Getting Started with GIT

greatlearning
Power Ahead

Source Code Management - GIT

1 : What is Version Control

2 : Why do we need a VCS?

3 : What are the different VCS available?

4 : Git Overview

5 : Getting started with Github

6 : Git Installation & Configurations

Source Code Management - GIT

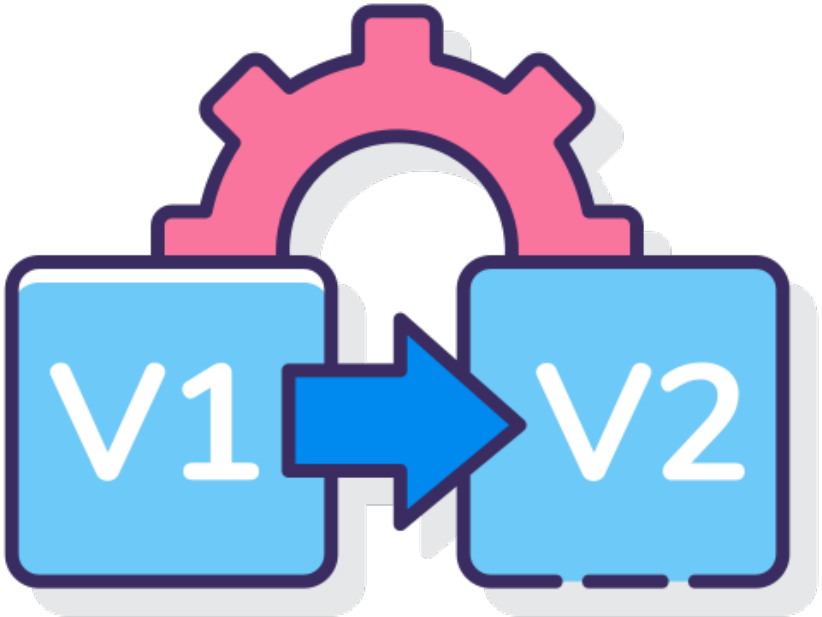
7 : Git basic commands

8 : branches

9 : Merging

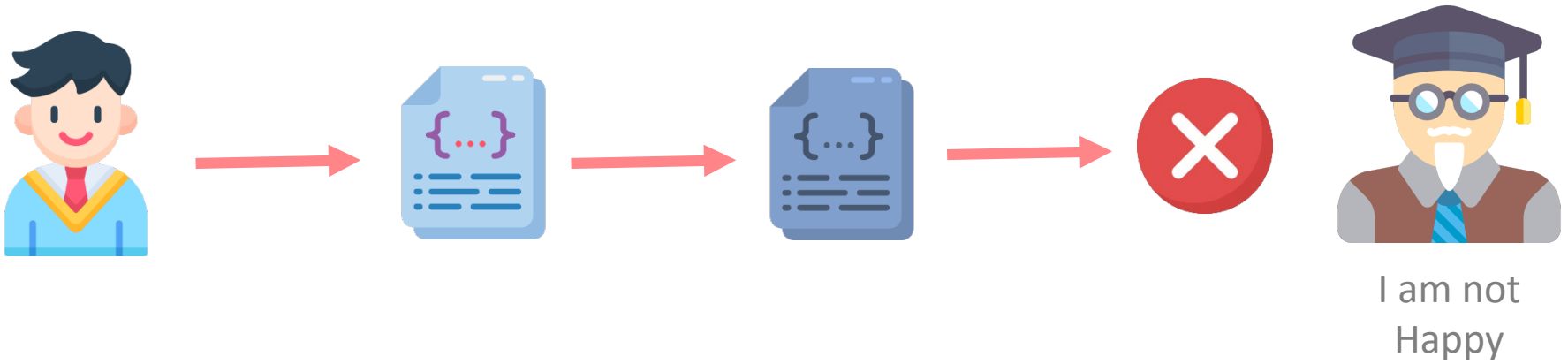
10 : Git workflow

11 : Git Best Practices

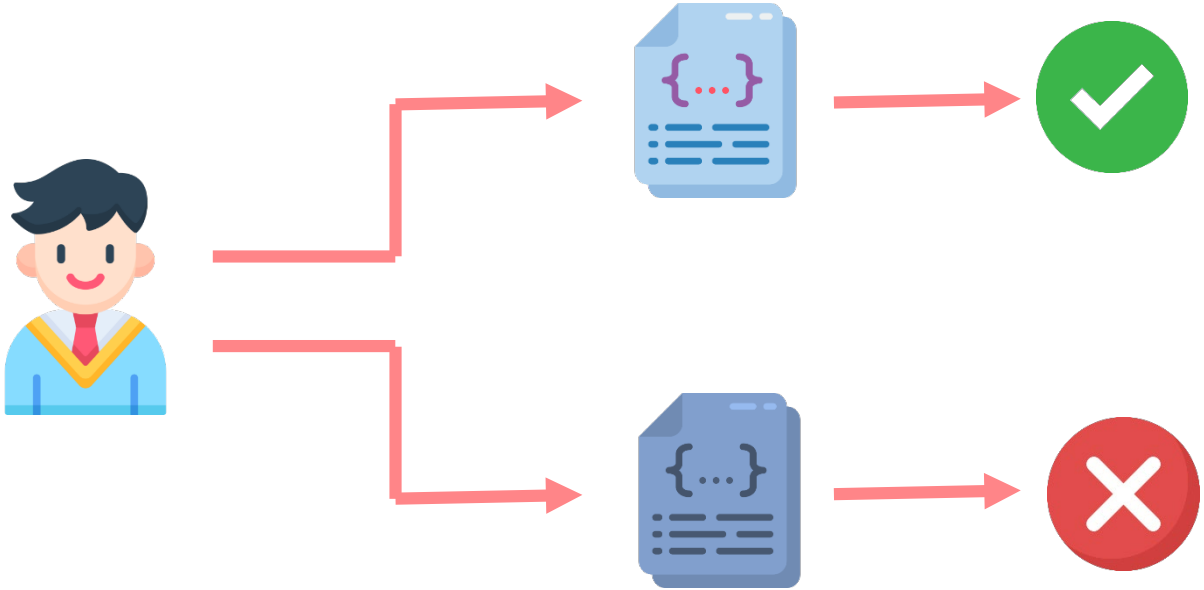


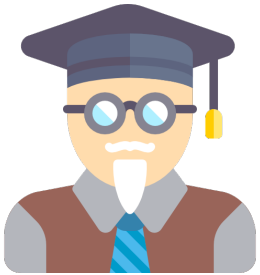
What is Version Control?

What is Version Control?



What is Version Control?





I am Happy

What is Version Control?

These days when software is developed, It is not developed with the mind-set that there will only be one piece of code that will be deployed and that's it. These days smaller snippets of code are deployed in regular successions with regular feedbacks. This leads to many different versions of the code.

And that creates a need to organise the code and all of its different version of it. This is where Version Control comes in. It is a practice of managing and storing different version of a source code.

This is especially the case with Larger companies that have multiple projects and multiple teams working within it.

What is Version Control?

A1, A2, A3

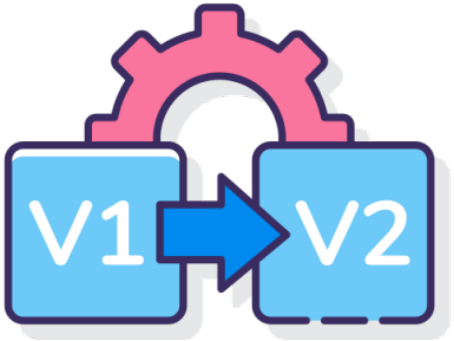
B1, B2, B3

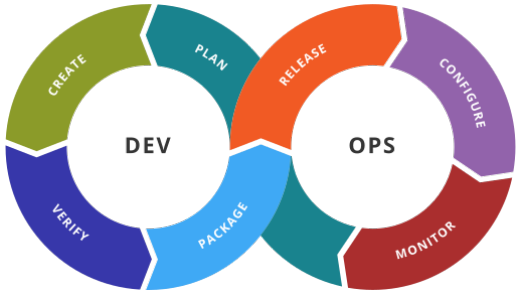
C1,C2,C3

Why do we need a VCS?

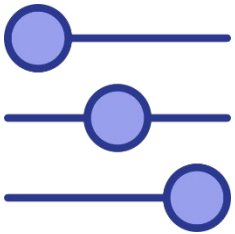
Why do we need a VCS?

Just as we discussed, the purpose of a version control system is as the name suggests, It controls the different versions of the code. The idea of a VCS meshes very well with the DevOps ideology. It allows for quick delivery of code and automatic job triggering. This allows for easy automation of the whole software development pipeline.





Benefits of VCS

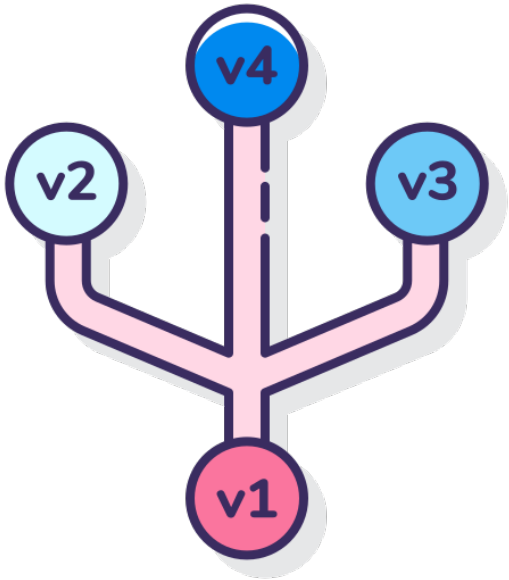


Control

Security

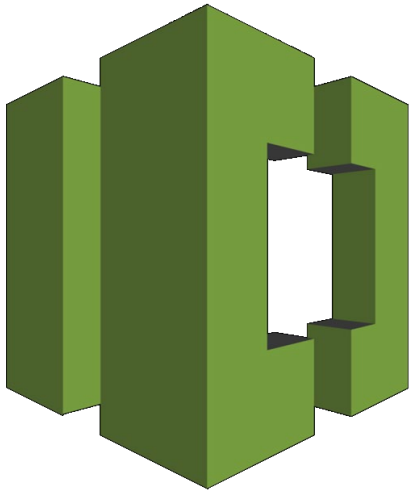
Fast
Delivery

Availability



What are the different VCS available?

What are the different VCS available?

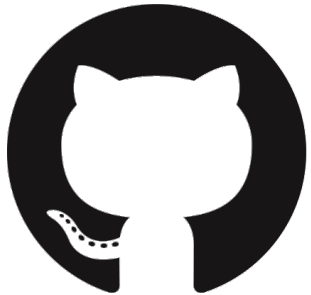


greatlearning
Power Ahead

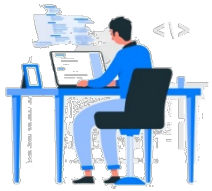
Git Overview

What is Git?

Git (global information tracker) is an open source distributed version control system or a source code management system. . It was developed by a group of Linux kernel developer who were not satisfied with any free source code manages that were available to them back in 2005.



What is Git?

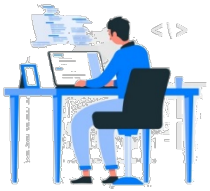


Developer 1

git

GitHub

git

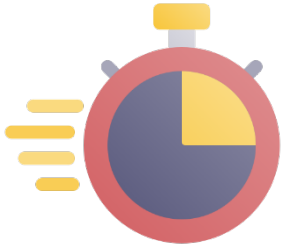


Developer 2

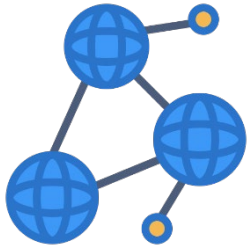
Git Features

System Compatibility

Collaboration



Speed

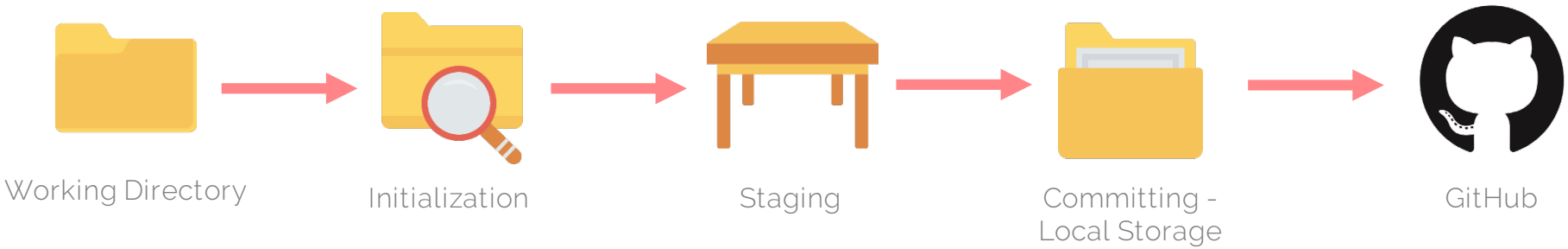


Distributed System & Reliability

Security

What is Git?

The lifecycle of the code within Git.



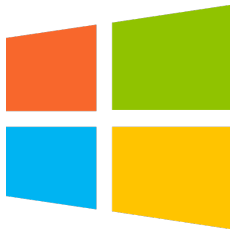
Getting started with GitHub

Installing & Configuring Git

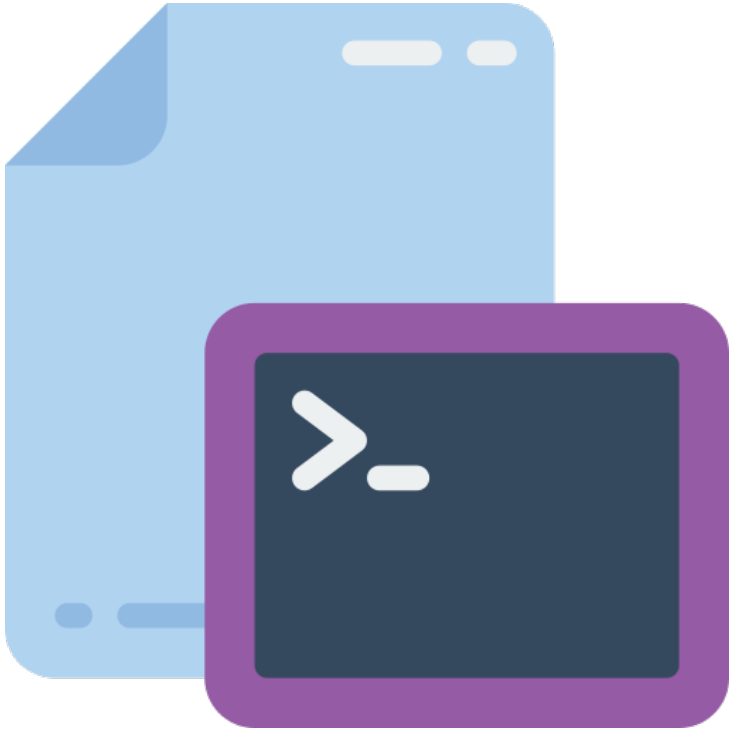
Git Installation & Setup

Linux

Windows



MAC

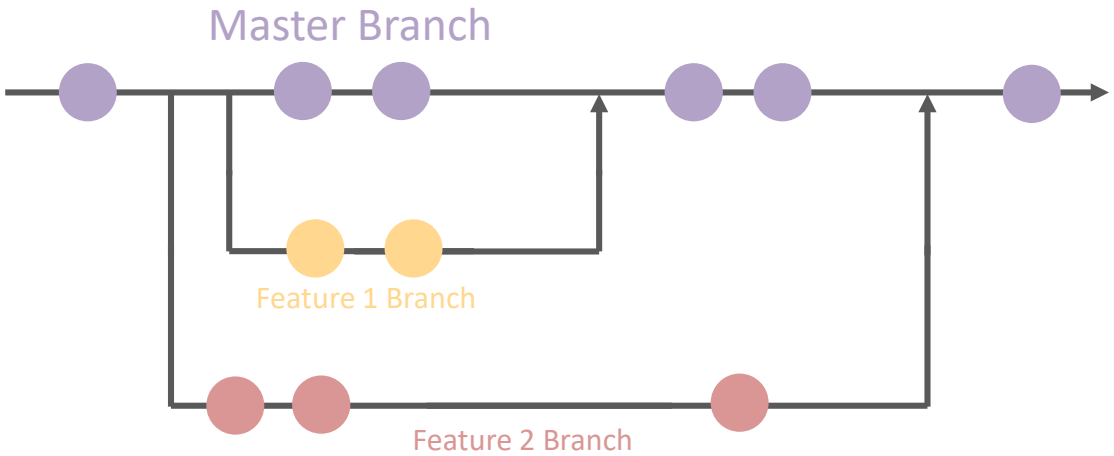


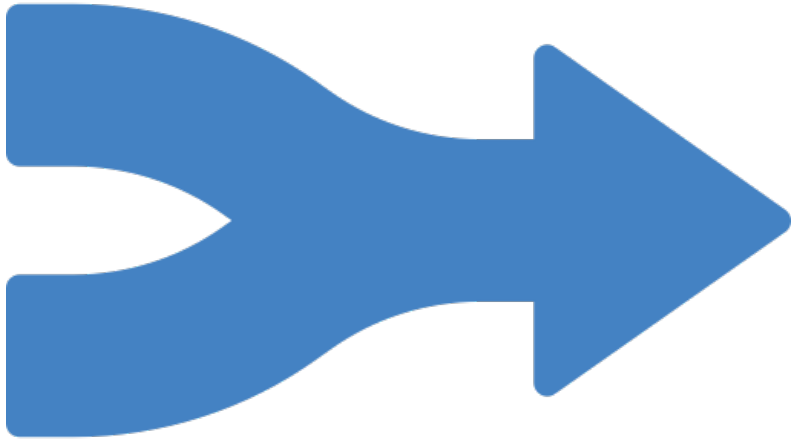
Git Basic Commands

Branches

Branches

Branches allow easy management of the different parts of a software, for e.g. Let's say your team is working on delivering a website. This website will have different features right? Each of these features will be counted as branch that will be merged onto the master branch. The master branch will be the one that goes into production.

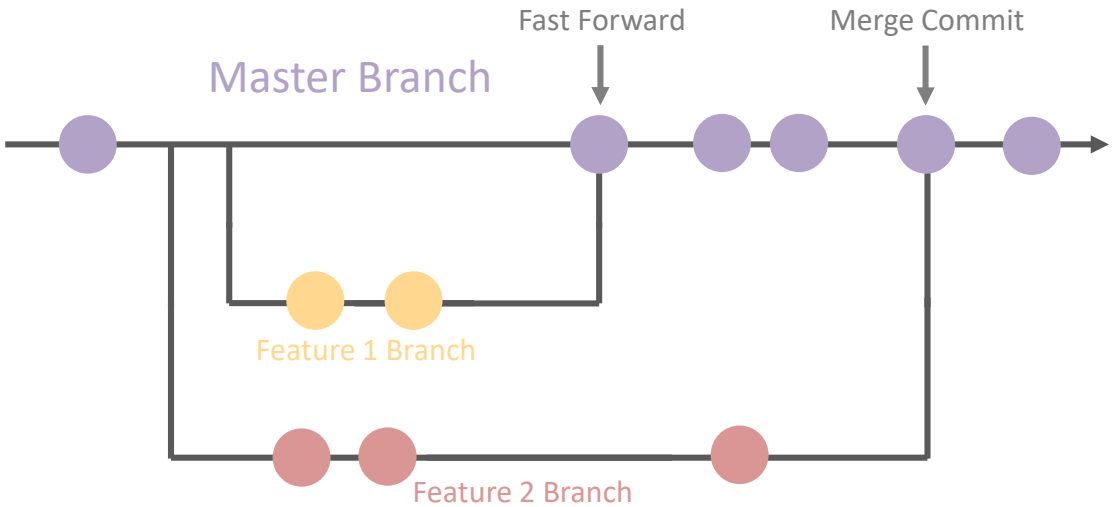




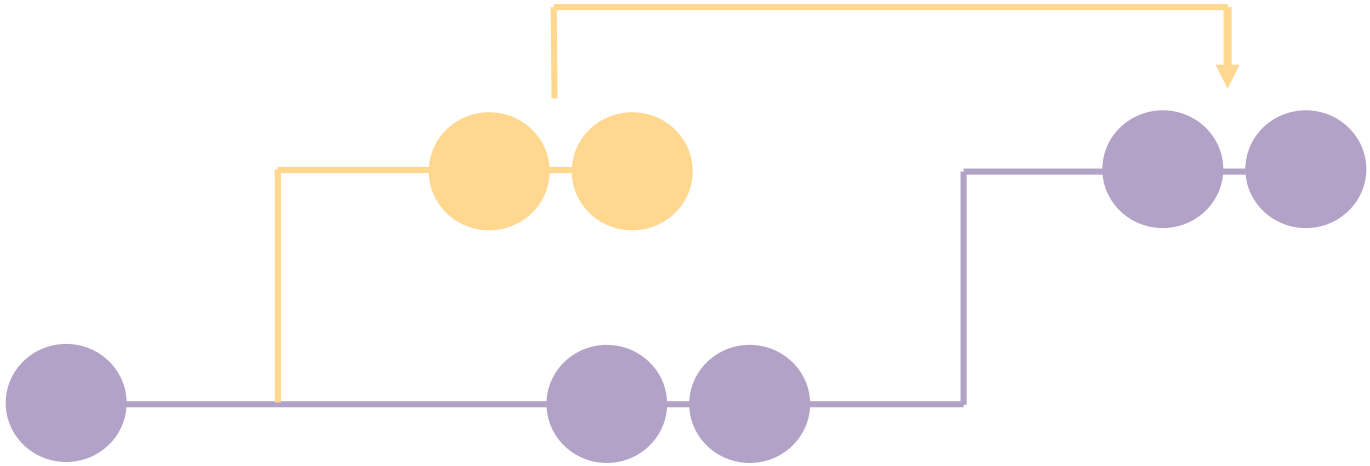
Merging

Merging

The git merge allows a user to merge different branches into one. It will create a new commit on merging. Here the history of commits are not transferred.



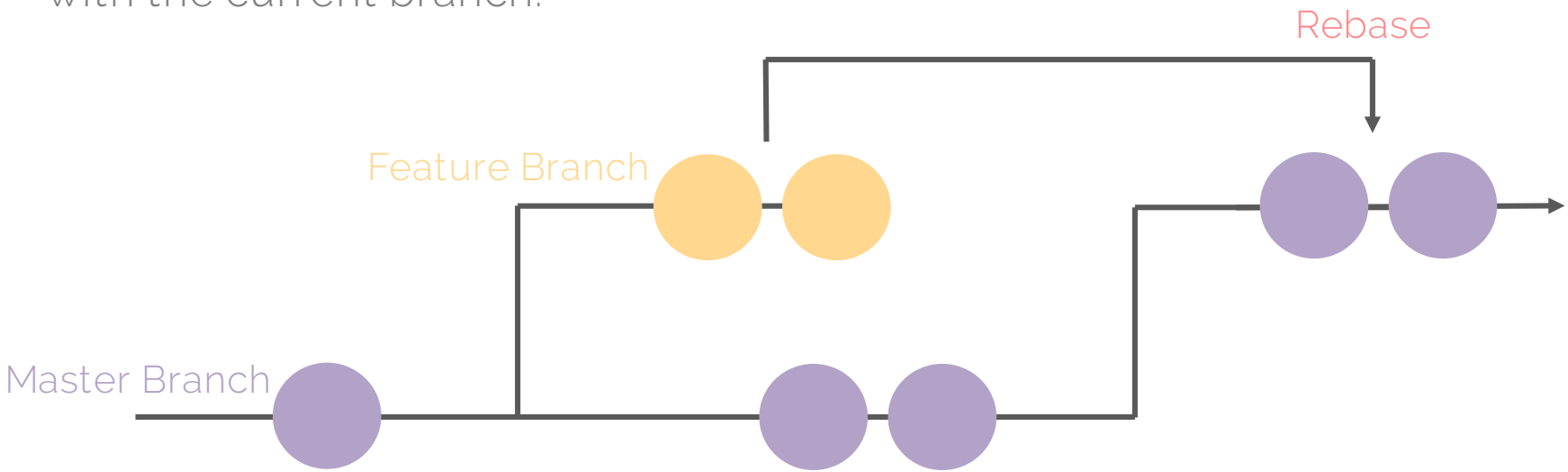
Git Merge Conflicts

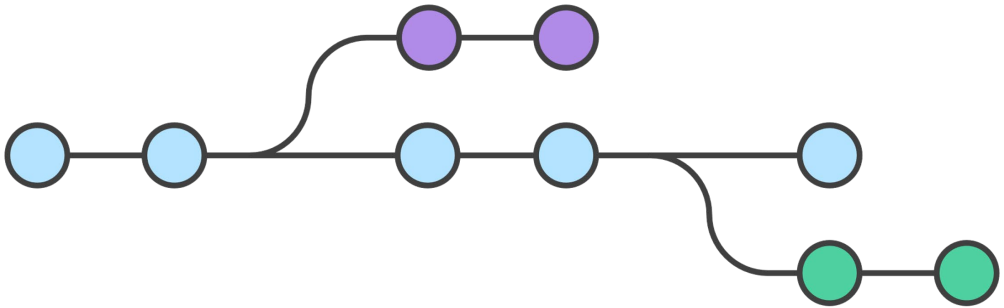


Rebasing

Rebasing

The git rebase command allows a user to shift a set of commits and give them a new base. Its similar to git merge but the difference here is that the history of commits of the branch is also carried over when it is joined with the current branch.

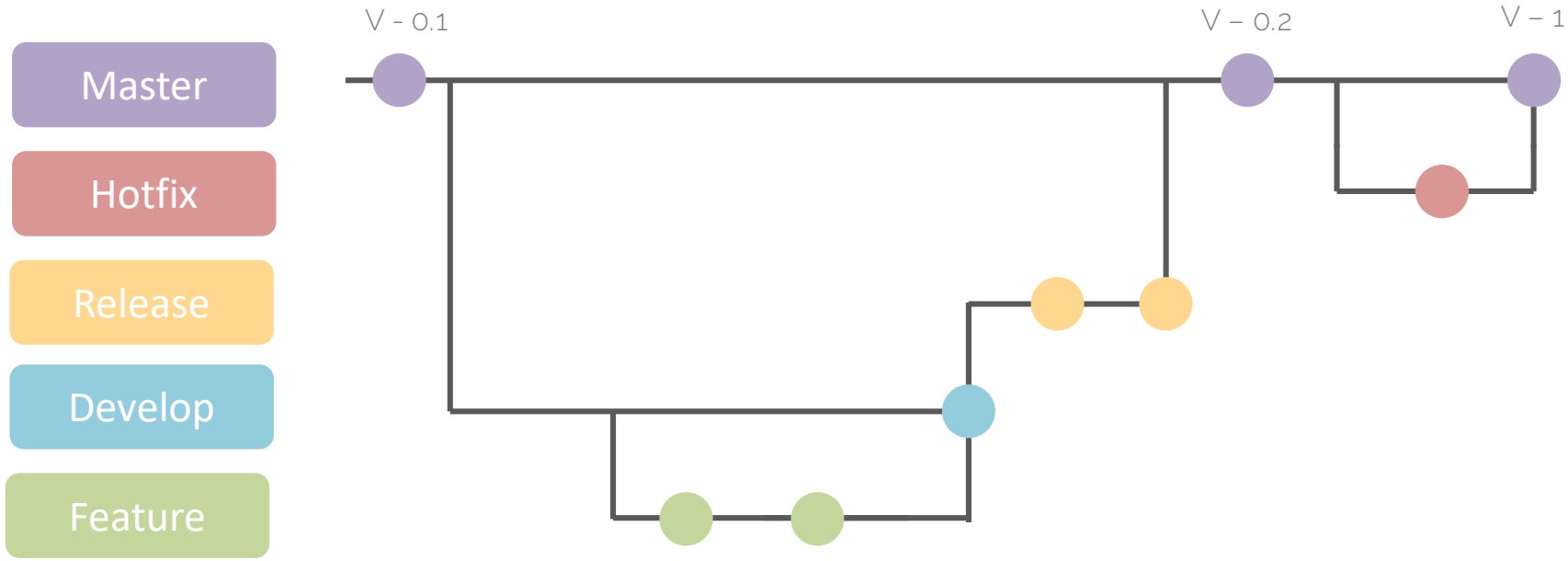




Git Workflow

Git Workflow

Git is a powerful tool, leveraging it effectively in software development is important. This is where the approach of the git workflow comes in. git needs to be designed.



Git Best Practices

greatlearning
Power Ahead

Git Best Practices

- Write Relevant Commit Messages
- Have Focused Commits
- Commit Frequently
- Only Commit Relevant Files
- Never Alter the Published History
- Rebase Frequently to stay updated
- Have Standards
- Avoid Committing very large files to GitHub