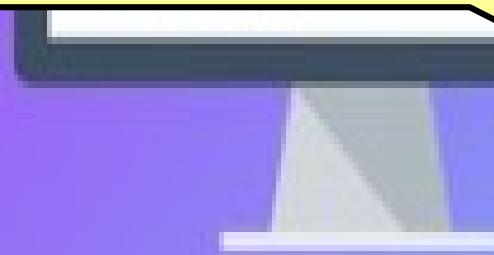




*Hello, my dear student of Zero To Mastery 2018.
Before you start reading the guide, please remember the following statement:*

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

C:\Users\Dima Mironov\Desktop\test\background-generator\.git

Share View

> test > background-generator > .git

Search .git

Name	Date modified	Type	Size
hooks	27.09.2018 8:29	File folder	
info	27.09.2018 8:29	File folder	
logs	27.09.2018 8:29	File folder	
objects	27.09.2018 8:29	File folder	
refs	27.09.2018 8:29	File folder	
config	27.09.2018 8:29	File	1 KB
description	27.09.2018 8:29	File	1 KB
FETCH_HEAD	27.09.2018 8:29	File	1 KB
HEAD	27.09.2018 8:29	File	1 KB
index	27.09.2018 8:30	File	1 KB
ORIG_HEAD	27.09.2018 8:29	File	1 KB
packed-refs	27.09.2018 8:29	File	1 KB

C:\Users\Dima Mironov\Desktop\Dima Studies\Andrei Negoie\The Complete Web D...

Share View

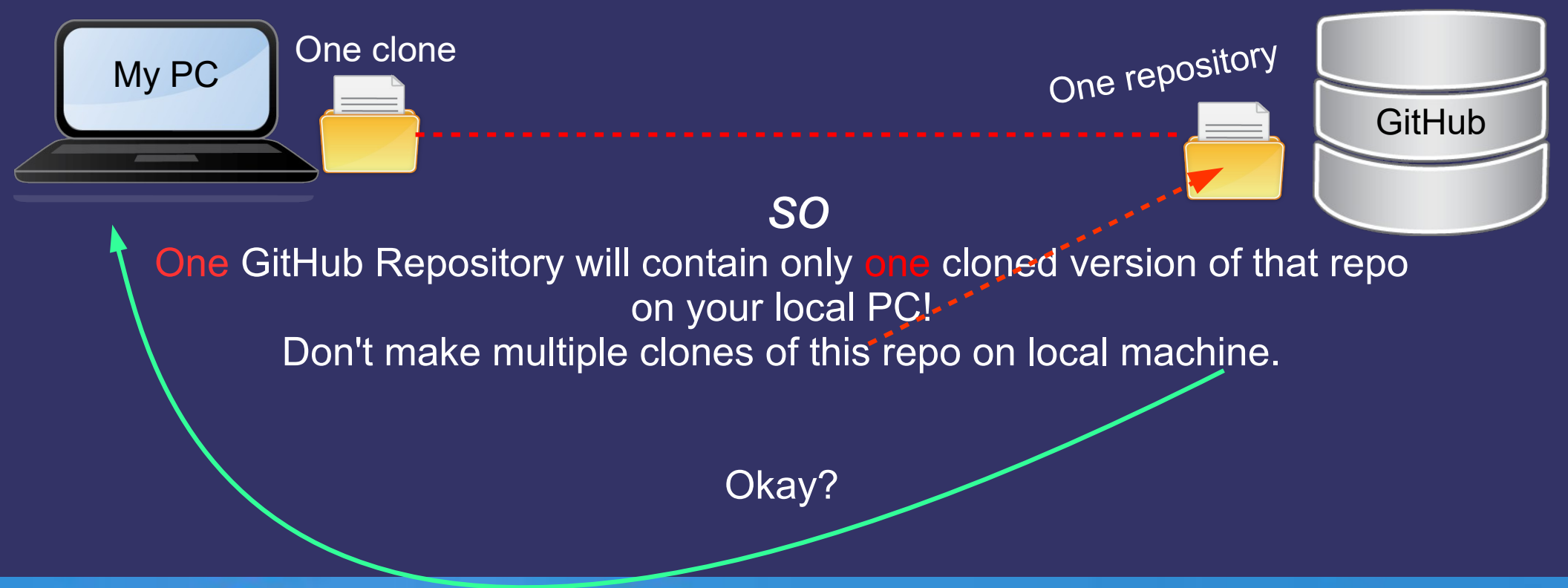
<< background-generator > .git

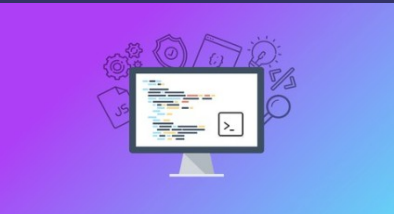
Search .git

Name	Date modified	Type
hooks	13.09.2018 8:25	File folder
info	13.09.2018 8:25	File folder
logs	13.09.2018 8:25	File folder
objects	27.09.2018 8:12	File folder
refs	13.09.2018 8:25	File folder
COMMIT_EDITMSG	27.09.2018 8:12	File
config	27.09.2018 8:14	File
description	13.09.2018 8:25	File
FETCH_HEAD	27.09.2018 8:24	File
HEAD	27.09.2018 8:09	File
index	27.09.2018 8:12	File
ORIG_HEAD	27.09.2018 8:24	File
packed-refs	13.09.2018 8:25	File

```
# pack-refs with: peeled fully-peeled sorted
fd8f4fe93b0078114b0f846b7e5345c9367f9e
refs/remotes/origin/master
```

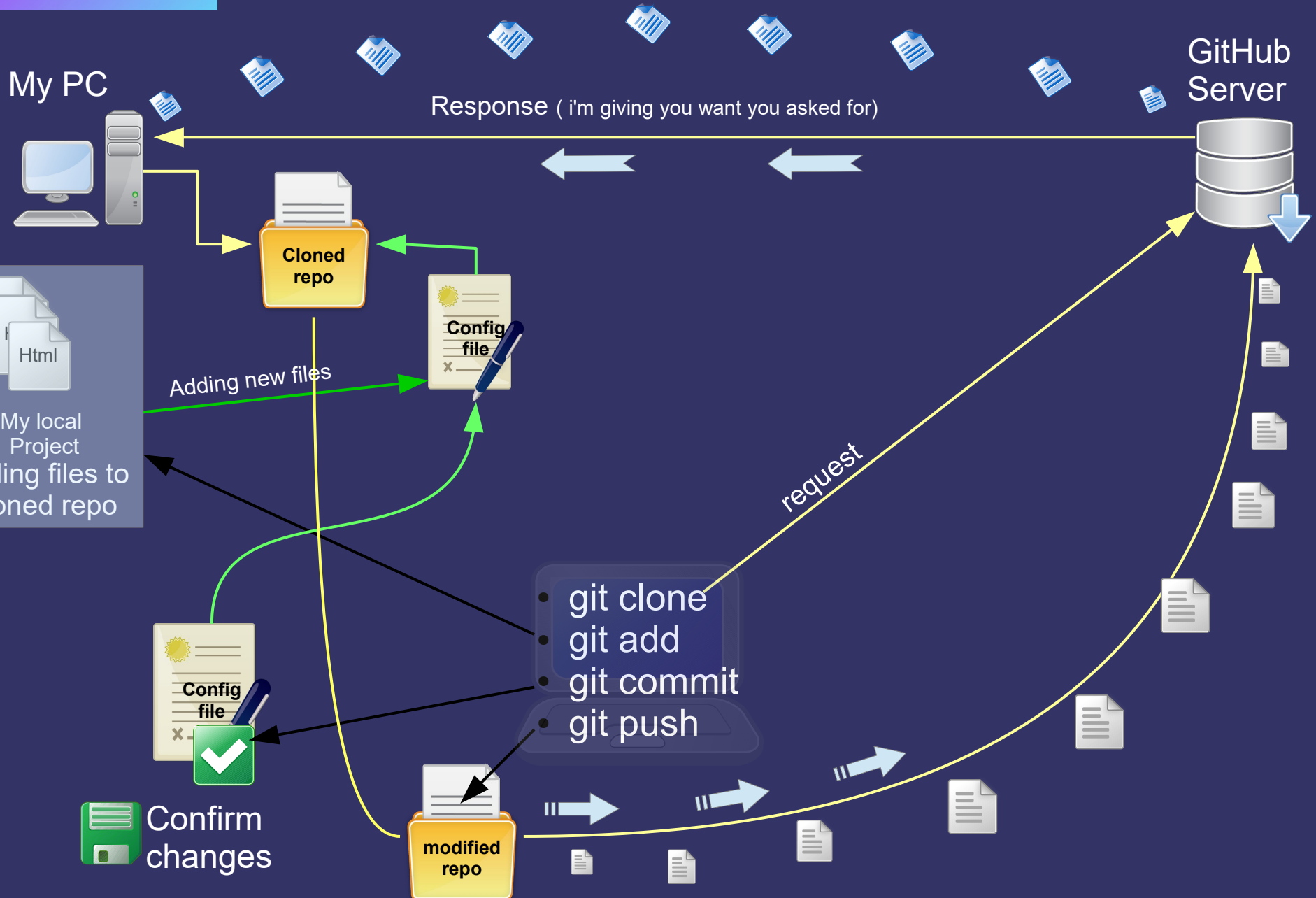
```
# pack-refs with: peeled fully-peeled sorted
59dc707c7403054f2ed7033ab04f5debca1947
refs/remotes/origin/master
```





Road Map

The general idea and the flow chart behind this guide



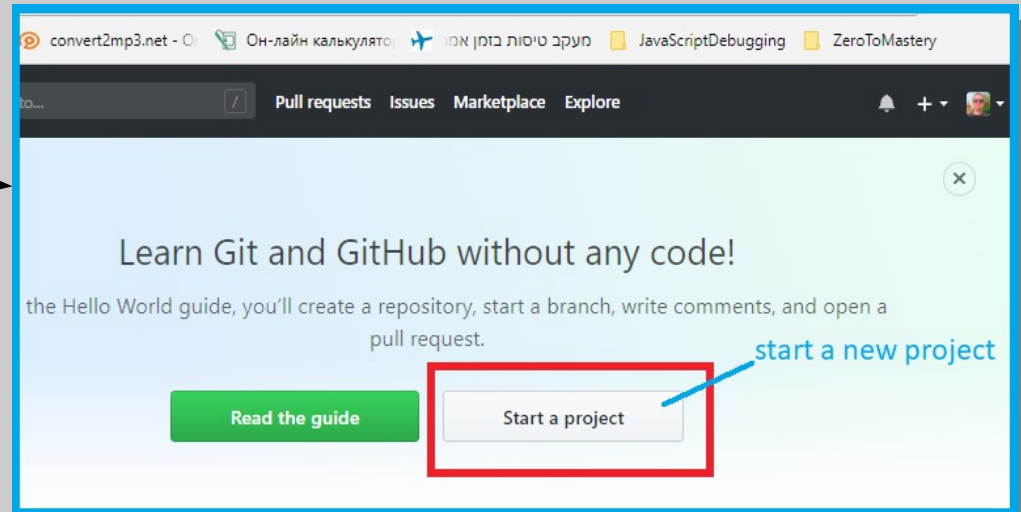
Git and GitHub Source Control

Part1 (Working on a new repository)

The Complete Web Developer in 2018: Zero to Mastery

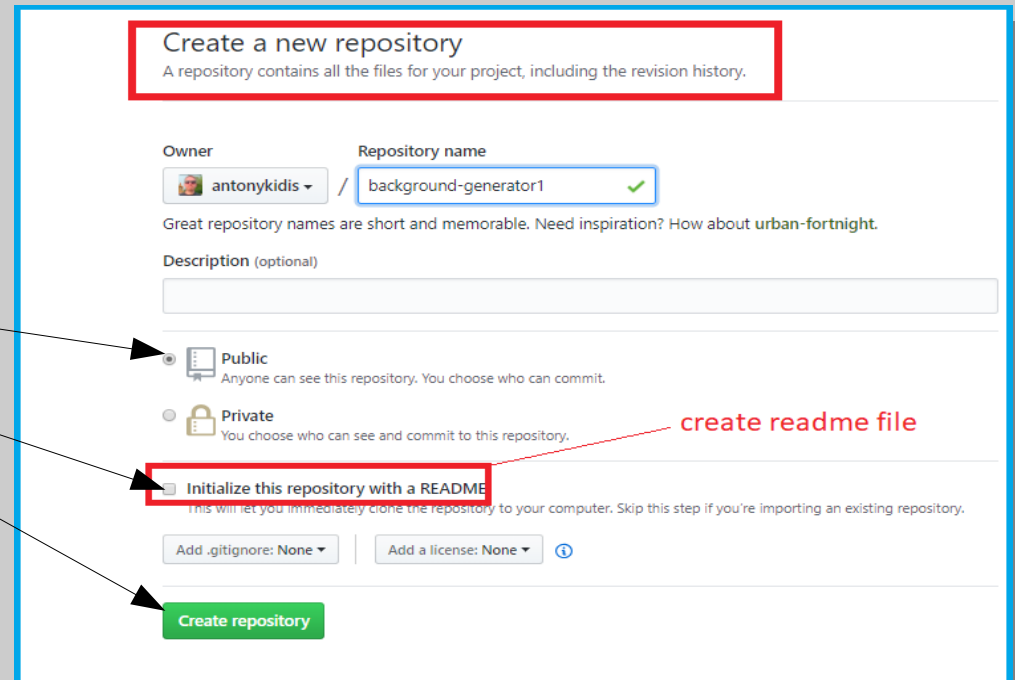
Step 1

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



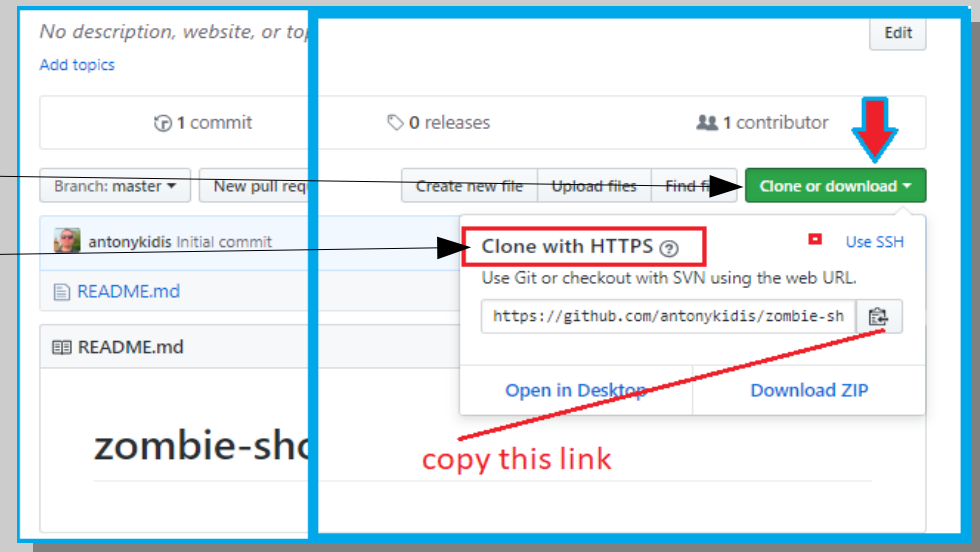
Step 2

- 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 preferred terminal.
- Navigate to a project you currently work on.
- Enter the following command:
`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

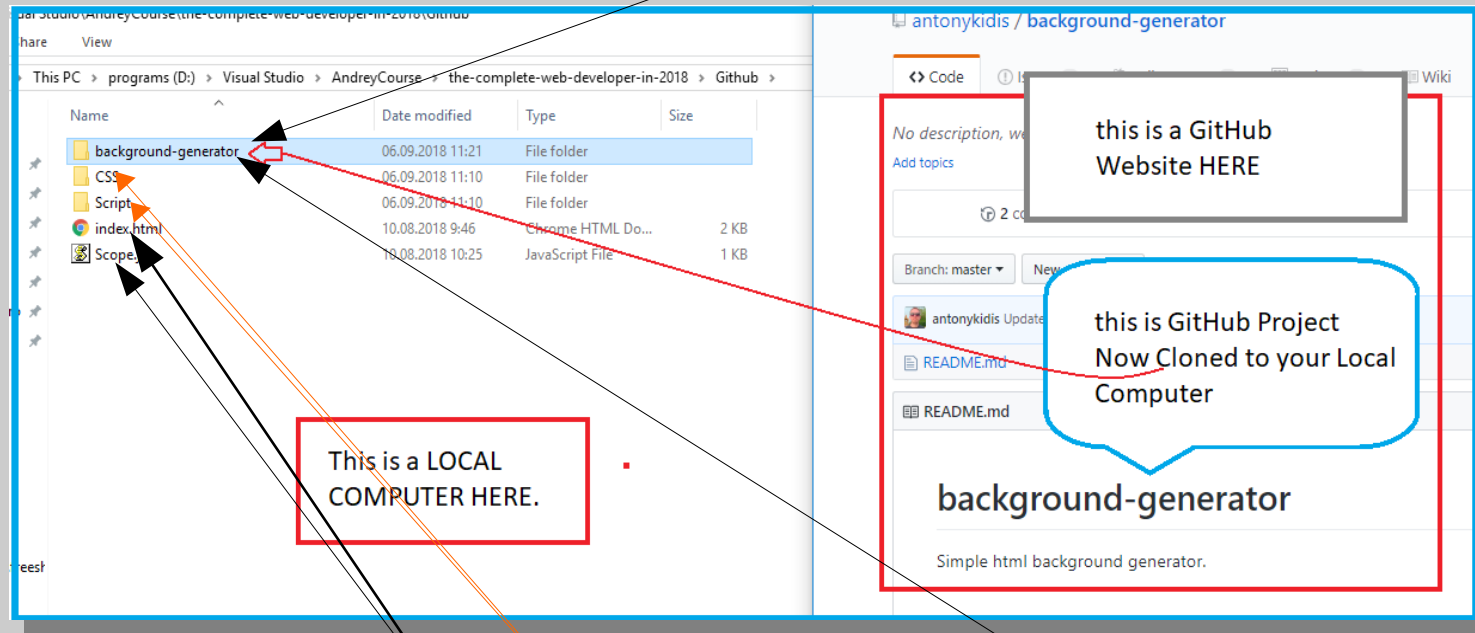
```
MINGW64:/d/Visual Studio/AndreyCourse/the-complete-web-developer-in-2018
Dima Mironov@Dmitry-M MINGW64 /d/Visual Studio/AndreyCourse/the-complete-web-developer-in-2018
$ cd Github
$ ls
CSS/ index.html Scope.js Script/

Dima Mironov@Dmitry-M MINGW64 /d/Visual Studio/AndreyCourse/the-complete-web-developer-in-2018/Github
$ git clone https://github.com/antonykidis/background-generator.git
Cloning into 'background-generator'...
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), done.

Dima Mironov@Dmitry-M MINGW64 /d/Visual Studio/AndreyCourse/the-complete-web-developer-in-2018/Github
$ ls
background-generator/ CSS/ index.html Scope.js Script/

Dima Mironov@Dmitry-M MINGW64 /d/Visual Studio/AndreyCourse/the-complete-web-developer-in-2018/Github
```

We've just copied our repo to a local computer.
You can see the **background-generator** appeared in the left window

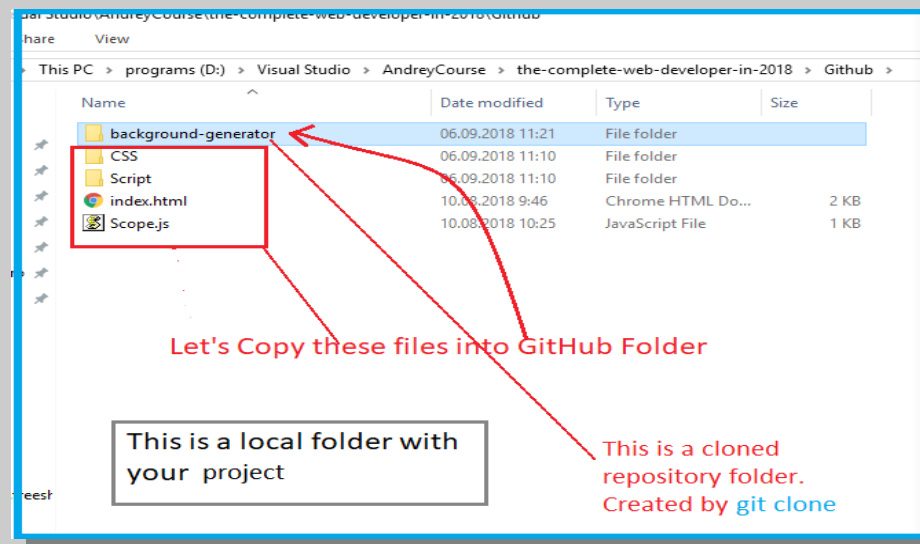


Step 5

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 directory 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

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

This folder will contain the project files plus the readme file

Step 7

- We now can check the status of our files since we've copied our files into the repository folder (communication 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

```
MINGW64:/d/Visual Studio/AndreyCourse/the-complete-web-developer-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-developer-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-developer-in-2018/Github/background-generator (master)
$
```



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

Adding files and folders to configuration file

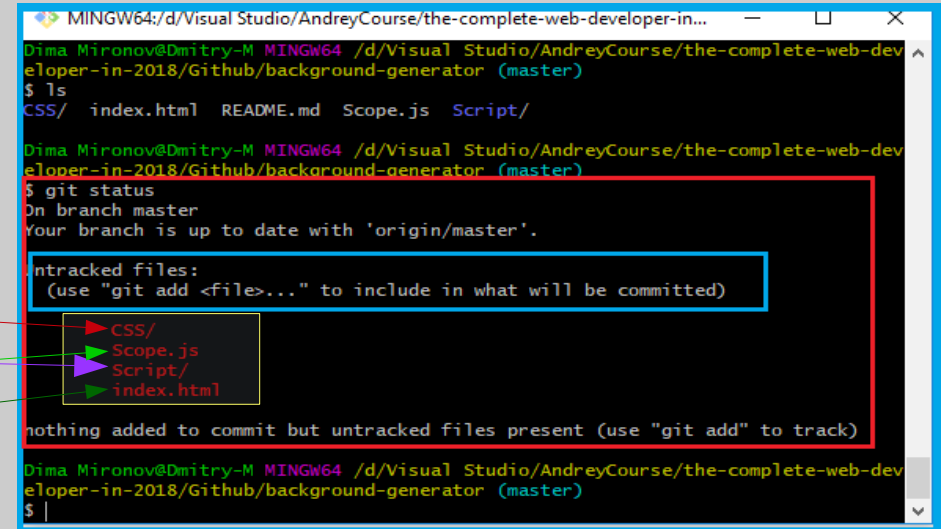
Step 8

- Let's Add untracked 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 .`



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

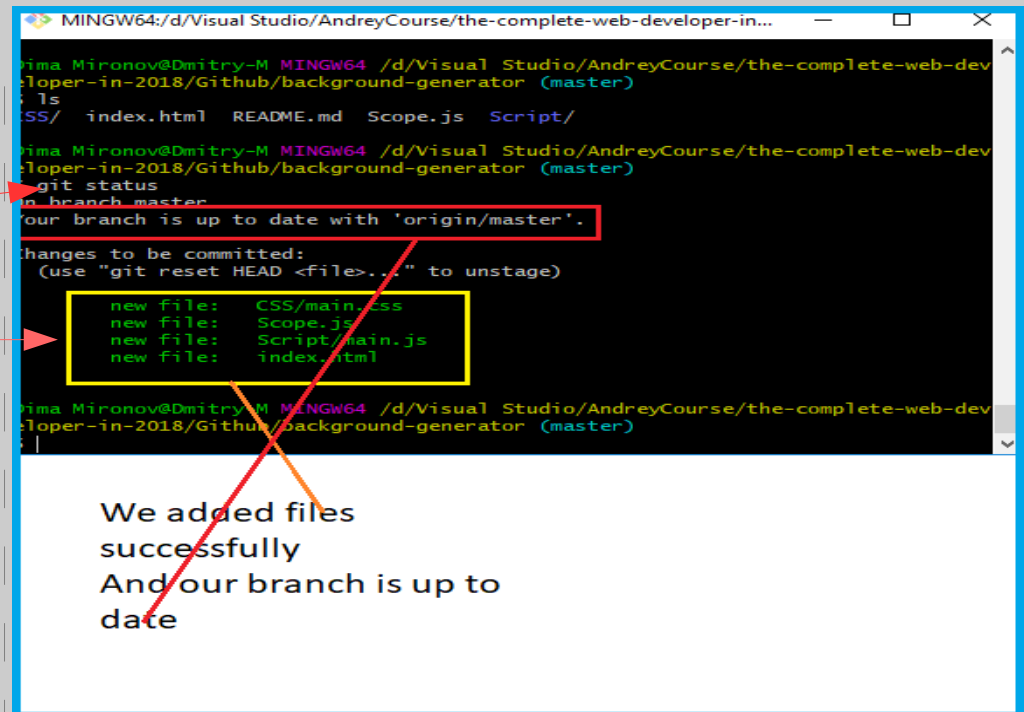
Dima Mironov@Dmitry-M MINGW64 /d/Visual Studio/AndreyCourse/the-complete-web-developer-in-2018/Github/background-generator (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

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)
```

Step 9

- After adding each file, and folder
- Type `git status`
 - You should see the following output
 - Okay we've just set the configuration file. Great!
 - Next step is to commit these files into the configuration file.



```
MINGW64:/d/Visual Studio/AndreyCourse/the-complete-web-developer-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-developer-in-2018/Github/background-generator (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:   CSS/main.css
        new file:   Scope.js
        new file:   Script/main.js
        new file:   index.html

We added files successfully
And our branch is up to date
```

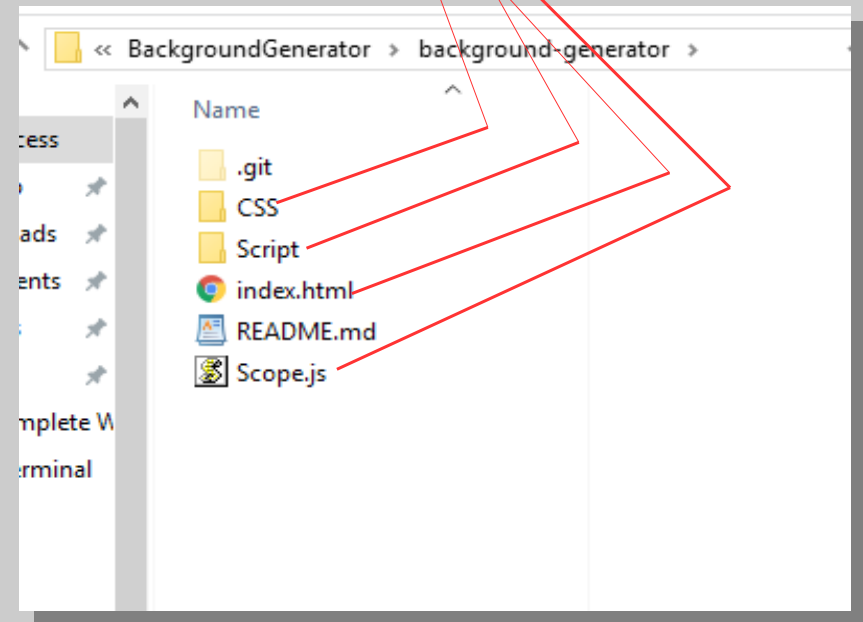
Commit the changes we've made in previous steps

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

```
ima Mironov@Dmitry-M MINGW64 /d/Visual Studio/AndreyCourse/the-complete-web-dev
loper-in-2018/Github/background-generator (master)
git commit -m "adding starting project. Say hi to marcy"
master 31a4720] adding starting project. Say hi to marcy
4 files changed, 82 insertions(+)
create mode 100644 CSS/main.css
create mode 100644 Scope.js
create mode 100644 Script/main.js
create mode 100644 index.html
```



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

his PC > programs (D:) > Visual Studio > AndreyCourse > the-complete-web-developer-in-2018 > Github > background-generator > .git

Name	Date modified	Type	Size
hooks	06.09.2018 11:19	File folder	
info	06.09.2018 11:19	File folder	
logs	06.09.2018 11:19	File folder	
objects	06.09.2018 13:51	File folder	
refs	06.09.2018 11:19	File folder	
COMMIT_EDITMSG	06.09.2018 13:51	File	1 KB
config	06.09.2018 11:19	File	1 KB
description	06.09.2018 11:19	File	1 KB
HEAD	06.09.2018 11:19	File	1 KB
index	06.09.2018 13:51	File	1 KB
packed-refs	06.09.2018 11:19	File	1 KB

```
MINGW64:/d/Visual Studio/AndreyCourse/the-complete-web-developer-in-2018/Github/background-generator (master)
$ git add CSS

Dima Mironov@Dmitry-M MINGW64 /d/Visual Studio/AndreyCourse/the-complete-web-developer-in-2018/Github/background-generator (master)
$ git add Script

Dima Mironov@Dmitry-M MINGW64 /d/Visual Studio/AndreyCourse/the-complete-web-developer-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-developer-in-2018/Github/background-generator (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:   CSS/main.css
    new file:   Scope.js
    new file:   Script/main.js
    new file:   index.html

Dima Mironov@Dmitry-M MINGW64 /d/Visual Studio/AndreyCourse/the-complete-web-developer-in-2018/Github/background-generator (master)
$ git commit -m "adding starting project. Say hi to marcy"
[master 31a4720] adding starting project. Say hi to marcy
 4 files changed, 82 insertions(+)
 create mode 100644 CSS/main.css
 create mode 100644 Scope.js
 create mode 100644 Script/main.js
 create mode 100644 index.html

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

index - WordPad

CSS/main.css

Scope.js

Script/main.js

index.html

Think of it as a Configuration file
This file will be PUSHED to GitHub to make changes

index file = 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 successfully wrote objects to <http://github.com...>

```
Dima Mironov@Dmitry-M MINGW64 /d/Visual Studio/AndreyCourse/the-complete-web-developer-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

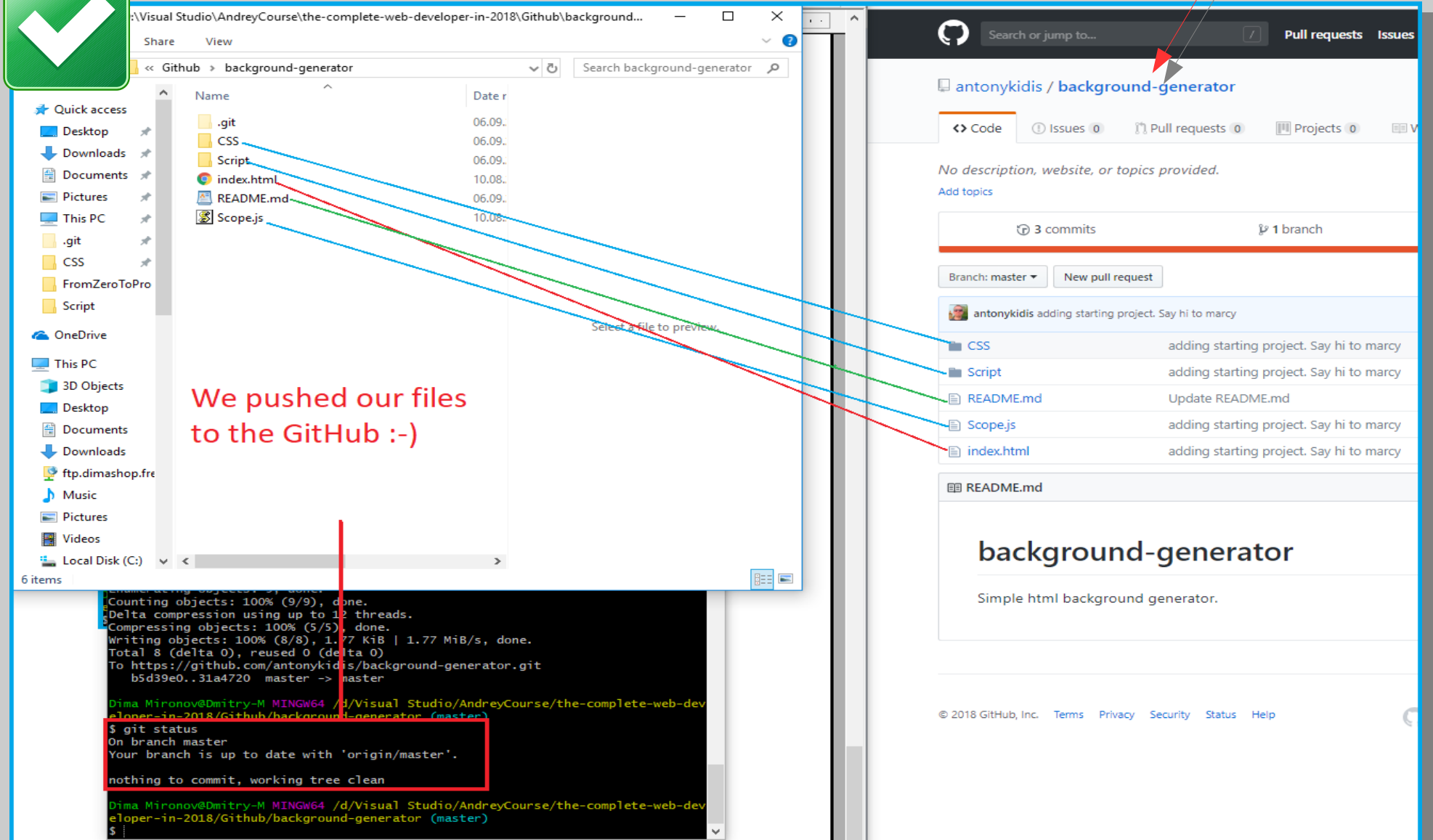
Step 12

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 successfully added a new files into GitHub repository



Visual Studio window showing the file explorer for the repository `background-generator`. The files listed are:

- .git
- CSS
- Script
- index.html
- README.md
- Scope.js

The terminal output shows the `git status` command result:

```
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

GitHub repository view for `antonykidis / background-generator`. The repository has 3 commits and 1 branch. The files listed are:

- CSS: adding starting project. Say hi to marcy
- Script: adding starting project. Say hi to marcy
- README.md: Update README.md
- Scope.js: adding starting project. Say hi to marcy
- index.html: adding starting project. Say hi to marcy

The repository description is: `background-generator` Simple html background generator.



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

The screenshot shows a Windows File Explorer window on the left and a Windows Terminal window on the right. The File Explorer window is displaying the contents of the 'test' folder on the Desktop, which now includes a '.git' folder, 'CSS', 'Script', 'index.html', 'README.md', and 'Scope.js'. A red box highlights these files. The Windows Terminal window shows the command prompt output for cloning the repository. A red arrow points from the 'git clone' command in the list to the terminal output. Another red arrow points from the 'test' folder in the File Explorer to the terminal output. A red text overlay says 'we've just cloned our repo to a local computer'.

we've just cloned our repo to a local computer

```
MINGW64:/c/Users/Dima Mironov/Desktop/test
'Adobe After Effects CC 2018.lnk'*
'Adobe Audition CC 2018.lnk'*
'Adobe Photoshop CC 2018.lnk'*
desktop.ini
'Dima Studies'/'
'GitHub Desktop.lnk'*
githubDependency.png
'Google Earth Pro.lnk'*
Grammarly.lnk*
'Javascript seriesBooks - Shortcut.lnk'*
'Microsoft Edge.lnk'*
'New Rich Text Document.rtf'
test/
'the-complete-web-developer-in-2018 - Shortcut.lnk'*
'This PC - Shortcut.lnk'*
'Visual Studio 2017.lnk'*

Dima Mironov@Dmitry-M MINGW64 ~/Desktop
$ cd test

Dima Mironov@Dmitry-M MINGW64 ~/Desktop/test
$ ls
index.html

Dima Mironov@Dmitry-M MINGW64 ~/Desktop/test
$ ls

Dima Mironov@Dmitry-M MINGW64 ~/Desktop/test
$ git clone https://github.com/antonykidis/background-generator.git
Cloning into 'background-generator'...
remote: Counting objects: 14, done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 14 (delta 0), reused 8 (delta 0), pack-reused 0
Unpacking objects: 100% (14/14), done.

Dima Mironov@Dmitry-M MINGW64 ~/Desktop/test
$ ls
background-generator/

Dima Mironov@Dmitry-M MINGW64 ~/Desktop/test
$
```


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)

```
MINGW64:/c:/Users/Dima Mironov/Desktop/Dima Studies/Andrei Negoie/T...
$ ls
CSS/  index.html  README.md  Scope.js  Script/

Dima Mironov@Dmitry-M MINGW64 ~/Desktop/Dima Studies/Andrei Negoie/The Complete
Web Developer 2018/BackGroundGenerator/background-generator (master)
$ start code index.html

Dima Mironov@Dmitry-M MINGW64 ~/Desktop/Dima Studies/Andrei Negoie/The Complete
Web Developer 2018/BackGroundGenerator/background-generator (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")

Dima Mironov@Dmitry-M MINGW64 ~/Desktop/Dima Studies/Andrei Negoie/The Complete
Web Developer 2018/BackGroundGenerator/background-generator (master)
$
```

The screenshot shows the Visual Studio Code editor with the `index.html` file open. The title has been changed to "Cool Generator". A speech bubble points to the change with the text "title has been changed now to cool Generator". Below the editor, a terminal window shows the command `$ git commit -m "title has been changed"` being executed. To the right, the GitHub web interface is visible, showing the commit history for the repository `antonykidis / background-generator`. The latest commit, "title has been changed", is highlighted with a red box.

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

4 commits

Showing 1 changed file with 1 addition and 1 deletion.

2 index.html

title has changed

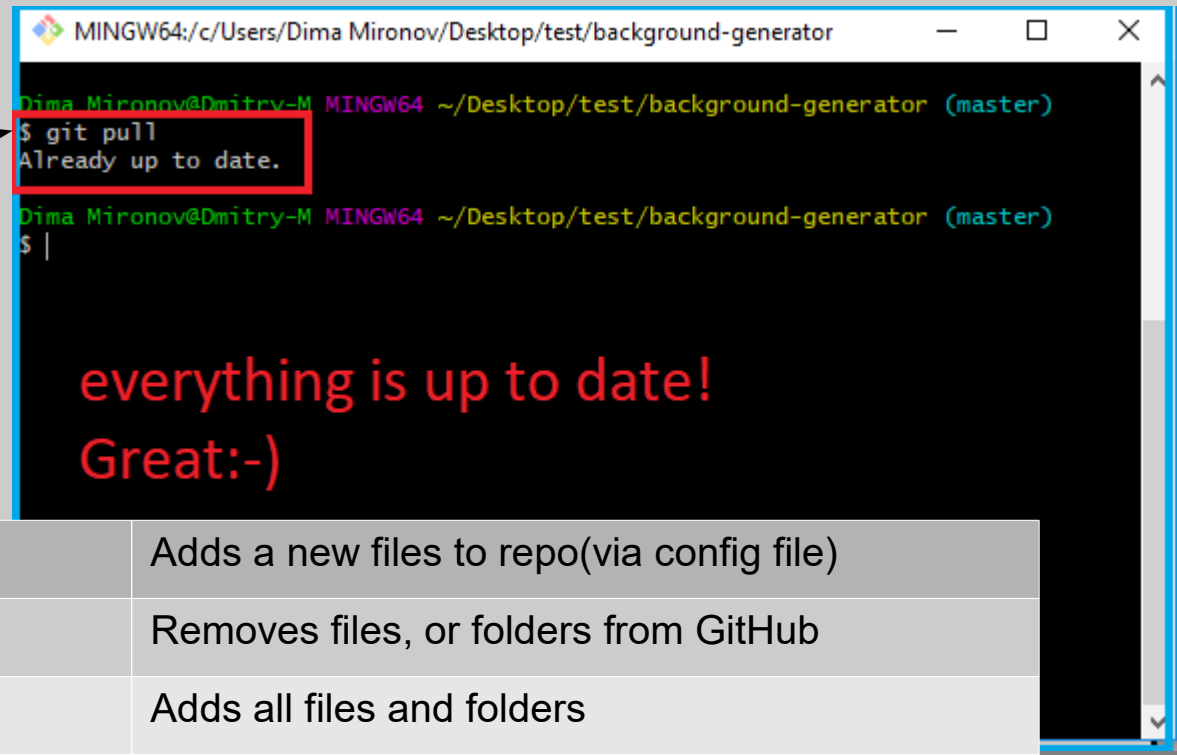
		@@ -11,7 +11,7 @@
11	11	</head>
12	12	<body id="gradient">
13	13	<div class="cntr">
14		- <h1>Background Generator</h1>
	14	+ <h1>Cool Generator</h1>
15	15	<input class="color1" type="color" name="color1" value="#00ff00">
16	16	<input class="color2" type="color" name="color2" value="#ff0000">
17	17	

0 comments on commit 5c9dc71

We can pull the latest version from GitHub

- Type **git pull**

Useful
commands



A terminal window titled 'MINGW64:/c/Users/Dima Mironov/Desktop/test/background-generator'. The prompt is 'Dima Mironov@Dmitry-M MINGW64 ~/Desktop/test/background-generator (master)'. The command '\$ git pull' is entered and highlighted with a red box. The output is 'Already up to date.'. Below the terminal output, the text 'everything is up to date!' and 'Great:-)' is written in red.

```
MINGW64:/c/Users/Dima Mironov/Desktop/test/background-generator
Dima Mironov@Dmitry-M MINGW64 ~/Desktop/test/background-generator (master)
$ git pull
Already up to date.
Dima Mironov@Dmitry-M MINGW64 ~/Desktop/test/background-generator (master)
$ |
```

everything is up to date!
Great:-)

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](#)