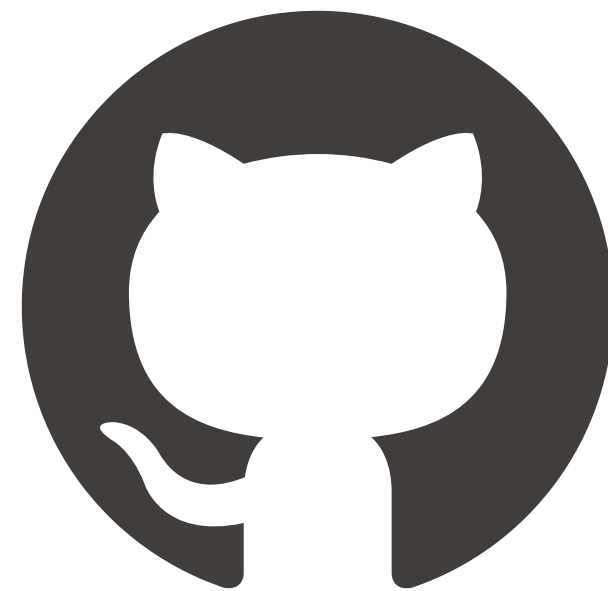


YHA Computer



GIT &
GITHUB

`</>` Cheat Sheet `</>`



OUTLINE

- INTRODUCTION TO VERSION CONTROL
- BEFORE VERSION CONTROL
- GIT & GITHUB
- UNDERSTANDING GIT/LINUX COMMAND
- USING GITHUB
- USING A REMOTE REPOSITORY



INTRODUCTION TO **VERSION CONTROL**

Version Control, also known as source control, is all about tracking and managing changes to software code.



BEFORE

Version Control

Before VCS, programmers managed code through manual backups, comments for tracking changes, naming conventions for versions, shared folders, code diff tools, manual merging, archives, custom backup scripts, change logs, and close communication.





YHA Computer

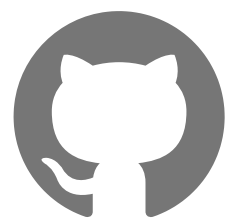
Git & GitHub

Git

Git is a version control system that allows developers to track changes in their code.

GitHub

GitHub is a web-based hosting service for git repositories.



setup & init

Configure user information, initializing and cloning repositories.

YHA Computer

\$ git init

Initialize an existing directory as a Git repository



Normal Directory

git init

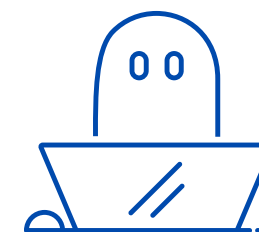


Git Repository

\$ git clone [url]

Retrieve an entire repository from a hosted location via URL.

git clone [URL]



Local Copy of the Repository

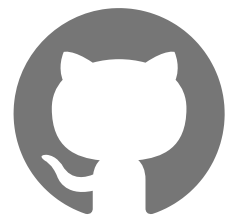


Remote Repository at [URL]

git clone [URL]



Local Copy of the Repository



stage & snapshot

Working with snapshots and the Git staging area.

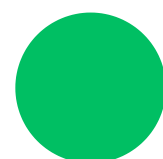
YHA Computer

```
$ git status
```

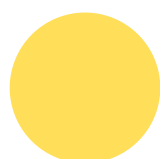
show modified files in working directory, staged for next commit.



Working Directory



Staged snapshot



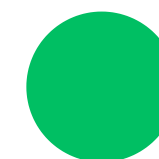
Unstaged snapshot

```
$ git add [file]
```

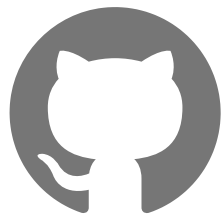
Add a file as it looks now to your next commit (stage).



Working Directory



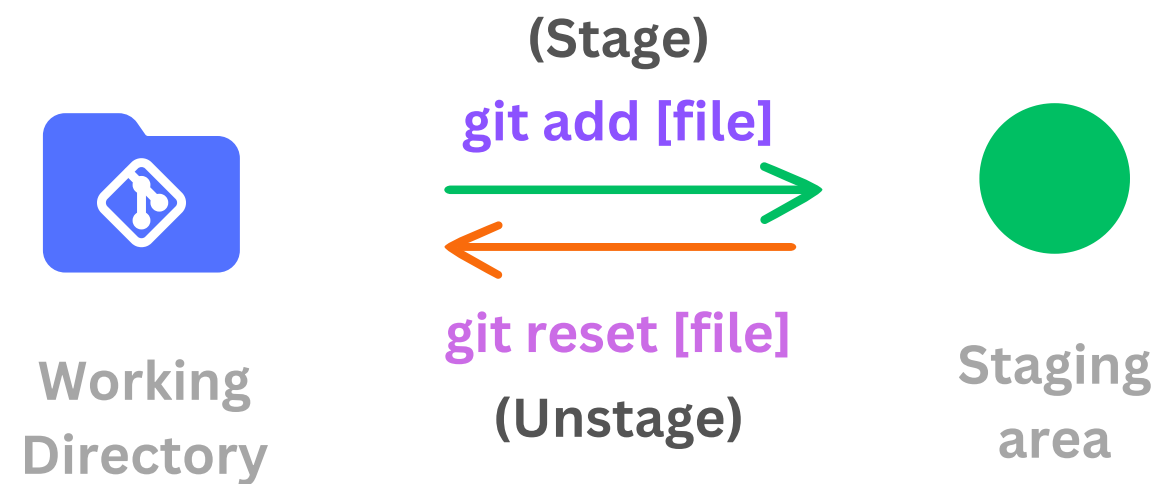
Staged snapshot



YHA Computer

`$ git reset [file]`

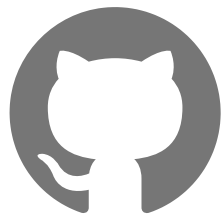
Unstage a file while retaining the changes in working directory



`$ git diff`

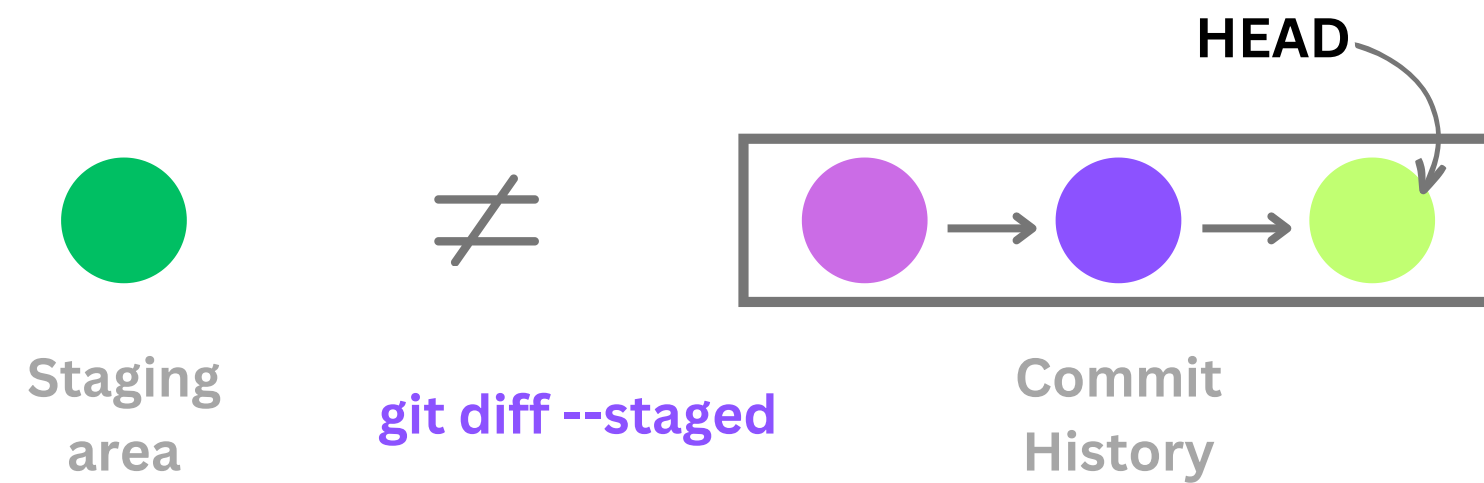
Diff of what is changed but not staged.





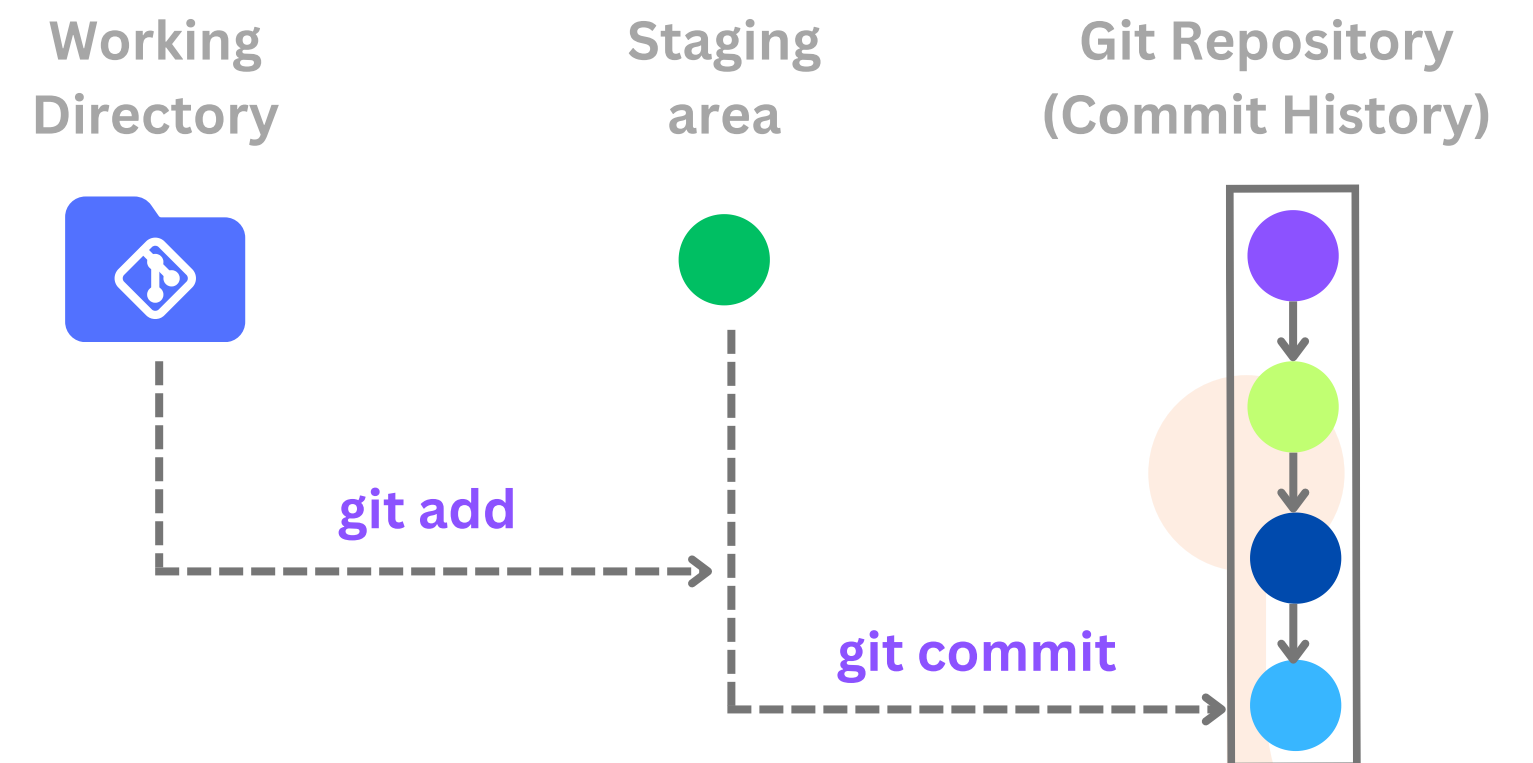
```
$ git dif --staged
```

Diff of what is staged but not yet committed.



```
$ git commit -m "[message]"
```

Commit your staged content as a new commit snapshot.





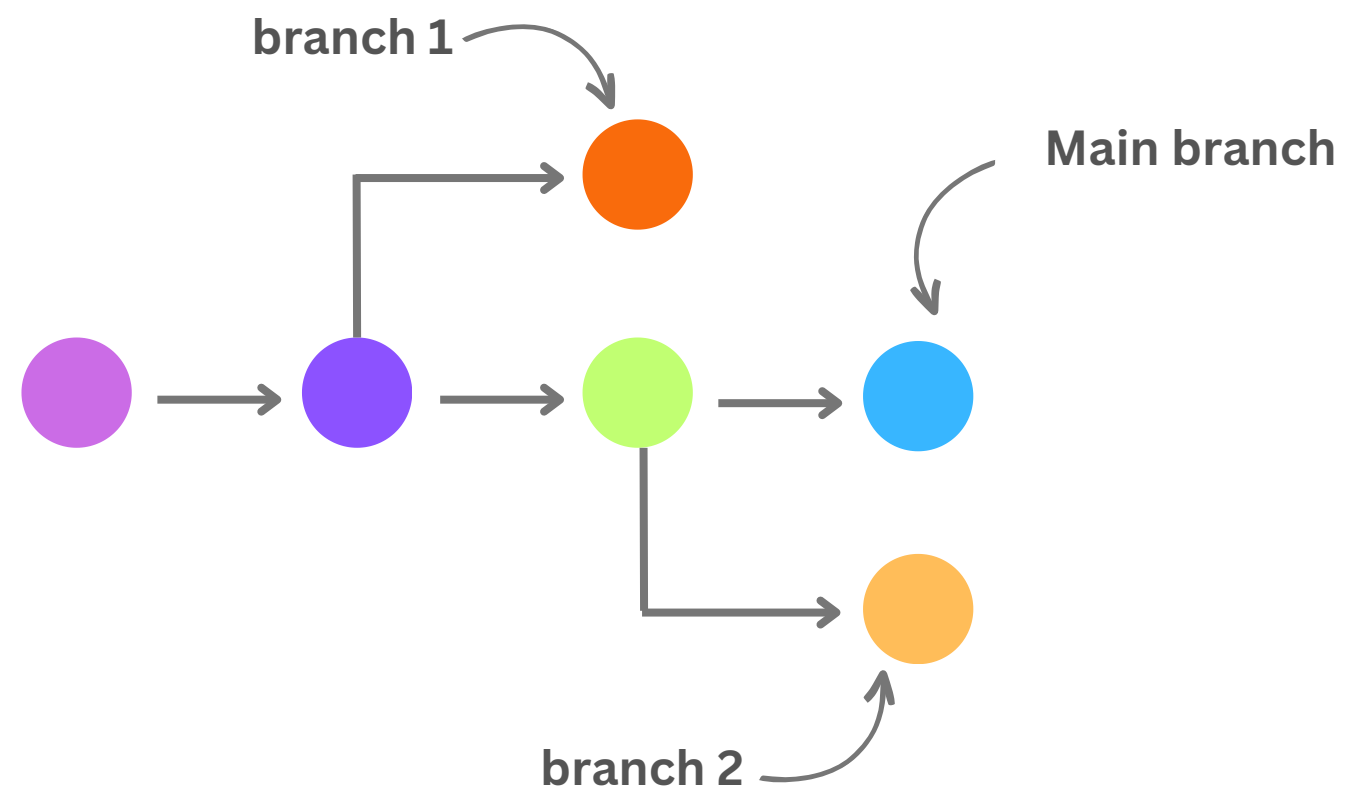
Branch & Merge

YHA Computer

List your branches. A * will appear next to the currently active branch.

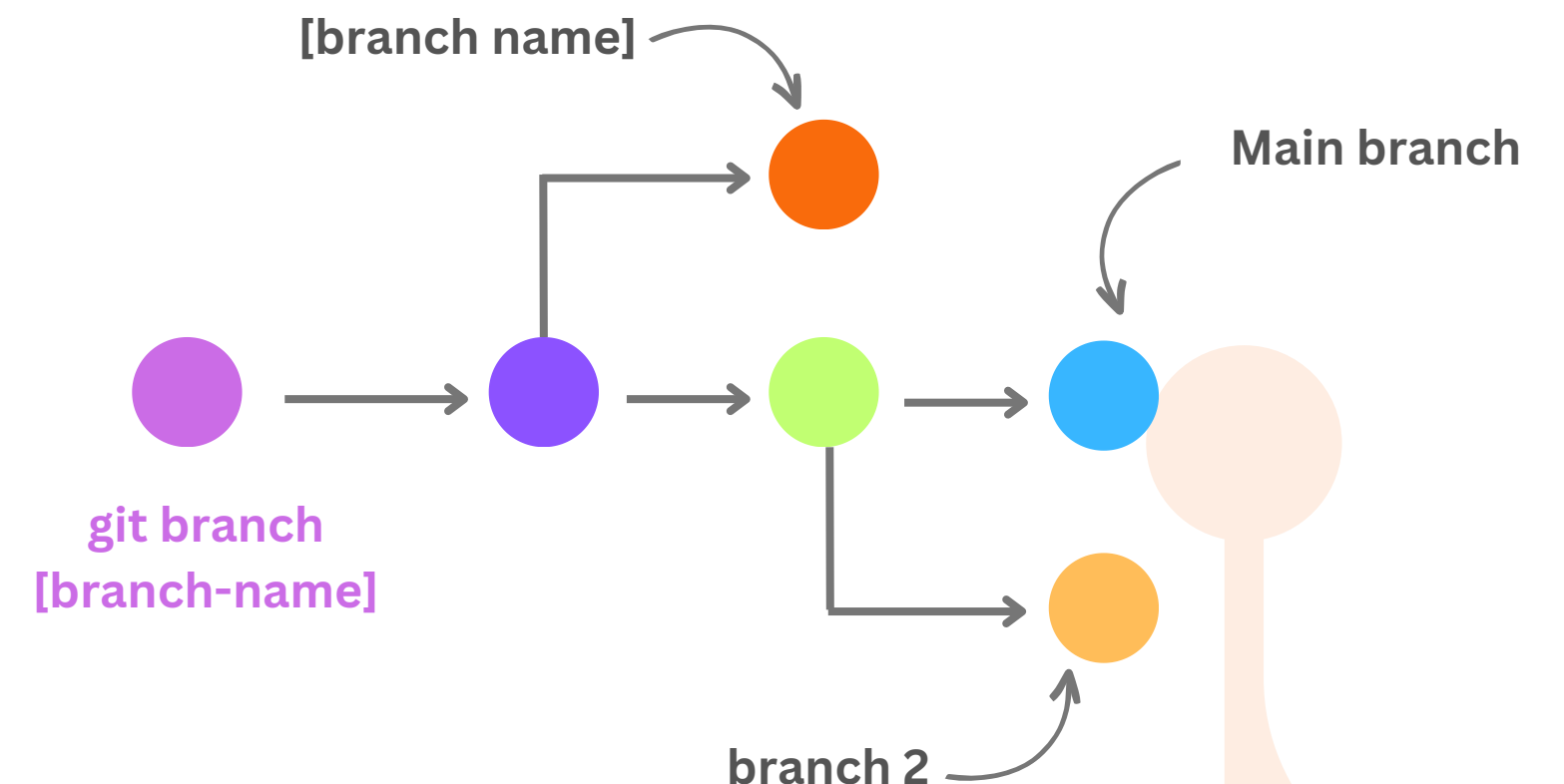
```
$ git branch
```

List your branches. A * will appear next to the currently active branch.



```
$ git branch [branch-name]
```

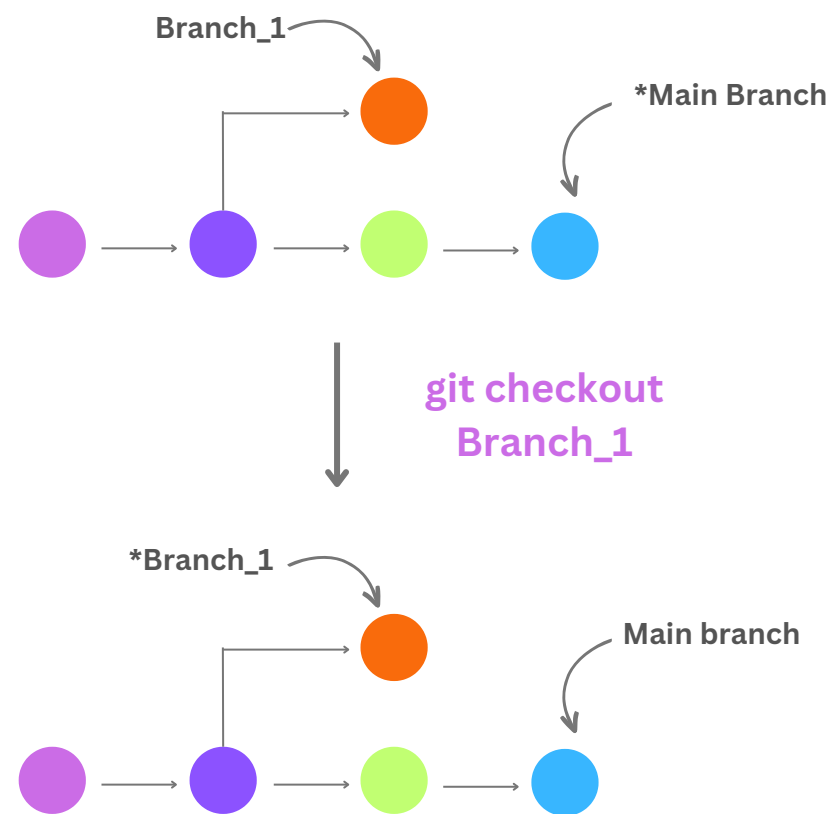
Create a new branch at the current commit.





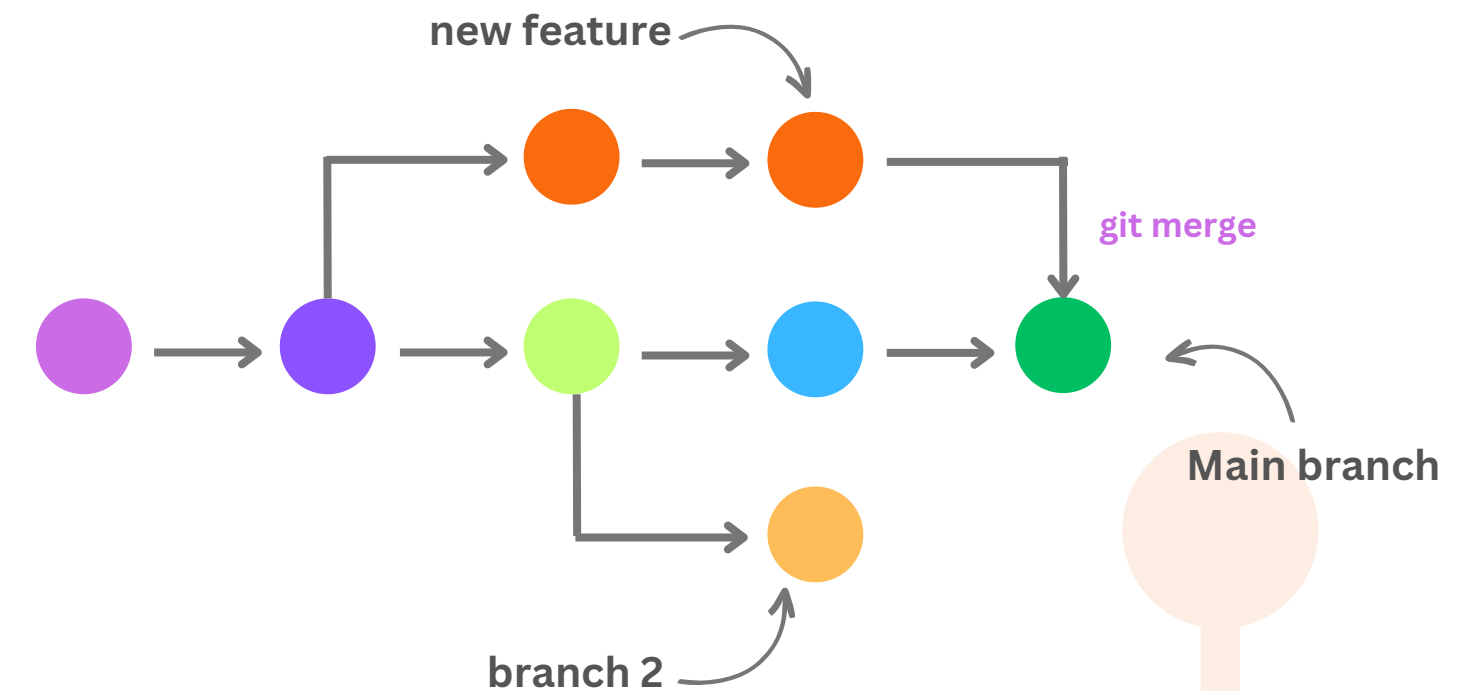
\$ git checkout

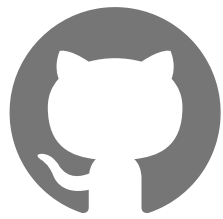
Switch to another branch and check it out into your working directory.



\$ git merge [branch]

Merge the specified branch's history into the current one.

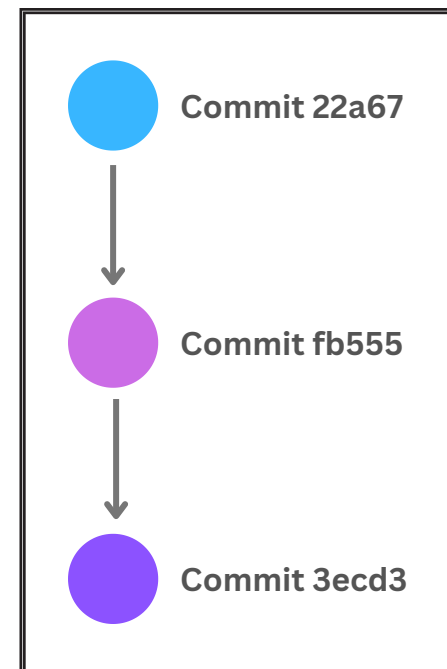




\$ git log

Add a file as it looks now to your next commit (stage).

Git Repository



git log

Git Logs

Commit 22a67
Author: xyz
Date: Wed, Oct 25 12:04:45
Commit Message

Commit 22a67
Author: ABC
Date: Thur, Oct 26 16:05:35
Commit Message

Commit 22a67
Author: XYZ
Date: Fri, Oct 27 20:25:10
Commit Message



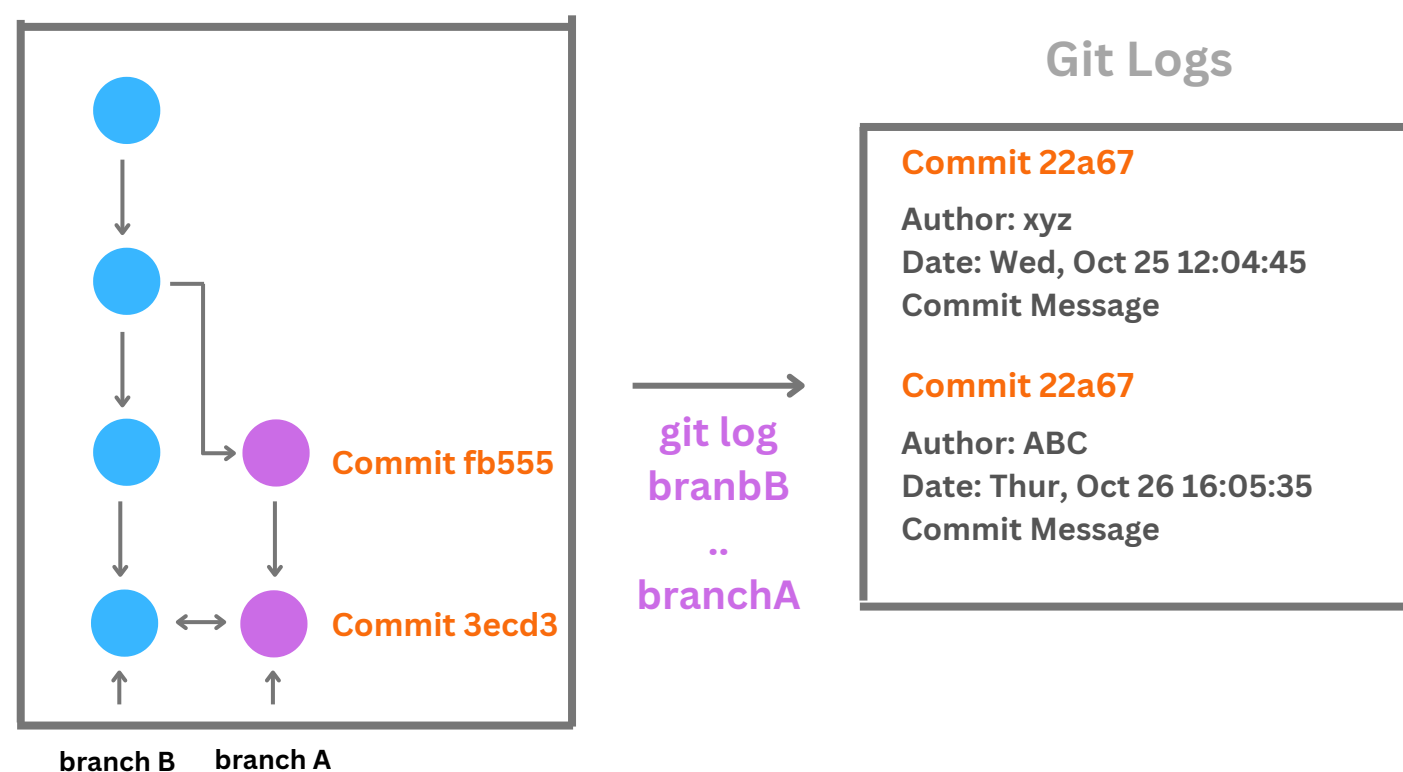
Inspect & Compare

Configuring user information, initializing and cloning repositories.

YHA Computer

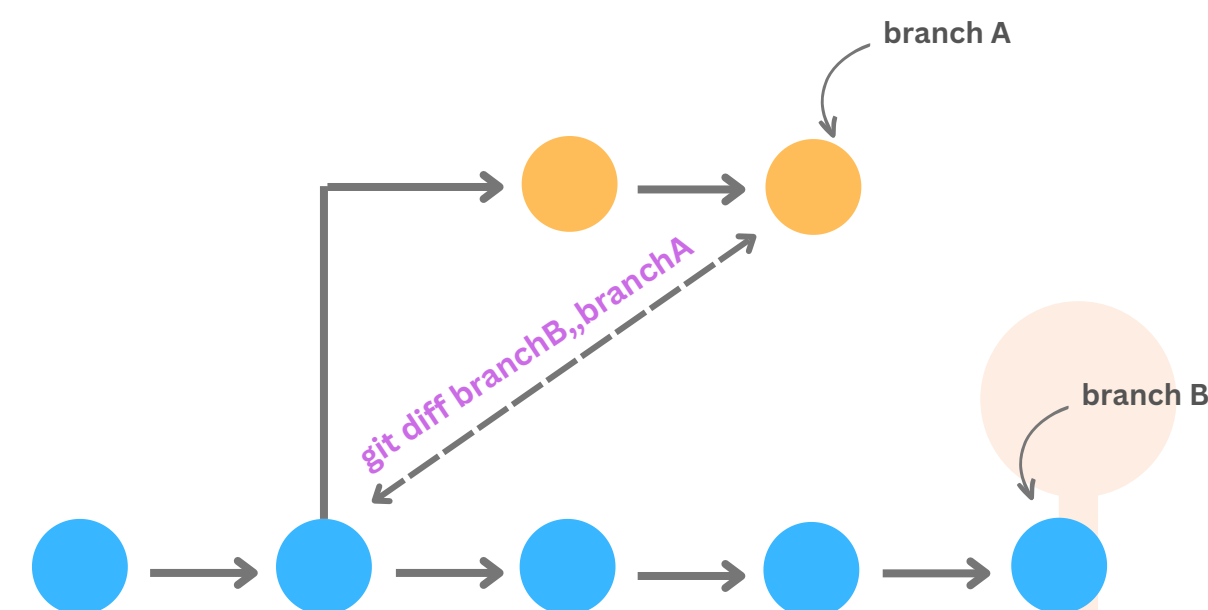
```
$ git log branchB..branchA
```

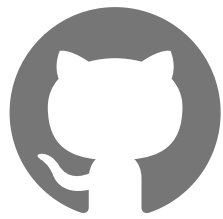
Show the commits on branchA that are not on branchB.



```
$ git diff branchB..branchA
```

Show the diff of what is in branchA that is not in branchB.





`$ git show [SHA]`

Show any object in Git in human-readable format

commit 22a67



`git show 22a67`

Commit 22a67
Author: xyz
Date: Wed, Oct 25 12:04:45
Commit Message

File1
File1 changes

File2
File2 changes

File3
File3 changes

....

`$ git log --follow [file]`

Show the commits that changed file,
even across renames.



Tracking Path Changes

Versioning file removes and path changes.

YHA Computer

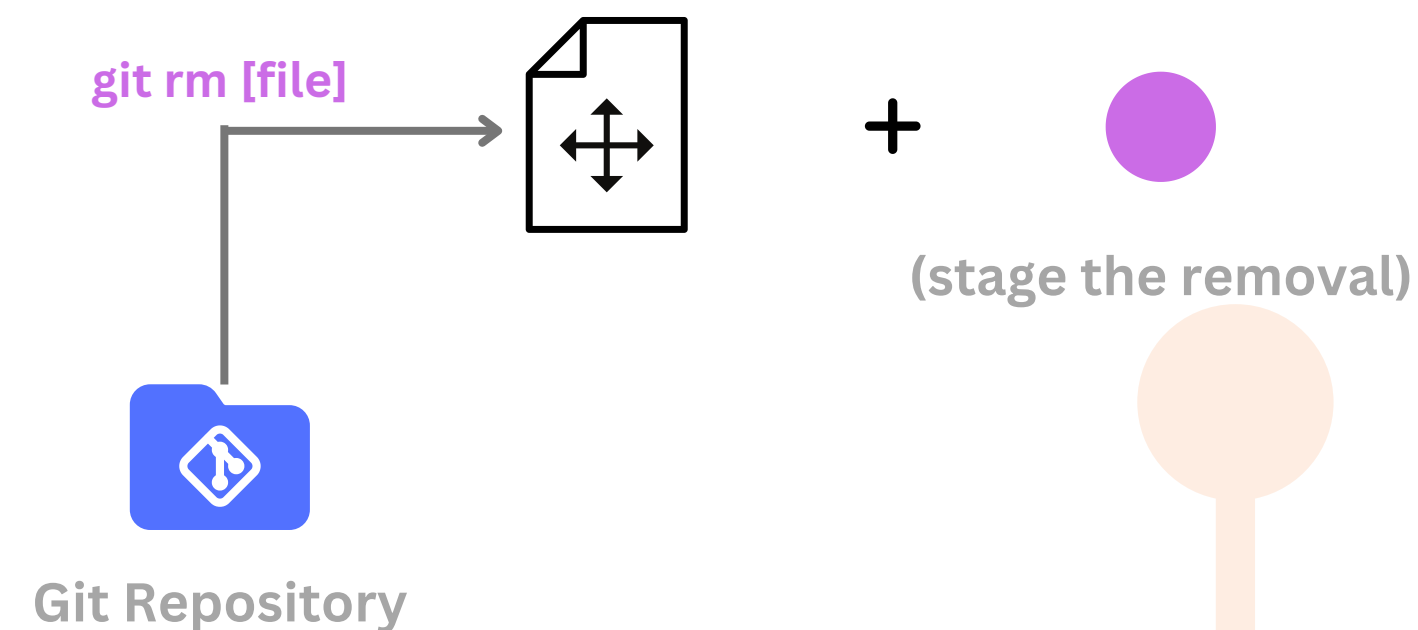
```
$ git rm [file]
```

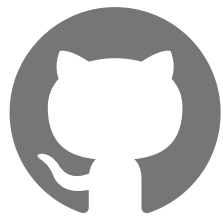
Delete the file from the project and stage the removal for commit.



```
$ git mv [existing-path] [new-path]
```

Change an existing file path and stage the move





```
$ git log --stat -M
```

Show all commit logs with indication
of any paths that moved.

Git Logs

Commit 22a67

Author: xyz

Date: Wed, Oct 25 12:04:45

.....

Commit Message

Commit 22a67

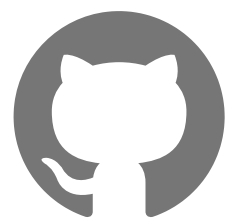
Author: xyz

Date: Wed, Oct 25 12:04:45

.....

Commit Message

.....

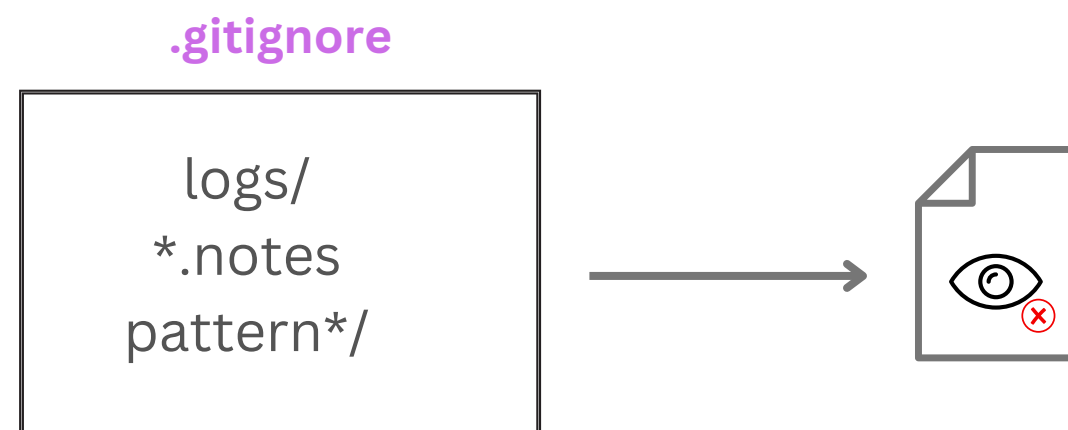


Ignoring Patterns

Preventing unintentional staging or committing of files.

```
logs/  
*.notes  
pattern*/
```

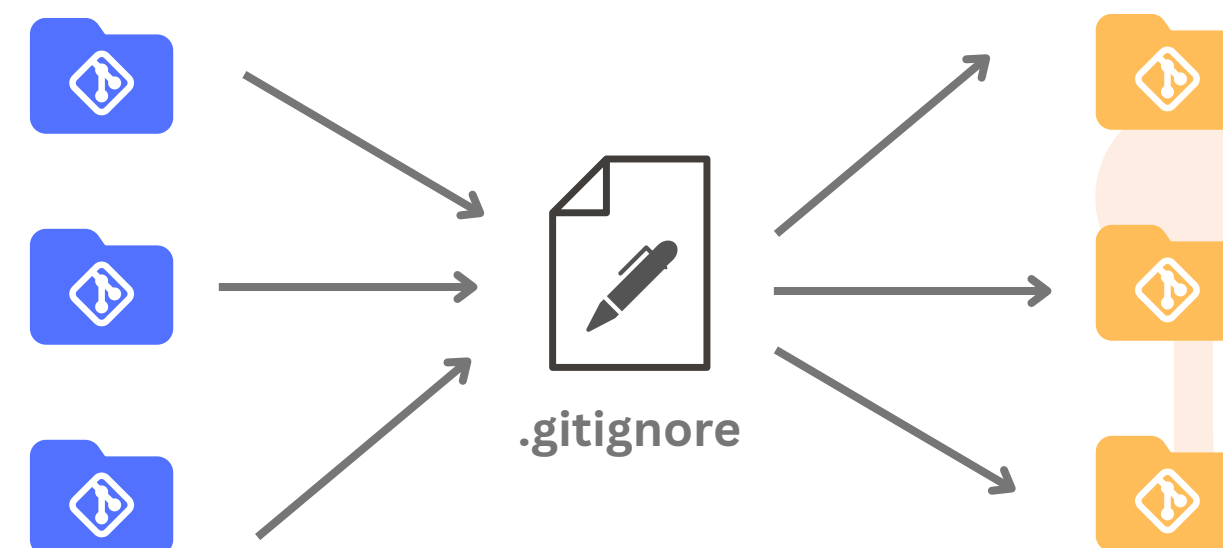
Save a file with desired patterns as .gitignore with either direct string matches or wildcard globs.



YHA Computer

```
$ git config --global  
core.excludesfile [file]
```

Change an existing file path and stage the move



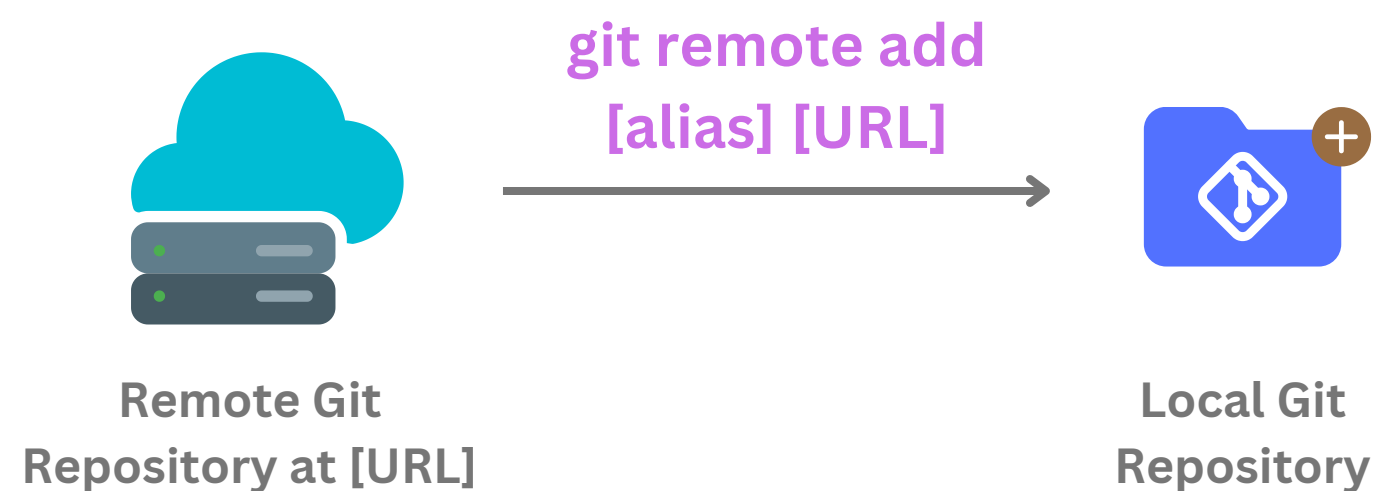


Share and Update

Retrieving updates from another repository and updating local repos.

```
$ git remote add [alias] [url]
```

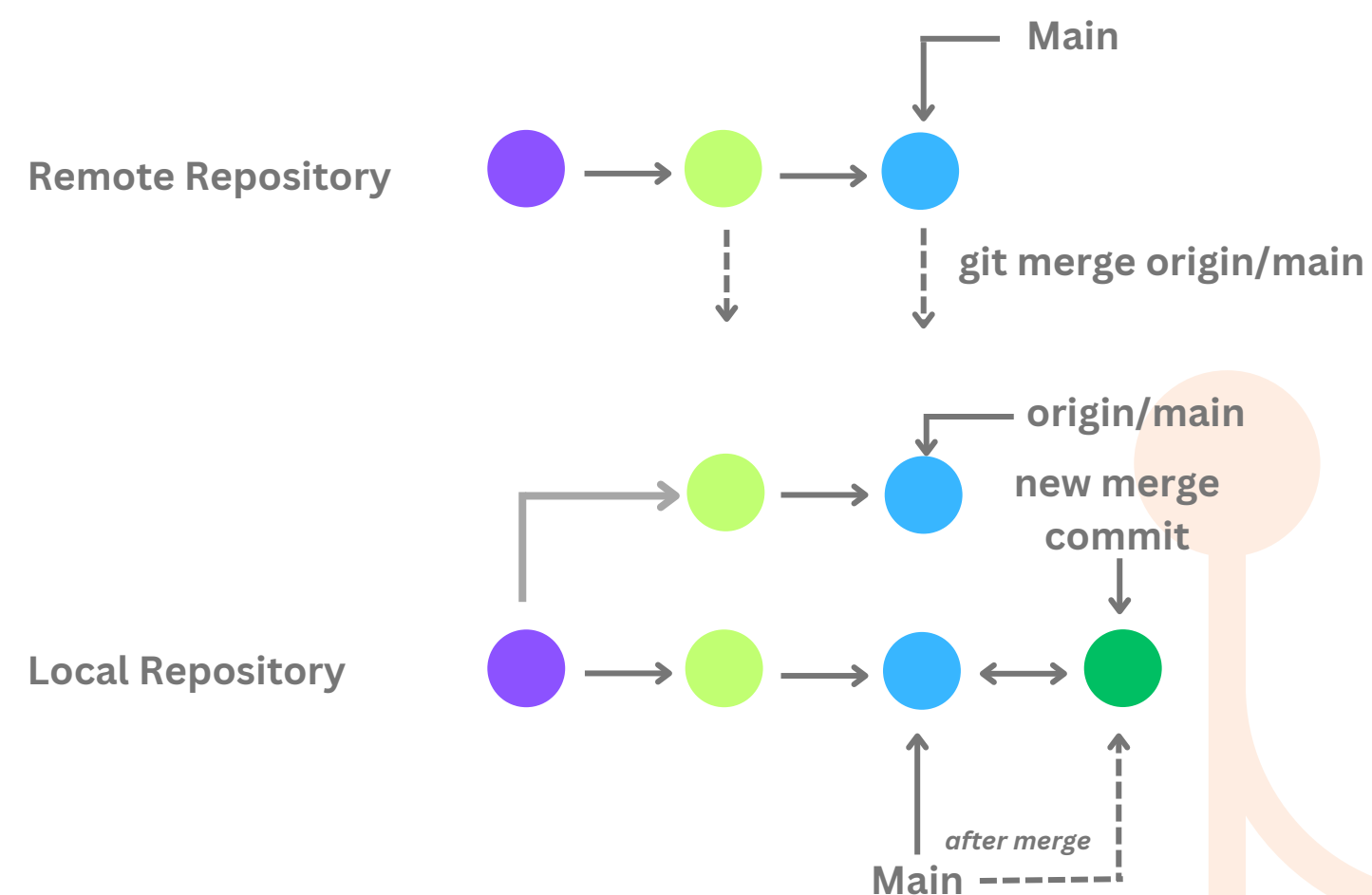
Add a git URL as an alias.

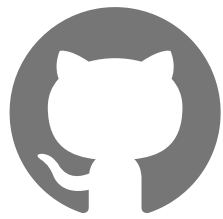


YHA Computer

```
$ git merge [alias]/[branch]
```

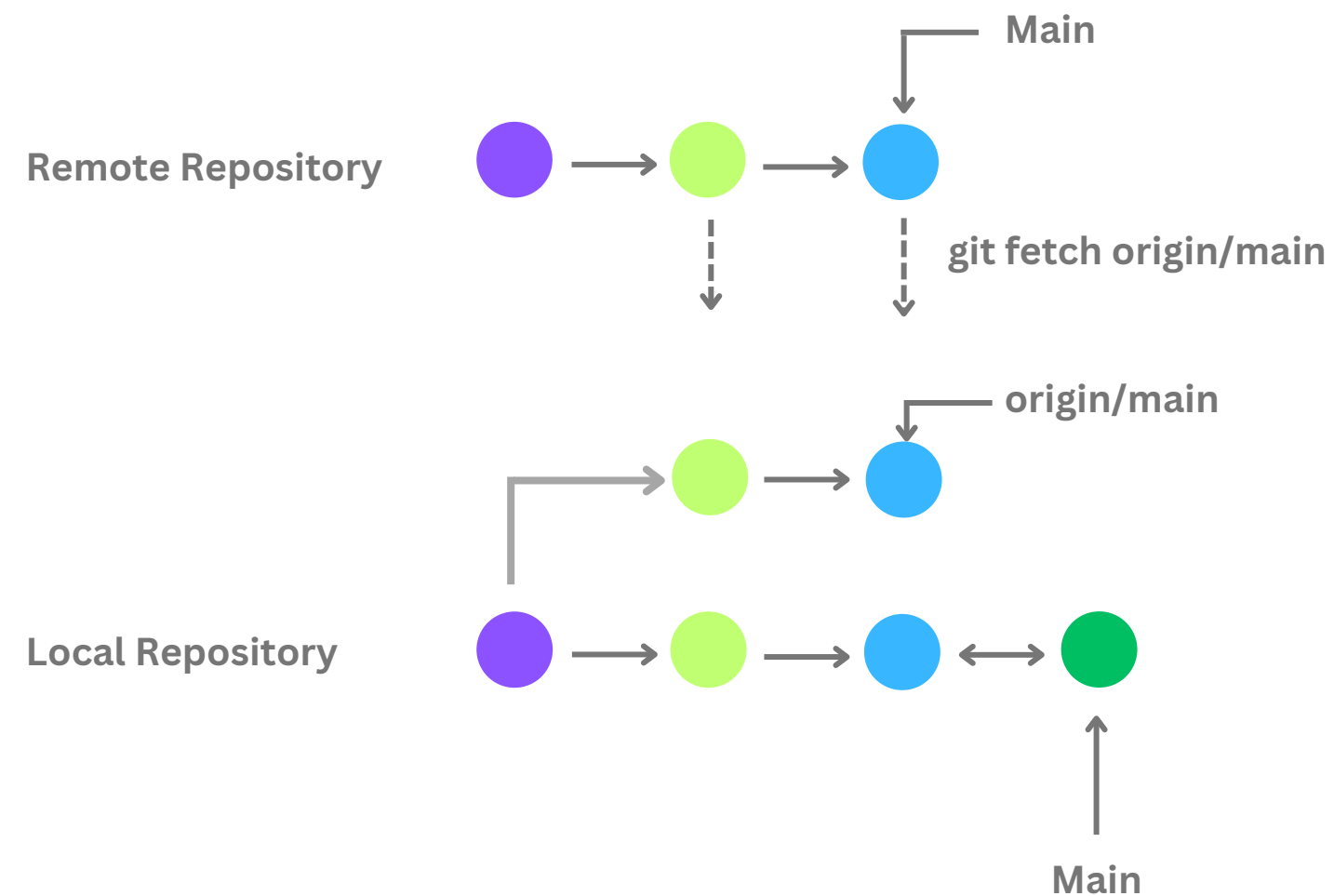
Merge a remote branch into your current branch to bring it up to date.





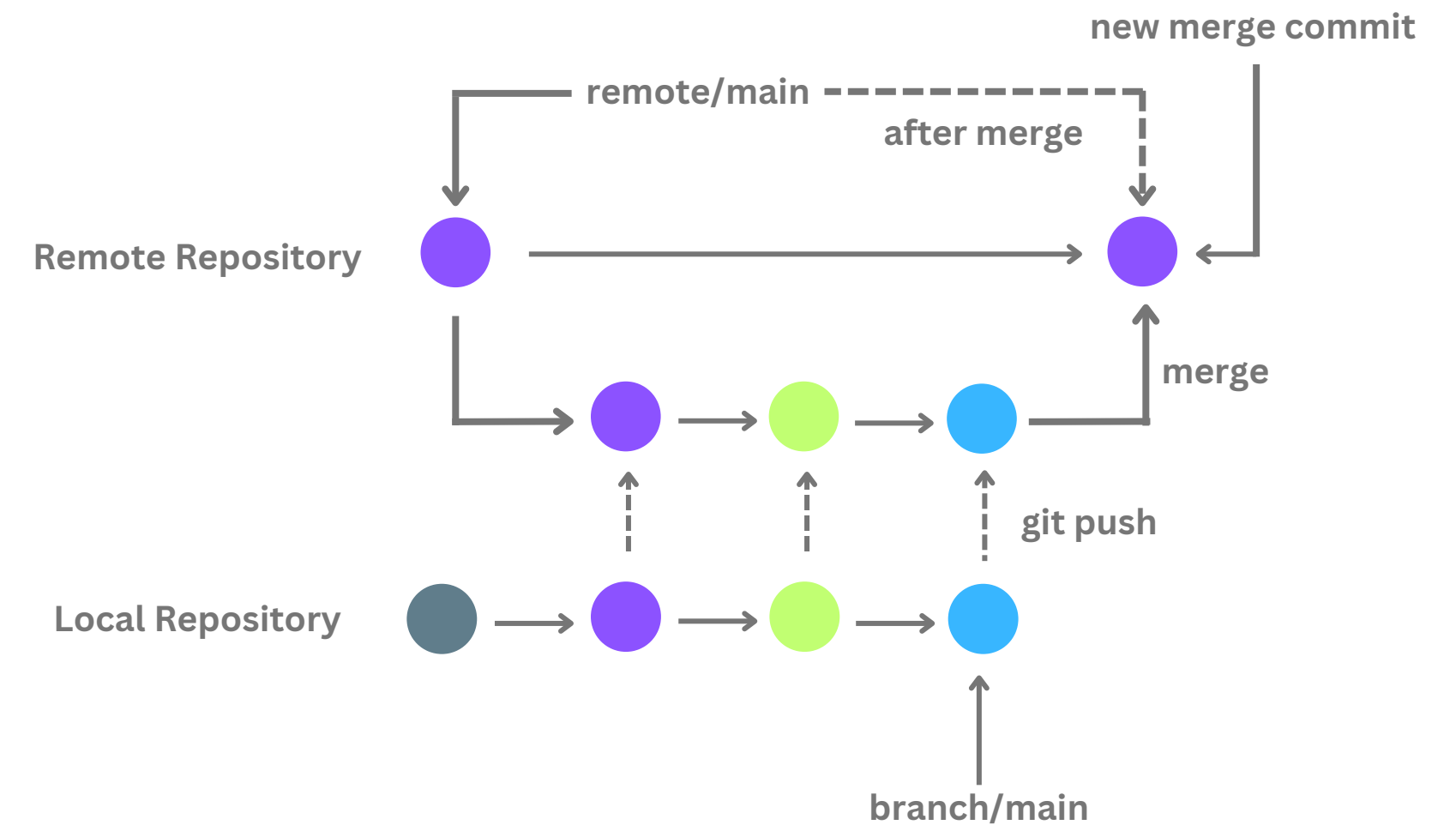
`$ git fetch [alias]`

Add a git URL as an alias.



`$ git push [alias] [branch]`

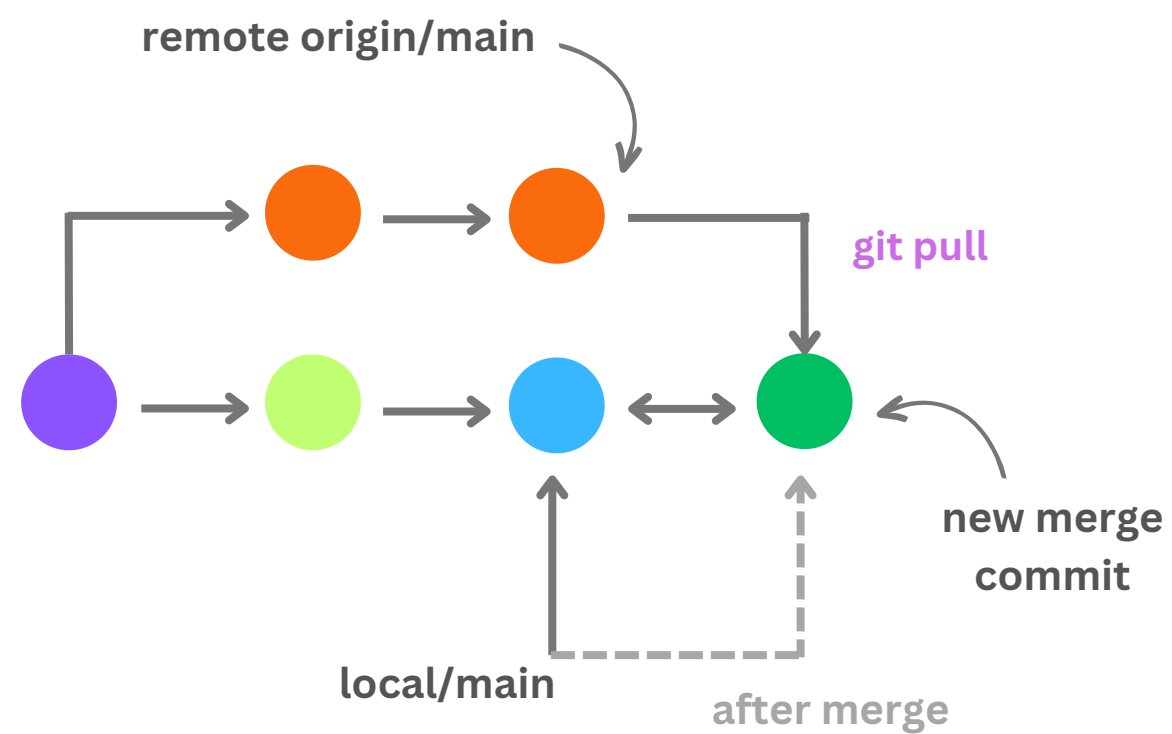
Merge a remote branch into your current branch to bring it up to date.





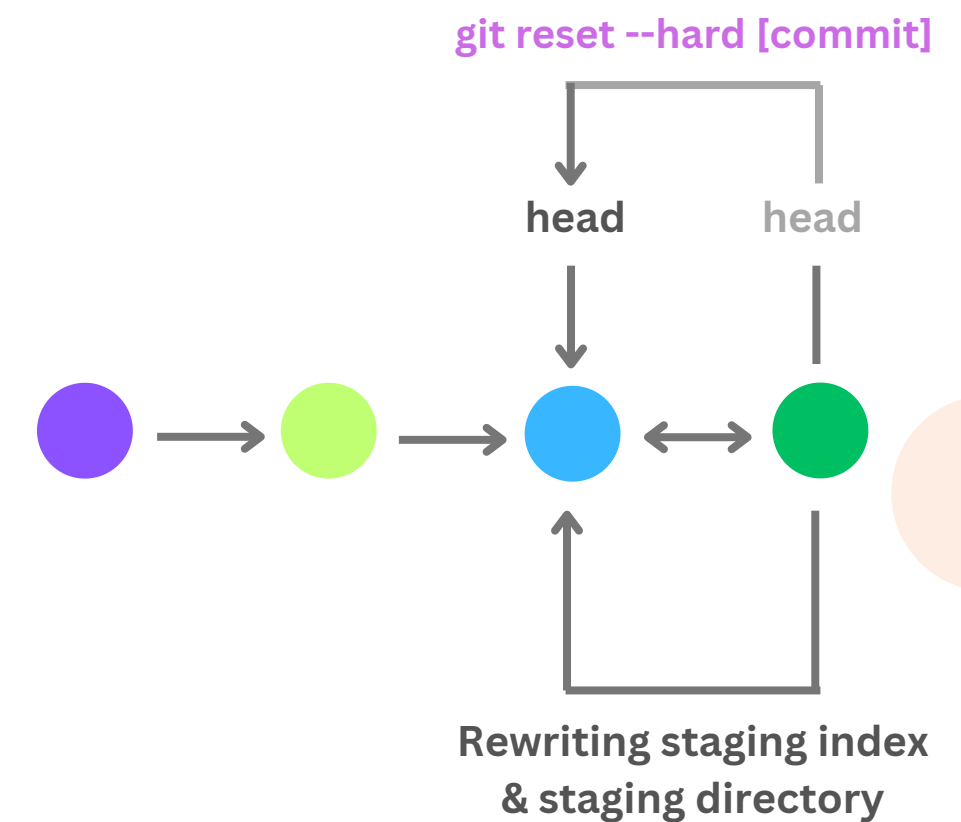
`$ git pull`

Fetch and merge any commits from the tracking remote branch.



`$ git reset --hard [commit]`

Clear staging area, rewrite working tree from specified commit.





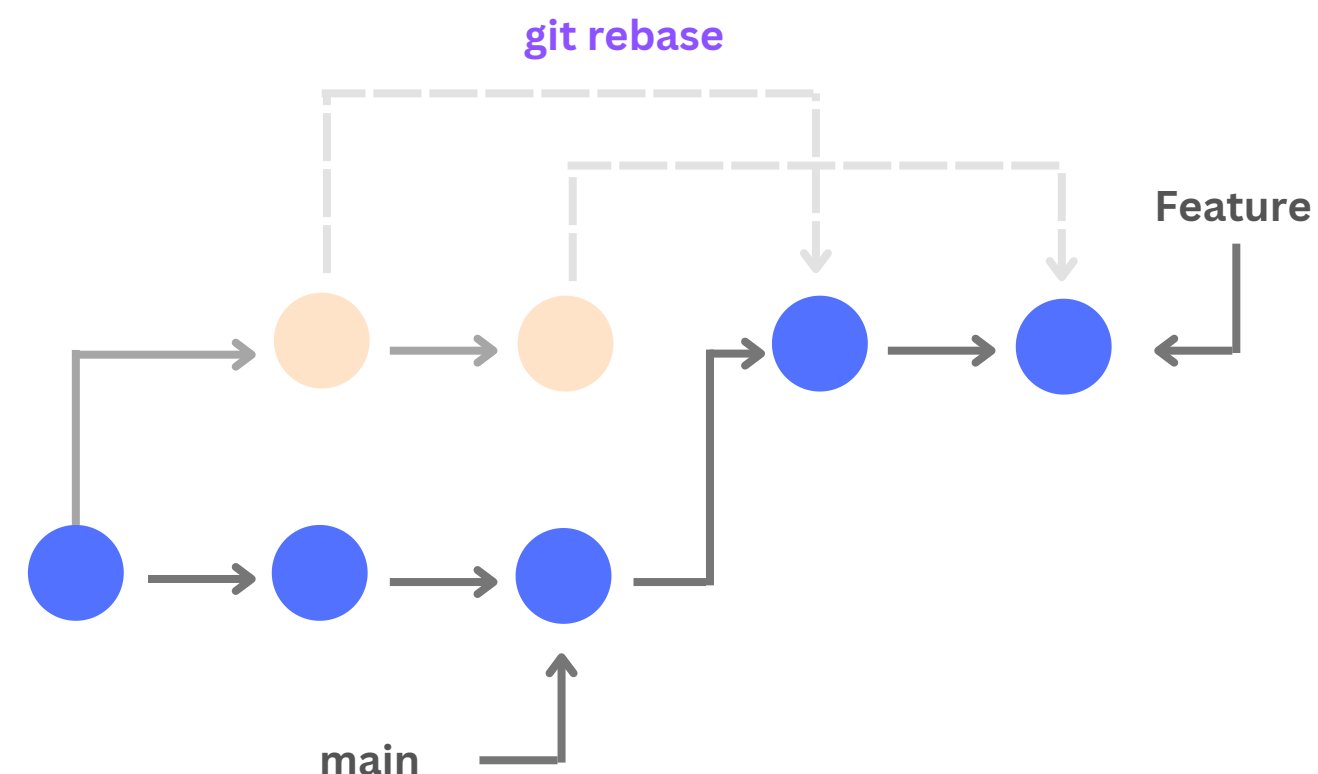
Rewrite History

Rewriting branches, updating commits and clearing history.

YHA Computer

```
$ git rebase [branch]
```

Apply any commits of the current branch ahead of specified one.





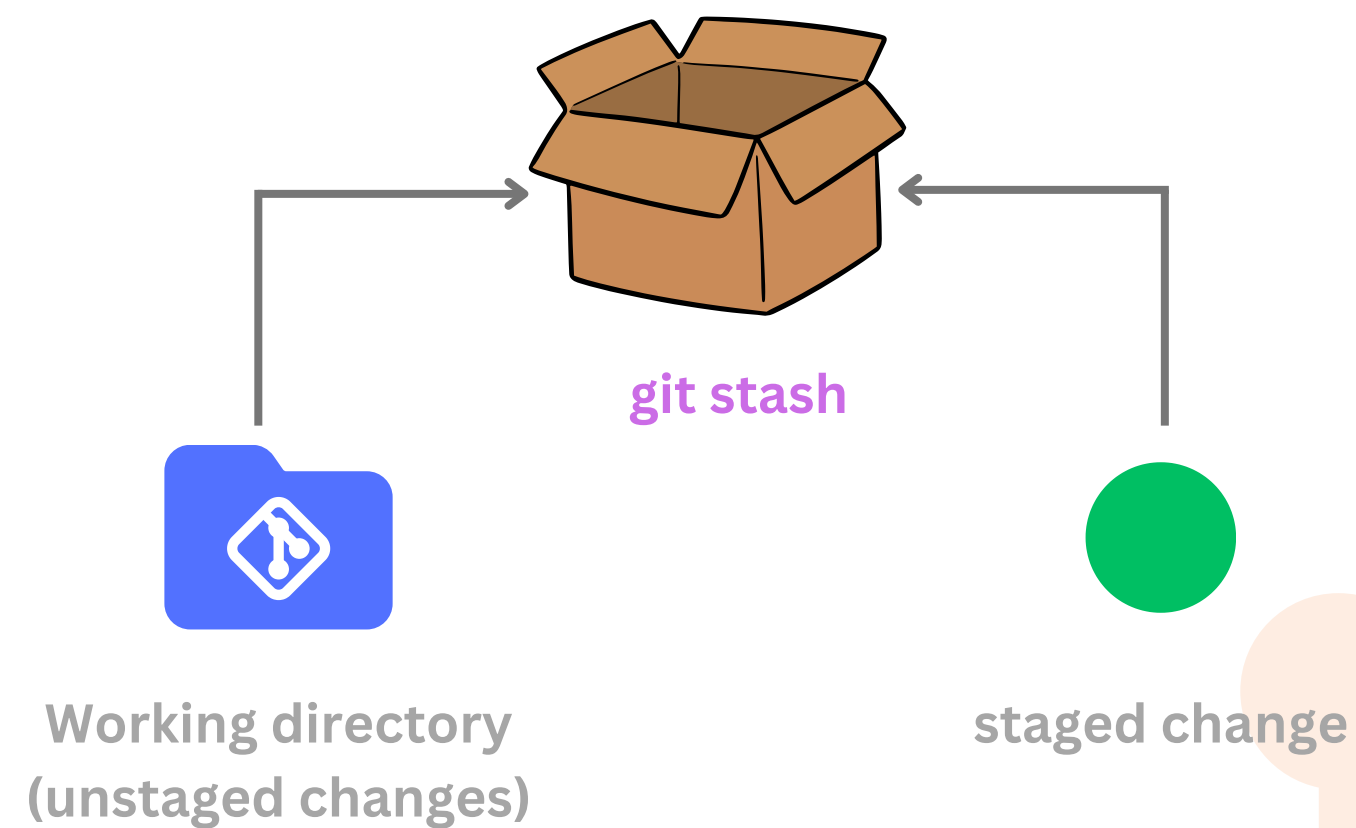
Temporary Commits

Temporarily store modified, tracked files in order to change branches.

YHA Computer

\$ git stash

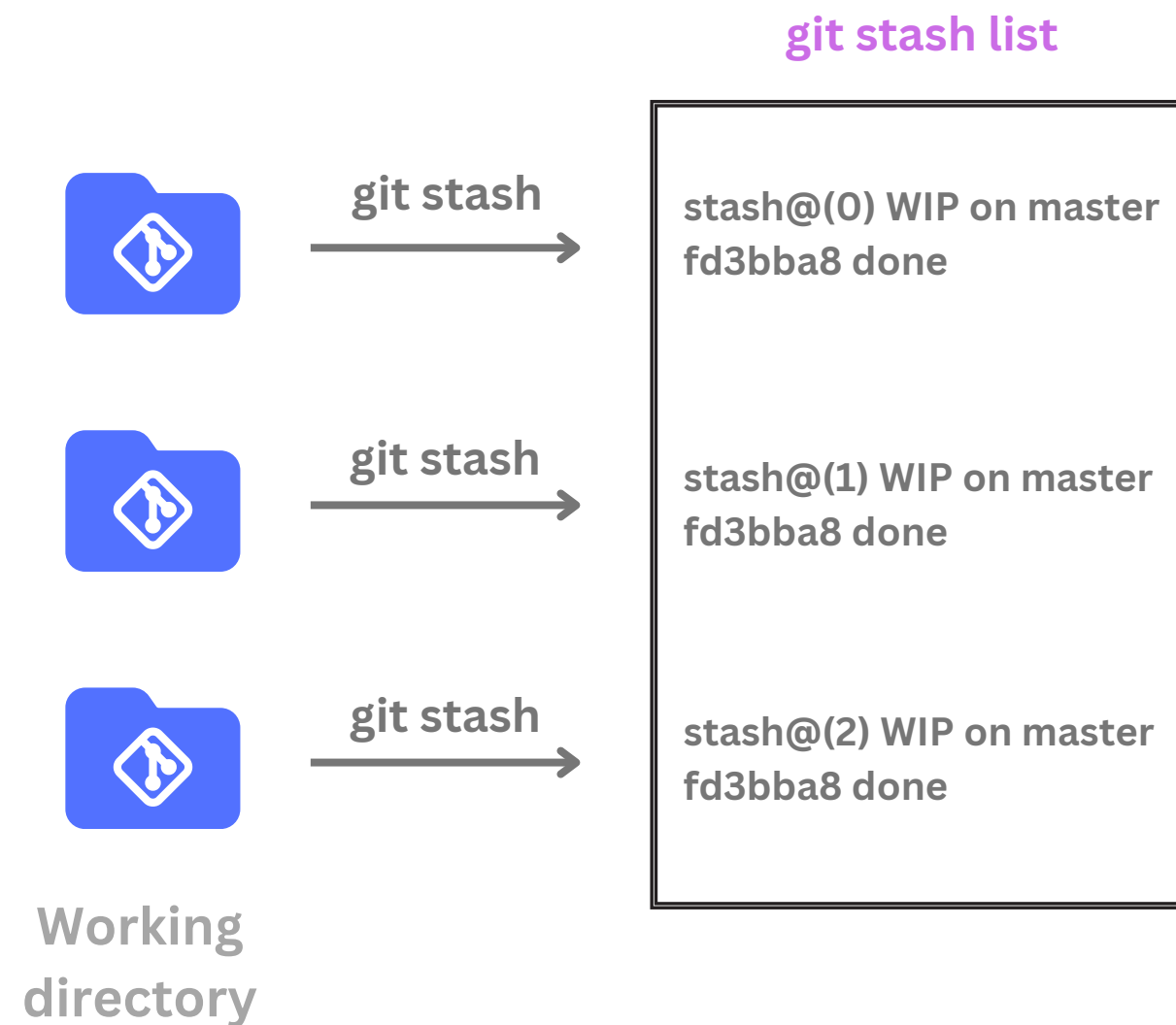
Save modified and staged changes.





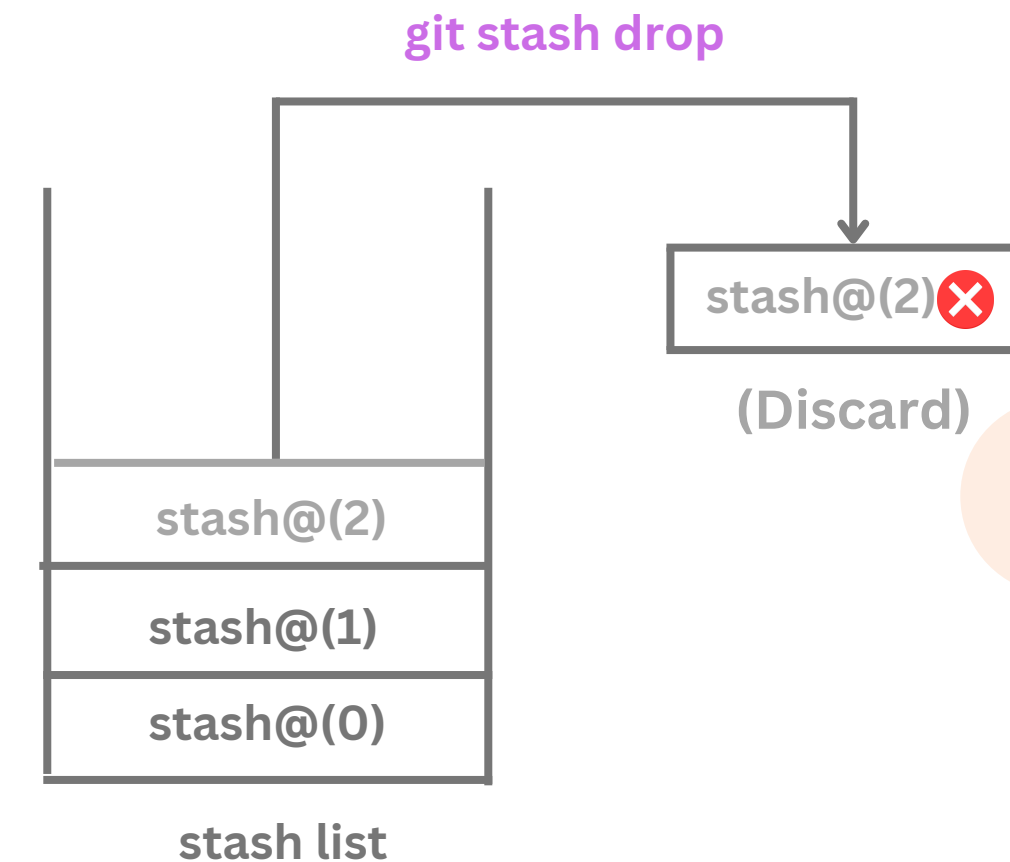
\$ git stash list

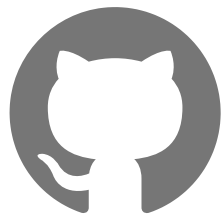
List stack-order of stashed file changes.



\$ git stash drop

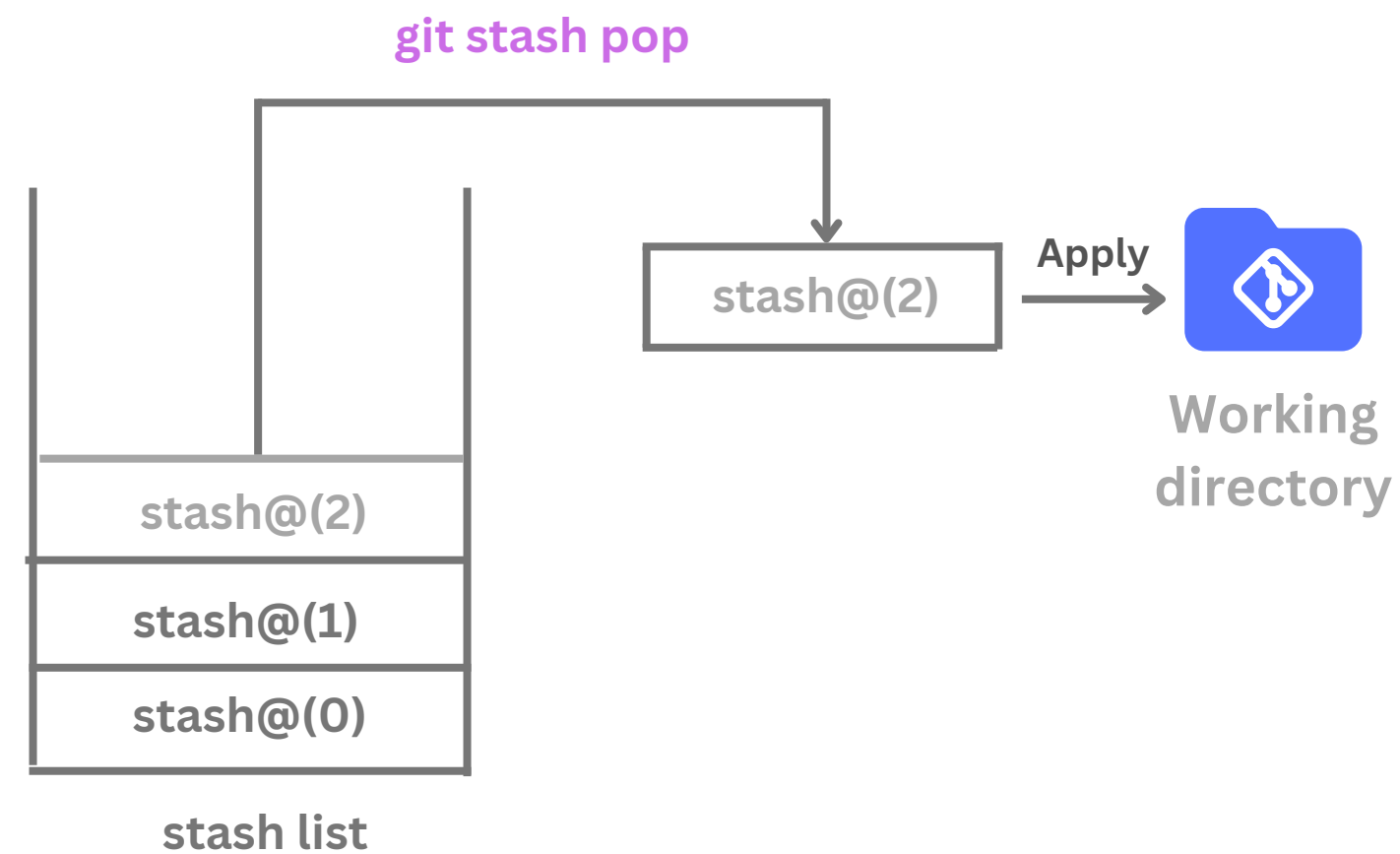
Discard the changes from the top of the stash stack.





```
$ git stash pop
```

Write working from the top of the
stash stack.





Reference

[Git Documentation](#)
[SlideShare.com](#)
[Atlassian.com](#)
[ScalerTopic.com](#)
[javatpoint.com](#)
[geeksforgeeks.com](#)