

GIT Workshop Review

TYH Lab git Workshop
by Paul Z. Cheng

Github House Cleaning: August 13 Update

- Github no longer support password authentication!
- Remember to switch to either SSH or 2 factor authentication.

Any questions with

```
git status  
git add .  
git commit -m "xx files update"  
git log
```

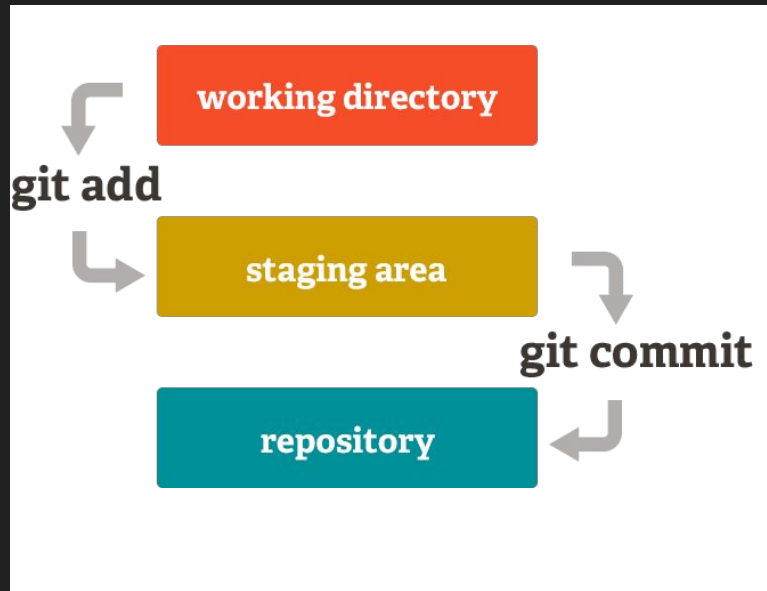
Daily edited and saving

```
## 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
```



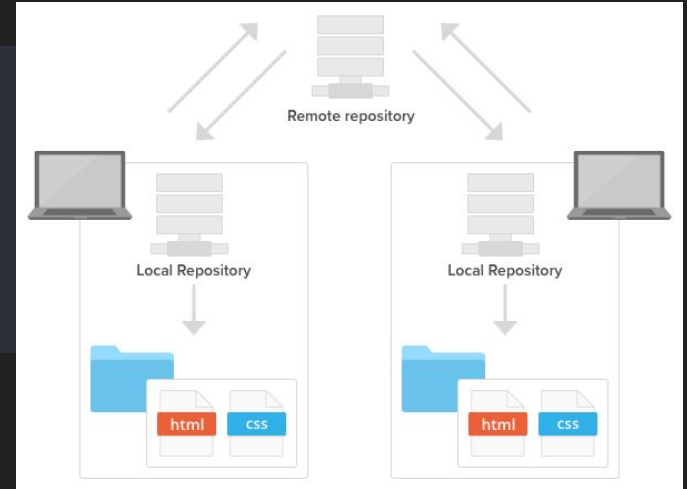
Pop Quiz!

- How to copy a git repo into your current computer?
- What is difference the between an annotated tag and a branch ?
- What are the two methods one can update current working directory with respect to their remote repo?
- What is the difference between these two?
- What is git?

Copying git repo

Copying git repo into your current Working directory

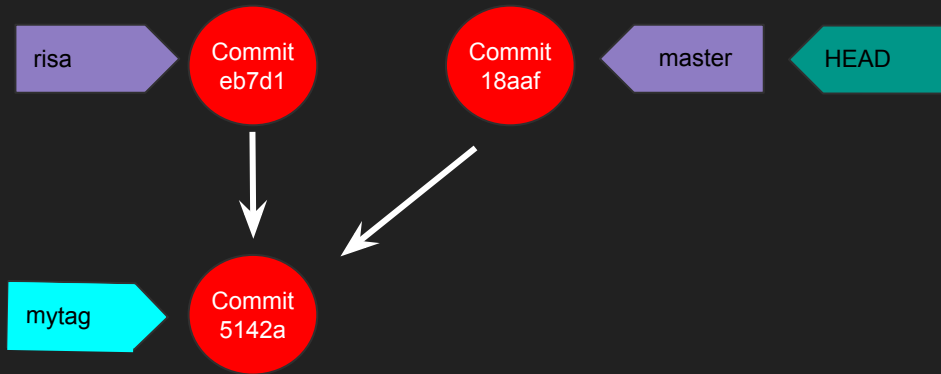
```
## Clone a directory  
git clone < git URL >  
  
# git remote setup  
git init  
git remote add origin <Remote URL>  
git pull origin master
```



Tag V Branch

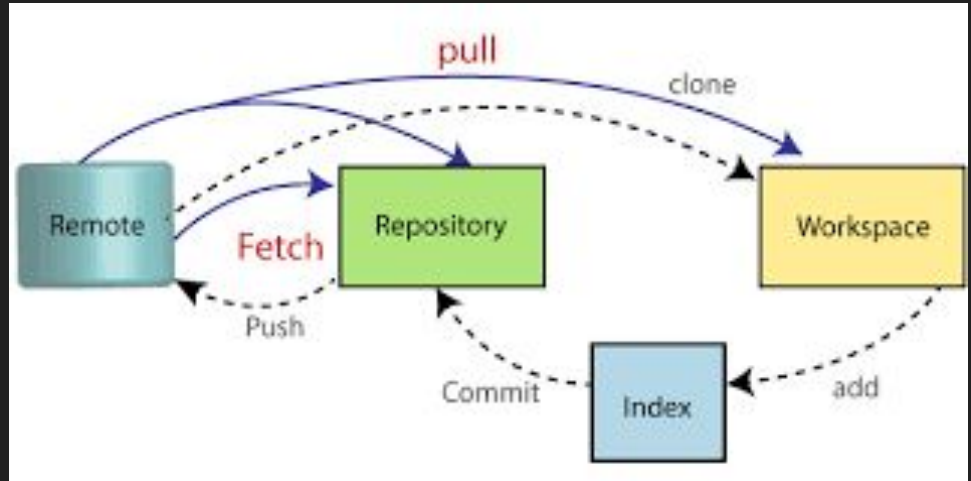
What is difference between annotated tag and branch?

- Annotated tag is unmovable marker pointed toward a commit
- Branch is move marker pointed toward commit, and it updates itself with every new commit.
- Basically, n Annotated tag can't be move and branch can.
- They are essentially the same, tag



Fetch or 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
```



What is GIT?

Onion





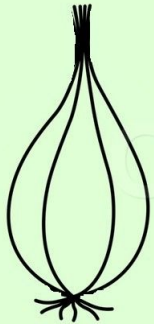
Git is a
Persistent Map

OpenGenus

Banana bread

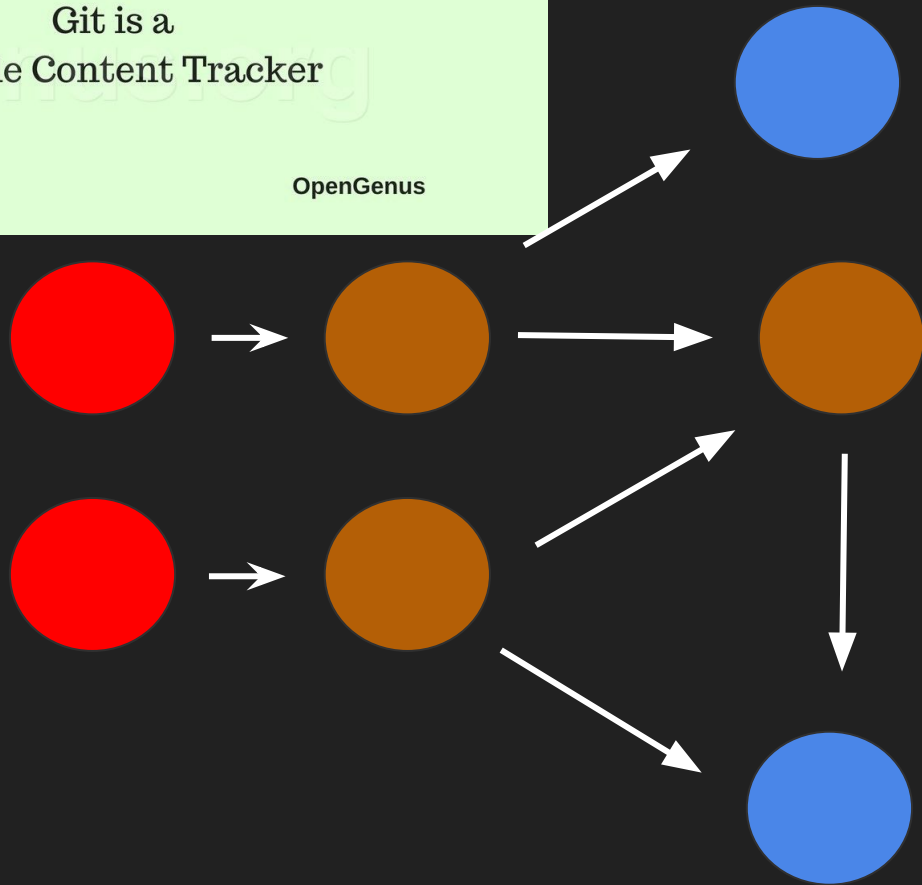


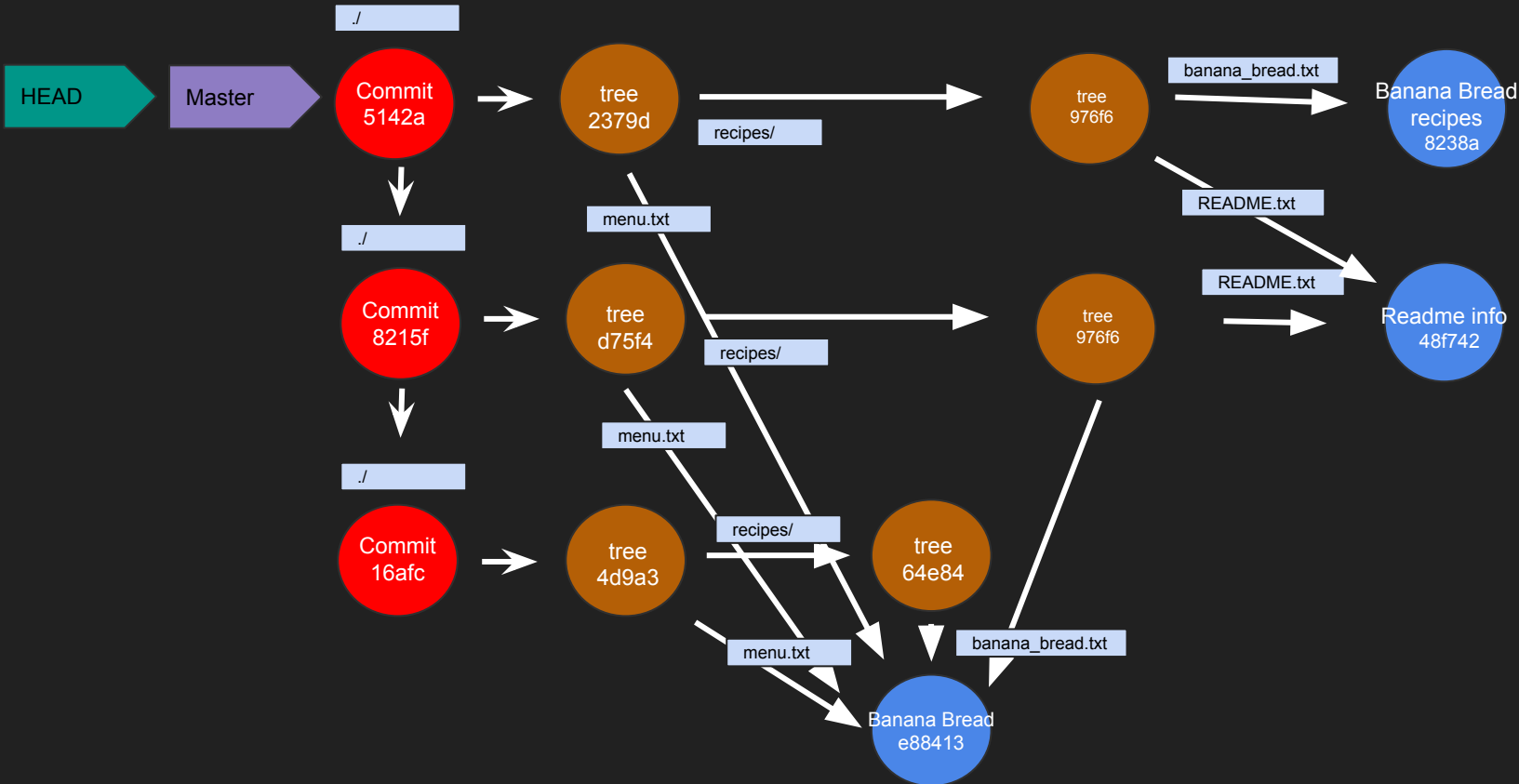
e884135b2103c673e876e850879539079033e670

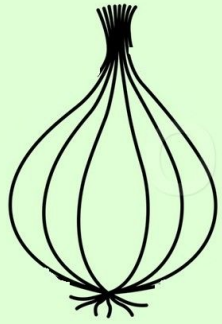


Git is a Simple Content Tracker

OpenGenus

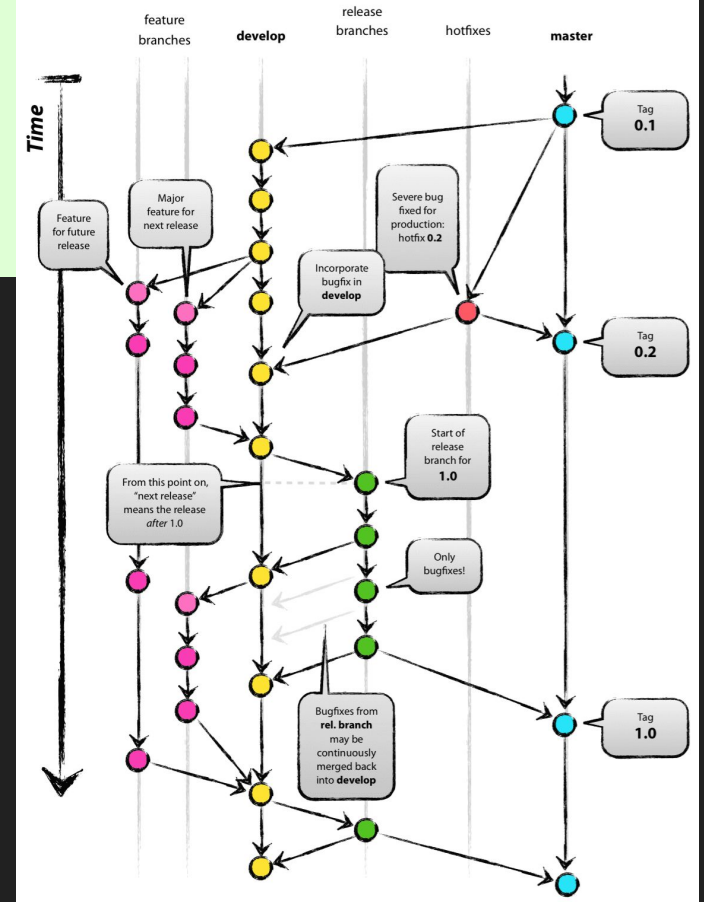




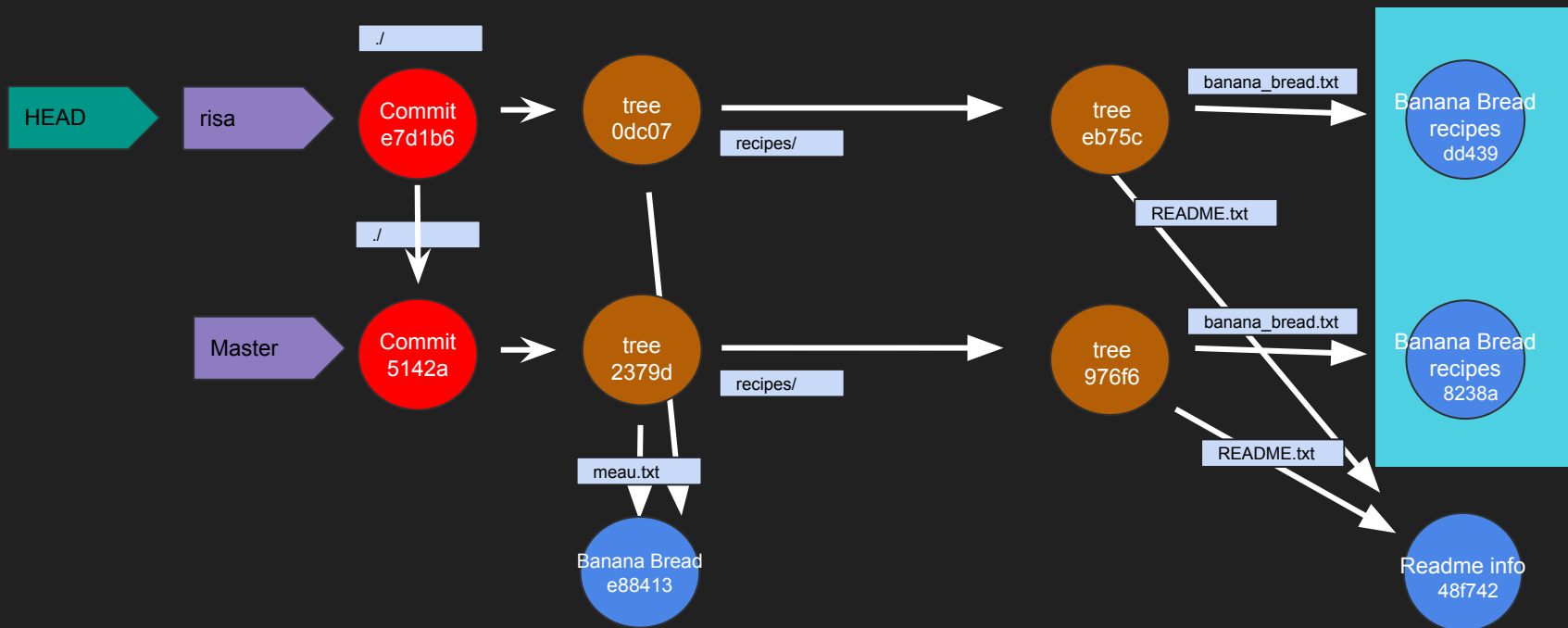


Git is a Revision Control System

OpenGenus



Branching



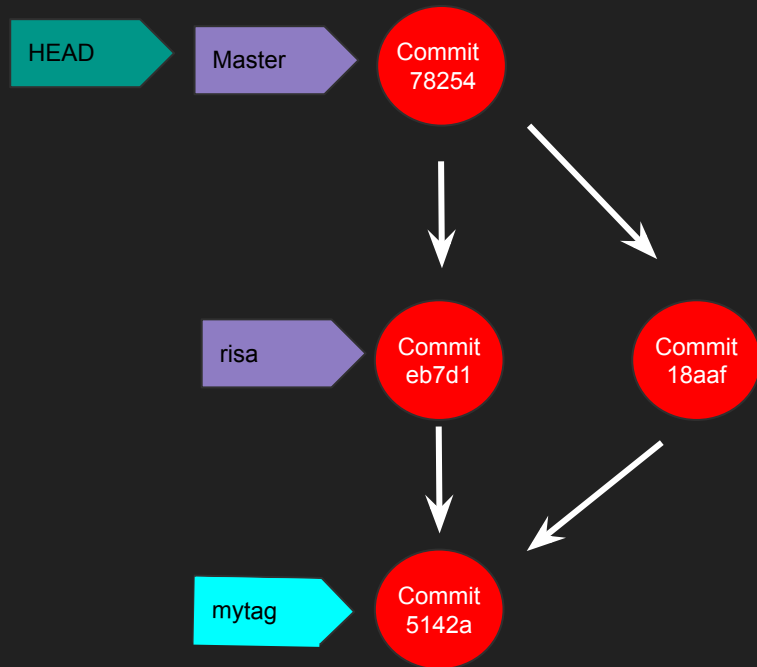
git diff risa master

```
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
```

Merge conflict

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



Resolving merge conflict.

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

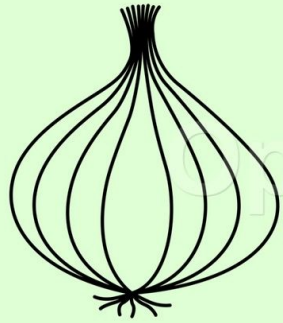
"Fix conflicts and then commit the result."

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.

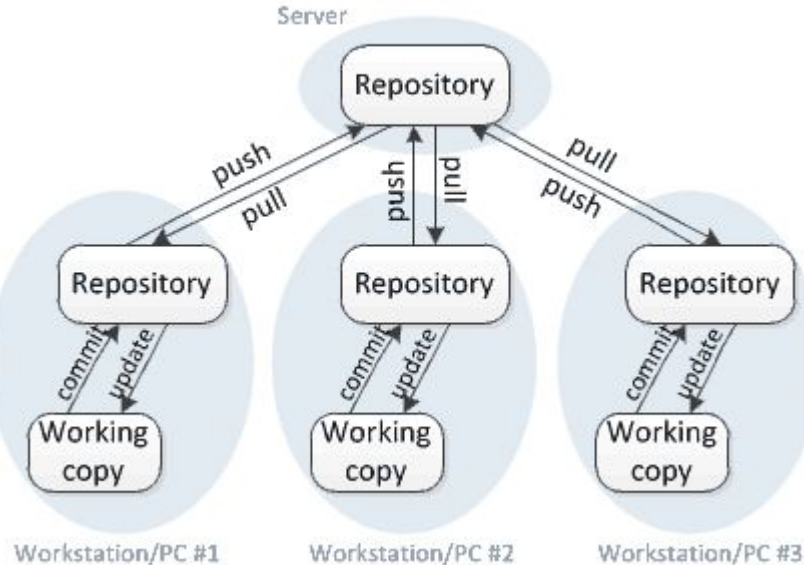
Before continuing, you may run **git status** and read what it says.



Git is a
Distributed Revision Control System

OpenGenus

Distributed version control



Revert time exercise.

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

- write a python function that find a missing number in a list

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

- write a python function that find missing numbers in a list

Results:

Step 3: Checkout to step one and change file name.

Master with MissNum.py and MissNum_2.py

- Check out Step 1
- Change the file name of MissNum.py to MissNum_2.py or cp.

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

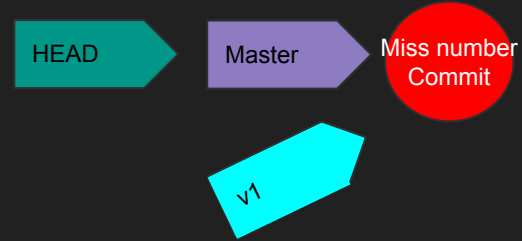
- Make a new branch with name: reverttime

Step 5: Merge branches

- Merge reverttime with master

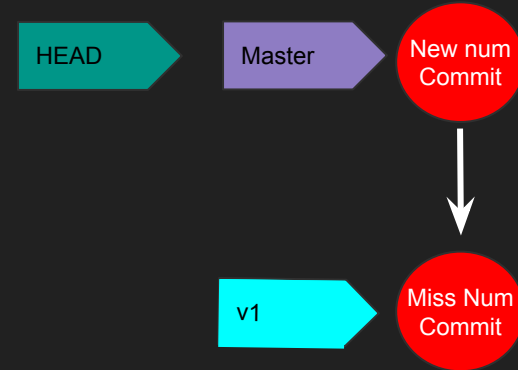
Miss number code and make a tag

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



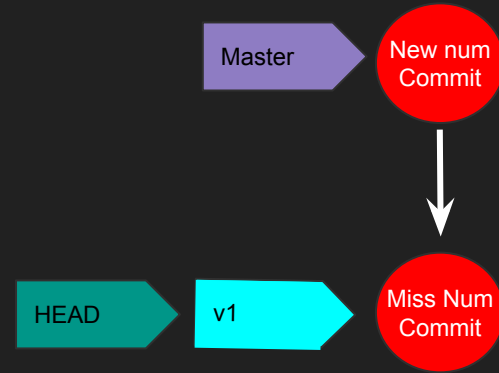
Update the function

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



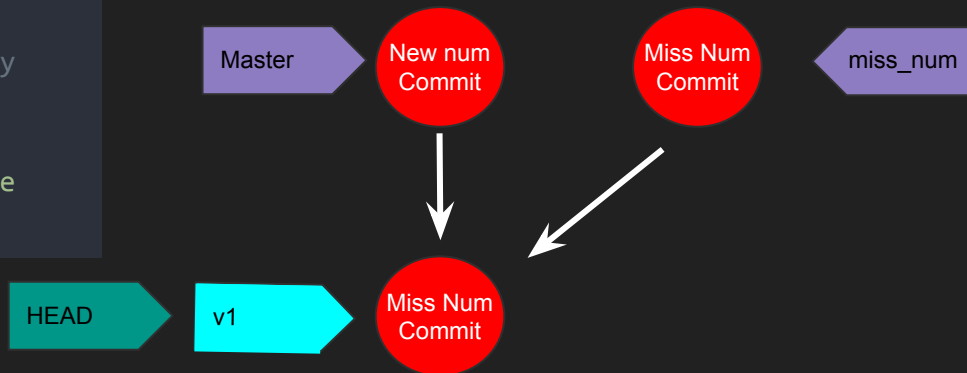
Checkout old tag, change name, create branch

```
git checkout v1
```



Create a new branch

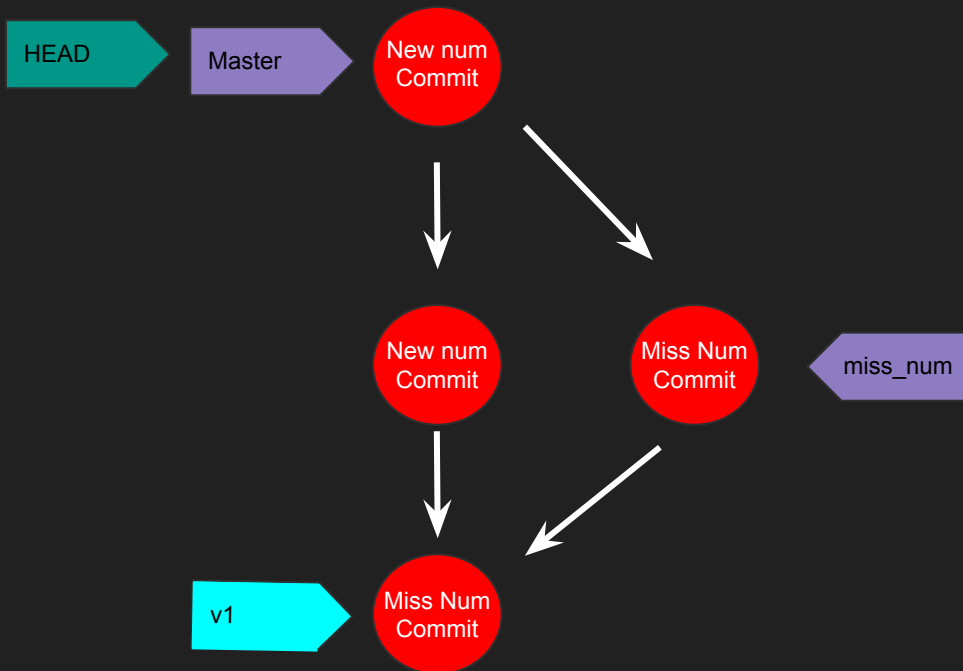
```
git branch miss_num
git checkout miss_num
mv miss_num.py miss_num_v1.py
atom miss_num_v1.py # add some comments in the .py
file
git add miss_m_v1.py
git commit -m "Missing num version only detect one
missing number"
```



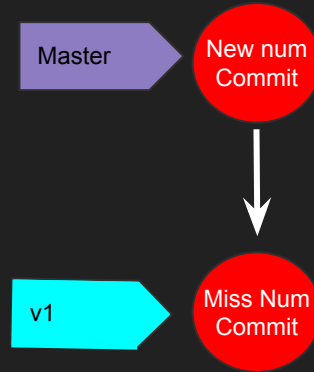
Merge

```
git checkout master
git merge master miss_num

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

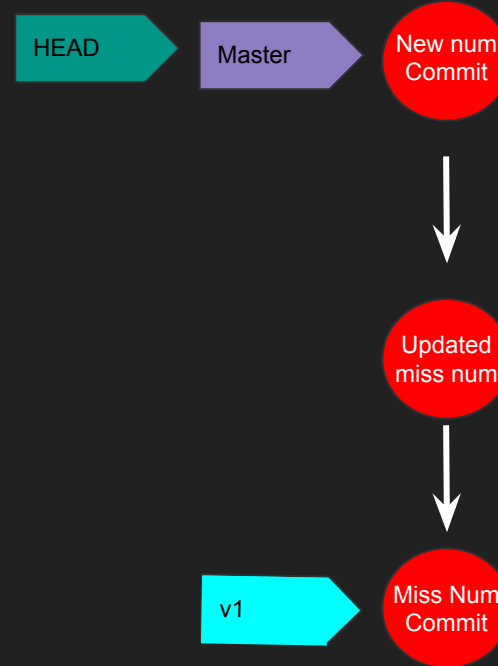


Solution 2



git checkout only one files

```
# rename current file name so no conflict
mv miss_num.py miss_num_v2.py
# retrieve one file from earlier version
# git checkout HEAD <file path>
git checkout v1 miss_num.py
git add .
git commit -m "get v1 version of miss_num in
current repo"
```



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.

Log reading

Look at Group one and two logs

```
git log --graph --decorate --oneline
```

git status



git log



Work on Branching

Try to work 8 exercises on git remote (1: Clone Intro)

<https://learngitbranching.js.org/>



The solution to mass Data!!