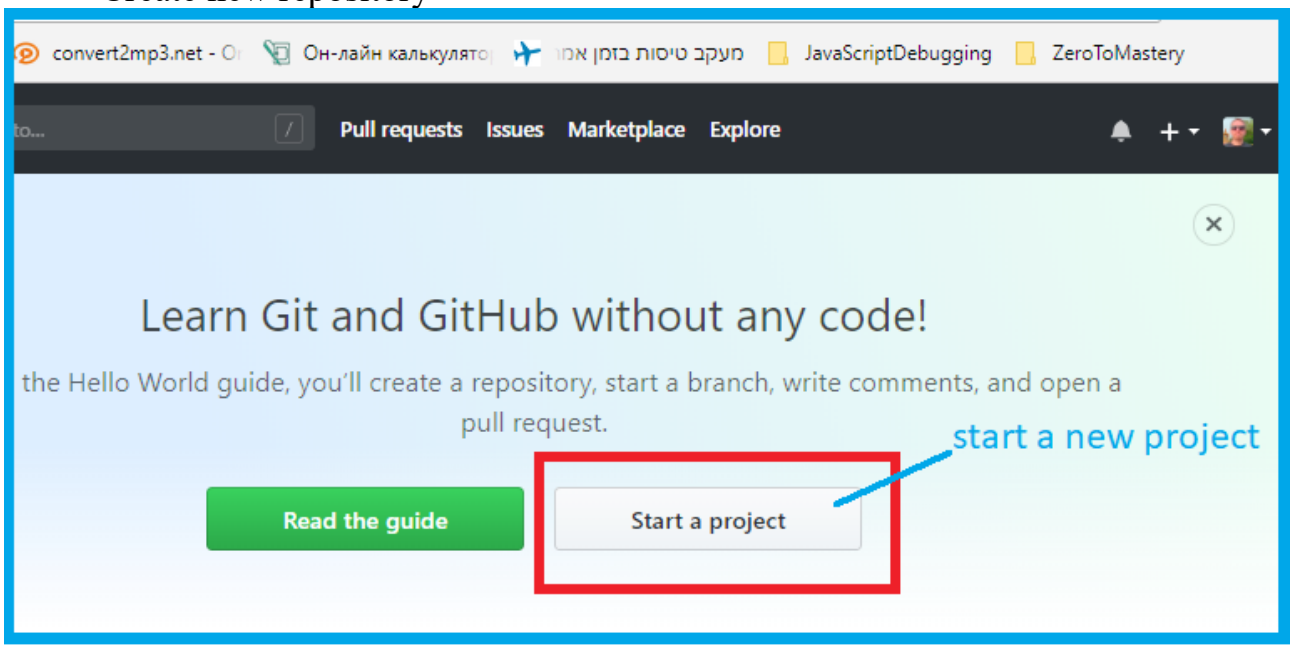
Git and GitHub (Source Control)

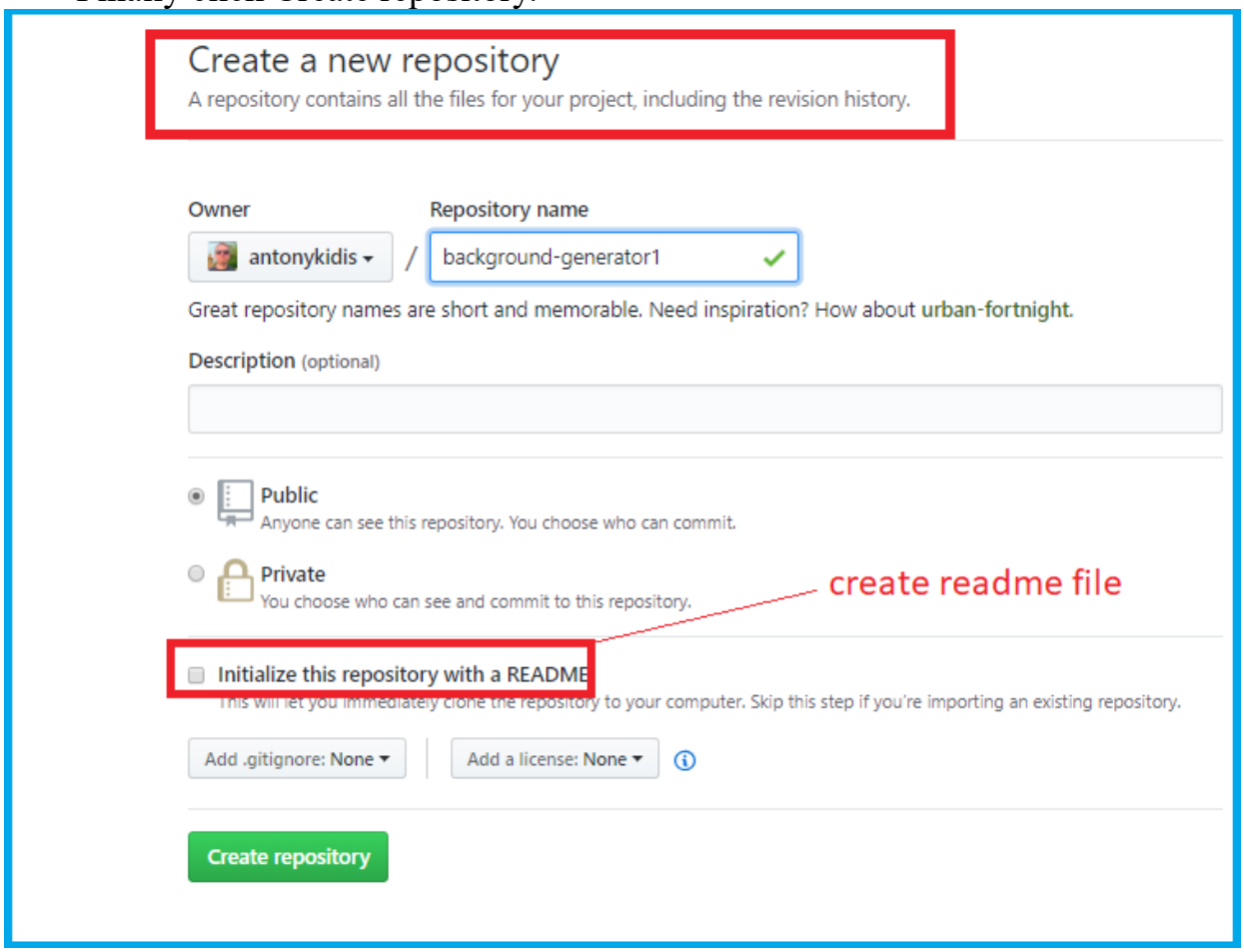
Step 1

- |Navigate to https://github.com/
- Create new repository

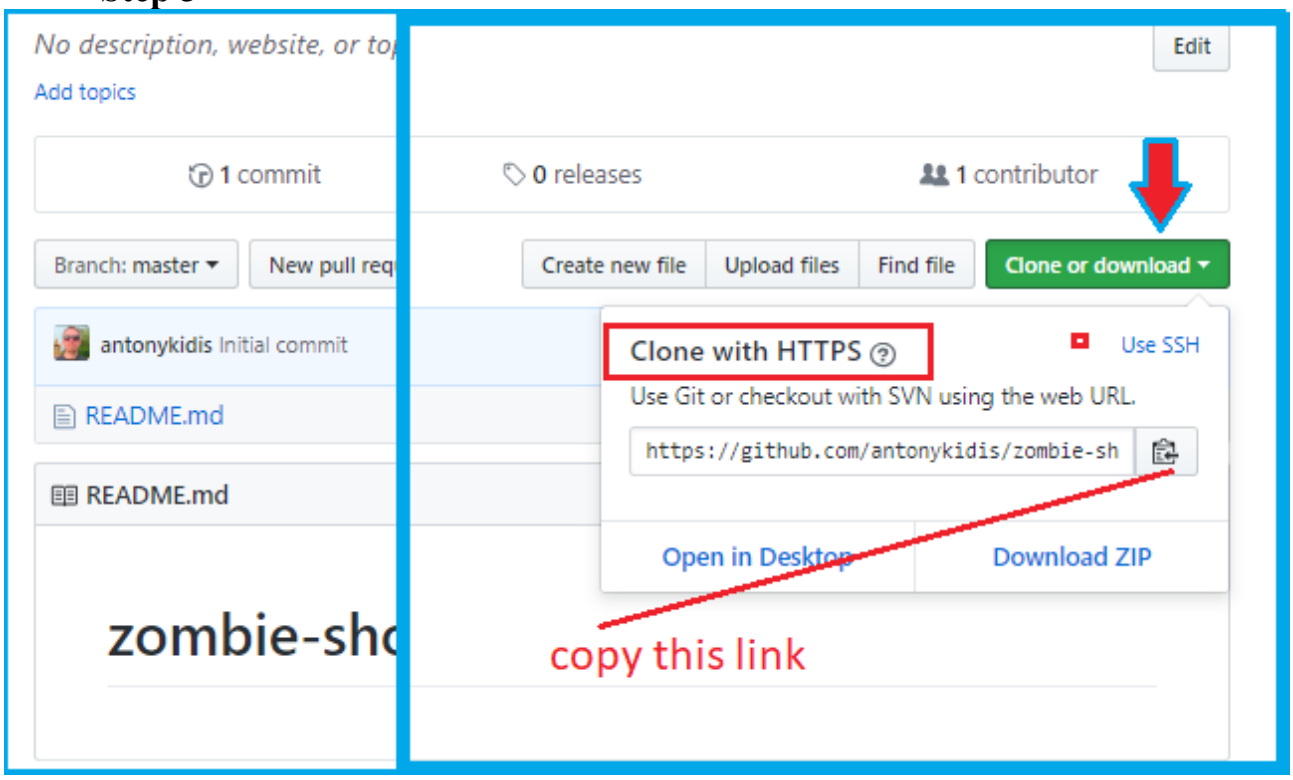


Step 2

- Name your project.
- check to create a readme file.
- Finally click Create repository.



- **Step 3**



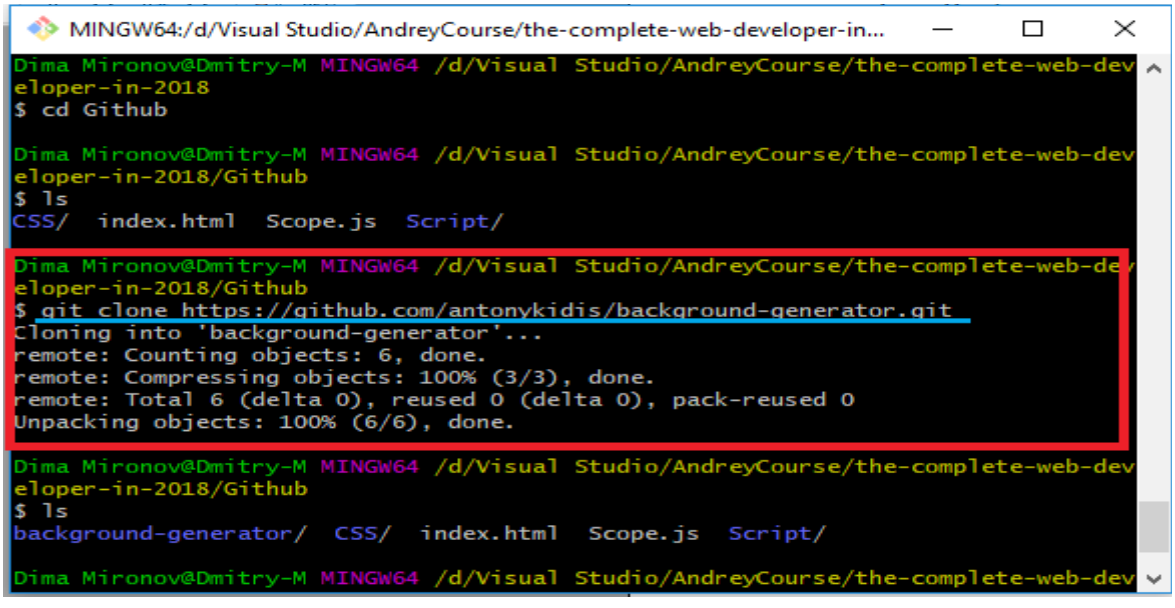
- select use HTTPS, then copy the **URL** to a clip board

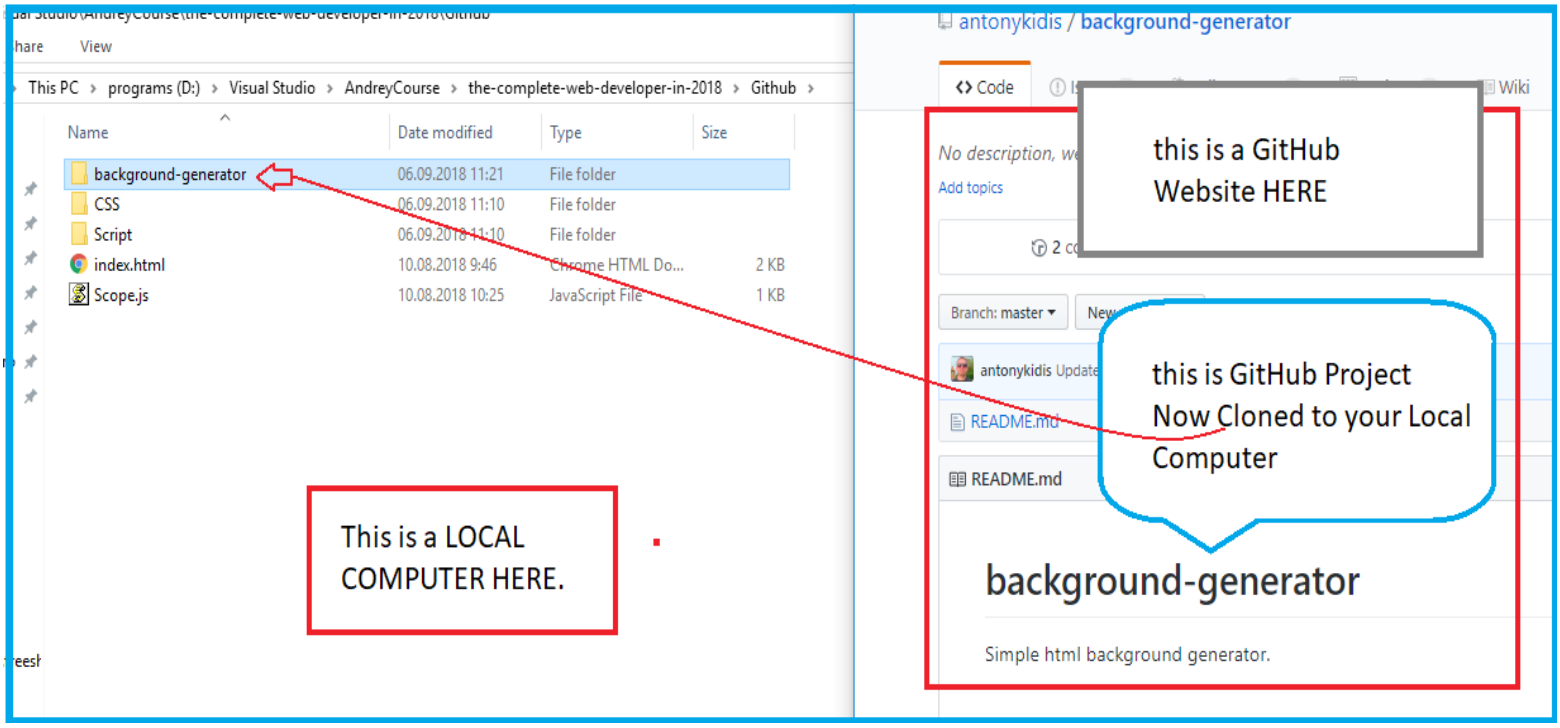
Step 4

- Open terminal(bash.exe/powershell) or other shell installed on your PC and navigate(cd...) to your working folder - the place where you build your project locally (on your computer)
- **cd *to your folder***
- Example:
- `Dima Mir@Dmitry-M MINGW64 /d/Visual Studio/AndreyCourse/the-complete-web-developer-in-2018/BackgroundGenerator`

Step 5

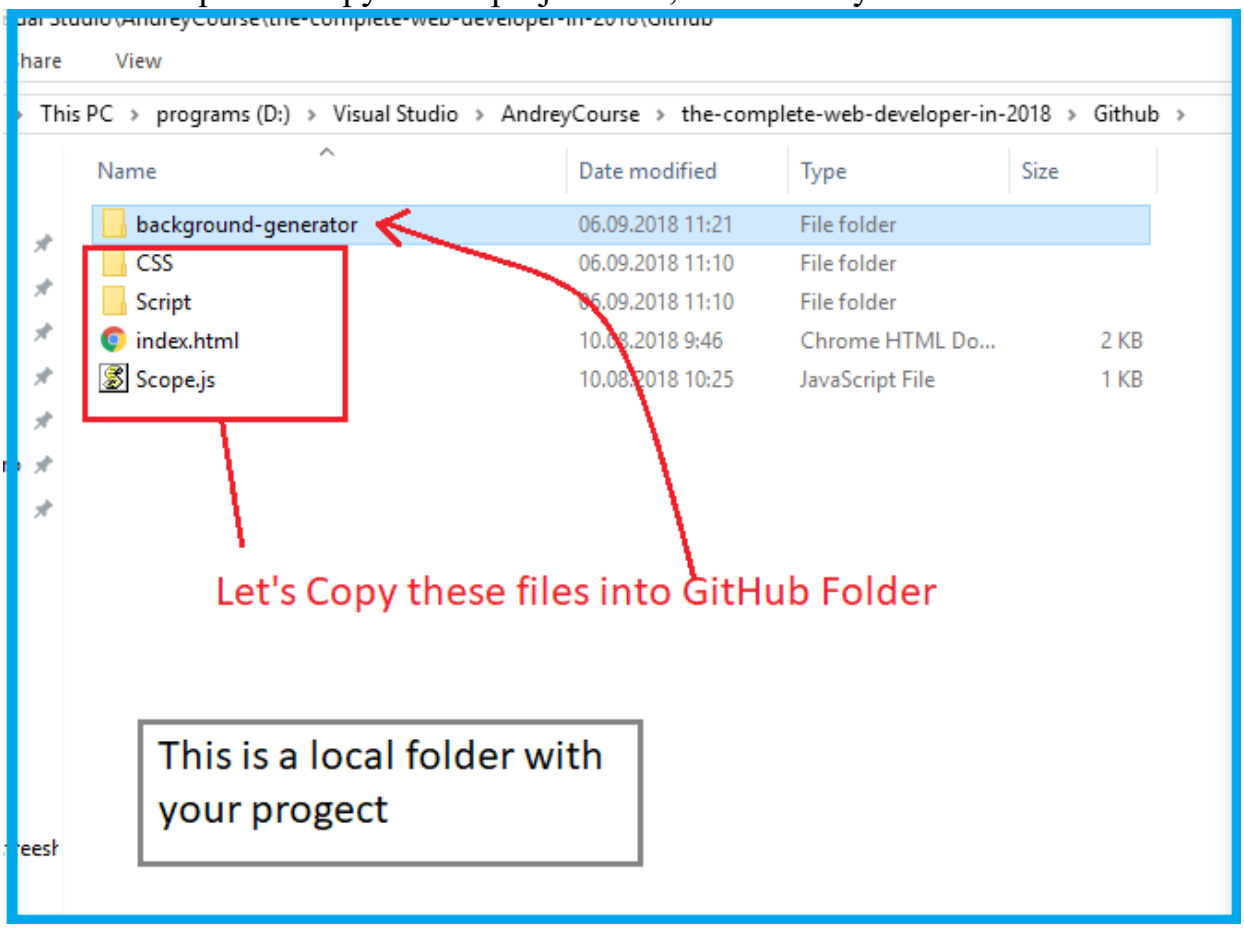
- Once you inside the project folder, open terminal window
- and enter the following command.
- git clone https://github.com/antonykidis/background-generator.git
- The above command will **COPY** GitHub's project to your **LOCAL** project's
- The background-generator cloned now to a local computer





Step6

- Next step is to Copy all the project files, into a newly cloned folder



Step 7

- Next Step is to CD into background-generator folder
- cd background generator will bring us to a similar output

```
Dima Mironov@Dmitry-M MINGW64 /d/Visual Studio/AndreyCourse/the-complete-web-dev
eloper-in-2018/Github
$ cd background-generator

Dima Mironov@Dmitry-M MINGW64 /d/Visual Studio/AndreyCourse/the-complete-web-dev
eloper-in-2018/Github/background-generator (master)
$
```

- We now can view all the files as we usually do in terminal.
- By typing ls.
- We'll get the following output:

```
Dima Mironov@Dmitry-M MINGW64 /d/Visual Studio/AndreyCourse/the-comp
eloper-in-2018/Github/background-generator (master)
$ ls
CSS/  index.html  README.md  Scope.js  Script/

Dima Mironov@Dmitry-M MINGW64 /d/Visual Studio/AndreyCourse/the-comp
eloper-in-2018/Github/background-generator (master)
$
```

This folder wil contain the project files plus the readme file

Step 8

- We can now check the status of our files
- use the following command
- git status
- The above will invoke the following output

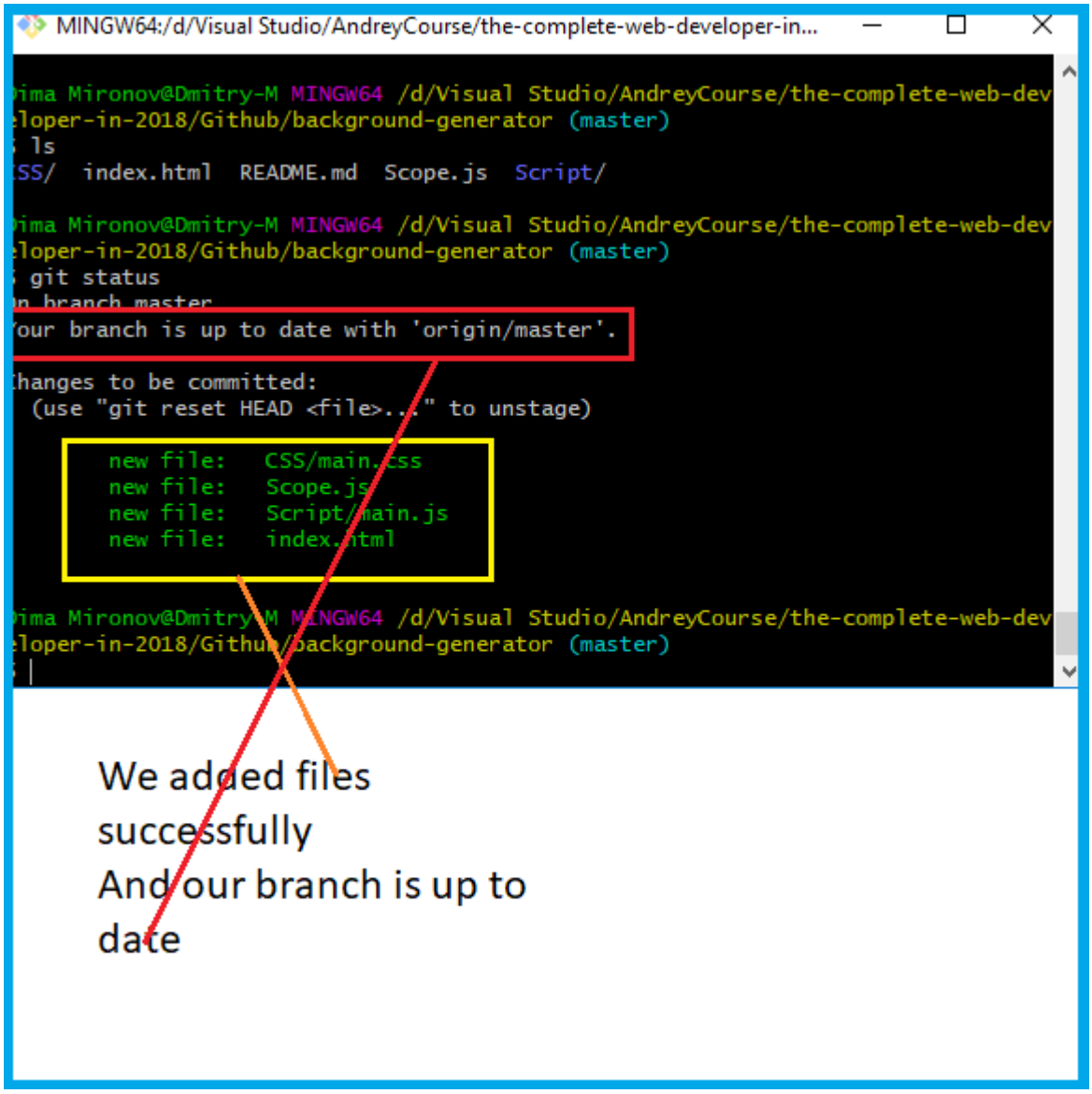
```
MINGW64:/d/Visual Studio/AndreyCourse/the-complete-web-developer-in...
Dima Mironov@Dmitry-M MINGW64 /d/Visual Studio/AndreyCourse/the-complete-web-dev
eloper-in-2018/Github/background-generator (master)
$ ls
CSS/  index.html  README.md  Scope.js  Script/

Dima Mironov@Dmitry-M MINGW64 /d/Visual Studio/AndreyCourse/the-complete-web-dev
eloper-in-2018/Github/background-generator (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)

        CSS/
        Scope.js
        Script/
        index.html

nothing added to commit but untracked files present (use "git add" to track)

Dima Mironov@Dmitry-M MINGW64 /d/Visual Studio/AndreyCourse/the-complete-web-dev
eloper-in-2018/Github/background-generator (master)
$
```

- As we see we have **Untracked** files

- We can now add these files to a GitHub Configuration file before upload it on Github

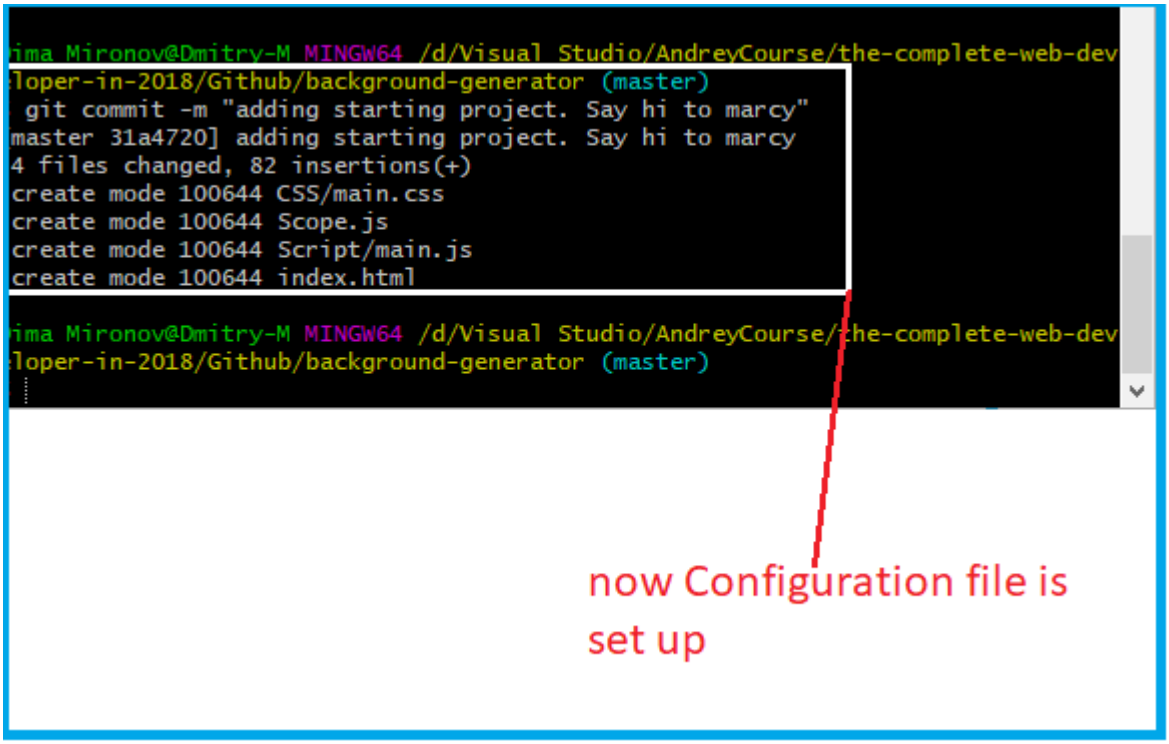
Step 9

- Let's **Add** our local project's (*Untracked files*) to a **Special configuration file**
- Follow the bellow syntax:
- **git add CSS** (folder)
- **git add Script**(folder)
- **git add Scope.js** (file)
- **git add index.html** **(file)**
-



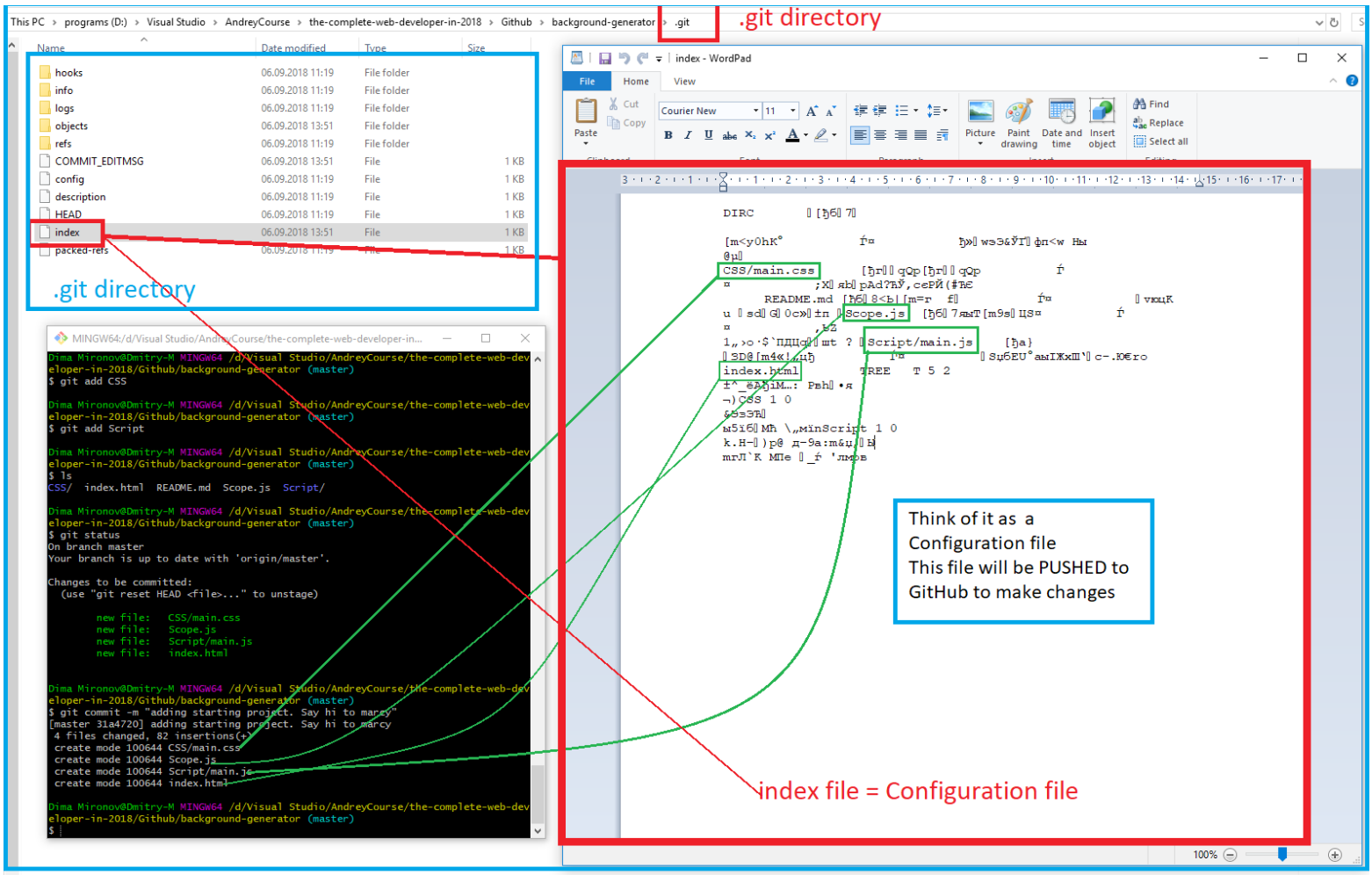
Step 10

- Finally We have to commit the ***special configuration file***
- commit = means to save all the changes to configuration file
- another words:
- *Yes, these are the changes we've made to the configuration file so far.*
- *we commit it (we approve it)*
- Okay we've just set up a special configuration file to be sent to the GitHub. Great!
- Type git commit -m “enter your message here”



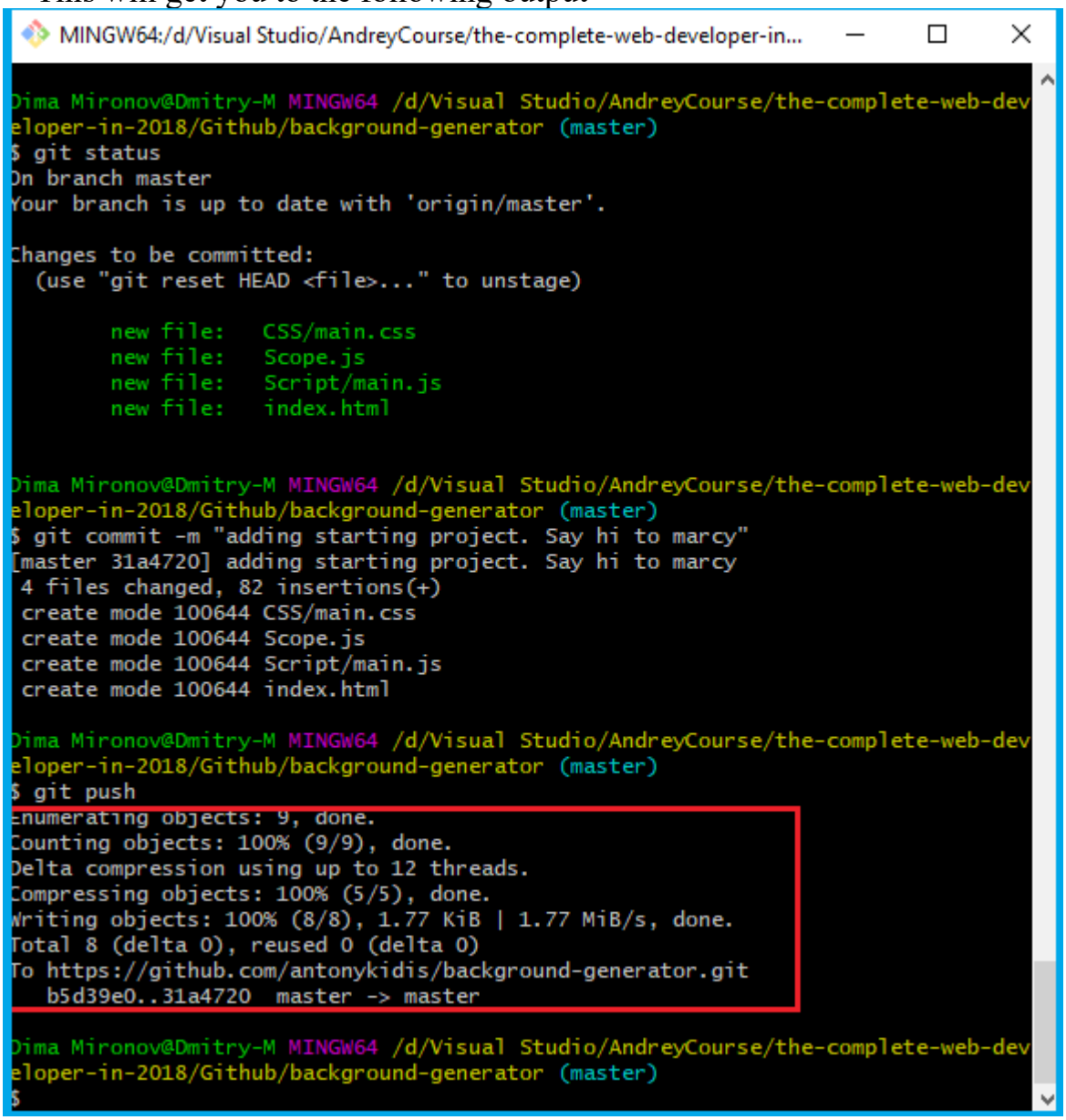
Here is the configuration file

- A .git directory contains a file called index.
- You can think of it as a configuration file



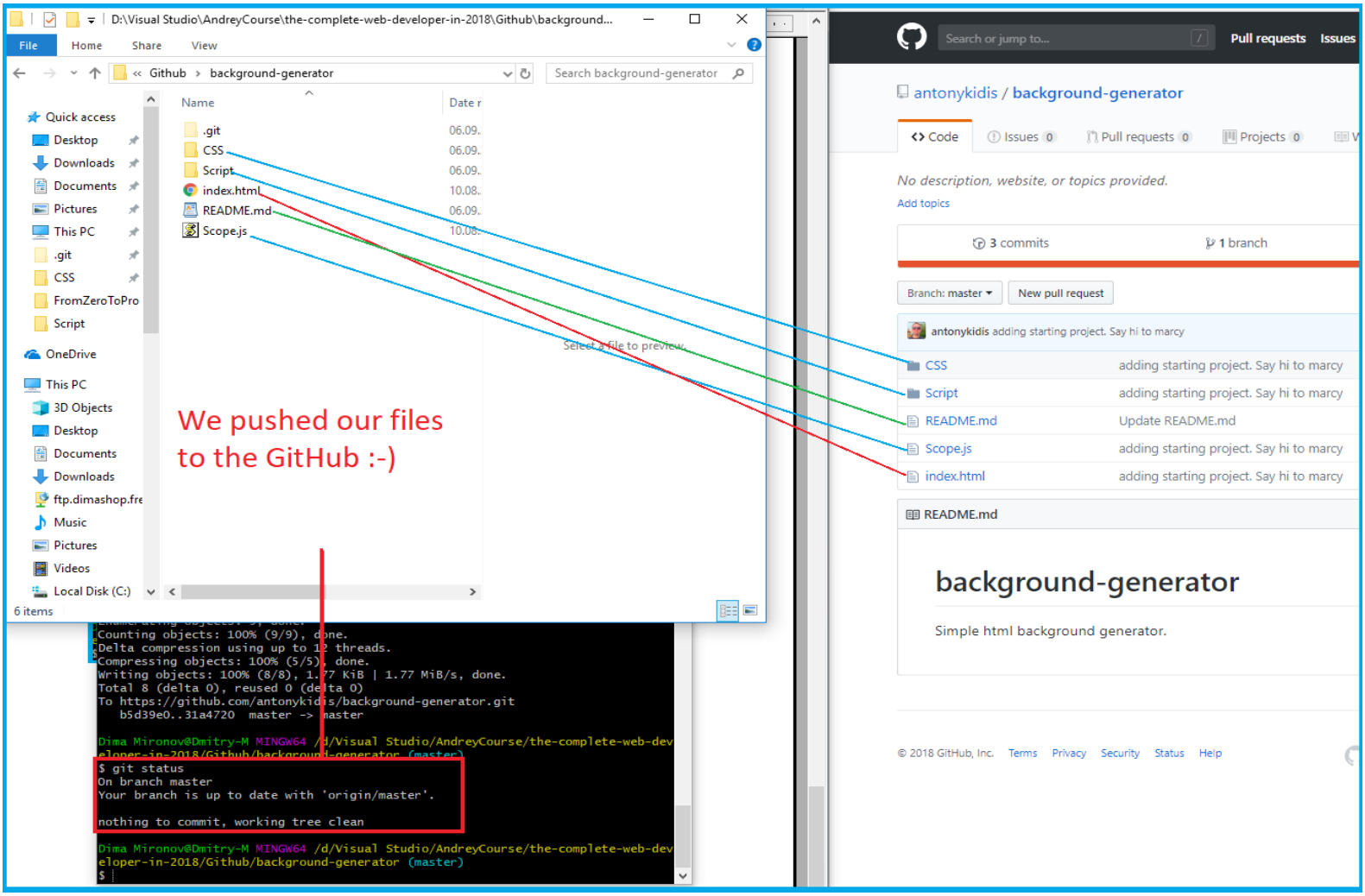
- **Step 11**
- Finally **Push** the configuration file to a GitHub server

- Use the following command
- get push
- This will get you to the following output



Step 12

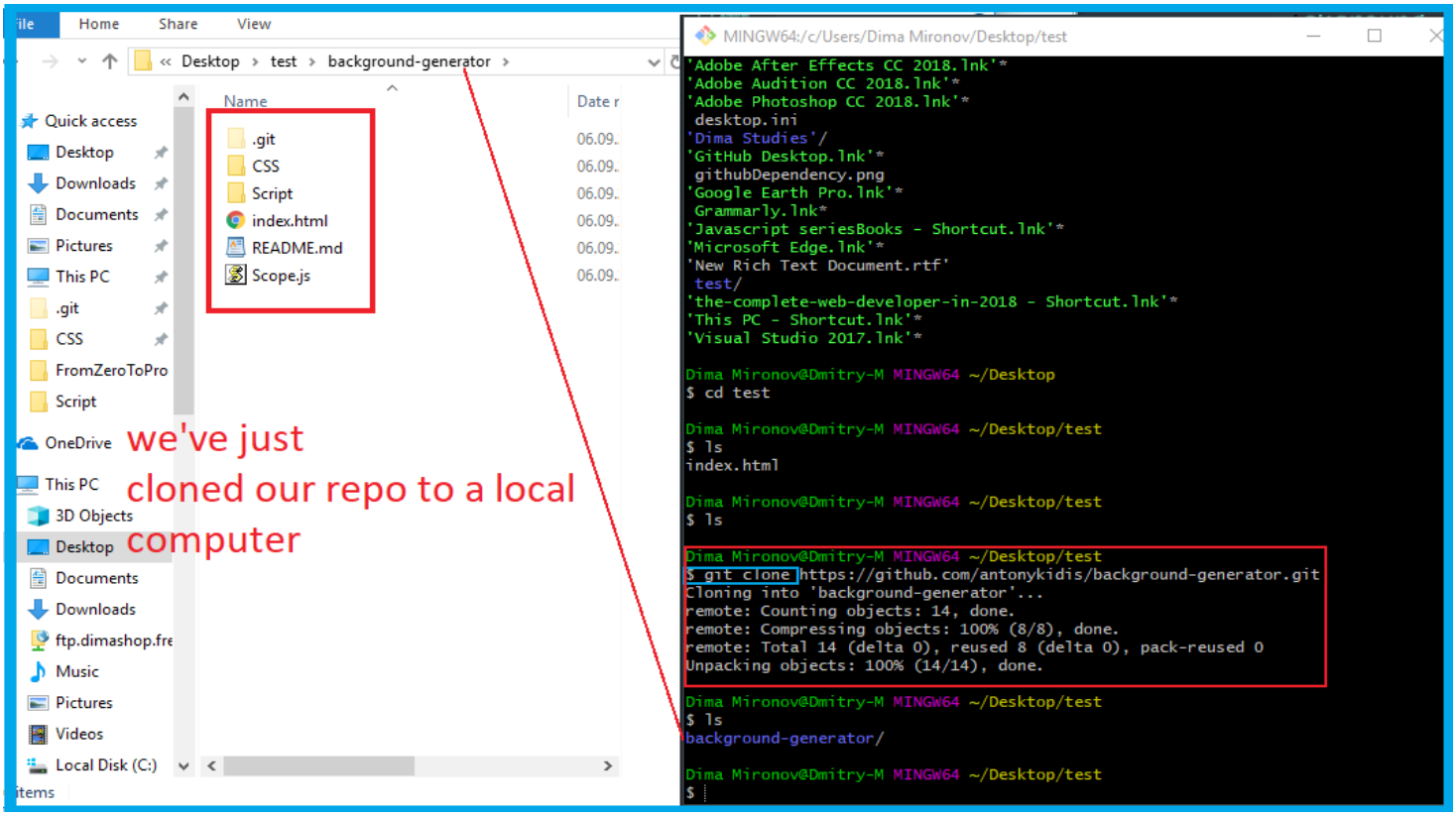
- Test the pushed files on https://github.com



We successfully pushed files on GitHub

Next Step How to work on existing Repository?	Working on existing GitHub Repository

1. Go to your folder (where you want to put a GitHub repository)
2. Type the following command to copy GitHub's repository to your computer.
   - git clone https://github.com/antonykidis/background-generator.git
3. Check if a cloned files appears in the folder (see picture)

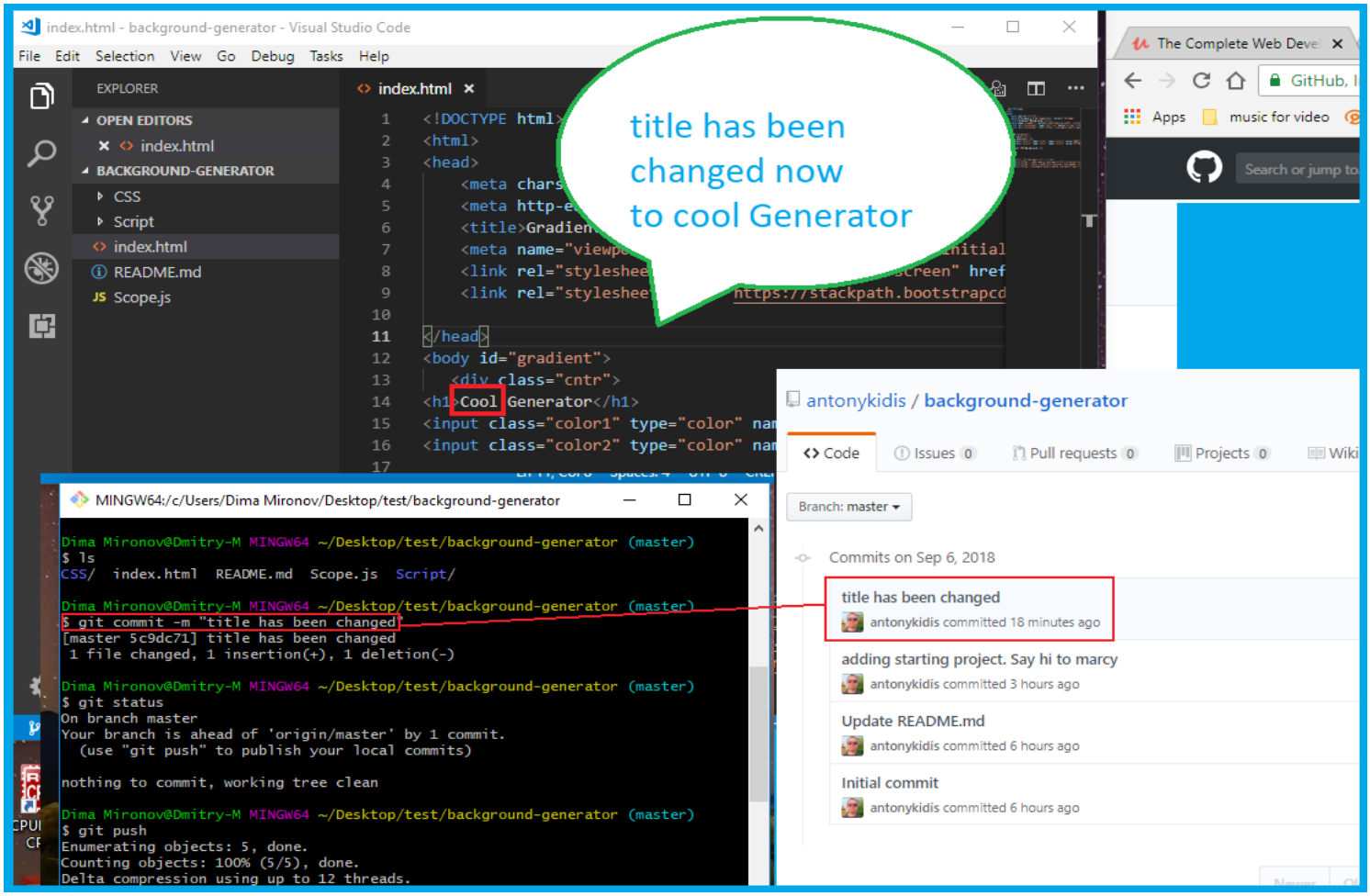


How To push a modified files back to GitHub?

1. Open **index html** and make some changes.
2. Then comeback to a terminal window and type git status .
3. You should see a similar picture

4. After modifying the **index.html we have to add it to** the configuration file again.
5. git add index.html
6. Then type git commit to update, and prepare the config file.
7. finally type git push to push a configured file bac to a GitHub server.
8. finally go to https://github.com and see if the repository been updated

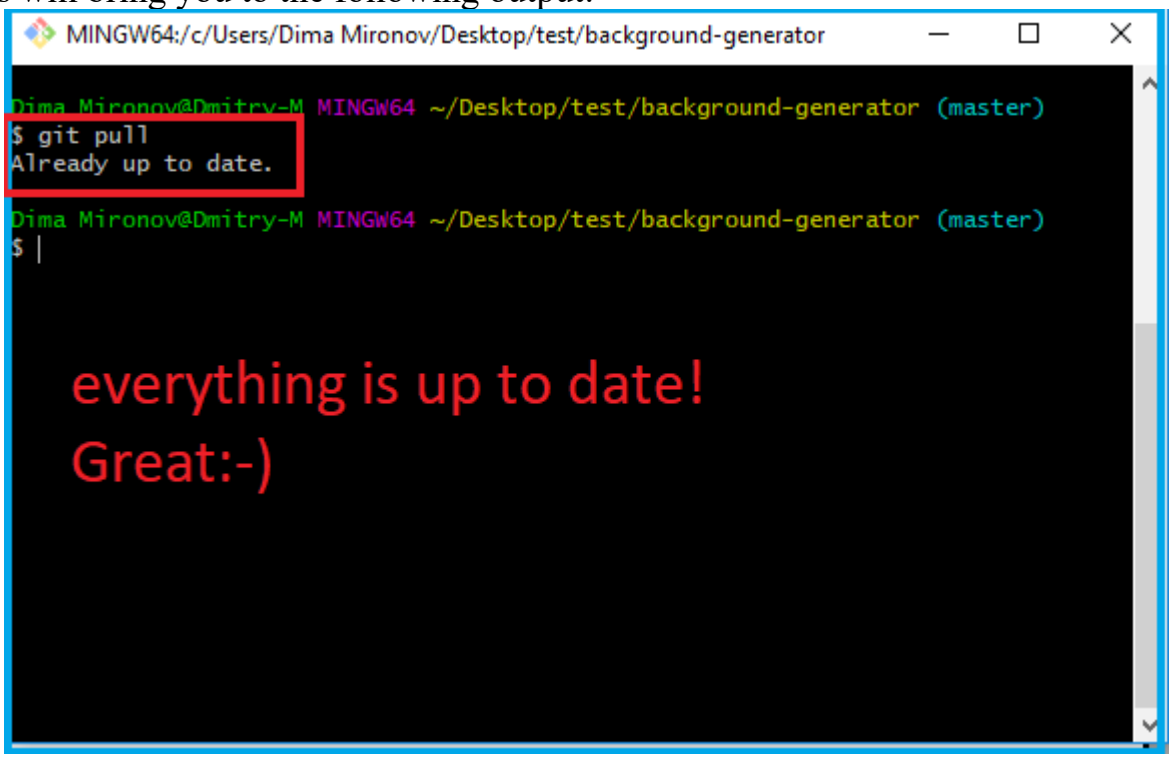
See the illustration



We now can navigate to a *commits link* and see the actual changes we made
As we see the title has changed

9. Finally we can Pull the latest version from a Github
like this

Type the following command git pull
This will bring you to the following output:



Always follow the below command rules
But these vary depending on the situation. Because sometimes you will need to get pull first. Then start working on the project (to get the latest version available)

Useful commands	
• git add	adds to Config.File
• git add .	adds all files
• git commit	closing the config.file and prepare for pushing into the server
• git push	pushes the configuration file and updates the repository
• git pull	PULL THE LATEST CHANGES