47...

blob

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
     <title>Hello World</title>
</head>
```





fa49b07...

commit

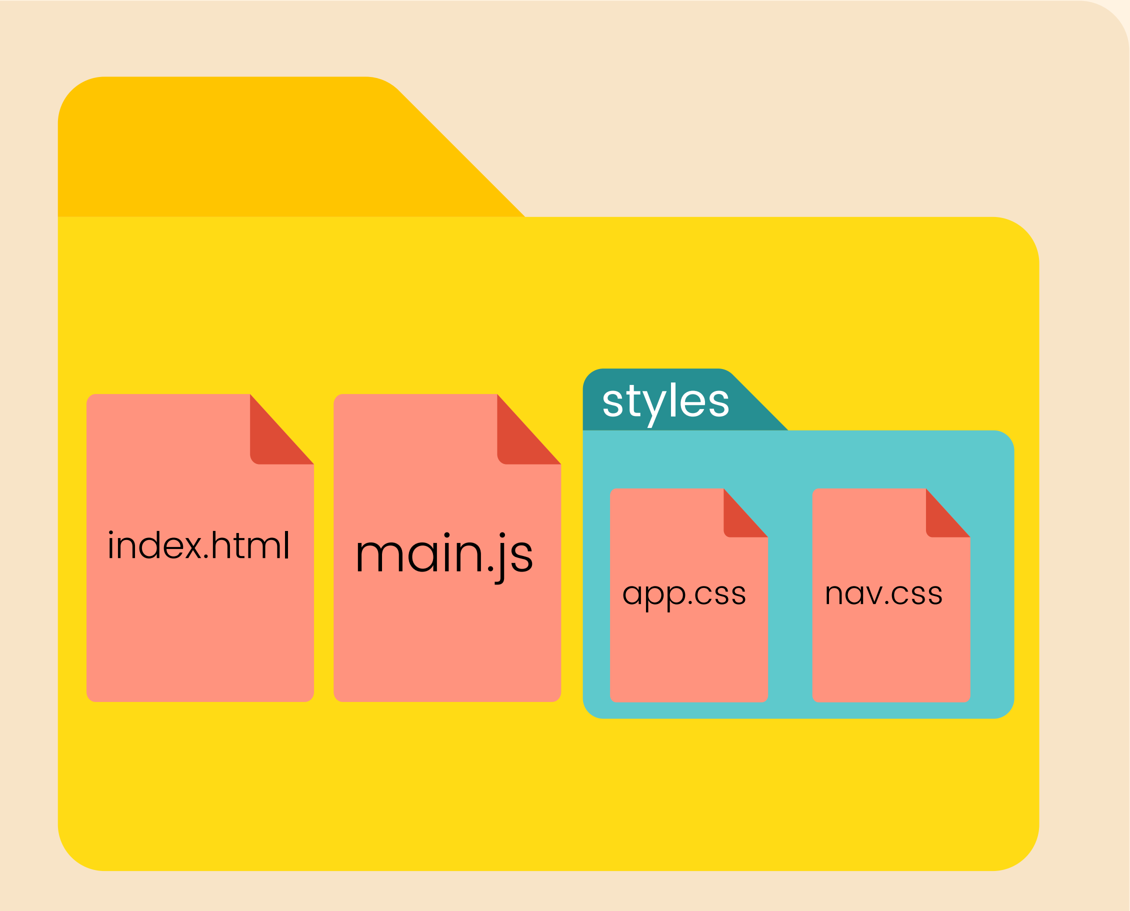
tree	c38719da...
parent	ae234ffa...
author	Sirius
committer	Sirius

this is my commit message

c38719da...

tree

blob	1f7a7a47...	index.html
blob	be321a77...	main.js
tree	982871aa...	styles

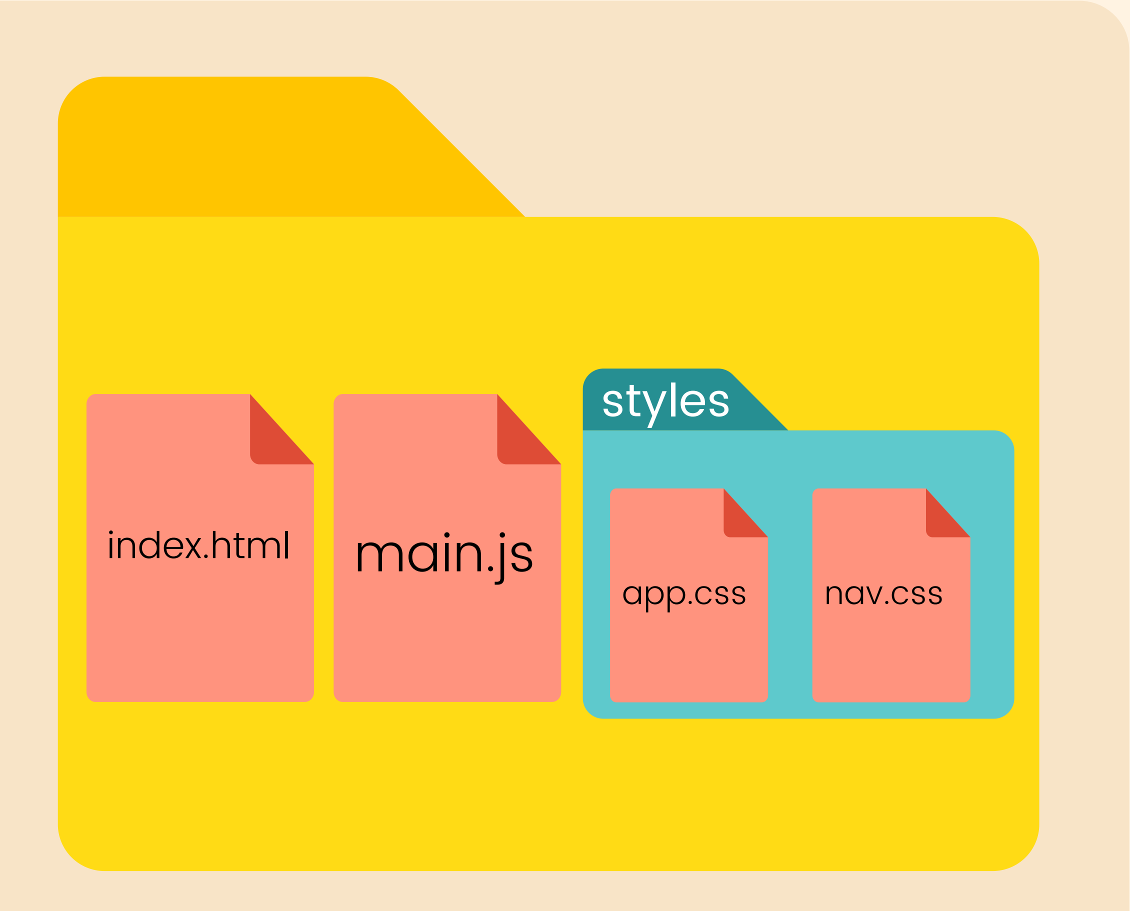


fa49b07...

commit

tree	c38719da...
parent	ae234ffa...
author	Sirius
committer	Sirius

this is my commit message


index.html
main.js
styles
app.css
nav.css