

# Git workshop III

TYH Lab git Workshop II  
by Paul Z Cheng

# Review

- git daily commands
- Onion model
- Flow with the log
- resolving merge conflict

# After install connecting to online platform

Before all the fun of `git clone`, `git fetch`, `git pull`, or `git push`

- ssh

<https://docs.github.com/en/github/authenticating-to-github/connecting-to-github-with-ssh>

- 2FA

<https://docs.github.com/en/github/authenticating-to-github/securing-your-account-with-two-factor-authentication-2fa>

- http

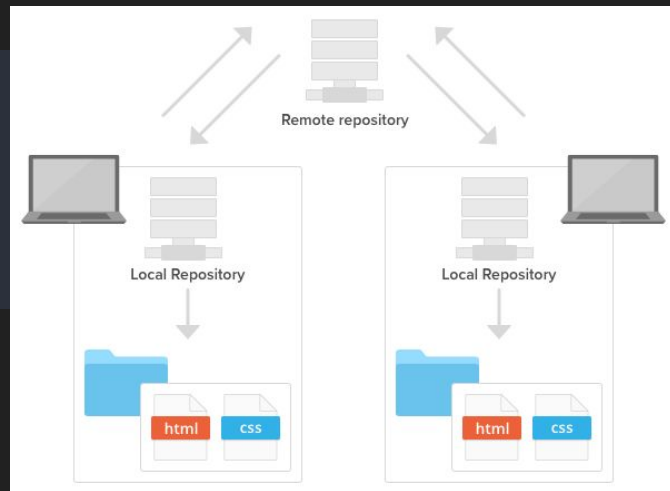
# configure your settings

```
git init
git config --global user.name "Paul Z. Cheng"
git config --global user.email
paul.z.cheng@gmail.com
git config --global core.editor atom
```



# set up your repository

```
## Clone a directory  
git clone < git URL >  
# ..... some awesome code session later and as above  
# git remote setup  
git remote add origin <Remote URL>
```



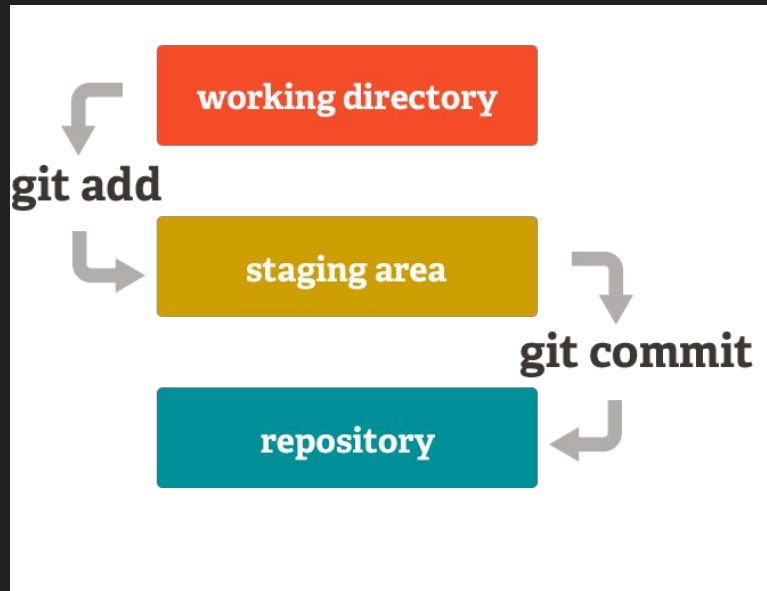
# Daily edited and saving

```
## some awesome coding and editing of your working area later
# check working area has any need to update
git status

# Stage files
git add "file names"
git add .

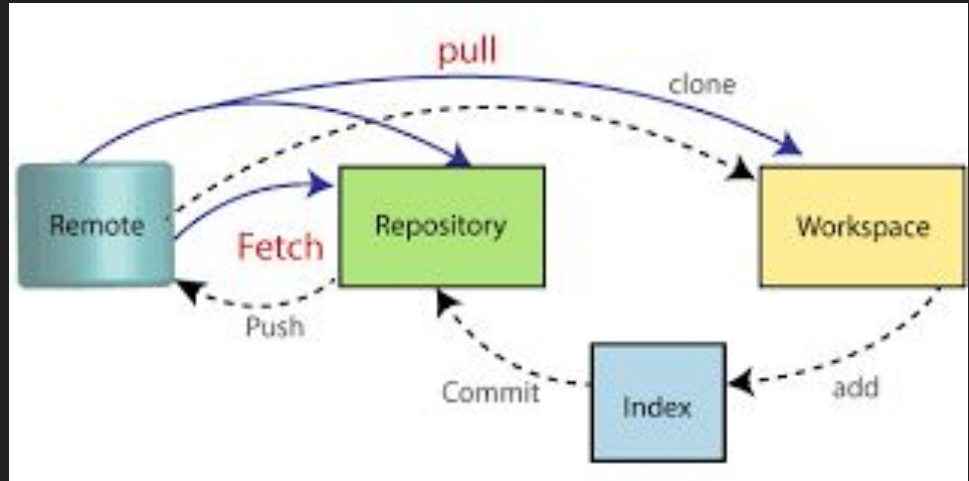
# everything that has been changed
# Commit related to "document your add"
git commit -m "xx files update"
# change message
git commit -amend -m "change"

# check out commits
git log
```



# Push and pull

```
# fetch method  
git fetch origin master  
# solve conflict or not  
git merge origin master  
  
# Pull method  
git pull origin master  
# update remote  
git push master origin
```

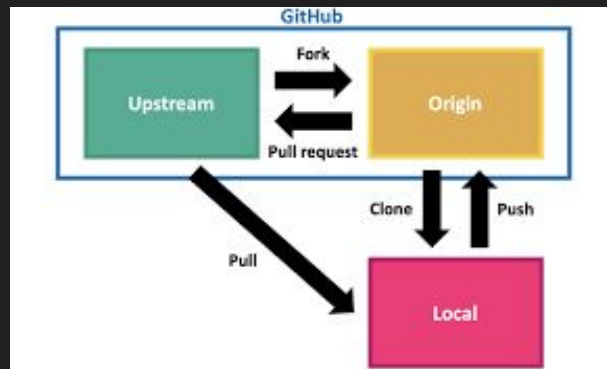


# Exercise to setup remote connection with master

```
# after you set everything up with your own fork
git remote add upstream https://github.com/paul30402/gitasktic.git

# solve merge conflict if you did something different to master
git fetch upstream
git diff upstream master
git merge upstream master

# push it to your own fork
git checkout master
git push master origin
```





# GIT IT?



146

# Onion





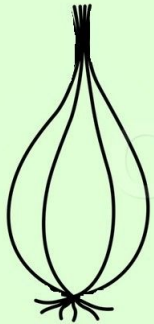
Git is a  
Persistent Map

OpenGenus

Banana bread

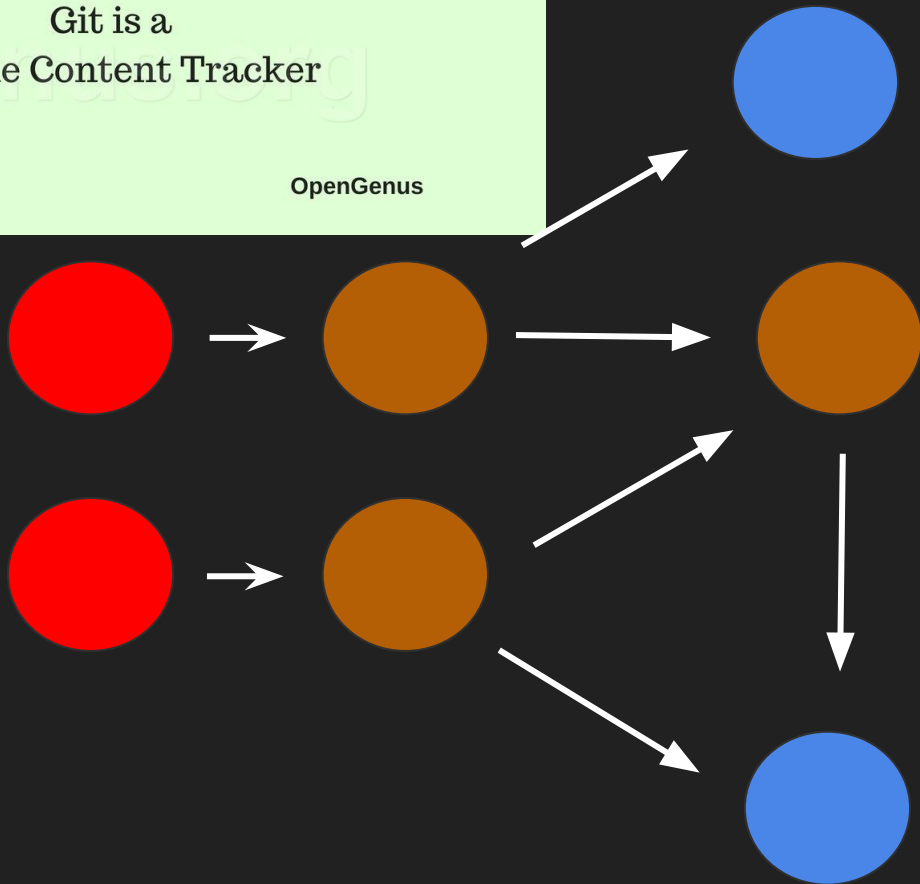


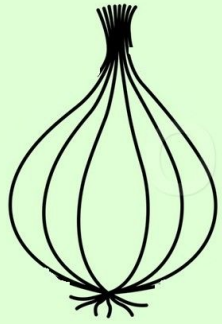
e884135b2103c673e876e850879539079033e670



# Git is a Simple Content Tracker

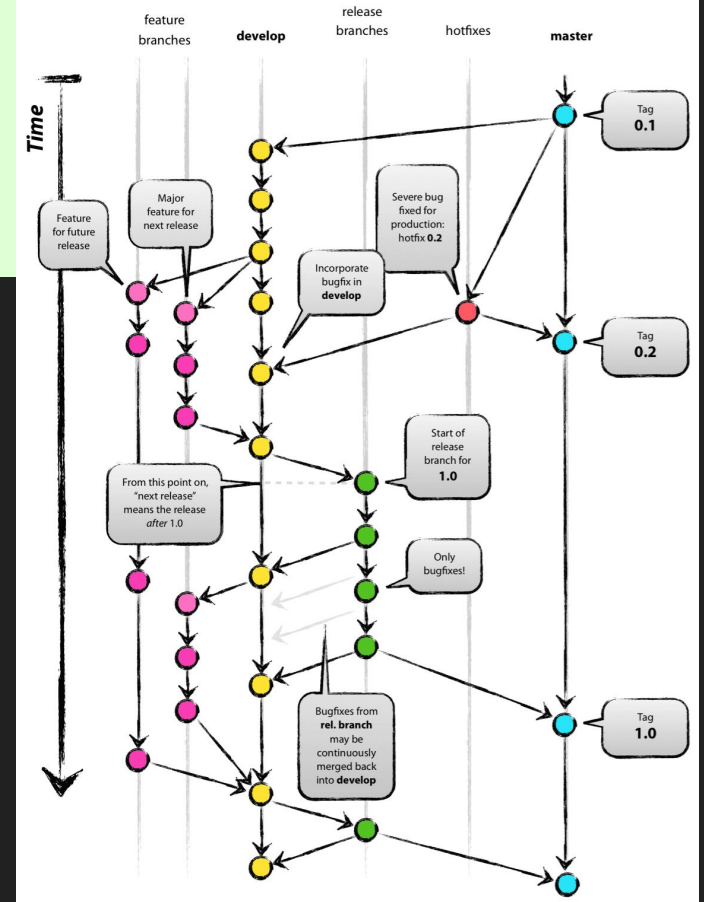
OpenGenus

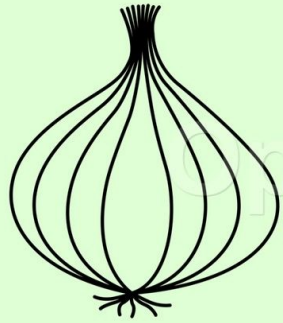




# Git is a Revision Control System

OpenGenus

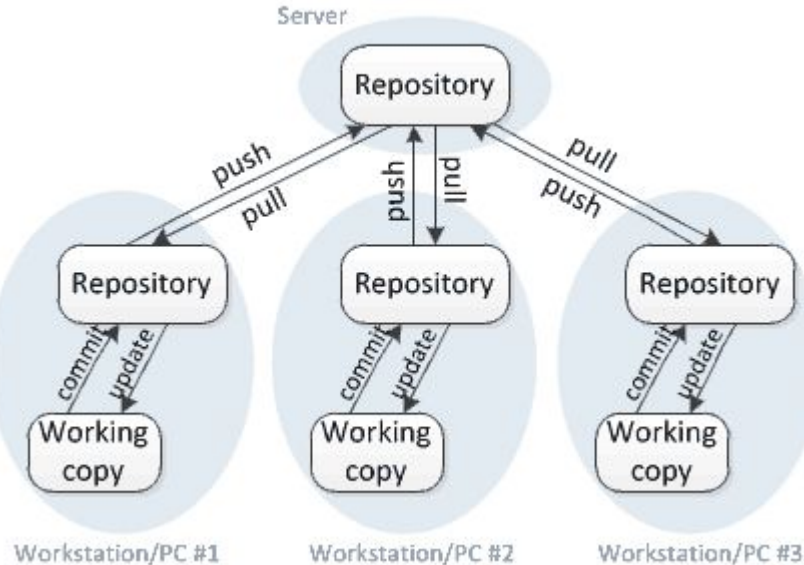




Git is a  
Distributed Revision Control System

OpenGenus

## Distributed version control



# Flow with log



# Review the blob blob blob blob

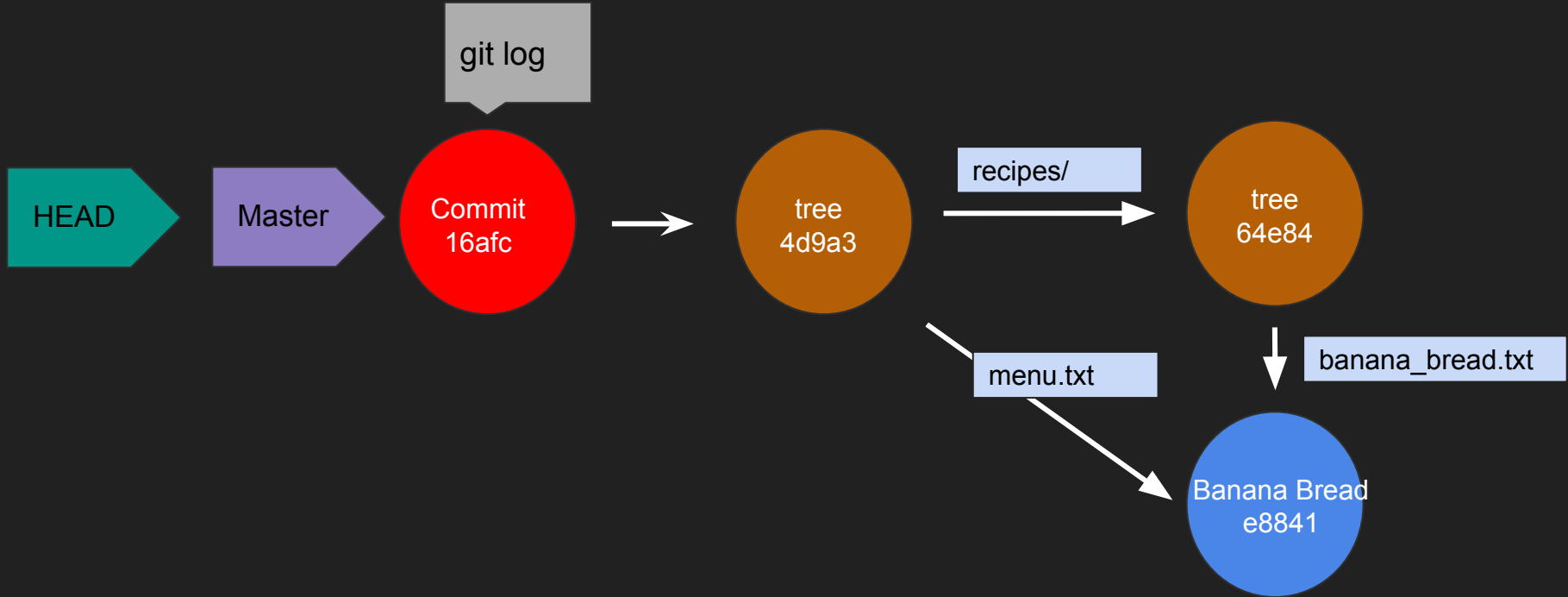


blob

e884135b2103c673e876e850879539079033e670



# git commit : Tree and blob



```
git init  
git log
```

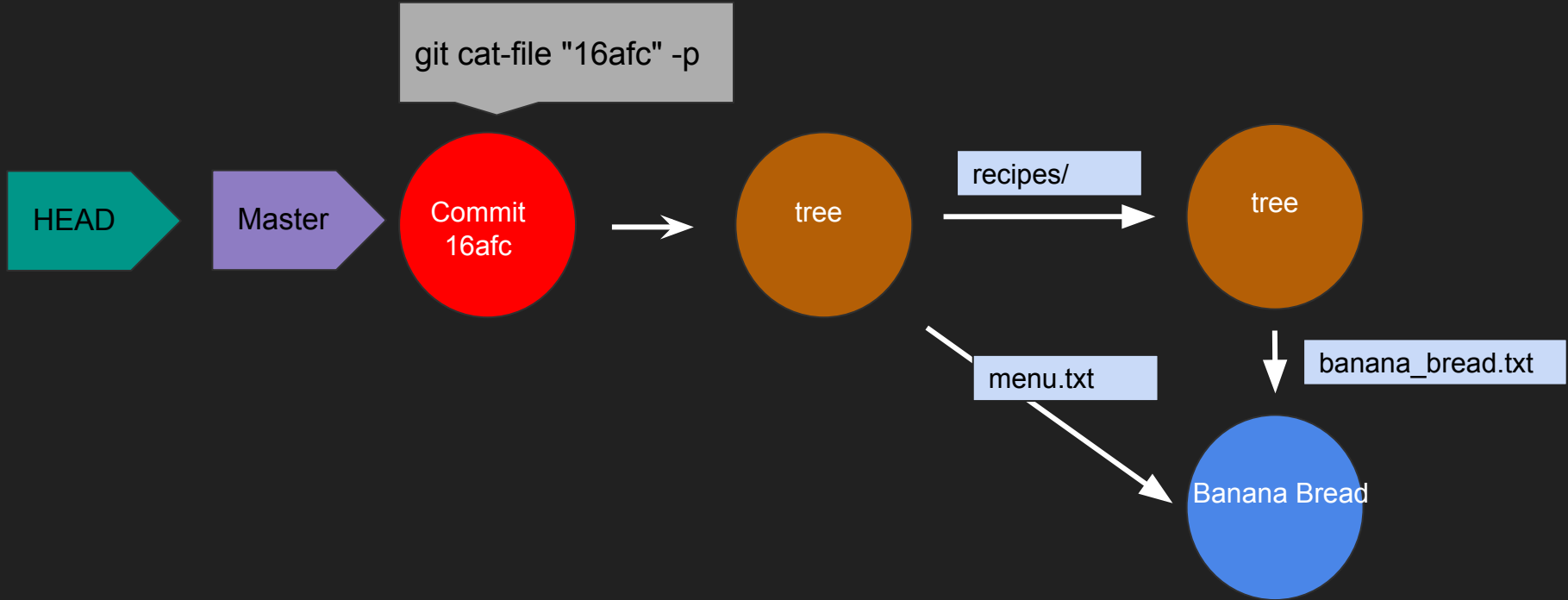
```
commit 16afc70b0211807255c00e9ac7d5bee648075a43 (HEAD -> master)
```

```
Author: Paul Z. Cheng <paul.z.cheng@gmail.com>
```

```
Date: Sat May 8 15:00:49 2021 +0800
```

```
initial commit
```

# git commit : Tree and blob



# What is inside of initial commit

```
git cat-file "16afc70" -t
```

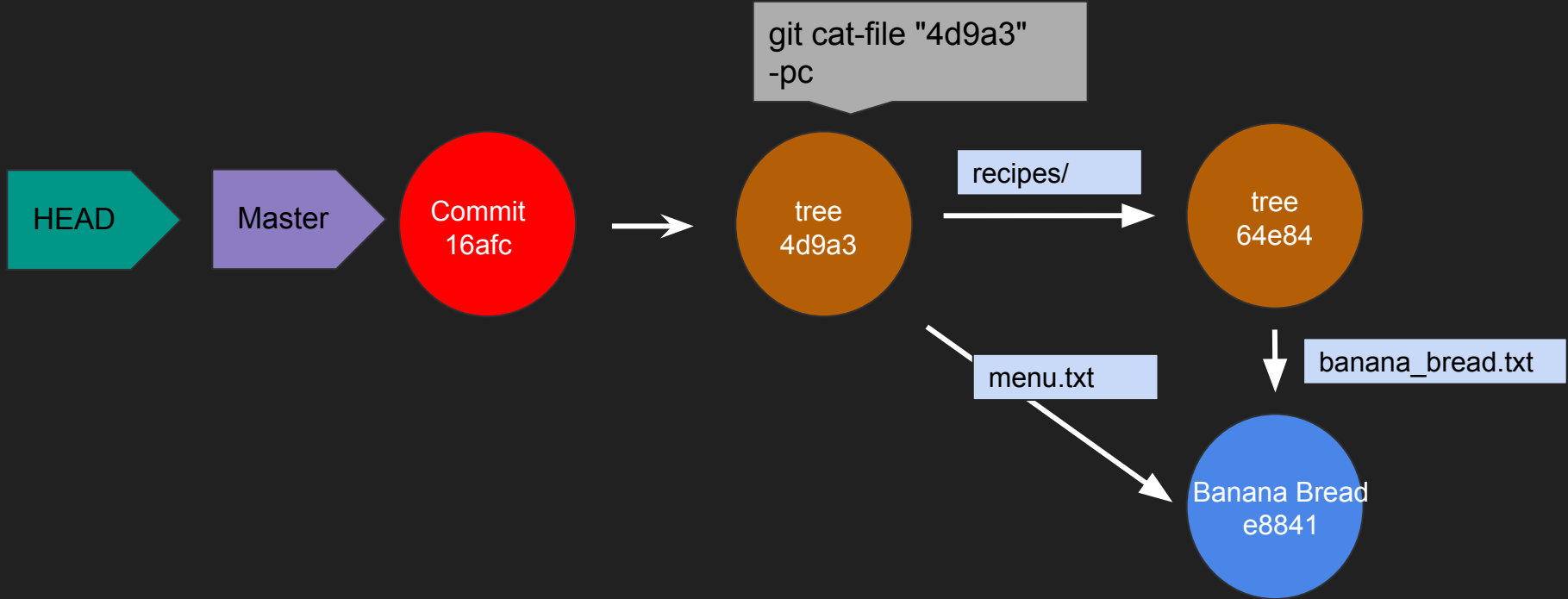
```
commit
```

```
git cat-file "16afc70" -p
```

```
tree 4d9a300bc821b3cb30ef3ae9e63d48f22cf0bc6c
author Paul Z. Cheng <paul.z.cheng@gmail.com> 1620457249 +0800
committer Paul Z. Cheng <paul.z.cheng@gmail.com> 1620457249 +0800
```

```
initial commit
```

# git commit : Tree and blob



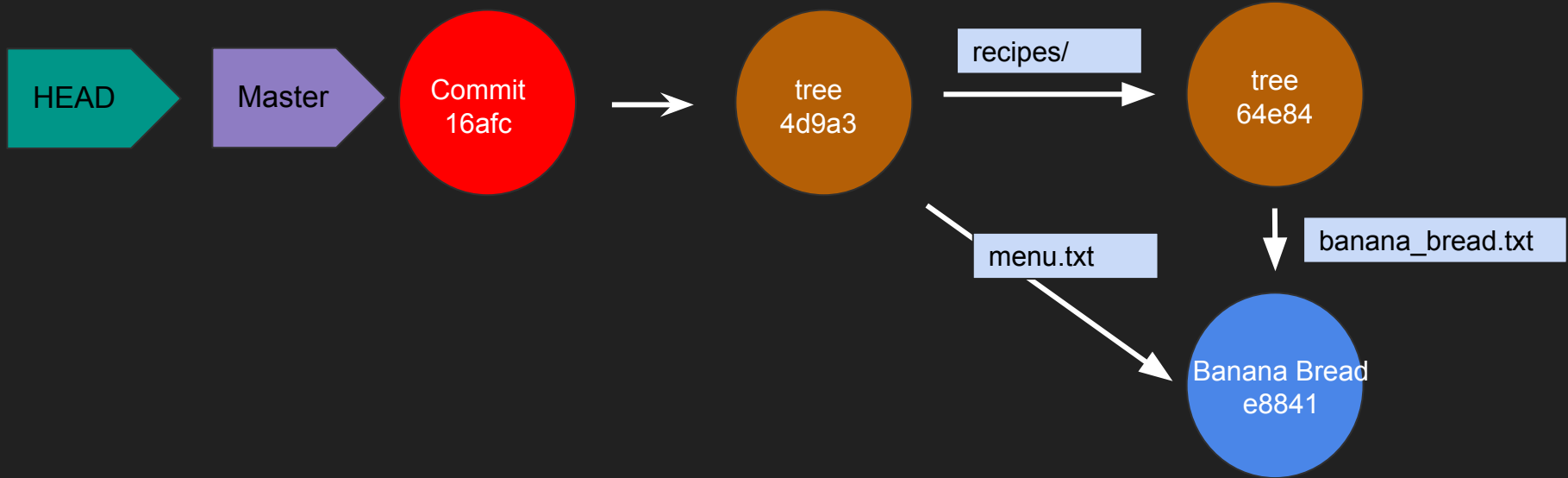
```
git cat-file "4d9a3" -t
```

```
tree
```

```
git cat-file "4d9a3" -p
```

```
100644 blob e884135b2103c673e876e850879539079033e670 menu.txt  
040000 tree 64e841cef75b1b1c4f9a5fe8e0e0559e8d806842 recipes
```

# git commit : Tree and blob



# Add some changes to README file

```
git commit -m "Add descriptor README file"
```

```
git log
```

```
commit 8215f3b135223478fbde164d42b20263110d8739 (HEAD, master)
```

```
Author: Paul Z. Cheng <paul.z.cheng@gmail.com>
```

```
Date: Sat May 8 15:18:41 2021 +0800
```

```
Add descriptor README file
```

```
commit 16afc70b0211807255c00e9ac7d5bee648075a43
```

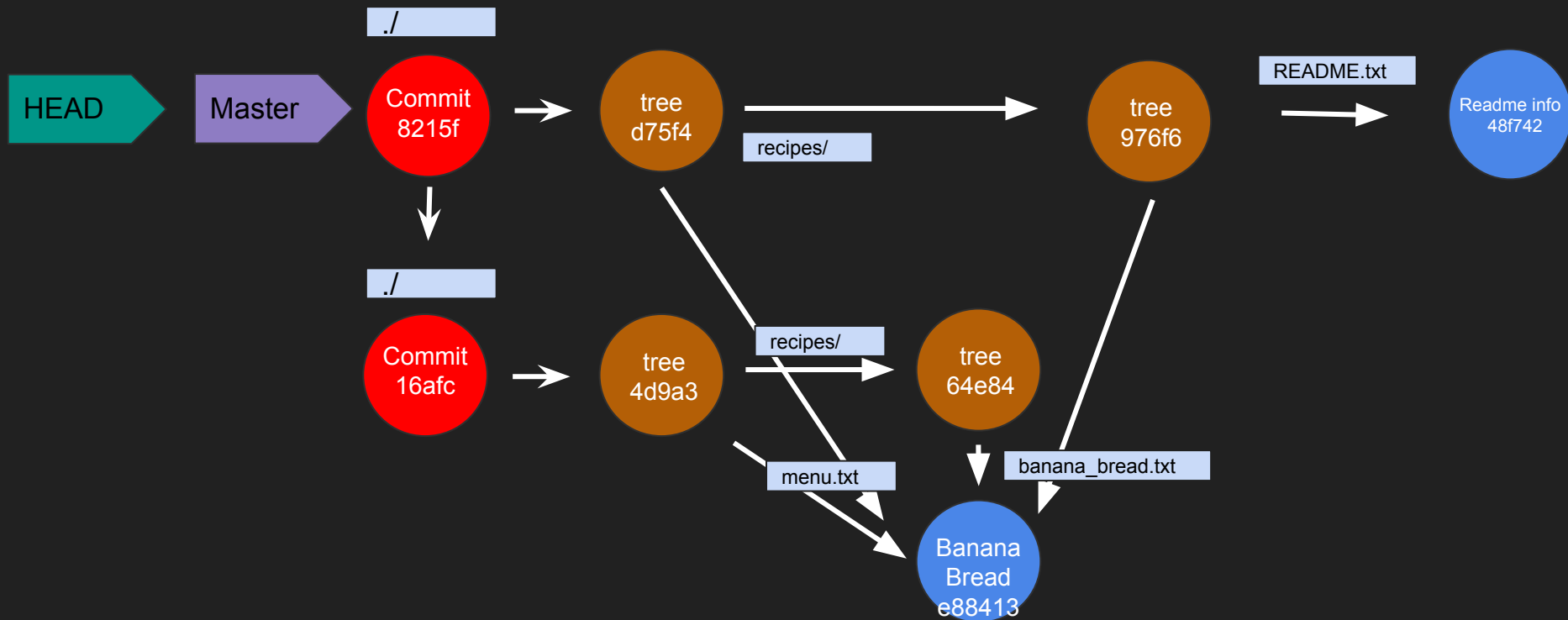
```
Author: Paul Z. Cheng <paul.z.cheng@gmail.com>
```

```
Date: Sat May 8 15:00:49 2021 +0800
```

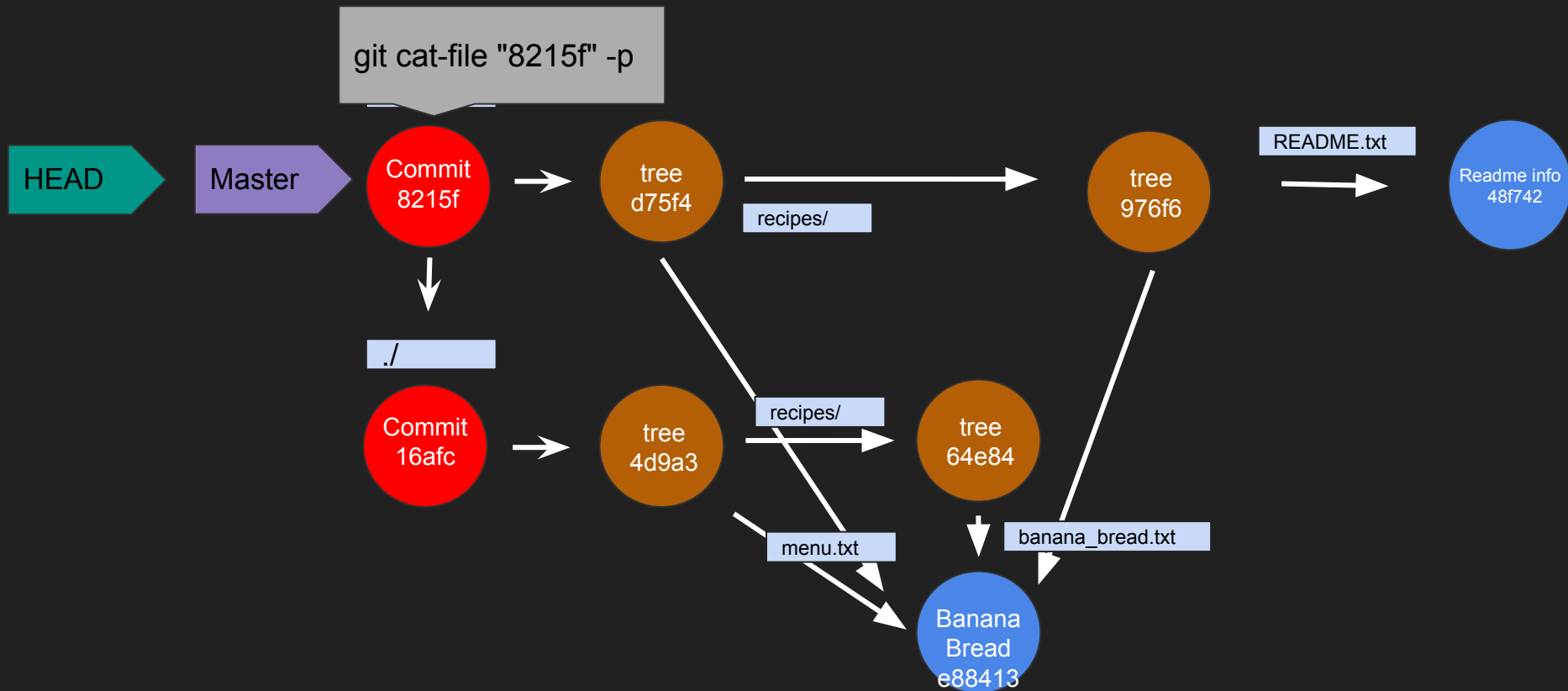
```
initial commit
```



# Commit README.txt



# Commit README.txt



```
git cat-file "8215f" -p
```

```
tree d75f44bf1fb72cde2479ba4fc68b269b37a92298
parent 16afc70b0211807255c00e9ac7d5bee648075a43
author Paul Z. Cheng <paul.z.cheng@gmail.com> 1620458321 +0800
committer Paul Z. Cheng <paul.z.cheng@gmail.com> 1620458321 +0800
```

```
Add descriptor README file
```

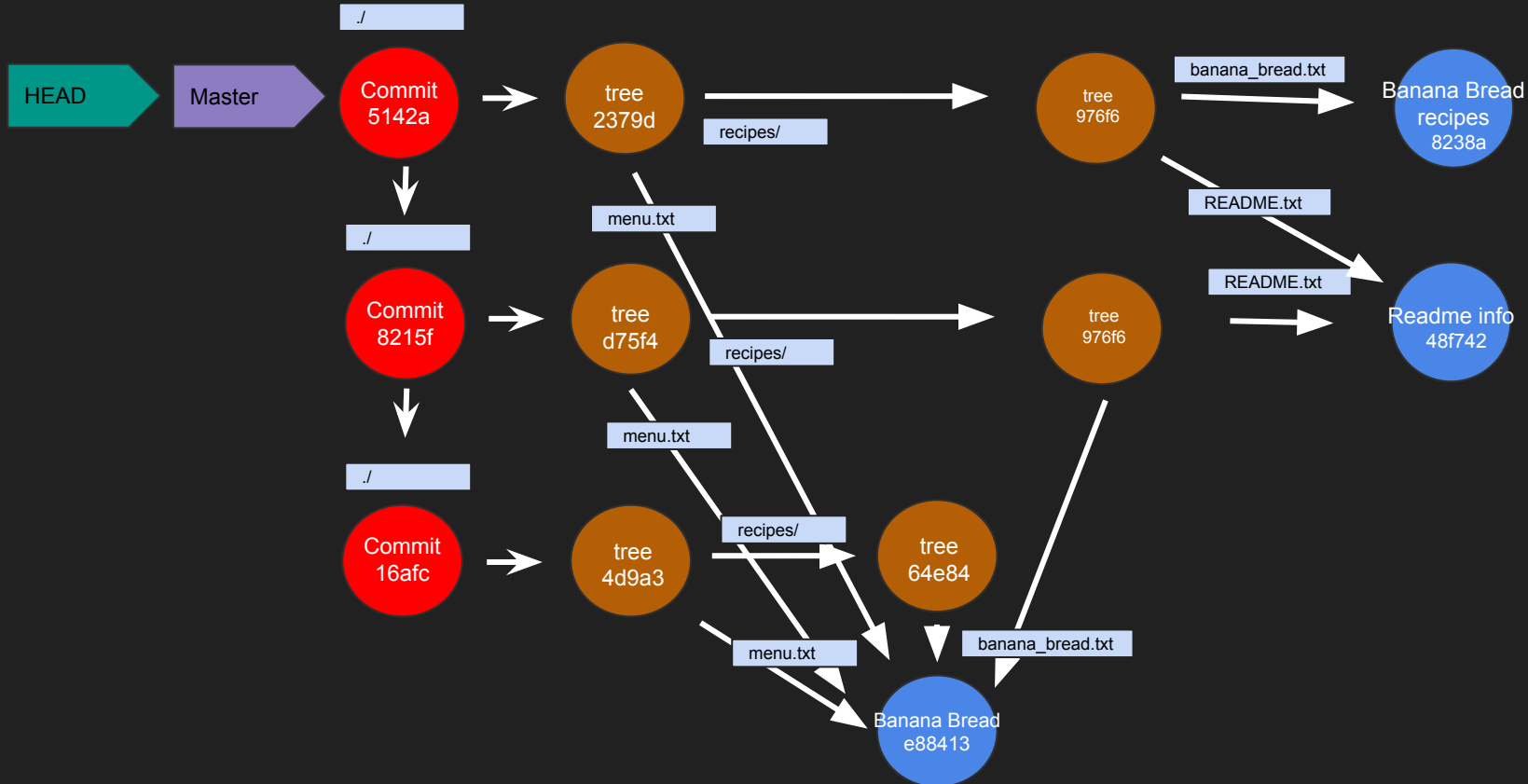
# Add banana bread recipe

```
git add banana_bread.txt  
git commit -m "paul's banana_bread recipe"
```

## Banana Bread

3 ripe bananas  
1/3 cup melted unsalted butter  
1 teaspoon baking soda  
1 pinch of salt  
3/4 cup sugar  
1 egg  
1 teaspoon vanilla extract  
3/2 cups all-purpose flour

# Commit add banana\_bread.txt recipes



# Annotated tag

```
git tag -a mytag  
# check out tag  
git tag
```

```
mytag
```

tag is saved inside of .git/refs/tags/mytag with flowing sha1  
386d6273736b896086867f6439bc536920735594

tag is store inside the .git/refs/tags

```
.  
├── branches  
├── COMMIT_EDITMSG  
├── config  
├── description  
├── HEAD  
├── hooks  
├── index  
├── info  
├── logs  
├── objects  
├── refs  
│   ├── heads  
│   │   └── master  
│   └── tags  
│       └── mytag
```

# Annotated tag, What is inside the tag

```
git cat-file "386d62" -p
```

```
object 5142a9653e8795ae428751c70782d9673d5bce41
type commit
tag mytag
tagger Paul Z. Cheng <paul.z.cheng@gmail.com> 1620564511 +0800

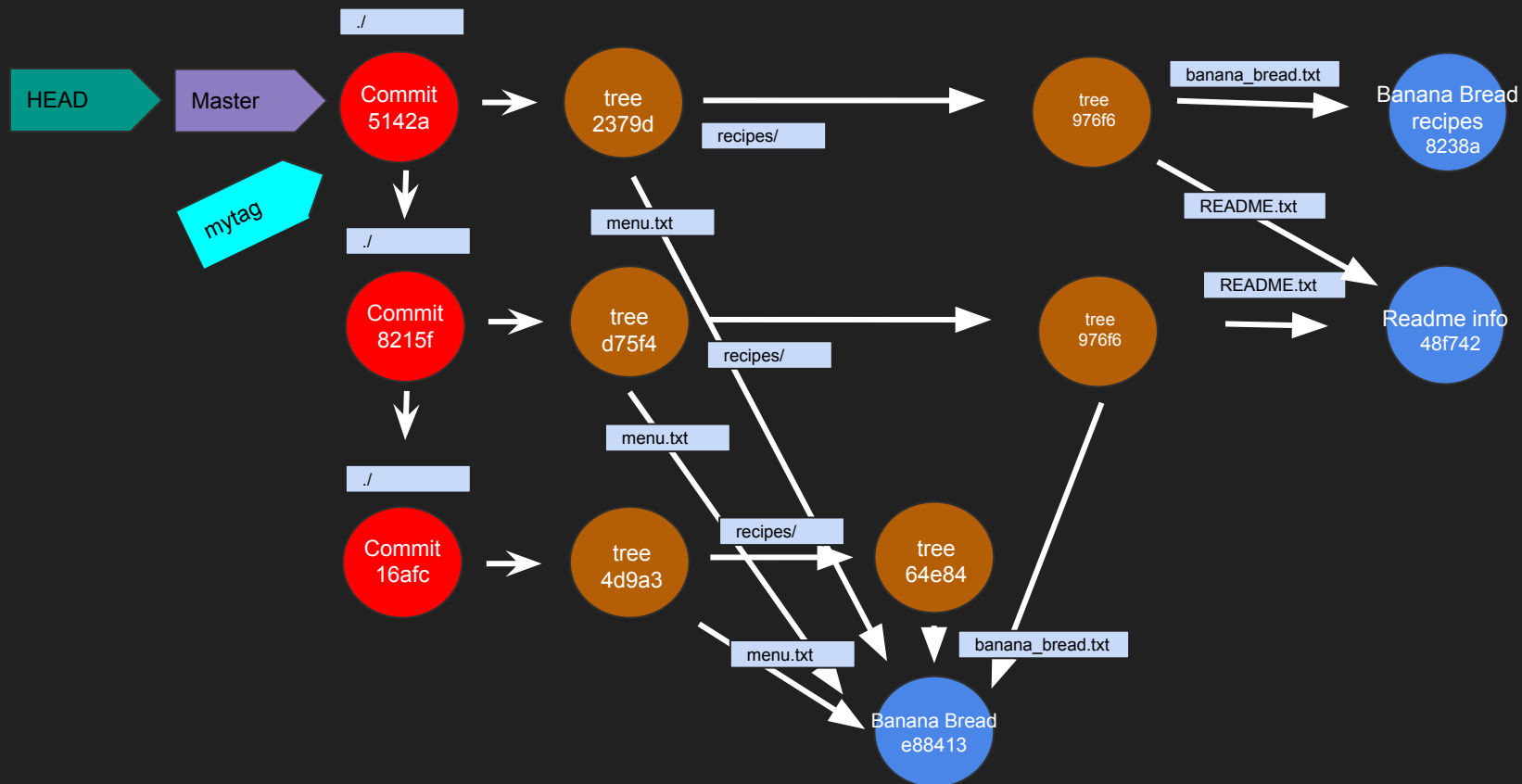
my tag
```

```
git cat-file "5142a" -p
```

```
tree 2379dc77acaa599e55b713f128b34f9dc9cbce25
parent 8215f3b135223478fbde164d42b20263110d8739
author Paul Z. Cheng <paul.z.cheng@gmail.com> 1620459827 +0800
committer Paul Z. Cheng <paul.z.cheng@gmail.com> 1620459827 +0800

paul's banana_bread receipt
```

# Annotated Tag





# Similar to detached head

```
git checkout mytag
```

You are in `'detached HEAD'` state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using `-c` with the switch `command`. Example:

```
git switch -c <new-branch-name>
```

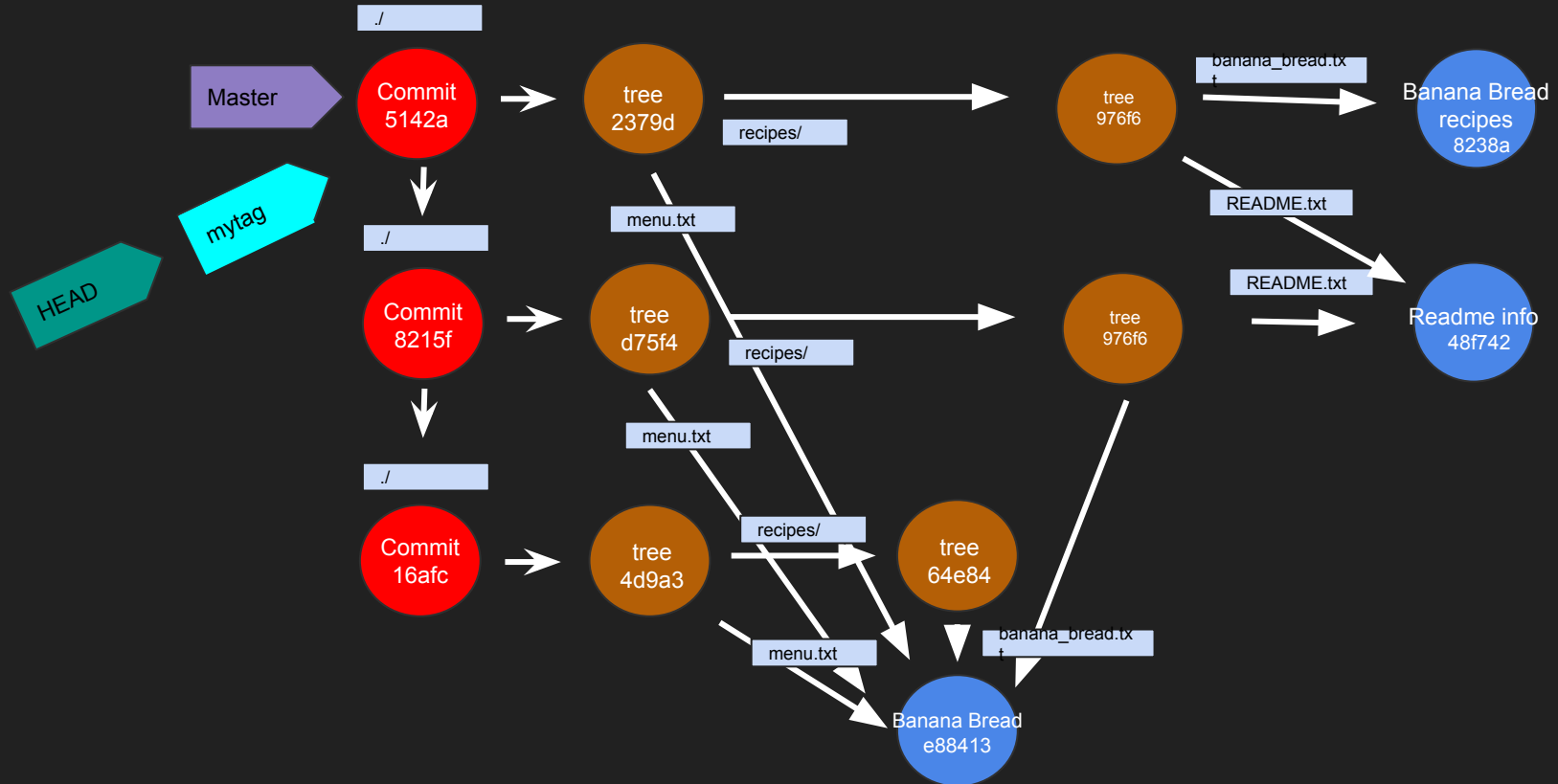
Or undo this operation with:

```
git switch -
```

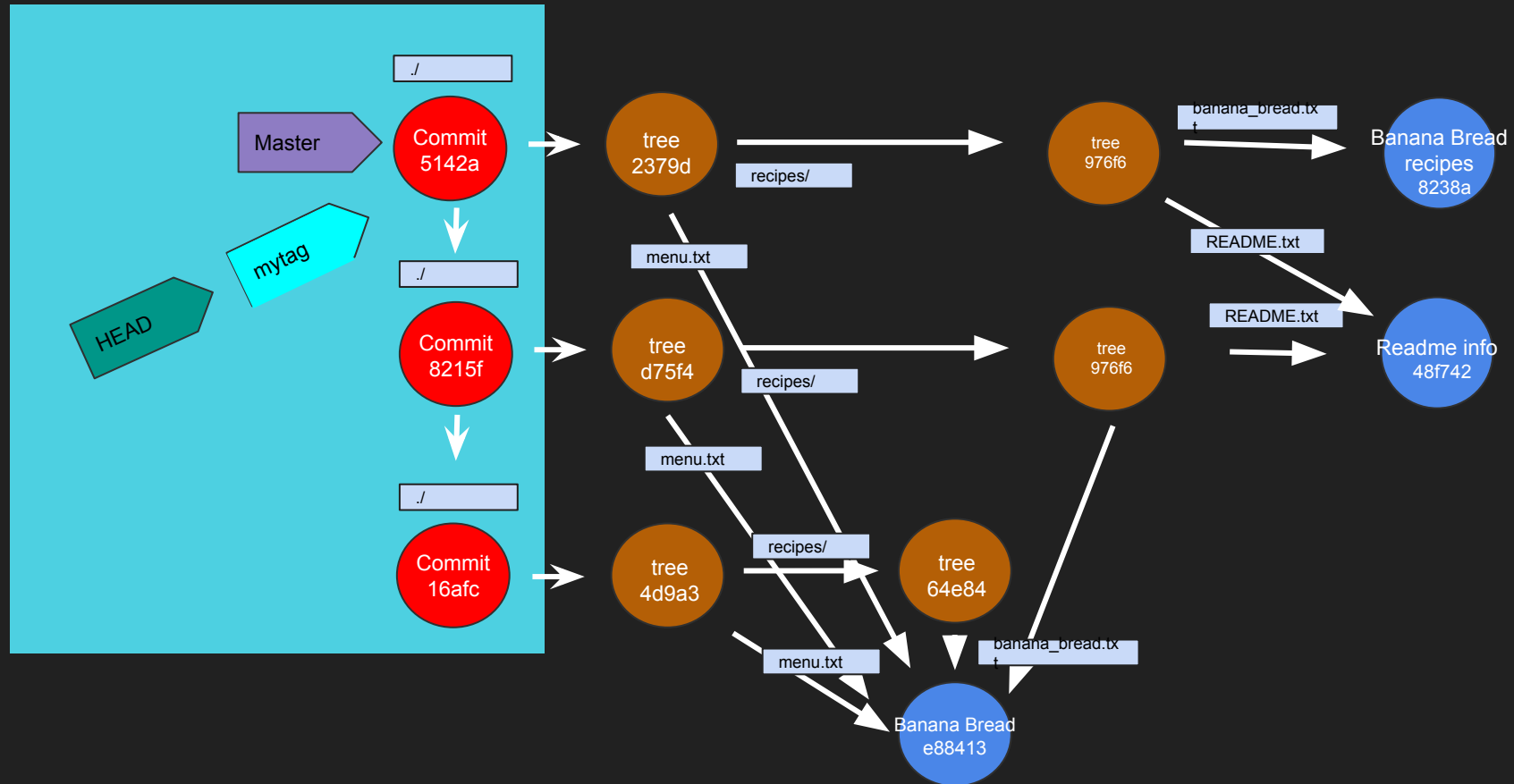
Turn off this advice by setting config variable `advice.detachedHead` to `false`

```
HEAD is now at 5142a96 paul's banana_bread receipt
```

# Detached head checkout tag



# git log



# git log

```
commit 5142a9653e8795ae428751c70782d9673d5bce41 (Head, tag: mytag, master)
```

```
Author: Paul Z. Cheng <paul.z.cheng@gmail.com>
```

```
Date: Sat May 8 15:43:47 2021 +0800
```

```
paul's banana_bread recipe
```

```
commit 8215f3b135223478fbde164d42b20263110d8739
```

```
Author: Paul Z. Cheng <paul.z.cheng@gmail.com>
```

```
Date: Sat May 8 15:18:41 2021 +0800
```

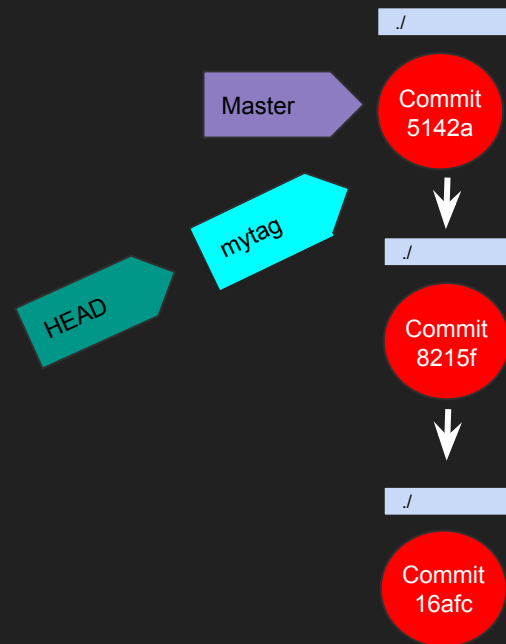
```
Add descriptor README file
```

```
commit 16afc70b0211807255c00e9ac7d5bee648075a43
```

```
Author: Paul Z. Cheng <paul.z.cheng@gmail.com>
```

```
Date: Sat May 8 15:00:49 2021 +0800
```

```
initial commit
```



# Branching visualization

```
git checkout master
git branch risa
"make some changes to banana bread.txt"
git add banana_bread.txt
git commit -m "Risa's recipe."
```

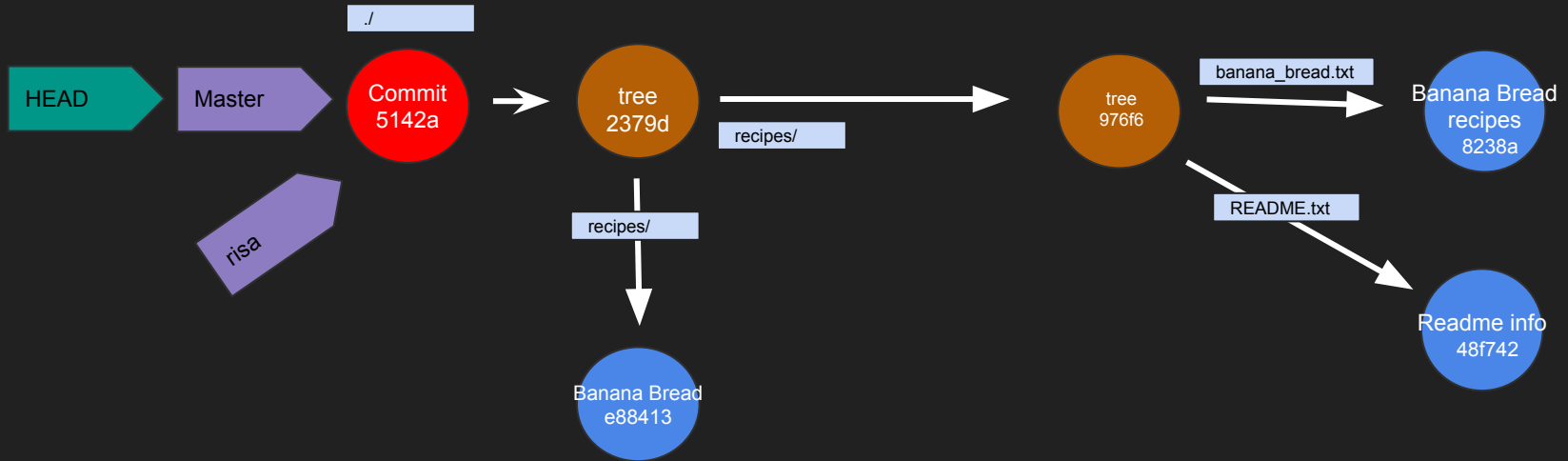
## Banana Bread

4 ripe bananas  
1/3 cup melted unsalted butter  
1 teaspoon baking soda  
1 pinch of salt  
2 cup sugar  
2 egg  
1 teaspoon vanilla extract  
3/2 cups all-purpose flour

risa branch is store inside the .git/refs/heads

```
.
├── branches
├── COMMIT_EDITMSG
├── config
├── description
├── HEAD
├── hooks
├── index
├── info
├── logs
├── objects
└── refs
    ├── heads
    │   ├── master
    │   └── risa
    ├── tags
    │   └── mytag
```

# git branch risa

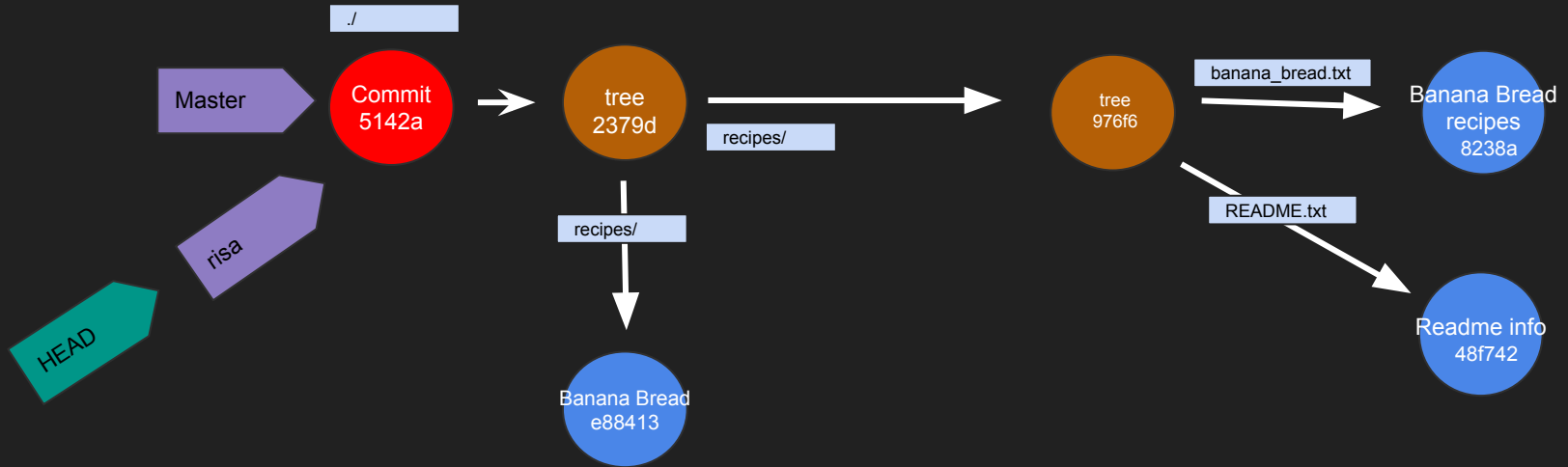


# Branching

Creating a tag with reference to commit



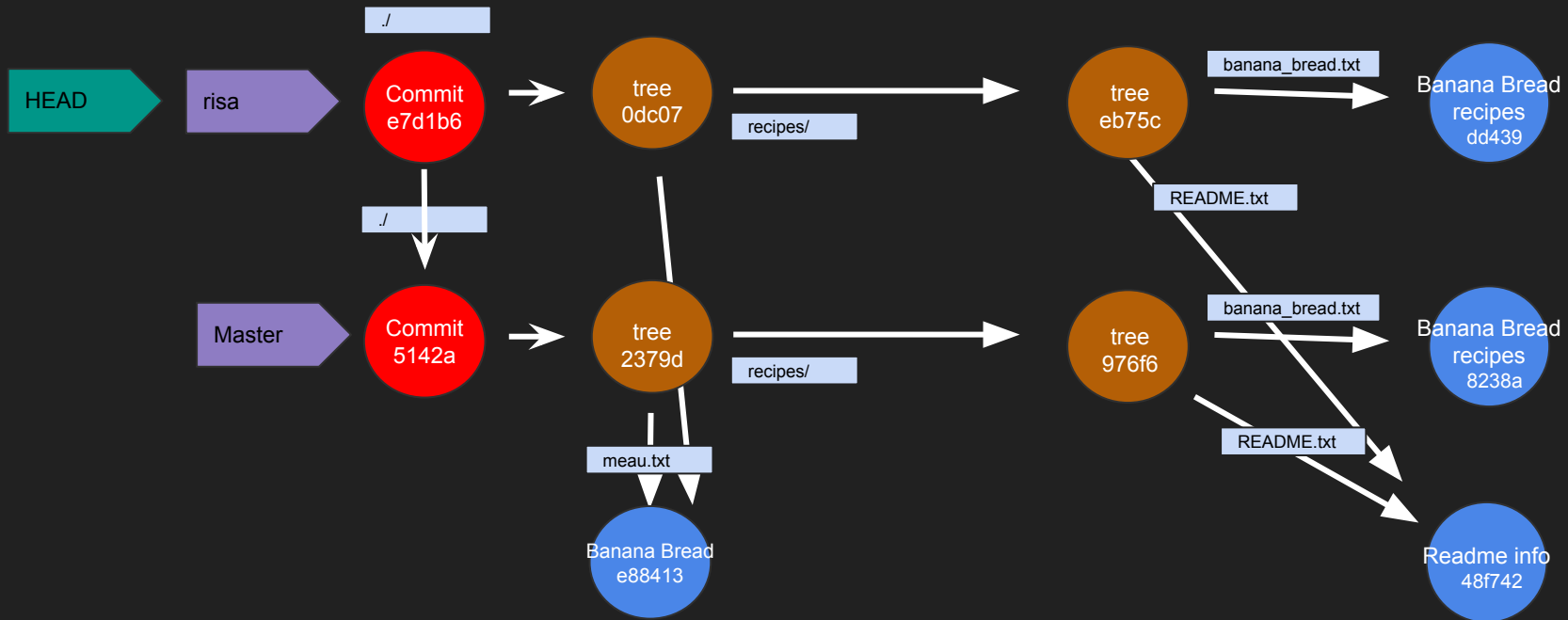
# git checkout risa



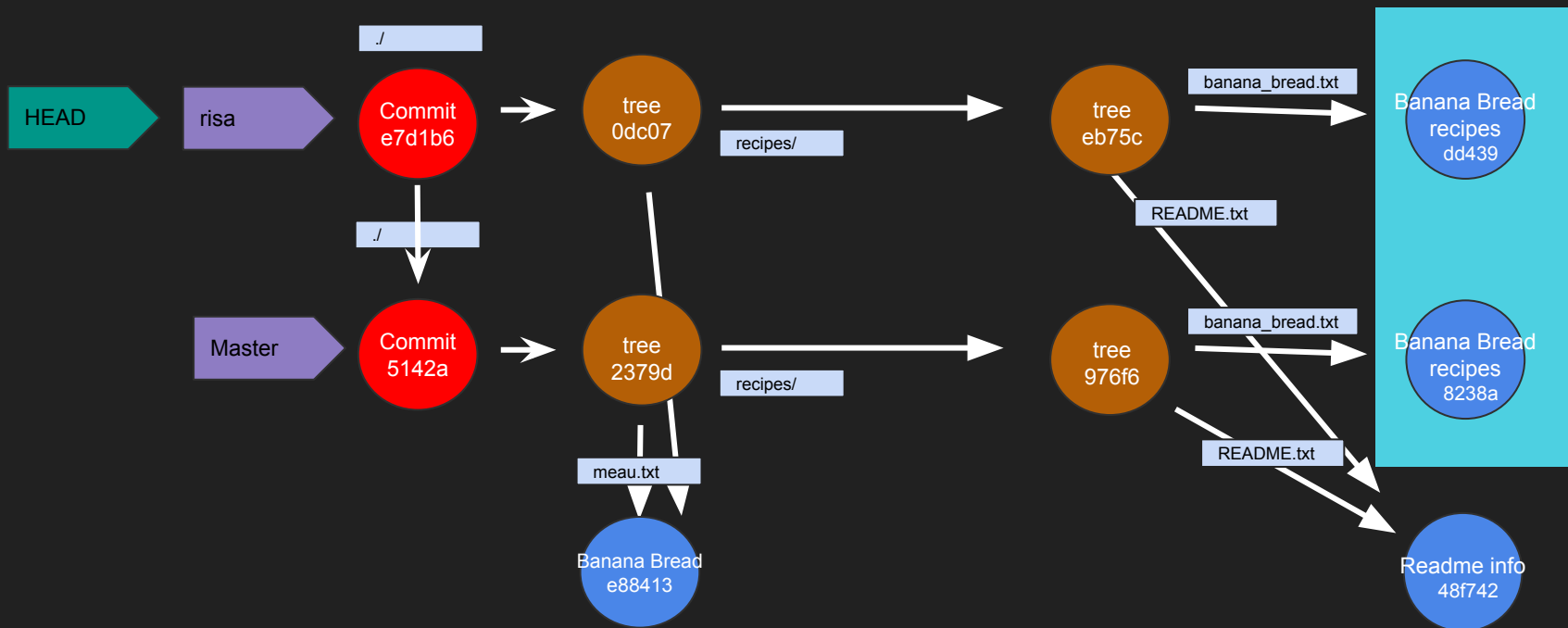
# Branching

Moving Head to referencing to a branch or commit

# git commit -m "Risa's recipe."



# git diff risa master

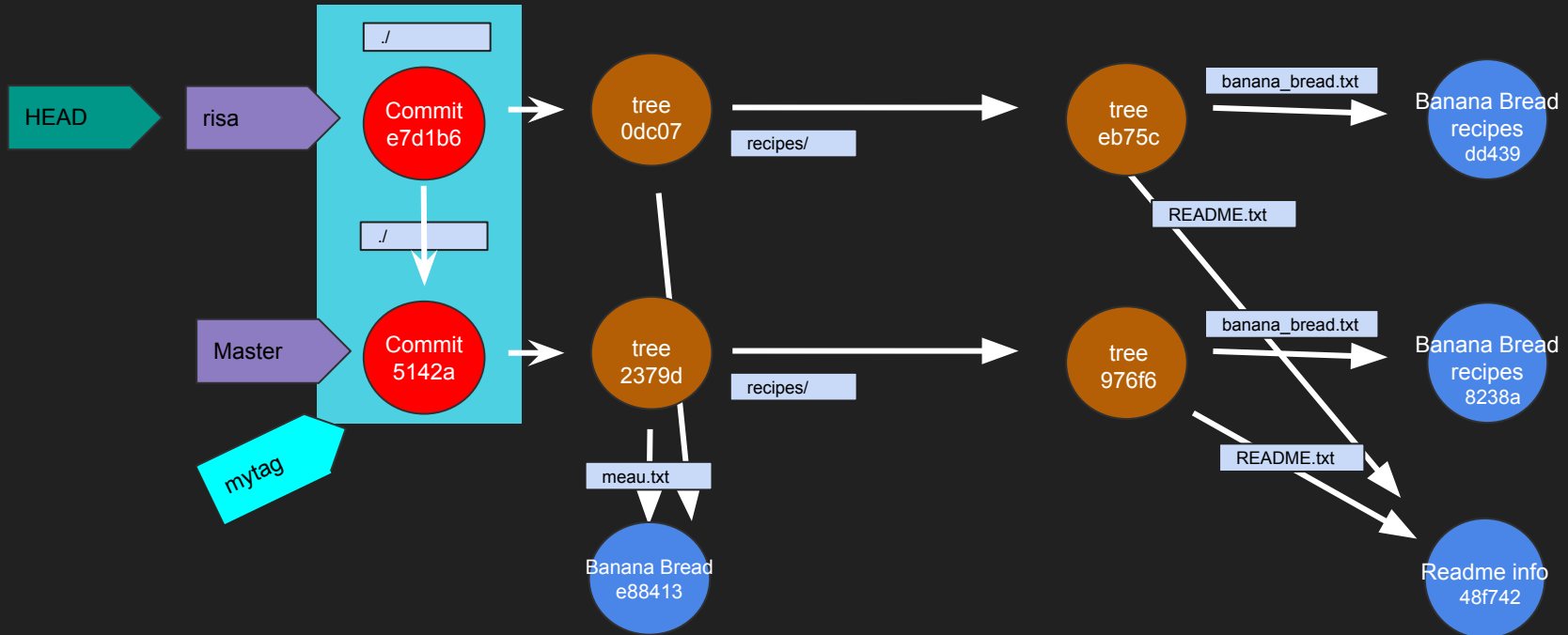


# git diff risa master

```
diff --git a/recipes/banana_bread.txt b/recipes/banana_bread.txt
index dd43918..8238a5e 100644
--- a/recipes/banana_bread.txt
+++ b/recipes/banana_bread.txt
@@ -1,10 +1,10 @@
 Banana Bread

-4 ripe bananas
+3 ripe bananas
 1/3 cup melted unsalted butter
 1 teaspoon baking soda
 1 pinch of salt
-2 cup sugar
-2 egg
+3/4 cup sugar
+1 egg
 1 teaspoon vanilla extract
 3/2 cups all-purpose flour
```

# The relationship between risa and master



# git log

```
commit e7d1b63cee8d473945c4794faebe8340d1257bb4 (HEAD -> risa)
```

```
Author: Paul Z. Cheng <paul.z.cheng@gmail.com>
```

```
Date: Sun May 9 21:40:37 2021 +0800
```

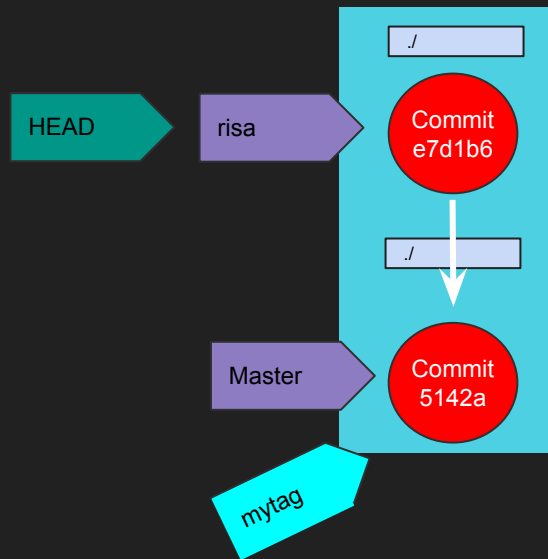
```
Risa's recipe.
```

```
commit 5142a9653e8795ae428751c70782d9673d5bce41 (tag: mytag, master)
```

```
Author: Paul Z. Cheng <paul.z.cheng@gmail.com>
```

```
Date: Sat May 8 15:43:47 2021 +0800
```

```
paul's banana_bread receipt
```



# Resolving merge conflict.

When merging two branch, git tells us what to do next:

"Fix conflicts and then commit the result."

Step 1: Open the conflicted file and merge the changes with edits.

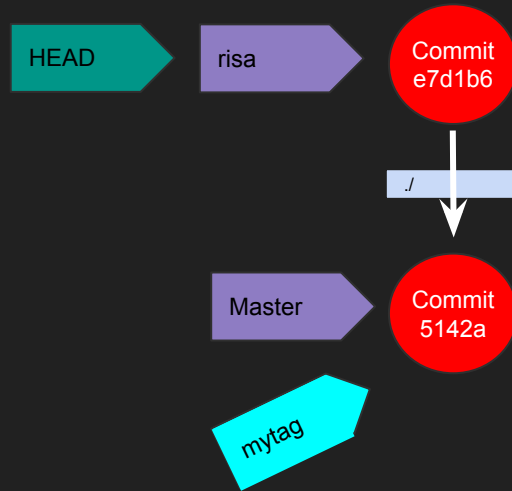
Step 2: Stage the change.

Step 3: Run **git commit** to finish the merge process.

Before continuing, you may run **git status** and read what it says.



# Example



```
diff --git a/recipes/banana_bread.txt b/recipes/banana_bread.txt
index d443918..8238a5e 100644
--- a/recipes/banana_bread.txt
+++ b/recipes/banana_bread.txt
@@ -1,10 +1,10 @@
 Banana Bread

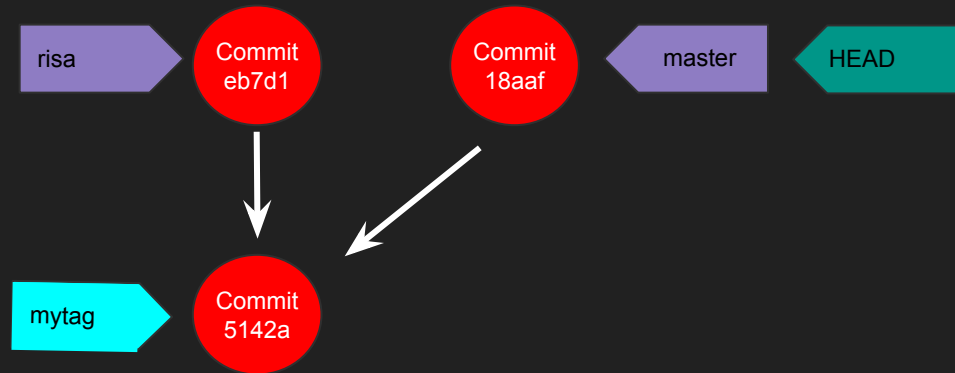
-4 ripe bananas
+3 ripe bananas
 1/3 cup melted unsalted butter
 1 teaspoon baking soda
 1 pinch of salt
-2 cup sugar
-2 egg
+3/4 cup sugar
+1 egg
 1 teaspoon vanilla extract
 3/2 cups all-purpose flour
```

# Make some changes to recipe

## Banana Bread

3.5 ripe bananas  
1/3 cup melted unsalted butter  
1 teaspoon baking soda  
1 pinch of salt  
1 cup sugar  
1 egg  
1 teaspoon vanilla extract  
3/2 cups all-purpose flour

```
git add .  
git commit -m "New recipe after seen risa."
```



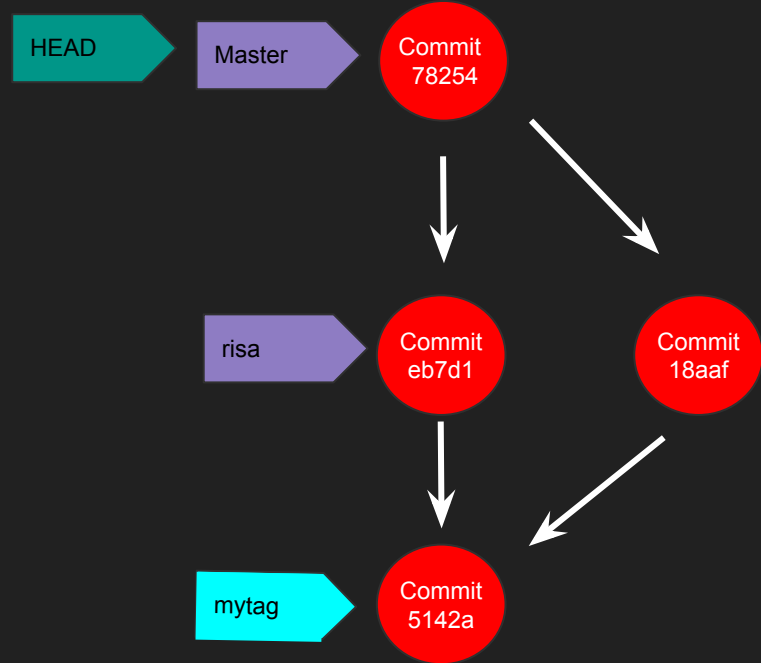
```
<<<<<< HEAD
3.5 ripe bananas
1/3 cup melted unsalted butter
1 teaspoon baking soda
1 pinch of salt
1 cup sugar
1 egg
=====
4 ripe bananas
1/3 cup melted unsalted butter
1 teaspoon baking soda
1 pinch of salt
2 cup sugar
2 egg
>>>>>> risa
1 teaspoon vanilla extract
3/2 cups all-purpose flour
```

# save this

3.5 ripe bananas  
1/3 cup melted unsalted butter  
1 teaspoon baking soda  
1 pinch of salt  
1 cup sugar  
1 egg  
1 teaspoon vanilla extract  
3/2 cups all-purpose flour

# Merge conflict exercise

```
git add .  
git commit -m "merge commit"
```



# Wa la

```
commit 782546681f72e0c38c6d9fb9b6e2b505db0a7dda (HEAD -> master)
```

```
Merge: 18aaf3f e7d1b63
```

```
Author: Paul Z. Cheng <paul.z.cheng@gmail.com>
```

```
Date: Mon May 10 11:03:34 2021 +0800
```

```
merge commit
```

```
commit 18aaf3fc4dd2dc9abf95d58261e1a7415d26279d
```

```
Author: Paul Z. Cheng <paul.z.cheng@gmail.com>
```

```
Date: Mon May 10 09:08:22 2021 +0800
```

```
New recipe after seen risa.
```

```
commit e7d1b63cee8d473945c4794faebe8340d1257bb4 (risa)
```

```
Author: Paul Z. Cheng <paul.z.cheng@gmail.com>
```

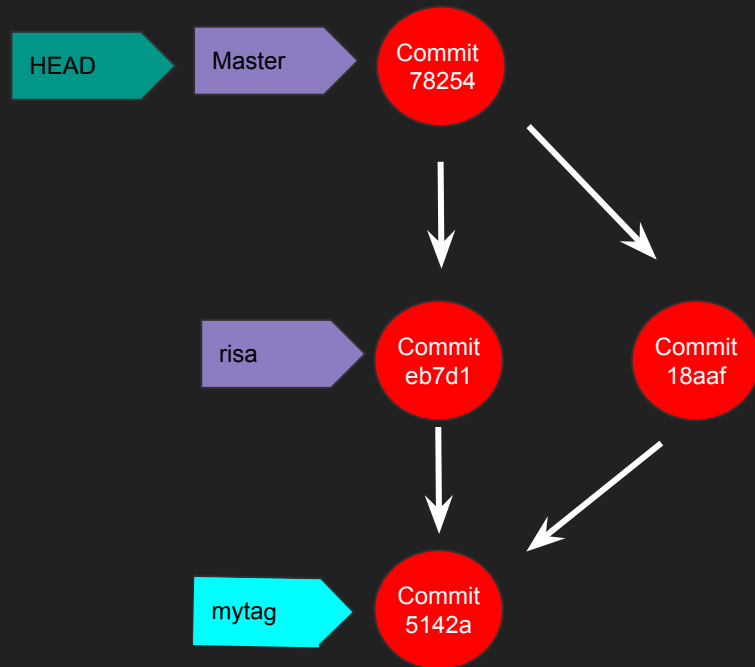
```
Date: Sun May 9 21:40:37 2021 +0800
```

```
Risa's recipe.
```

```
commit 5142a9653e8795ae428751c70782d9673d5bce41 (tag: mytag)
```

```
Author: Paul Z. Cheng <paul.z.cheng@gmail.com>
```

```
Date: Sat May 8 15:43:47 2021 +0800
```





commit 782546681f72e0c38c6d9fb9b6e2b505db0a7dda (HEAD -> master)

Merge: 18aaf3f e7d1b63

Author: Paul Z. Cheng <paul.z.cheng@gmail.com>

Date: Mon May 10 11:03:34 2021 +0800

merge commit

commit 18aaf3fc4dd2dc9abf95d58261e1a7415d26279d

Author: Paul Z. Cheng <paul.z.cheng@gmail.com>

Date: Mon May 10 09:08:22 2021 +0800

New recipe after seen risa.

commit e7d1b63cee8d473945c4794faebe8340d1257bb4 (risa)

Author: Paul Z. Cheng <paul.z.cheng@gmail.com>

Date: Sun May 9 21:40:37 2021 +0800

Risa's recipe.

commit 5142a9653e8795ae428751c70782d9673d5bce41 (tag: mytag)

Author: Paul Z. Cheng <paul.z.cheng@gmail.com>

Date: Sat May 8 15:43:47 2021 +0800

# Home Work

## Revert time exercise.

### Step 1: setting up your repository with a python "MissNum.py" file.

(1) write a python function that find a missing number in a list

ex:

```
iter = [0,1,2,4,5,6]
miss = miss_num_func(iter)
print(miss) # 3
```

(2) add and commit the changes

### Step 2: Make some changes to "MissNum.py" file.

(1) write a python function that find missing numbers in a list

ex:

```
iter = [0,1,2,4,5,6,8,9]
miss = miss_num_func(iter)
for i in miss:
    print(i) # 3 7
```

(2) add and commit the changes after editing MissNum.py

### Step 3: Reverting to step one and change file name.

(1) Check out Step 1

(2) Change the file name of MissNum.py to MissNum\_2.py or cp.

### Step 4: Create a new branch with reverted time step three.

(1) Make a new branch with name: reverttime

### Step 5: Merge branches

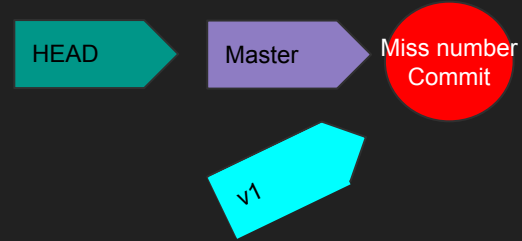
(1) Merge reverttime with master

### Results:

**Master with MissNum.py and MissNum\_2.py**

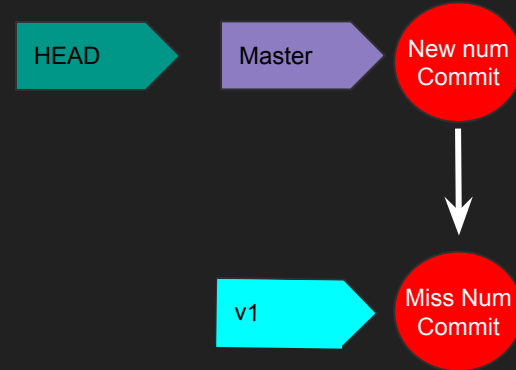
# Miss number code and make a tag

```
git add miss_num.py  
git commit -m "Missing number version I"  
git tag v1
```



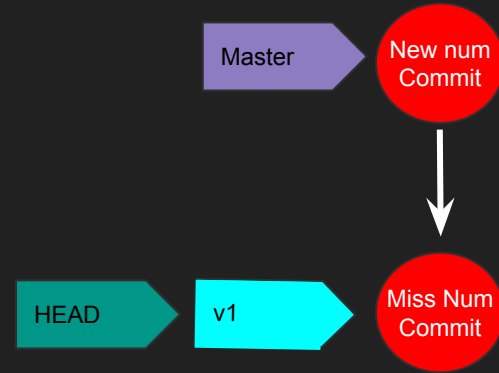
# Update the function

```
git add miss_num.py  
git commit -m "missing number takes in multiple missing  
number."
```



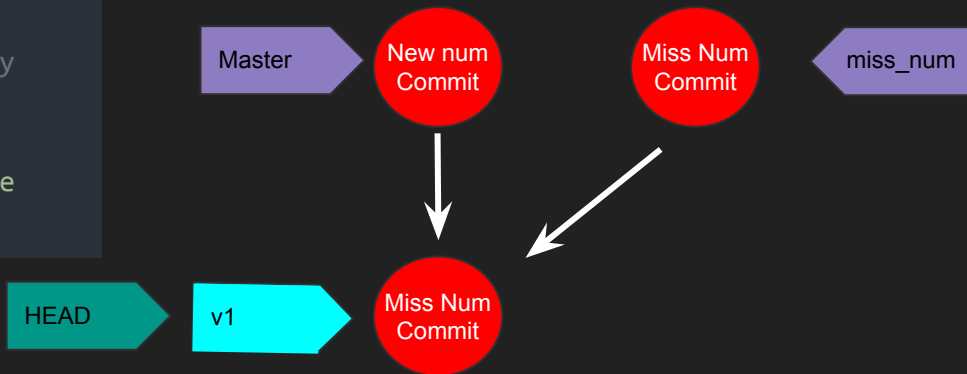
# checkout old tag, change name, create branch

```
git checkout v1
```



# Create a new branch

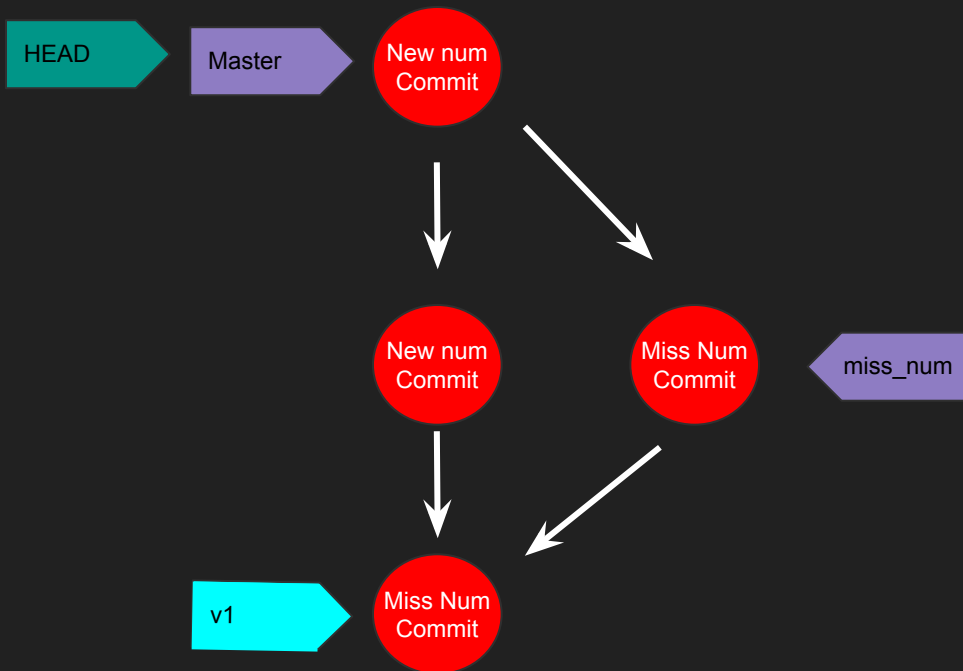
```
git branch miss_num
mv miss_num_v1.py
atom miss_num_v1.py # add some comments in the .py
file
git add miss_num_v1.py
git commit -m "Missing num version only detect one
missing number"
```



# Merge

```
git checkout master
git merge master miss_num

# ... moments later solving
conflict
git add .
git commit -m "Merge commit
with both missing number
versions"
```



Check your log : log sharing



# Team building exercise

**Step 1: Pull the newest updated gitastic github from paul.**

**Step 2: Push the two function files into your fork**

**Step 3: Choose a partner to work and share each other fork and solve the merge conflict of the two function**

**Step 4: Create a new python function that incorporate both functions (Make sure to update each as you work.)**

- Able to run as independent file: "python miss\_number.py"
- screen will print out: "Take in numbers:" and return missing value.
  - I expect to have if else statement in this.
  - if missing one number : print("There is one missing number : \*\*\*")
  - if multiple: print("there is multiple missing numbers : \*\*\*")

**Step 5: Send Paul a pull request with working miss\_number.py.**