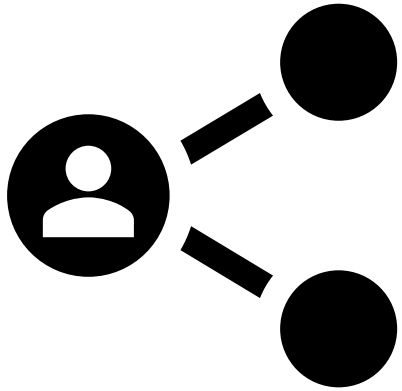
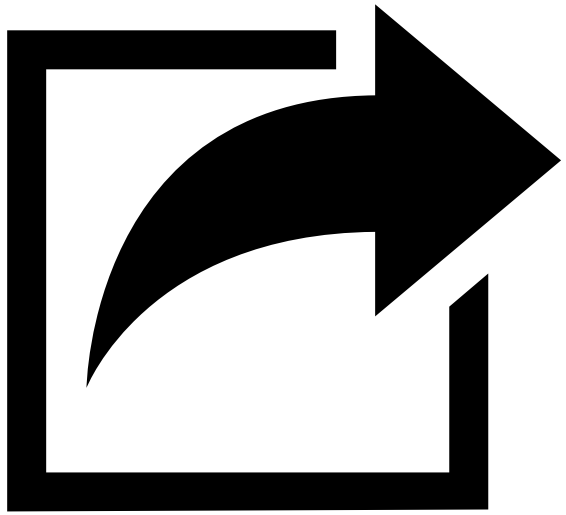
GitHub Essentials

By Gourab Sahu

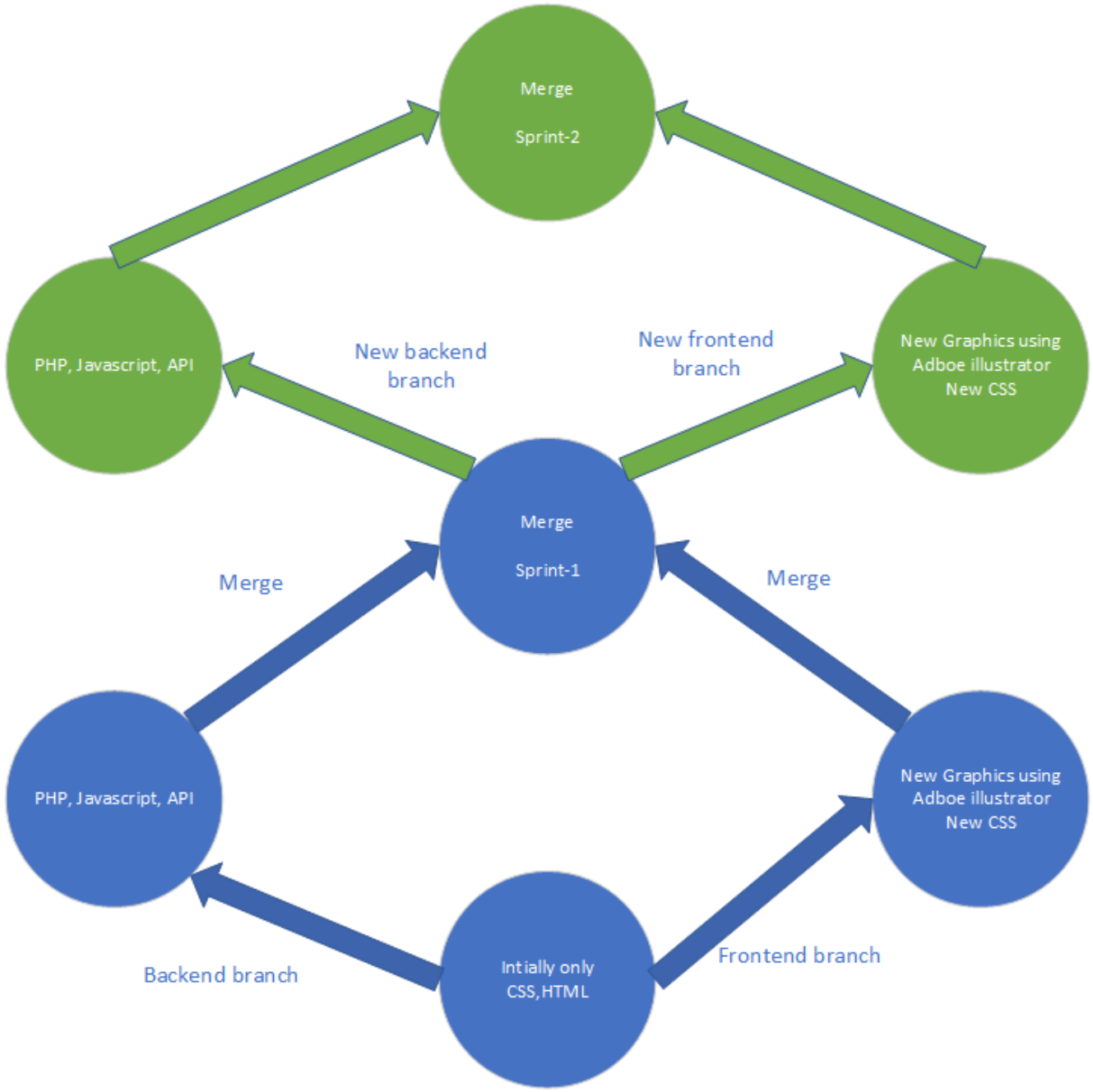
Why use GitHub?

- Version Control
- Software Backup / Sharing data





A small example of *version control* in web development.



Important Git commands in 4 categories

Installation

Saving Changes

Collaboration

Branches

Installation

- **git init**

  Initializes a normal directory as a git directory.

- **git clone „link“**

  First runs the git init command internally and then downloads a copy from remote repo to local direcotry.

- git config

  To setup configuration at local or global level.

Saving Changes

- **git add**

  Adds files into staging area.

- **git commit**

  If the staged changes are satifactory then this command commits the new changes into the project history.

- **git stash / git stash pop**

  Not sure about the current changes/ Unfinished work.

- **git ignore**

  Log files or System variables or Security information should not be tracked.

Collaboration

- **git remote**

  Every repo has 1 compulsory remote repo called origin.

- **git fetch**

  Gives a temporary head to check changes in the remote repo. *Safer than git pull.*

- **git pull**

  = git fetch + git merge

- **git push**

  Pushes locally committed changes to the remote repository.

Branches

- **git branch**

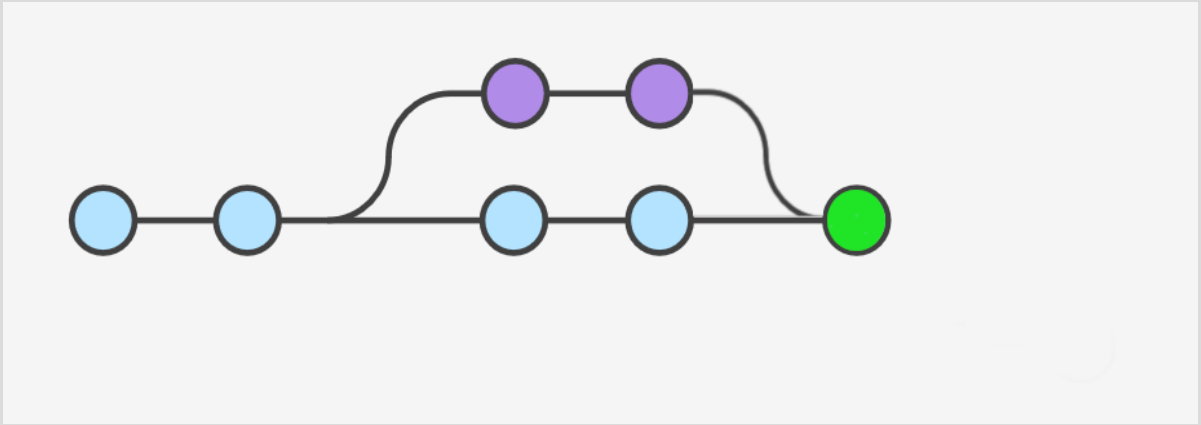
  To view/create existing/new branches.

- **git checkout**

  To navigate between the branches.

- **git merge**

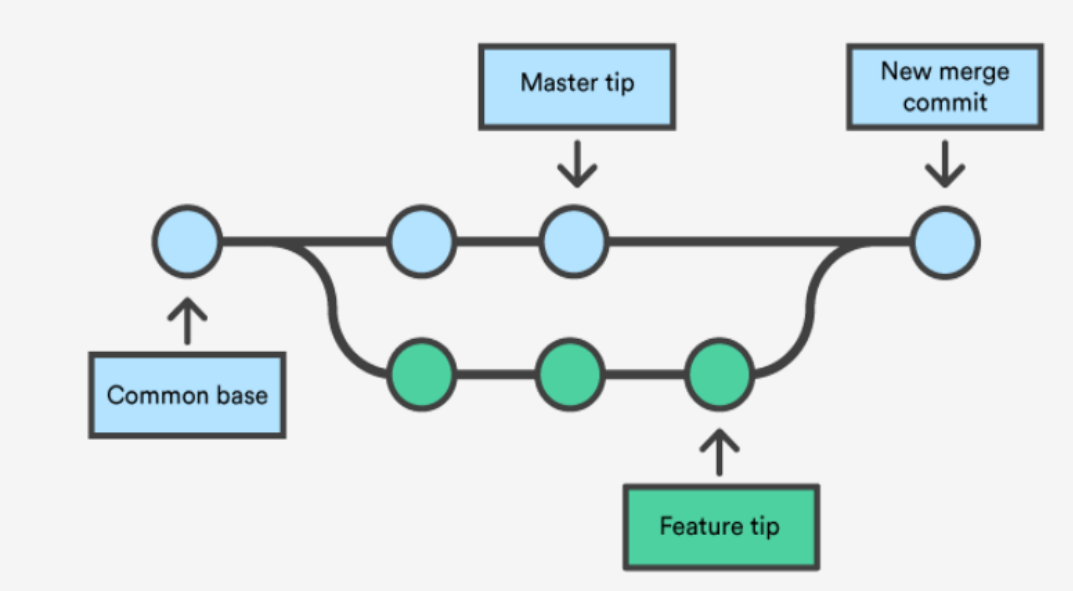
  Combines code from one branch into another.



How to resolve merge conflicts?

Conflict can happen in following two cases:

- During untracked changes
- Both branches have conflicting changes



How to resolve merge conflicts?

Conflicts can be resolved by opening the disputed files and keeping the changes asked by descrptive git texts such as follows:

```
$ cat merge.txt
<<<<<<< HEAD
this is some content to mess with
content to append
=======
totally different content to merge later
>>>>>>> new_branch_to_merge_later
```

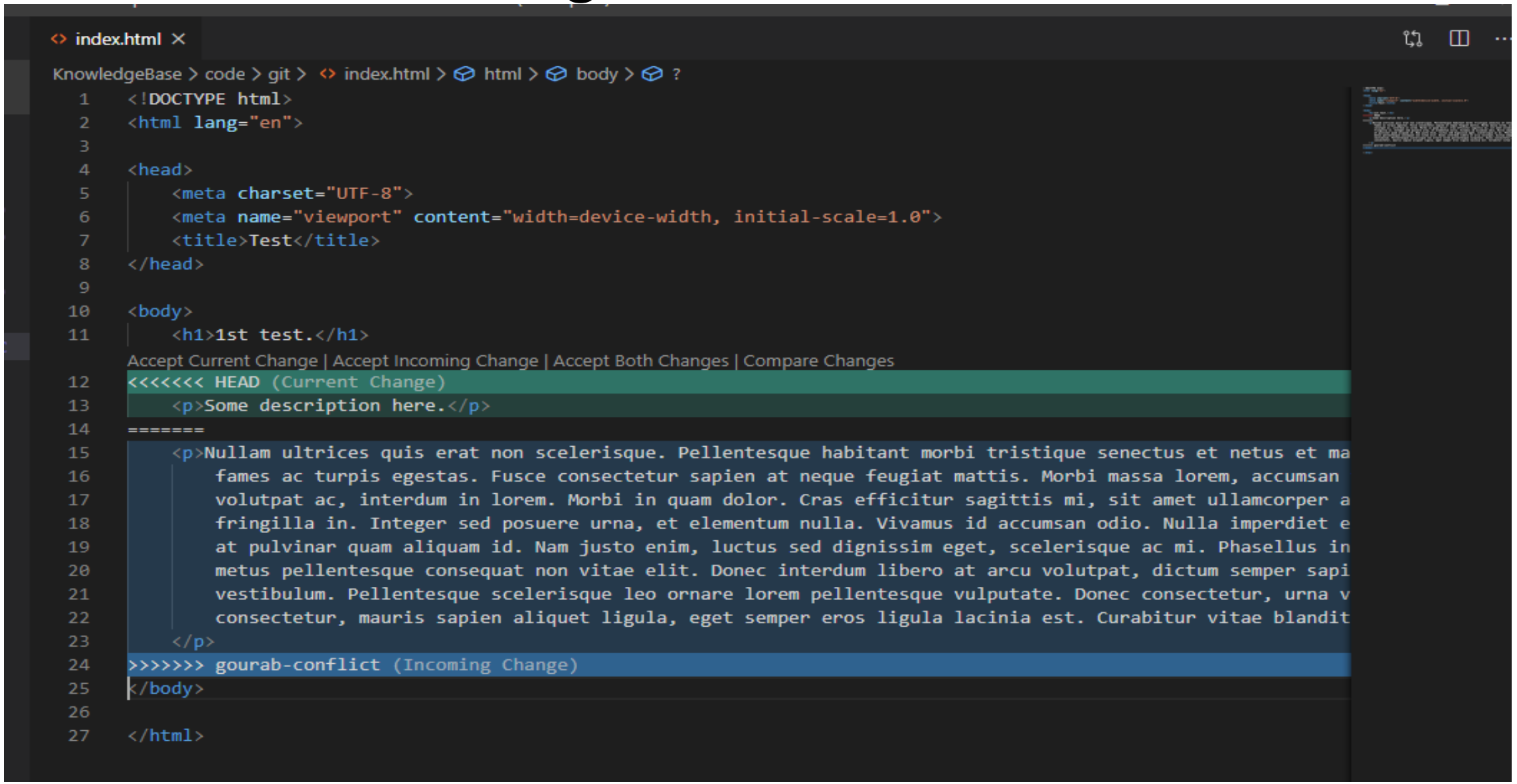
After making necessary changes **git add** and **git commit** can be used to finally merge the file.

How to resolve merge conflicts?

Conflicts in a disputed file can be resolved using any plain text editor.

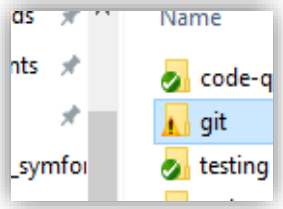
However it is recommended to use specialized tools such as **TortoiseGit** or **VSCode** if conflicts need to be resolved in multiple files.

How to resolve merge conflicts?

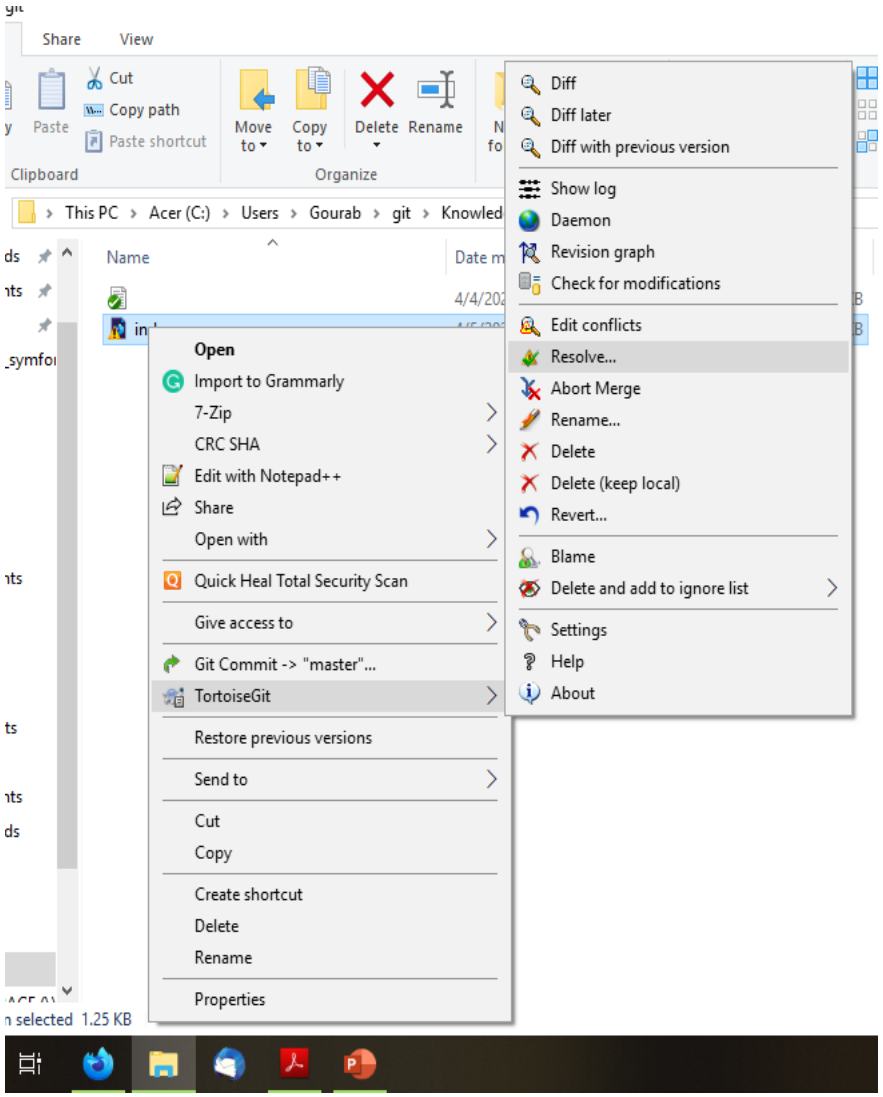


- Visual studio interface for conflict resolution. It also provides a visual in the right side with colored lines to reach them faster.

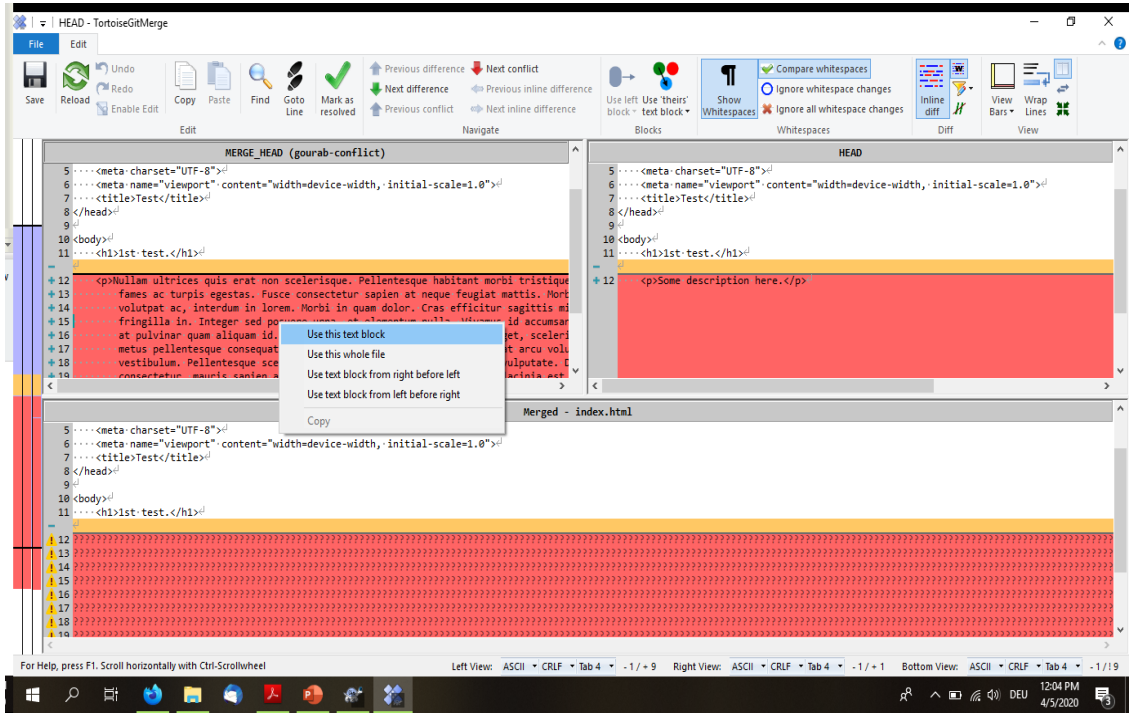
How to resolve merge conflicts?



TortoiseGit conflict notification.



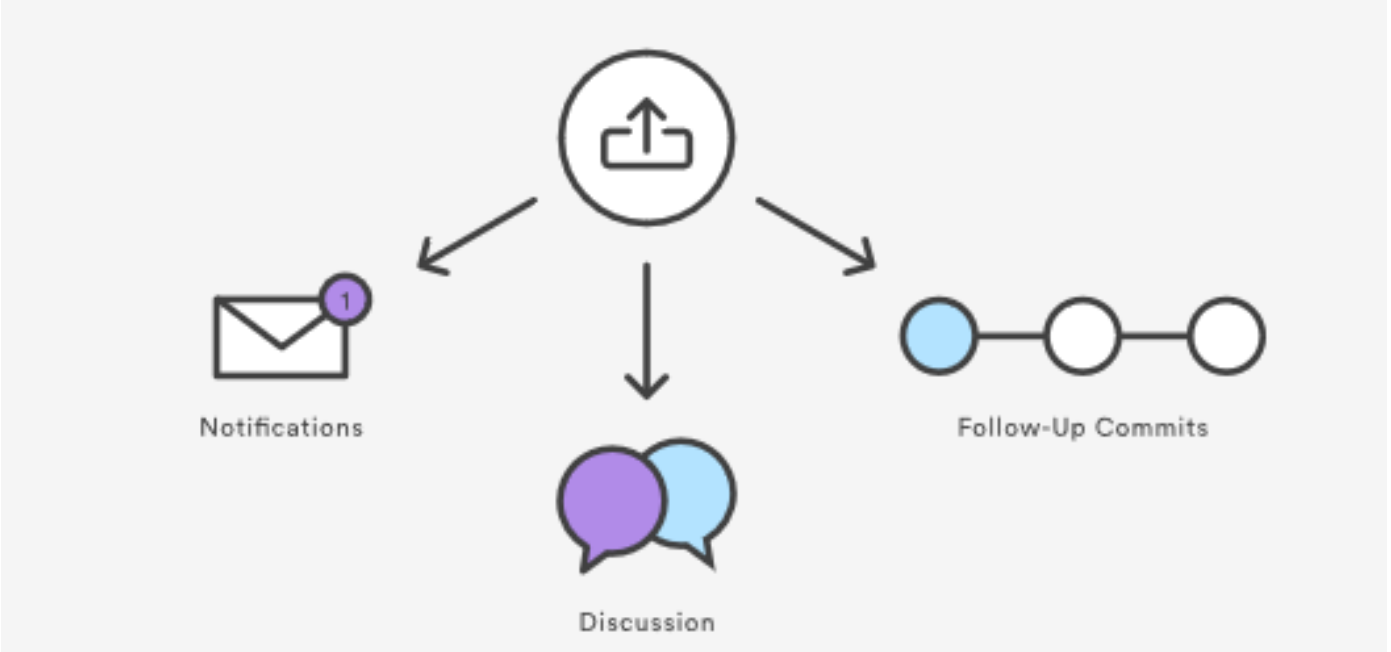
TortoiseGit user-menu.



TortoiseGit conflict resolution Interface

Difference between pull request and git merge?

- git merge is just a command to merge a feature branch into working/master branch.
- Pull request is a **notification** from a developer to another developer to merge a feature branch.
- Pull request also provides a platform for discusssion and code review by other team members.



git pull request

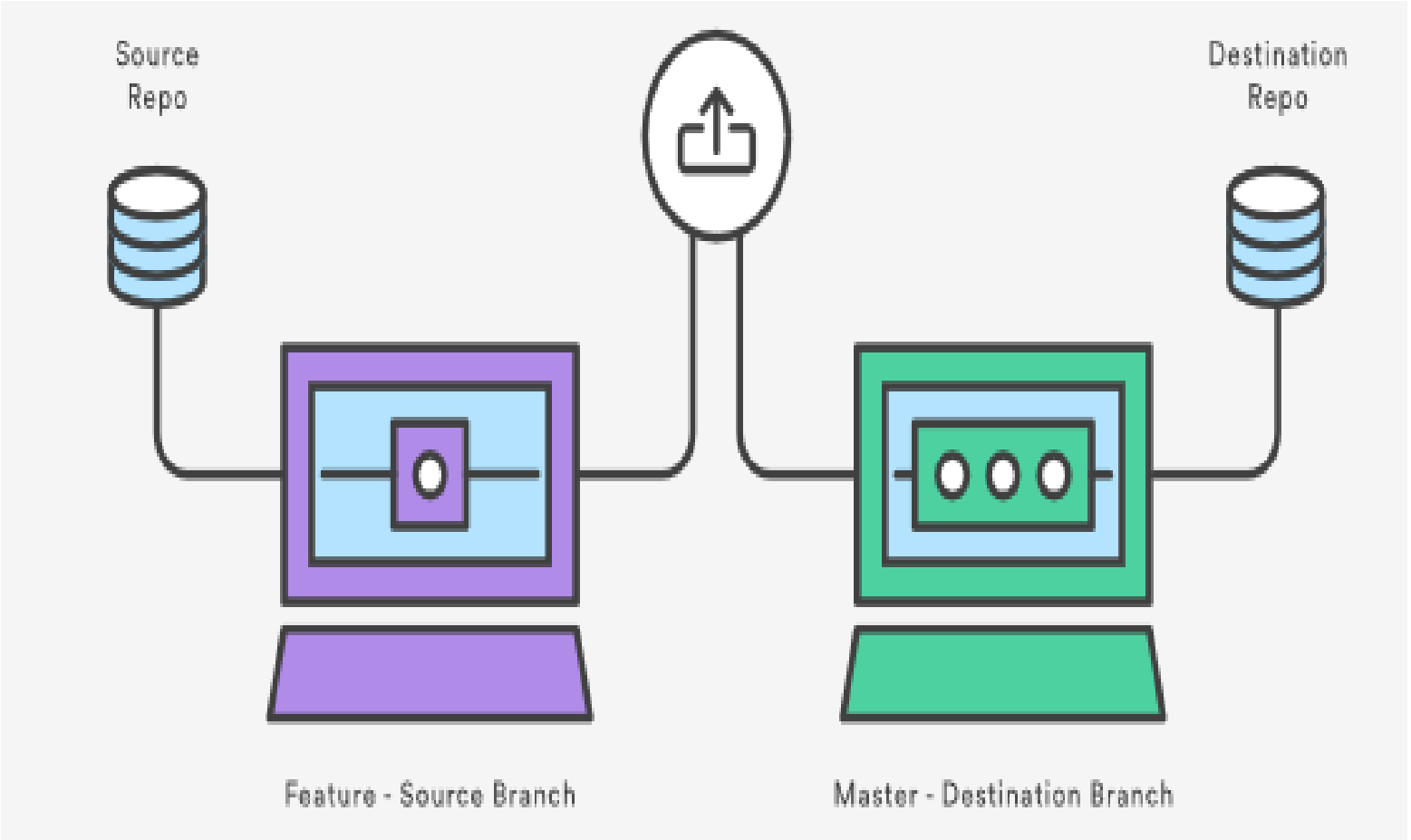
How pull request works?

1. A team-member creates a new branch in the same repo or in a forked repo.

2. The team-member adds new feature in the branch and makes a pull-request.

3. All team-members review new feature and provide comments.

4. Once approved, the git-master merges the requested branch into master branch.



.gitignore

Inside a git folder a file can be tracked, untracked or ignored.

Following category of files should not be tracked by git:

a) Javascript dependencies such as node_modules.

b) System generated files such as .DS_Store, Thumbs.db or .log files.

c) Security files with password or secret key.

To ignore any file following 2 steps can be used:

1. Make a file called .gitignore and put the name of the file to be ignored into it.
2. Or Use "git rm –cached file_to_ignore “ command.

References

- Atlassian. “Learn Git- Git Tutorials, Workflows and Commands: Atlassian Git Tutorial.” *Atlassian*, www.atlassian.com/git.

- “Reference.” *Git*, git-scm.com/docs.