Hello, my dear student of Zero To Mastery 2018.
Before you start reading the guide, please remember the following statement:

One GitHub Repository = One cloned repo!

Do not work on the same cloned repo

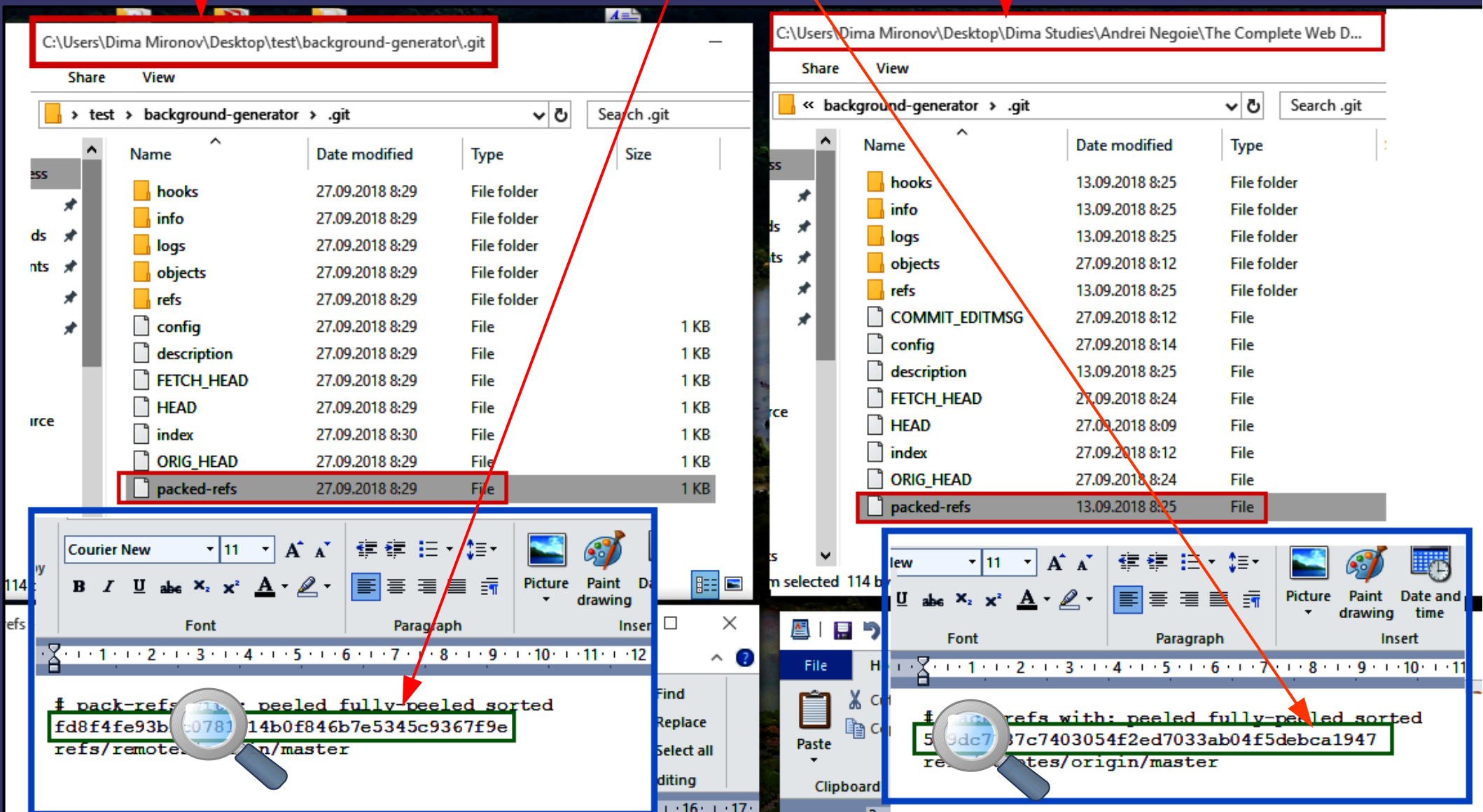
from ***two*** *different paths on your local PC*

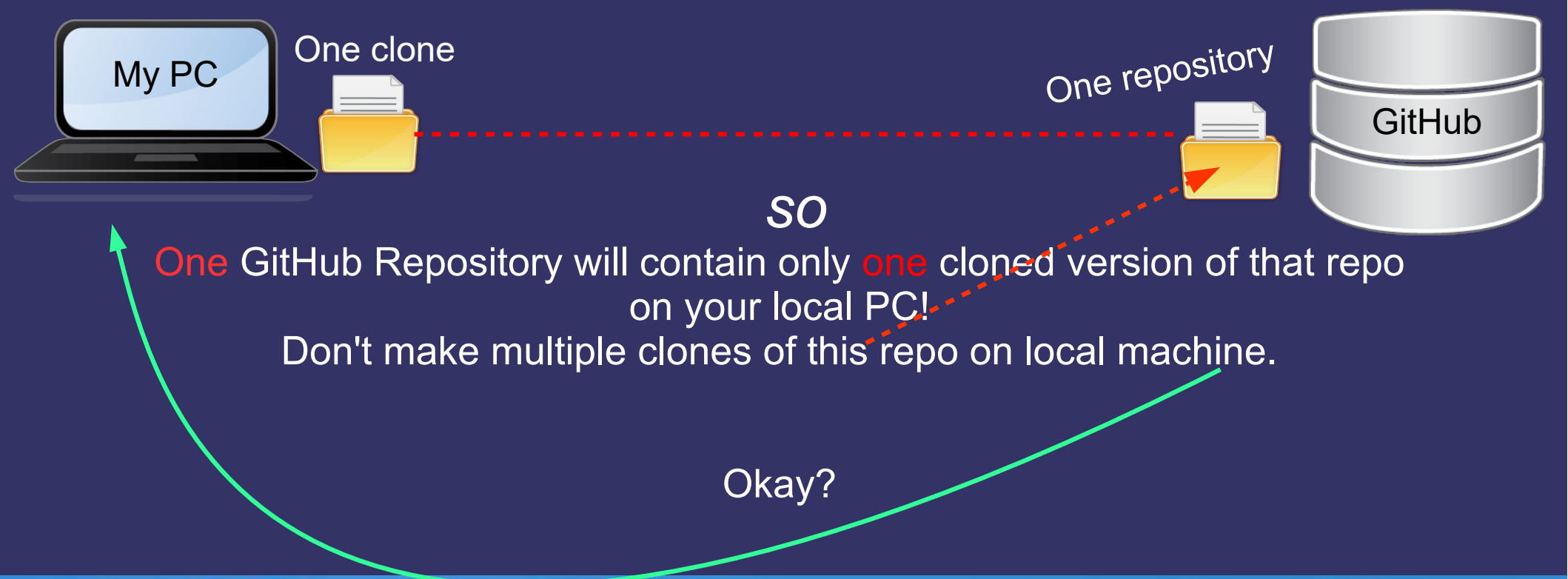
Remember this, or It might cause unexpected results...

The very same cloned repo in different places on your PC will have different package references id's after you **manipulate the original repo!**

Second path =Desktop cloned repo

First path = Original cloned repo





SO

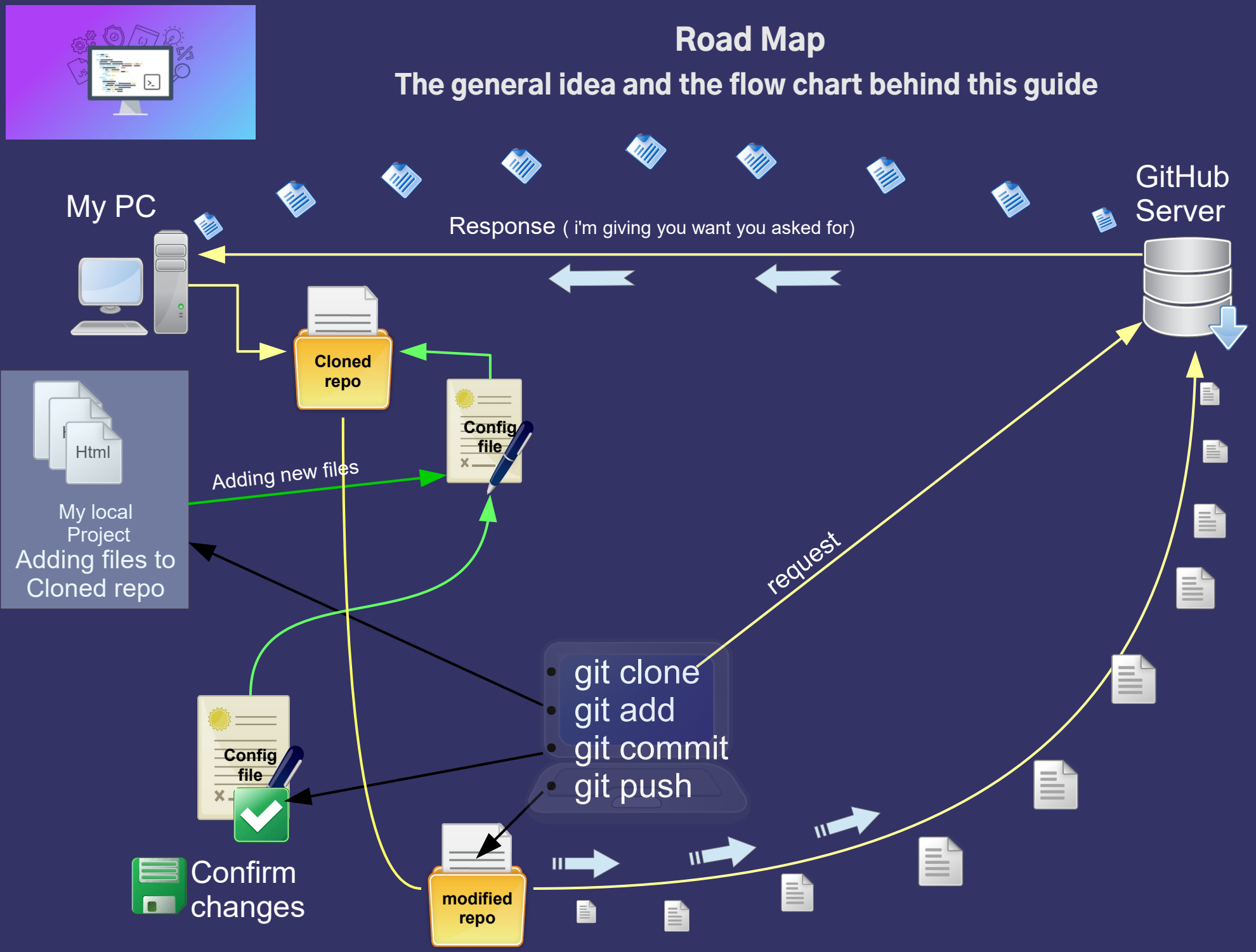
One GitHub Repository will contain only one cloned version of that repo on your local PC!

Don't make multiple clones of this repo on local machine.

Okay?

Road Map

The general idea and the flow chart behind this guide



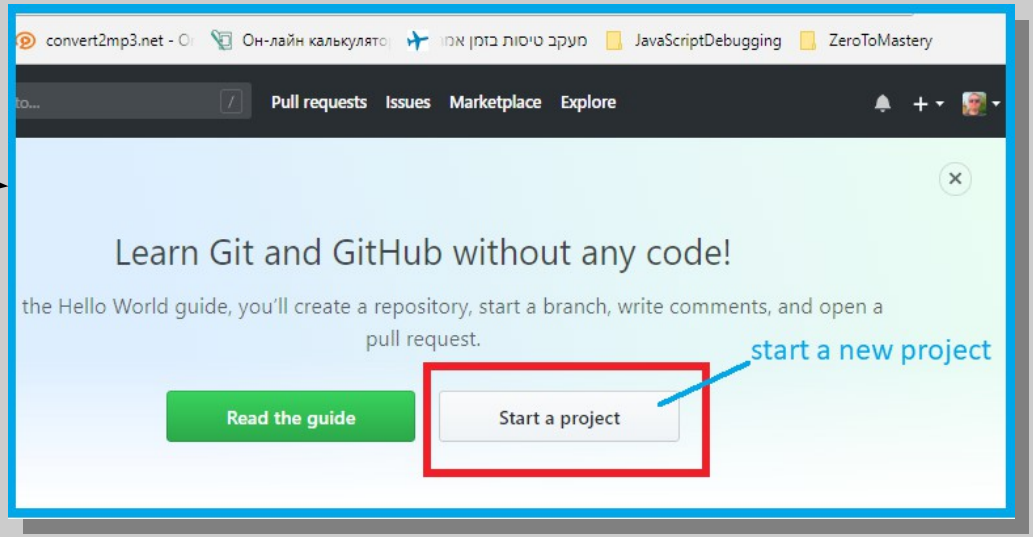
Git and GitHub Source Control

Part1 (Working on a new repository)

The Complete Web Developer in 2018: Zero to Mastery

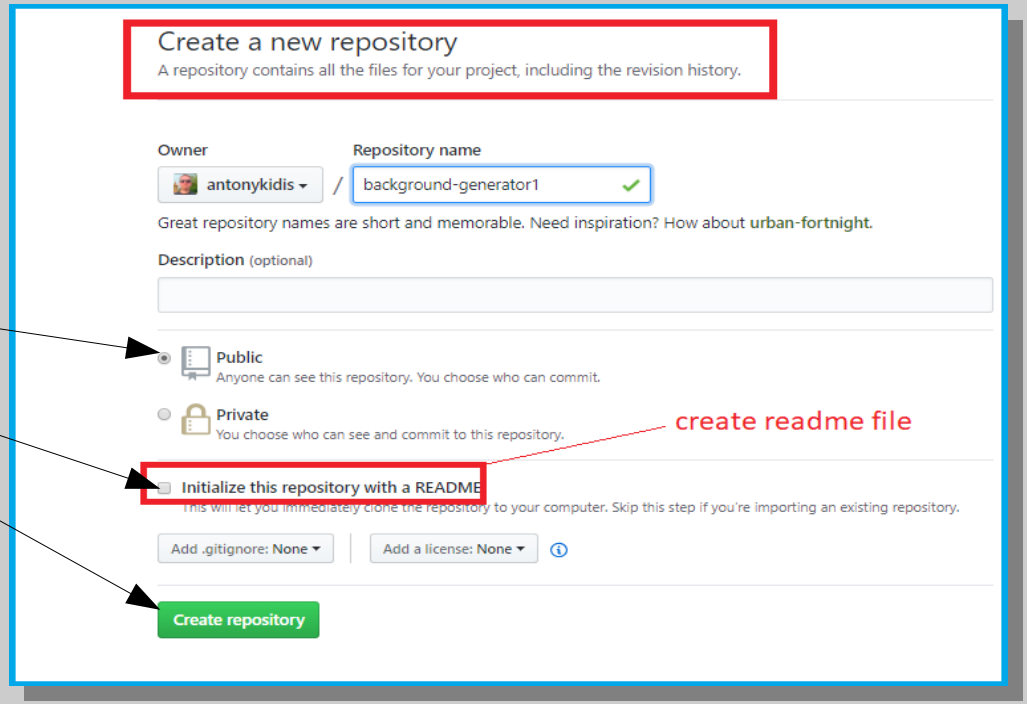
Step 1

- Navigate to https://github.com
- Start a new project



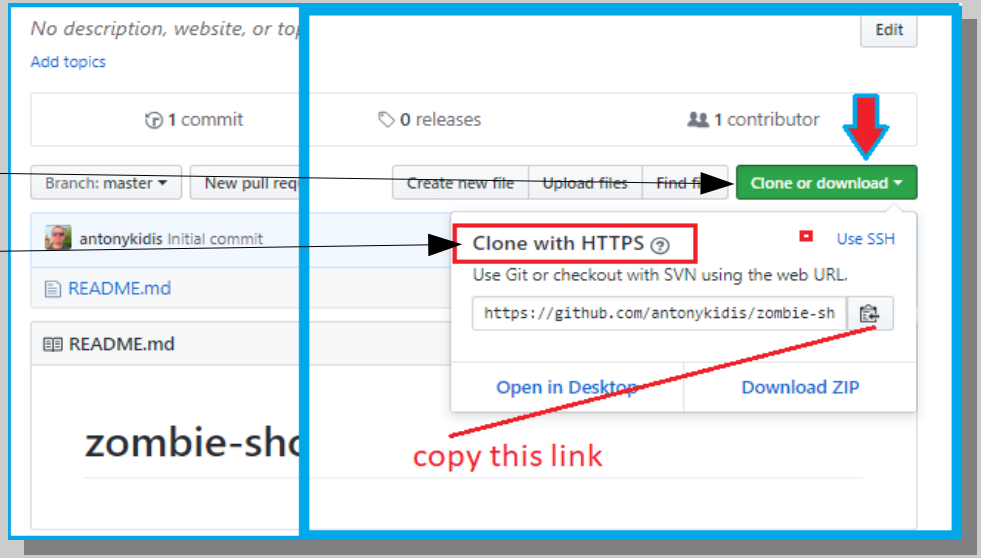
Step 2

- Name your project background-generator
- Select Public
- Check to create a README file
- Finally click create repository button



Step 3

- Click Clone or download button
- Select use HTTPS
- Copy the URL to a clipboard by clicking the Little icon



Step 4

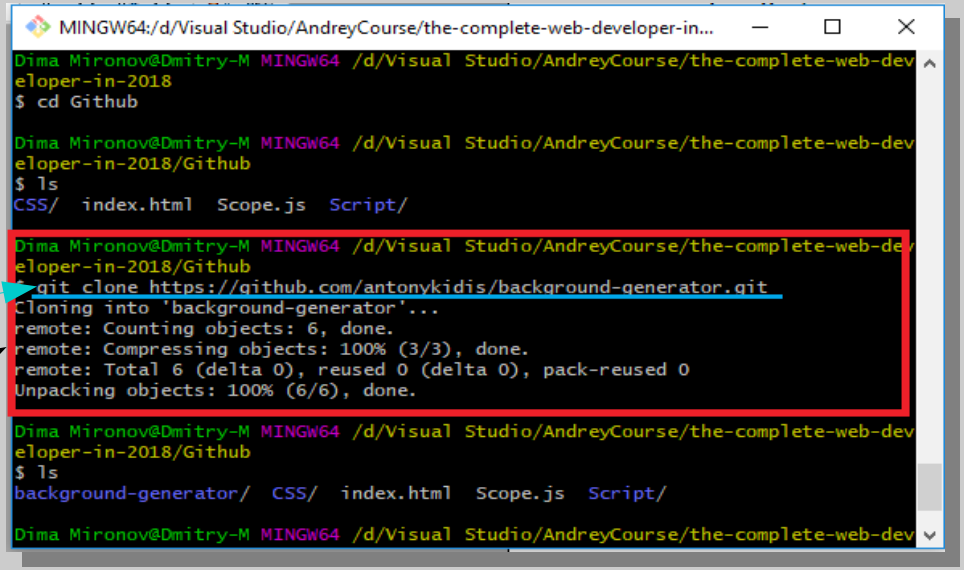
- Open terminal(bash,or powershell) or other prefered terminal.
- Navigate to a project you currently work on.
- Enter the following commad:

  git clone https://github.com/antonykidis/background-generator.git

This will clone GitHub's repository to your computer.

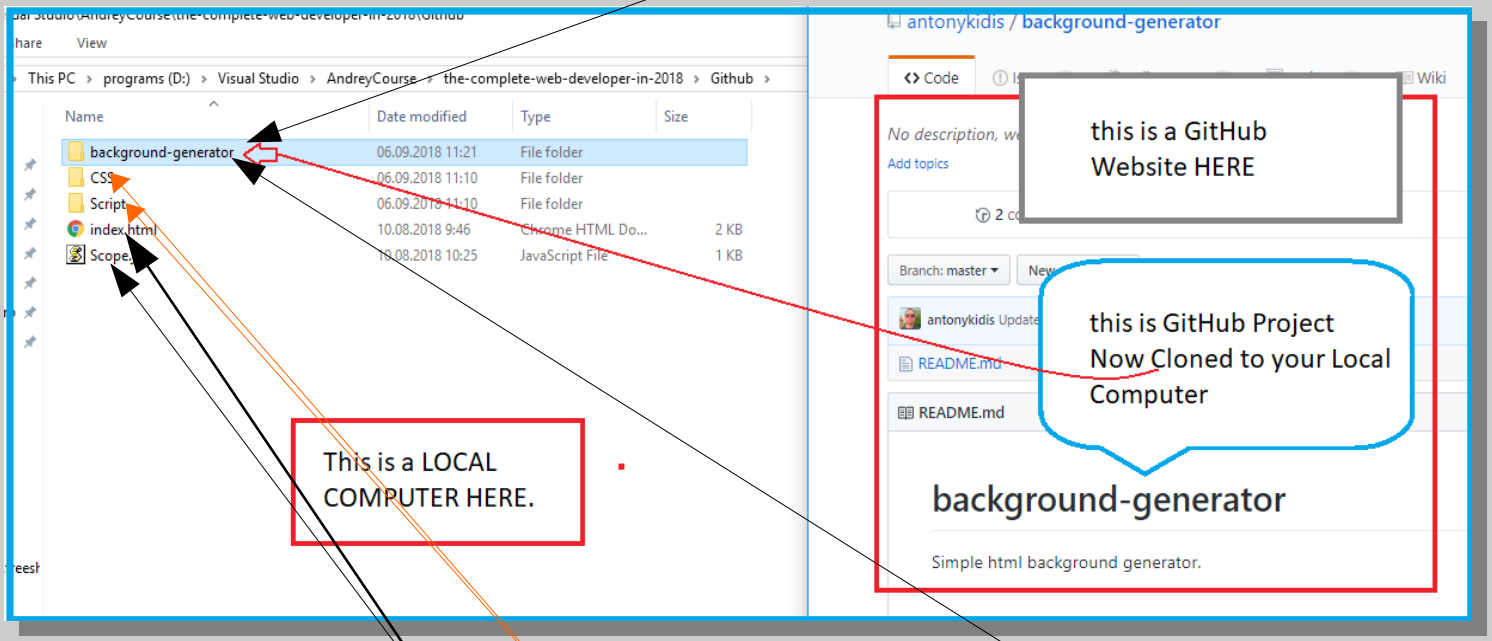
- Here is the following output

  Unpacking objects 100% (6 / 6) done.
  The background-generator cloned now to your computer



We've just copied our repo to a local computer.

You can see the background-generator appeared in the left window



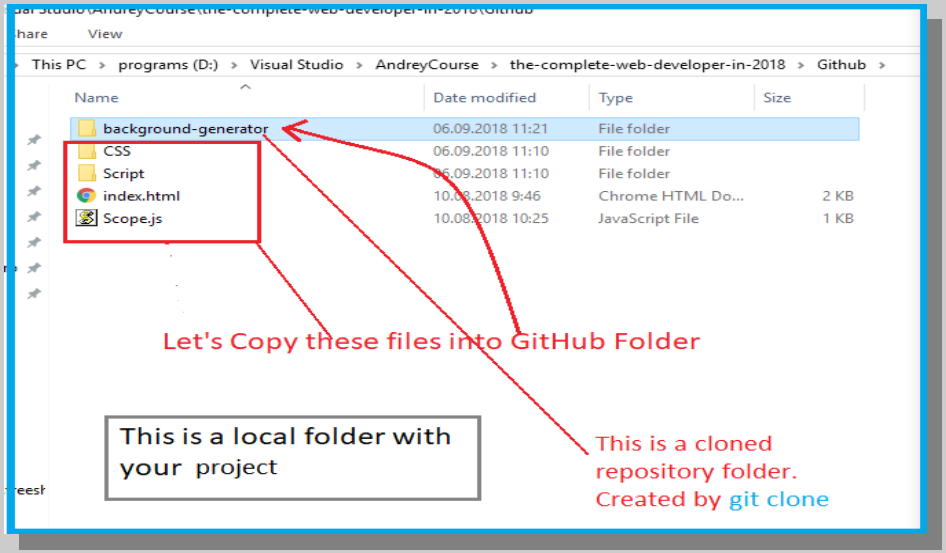
Step 5

Next Step is to take all of our local project's files and folders, and copy them into the background-generator repository folder.

You can copy these files using the IDE, or a command line (terminal)

To copy files and folders via command line please refer to the terminal section of the course

Illustration

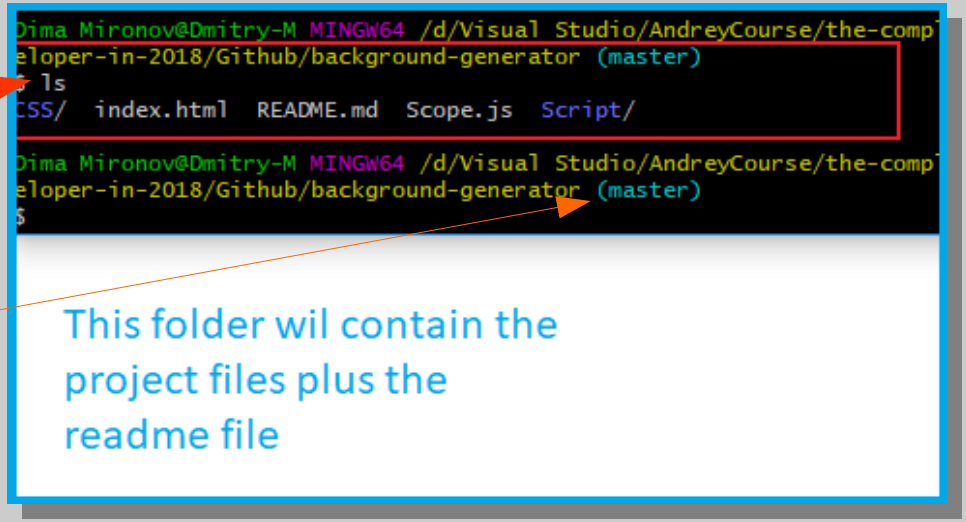


Step 6

1. Back to a terminal window and cd (navigate) to a newly created cloned direcory named background-generator.
   Example:
   cd background-generator
2. Then type ls (dir in windows) to
   list the contents of a background-generator folder.

3. You should see the similar output with all the copied files from the previous step.

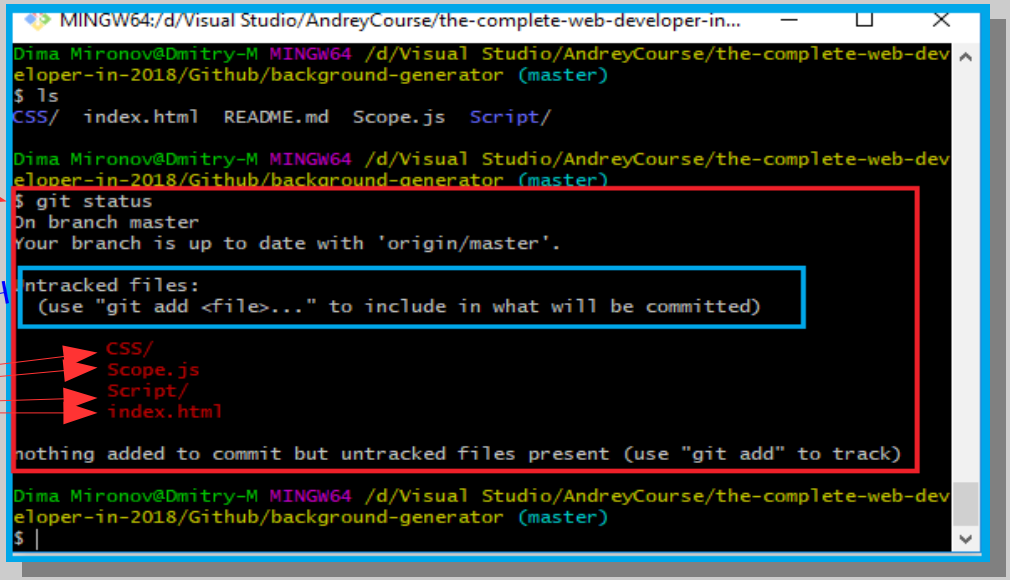
(master) means that we now working in the repository folder



Step 7

- We now can check the status of our files since we've copied our files into the repository folder(comunication folder)
- While in terminal type git status
- This will invoke the following output

You can see that we have untracked files and folders
We can now add these files to a GitHub **Configuration file** before upload it back to GitHub



Configuration file. We will explore it in the following sections.

Adding files and folders to configuration file

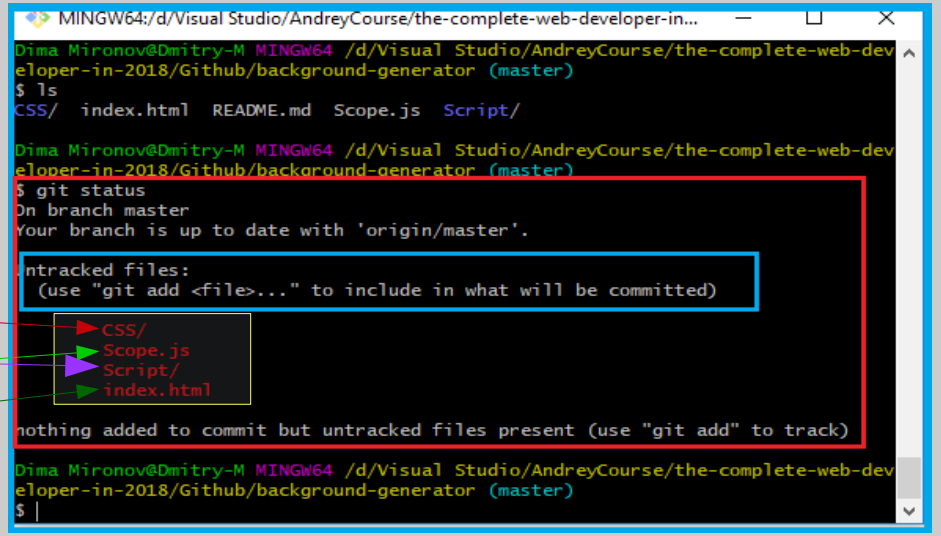
Step 8

- Let's Add utracked files to a special Configuration file

  Type the following commands:

- git add CSS (folder)
- git add Script(folder)
- git add Scope.js(file)
- git add index.html (file)

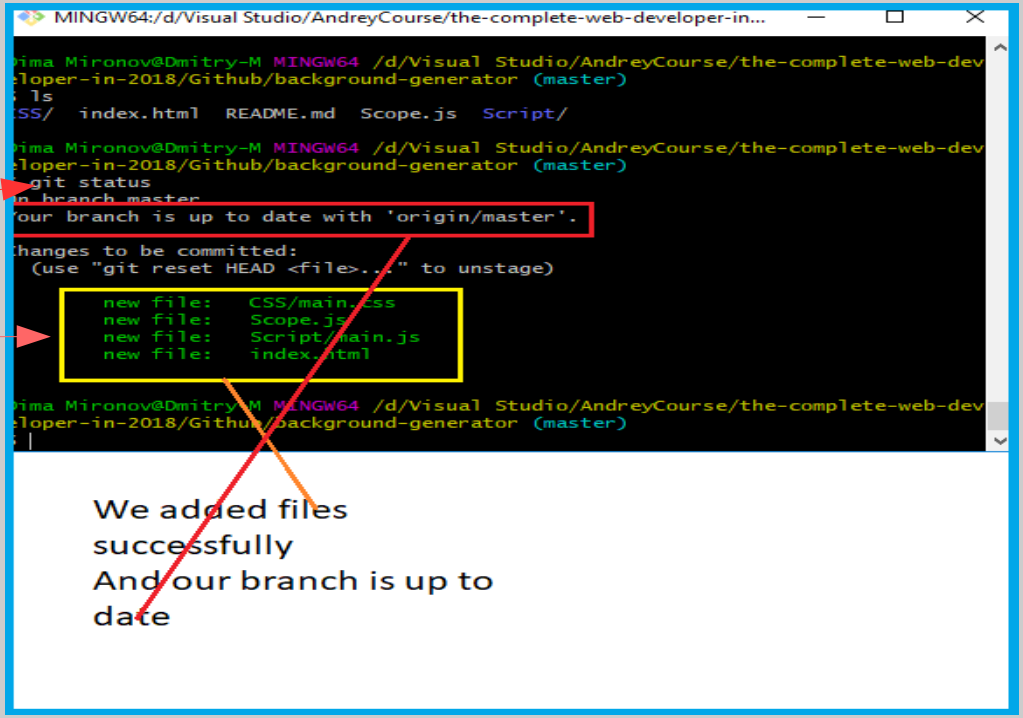
To add all files and folders at once type **git add .**



Step 9

- After adding each file, and folder

1. Type git status
2. You should see the following output
3. Okay we've just set the configuration file. Great!
4. Next step is to commit these files into the configuration file.

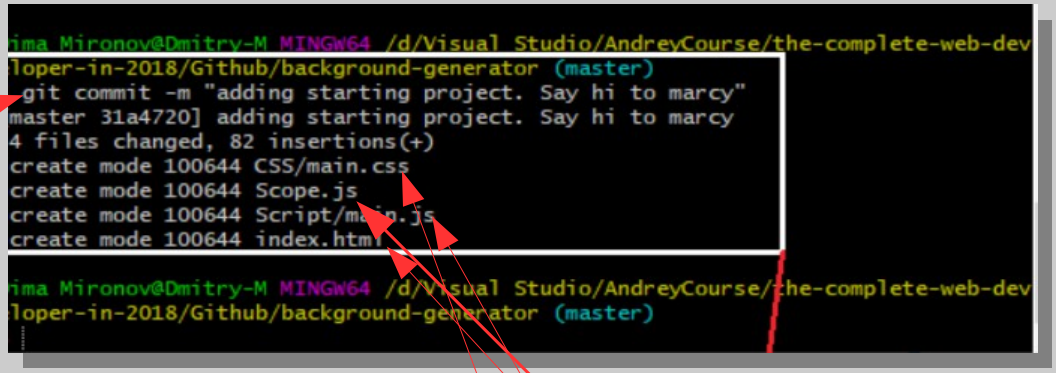


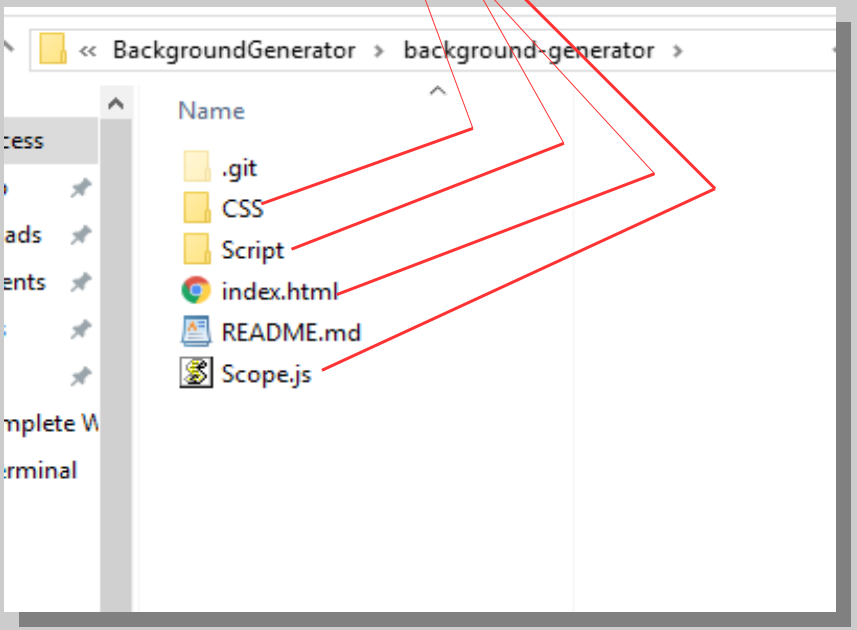
Commit the changes we've made in previous steps

Step 10

- Let's commit previously added files and folders.
- Enter the following command
- git commit -m "adding starting project"

Commit means **Save all** the changes to config.file before pushing it back to the GitHub server

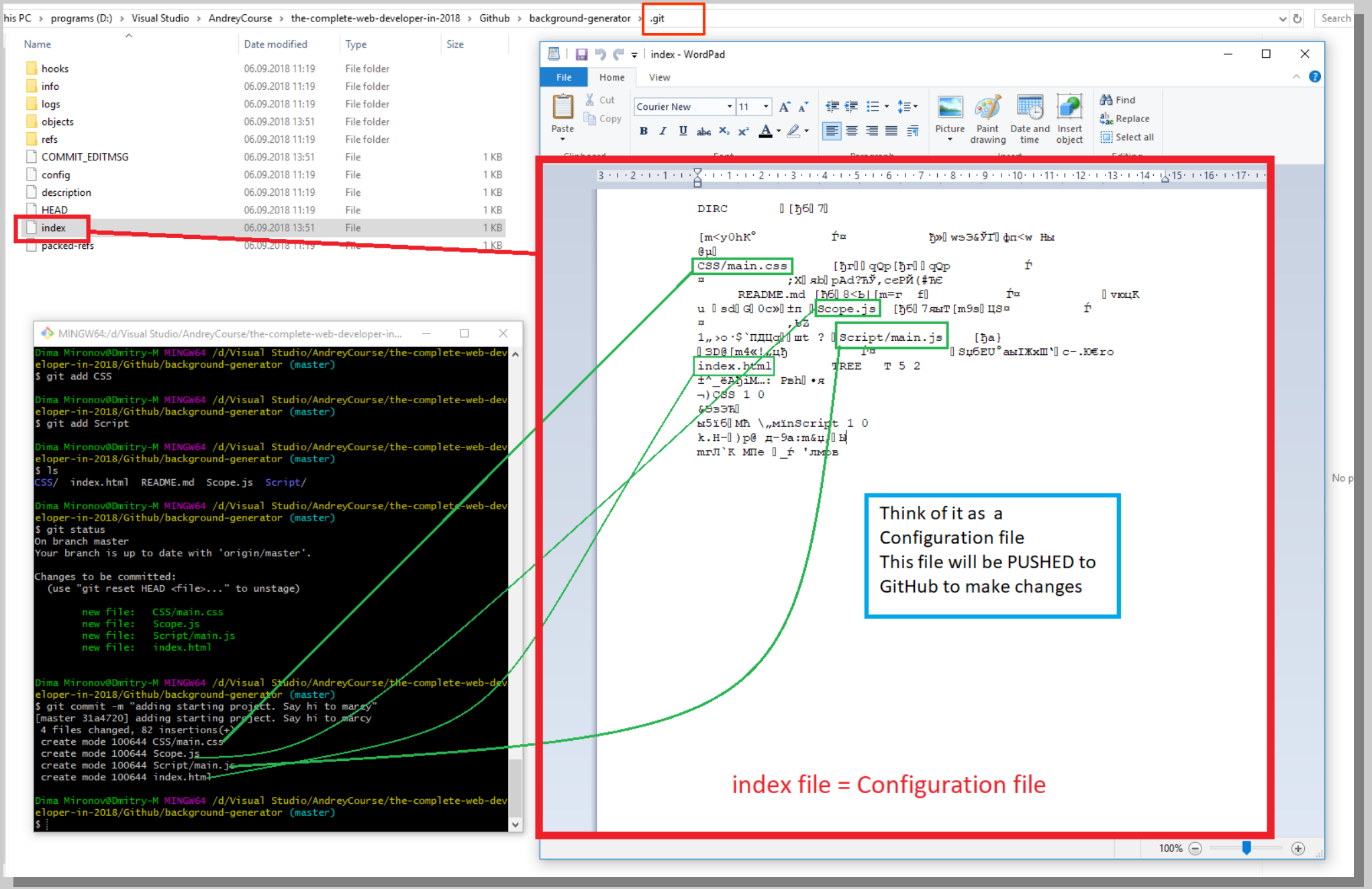




We've just set up the configuration file

- The last step is to push these changes back to the GitHubServer. But let's explore the configuration file first.

Explore the configuration file



Push changes to GitHub

Step 11

- Finally Push the changes to the server
- Use the following command
- git push
- This will get you to the following output
- We've succesfully wrote objects to http://github.com...

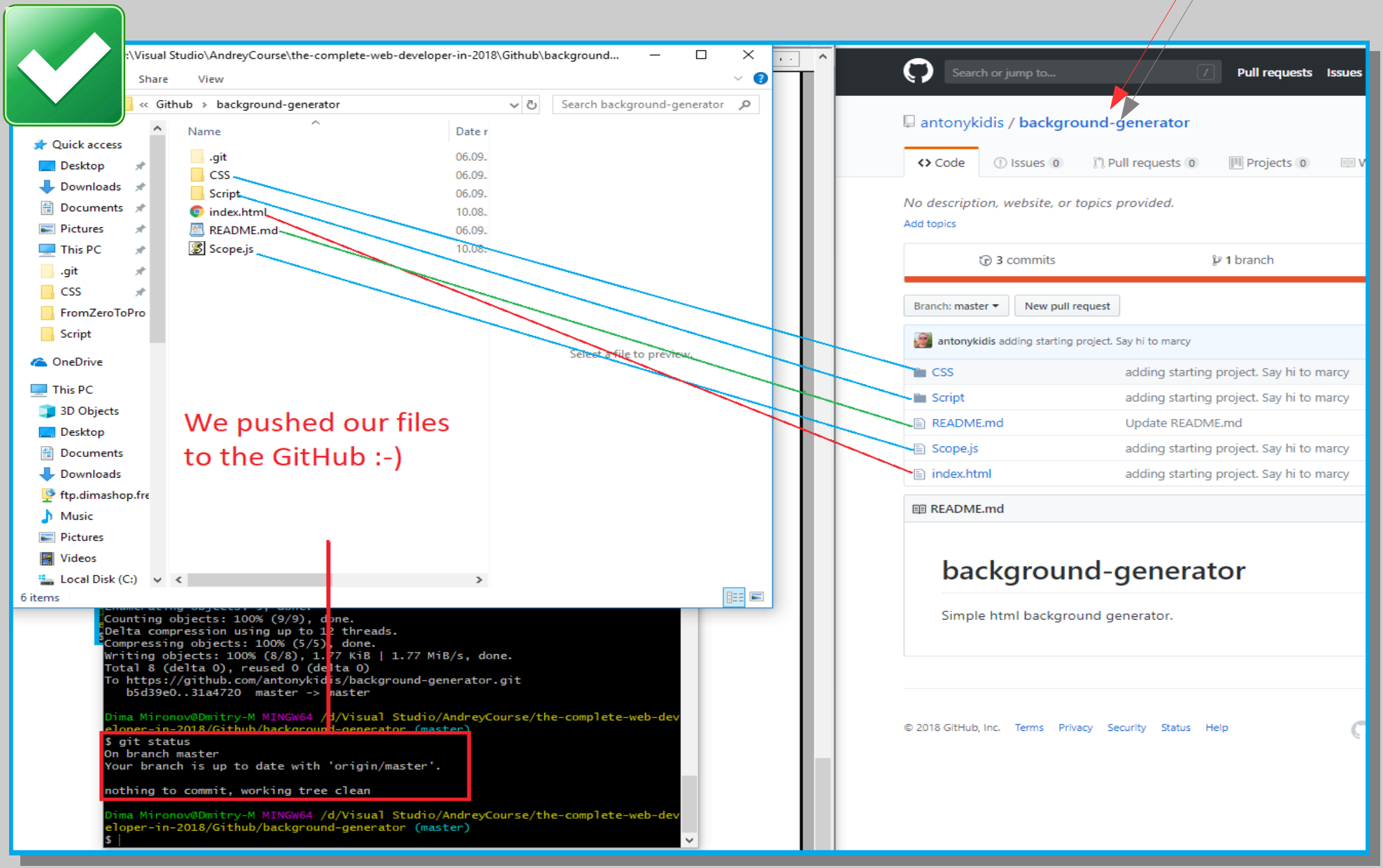
```
Dima Mironov@Dmitry-M MINGW64 /d/Visual Studio/AndreyCourse/the-complete-web-dev
eloper-in-2018/Github/background-generator (master)
$ git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 12 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (8/8), 1.77 KiB | 1.77 MiB/s, done.
Total 8 (delta 0), reused 0 (delta 0)
To https://github.com/antonykidis/background-generator.git
   b5d39e0..31a4720  master -> master
```

Great!
Now a background-generator is online

Step 12

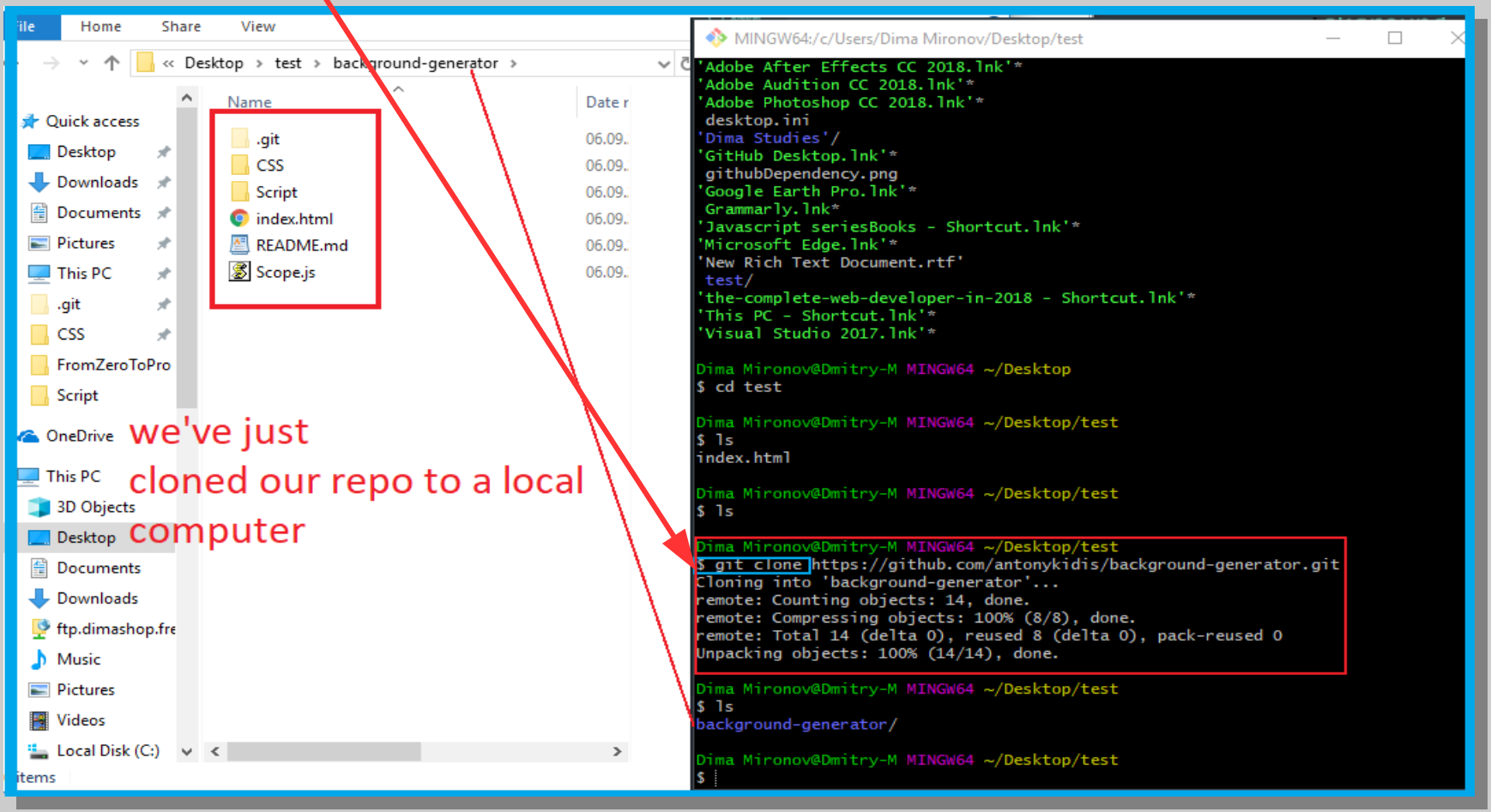
Go back to https://github.com and check
What you've done so far
(Test pushed files on GitHub)

What we've done so far
We succesfully added a new files into GitHub repository



Next Step - Work on Existing Repository

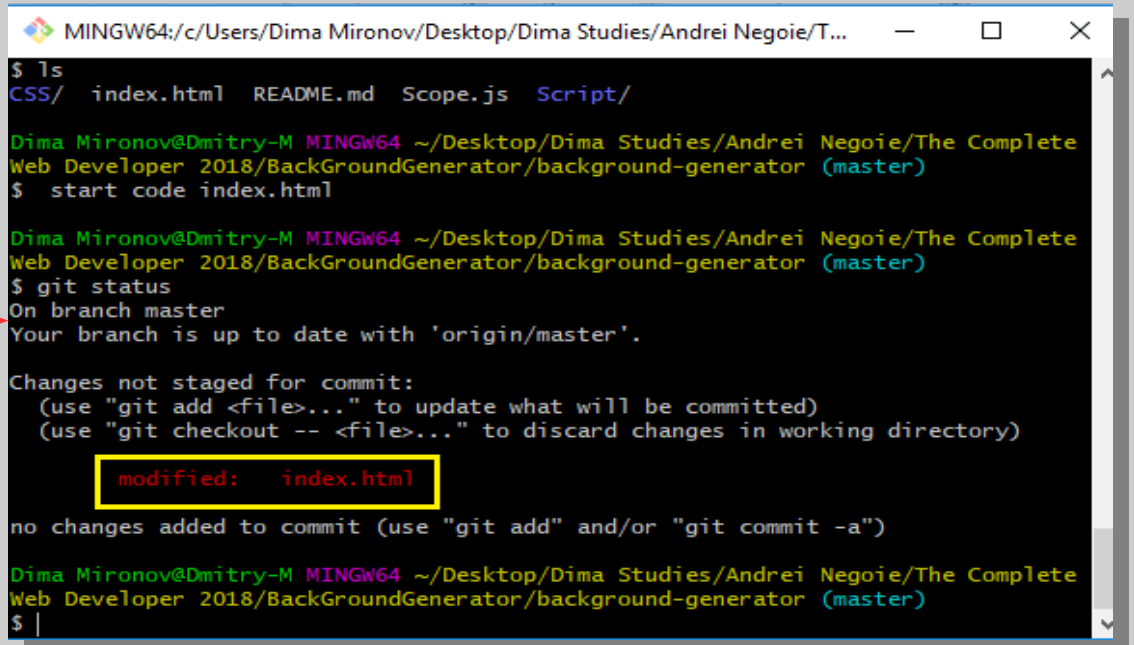
- Open Terminal(or cmd/bash/powrshell in windows)
- Navigate to the folder where you want to download a GitHub existing repository
- Copy repository's url from GitHub
- Type the following command (Example)
  git clone https://github.com/antonykidis/background-generator.git
- Finally check if a cloned repository appears in the desired folder



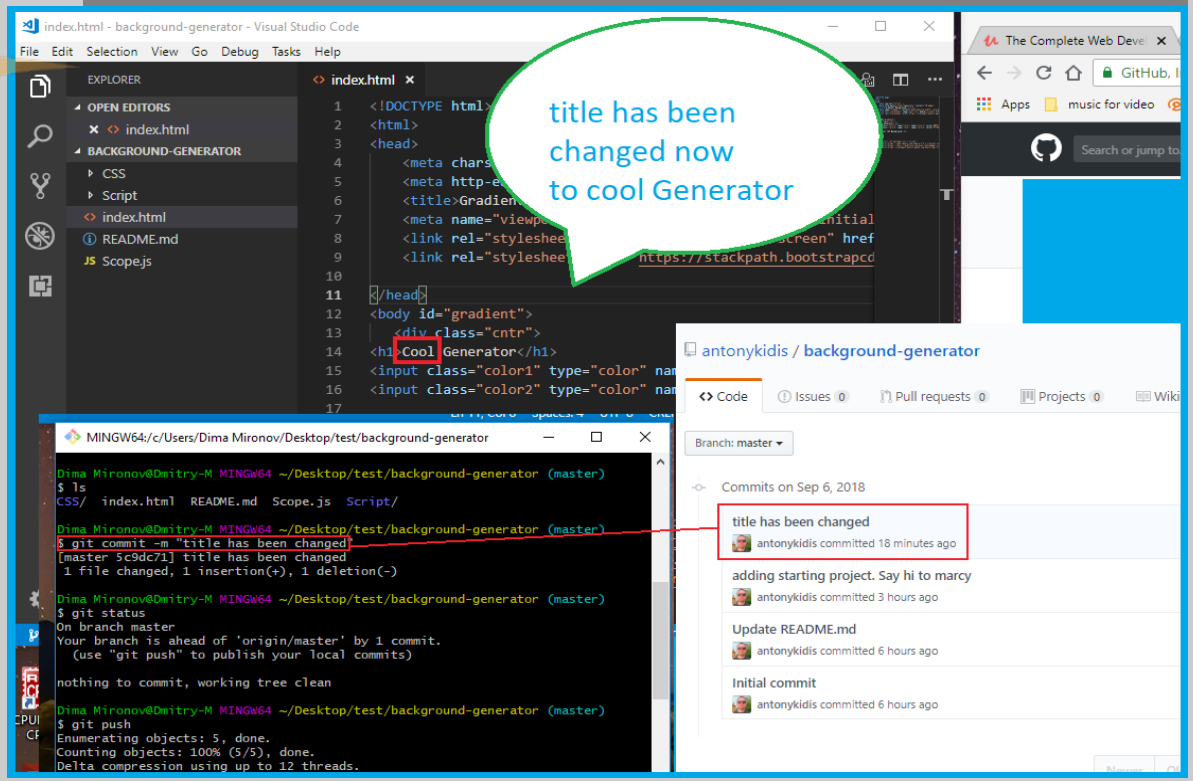
Next Step - Pushing a modified files back to a GitHub

1. For example:
   open index.html and change the title<h1>here</h1>
   to a cool generator
2. Save it
3. Comeback to a terminal window, and type
   git status

   You should see a similar output
4. Type git add (adding the index.html)
5. Type git commit -m "title has been changed"
   (save changes)
6. Type git push (pushing the changes back to
   server)

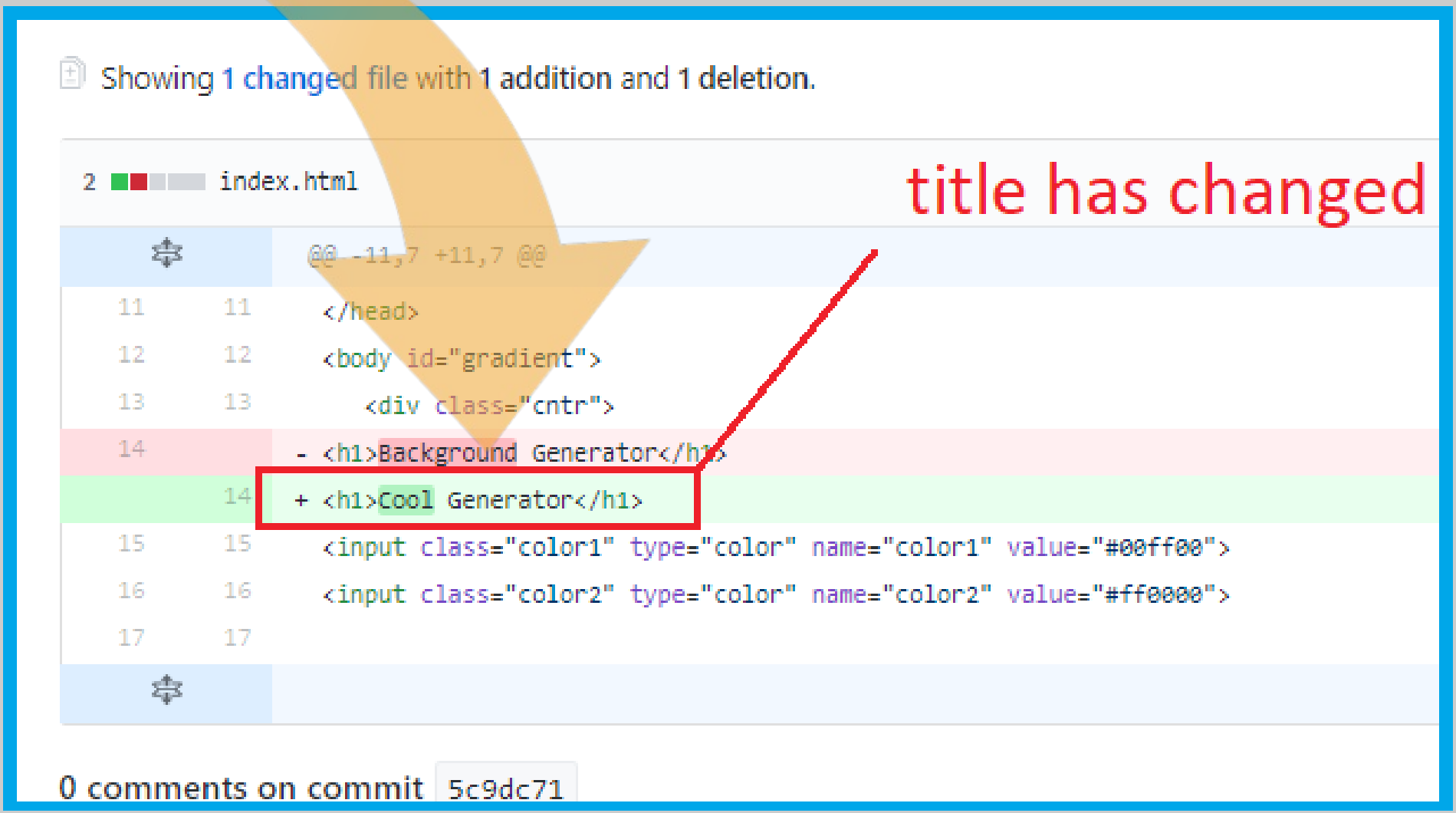


7.Go to GitHub



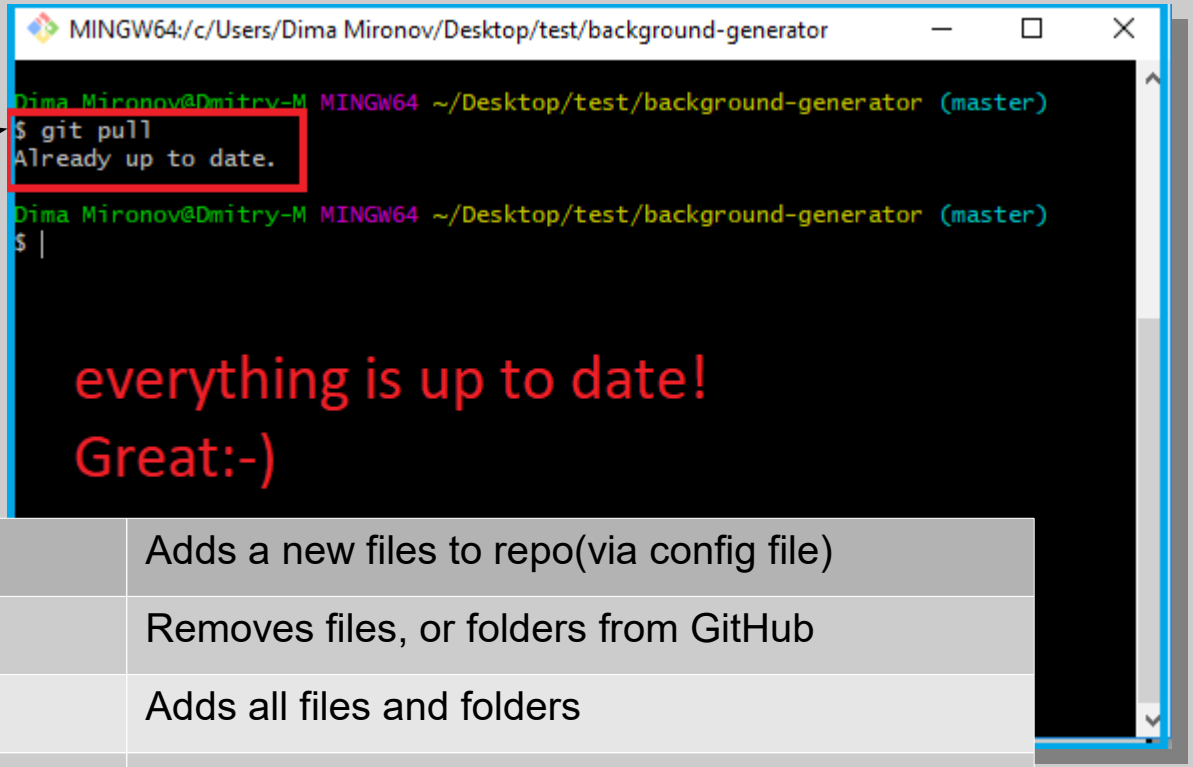
Navigate to commits link

And see the actual changes we've just made
As you can see the title has been changed to a Cool Generator



We can pull the latest version from GitHub

- Type git pull



Useful commands

git add <files, or folders>	Adds a new files to repo(via config file)
git rm <files or folders>	Removes files, or folders from GitHub
git add .	Adds all files and folders
git commit	Save changes to config file, and prepare pushing it to the server
git push	Pushes the new changes back to server, and updates the remote repository
git pull	Pulls the latest version of the repository from a GitHub
git status	Checks the overall status of repository
git branch -d branch_name or git branch -D branch_name	Removes the named branch
git revert dd61ab32	Removes the last commit mistakenly pushed to the GitHub By commit ID. So the commit id in this case is dd61ab32

git push origin --delete <branch_name>	Delete a remote GIT branch
git push <remote_name> :<branch_name> Remote name means this: git push origin :<branch_name>	Delete a remote GIT branch

The end of part 1
Please read the GitHub guide part 2