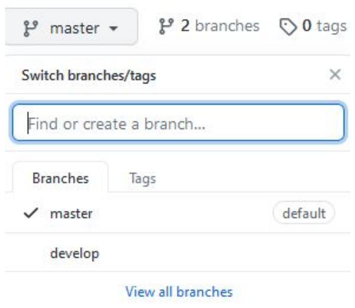
Exercise 3

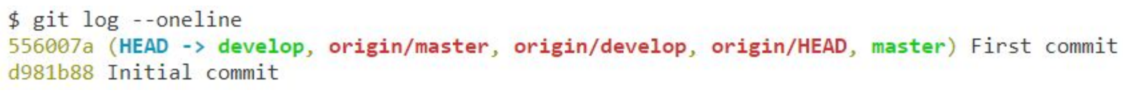
Use bash and git commands

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.
    a. git clone <repo-url>
    b. cd exercise-3
    c. code .; exit
4. Edit the text in README.md to **# Exercise 3** and create **.gitignore** in the root of the repo.
    a. Open README.md in the editore and edit it
    b. touch .gitignore
5. **Commit** the changes with the message "**First commit**" and then **push** the changes to the remote repo.
    a. git add . && git commit -m "First commit"
    b. git push
6. Run **git log --oneline**, **git branch** and **Create** a new branch, **temp** and run **git branch**.
    a. git log --oneline
    b. git branch
    c. git branch temp
    d. git branch
7. **Push** the new branch to the **remote** and **switch** to **temp**.
    a. git push -u origin temp
    b. git switch temp
8. **Try** to **delete** branch **temp**. Is it possible?
    a. git branch -d temp
    b. No. We can not delete a branch when we are on the branch
9. **Switch** to master and then **delete** branch **temp**. Run **git branch**.
    a. git switch master
    b. git branch -d temp
    c. git branch
10. **Push** the change to the remote and check if **temp** has been **deleted** on the **remote repo**.
    a. git push -d origin temp
11. In the remote repo create a new branch, **develop** .



12. In the **local** repo, **try** to **switch to develop**. Is it possible?
    a. git switch develop
    b. No. Because develop does not exist on the local

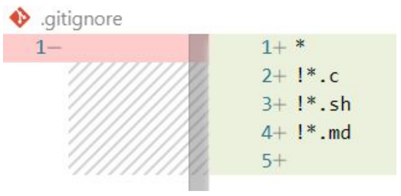
13. Run **git pull** and **switch to develop**. Run **git log --oneline**.



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

   a. git pull
   b. git switch develop
   c. git log --oneline

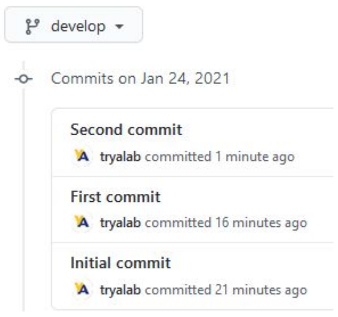
14. Ignore all files whose extensions are not **.c**, **.md** and **.sh**   .



```
.gitignore
1-        1+ *
          2+ !*.c
          3+ !*.sh
          4+ !*.md
          5+
```

15. **Commit** the change with the message “**Second commit**” and run **git log --oneline**.
   a. git add .gitignore && git commit -m “Second commit”
   b. git log --oneline

16. **Push** the changes to the remote repository   .



develop

Commits on Jan 24, 2021

Second commit
tryalab committed 1 minute ago

First commit
tryalab committed 16 minutes ago

Initial commit
tryalab committed 21 minutes ago

   a. git push

17. **Amend** the message of the last commit to “**Ignored all files except .c, .md and .sh files**”
   a. git commit --amend -m "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?
   a. git log --oneline
   b. git push
   c. No. Because when we rewrite the history we need to use -f or --force

19. Use **git push -f** to push the amended commit to the remote repo.
   a. git push -f

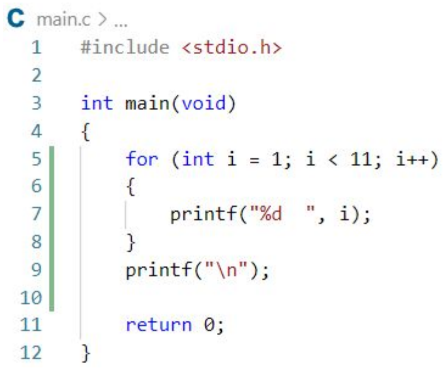
20. Create a new file, **run.sh**, in the repository and write **clear; gcc main.c -o main; ./main** to the file.
   a. echo "clear; gcc main.c -o main; ./main" > run.sh

21. Create a new file, **main.c**, in the repository and make an **empty c program**.
   a. printf "#include <stdio.h>\n\nint main(void)\n{\n\treturn 0;\n}" > main.c

22. Run **git status** and **commit** the changes with the message “**Made the base of the program**”.
   a. git status
   b. git add .
   c. git commit -m "Made the base of the program"

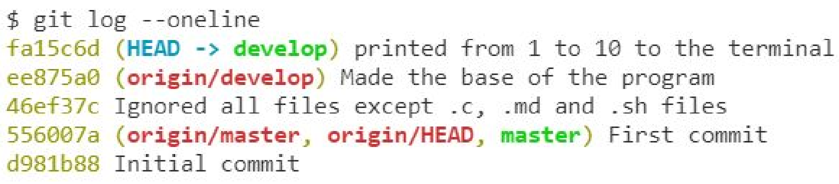
23. Run **git log --oneline** and **push** the commit to the remote repo.
   a. git log --oneline
   b. git push

24. Make a program to **print numbers in the range of 1 to 10** to the terminal.



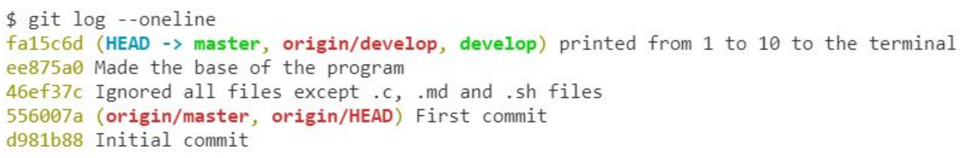
```c
main.c > ...
#include <stdio.h>

int main(void)
{
    for (int i = 1; i < 11; i++)
    {
        printf("%d  ", i);
    }
    printf("\n");

    return 0;
}
```

25. Run **sh run.sh** in the terminal and ensure that the program works.
    a. sh run.sh
26. **Try** to switch to master. Is it possible? **Stash** the changes and then switch to master.
    a. git switch master
    b. No. Because there are some changes in the working directory.
    c. git stash
    d. git switch master
27. Switch to **develop** and **restore the stashed changes**.
    a. git switch develop
    b. git stash pop
28. **Commit** the changes with the message "**printed from 1 to 10 to the terminal**"
    a. git add . && git commit -m "printed from 1 to 10 to the terminal"
29. Run **git log --oneline** and then **push** the changes to the remote repo.



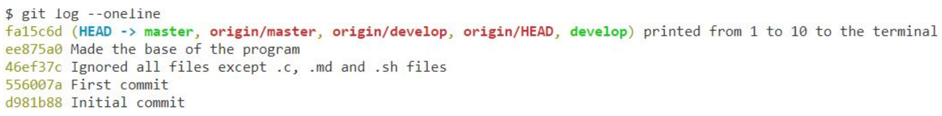
```
$ git log --oneline
fa15c6d (HEAD -> develop) printed from 1 to 10 to the terminal
ee875a0 (origin/develop) Made the base of the program
46ef37c Ignored all files except .c, .md and .sh files
556007a (origin/master, origin/HEAD, master) First commit
d981b88 Initial commit
```

    a. git log --oneline
    b. git push
30. **Switch** to **master** and **merge develop** into **master.**.
    a. git switch master
    b. git merge develop
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
```

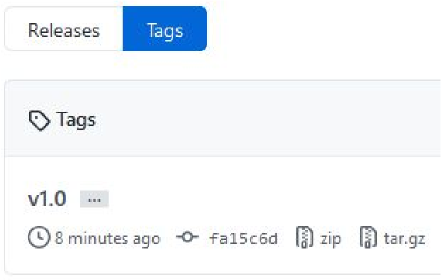
    a. git log --oneline
    b. git push
32. Run **git log --oneline** and **make a tag on the last commit**. the tag name shall be **v1.0**



```
$ 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
```

    a. git tag v1.0

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



Releases Tags

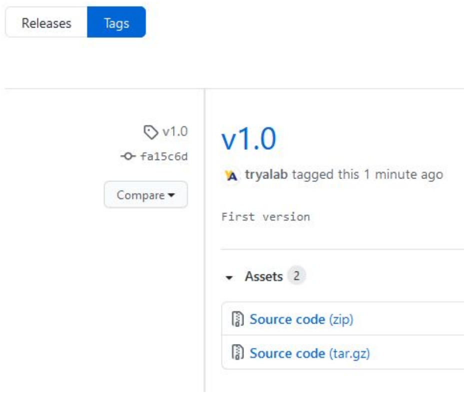
Tags

v1.0 ···

8 minutes ago fa15c6d zip tar.gz

   a. git log --oneline
   b. git push --tags

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



Releases Tags

v1.0
fa15c6d
Compare

v1.0

tryalab tagged this 1 minute ago

First version

Assets 2

Source code (zip)

Source code (tar.gz)

   a. git tag -af v1.0 -m "First version"
   b. git push -f --tags

35. **Switch** to **develop** and change the program to print from 1 to 15 to the terminal. Ensure that it works.
   a. git switch develop
   b. for (int i = 1; i < 16; i++)

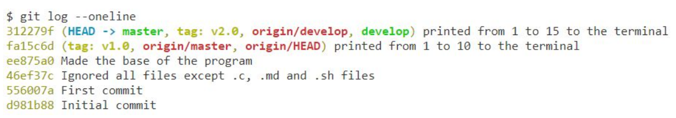
36. Run **git status** and commit the change with message "**printed from 1 to 15 to the terminal**"
   a. git status
   b. git add main.c && git commit -m "printed from 1 to 15 to the terminal"

37. Run **git log --oneline** and then **push** the commit to the remote repo.
   a. git log --oneline && git push

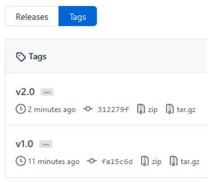
38. Switch to **master** and then merge **develop** into it with the message "**Merged with develop**"
   a. git switch master
   b. git merge develop -m "Merged with develop"

39. Run **git log --oneline**. Tag the last commit with tag name **v2.0** and the annotation "**Second Version**".
   a. git log --oneline
   b. git tag -a v2.0 -m "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
```



Releases Tags

Tags

v2.0 ···

2 minutes ago 312279f zip tar.gz

v1.0 ···

11 minutes ago fa15c6d zip tar.gz

   a. git log --oneline
   b. git push && git push --tags