Troubleshooting

- Error: Your configuration specifies a merge with the ref 'refs/heads/master' from the remote, but no such ref was fetched.

```
git branch --unset-upstream
```

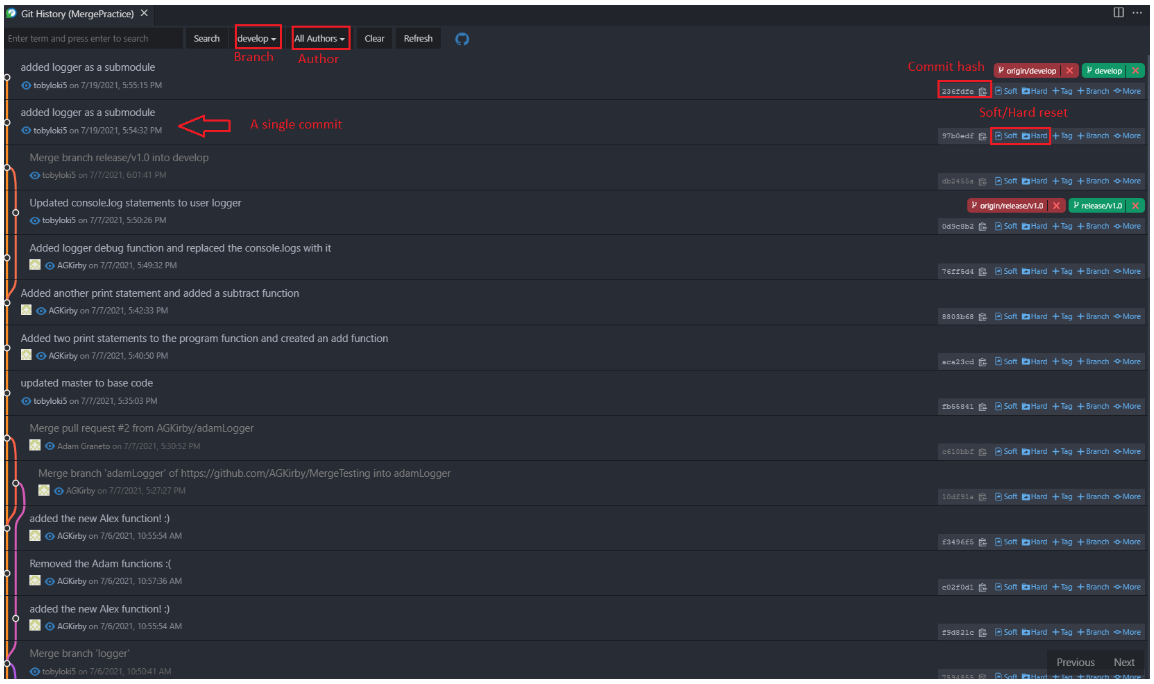
- Error: Undo the merge

```
git merge --abort
```

Initialize repo to current empty folder

```
git remote add origin <repo>
```

VSCode Extensions

- Git Lens: https://marketplace.visualstudio.com/items?itemName=eamodio.gitlens
    - Shows a hint on changes made to each line when you click on line of code and who made those changes and how long ago
    - In Source Control, you can use the Stash
- Git History: https://marketplace.visualstudio.com/items?itemName=donjayamanne.githistory
    - View entire git history
        - Hotkey: `Ctrl + Shift + P` → Git History
    - 

```
added logger as a submodule
tobyloki5 on 7/19/2021, 5:55:15 PM                                  236fdfe
added logger as a submodule
tobyloki5 on 7/19/2021, 5:54:32 PM                                  97b0edf
Merge branch release/v1.0 into develop
tobyloki5 on 7/7/2021, 6:01:41 PM                                   db2455a
Updated console.log statements to user logger
tobyloki5 on 7/7/2021, 5:50:26 PM                                   0d9c8b2
Added logger debug function and replaced the console.logs with it
AGKirby on 7/7/2021, 5:49:32 PM                                     76ff5d4
Added another print statement and added a subtract function
AGKirby on 7/7/2021, 5:42:33 PM                                     8803b68
Added two print statements to the program function and created an add function
AGKirby on 7/7/2021, 5:40:50 PM                                     aca23cd
updated master to base code
tobyloki5 on 7/7/2021, 5:35:03 PM                                   fb55841
Merge pull request #2 from AGKirby/adamLogger
Adam Graneto on 7/7/2021, 5:30:52 PM                                c610bbf
Merge branch 'adamLogger' of https://github.com/AGKirby/MergeTesting into adamLogger
AGKirby on 7/7/2021, 5:27:27 PM                                     10df91a
added the new Alex function! :)
AGKirby on 7/6/2021, 10:55:54 AM                                    f3496f5
Removed the Adam functions :(
AGKirby on 7/6/2021, 10:57:36 AM                                    c02f0d1
added the new Alex function! :)
AGKirby on 7/6/2021, 10:55:54 AM                                    f9d821c
Merge branch 'logger'
tobyloki5 on 7/6/2021, 10:50:41 AM                                  7594855
```

Git clone

- Description
    - Essentially downloads the repo, but more advanced than the green button that downloads a zip file.

- Commands
  - Clone repo (vanilla style)

```
git pull <repo>
```

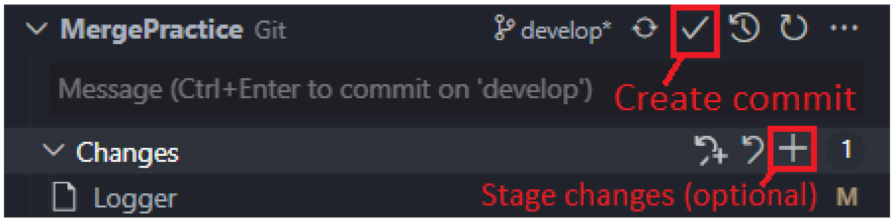
  - Clone repo at a specific branch (can always change branch later with git checkout)

```
git pull <repo> -b <branch>
```

  - Clone repo and pull all submodules too (submodules are not downloaded without the --recursive flag)

```
git pull --recursive <repo>
```

<u>Git commit</u>

- Description
  - Saves all your work into a commit. Commits are stored locally until you push.
- VSCode way
  - 

- Command line way

```
git add .
```

```
git commit -m "<message>"
```

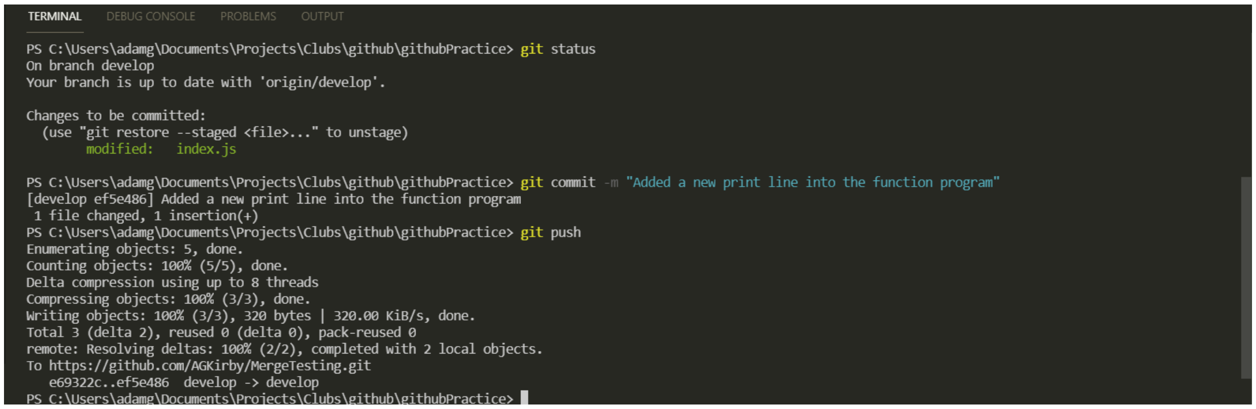
<u>Git push</u>

- Description
  - Add the changes you made and committed in your local repository to the remote repository.
- Commands

```
git push
```

```
git push [<repo> <branch>]
```

- With the terminal:

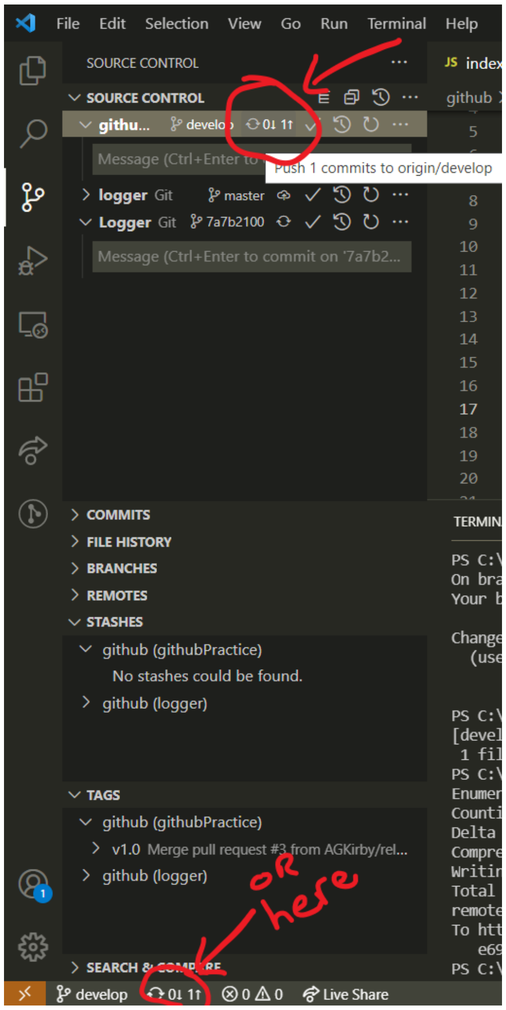


```
PS C:\Users\adamg\Documents\Projects\Clubs\github\githubPractice> git status
On branch develop
Your branch is up to date with 'origin/develop'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.js

PS C:\Users\adamg\Documents\Projects\Clubs\github\githubPractice> git commit -m "Added a new print line into the function program"
[develop ef5e486] Added a new print line into the function program
 1 file changed, 1 insertion(+)
PS C:\Users\adamg\Documents\Projects\Clubs\github\githubPractice> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 320 bytes | 320.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/AGKirby/MergeTesting.git
   e69322c..ef5e486  develop -> develop
PS C:\Users\adamg\Documents\Projects\Clubs\github\githubPractice>
```

- With VSCode GUI:


File Edit Selection View Go Run Terminal Help
SOURCE CONTROL
githu... develop 0↓ 1↑
Push 1 commits to origin/develop
logger Git master
Logger Git 7a7b2100
Message (Ctrl+Enter to commit on '7a7b2...
COMMITS
FILE HISTORY
BRANCHES
REMOTES
STASHES
github (githubPractice)
No stashes could be found.
github (logger)
TAGS
v1.0 Merge pull request #3 from AGKirby/rel...
SEARCH & COMPARE
develop 0↓ 1↑ ⊗ 0 ⚠ 0 Live Share
OR here

- Resources
  - https://git-scm.com/docs/git-push

Git pull

- Description
  - Pulls latest changes from remote repo to local

- Commands

```
git pull
```

- With the VSCode GUI, it is the same button as it is to push.
    - It will indicate that there is something to pull by having a 1 next to the down arrow on the left (instead of a 0 like in the picture above).
- Resources:
    - https://git-scm.com/docs/git-pull

Git checkout

- Description
    - Check out a specific branch in a repo.
- Commands

```
git checkout <branch>
```

Git Merge

```
git merge <branch>
```

- Merges the specified branch into the current branch

```
git merge --abort
```

- In the event of merge conflicts, this can be used to back out of the merge attempt

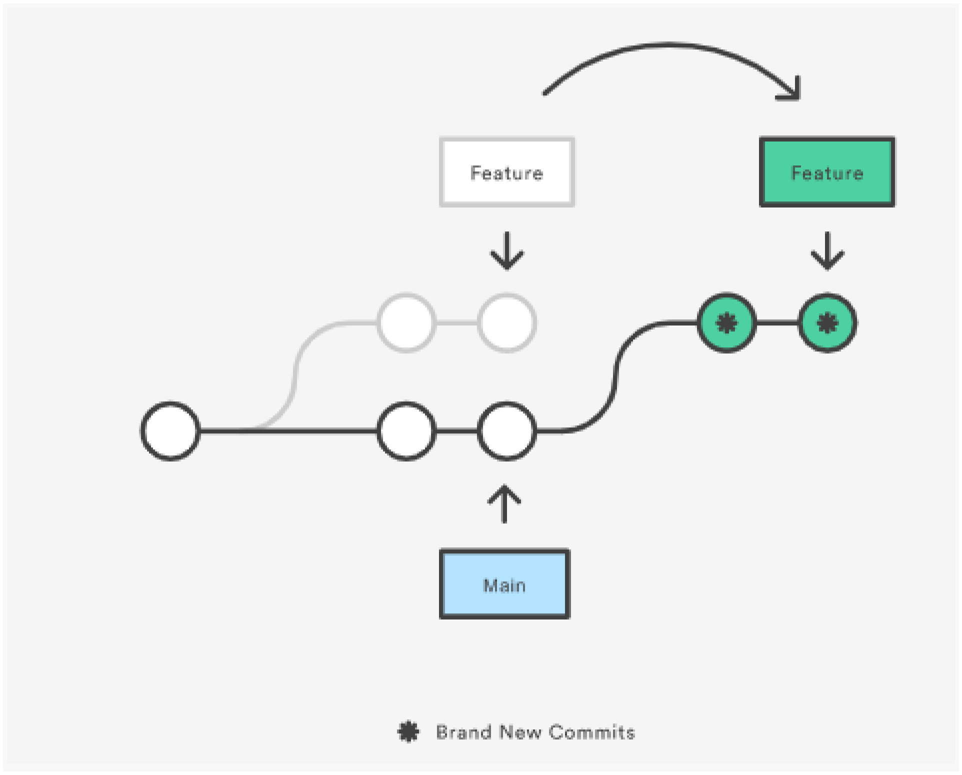
```
git merge --continue
```

- In the event of a merge conflict, you can have git attempt to solve the conflict by using this command, it will check if there is an interrupt, if not it will mark the conflicts, edit the files into a form that can be merged, and add them to the current index
- Resources
    - https://git-scm.com/docs/git-merge
    - https://www.atlassian.com/git/tutorials/merging-vs-rebasing

Git Rebase (use Git Merge instead if possible)

- Changes the base of a branch (the commit from which it was spawned off of).
    - Used to keep the project's branches ordered in a linear fashion.

```
git rebase <base>
```



- By rebasing a feature branch before merging back with the main branch, you can effectively pull upstream changes and avoid a merge conflict.
- You can use the interactive flag (`--i`) to alter individual commits during the process of rebasing
- Resources
    - https://www.atlassian.com/git/tutorials/merging-vs-rebasing

```
git reset --i <base>
```

    - However this can be dangerous because it can result in lost commits
- In the event of merge conflicts in the process of rebasing, you can use the `--continue` flag and the `--abort` flag to advance or reset the process of rebasing, respectively.
- https://www.atlassian.com/git/tutorials/rewriting-history/git-rebase

Git Soft/Hard Reset

```
git reset --soft <commit>
```

- Moves the pointer (head) to the specified commit
    - Index is not changed, so doing `git commit` immediately will restore the version prior to the reset

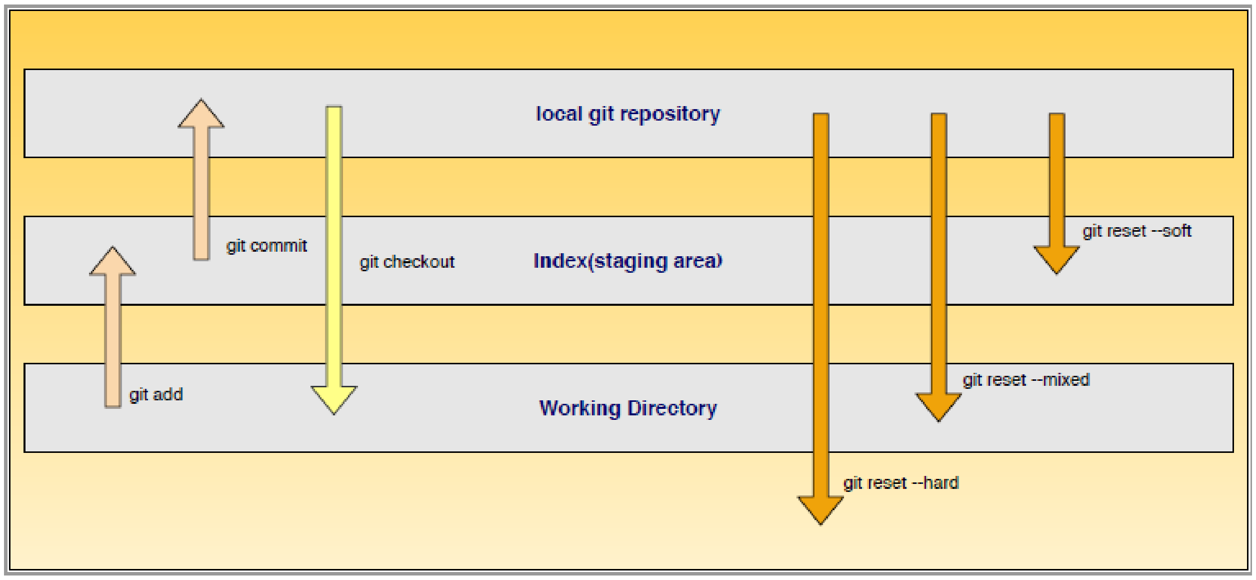
```
git reset -- mixed <commit>
```

- Mixed is the default option
- Moves pointer and adjusts the index to match
    - Therefore, doing a git commit immediately afterwards will not do anything

- Changes made before the reset will still be saved in the working directory, so they can be restored by doing a `git add .` and then a `git commit`

```
git reset --hard <commit>
```

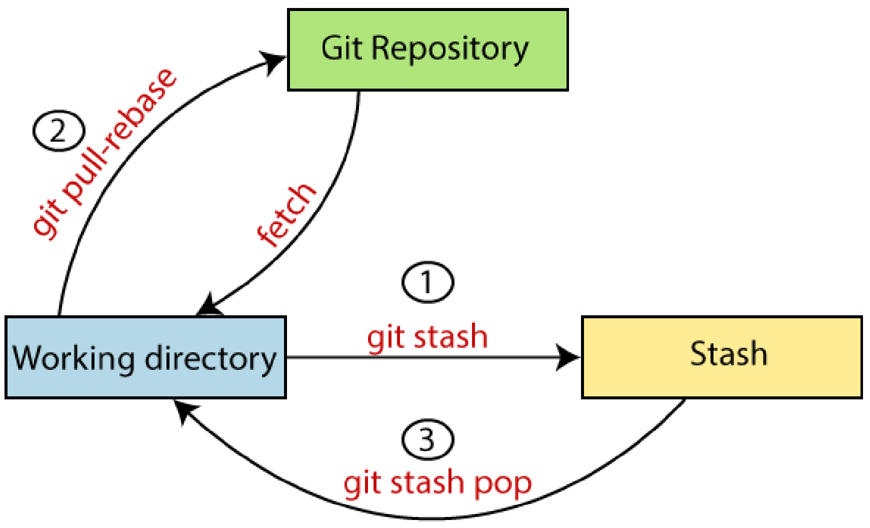
- The pointer (head) will be moved, the index will be adjusted, and the previous changes made will be removed from the working directory, meaning they have been removed altogether
    - Be careful! You will lose any uncommitted changes!



- https://stackoverflow.com/questions/3528245/whats-the-difference-between-git-reset-mixed-soft-and-hard

Git Stash

- Description
    - When working on changes locally, but want to revert to a previous commit without losing your changes, you can use git stash to save your current changes before going back to the previous commit



- Example: When there is a merge conflict upstream, you can stash your changes, pull, and then restore your changes:

```
git pull
```

- Merge conflict!

```
git stash
```

- Save your changes

```
git pull
```

```
git stash pop
```

- Restore changes
- https://git-scm.com/docs/git-stash
- https://www.youtube.com/watch?v=KLEDKgMmbBI&ab_channel=CoreySchafer

Git Submodules

- Description
  - Submodules are a reference (like a pointer) to another repo in your repo. Instead of pulling all the code from the external repo into your own repo, it simply references it.
- Commands
  - Clone a repo

```
git clone --recursive <repo>
```

  - Pulls the submodules in your local repo for the first time

```
git submodule update --init --recursive
```

  - Update submodules to latest versions

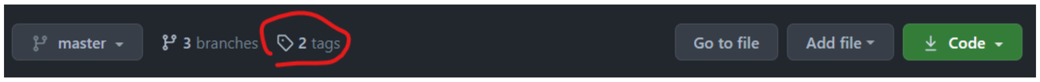
```
git submodule update --recursive --remote
```

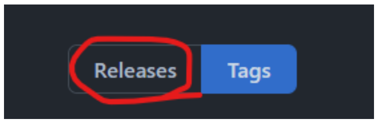
  - Update a specific submodule to a specific commit hash

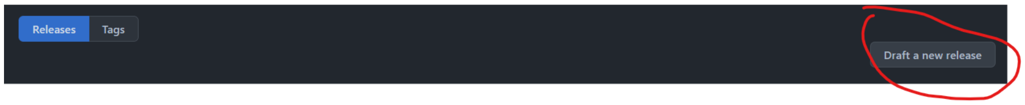
```
cd <submodule directory>
git checkout <commit hash>
```

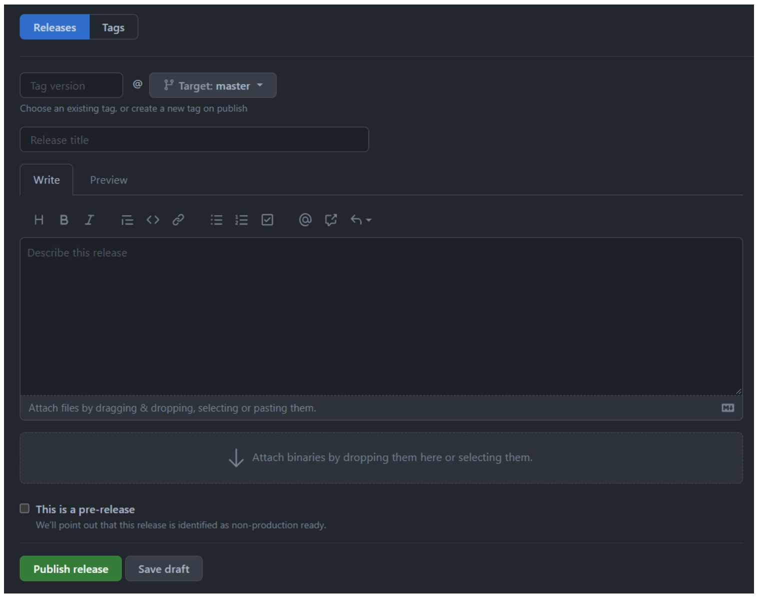
  - After making changes to version of submodule, make sure to commit and push changes
- Resources
  - https://stackoverflow.com/questions/1030169/easy-way-to-pull-latest-of-all-git-submodules
  - https://stackoverflow.com/questions/10914022/how-do-i-check-out-a-specific-version-of-a-submodule-using-git-submodule

Git Tag

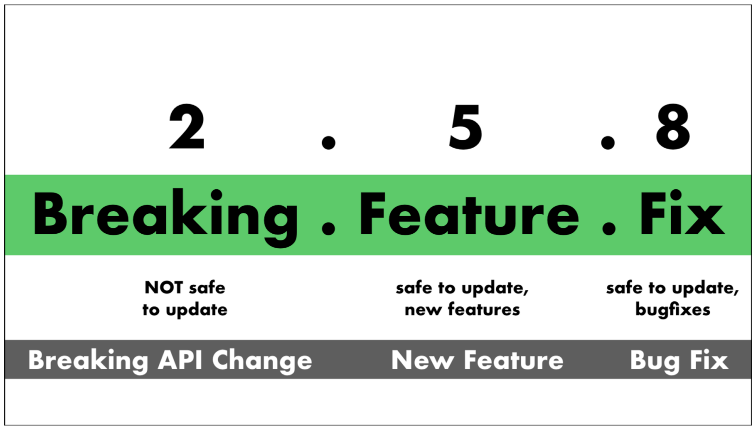
- Description
  - Tagging is a way to tag specific commits. For example, a commit hash like `5d7712f348f6694604288e664e5ed774bb65019b` can be tagged as v1.0. It is common practice to tag specific commits in the master branch to identify it as a specific release version.
- Instructions for creating a new tag/release
  - Go to tags
    - 

    - 

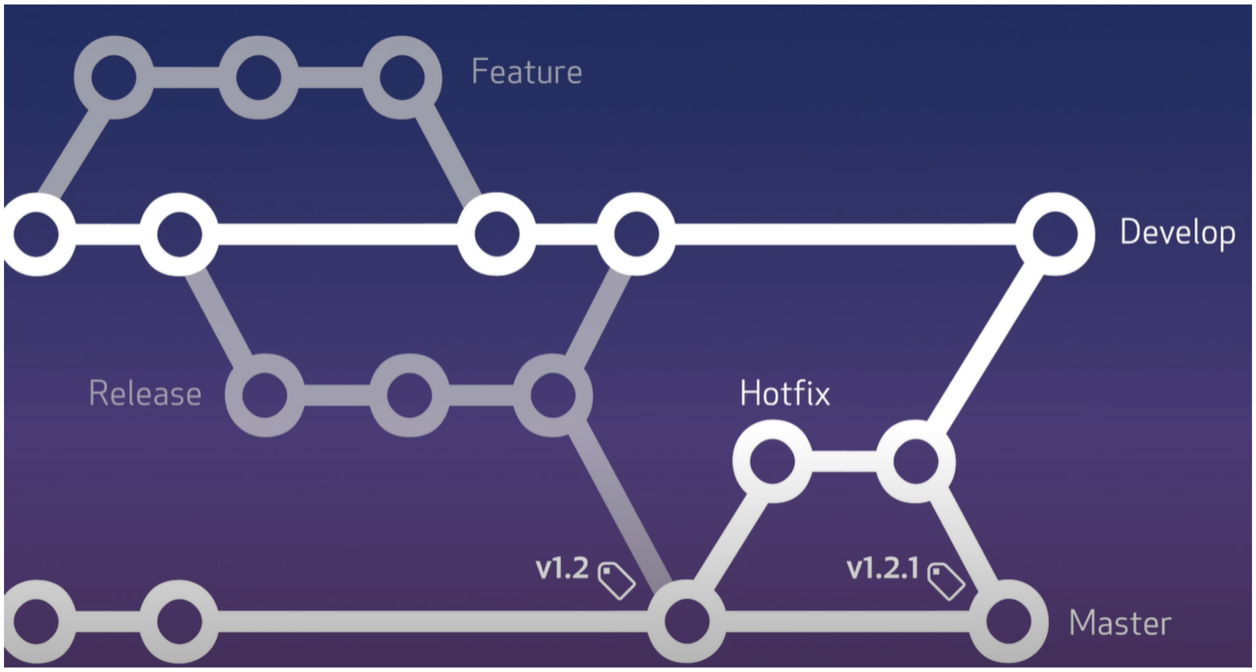
    - 

■ 
Releases
Tags
Tag version
@
Target: master
Choose an existing tag, or create a new tag on publish
Release title
Write
Preview
Describe this release
Attach files by dragging & dropping, selecting or pasting them.
Attach binaries by dropping them here or selecting them.
This is a pre-release
We'll point out that this release is identified as non-production ready.
Publish release
Save draft

Git Workflow Strategy

- Description
    - Develop branch: You work on this branch for work
    - Feature branch: You work on this branch when implementing a new feature and merge it back to develop when done
    - Release branch: You work on this branch when a new release is prepared to be published. Use this branch to make final adjustments and fixes before merging it into master.
    - Master branch: This is the main branch where main changes are made, typically merges from release.
    - Tags: They are tags to a commit made from a release branch merged into the master branch. They help to identify a specific release and are packaged up in the releases section of Github.
        - Notation: Major version . Minor version . Patch version
            - Major version: Breaking changes, not safe to update
            - Minor version: Feature changes, safe to update
            - Patch version: Hotfix / Bug fix, safe to update
            - 

    - Hotfix branch: These are (optionally temporary) branches that fix problems in a release version found in the master branch. They should be merged back into master when finished as a patch fix.

- 

- Resources
    - https://www.youtube.com/watch?v=aJnFGMclhU8&list=PLg7s6cbtAD15mfk2YVHy2U8Ileig_yEZW&ab_channel=GitHubTraining%26Guides

Git Rewriting commit history (not necessary)

- Resources
    - https://www.youtube.com/watch?v=ElRzTuYln0M&ab_channel=TheModernCoder