WEEK 3

MACHINE LEARNING OPERATIONS

UNLIMITED

Presented by **Asst. Prof. Dr. Tuchsanai Ploysuwan**

Week 3
Branches and Working with Others

Week 3 Branches and Working with Others

- **Week 3 Topics:**
  - Master/Main Branch and Branches
  - Understanding HEAD
  - Git Branch Commands:
    - **git branch**, **git switch**, **git checkout**
  - Delete or Rename Branch
  - Merging Branches and Conflicts
  - Exercise and Solution

Week 3
Branches

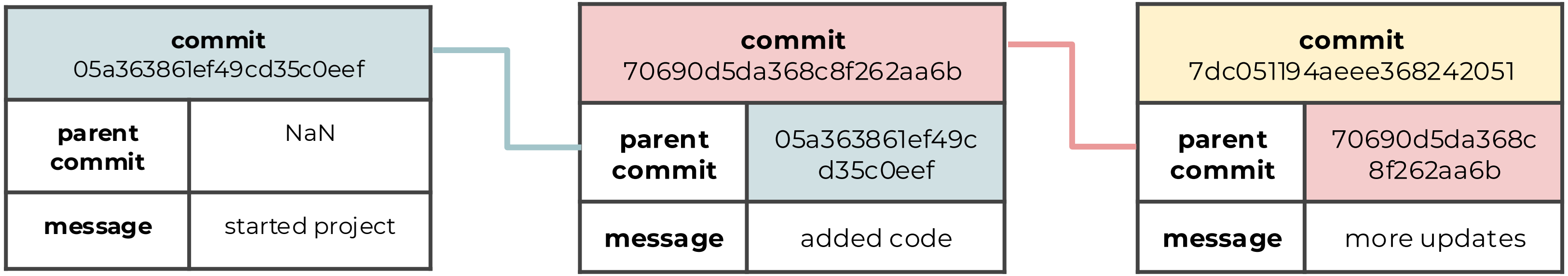
Week 3 Branches and Working with Others

- **Commit Process**
  - As we create commits, we are linking to a parent commit, showing the log of the commit history.

commit 05a363861ef49cd35c0eef	
parent commit	NaN
message	started project

Week 3 Branches and Working with Others

- **Commit Process**
  - As we create commits, we are linking to a parent commit, showing the log of the commit history.



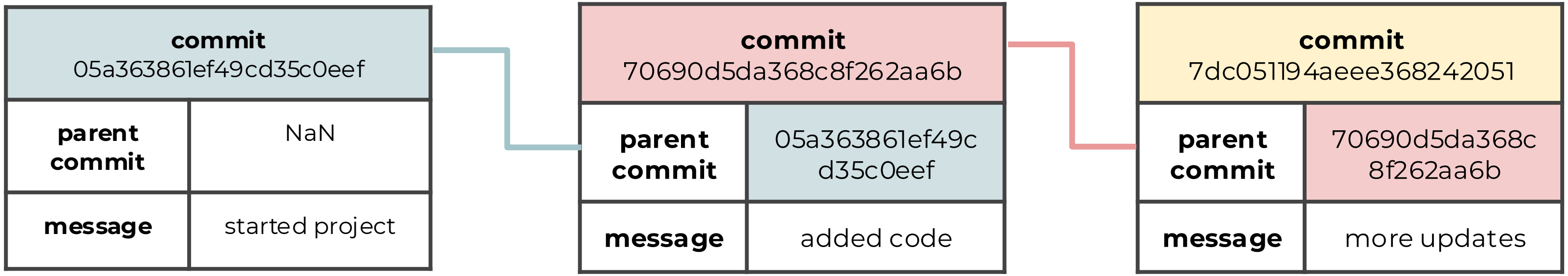
commit 05a363861ef49cd35c0eef	
parent commit	NaN
message	started project

commit 70690d5da368c8f262aa6b	
parent commit	05a363861ef49cd35c0eef
message	added code

commit 7dc051194aeee368242051	
parent commit	70690d5da368c8f262aa6b
message	more updates

Week 3 Branches and Working with Others

- **Commit Process**
  - As we need incorporate the workflows of others or be able to focus on new updates without breaking old code, we need **branches**.



commit 05a363861ef49cd35c0eef	
parent commit	NaN
message	started project

commit 70690d5da368c8f262aa6b	
parent commit	05a363861ef49cd35c0eef
message	added code

commit 7dc051194aeee368242051	
parent commit	70690d5da368c8f262aa6b
message	more updates

Week 3 Branches and Working with Others

- **Branches**
  - A branch represents an independent line of development.
  - Branches serve as an abstraction for the edit/stage/commit process.
  - They are a way to request a brand new working directory, staging area, and project history.

Week 3 Branches and Working with Others

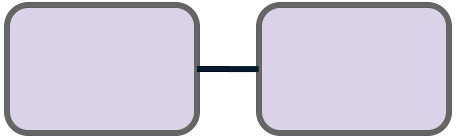
- **Branches**
  - Branches are just pointers to commits.
  - When you create a branch, all Git needs to do is create a new pointer, it doesn’t change the repository in any other way.
  - Let’s explore why branches are useful for workflows...

Week 3 Branches and Working with Others

- **Branches**

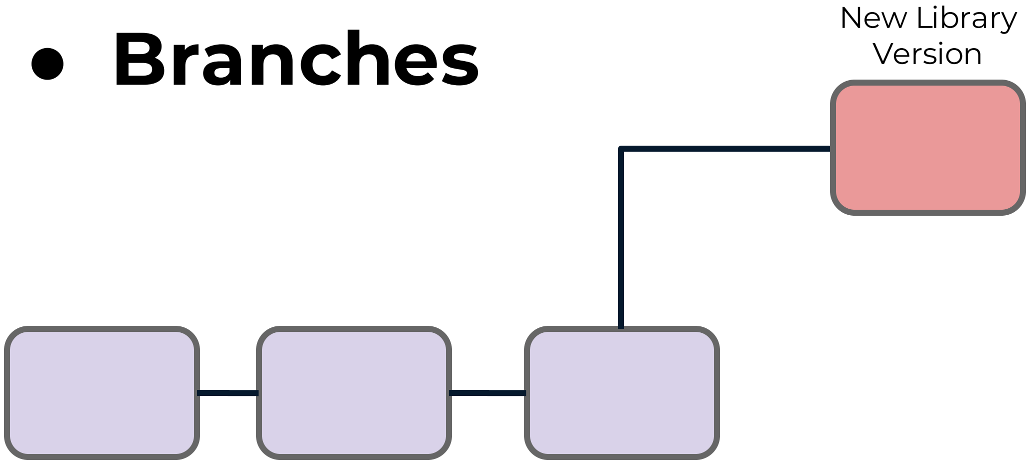
Week 3 Branches and Working with Others

- Branches



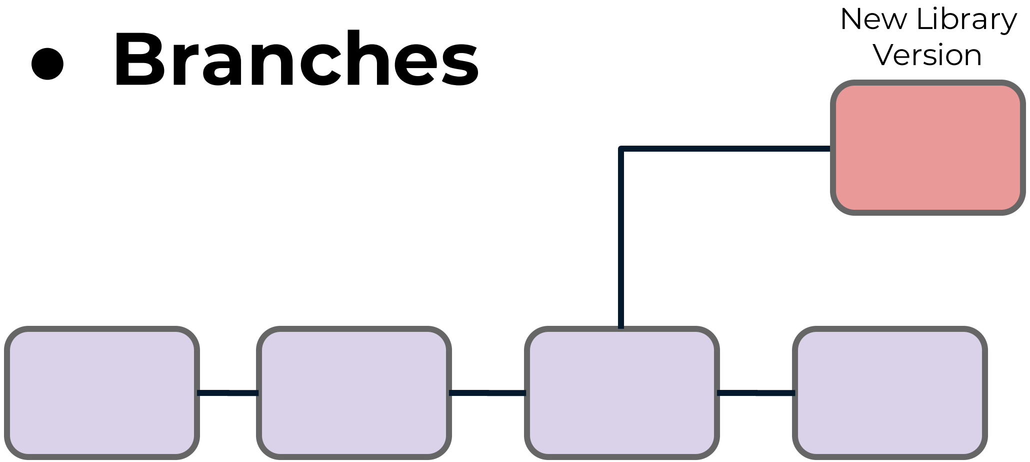
Week 3 Branches and Working with Others

- **Branches**



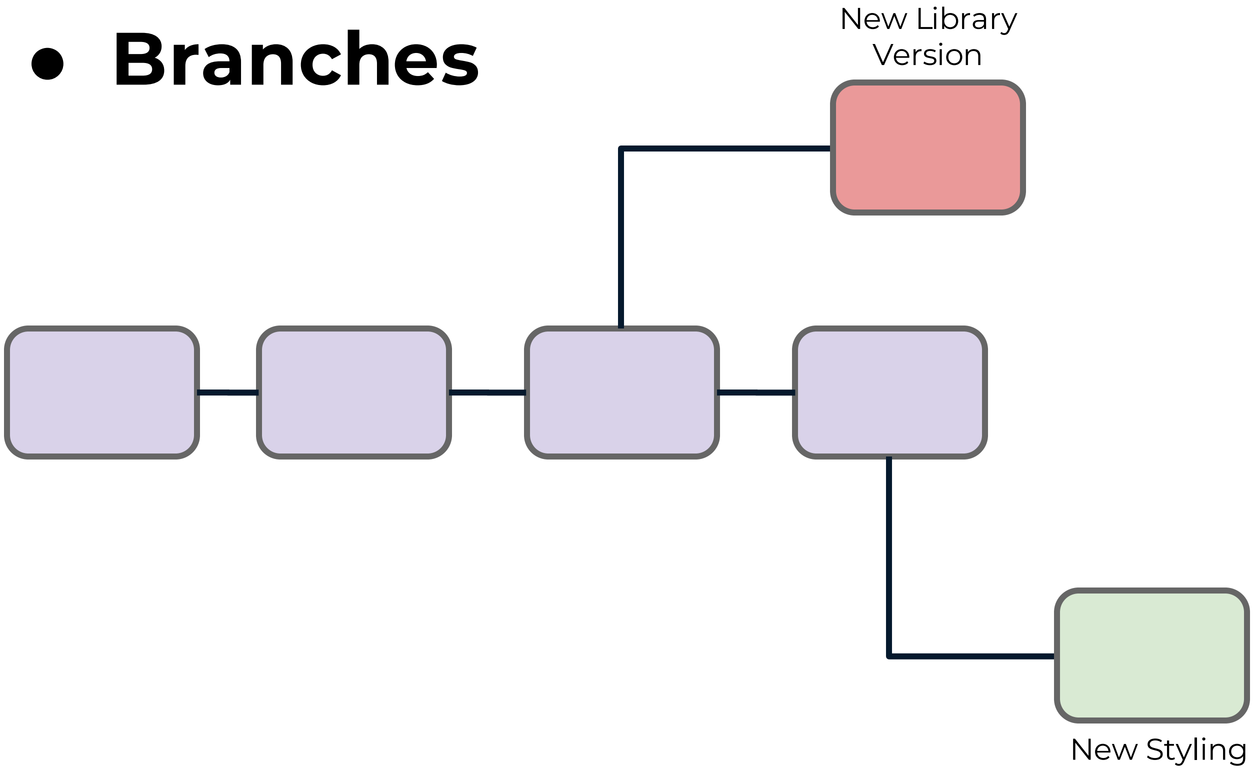
Week 3 Branches and Working with Others

- **Branches**



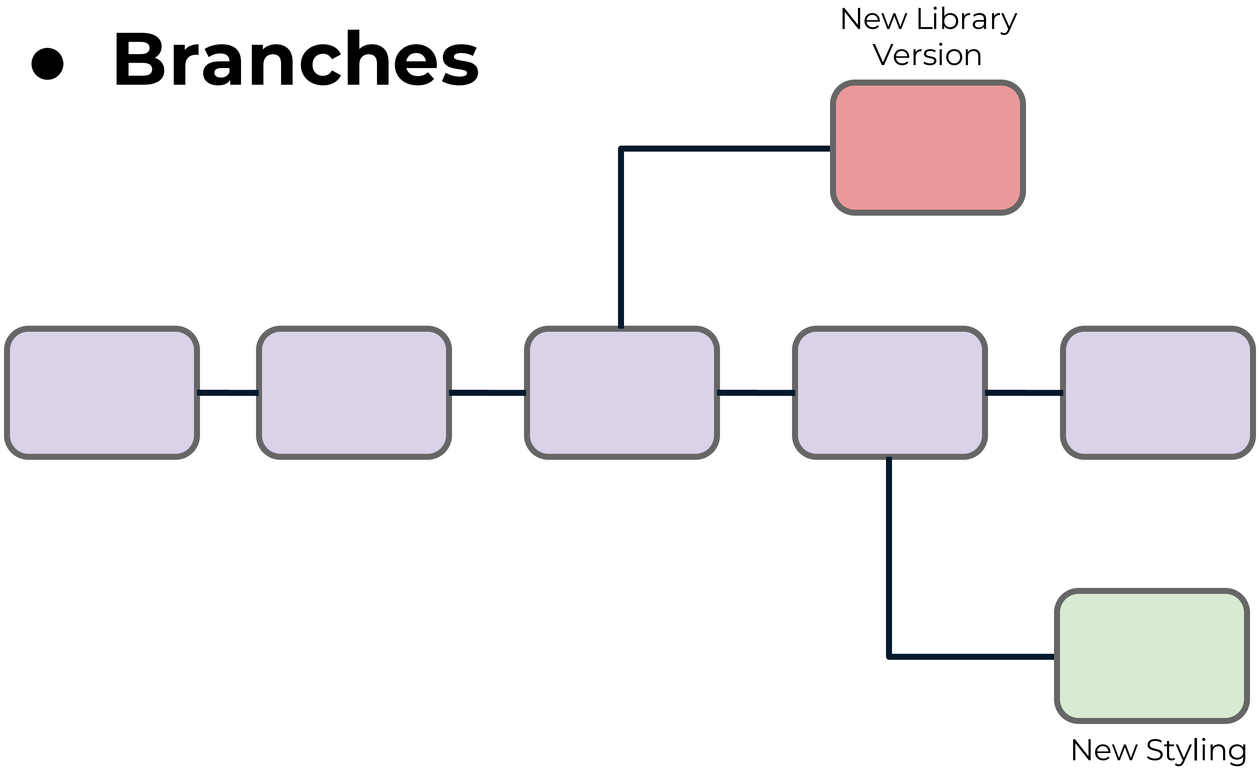
Week 3 Branches and Working with Others

- **Branches**



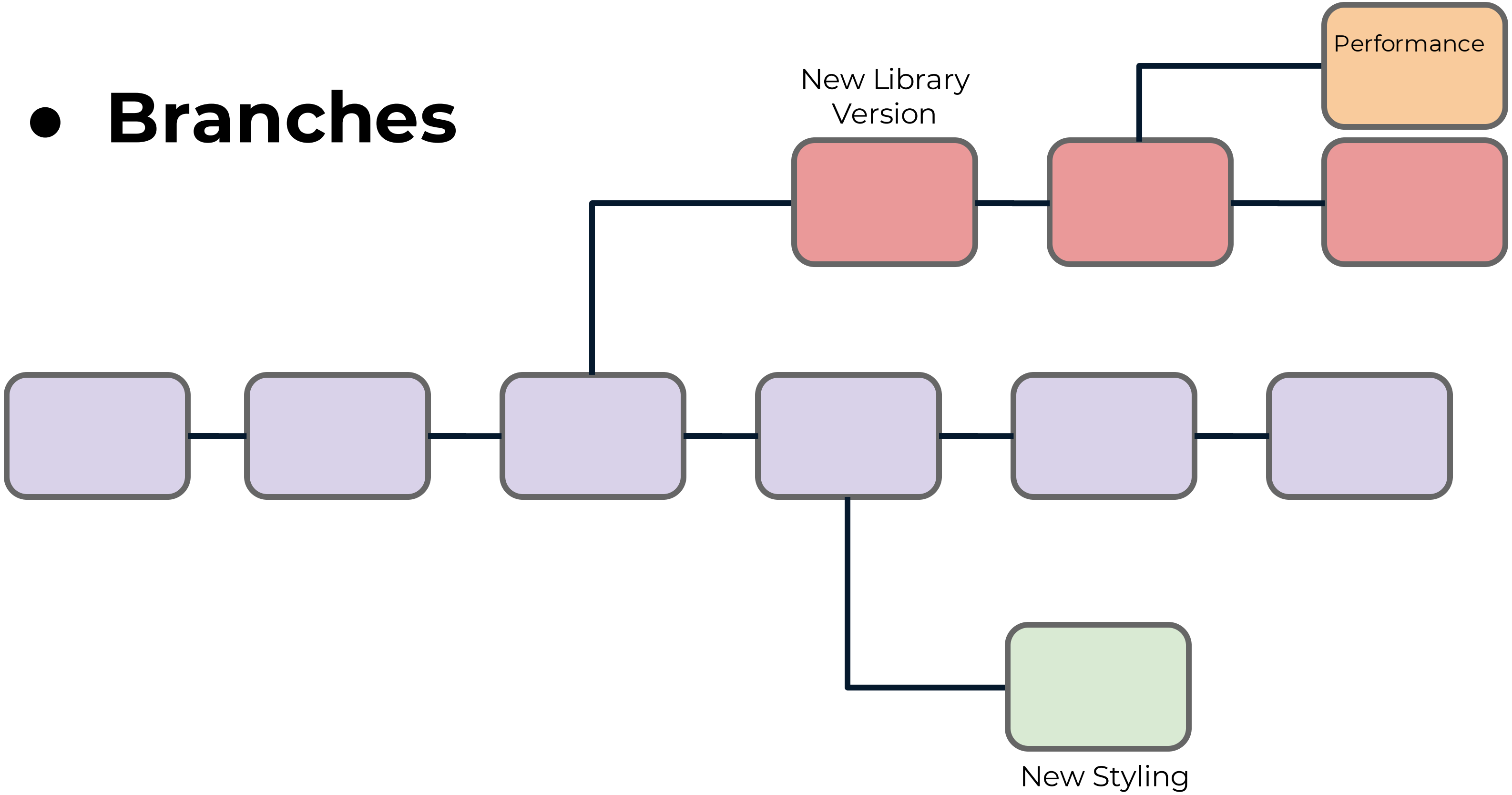
Week 3 Branches and Working with Others

- **Branches**



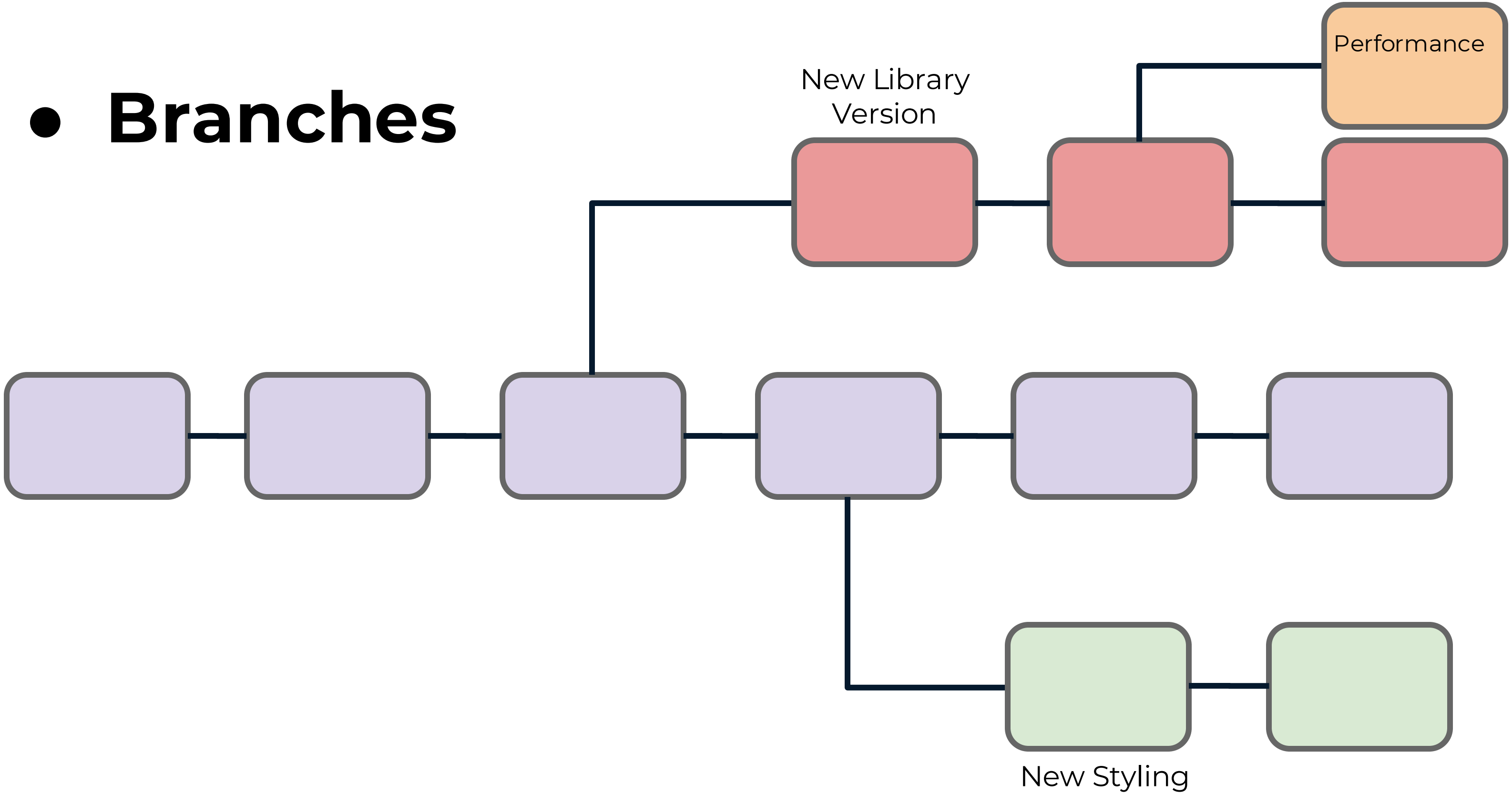
Week 3 Branches and Working with Others

- **Branches**



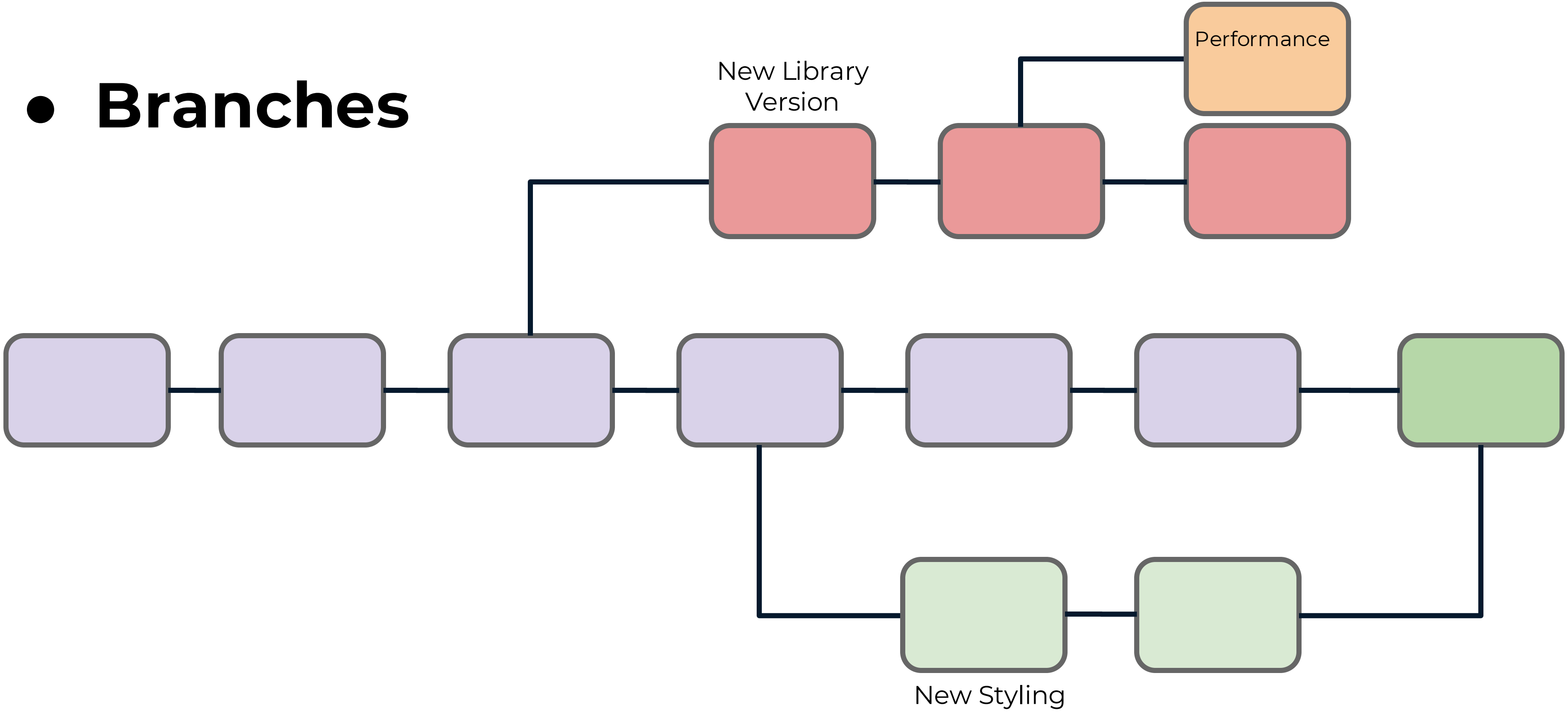
Week 3 Branches and Working with Others

- **Branches**



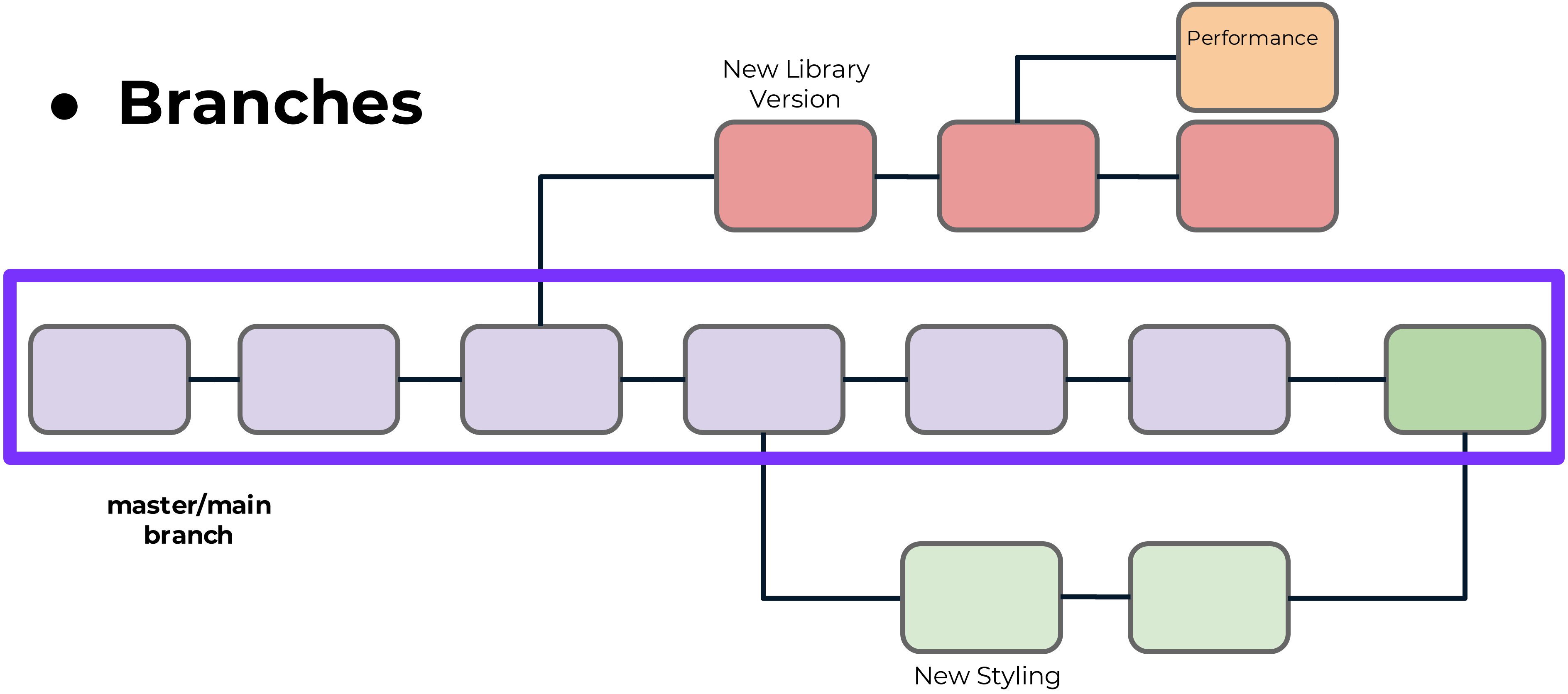
Week 3 Branches and Working with Others

- **Branches**



Week 3 Branches and Working with Others

- **Branches**

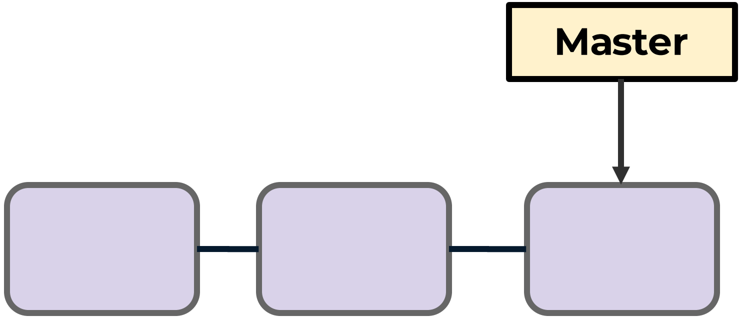


Week 3 Branches and Working with Others

- **Creating a New Branch**
  - Before we conclude, let’s quickly go into more detail about what happens when first create a new branch.
  - Branches are just pointers to commits.
  - When you create a branch, all Git needs to do is create a new pointer, it doesn’t change the repository in any other way.

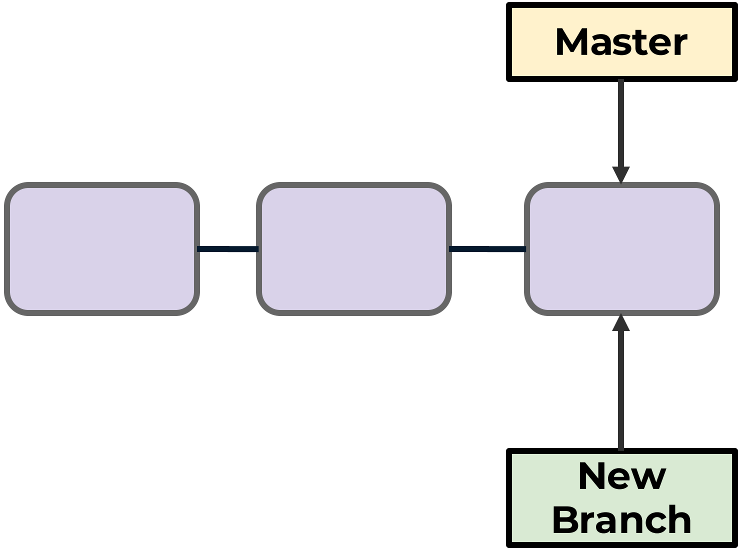
Week 3 Branches and Working with Others

- **Creating a New Branch**



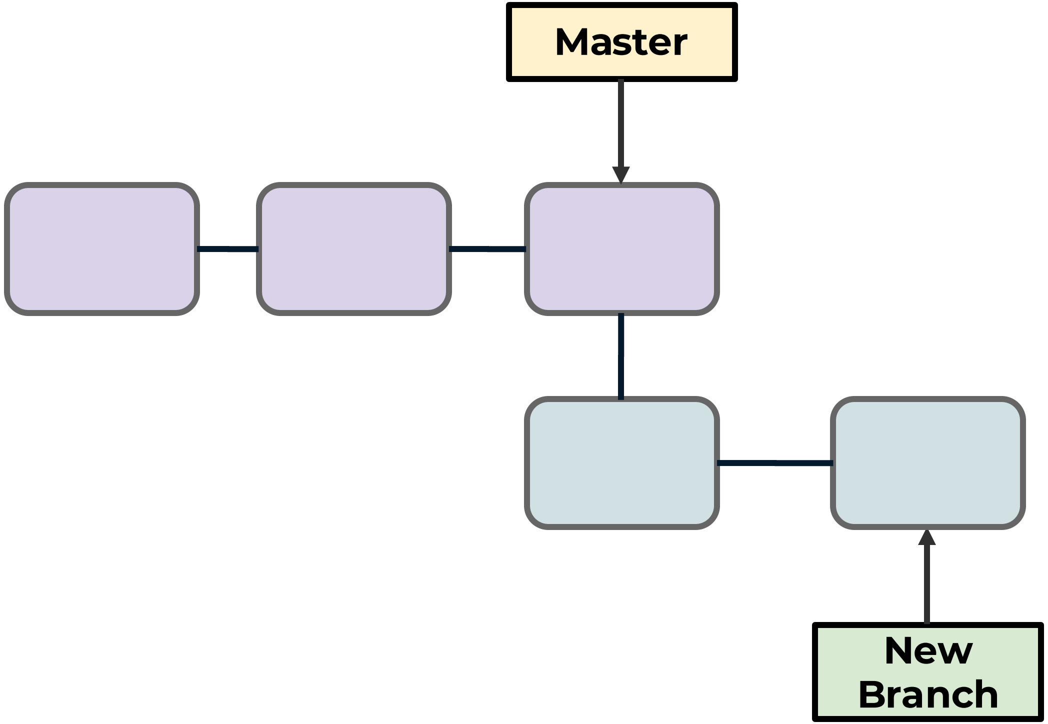
Week 3 Branches and Working with Others

- **Creating a New Branch**



Week 3 Branches and Working with Others

- **Creating a New Branch**



Week 3 Branches and Working with Others

- Now that we’ve seen how branches point to commits, we need to learn about HEAD.
- HEAD will help us understand what we are currently “viewing” or where we are “located” in regards to branches and commits.

Week 3 Branches and Working with Others

- **Up Next:**
    - We’ll explore and visualize specific actions and commands related to branches, including **HEAD, git checkout, git branch, git switch**, and more.

Week 3
Understanding HEAD

Week 3 Branches and Working with Others

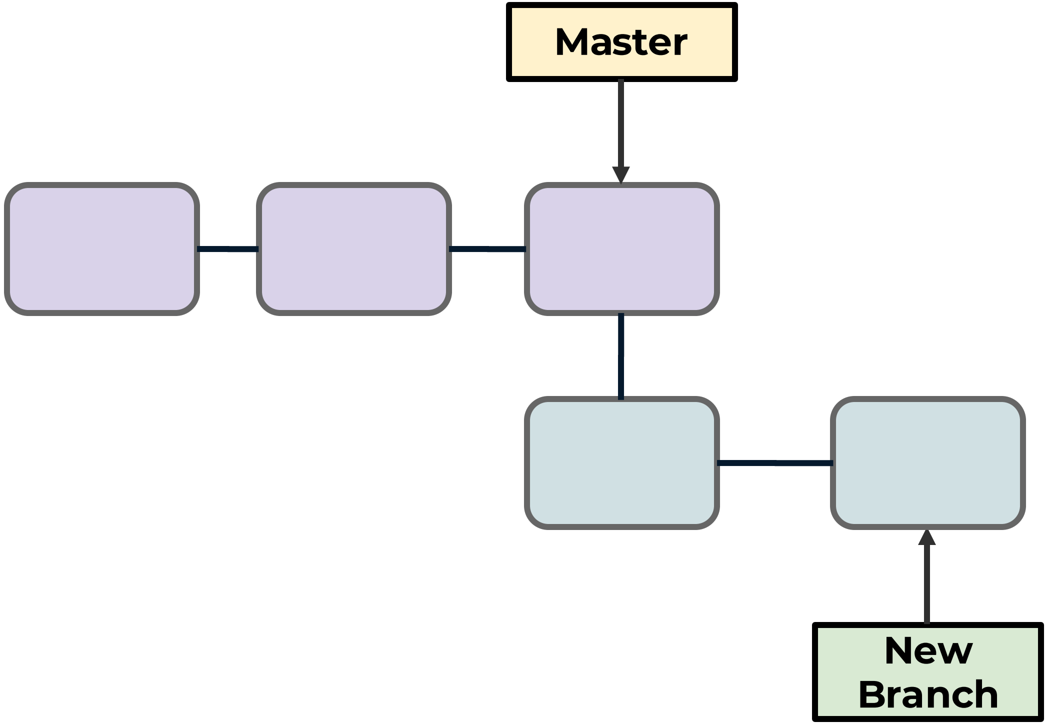
- **Branches**
  - As we work more with branches, you will probably notice a term show up during your commits: **HEAD**.
  - When viewing the most recent commit using **git log** you may see:
    - **commit 05as..3e2 (HEAD -> master)**

Week 3 Branches and Working with Others

- **HEAD**
  - In all of our examples so far, HEAD has always been pointing to the most recent commit in the master branch.
    - **HEAD -> master**

Week 3 Branches and Working with Others

- **Recall we have branch points (references)**



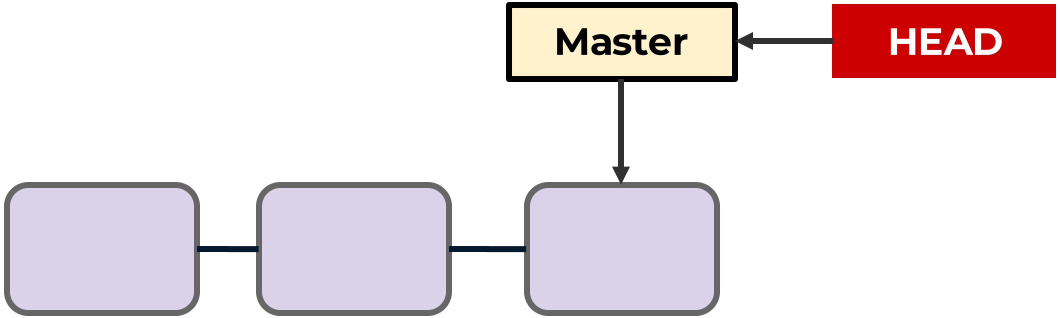
Week 3 Branches and Working with Others

- **Branches and Commits**
  - Git stores a branch as a reference to a commit.
  - In this sense, a branch represents the tip of a series of commits—it's not a container for commits.
  - The history for a branch is extrapolated through the commit relationships.

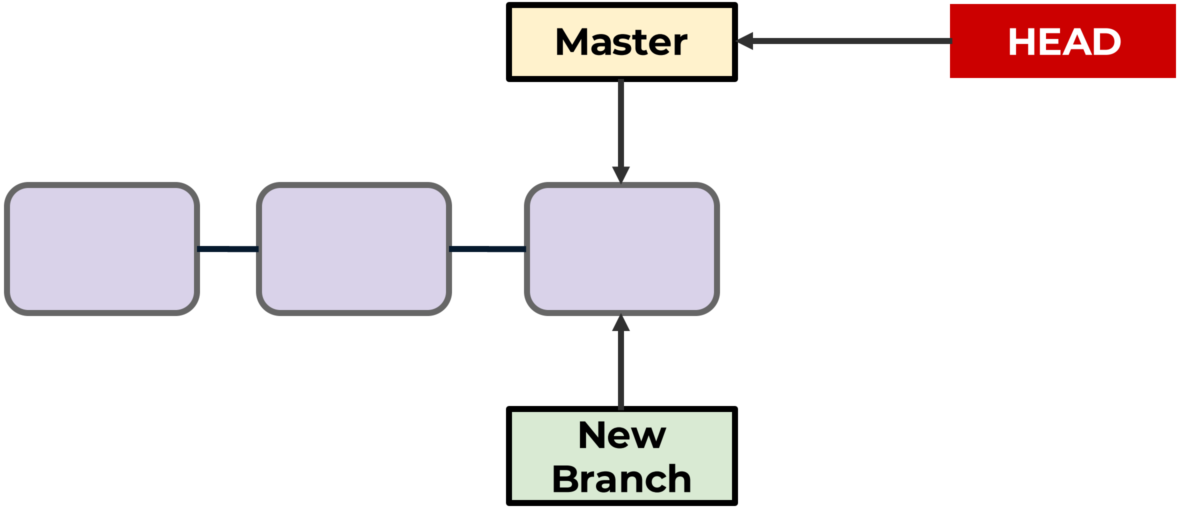
Week 3 Branches and Working with Others

- **HEAD**
  - A HEAD is simply a reference to a commit object.
  - We can think of HEAD as pointing to a specific commit in a branch that we are currently viewing.

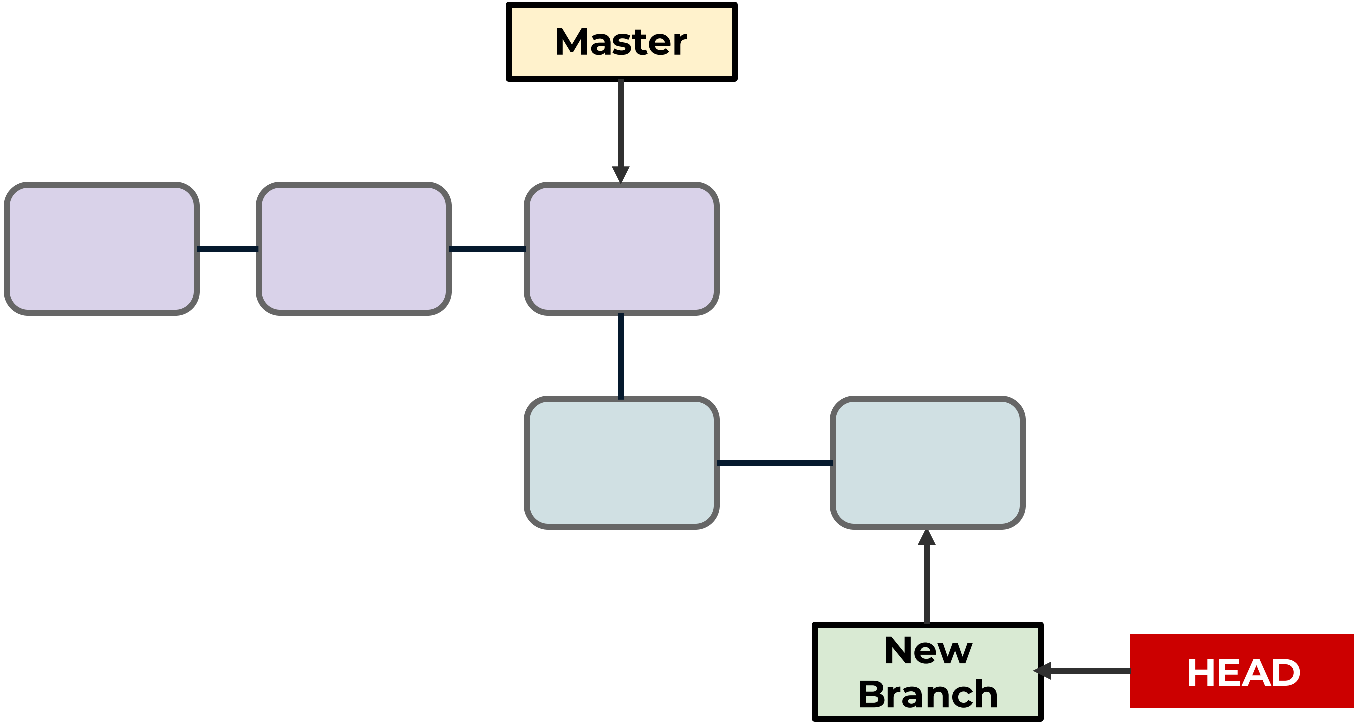
Week 3 Branches and Working with Others



Week 3 Branches and Working with Others



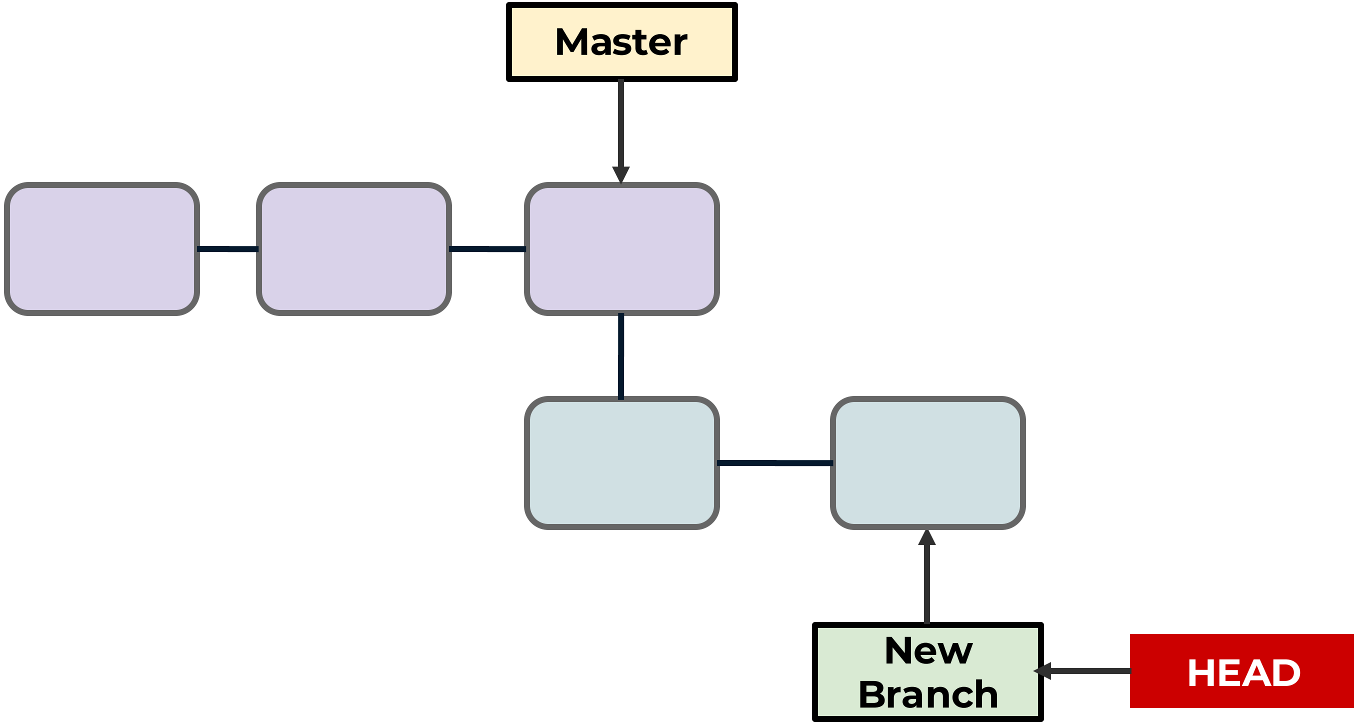
Week 3 Branches and Working with Others



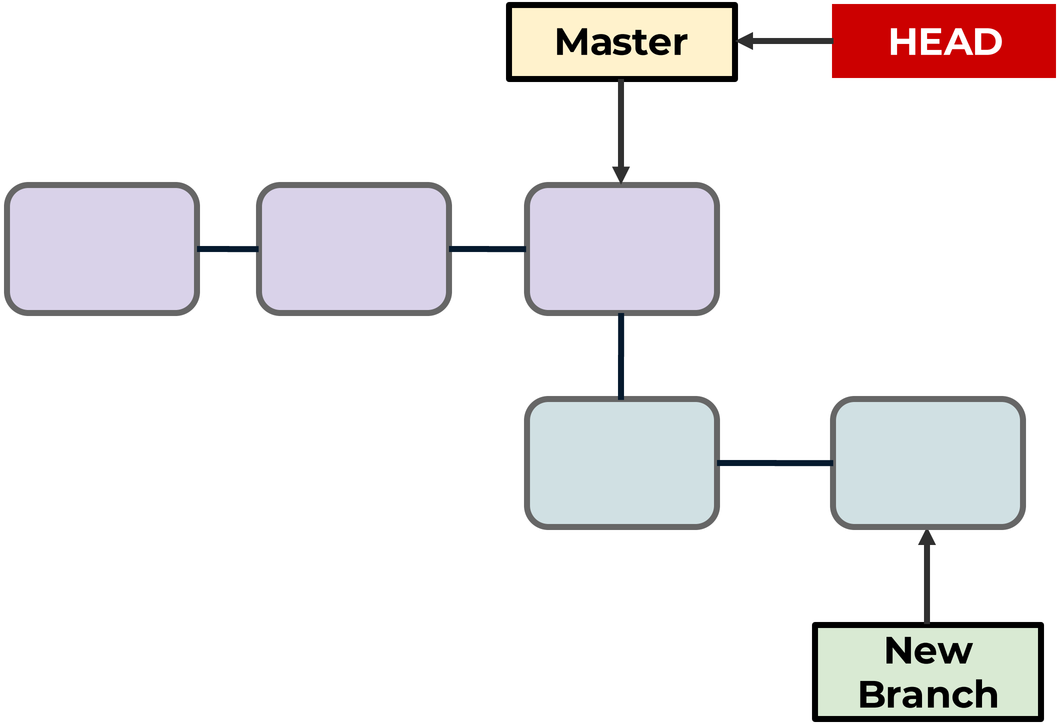
Week 3 Branches and Working with Others

- We can think of these branches as just references to a commit.
- Using HEAD tells us which branch reference we are currently “checking out”.
- We can always switch back out HEAD to some other branch (which is a pointer to a commit reference).

Week 3 Branches and Working with Others



Week 3 Branches and Working with Others



Master

HEAD

New Branch

Week 3 Branches and Working with Others

- **Up Next:**
  - Now that we understand the theory behind branches and HEAD, let’s begin to explore the actually commands that let us create branches and navigate between them.

Week 3
Git Branch Commands

Week 3 Branches and Working with Others

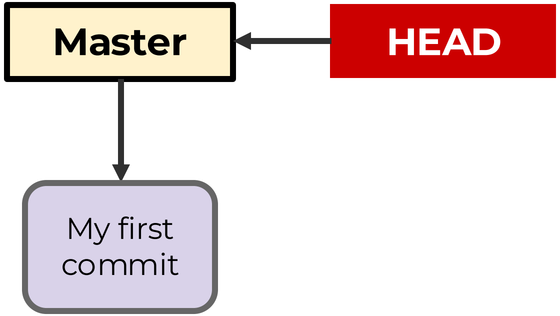
- **Git Branch Commands**
  - Create a New Repo
  - Add File
  - Create a New Branch
    - **git branch <branch_name>**
  - Report Branches
    - **git branch**
  - Switch Branches
    - **git switch or git checkout**

Week 3 Branches and Working with Others

- **Git Branch Commands**
  - Add and Commit Changes on New Branch
  - Use **git log** and **git switch** to explore differences between branches.

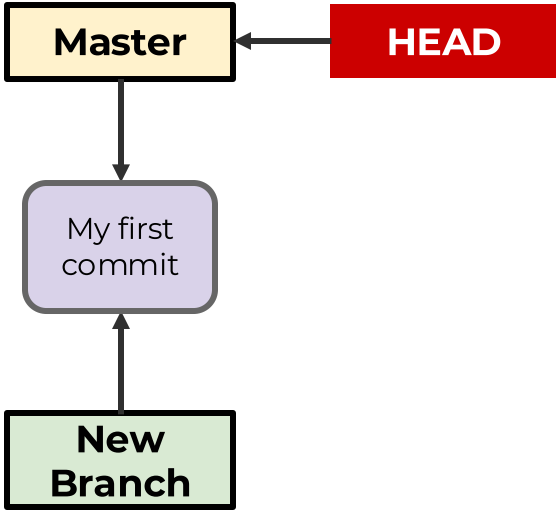
Week 3 Branches and Working with Others

- **git init, git add, git commit**



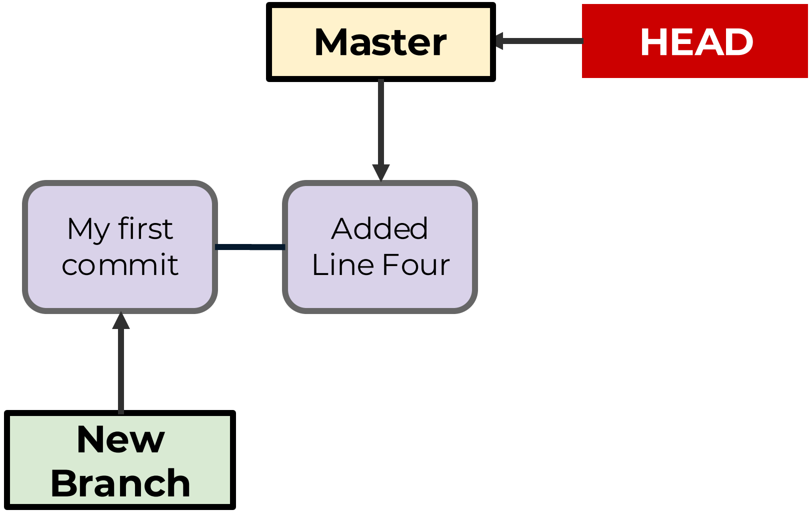
Week 3 Branches and Working with Others

- **git branch new_branch**

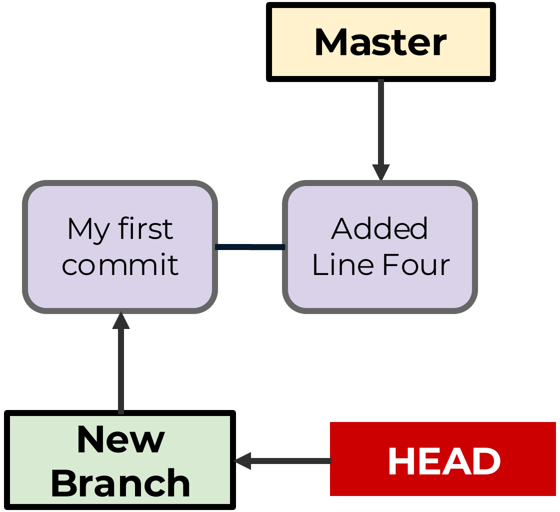


Week 3 Branches and Working with Others

- **git add, git commit, git log**



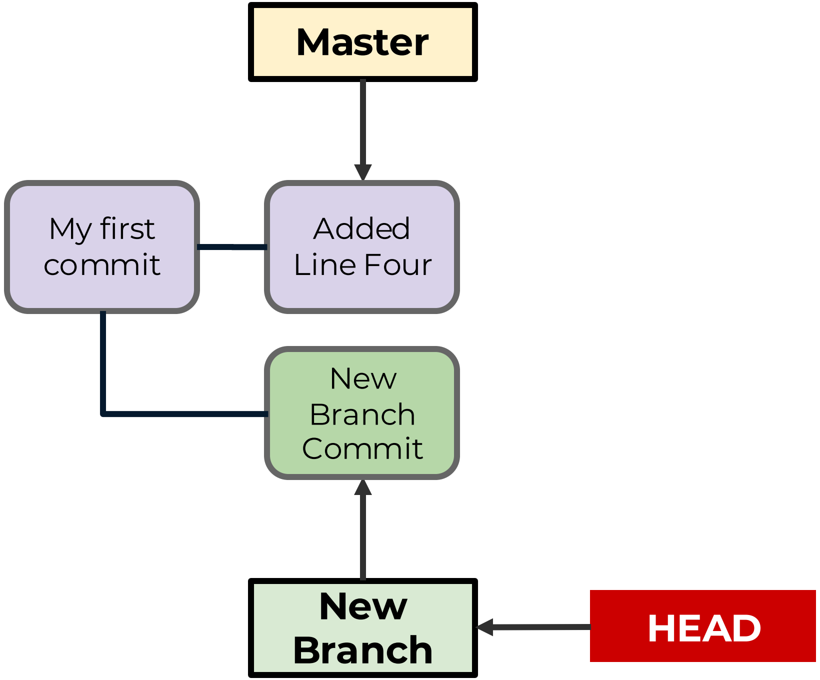
Week 3 Branches and Working with Others



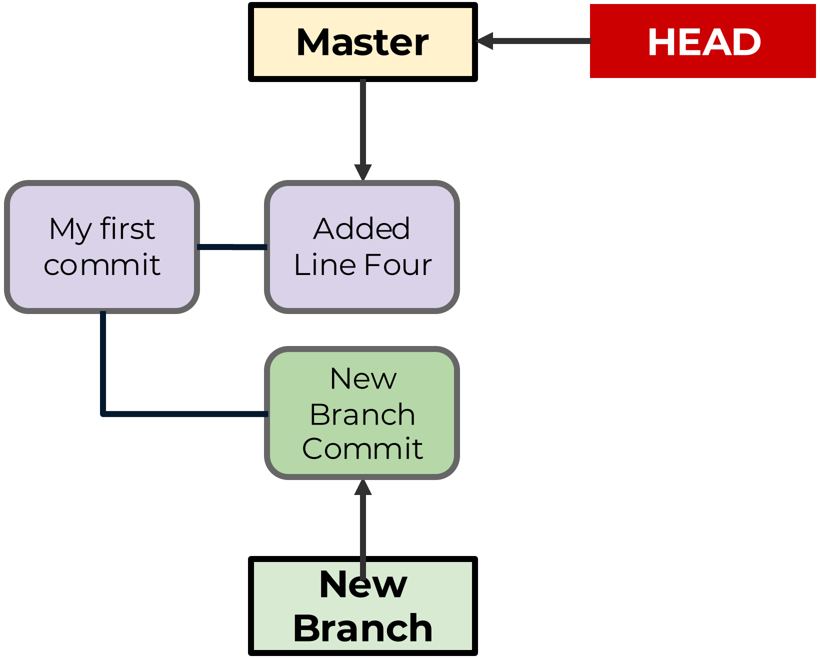
- git switch new_branch
or git checkout new_branch

Week 3 Branches and Working with Others

- **git add , git commit, git log**



Week 3 Branches and Working with Others



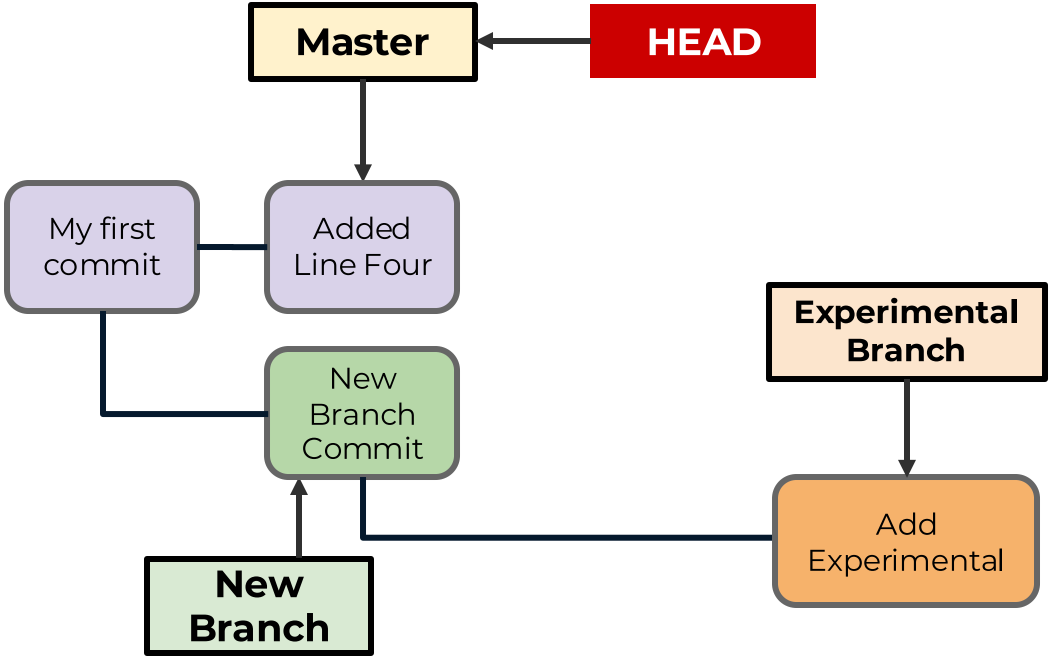
- **git switch master**

or git checkout master

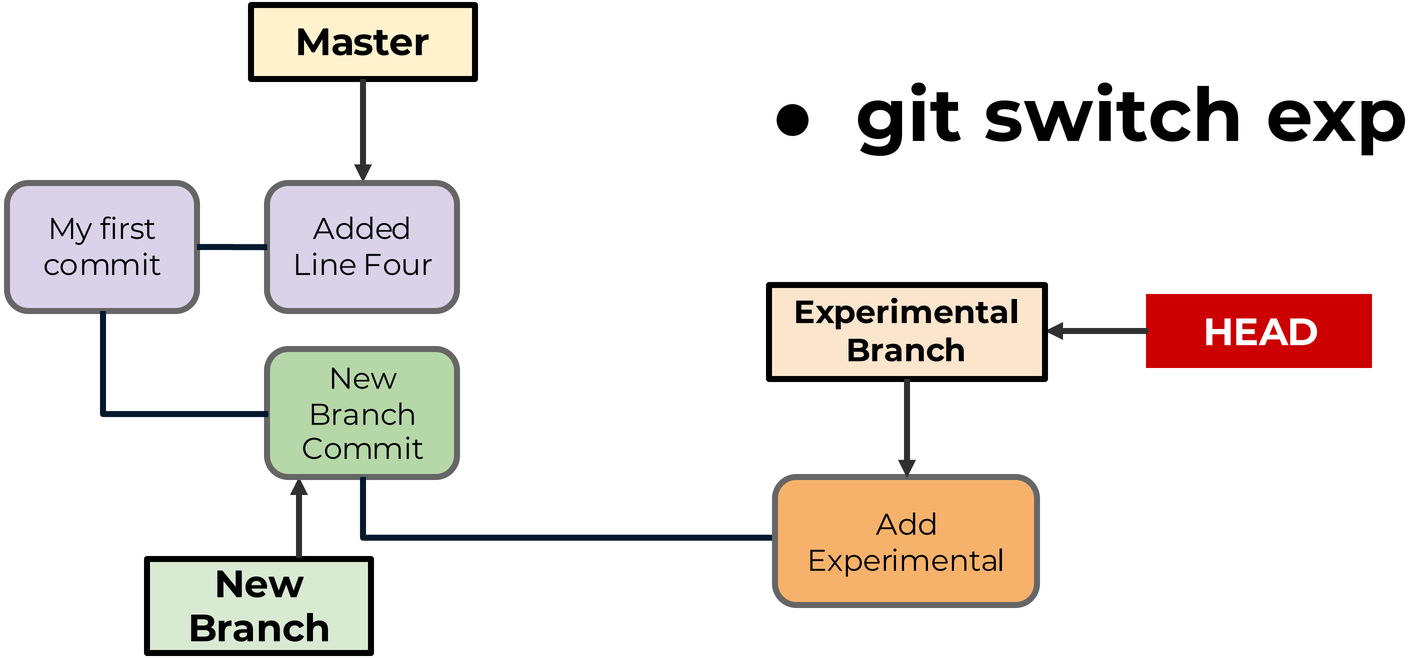
Week 3
Delete and Rename Branches

Week 3 Branches and Working with Others

- **Previously:**



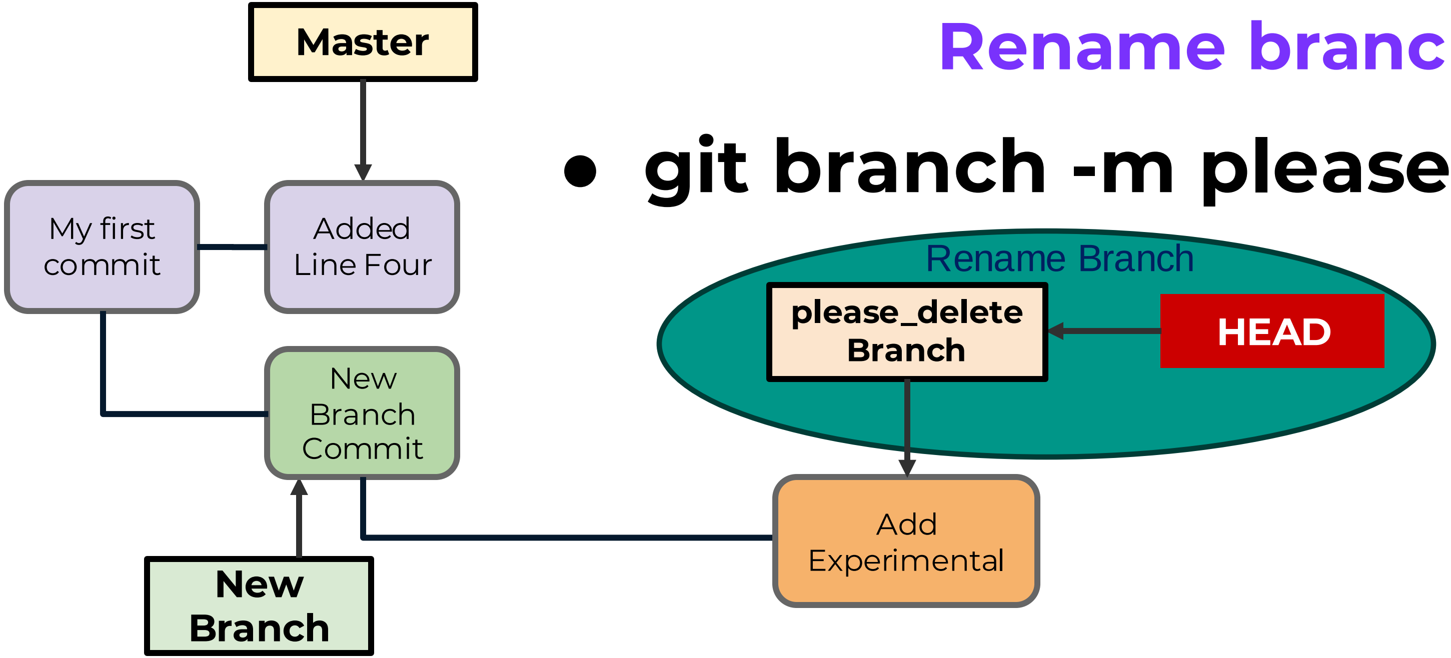
Week 3 Branches and Working with Others



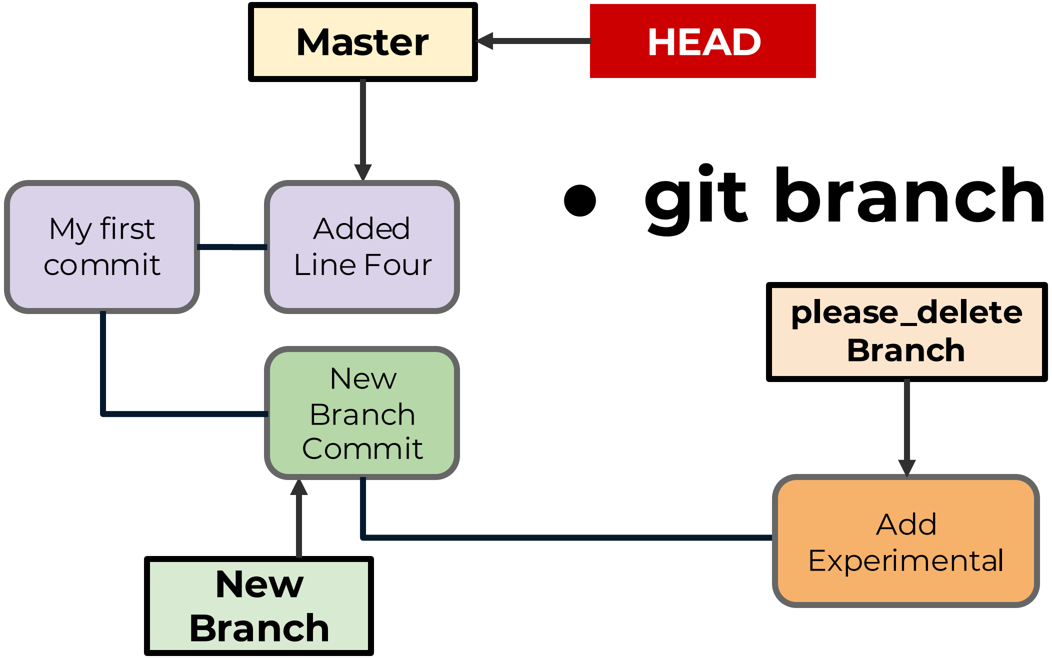
- **git switch experimental**

Rename branch

- git branch -m please_delete



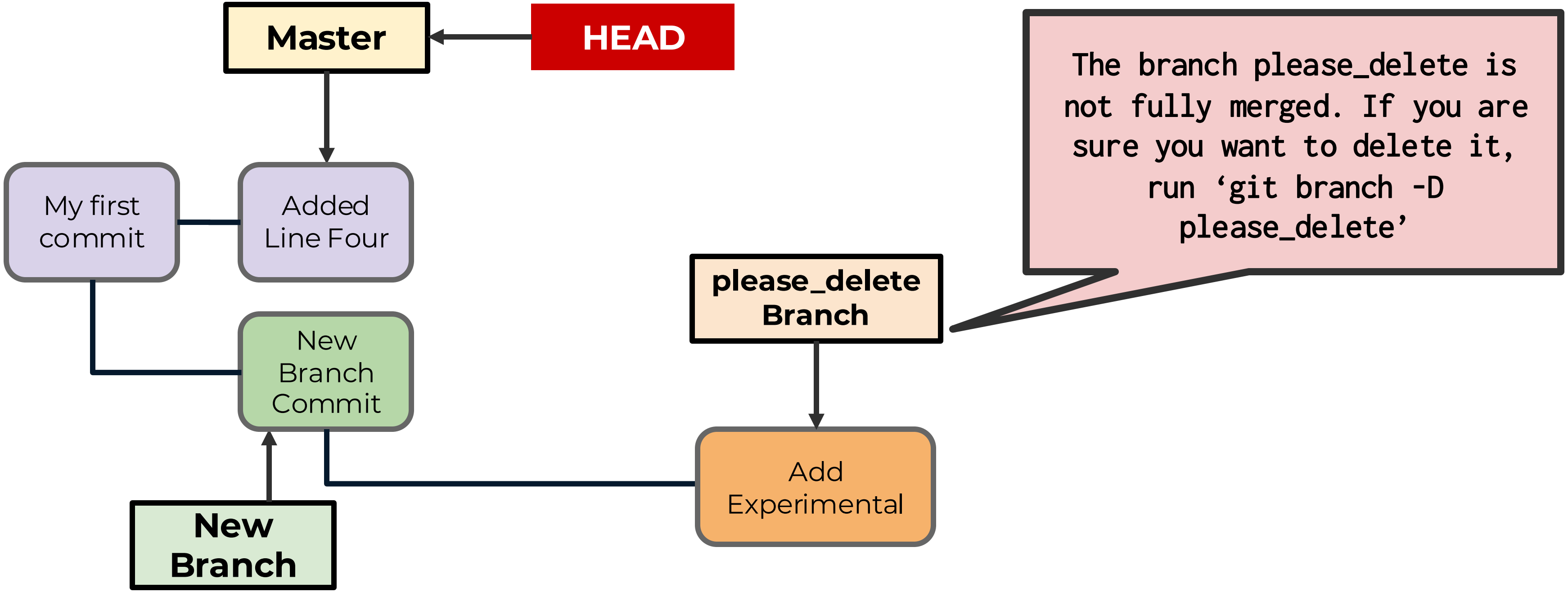
Week 3 Branches and Working with Others



- **git branch -d please_delete**

Week 3 Branches and Working with Others

- **git branch -d please_delete**



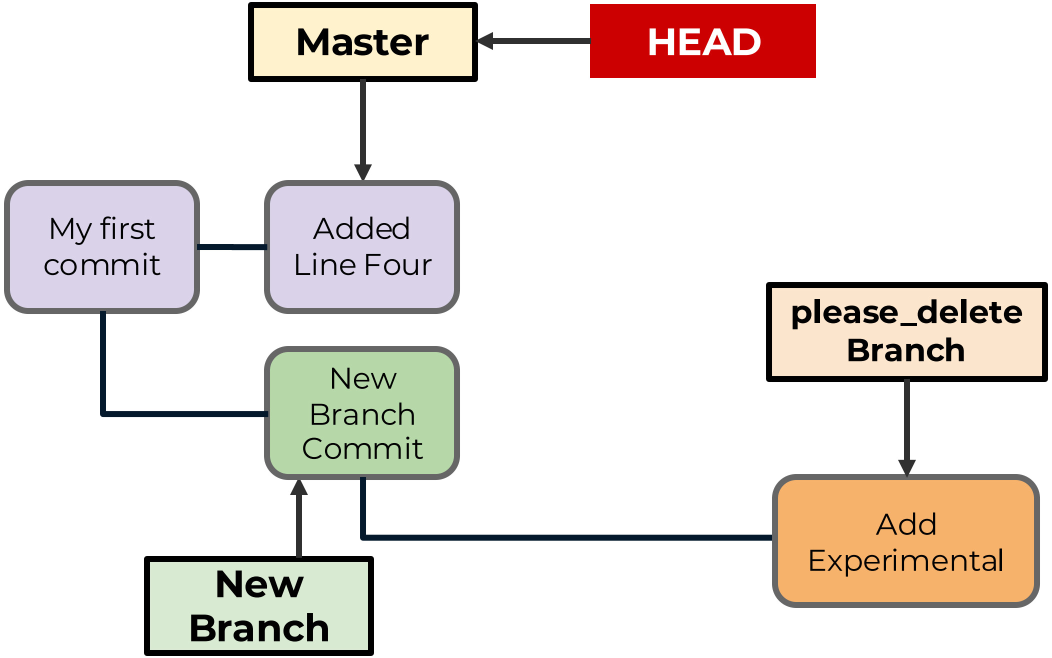
The branch please_delete is not fully merged. If you are sure you want to delete it, run 'git branch -D please_delete'

Week 3 Branches and Working with Others

- **Deleting a Branch**
    - **git branch -d branch_to_delete_name**
        - You can not delete a branch you are checked out at.
        - You also will get a warning if the branch is not merged.
            - You can confirm you want to do this anyways with **-D**

Week 3 Branches and Working with Others

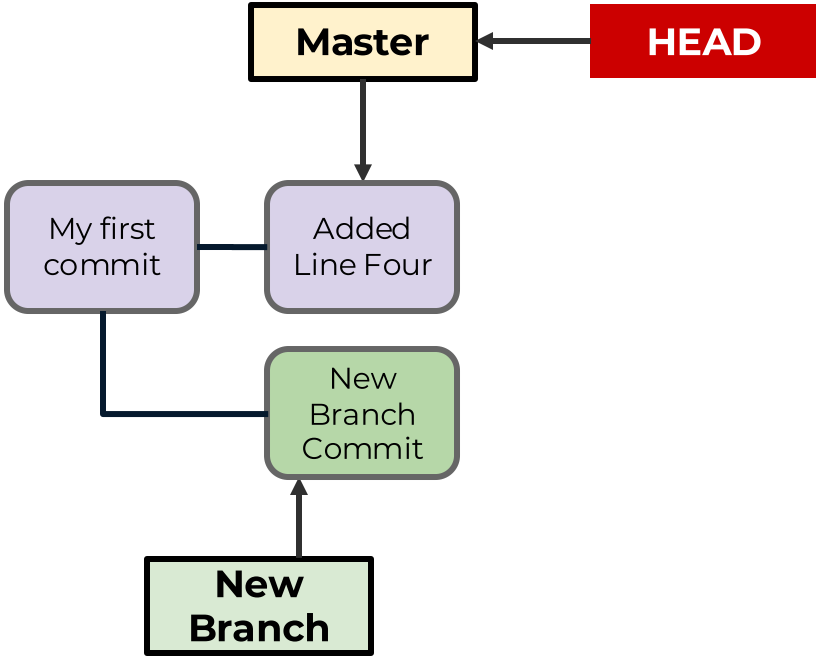
- **git branch -D please_delete**



The branch please_delete is not fully merged. If you are sure you want to delete it, run ‘git branch -D please_delete’

Week 3 Branches and Working with Others

- **git branch -D please_delete**



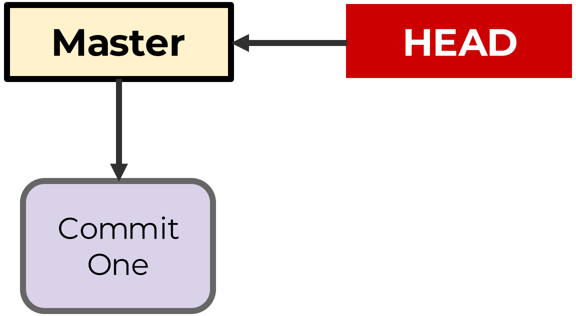
Week 3
Merging Branches and Conflicts

Week 3 Branches and Working with Others

- Now that we understand creating new branches, let’s shift focus to merging branches back together.
- Let’s explore a simple type of merge, where a new branch is created, but the original branch it stemmed from has no additional commits.
  - This is known as a “fast-forward” merge

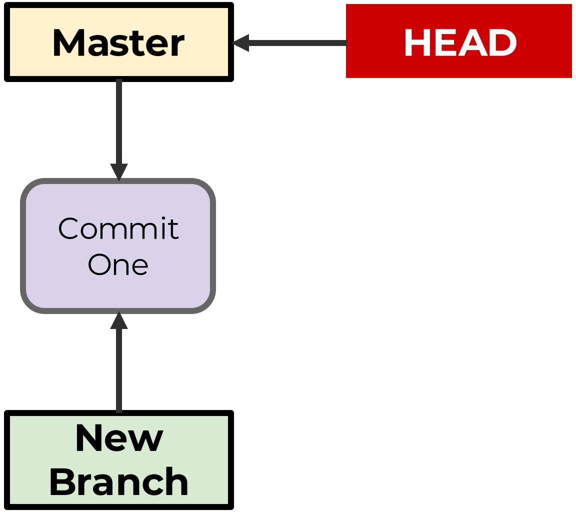
Week 3 Branches and Working with Others

- “Fast Forward” Merge



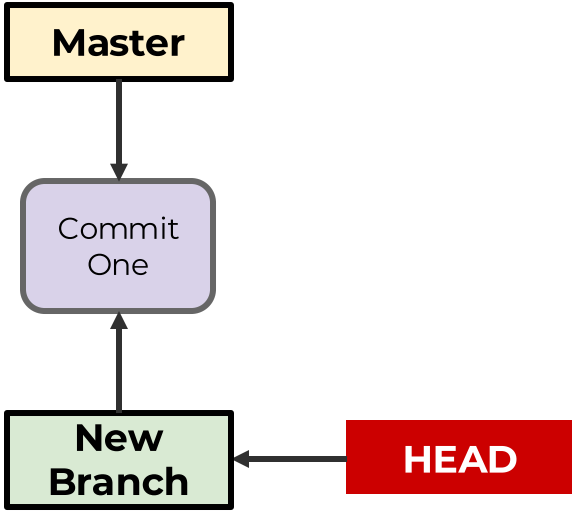
Week 3 Branches and Working with Others

- “Fast Forward” Merge



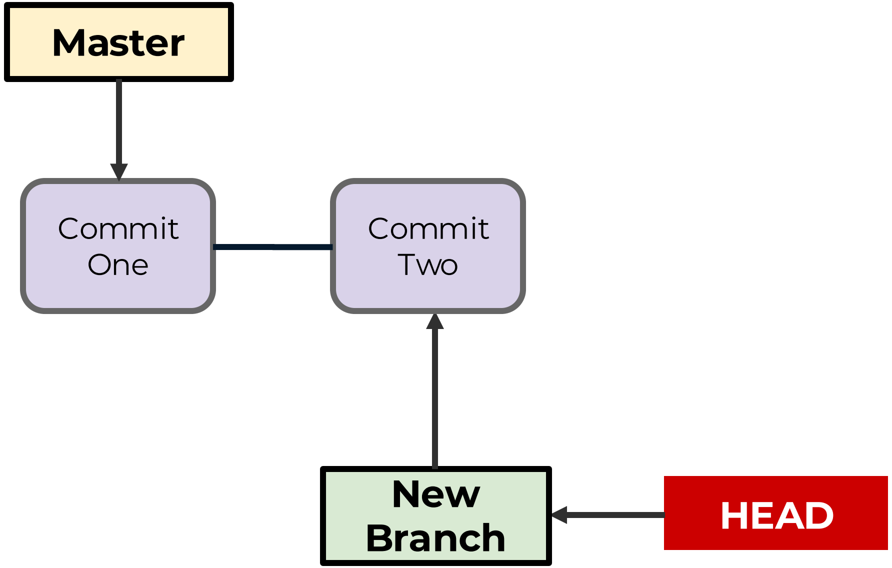
Week 3 Branches and Working with Others

- “Fast Forward” Merge



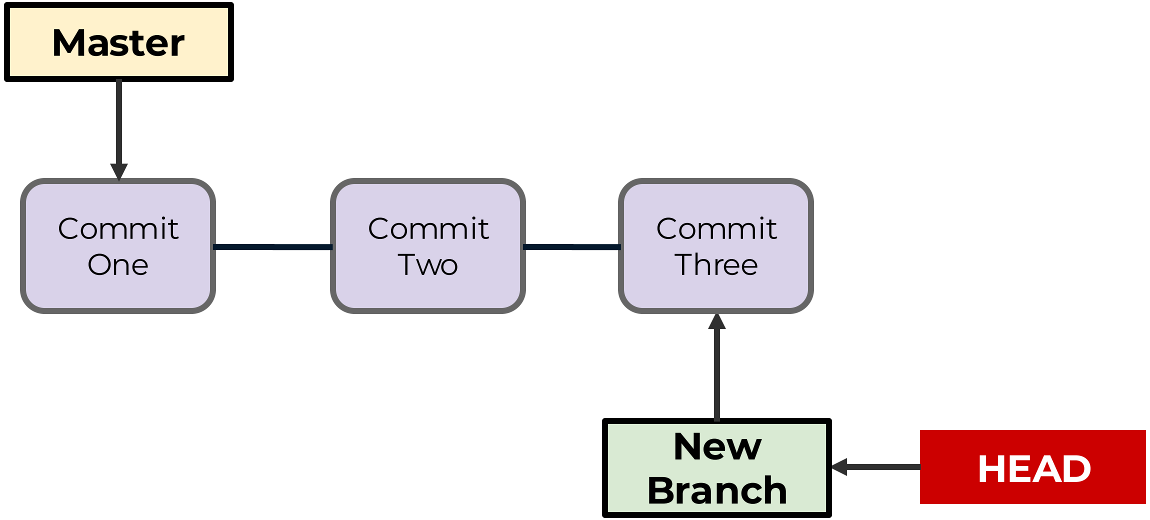
Week 3 Branches and Working with Others

- “Fast Forward” Merge



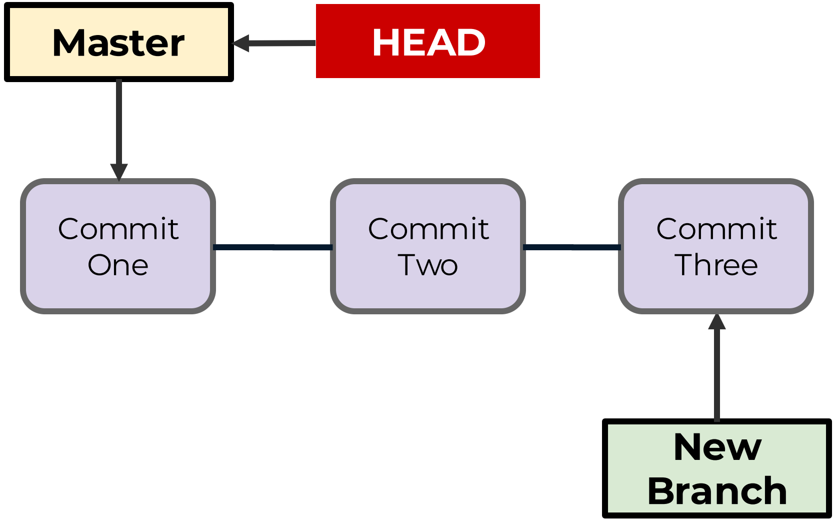
Week 3 Branches and Working with Others

- “Fast Forward” Merge



Week 3 Branches and Working with Others

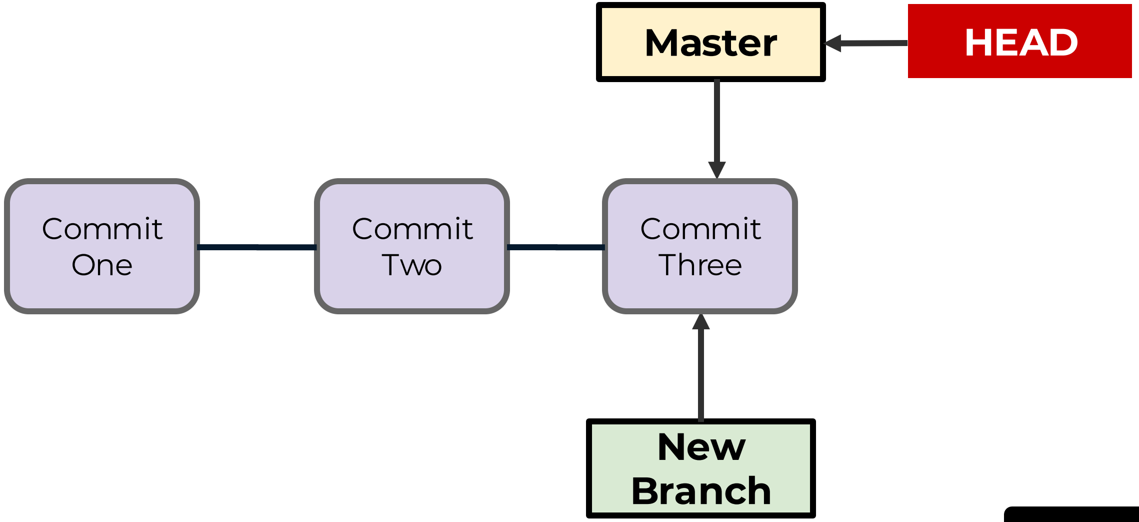
- “Fast Forward” Merge



```
>> git switch master
```

Week 3 Branches and Working with Others

- “Fast Forward” Merge



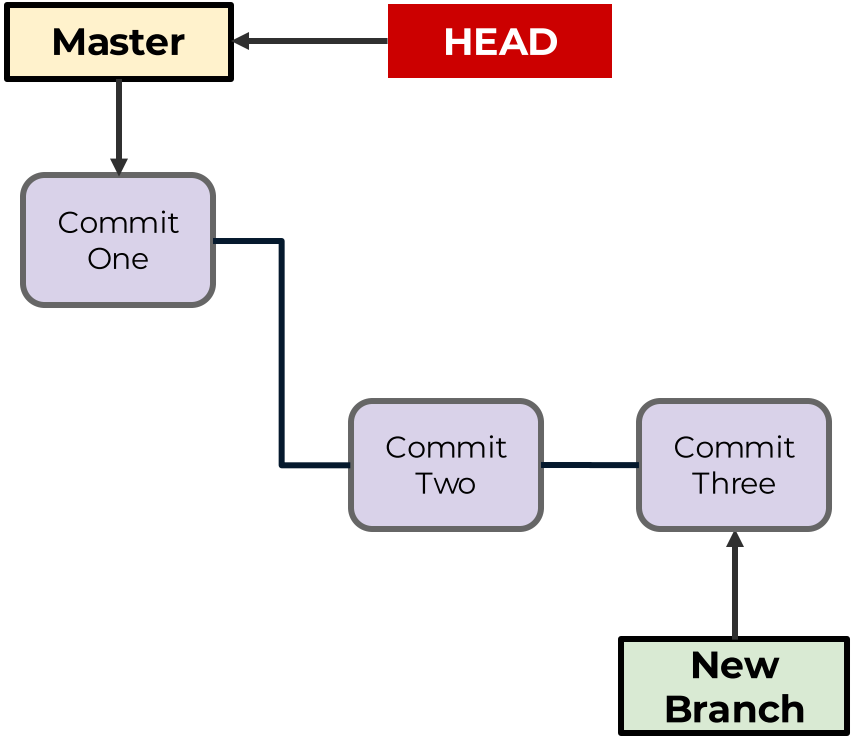
```
>> git merge new_branch
```

Week 3 Branches and Working with Others

- Now let’s explore what happens for a merge where we have different commits in the branches.

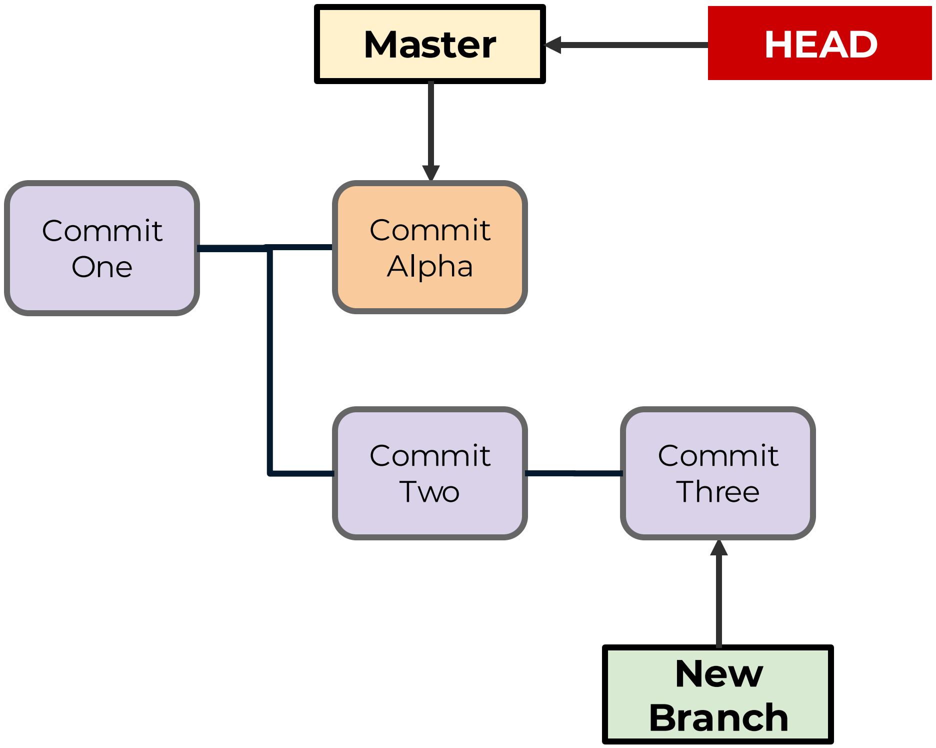
Week 3 Branches and Working with Others

- **Git Merge**



Week 3 Branches and Working with Others

- **Git Merge**



Master

HEAD

Commit One

Commit Alpha

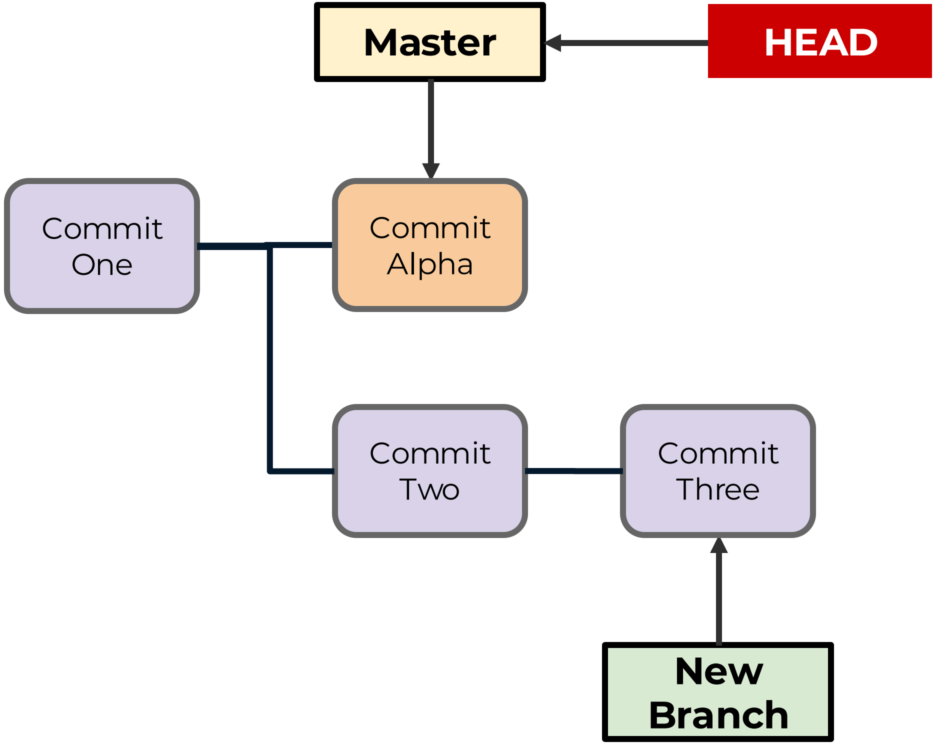
Commit Two

Commit Three

New Branch

Week 3 Branches and Working with Others

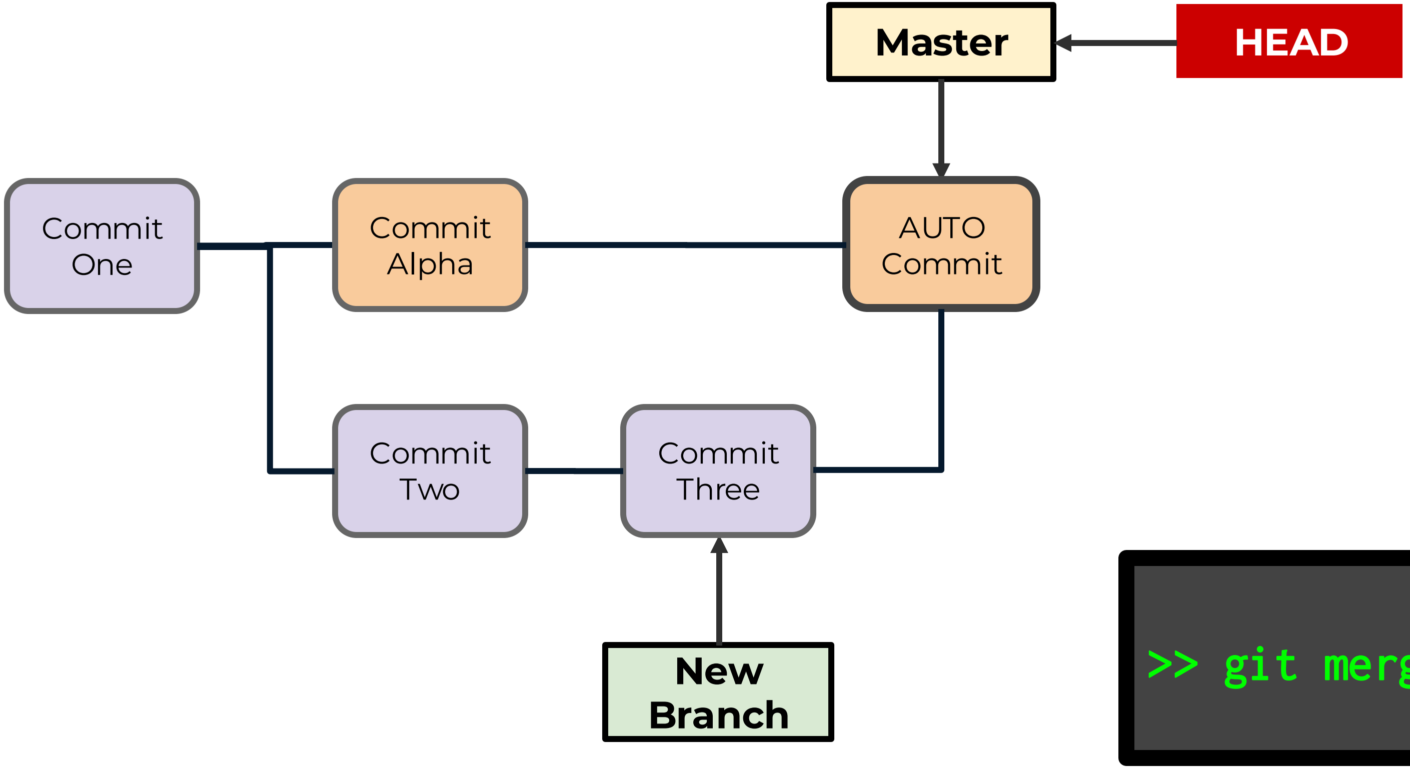
- **Git Merge**



```
>> git merge new_branch
```

Week 3 Branches and Working with Others

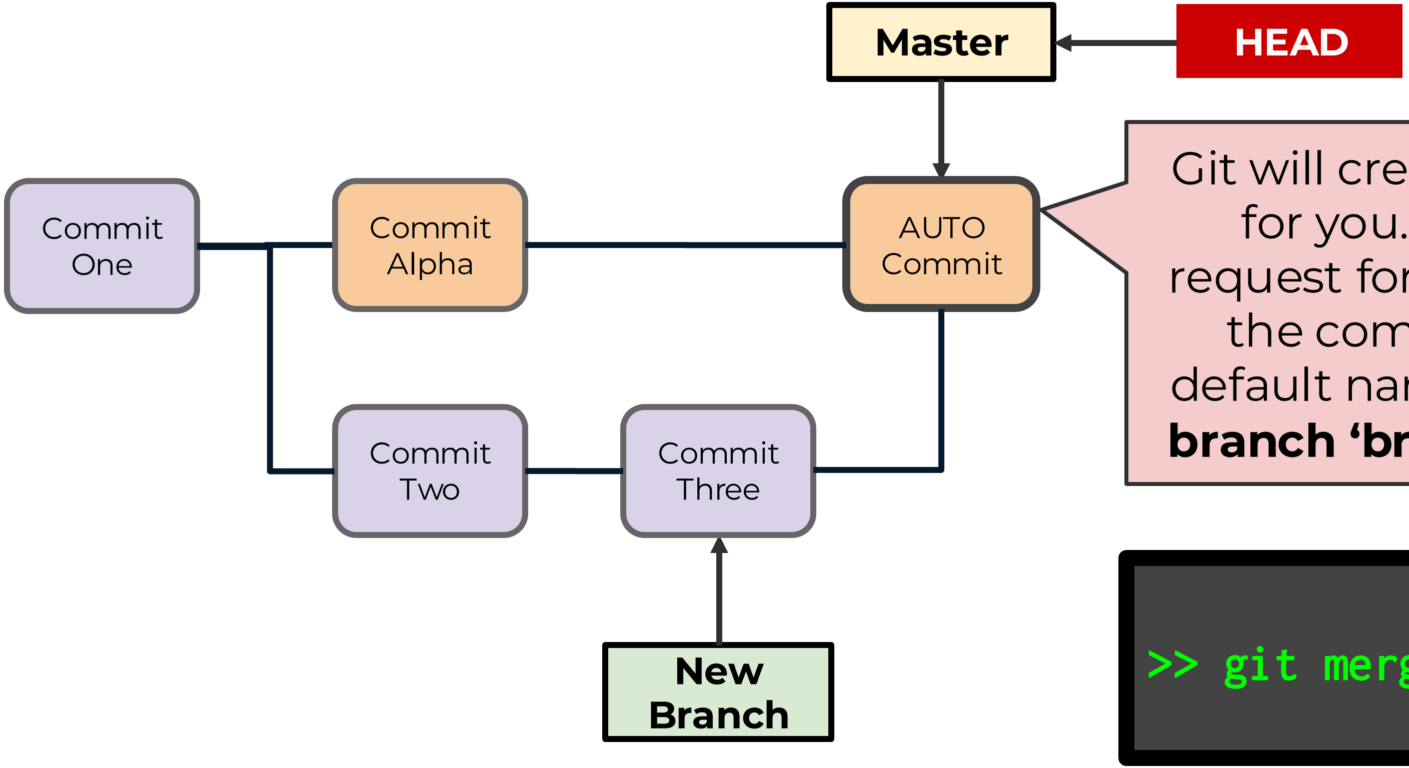
- **Git Merge**



```
>> git merge new_branch
```

Week 3 Branches and Working with Others

- **Git Merge**



Git will create a commit for you. It will also request for you to name the commit, with a default name of “**Merge branch ‘branch_name’**”

```
>> git merge new_branch
```

Week 3 Branches and Working with Others

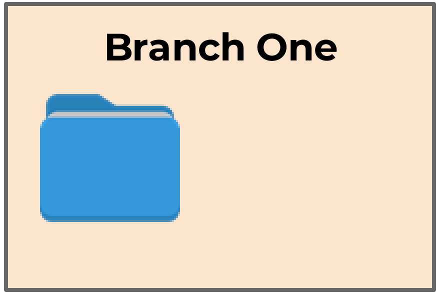
- Git creates the new commit for us, and will attempt the merge.
- Sometimes there are no conflicts, for example:
    - The branch only focused on files not in the receiving branch, thus the merge simply adds the new files to the receiving branch.

Week 3 Branches and Working with Others

- Git will try to automatically create the merge, and can do so in cases where no information from a specific branch would be lost, for example:

Week 3 Branches and Working with Others

- Git will try to automatically create the merge, and can do so in cases where no information from a specific branch would be lost, for example:



Branch One

Week 3 Branches and Working with Others

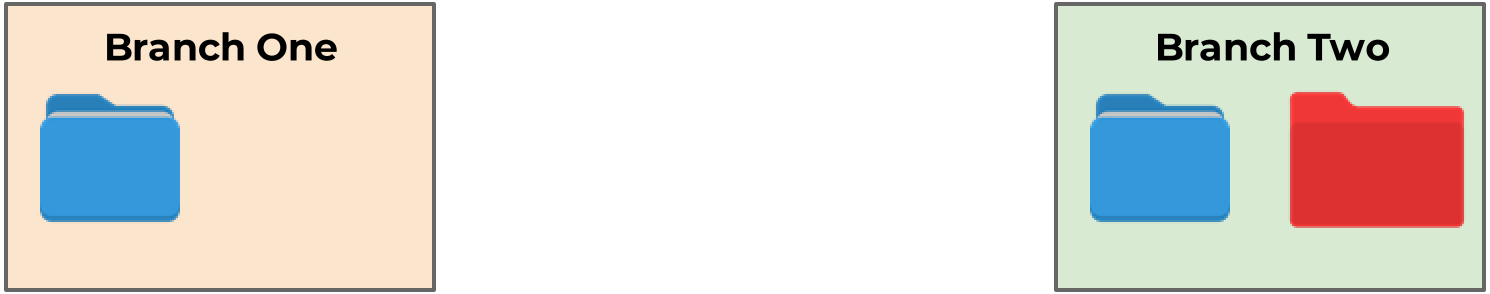
- Git will try to automatically create the merge, and can do so in cases where no information from a specific branch would be lost, for example:

Branch One

Branch Two

Week 3 Branches and Working with Others

- Git will try to automatically create the merge, and can do so in cases where no information from a specific branch would be lost, for example:

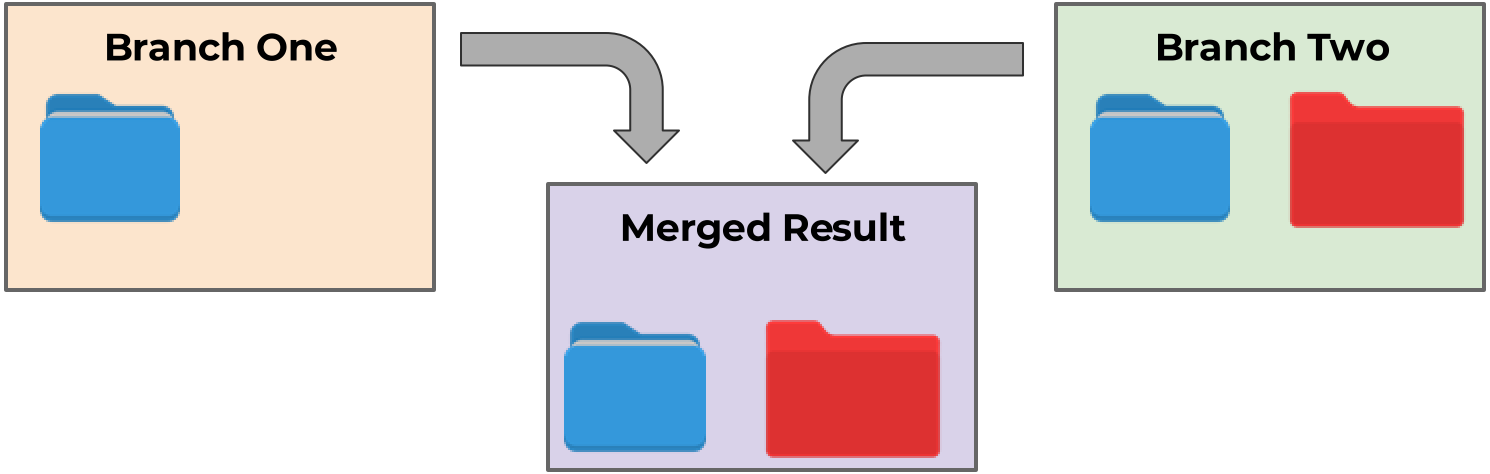


Branch One

Branch Two

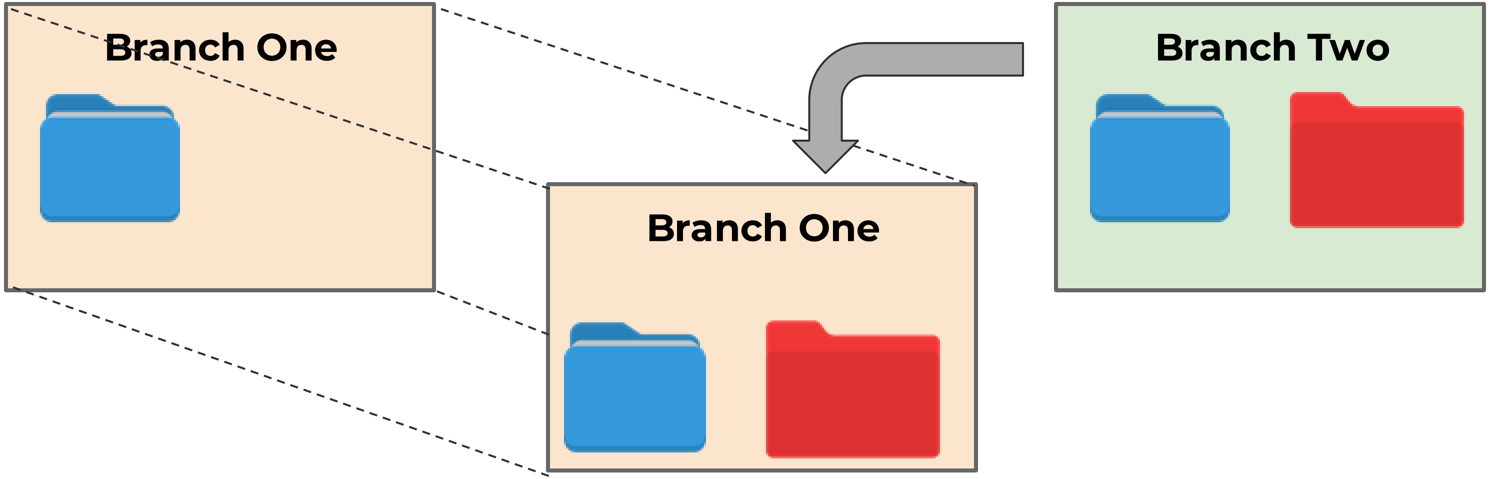
Week 3 Branches and Working with Others

- Git will try to automatically create the merge, and can do so in cases where no information from a specific branch would be lost, for example:



Week 3 Branches and Working with Others

- Git will try to automatically create the merge, and can do so in cases where no information from a specific branch would be lost, for example:


Branch One
Branch One
Branch Two