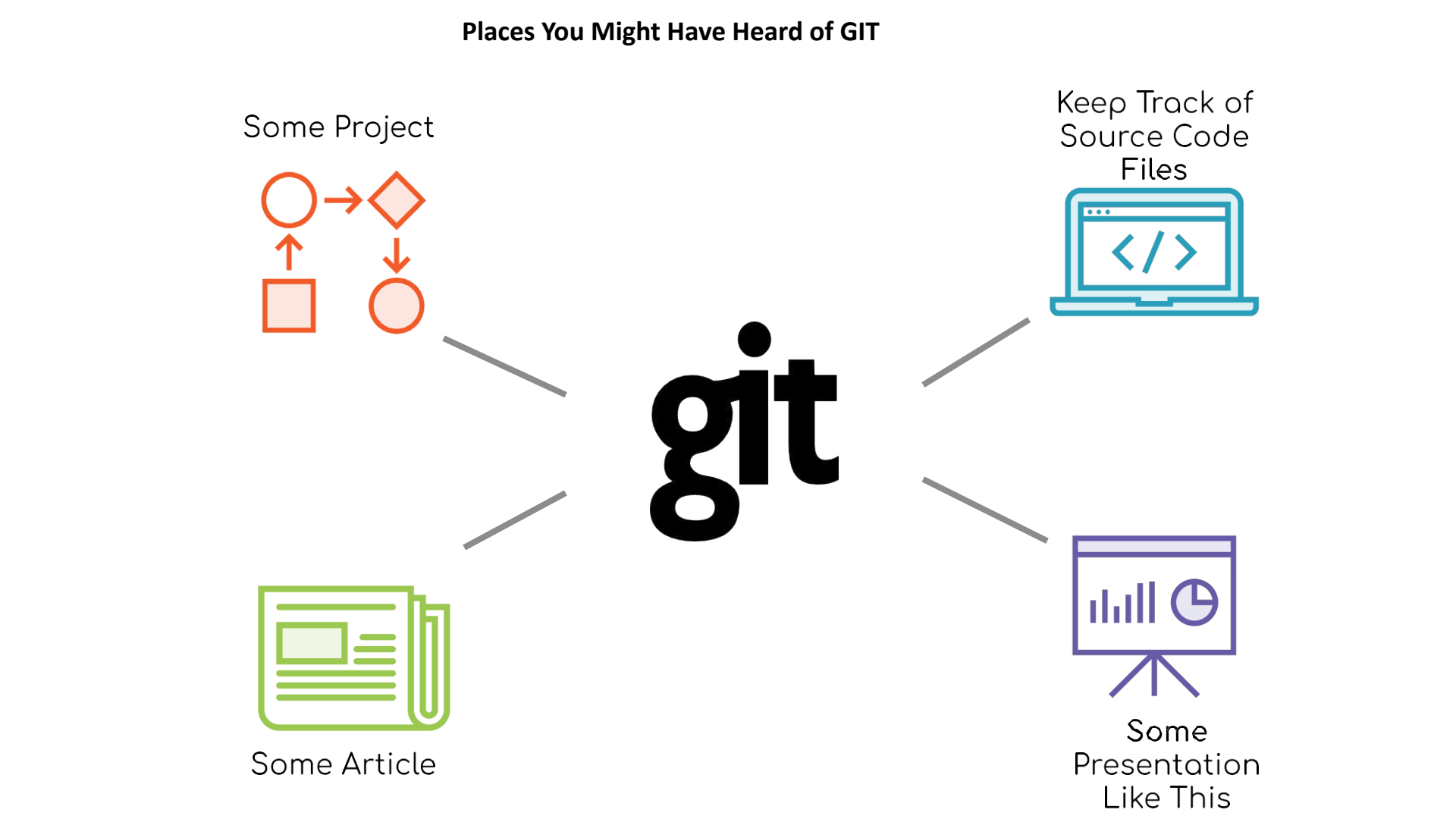
Getting Started with Git

WHAT IS GIT AND HOW DOES IT WORK?

By the end of this session, you will be able to understand why Git is such a powerful and popular tool among the software developers, team members and of course students, or anyone who wants to keep track of changes to any type of plain text file project.

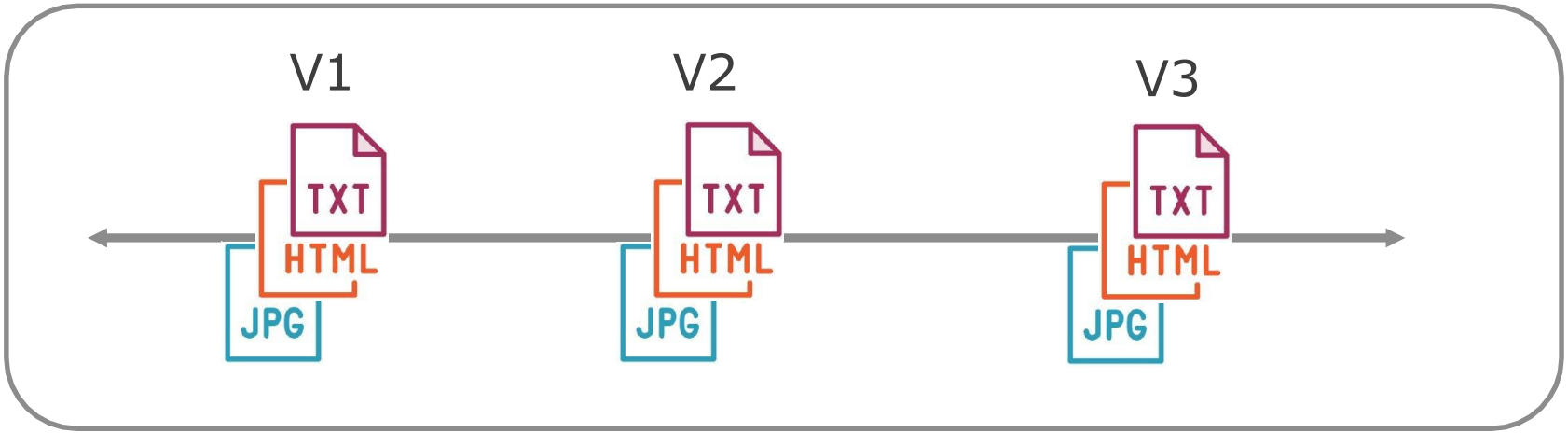
Places You Might Have Heard of GIT



What is Git?

Version Control System (VCS)

- Software designed to record changes made to files over time.
- Ability to revert back to previous file version or project version.
- Compare changes made to files from one version to another.
- Version control any plain text file, not just source code.

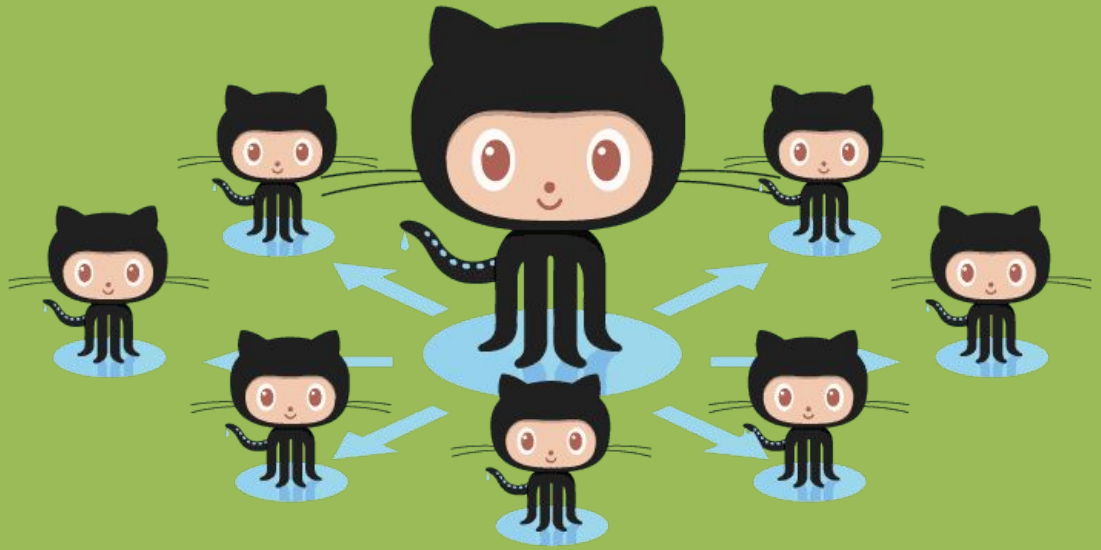


Separate Technologies

Complement each other very well

GitHub

GitHub is a web-based **version-control** and collaboration platform for software developers.



GitHub

Basic Terminologies

Repository

A repository is the most basic element of GitHub. They're easiest to imagine as a project's folder. A repository contains all of the project files (including documentation), and stores each file's revision history. Repositories can have multiple collaborators and can be either public or private.

Basic Terminologies

Fork

A fork is a personal copy of another user's repository that lives on your account. Forks allow you to freely make changes to a project without affecting the original. Forks remain attached to the original, allowing you to submit a pull request to the original's author to update with your changes. You can also keep your fork up to date by pulling in updates from the original.

Basic Terminologies

Clone

A clone is a copy of a repository that lives on your computer instead of on a website's server somewhere, or the act of making that copy. With your clone you can edit the files in your preferred editor and use Git to keep track of your changes without having to be online. It is, however, connected to the remote version so that changes can be synced between the two. You can push your local changes to the remote to keep them synced when you're online.

Basic Terminologies

Pull request

Pull requests are proposed changes to a repository submitted by a user and accepted or rejected by a repository's collaborators. Like issues, pull requests each have their own discussion forum. For more information, see "About pull requests."

git init

Usage: git init [repository name]

git clone

Usage: git clone [url]

git add

Usage: git add [file]

git commit

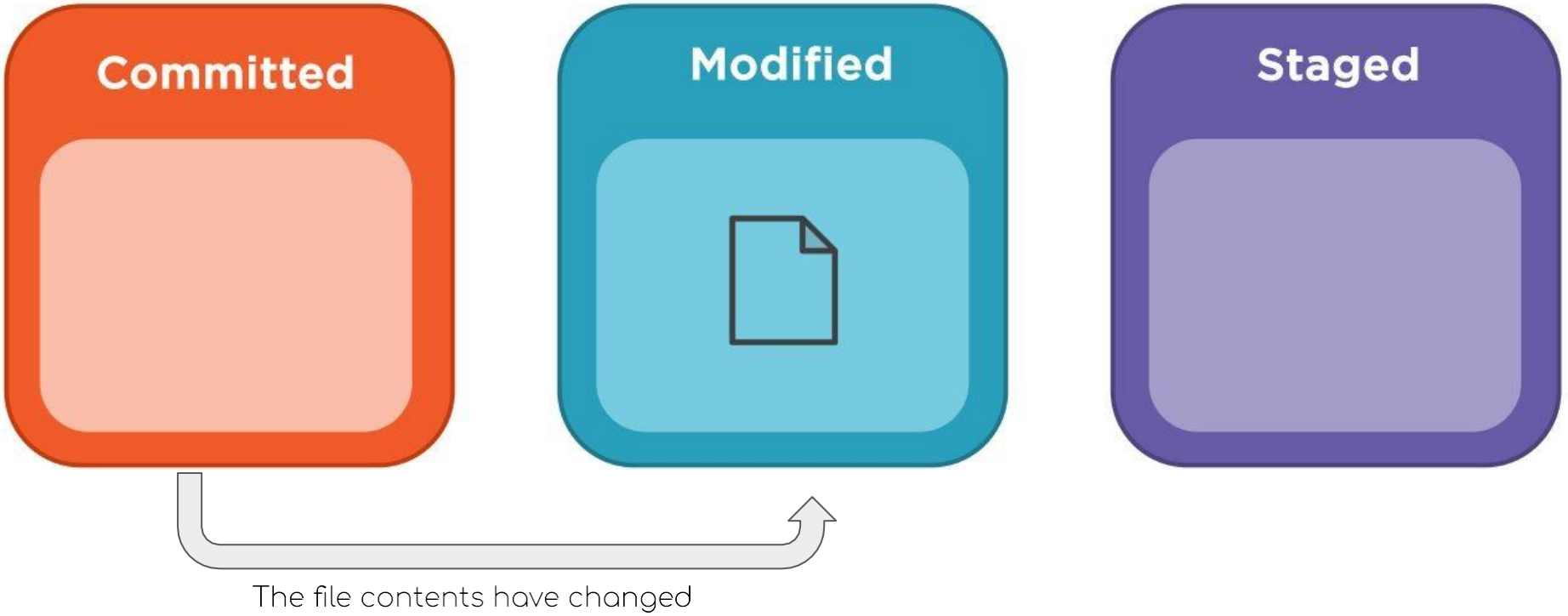
Usage: git commit -m “[Type in the commit message]”

git status

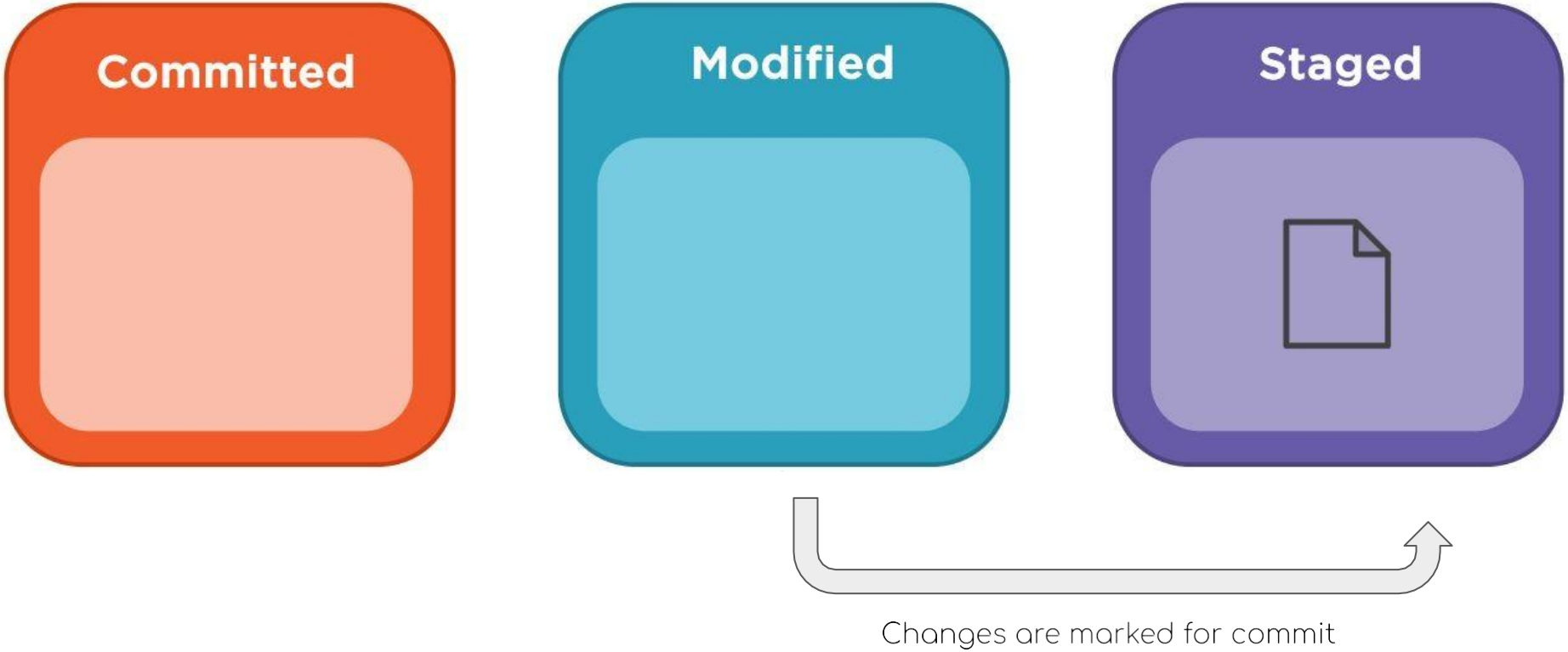
Usage: git status

3 stages of a file

3 stages of a file



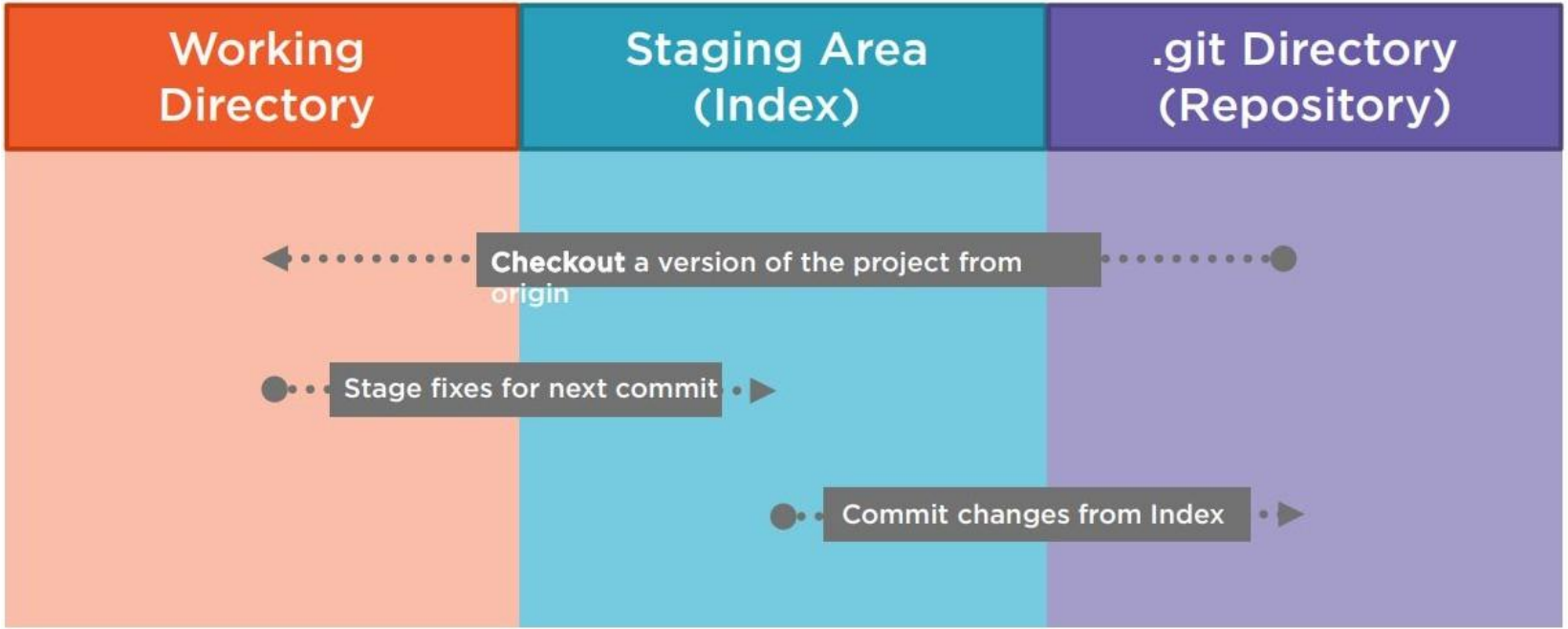
3 stages of a file



3 stages of a file

Marked changes have been added to the commit snapshot

3 stages of a git project



3 stages of a git project

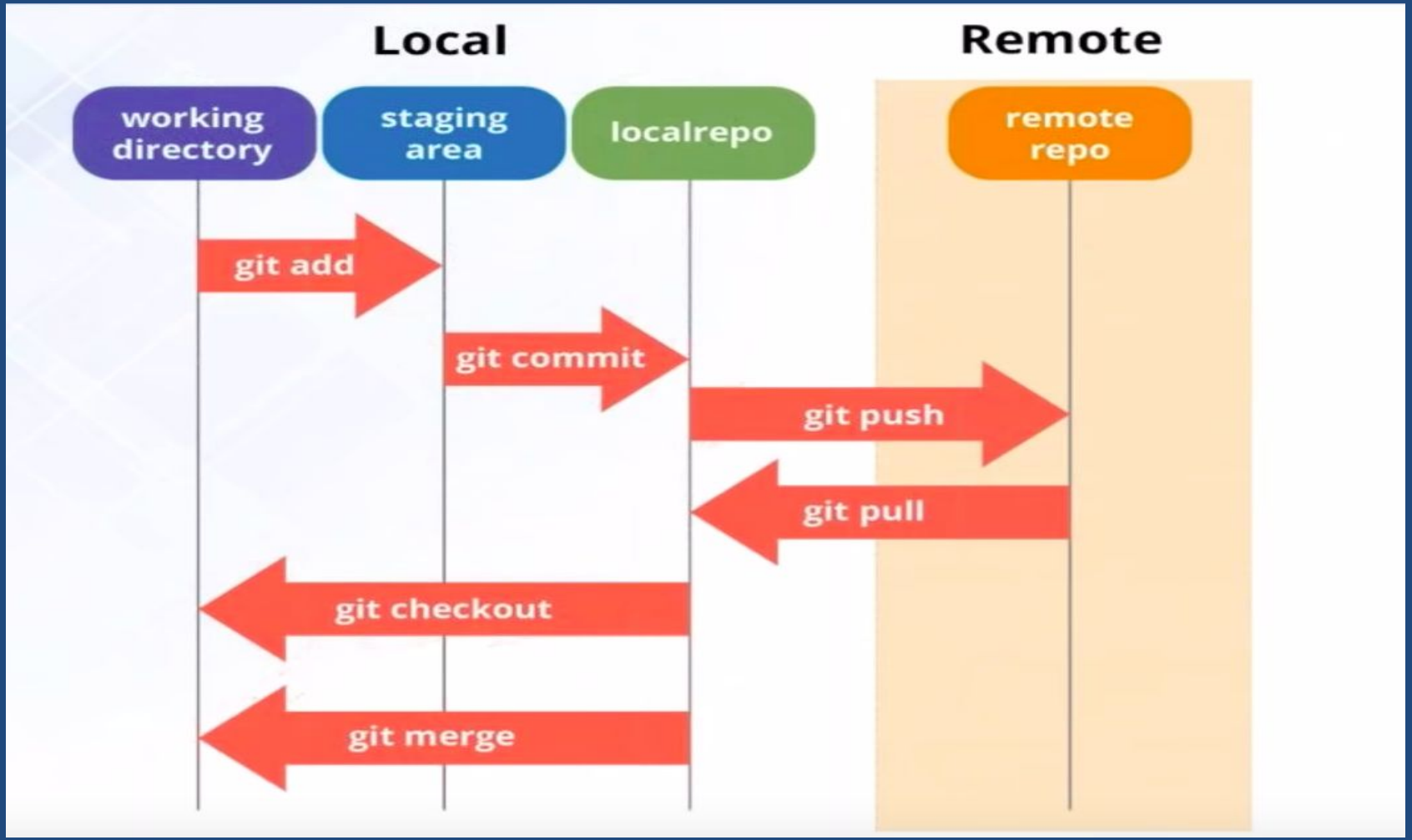
// TODO

- Install Git (Hopefully you guys did it)
- Configure Git
- Initialize a new project
- Create a github account
- Push our Git project to github

git config

git config --global user.name "BHAVYA HODA"

git config --global user.email "bhavyhoda@gmail.com"


Local
Remote
working directory
staging area
localrepo
remote repo
git add
git commit
git push
git pull
git checkout
git merge

```
git remote add origin "<>"

git push -u origin master
```

git log

Check commit history

Reverse Chronological Order

```
git log -2
git log --oneline
git log --stat
git log --patch
```

commit messages

https://chris.beams.io/posts/git-commit/

branches

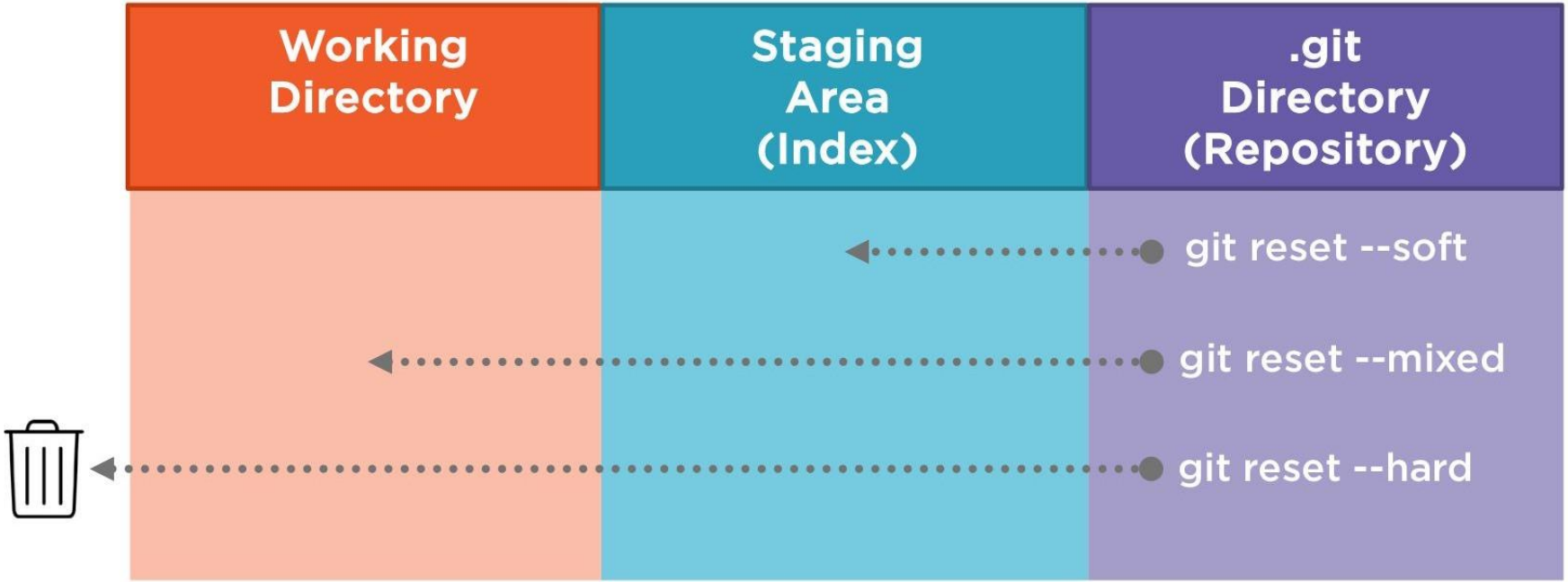
Visualization: https://git-school.github.io/visualizing-git

git branch <new_branch_name>

git checkout -b <new_branch_name>

git merge <branch_name_to_merge_into_current_active_branch>

git reset


Working Directory
Staging Area (Index)
.git Directory (Repository)
git reset --soft
git reset --mixed
git reset --hard