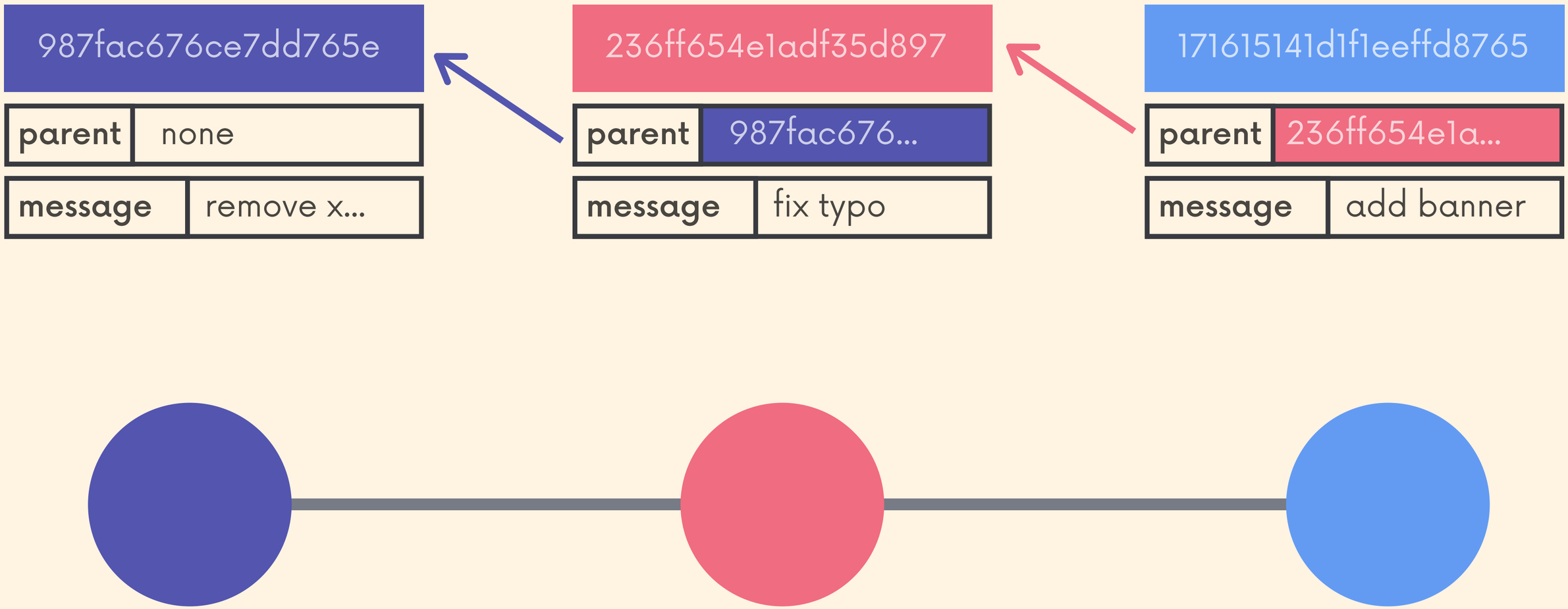
Git Branching


987fac676ce7dd765e
parent
none
message
remove x...
236ff654e1adf35d897
parent
987fac676...
message
fix typo
171615141d1f1eeffd8765
parent
236ff654e1a...
message
add banner

Contexts

On large projects, we often work in multiple contexts:

1. You're working on 2 different color scheme variations for your website at the same time, unsure of which you like best
2. You're also trying to fix a horrible bug, but it's proving tough to solve. You need to really hunt around and toggle some code on and off to figure it out.
3. A teammate is also working on adding a new chat widget to present at the next meeting. It's unclear if your company will end up using it.
4. Another coworker is updating the search bar autocomplete.
5. Another developer is doing an experimental radical design overhaul of the entire layout to present next month.

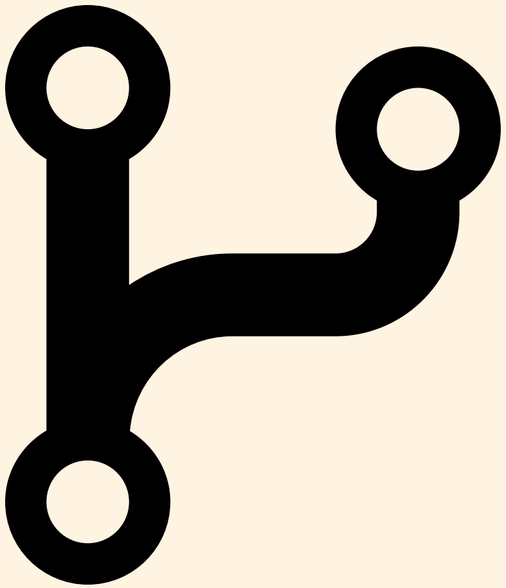
Branches

Branches are an essential part of Git!

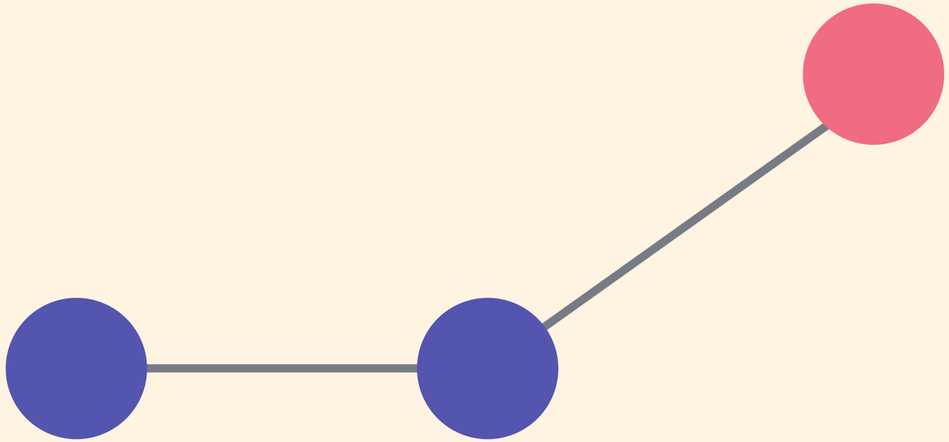
Think of branches as alternative timelines for a project.

They enable us to create separate contexts where we can try new things, or even work on multiple ideas in parallel.

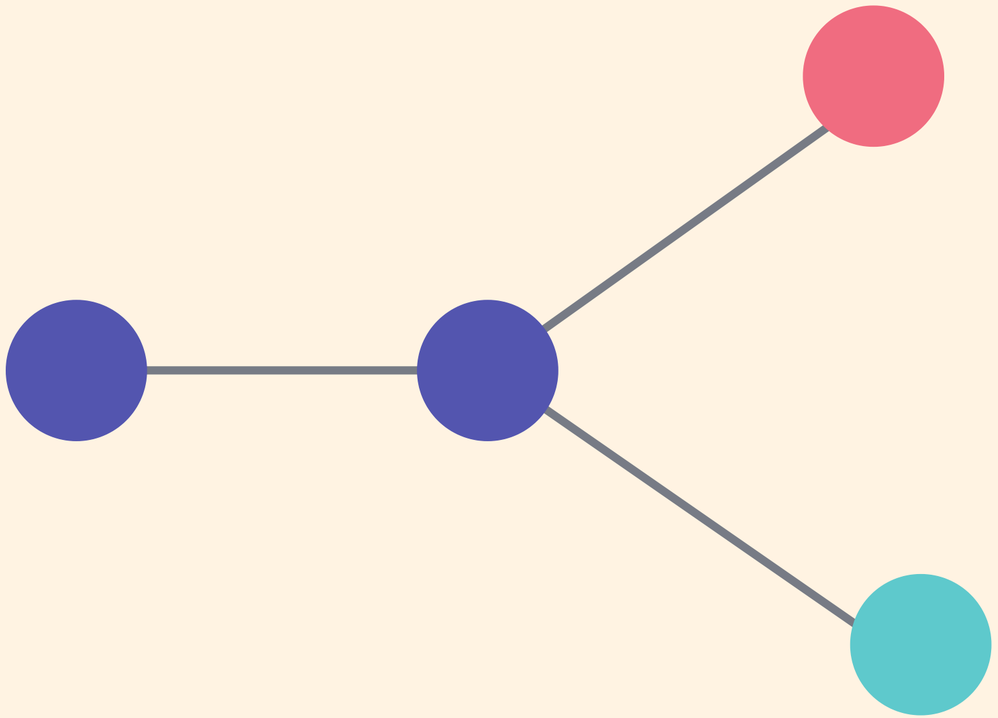
If we make changes on one branch, they do not impact the other branches (unless we merge the changes)



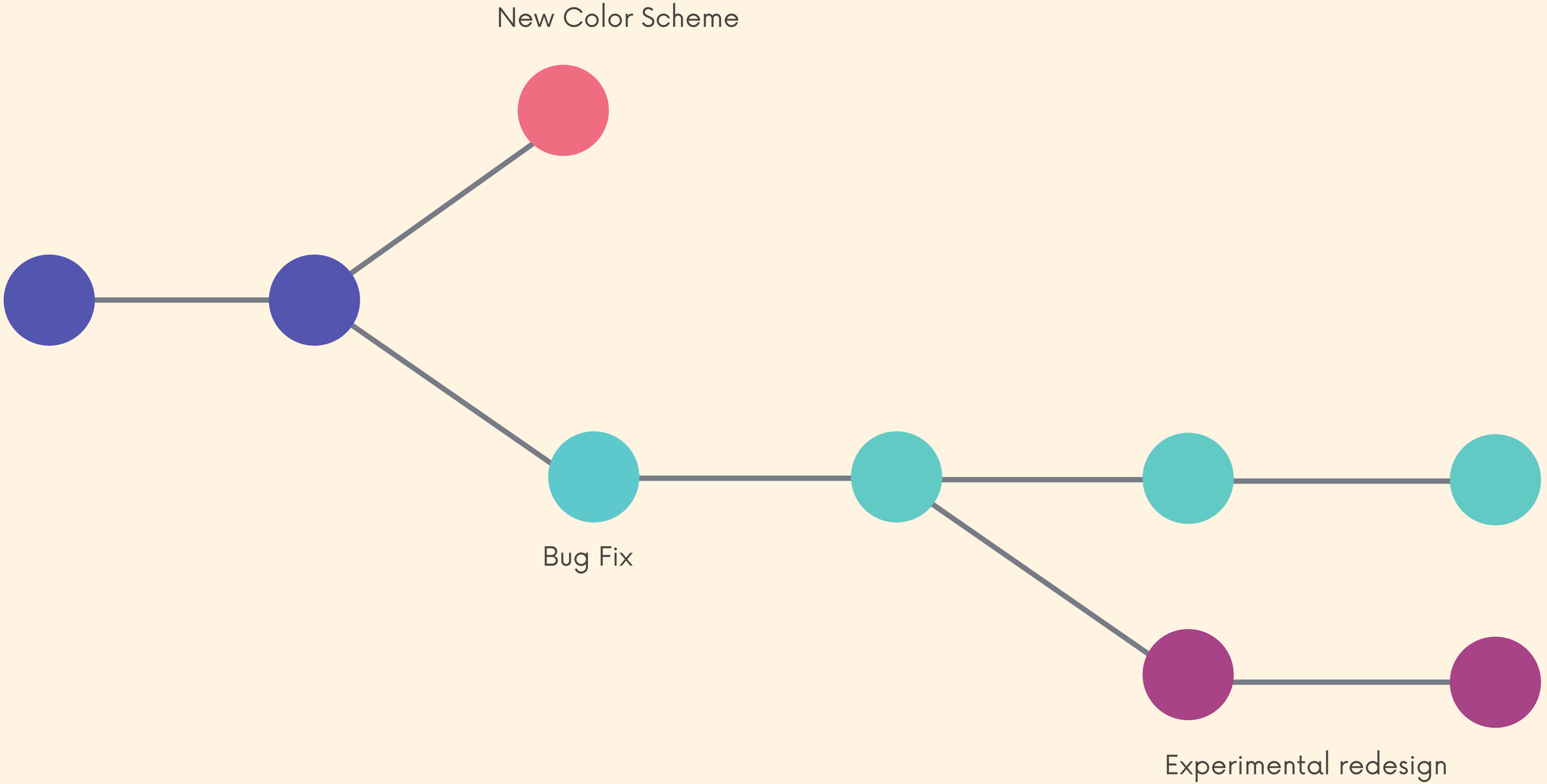
New Color Scheme

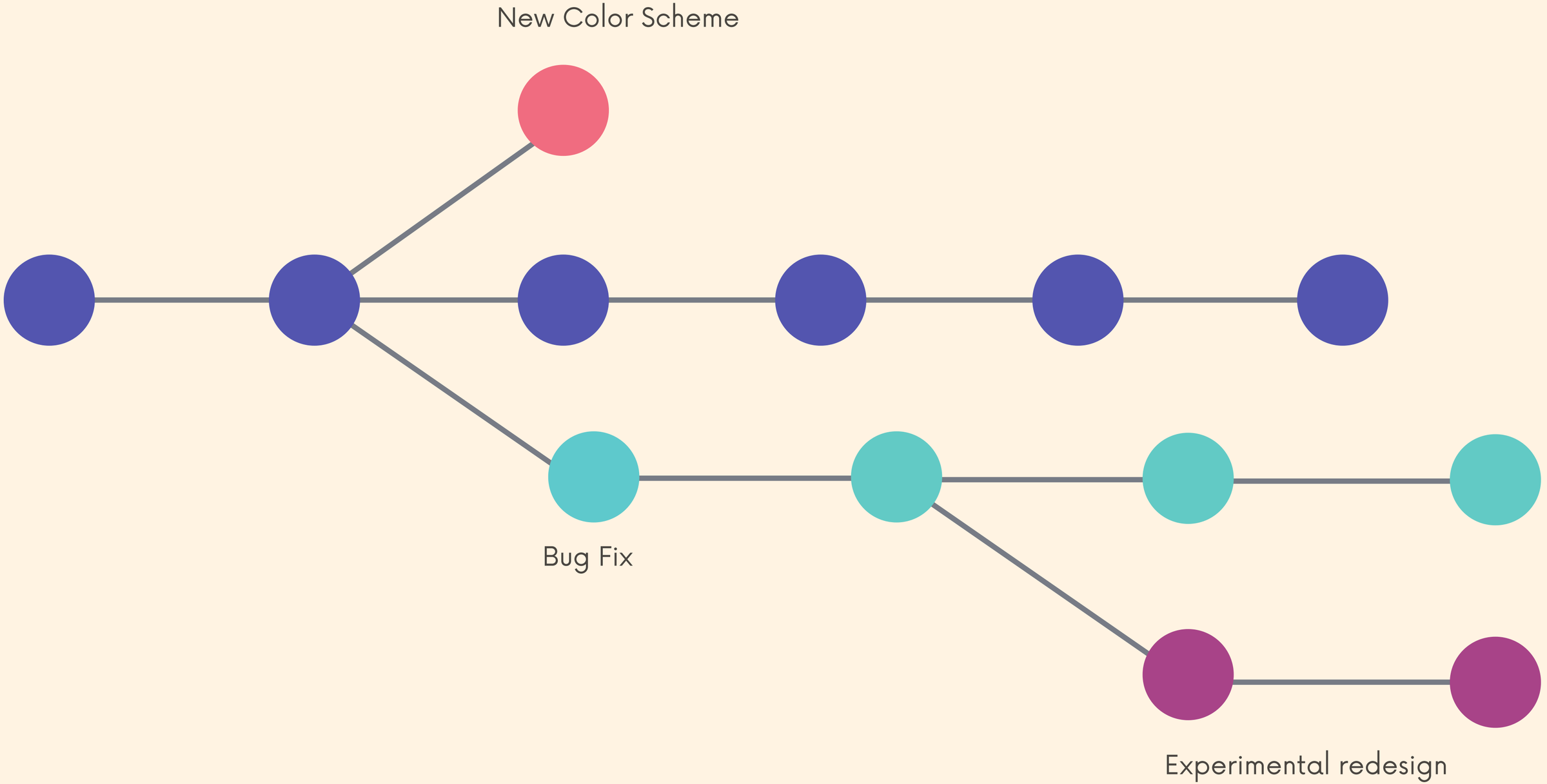


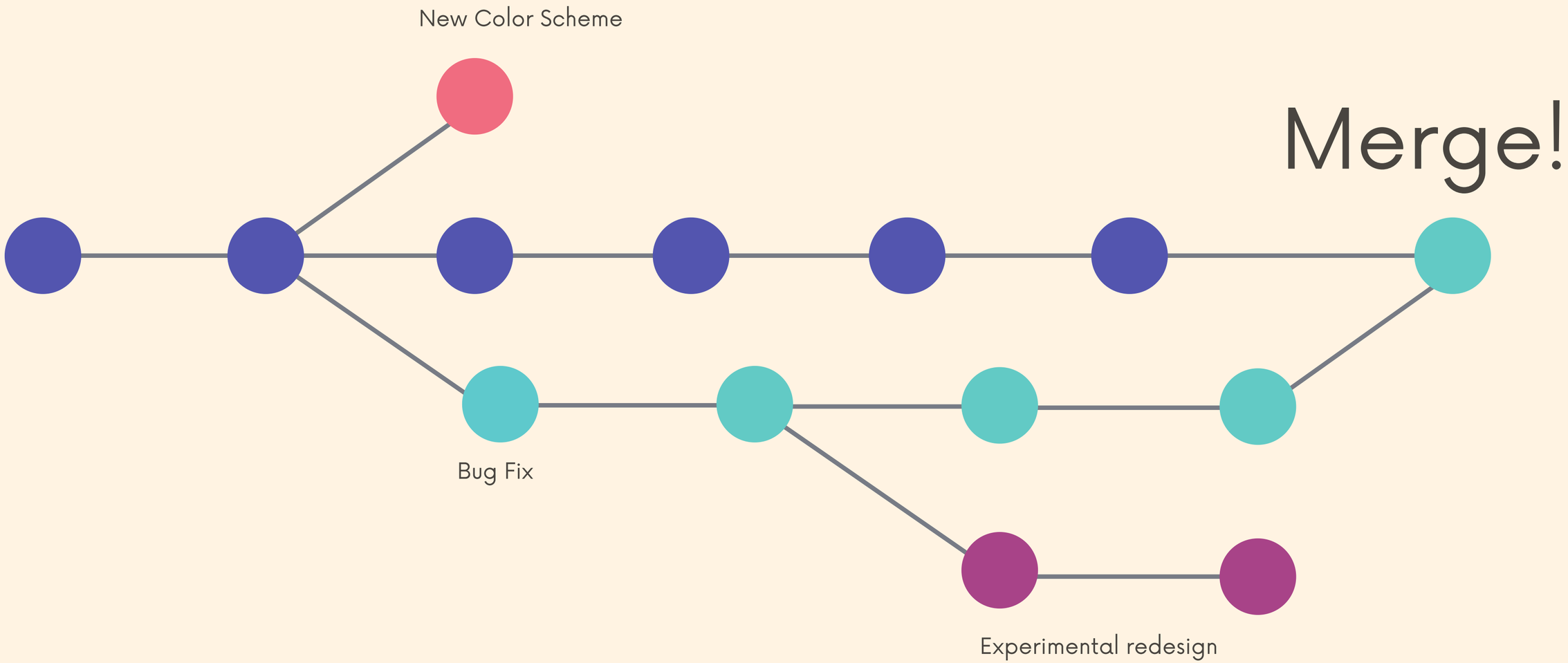
New Color Scheme



Bug Fix


New Color Scheme
Bug Fix
Experimental redesign


New Color Scheme
Bug Fix
Experimental redesign


New Color Scheme
Merge!
Bug Fix
Experimental redesign

You Probably Have Seen This Before

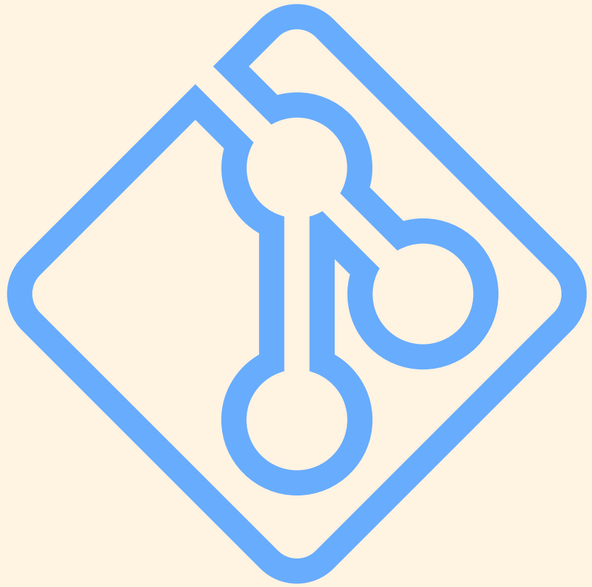
```
On branch master
nothing to commit
```

The Master Branch

In git, we are always working on a branch* The default branch name is **master**.

It doesn't do anything special or have fancy powers. It's just like any other branch.

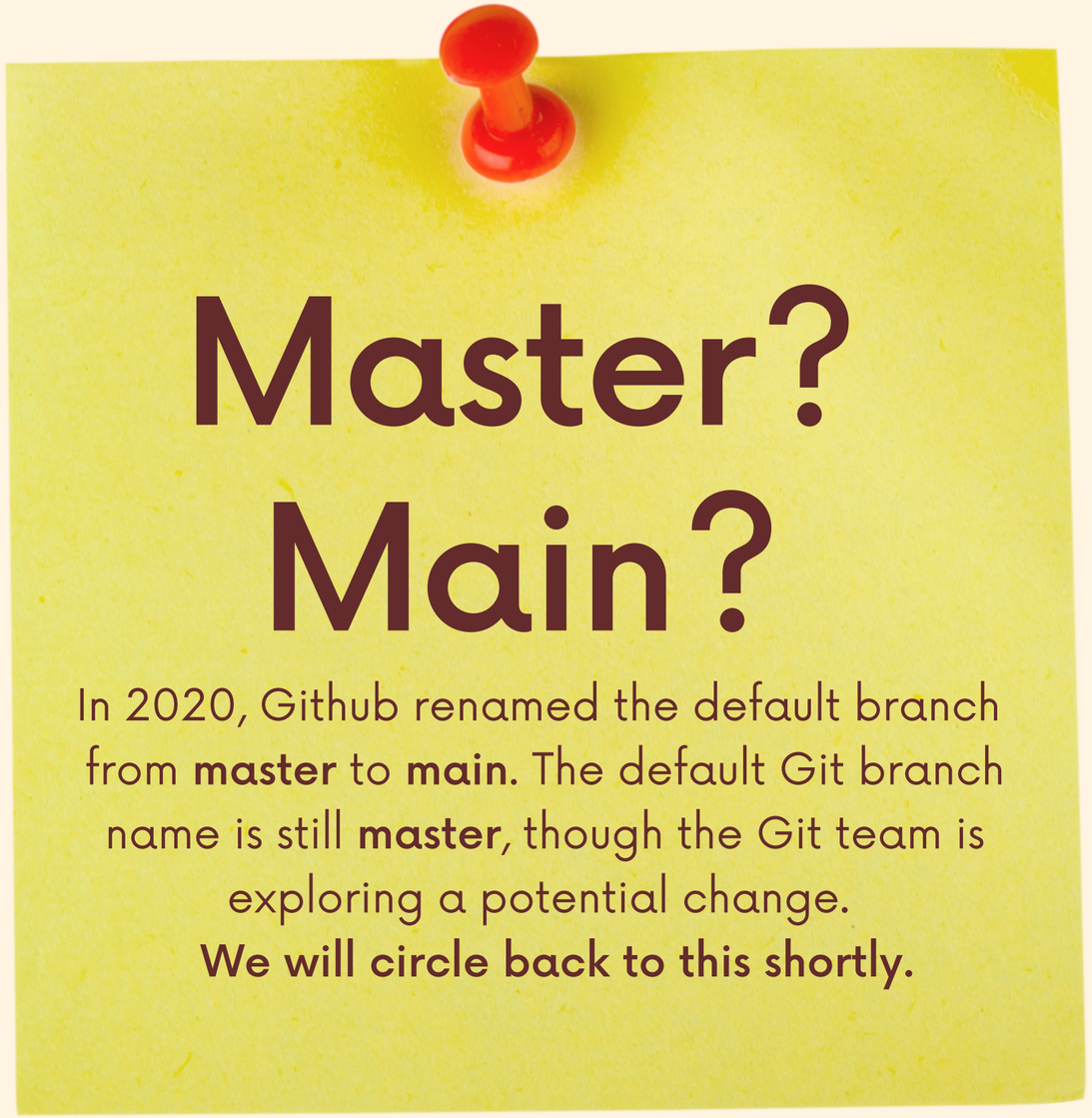
*technically that's not 100% true as we'll see later



Master

Many people designate the master branch as their "source of truth" or the "official branch" for their codebase, but that is left to you to decide.

From Git's perspective, the master branch is just like any other branch. It does not have to hold the "master copy" of your project.


Master?
Main?
In 2020, Github renamed the default branch from **master** to **main.** The default Git branch name is still **master**, though the Git team is exploring a potential change.
We will circle back to this shortly.

Branching

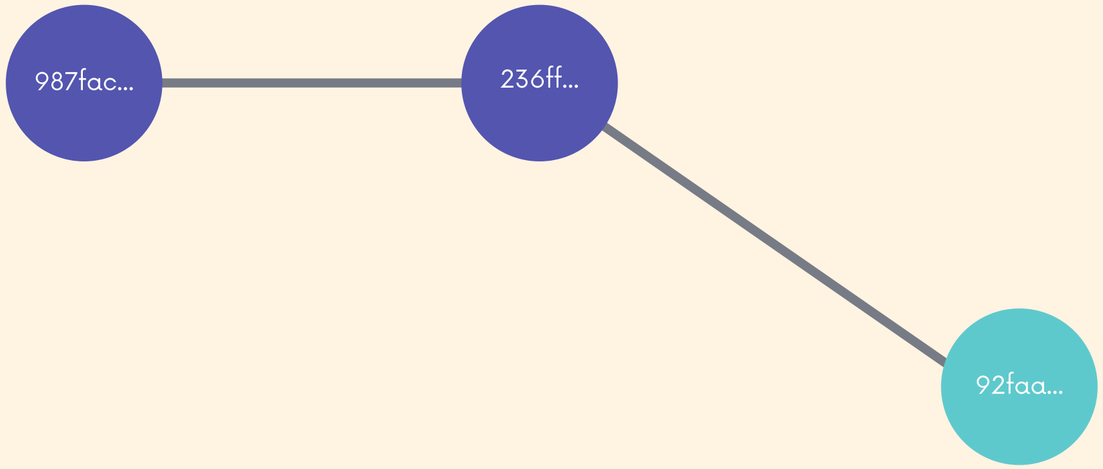
Master Branch

Branching

Master Branch

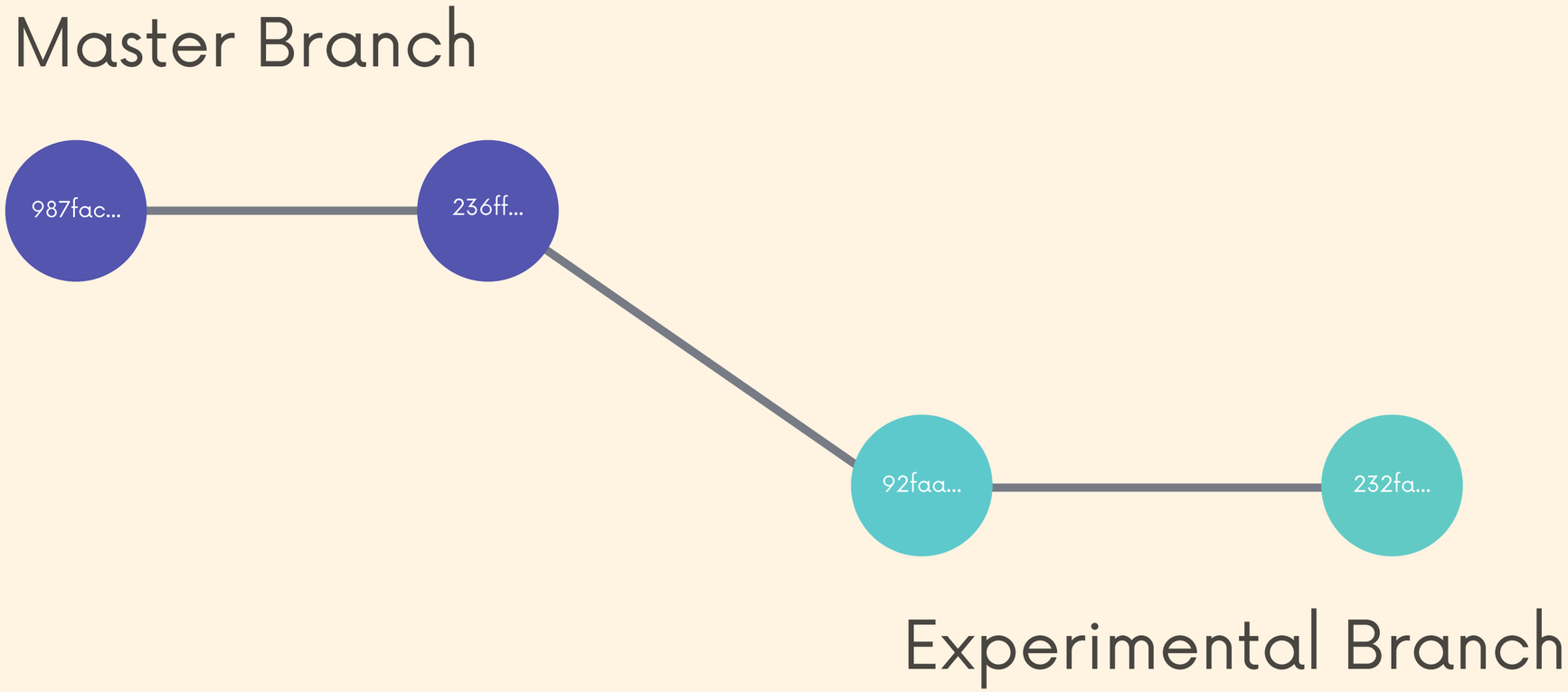
Branching

Master Branch

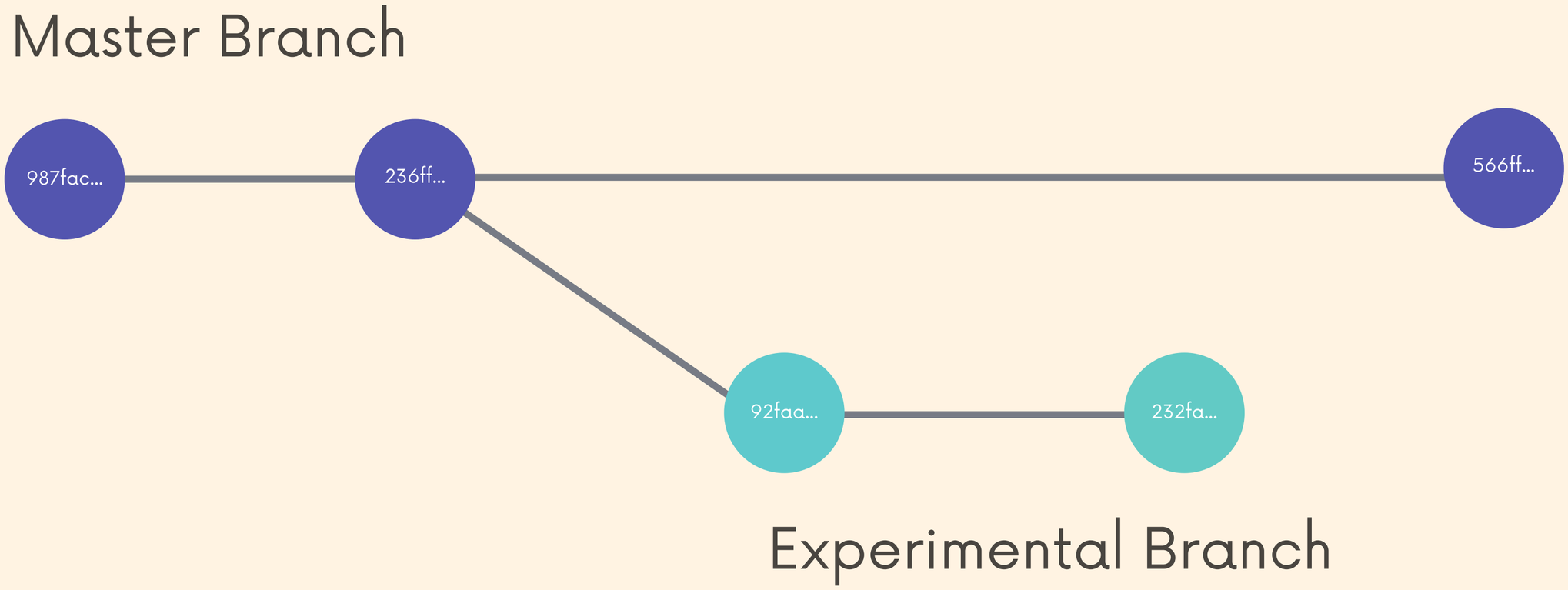


Experimental Branch

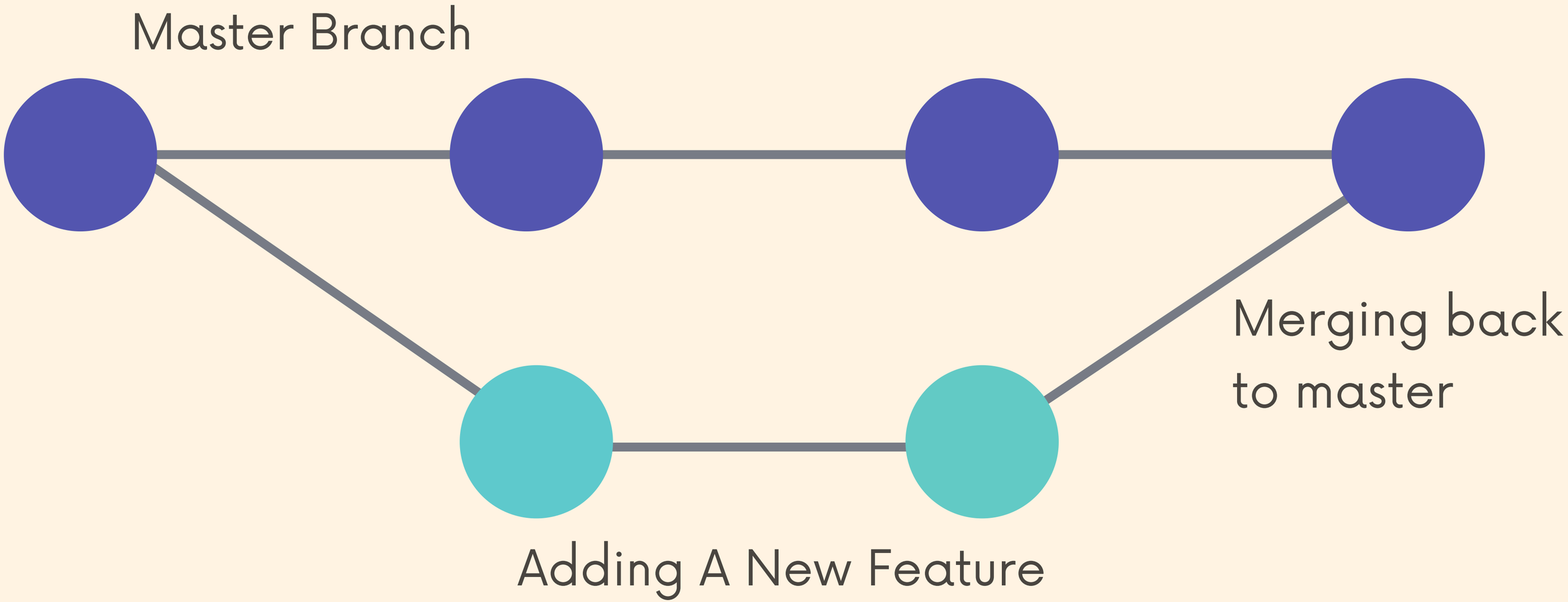
Branching

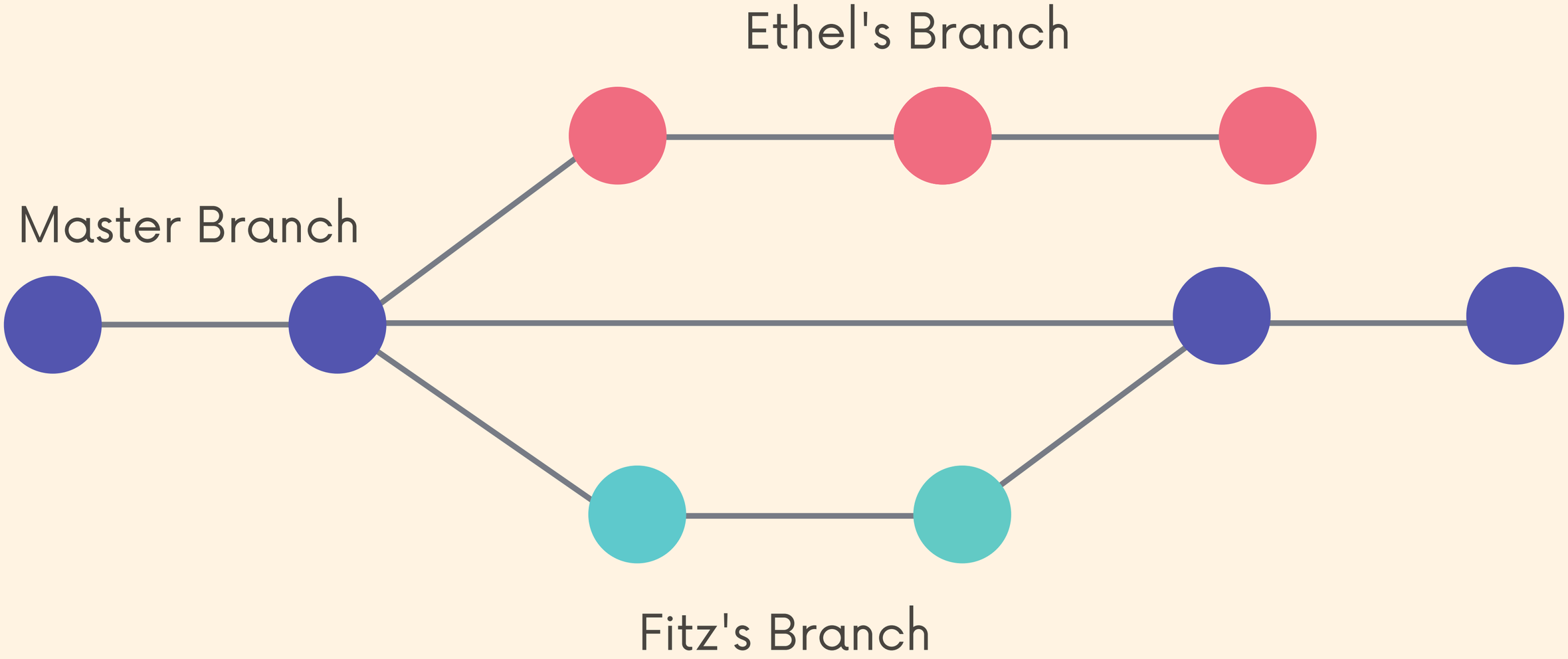

Master Branch
987fac...
236ff...
92faa...
232fa...
Experimental Branch

Branching


Master Branch
987fac...
236ff...
566ff...
92faa...
232fa...
Experimental Branch

A Common Workflow




Ethel's Branch
Master Branch
Fitz's Branch

commit bca5fc8b05deda4a13e7588c7ca352e47560c1dd
(HEAD -> master)

WTF IS THAT CRAZINESS

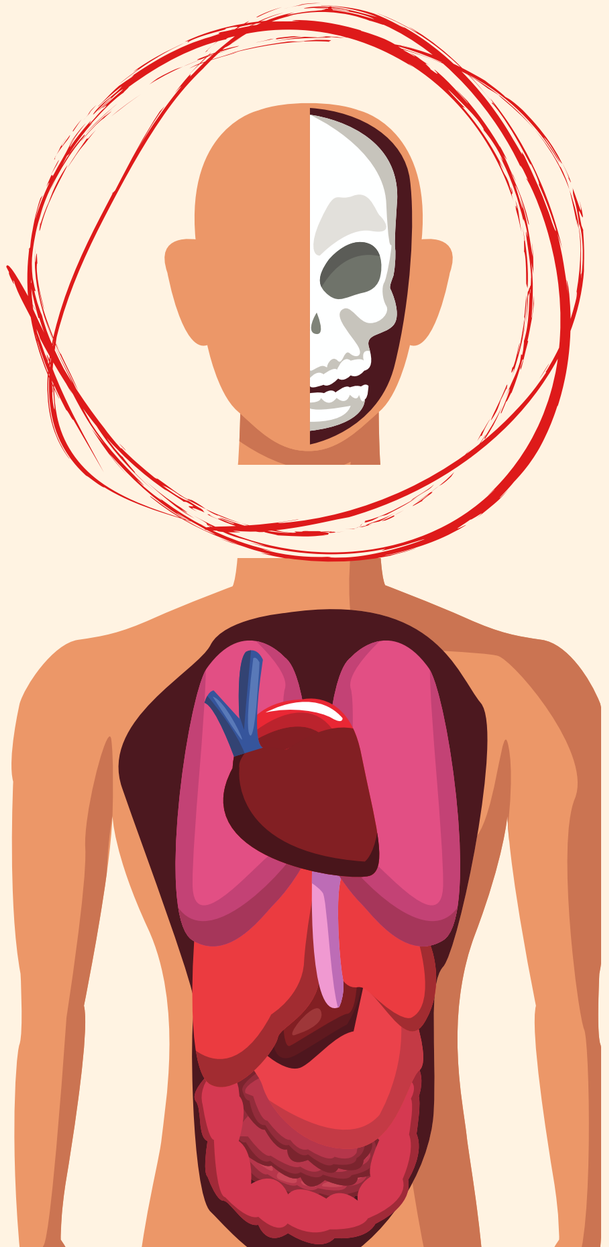
HEAD? What Is That?

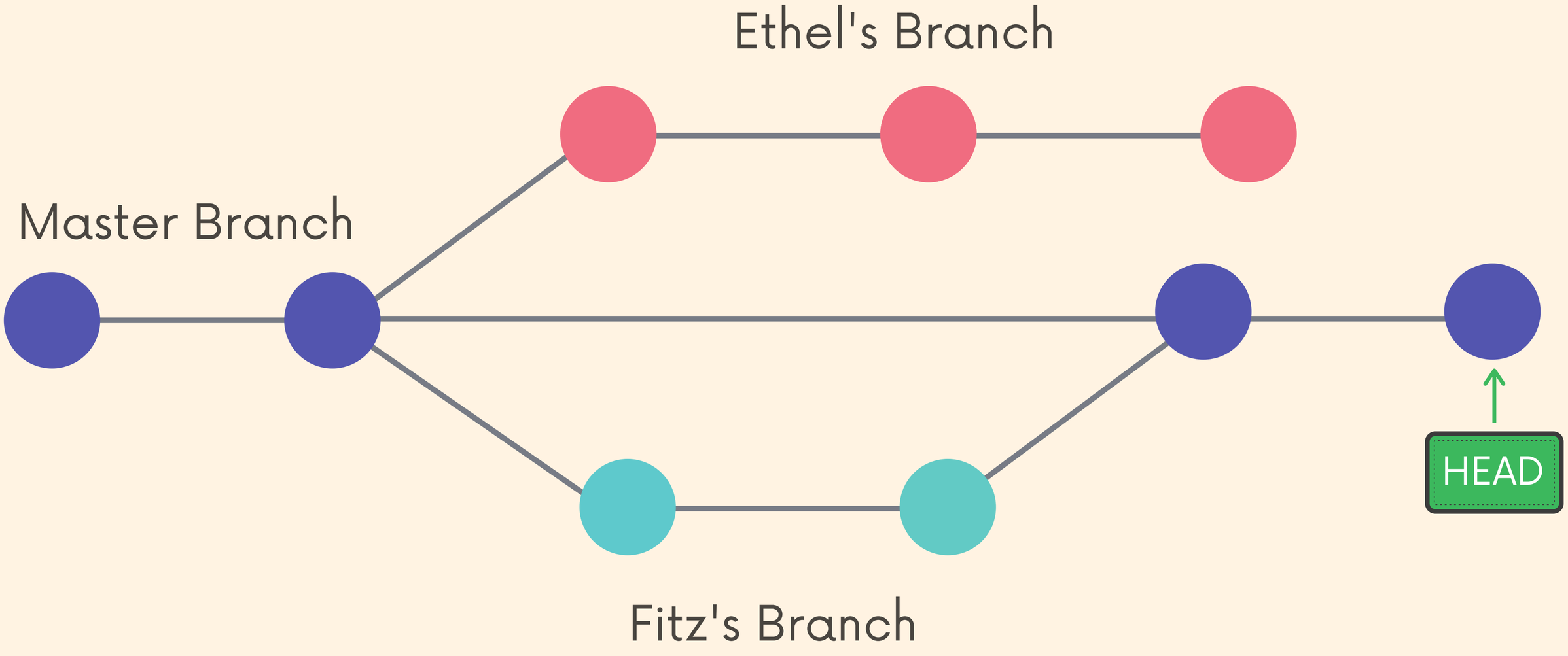
HEAD

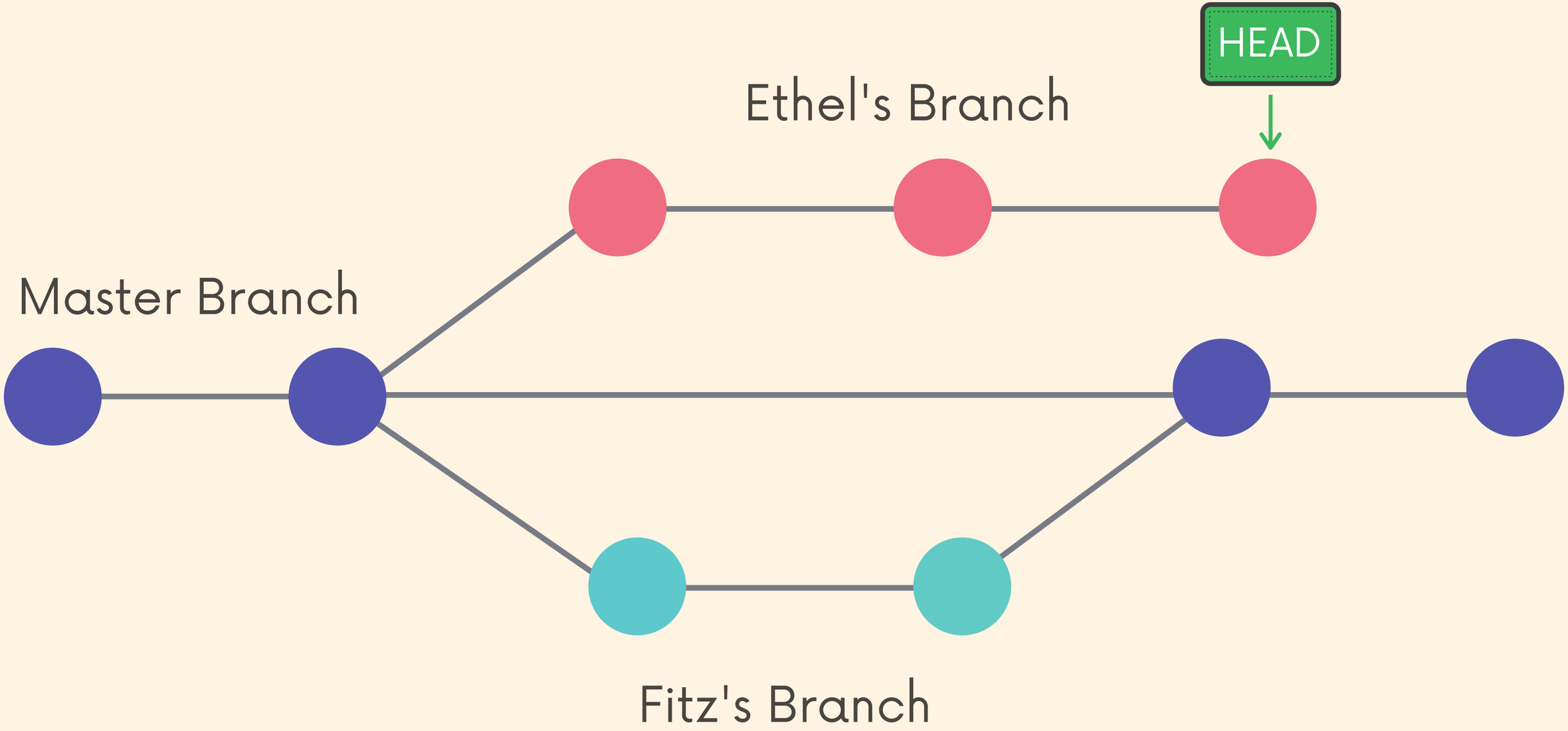
We'll often come across the term **HEAD** in Git.

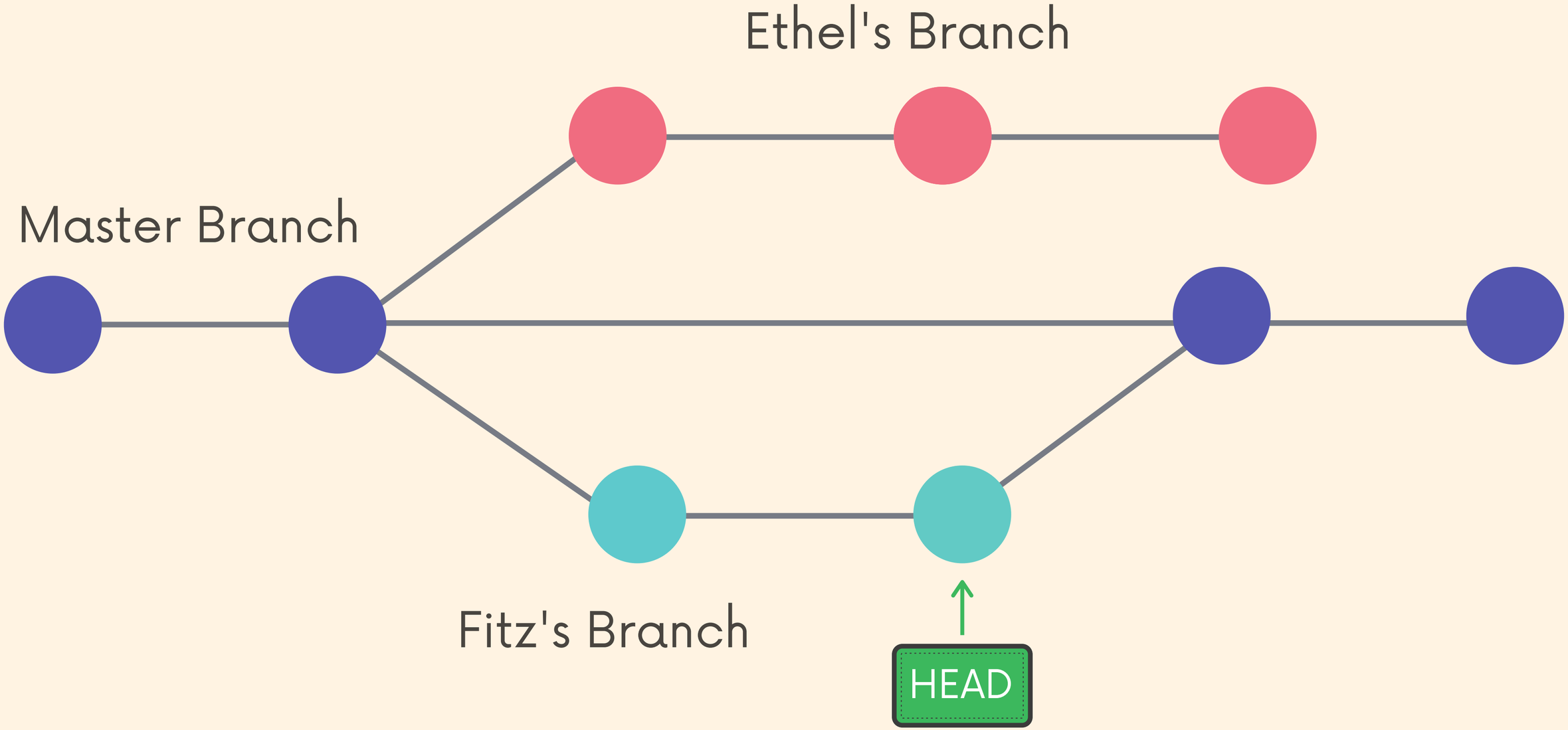
HEAD is simply a pointer that refers to the current "location" in your repository. It points to a particular branch reference.

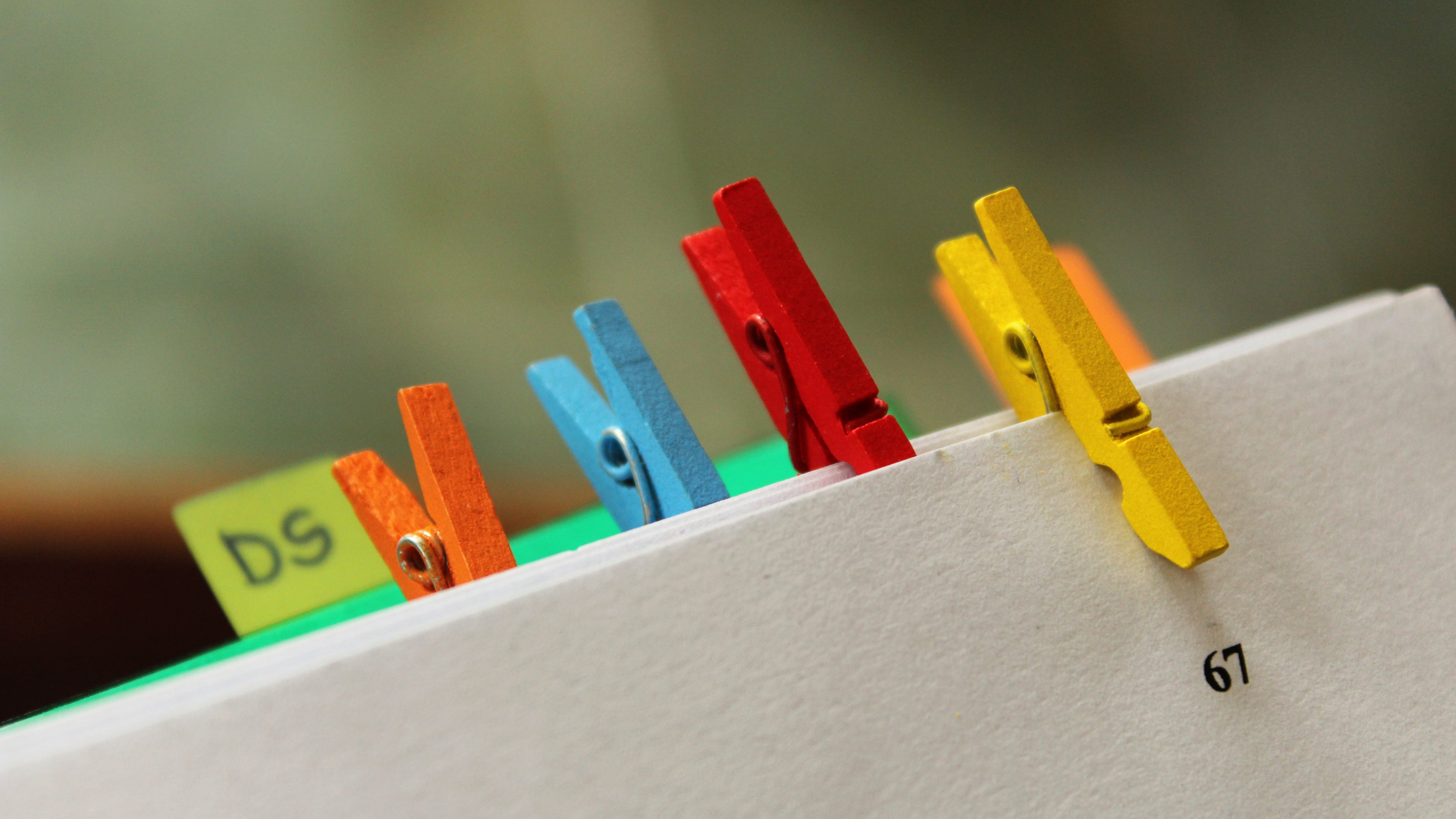
So far, HEAD always points to the latest commit you made on the master branch, but soon we'll see that we can move around and HEAD will change!




Ethel's Branch
Master Branch
HEAD
Fitz's Branch


HEAD
Ethel's Branch
Master Branch
Fitz's Branch


Ethel's Branch
Master Branch
Fitz's Branch
HEAD


DS
67

king and spitting.
Sovereign Lord helps me,
be disgraced.
ave I set my face like flint,
ow I will not be put to
me.
dicates me is near.
n will bring charges against

ce each other!
accuser?
confront me!
vereign Lord who helps me.
e that will condemn me?
ll wear out like a garment;
s will eat them up.

g you fears the Lord
s the word of his servant?
o walks in the dark,
no light,
name of the Lord
on his God.
ll you who light fires
ide yourselves with flaming
ches,
the light of your fires
he torches you have set

The islands will look to me
and wait in hope for my arm
6Lift up your eyes to the heavens
look at the earth beneath;
the heavens will vanish like sm
the earth will wear out like a
garment
and its inhabitants die like fl
But my salvation will last forev
my righteousness will never

7"Hear me, you who know wha
you people who have my law
hearts:
Do not fear the reproach of me
or be terrified by their insul
8For the moth will eat them up
garment;
the worm will devour them
wool.
But my righteousness will last
my salvation through all ger

9Awake, awake! Clothe yoursel
strength,
O arm of the Lord;
awake, as in days gone by,
as in generations of old.
Was it not you who cut Rahab to piece
who pierced that monster thr

the Lord Almighty is his name.
16I have put my words in your mouth
and covered you with the shadow of
my hand—
I who set the heavens in place,
who laid the foundations of the earth,
and who say to Zion, 'You are my
people.' "

The Cup of the Lord's Wrath

17Awake, awake!
Rise up, O Jerusalem,
you who have drunk from the hand of
the Lord
the cup of his wrath,
you who have drained to its dregs
the goblet that makes men stagger.
18Of all the sons she bore
there was none to guide her;
of all the sons she reared
there was none to take her by the
hand.
19These double calamities have come
upon you—
who can comfort you?—
ruin and destruction, famine and
sword—
who can[a] console you?
your sons have fainted;
lie at the head of every street,

Free yourself from the
neck,
O captive Daughte

3For this is what the

"You were sold for n
and without mone
redeemed."

4For this is what th

"At first my people
to live;
lately, Assyria h

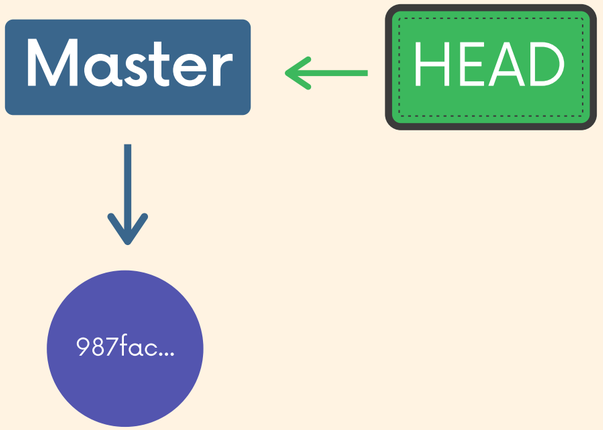
5"And now what
clares the Lord.

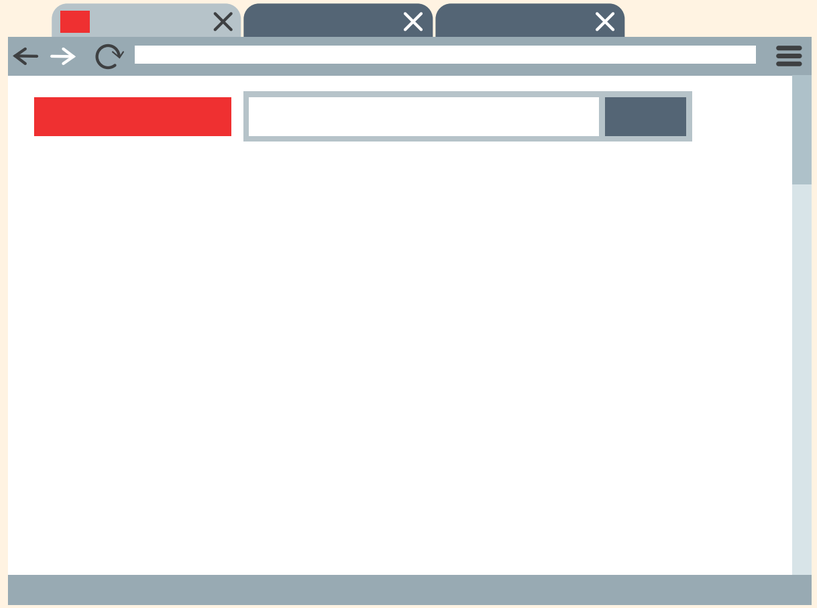
"For my people h
for nothi
and those who

"And all day lon
my name is c
6Therefore my pe
name;
therefore in
that it is I who
Yes, it is I."

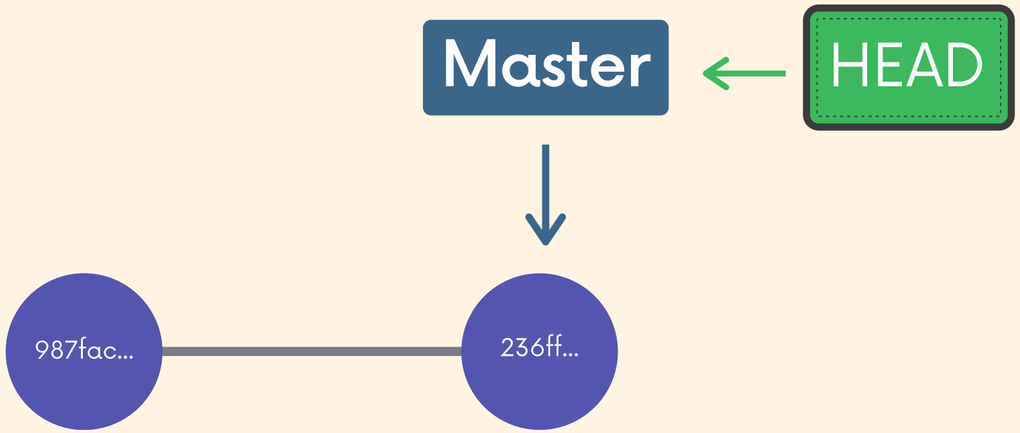
7How beautiful
are the feet

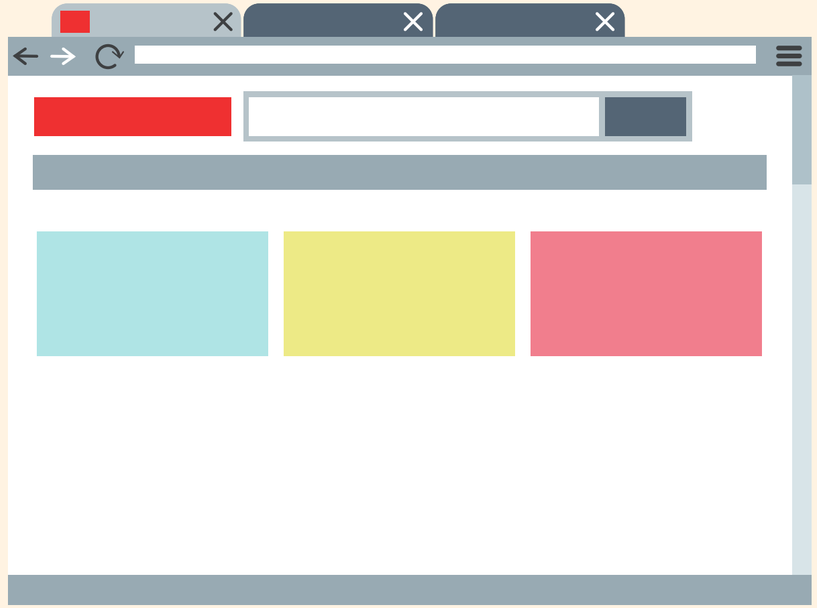
My First Commit On Master



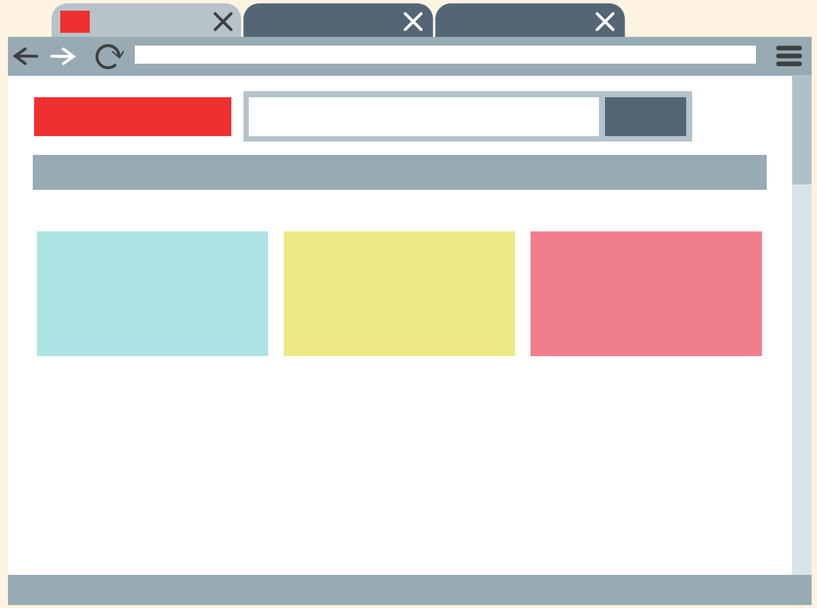


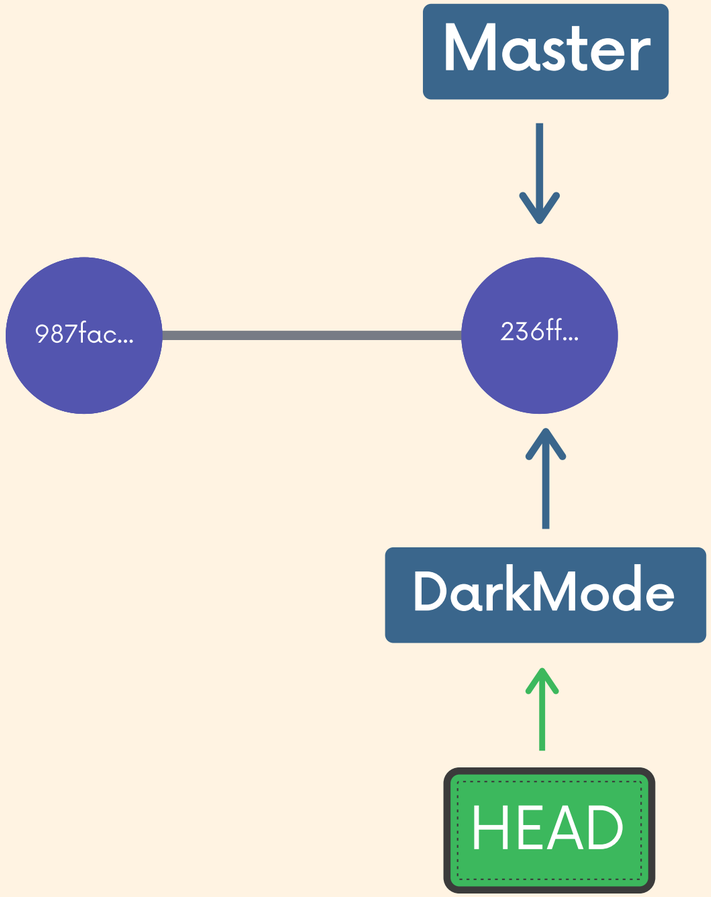
Another Commit



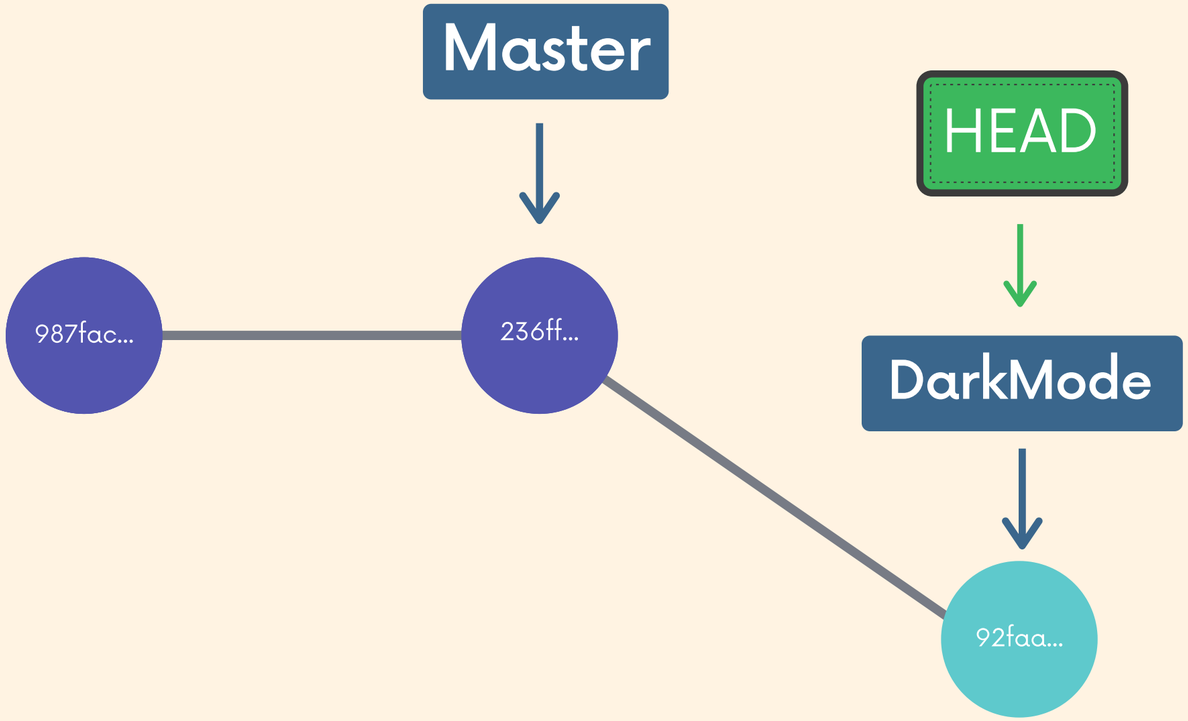


I make a new branch!

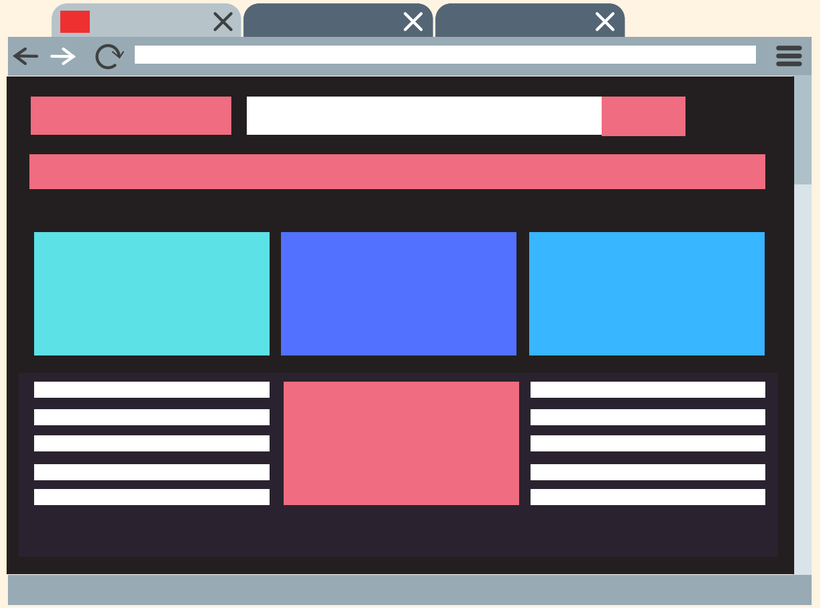


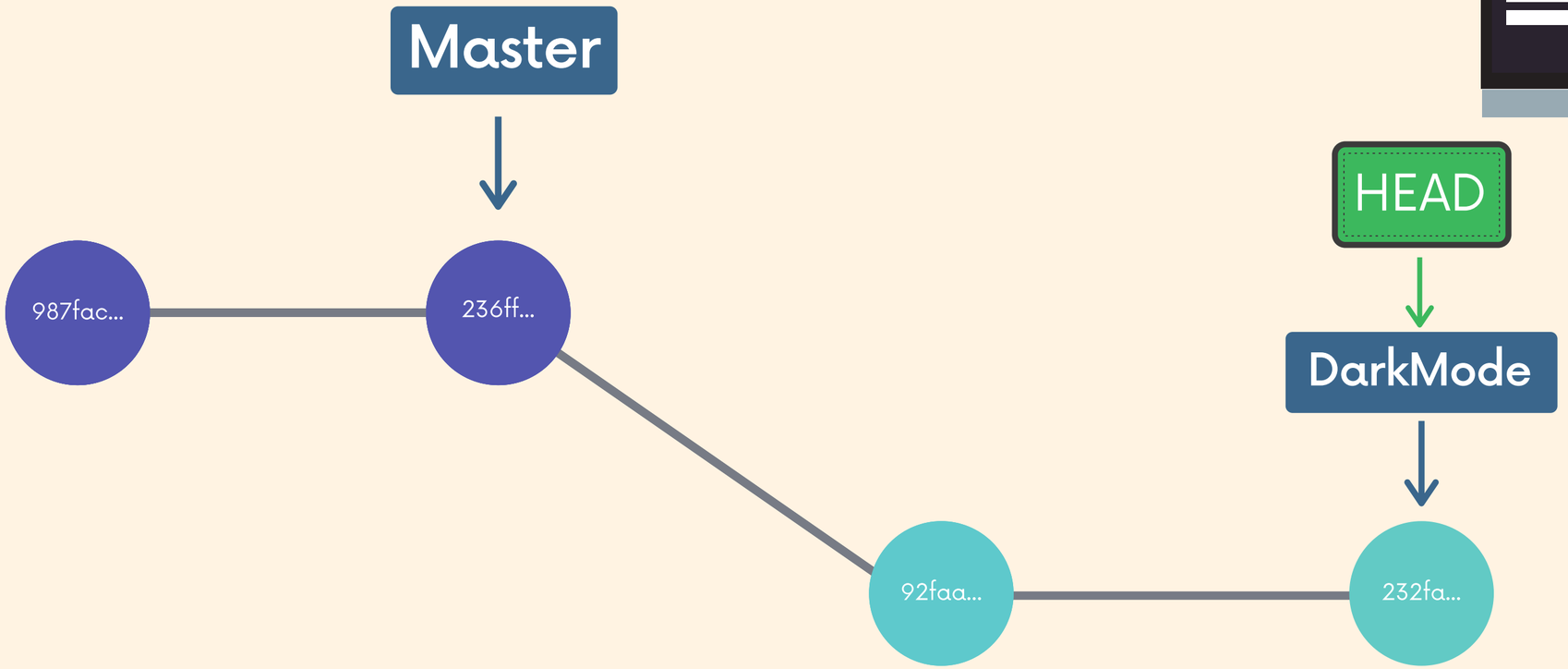


I make a commit on the new branch

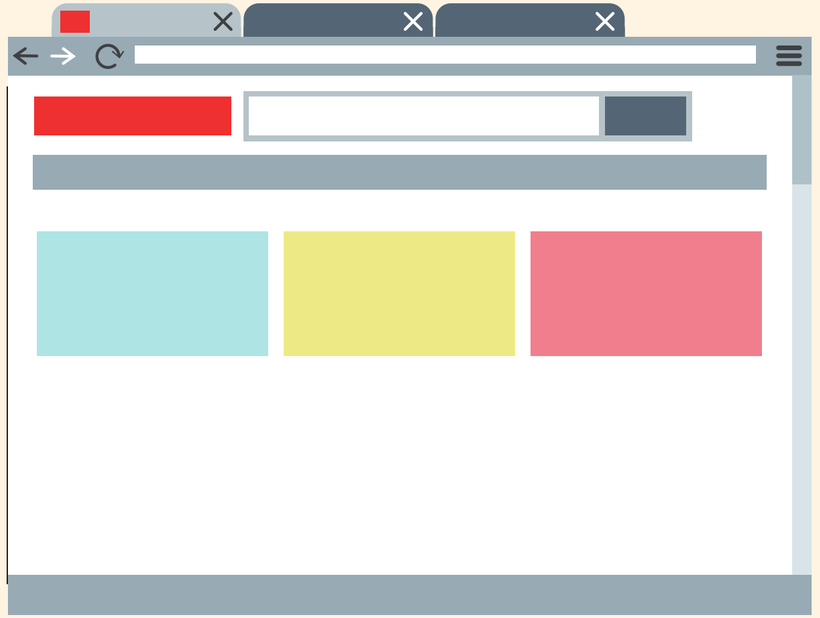


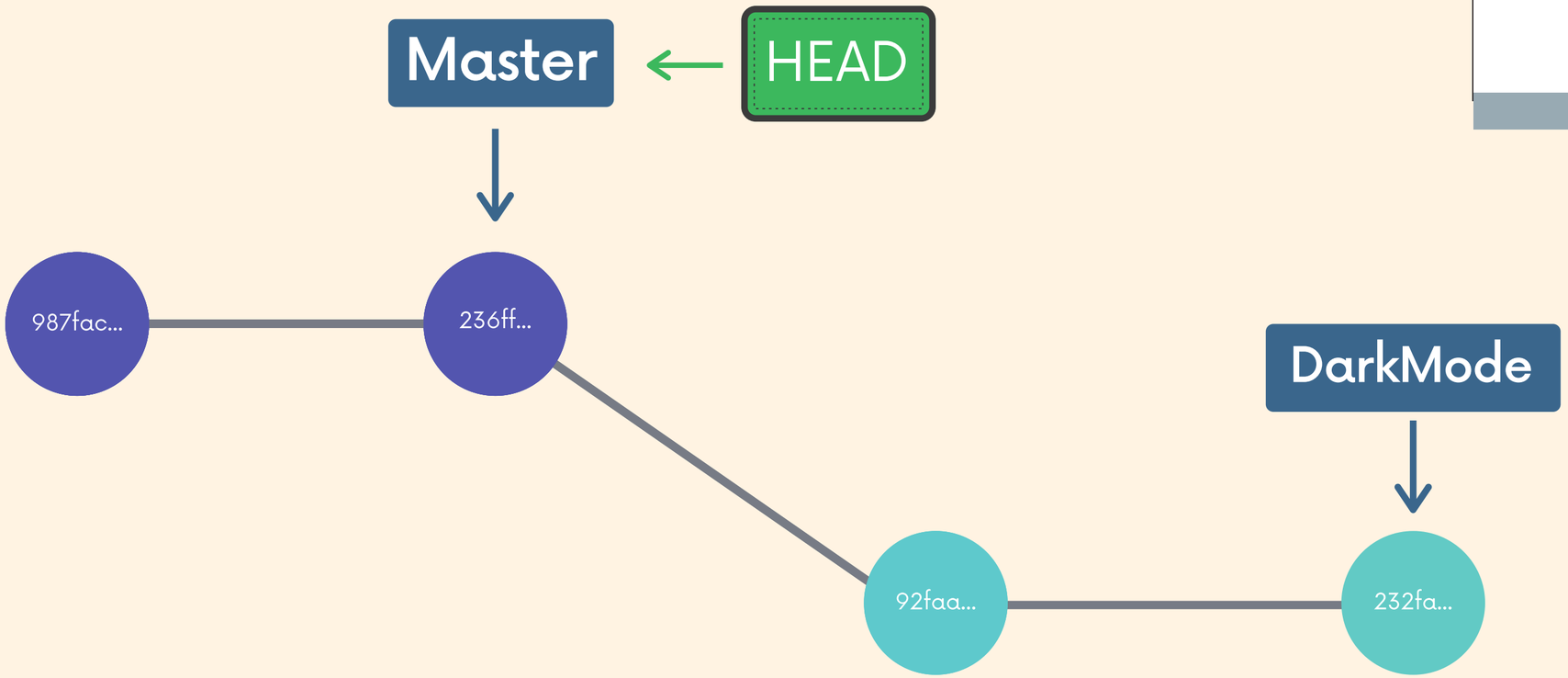
2nd Commit
On New Branch



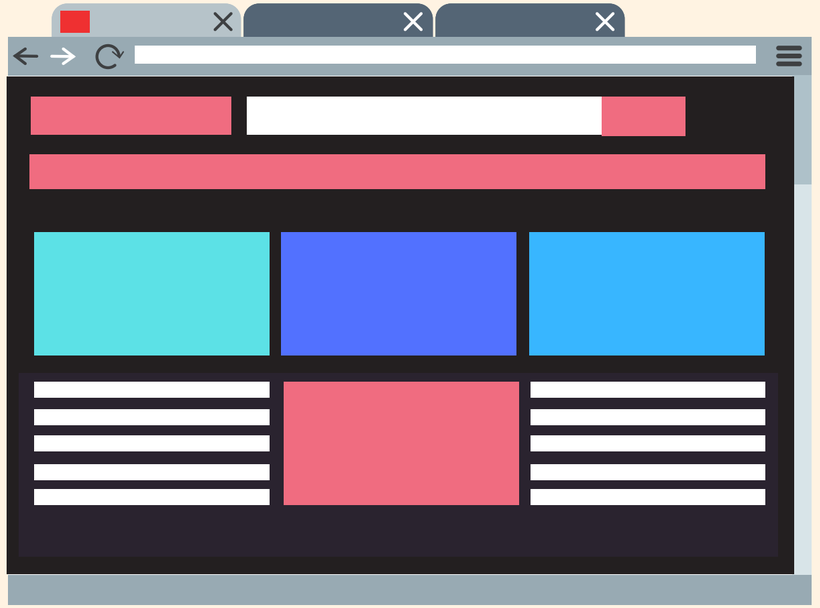


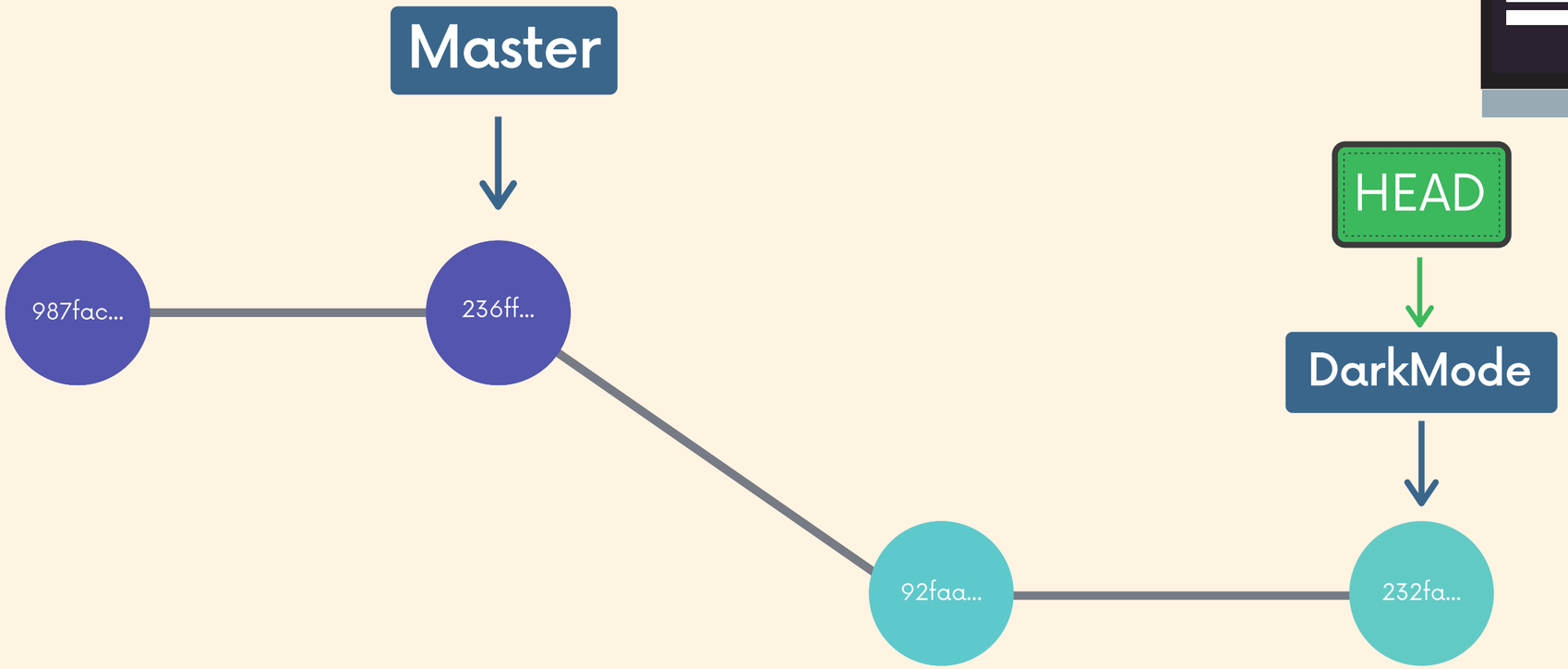
Return to Master



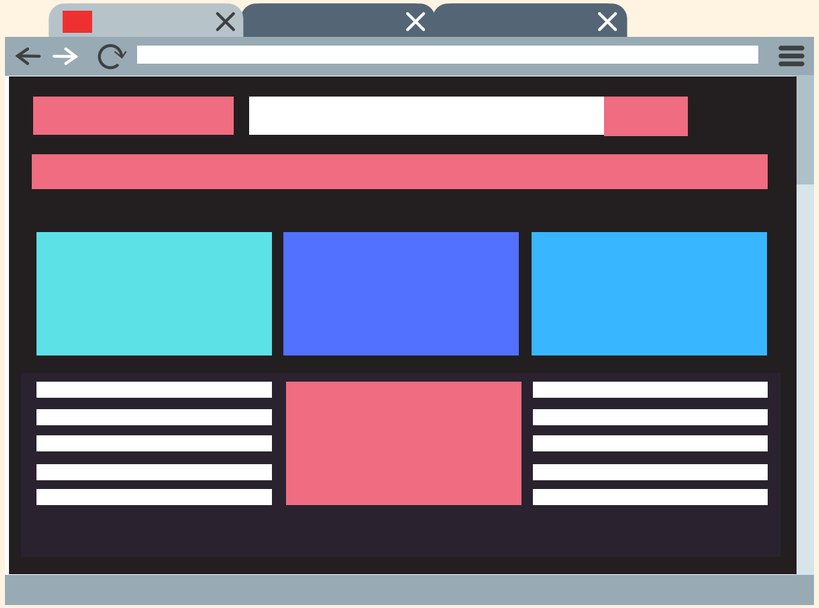


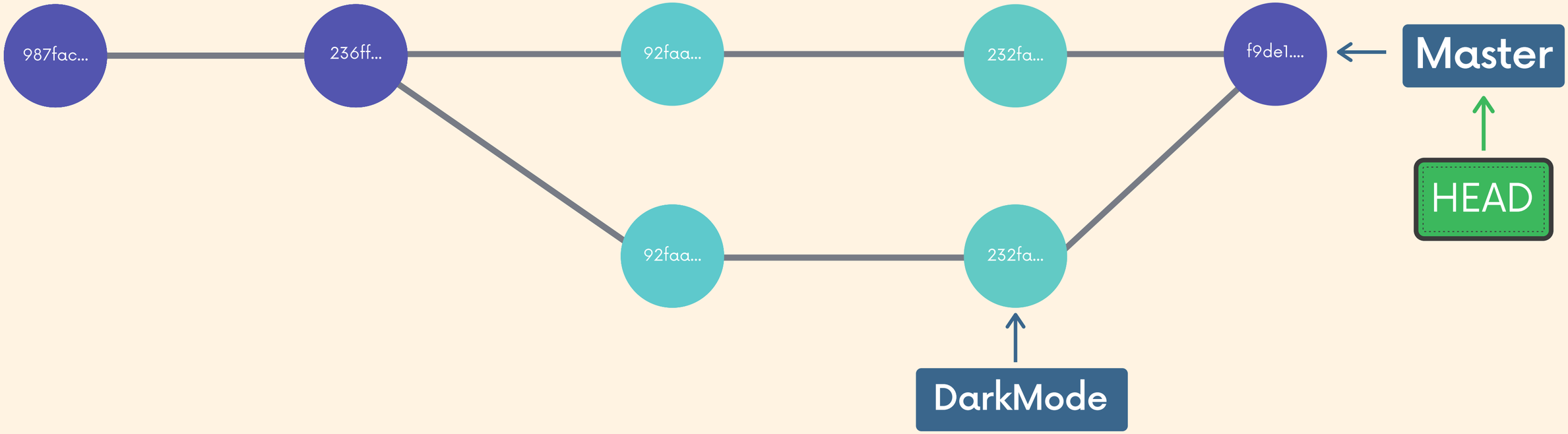
Going Back To My Dark Mode Branch





Merge Changes Into Master Branch?





Viewing Branches

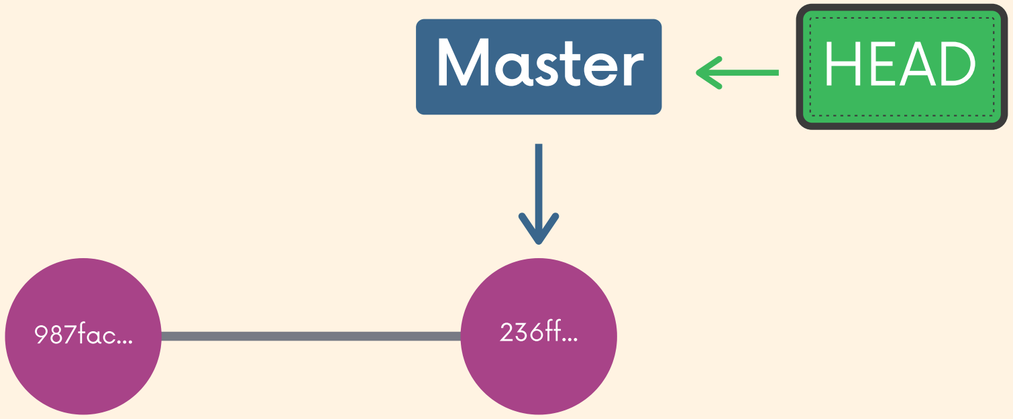
Use **git branch** to view your existing branches. The default branch in every git repo is master, though you can configure this.

Look for the * which indicates the branch you are currently on.

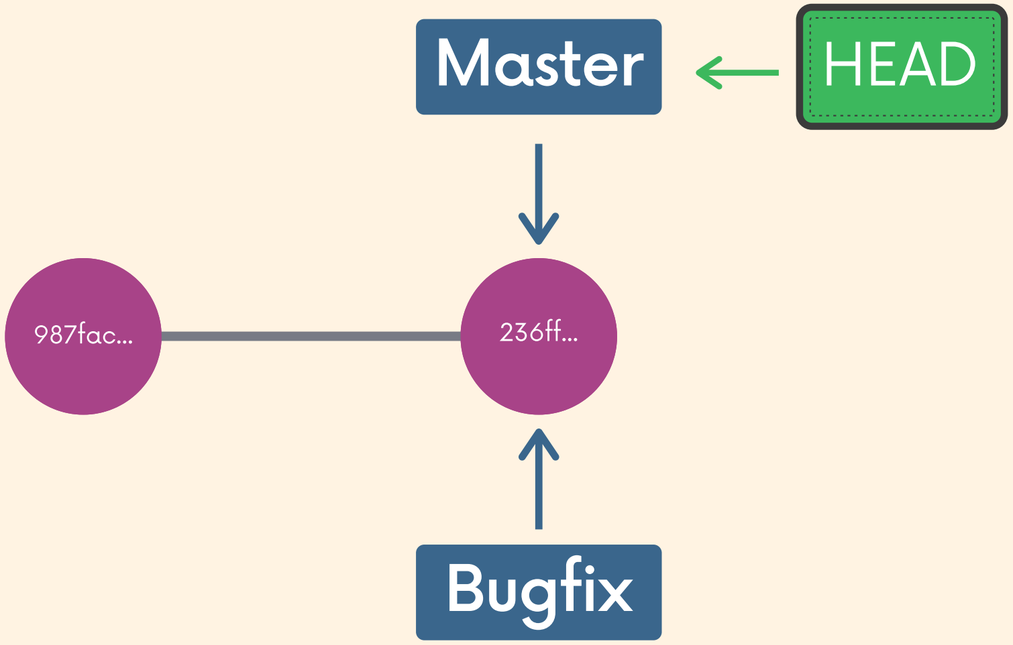
Creating Branches

Use **git branch <branch-name>** to make a new branch based upon the current HEAD

This just creates the branch. It does not switch you to that branch (the HEAD stays the same)


Master
HEAD
987fac...
236ff...

```
❯ git branch bugfix
```

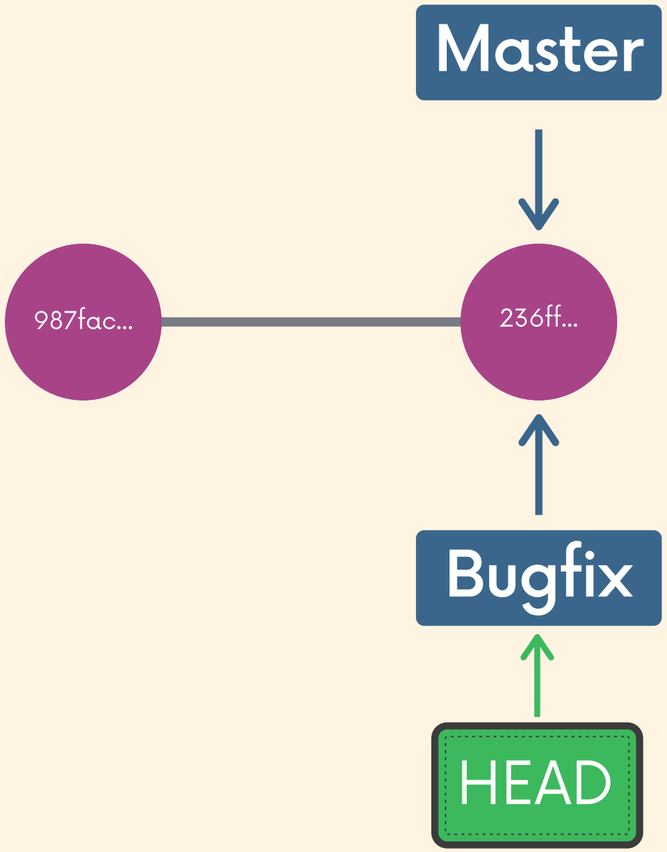


I've made a new branch,
but I'm still working on master.

Switching Branches

Once you have created a new branch,
use **git switch <branch-name>** to switch to it.

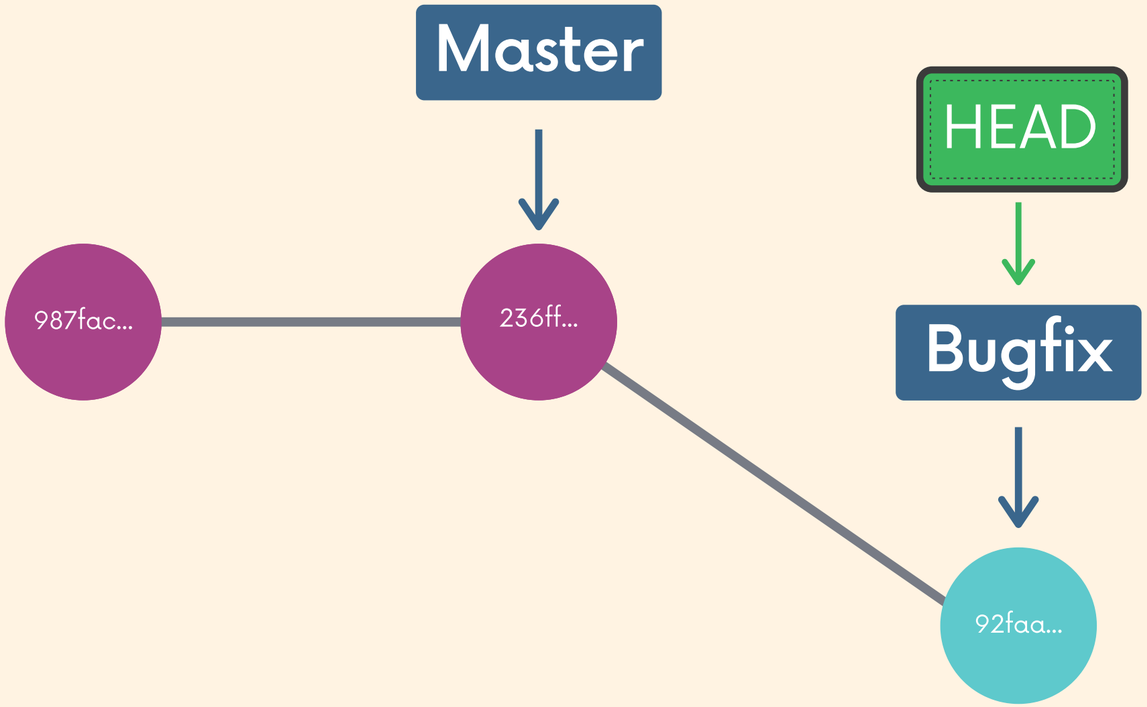
```
❯ git switch bugfix
```



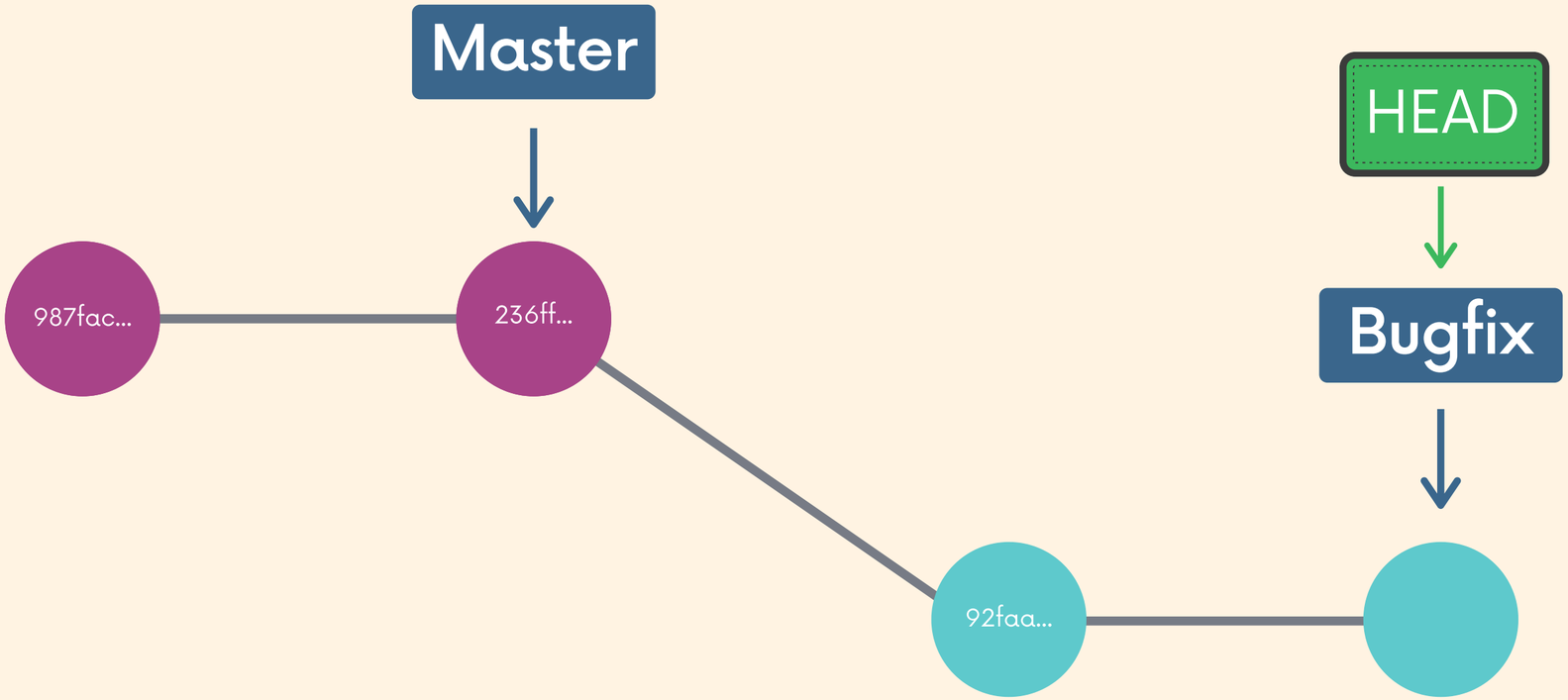
I've now switched over to the new bugfix branch!

Notice that HEAD is now pointing to Bugfix, not master.

When I make a new commit, it exists only on my new branch, not on master!

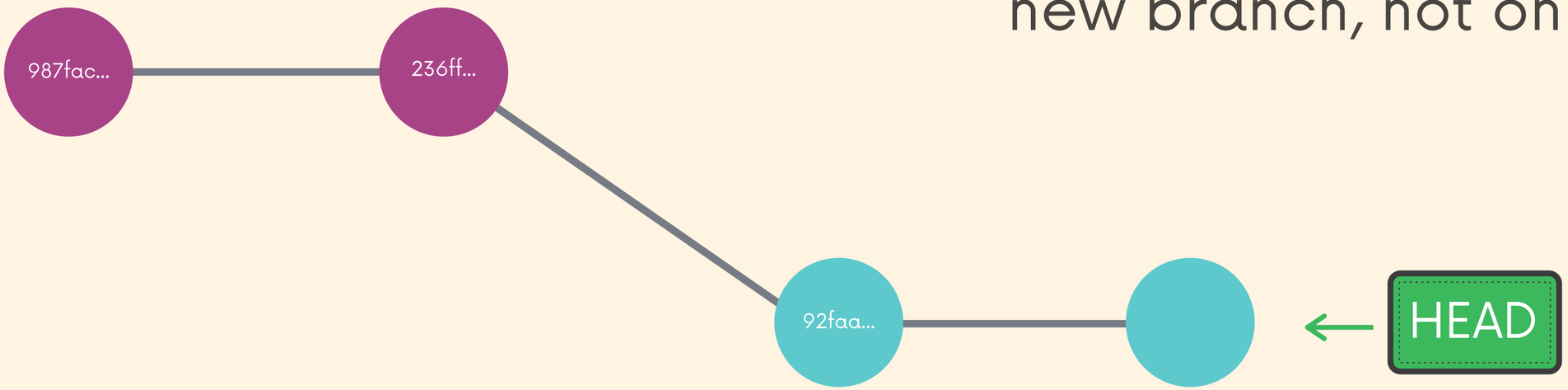


And another commit...



```
❯ git switch bugfix
```

master branch



bugfix branch

The commit exists only on my new branch, not on master!

Another way of switching??

Historically, we used git **checkout <branch-name>** to switch branches. This still works.

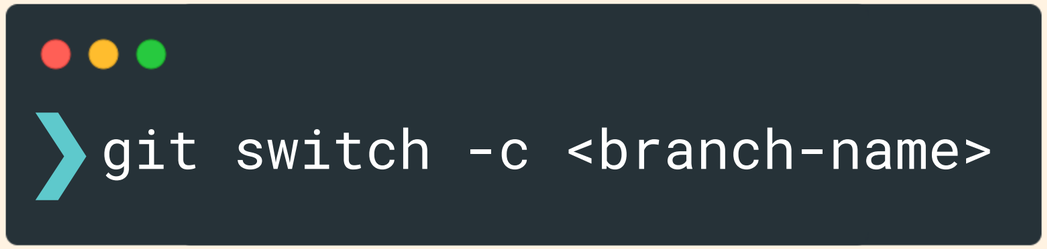
The **checkout** command does a million additional things, so the decision was made to add a standalone switch command which is much simpler.

You will see older tutorials and docs using checkout rather than switch. Both now work.

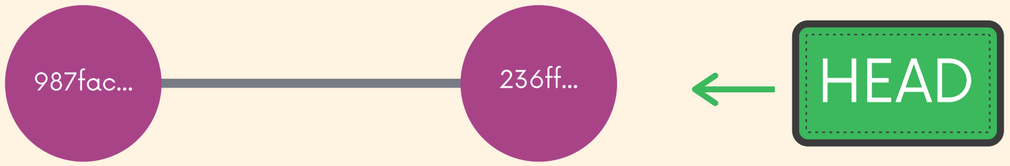
Creating & Switching

Use **git switch** with the **-c** flag to create a new branch AND switch to it all in one go.

Remember **-c** as short for "create"

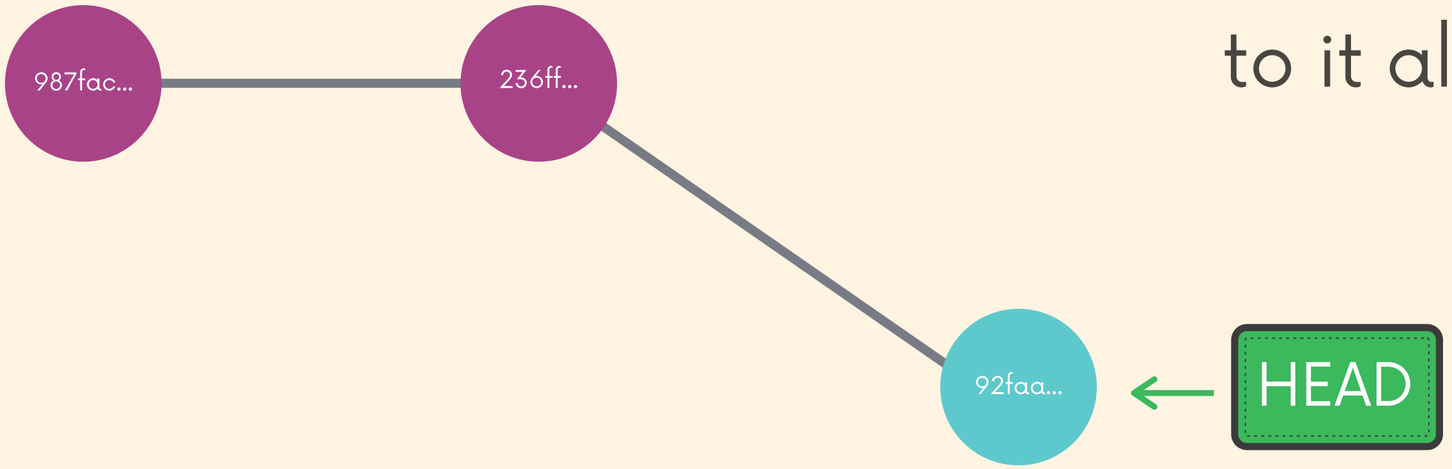


master branch



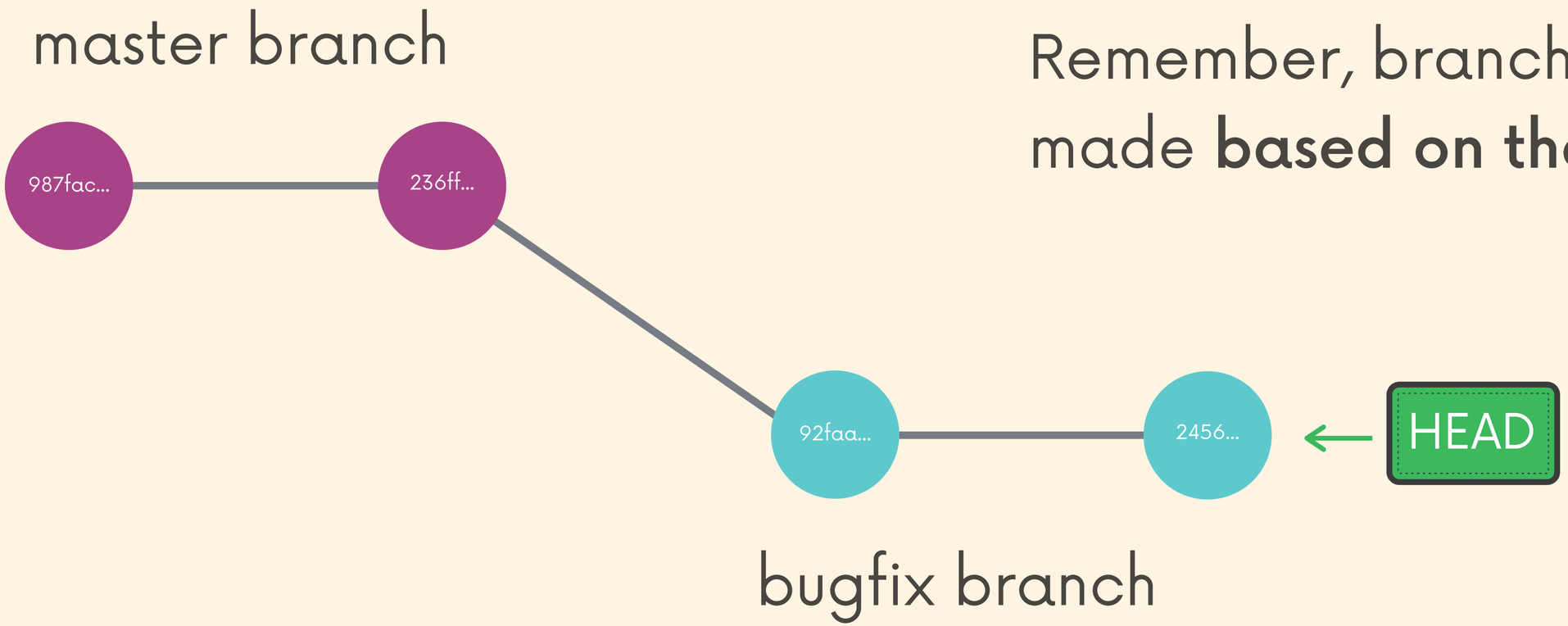
```
❯ git switch -c refactor
```

master branch

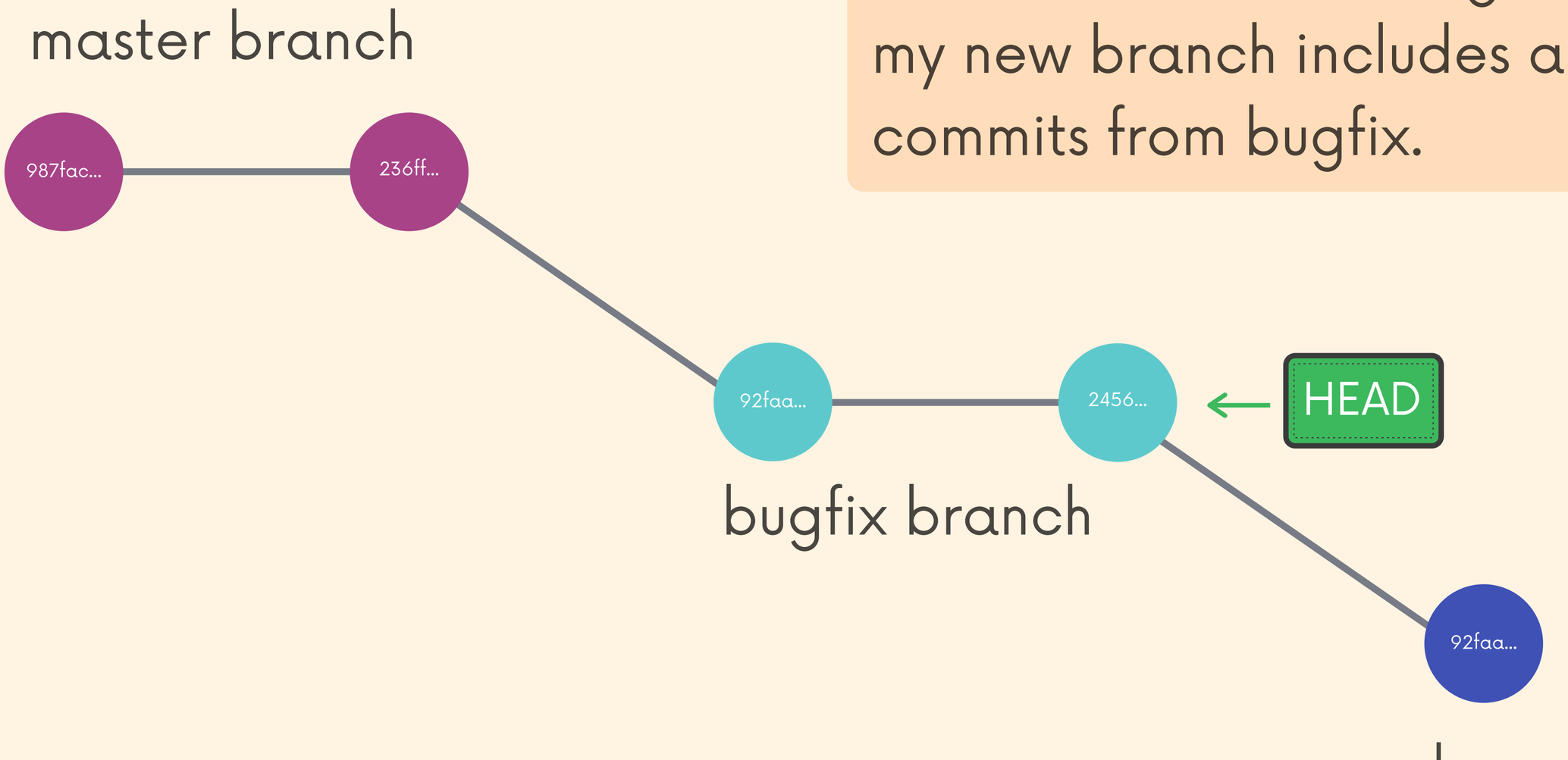


refactor branch

the **-c** flag makes the new refactor branch and switches to it all at once!


master branch
Remember, branches are made based on the HEAD
987fac...
236ff...
92faa...
2456...
HEAD
bugfix branch

If I branch from the bugfix branch, my new branch includes all the commits from bugfix.



Viewing More Info

Use the **-v** flag with **git branch** to view more information about each branch.

```
❯ git branch -v
```

```
* master      88df51f   add documentation
  bugfix      7c2a586   fix hover bug
  chatdemo    afadcf9   add chat widget
```