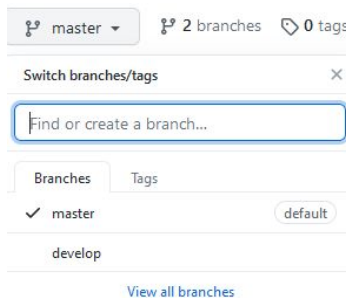


## Exercise 3

Use bash and git commands

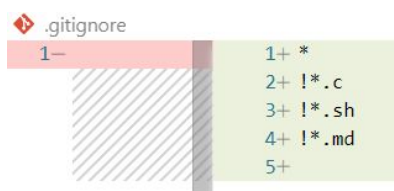
1. On **github.com** create a private repository, **exercise-3**. Initialize the repository with **README.md**.
2. If the default branch name is **main**, change it to **master** on **Github**.
3. **Clone** the repository on the desktop of your computer and open it in Visual Studio code.
4. Edit the text in README.md to **# Exercise 3** and create **.gitignore** in the root of the repo.
5. **Commit** the changes with the message "**First commit**" and then **push** the changes to the remote repo.
6. Run **git log --oneline**, **git branch** and **Create** a new branch, **temp** and run **git branch**.
7. **Push** the new branch to the **remote** and **switch** to **temp**.
8. **Try to delete** branch **temp**. Is it possible?
9. **Switch** to master and then **delete** branch **temp**. Run **git branch**.
10. **Push** the change to the remote and check if **temp** has been **deleted** on the **remote repo**.
11. In the remote repo create a new branch, **develop**.



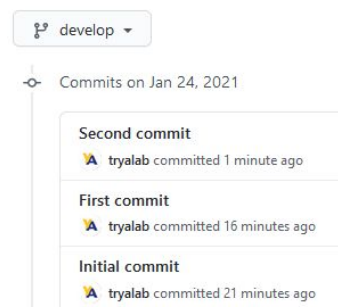
12. In the **local** repo, **try to switch to develop**. Is it possible?
13. Run **git pull** and **switch to develop**. Run **git log --oneline**.

```
$ git log --oneline
556007a (HEAD -> develop, origin/master, origin/develop, origin/HEAD, master) First commit
d981b88 Initial commit
```

14. Ignore all files whose extensions are not **.c**, **.md** and **.sh**.



15. **Commit** the change with the message "**Second commit**" and run **git log --oneline**.
16. **Push** the changes to the remote repository.



a. **git push**

17. **Amend** the message of the last commit to “**Ignored all files except .c, .md and .sh files**”
18. Run **git log --oneline**. Try to push the change to the **remote repo** using **git push**. Is it possible?
19. Use **git push -f** to push the amended commit to the remote repo.
20. Create a new file, **run.sh**, in the repository and write **clear; gcc main.c -o main; ./main** to the file.
21. Create a new file, **main.c**, in the repository and make an **empty c program**.
22. Run **git status** and **commit** the changes with the message “**Made the base of the program**”.
23. Run **git log --oneline** and **push** the commit to the remote repo.
24. Make a program to **print numbers in the range of 1 to 10** to the terminal.

```
C main.c > ...
1  #include <stdio.h>
2
3  int main(void)
4  {
5      for (int i = 1; i < 11; i++)
6      {
7          printf("%d ", i);
8      }
9      printf("\n");
10
11     return 0;
12 }
```

25. Run **sh run.sh** in the terminal and ensure that the program works.
26. **Try** to switch to master. Is it possible? **Stash** the changes and then switch to master.
27. Switch to **develop** and **restore the stashed changes**.
28. **Commit** the changes with the message “**printed from 1 to 10 to the terminal**”
29. Run **git log --oneline** and then **push** the changes to the remote repo.

```
$ git log --oneline
fa15c6d (HEAD -> develop) printed from 1 to 10 to the terminal
ee875a0 (origin/develop) Made the base of the program
46ef37c Ignored all files except .c, .md and .sh files
556007a (origin/master, origin/HEAD, master) First commit
d981b88 Initial commit
```

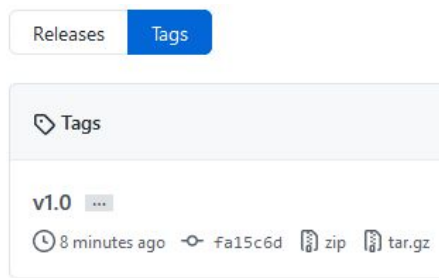
30. **Switch to master** and **merge develop into master**.
31. Run **git log --oneline** and then **push** the changes to the remote repo.

```
$ git log --oneline
fa15c6d (HEAD -> master, origin/develop, develop) printed from 1 to 10 to the terminal
ee875a0 Made the base of the program
46ef37c Ignored all files except .c, .md and .sh files
556007a (origin/master, origin/HEAD) First commit
d981b88 Initial commit
```

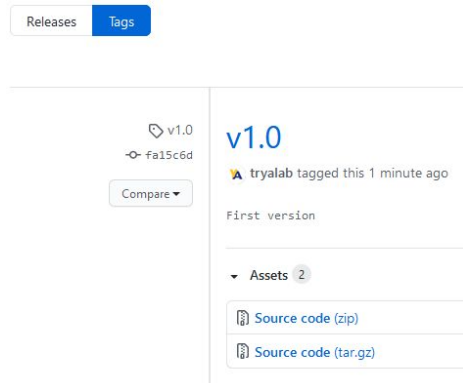
32. Run **git log --oneline** and **make a tag on the last commit**. the tag name shall be **v1.0**

```
$ git log --oneline
fa15c6d (HEAD -> master, origin/master, origin/develop, origin/HEAD, develop) printed from 1 to 10 to the terminal
ee875a0 Made the base of the program
46ef37c Ignored all files except .c, .md and .sh files
556007a First commit
d981b88 Initial commit
```

33. Run **git log --oneline** and **push the tag** to the remote repo. Ensure that the tag has been pushed.



34. **Annotate** the tag with the tag message “**First version**” and **push** the change to the remote repo.



35. **Switch** to **develop** and change the program to print from 1 to 15 to the terminal. Ensure that it works.
36. Run **git status** and commit the change with message “**printed from 1 to 15 to the terminal**”
37. Run **git log --oneline** and then **push** the commit to the remote repo.
38. Switch to **master** and then merge **develop** into it with the message “**Merged with develop**”
39. Run **git log --oneline**. Tag the last commit with tag name **v2.0** and the annotation “**Second Version**”.
40. Run **git log --oneline** and then **push** the changes to the remote repository.

```
$ git log --oneline
312279f (HEAD -> master, tag: v2.0, origin/develop, develop) printed from 1 to 15 to the terminal
fa15c6d (tag: v1.0, origin/master, origin/HEAD) printed from 1 to 10 to the terminal
ee875a0 Made the base of the program
46ef37c Ignored all files except .c, .md and .sh files
556007a First commit
d981b88 Initial commit
```

