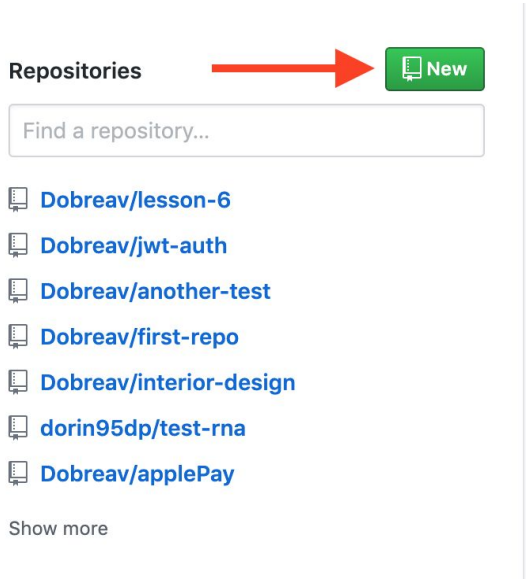
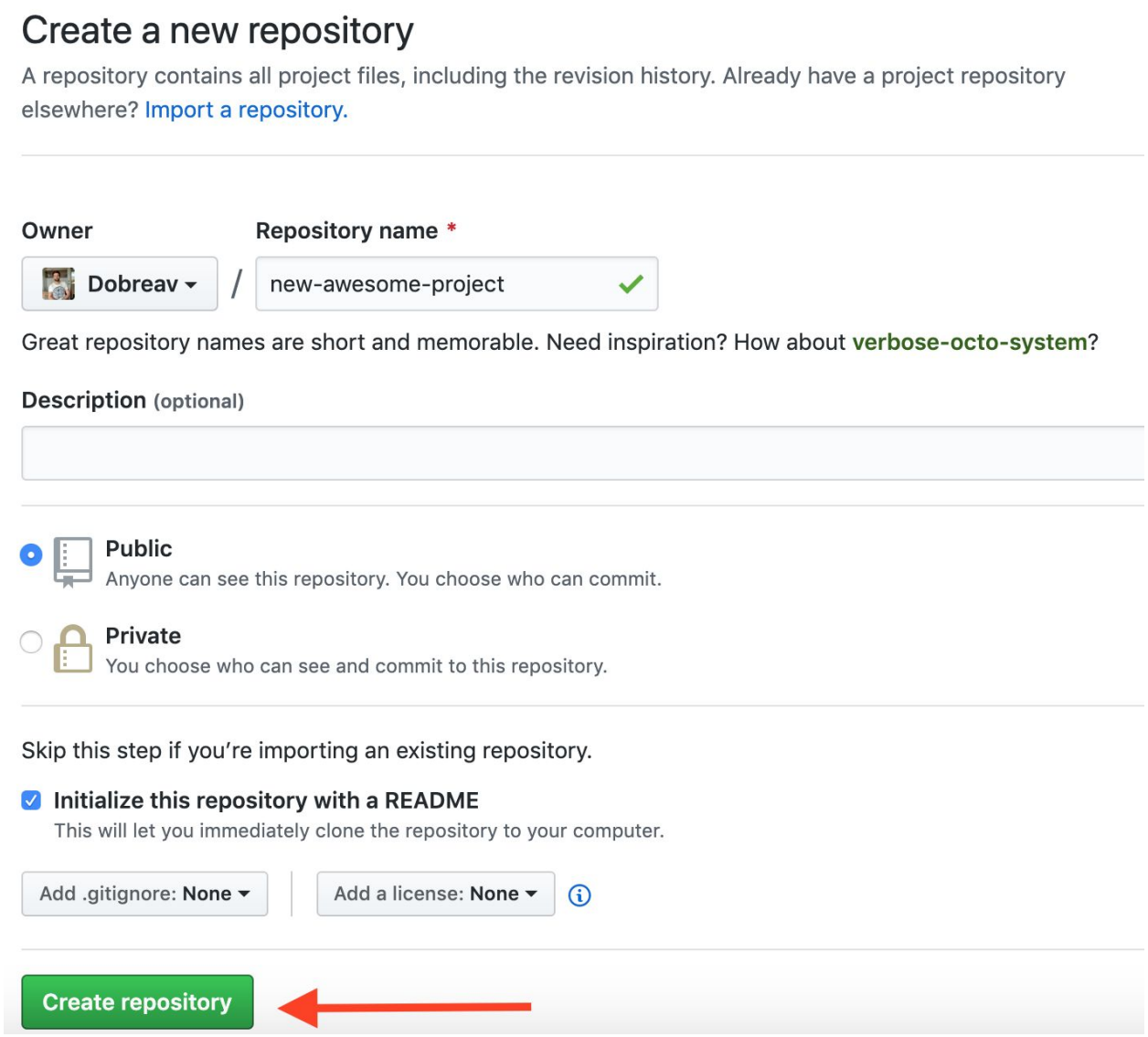
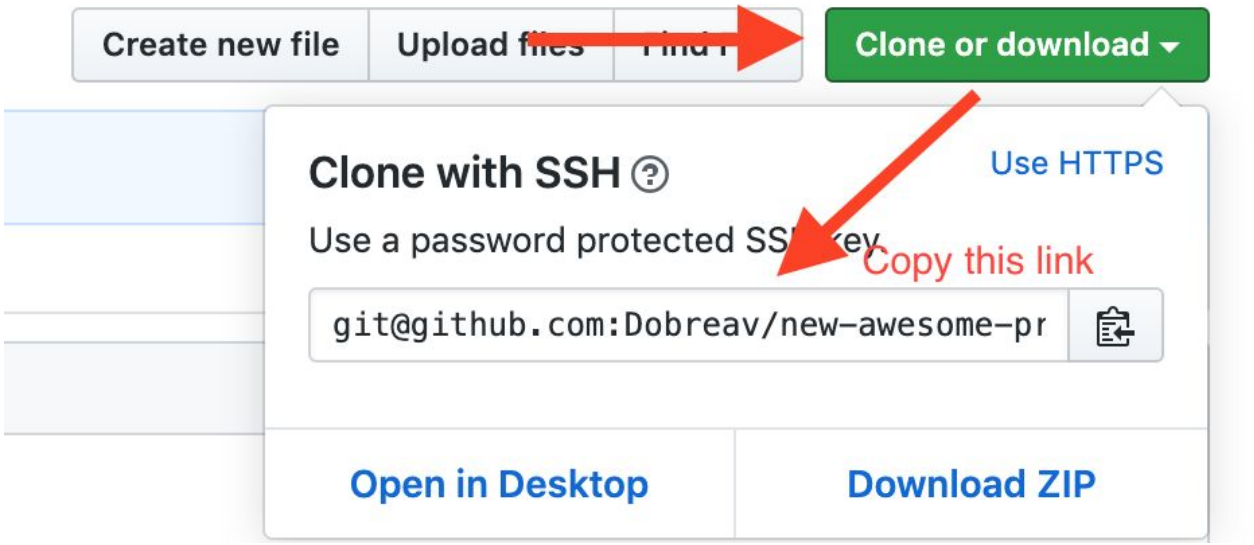
1. Go to github and create new repository



2. In page that opens, fill your project name, and press create repository



3. In newly created repository page, go and **clone and download**, and copy the link to our repo.



4. Now local on your computer open terminal (Linux or Mac's) or CMD on windows. (тут инфа про командные строки: https://tutorial.djangogirls.org/ru/intro_to_command_line/) Choose the directory you want to work in ex: C://users/userName/documents/, and execute this command: **git clone git@github.com:Dobreav/new-awesome-project.git**

```
09/10/19|7:31:36 ➜  lesson-6  $ git clone git@github.com:Dobreav/new-awesome-project.git
```

this will create a new folder with our project from github.

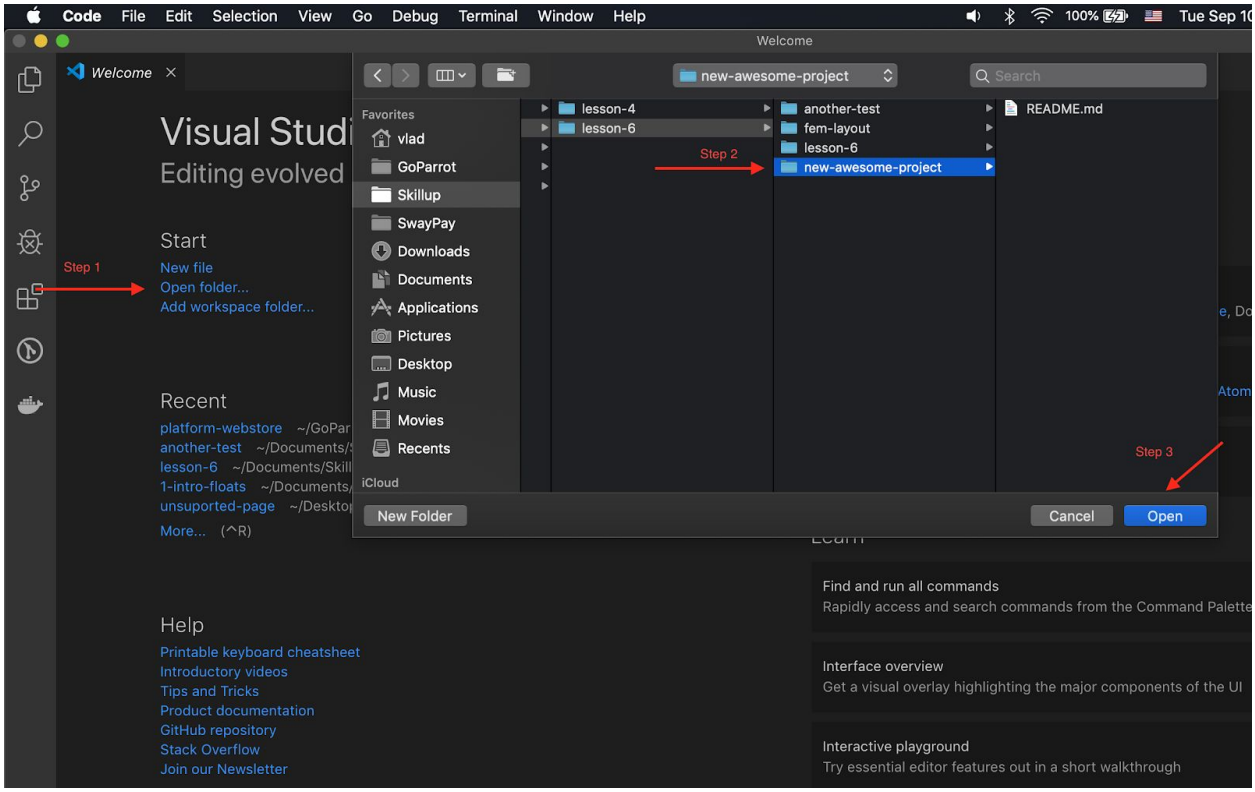
```
Cloning into 'new-awesome-project'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
09/10/19|7:34:58 ➜  lesson-6  $
```

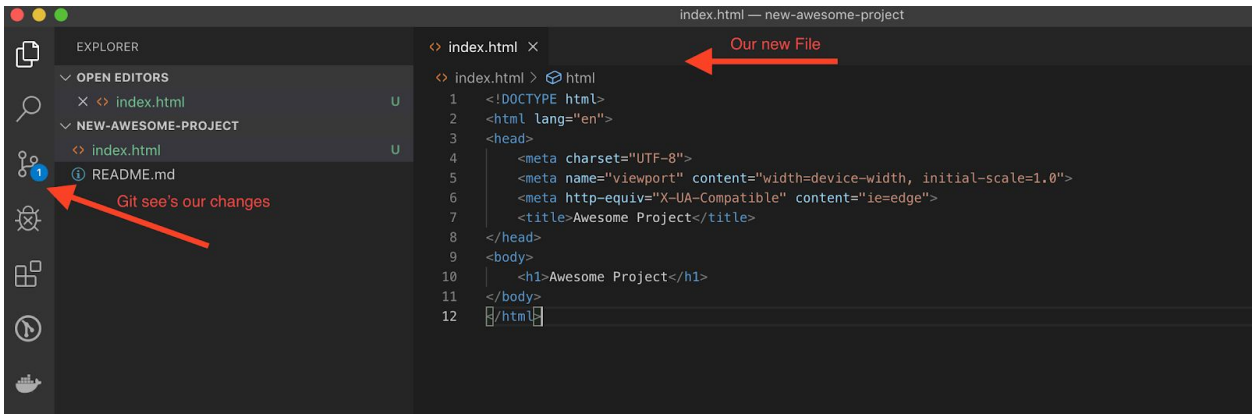
5. Then we need to change our directory to newly clone repo with this command: **cd new-awesome-project**
6. To check that it's a git repository, write this command: **git status**

```
09/10/19|7:36:23 ➜  new-awesome-project git:(master)  $ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
09/10/19|7:36:26 ➜  new-awesome-project git:(master)  $
```

7. You can now create new files and directories in this folder, and it will be watched by Git. For example let's create a new file, called index.html. From VS Code editor





8. After this we need to remember, we are in **working directory**, we have **untracked files** in it (our index.html). Go to terminal and execute **git status**. It should prin this:

```
09/10/19|7:36:26 ➜  new-awesome-project git:(master)  $ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        index.html

nothing added to commit but untracked files present (use "git add" to track)
09/10/19|7:46:55 ➜  new-awesome-project git:(master) ✗  $
```

9. Then we need to add files that we want to commit. This is called staged, it's a step before commit itself. Execute this command: **git add .** and after check with this: **git status**

```
09/10/19|7:46:55 ➜  new-awesome-project git:(master) ✗  $ git add .
09/10/19|7:57:53 ➜  new-awesome-project git:(master) ✗  $ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   index.html

09/10/19|7:57:57 ➜  new-awesome-project git:(master) ✗  $
```

10. Now finaly we can commit our changes with this commad: **git commit -m"Here you write your message".** The result should be

```
09/10/19|7:57:57 ➜  new-awesome-project git:(master) ✗  $ git commit -m"First commit"
[master beb8887] First commit
 1 file changed, 12 insertions(+)
 create mode 100644 index.html
09/10/19|8:01:06 ➜  new-awesome-project git:(master)  $
```

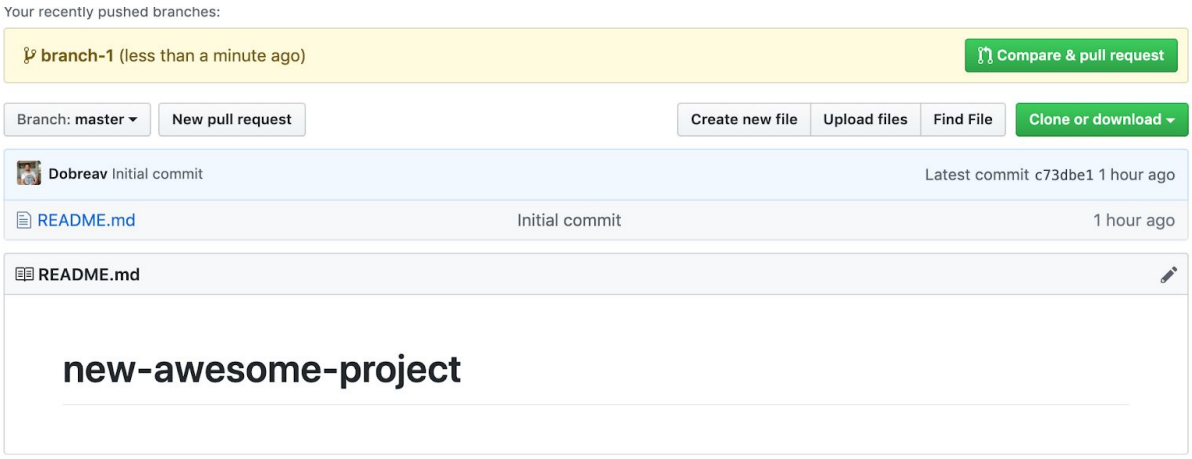
11. Then last step is to push our changes to github (remote origin) with this command: **git push origin master.** Meaning of this command
    a. Push - command that tells git to push all commits somewhere
    b. Origin - is the origin of our repo (in our case it's our github page)
    c. Master - is the name of branch.

```
09/10/19|8:01:06 ➜ new-awesome-project git:(master)  $ git push origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 477 bytes | 477.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:Dobreav/new-awesome-project.git
   c73dbe1..beb8887  master -> master
09/10/19|8:08:37 ➜ new-awesome-project git:(master)  $
```

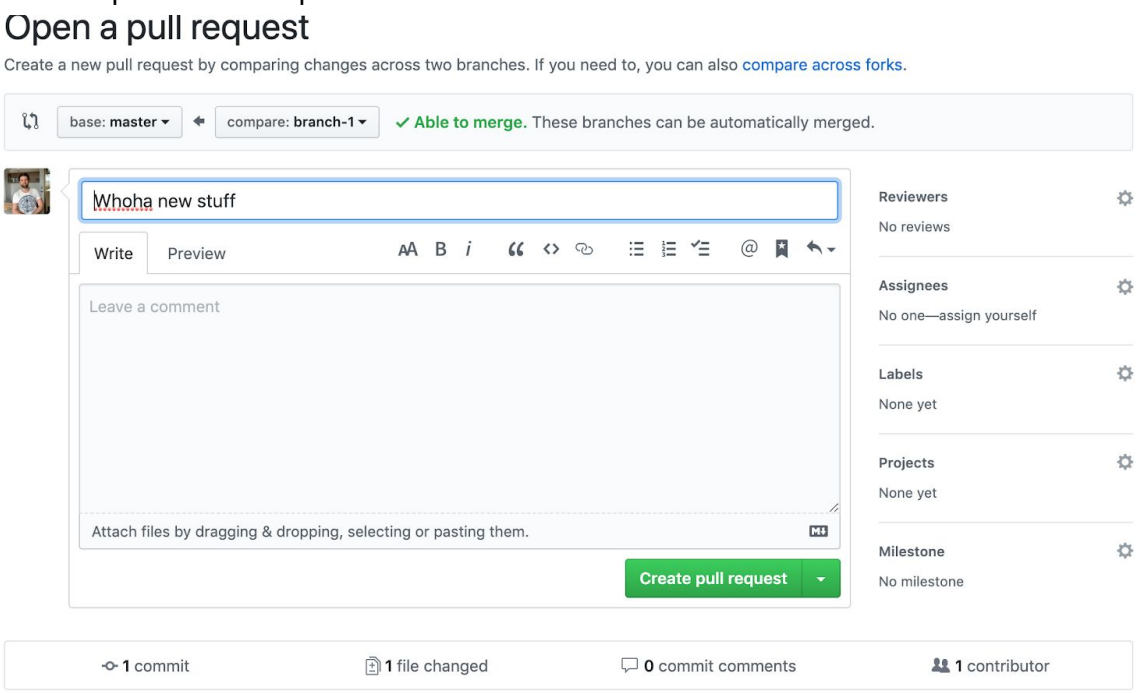
12. Then we can go to github and check our updates.
13. Then we can create another branch locally, for this we can use this command: **git checkout -b branch-1**

```
09/10/19|8:08:37 ➜ new-awesome-project git:(master)  $ git checkout -b branch-1
Switched to a new branch 'branch-1'
09/10/19|8:10:34 ➜ new-awesome-project git:(branch-1)  $
```

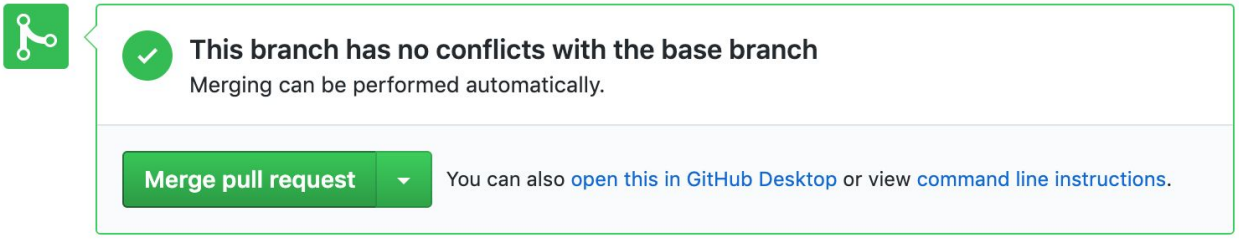
    This means that we created a new branch that is for now, exact copy of master branch.
14. We now can make changes to our index.html file. Save them. And repeat steps 8, 9, 10.
15. Then we need to push our changes to github. And for this use this command: **git push origin branch-1.** You can see that the only change from step 11 is the last word, which is the name of our new branch. If you go to github you should see something like this.



16. Go to Compare & Pull request



17. Create pull request and then merge



18. After this go to terminal and execute
    a. **git fetch** - this will get (update) all changes from remote (github)

    b. Now execute: **git checkout master**. We will switch our branch to master and see that we have our new changes on master!

19. That's the git flow that is used in 90% of cases in real world development so please go through steps and try it and try to understand. Cheers!