
How To Use Github

— Introduction To Programming —

What is Git

Git is a version control system that help programmers manage their projects efficiently. A Version Control System is where programmers can create different versions of their project and Git keeps track of these changes so that if he/she wants to go change between versions from version 3 to version 2 for example, git helps this process easily.

Git vs No Git (No Git version)

This is how people would usually manage different versions of their project without git. This kind of style would usually make a mess because there's



← File Version 1



← File Version 2

...



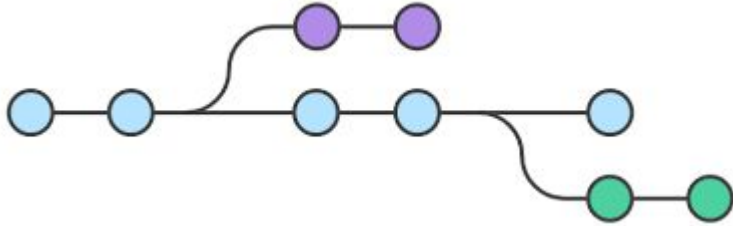
← File Version 25

too many files to save,

no control, no update

history, etc.

Git vs No Git (Git Version)



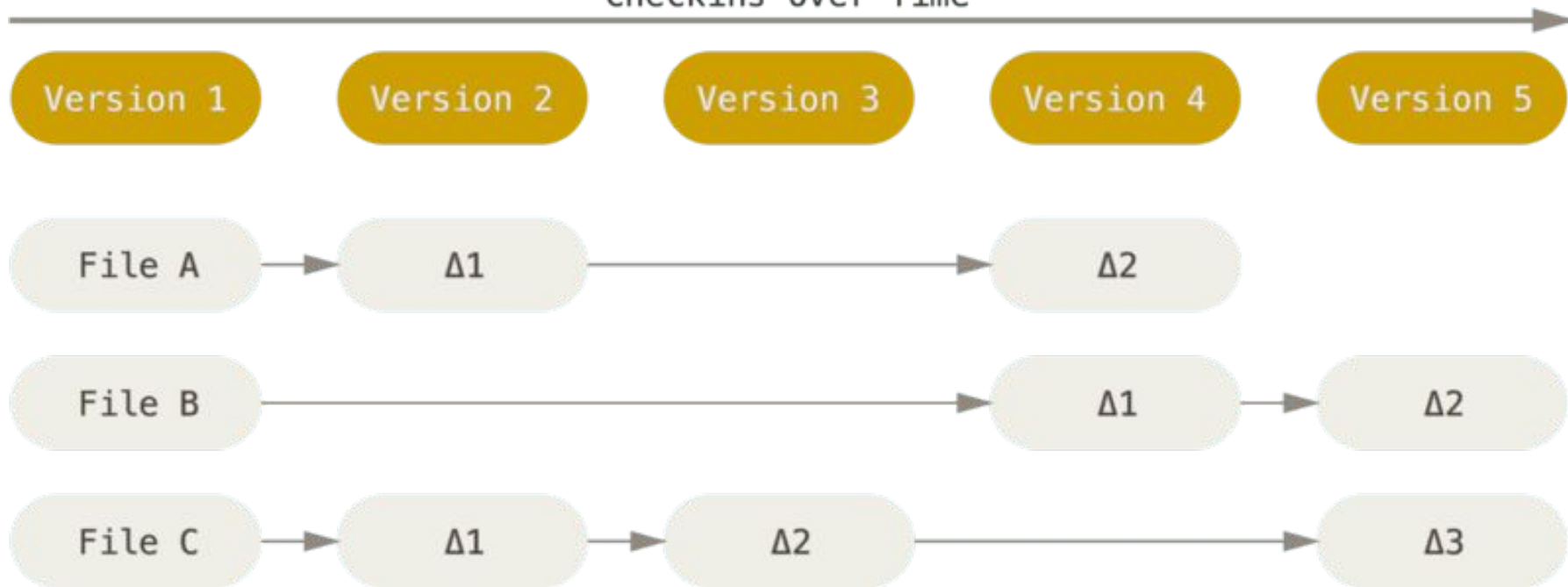
With Git every changes are tracked and recorded so that if you want to go to a previous state or version all the user has to do ask git to change the version and the project will change it to an older version and of course you can go back to the newer version. Other than changing versions, Git implements a system where it tracks all changes that are made to every file in the project and the programmer can save its current state so that later when changes are made to any of the project files, he/she can revert it back to the previous state that was saved if he/she does not like it.

How Does Git work?

Git doesn't think of or store its data this way. Instead, Git thinks of its data more like a set of snapshots of a miniature filesystem. Every time you commit, or save the state of your project in Git, it basically takes a picture of what all your files look like at that moment and stores a reference to that snapshot. To be efficient, if files have not changed, Git doesn't store the file again, just a link to the previous identical file it has already stored. Git thinks about its data more like a stream of snapshots.

How Does Git work?

Checkins Over Time

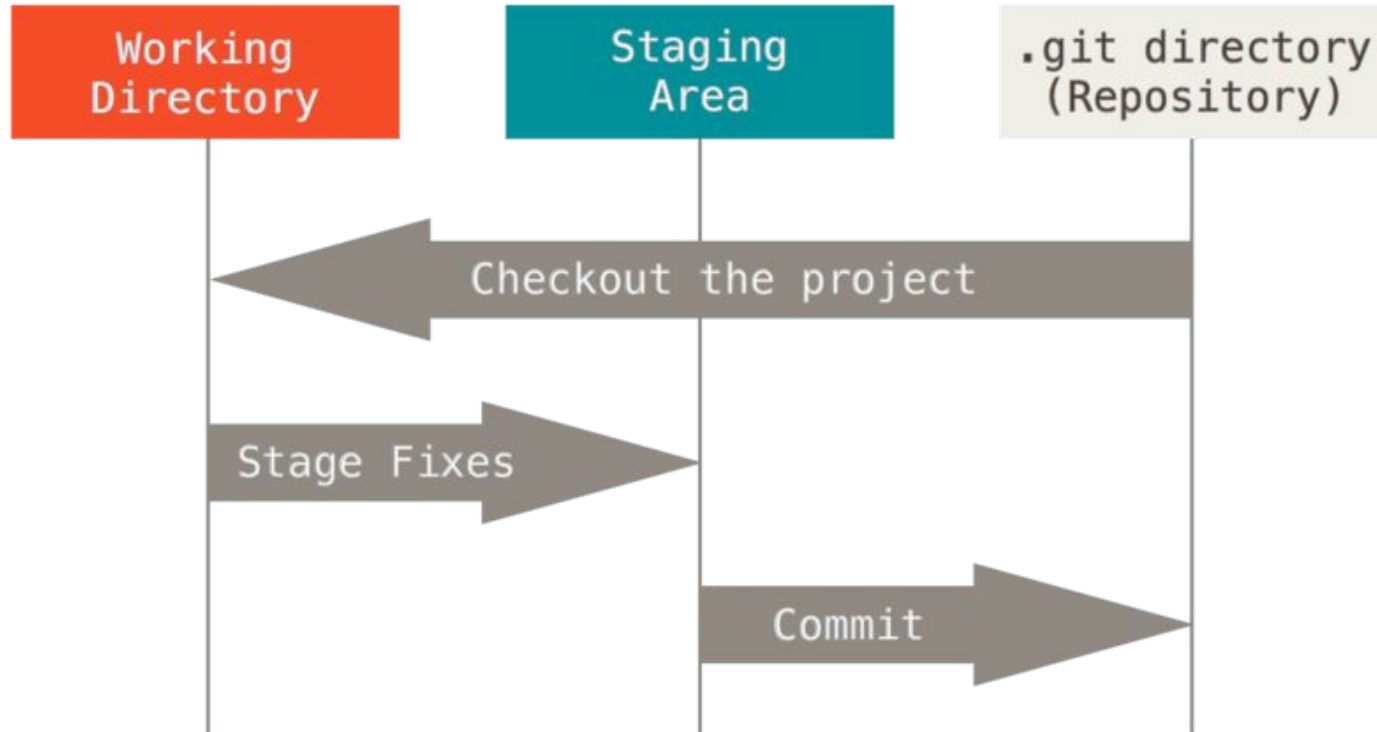


How Does Git work?

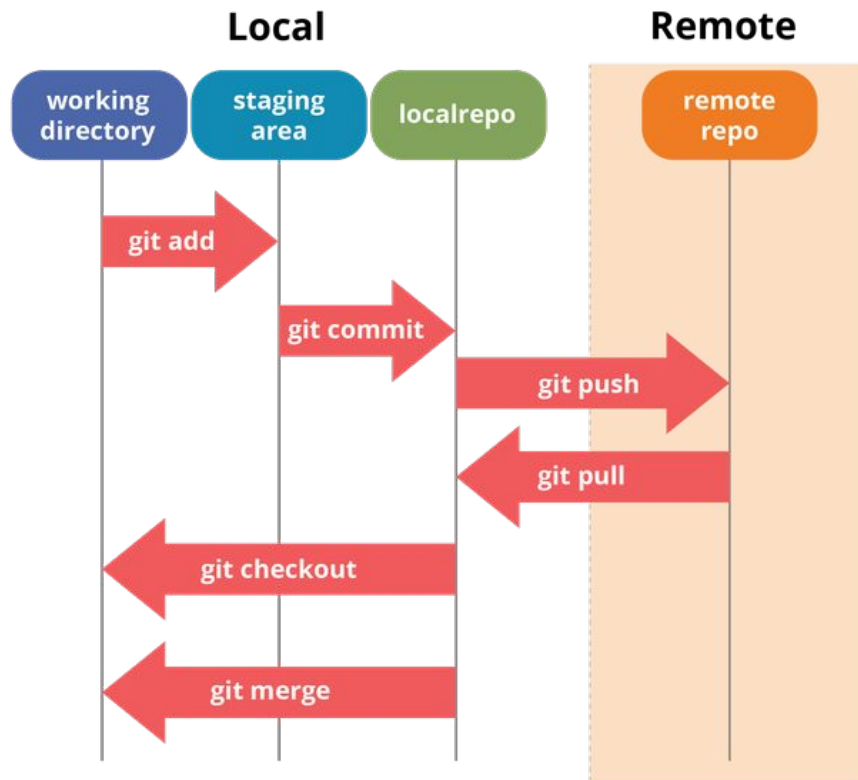
Now, pay attention. This is the main thing to remember about Git if you want the rest of your learning process to go smoothly. Git has three main states that your files can reside in: committed, modified, and staged. Committed means that the data is safely stored in your local database. Modified means that you have changed the file but have not committed it to your database yet. Staged means that you have marked a modified file in its current version to go into your next commit snapshot.

This leads us to the three main sections of a Git project: the Git directory, the working tree, and the staging area.

How Does Git work?



How Does Git work?



How to use GitHub?

Install Git:

- <https://git-scm.com/downloads>

Have some basic knowledge on how to use command line(windows)/terminal (Mac)

Have a Github Account

How to use Command Line/Terminal

cd: Change Directory (Directory = Folder)

Example: (Notice how we changed directory)

```
C:\Users\Lenovo>cd Documents
```

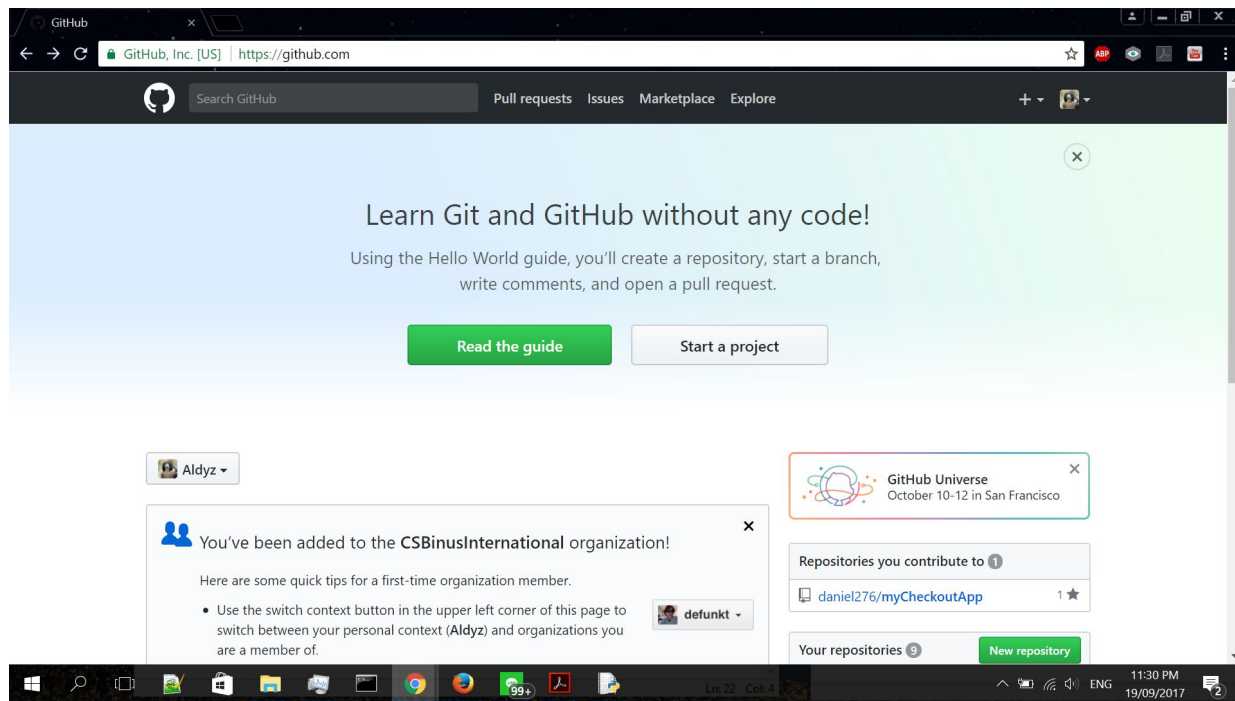
```
C:\Users\Lenovo\Documents>
```

dir(windows), ls(mac): This command list all files in the directory

mkdir [FolderName]: This command creates an empty folder with the name of the folder after the command, if you want a space in the name use double quotations, for example "A Project".

Pushing a Local Repository to a Remote Repository#1

Start a Project →



Pushing a Local Repository to a Remote Repository#2

Enter a repo name

And click create

repository.

Create a New Repository x

GitHub, Inc. [US] | https://github.com/new

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: Aldyz / Repository name: Dyz-Project ✓

Great repository names are short and memorable. Need inspiration? How about [stunning-octo-umbrella](#).

Description (optional)

☒ **Public**
Anyone can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** | Add a license: **None** ⓘ

Create repository

Pushing a Local Repository to a Remote Repository#3

Open Git CMD (Windows) or Terminal (Mac), Type and Enter These:

```
C:\Users\Lenovo>git config --global user.name [Github Username]
```

```
C:\Users\Lenovo>git config --global user.email [Used Email for Github]
```

Example:

```
C:\Users\Lenovo>git config --global user.name Aldyz
```

```
C:\Users\Lenovo>git config --global user.email supremealdi@gmail.com
```

NEVER FORGET THIS STEP, but you only need to do this once so next time you open Git, this step is not needed.

Pushing a Local Repository to a Remote Repository#4

Make a folder anywhere and in command line change the directory to the directory where the folder was created, For example:

```
C:\Users\Lenovo\Documents>mkdir GitHub_Project
```

```
C:\Users\Lenovo\Documents>cd GitHub_Project
```

```
C:\Users\Lenovo\Documents\GitHub_Project>
```

Pushing a Local Repository to a Remote Repository#5

Type the command below as this will initialize a local repository in the folder or basically the git system will start to actively work in the folder and start tracking files.

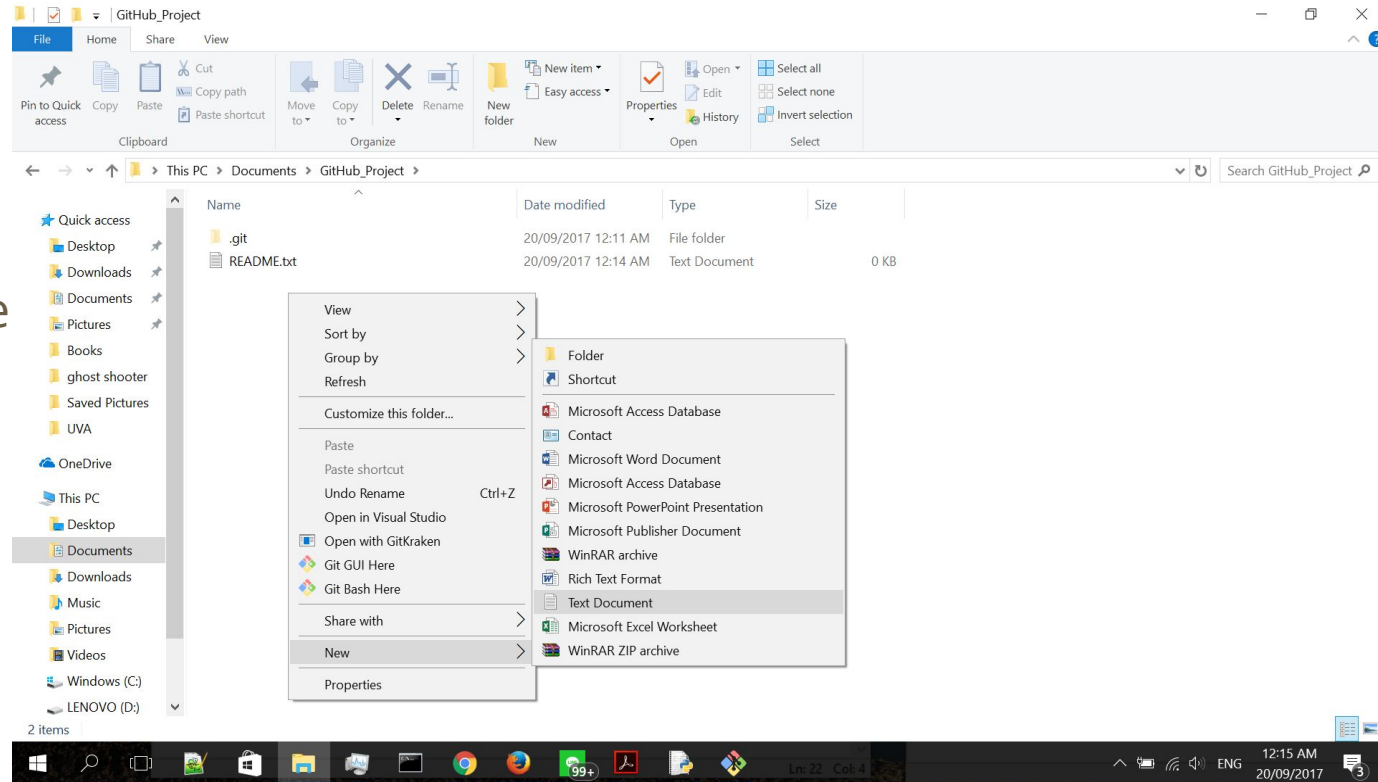
```
C:\Users\Lenovo\Documents\GitHub_Project>git init
```

The Command above should output:

```
Initialized empty Git repository in  
C:/Users/Lenovo/Documents/GitHub_Project/.git/
```


Pushing a Local Repository to a Remote Repository#6

Create a
README.txt file in
the folder, open the
file and insert any
random text.



Pushing a Local Repository to a Remote Repository#7

Go back to Git CMD or Terminal, then type the commands below:

```
C:\Users\Lenovo\Documents\GitHub_Project>git add .
```

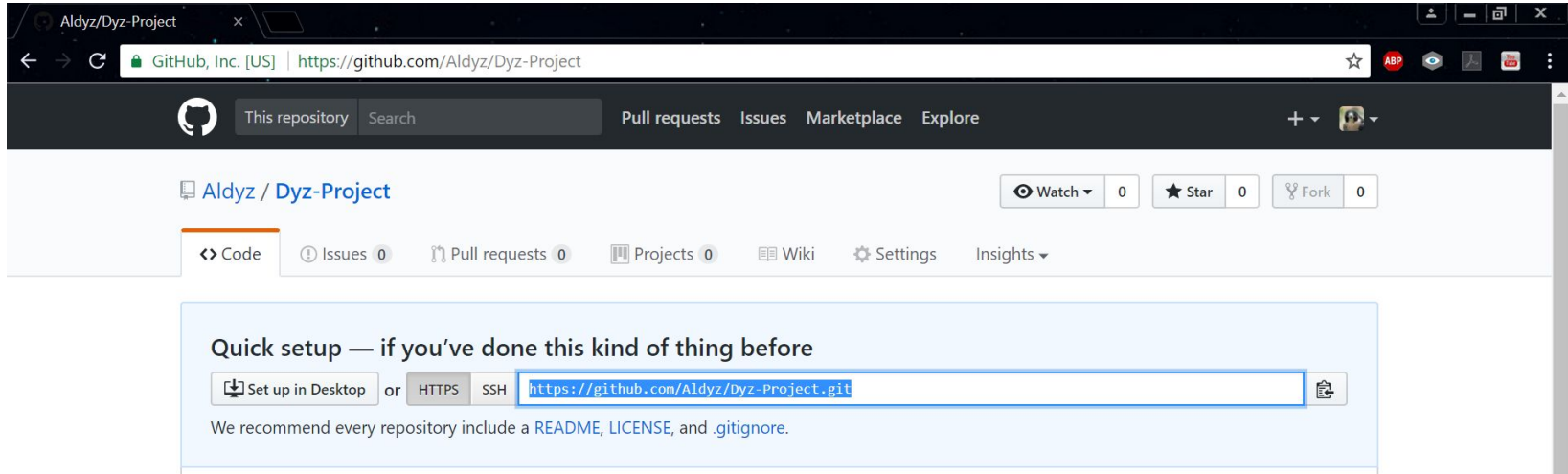
```
C:\Users\Lenovo\Documents\GitHub_Project>git commit -m "First Commit"
```

git add . : Tells Git to track all the files in the folder, make sure its ready for any commit.

git commit -m [Commit Message]: Tells Git to record/commit all changes in the directory and saves it as a state. Later the programmer can revert back to the saved state if he wants to cancel every update he has done in the folder.

Pushing a Local Repository to a Remote Repository#8

In order for the local repository or folder to know which remote/online repository it should push/upload its files to, open the Github repository you created and copy the https link.



Pushing a Local Repository to a Remote Repository#9

After Copying the link, open the Git CMD/Terminal, type and enter the following command:

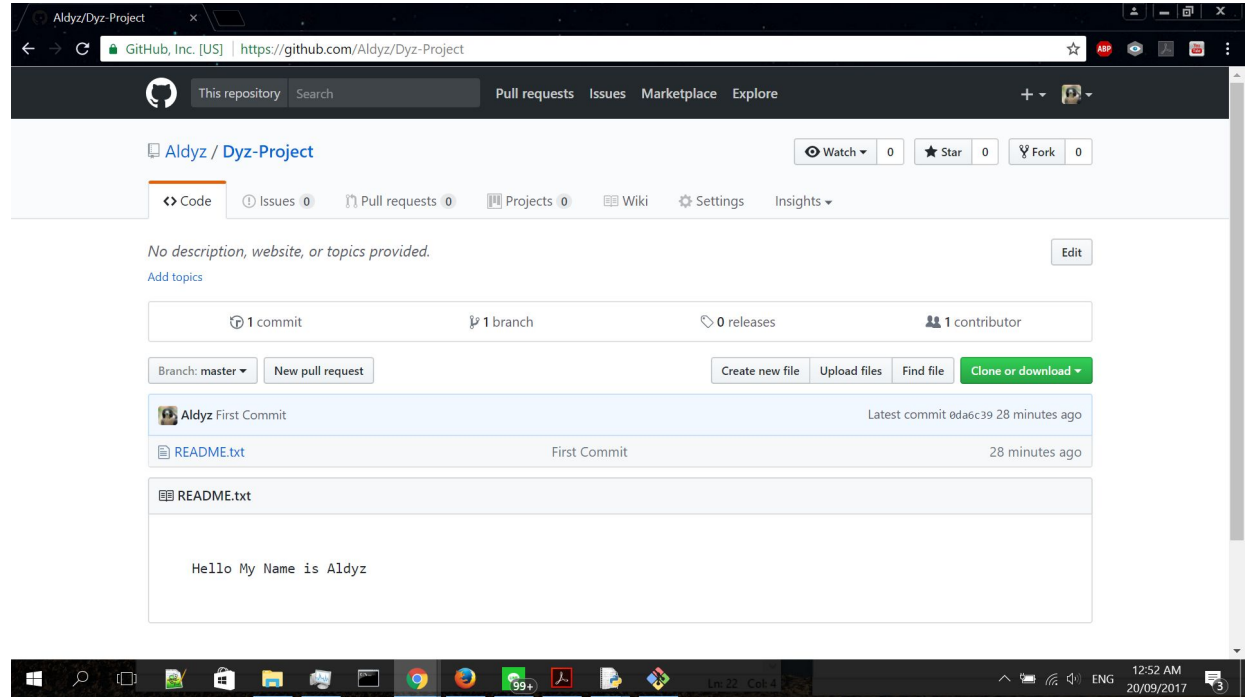
```
git remote add origin https://github.com/Aldyz/Dyz-Project.git
```

Notice how the link is inserted in the command as well. Now it's time to push the files in the local repository to the remote/online repository. Type and enter the following command.

```
git push -u origin master
```

Pushing a Local Repository to a Remote Repository#10

Notice how now the file
We have in our local
Repository has been
pushed/uploaded
To the remote/online
repository.



Pushing a Local Repository to a Remote Repository#11

Later if you add more files to the local repository or folder. You do not have to repeat the whole process you did earlier to push it again to the remote repository.

This PC > Documents > GitHub_Project >				
	Name	Date modified	Type	Size
Office T	.git	20/09/2017 12:40 AM	File folder	
System	Homework-1.py	20/09/2017 12:48 AM	Python File	1 KB
e	README.txt	20/09/2017 12:18 AM	Text Document	1 KB
Mather				
Structu				

Pushing a Local Repository to a Remote Repository#12

To push/upload the changes you have made to your project. Open your Git CMD/Terminal again. Type and Enter the following commands. **MAKE SURE YOU ARE IN THE LOCAL REPOSITORY's DIRECTORY**, if not change directory.

```
git add .
```

```
git commit -m "Second Commit"
```

```
git push
```

With only these these three commands, the local repository will upload its current state or files to the remote/online repository.

Pushing a Local Repository to a Remote Repository#13

The screenshot shows a web browser window displaying the GitHub repository page for 'Aldyz / Dyz-Project'. The browser's address bar shows the URL 'https://github.com/Aldyz/Dyz-Project'. The repository page includes a header with the repository name, a search bar, and navigation links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the header, there are buttons for 'Watch', 'Star', and 'Fork', each with a count of 0. The 'Code' tab is selected, showing a 'No description, website, or topics provided.' message and an 'Add topics' link. A summary bar indicates '2 commits', '1 branch', '0 releases', and '1 contributor'. Below this, there are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. The commit history shows two commits: 'Homework-1.py' (Second Commit, 5 minutes ago) and 'README.txt' (First Commit, 37 minutes ago). The 'README.txt' file content is displayed as 'Hello My Name is Aldyz'. The Windows taskbar at the bottom shows the time as 1:01 AM on 20/09/2017.

Aldyz / Dyz-Project

GitHub, Inc. [US] | https://github.com/Aldyz/Dyz-Project

This repository Search Pull requests Issues Marketplace Explore

Aldyz / Dyz-Project

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights

No description, website, or topics provided. Edit

Add topics

2 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

Aldyz Second Commit Latest commit d055af8 5 minutes ago

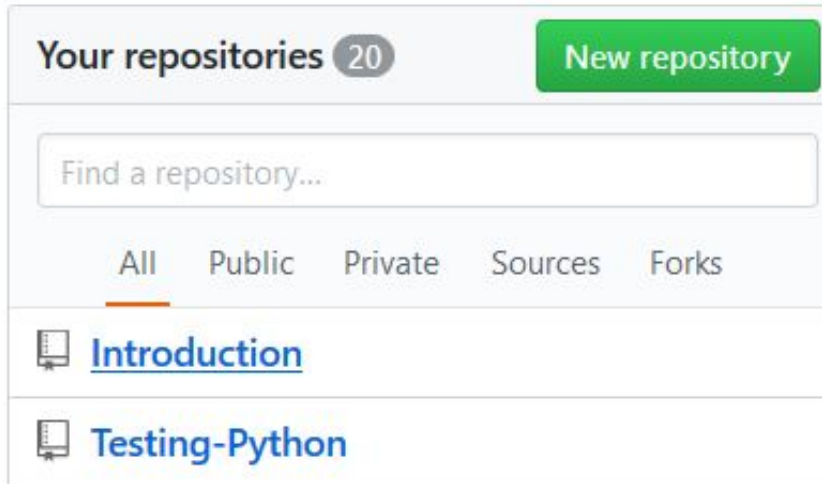
Homework-1.py	Second Commit	5 minutes ago
README.txt	First Commit	37 minutes ago

README.txt

```
Hello My Name is Aldyz
```


How to Clone A Repository that Already Exists on GitHub to Your Folder#1

First choose your GitHub repository



How to Clone A Repository that Already Exists on GitHub#2

Then, click the clone or download button.

The screenshot shows the GitHub interface for a repository named 'Introduction' by user 'veronika98stephanie'. The repository has 1 commit, 1 branch, 0 releases, and 1 contributor. The 'Clone or download' button is highlighted in green. A modal window is open, showing the 'Clone with HTTPS' option, which provides the URL 'https://github.com/veronika98stephanie/Intro' and a copy icon. Other options in the modal include 'Open in Desktop' and 'Download ZIP'. The repository's README file is visible in the background, titled 'Introduction'.

veronika98stephanie / Introduction

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights

No description, website, or topics provided. [Add topics](#) [Edit](#)

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request

Create new file Upload files Find file Clone or download

veronika98stephanie committed on GitHub Initial commit

README.md Initial commit

README.md

Introduction

Clone with HTTPS [Use SSH](#)

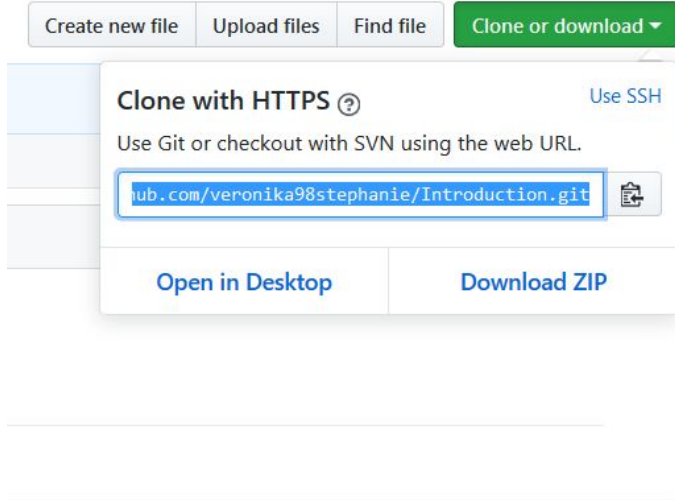
Use Git or checkout with SVN using the web URL.

<https://github.com/veronika98stephanie/Intro> [Copy](#)

[Open in Desktop](#) [Download ZIP](#)


How to Clone A Repository that Already Exists on GitHub#3

Make sure that you copy the repository's URL



How to Clone A Repository that Already Exists on GitHub#4

Now open your Git CMD or terminal from your Mac and go to the folder that you want to clone your repository to. Example:

 MINGW64:/e/Github

```
user@Veros-ASUS MINGW64 ~ (master)
$ cd E:

user@Veros-ASUS MINGW64 /e
$ cd Github

user@Veros-ASUS MINGW64 /e/Github (master)
$ |
```

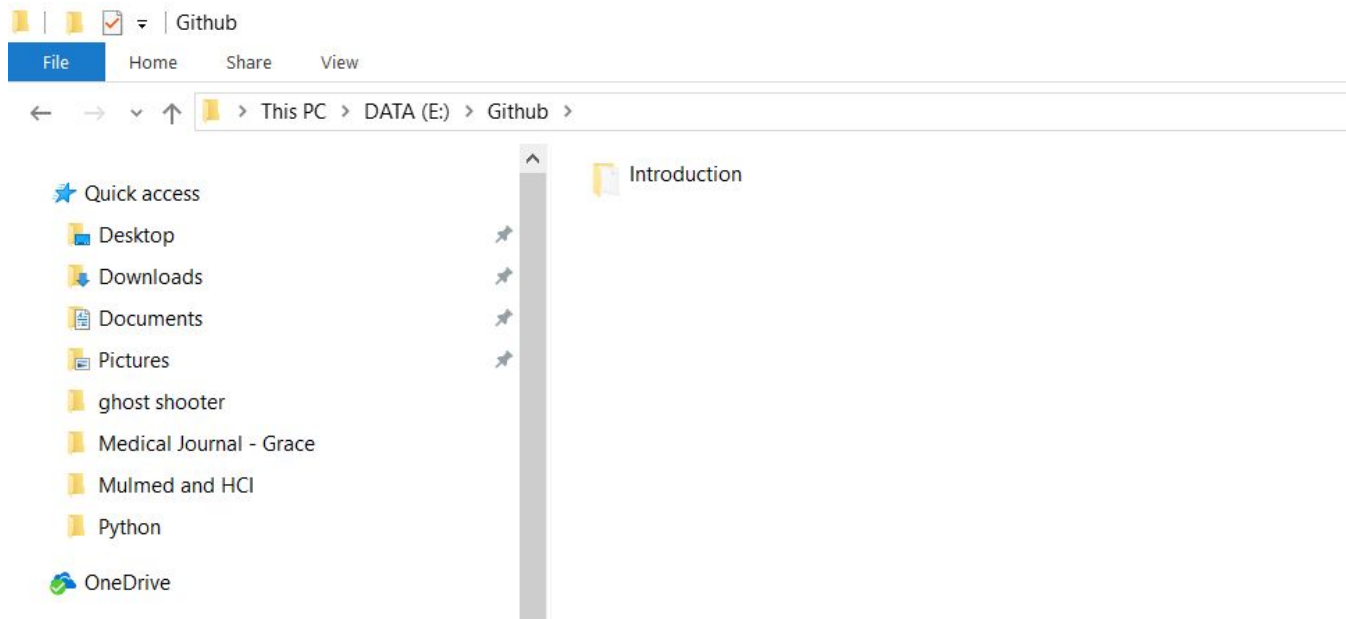
Clone the project by typing `git clone yourURL`. Your URL is the repository URL that you have copied before.

```
user@Veros-ASUS MINGW64 /e/Github (master)
$ git clone https://github.com/veronika98stephanie/Introduction.git
Cloning into 'Introduction'...
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.

user@Veros-ASUS MINGW64 /e/Github (master)
$ |
```

How to Clone A Repository that Already Exists on GitHub#5

Now that you are done, you can see a new folder named after your repository in your Github repository\folder that you have cloned



Editing and re-push data

For example you have a repository that consists of 3 files and you edit the content of one of the file. For this example the file that will be edited is the testing.py file

The screenshot displays a GitHub repository interface. At the top, it states 'No description, website, or topics provided.' with an 'Edit' button. Below this, repository statistics are shown: 2 commits, 1 branch, 0 releases, and 1 contributor. A navigation bar includes a dropdown for the 'master' branch, a 'New pull request' button, and buttons for 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. The commit history table shows three entries: the first push by 'veronika98stephanie' (latest commit b21e47f, 20 seconds ago), the initial commit of '.idea' (19 seconds ago), and the initial commit of 'README.md' (9 hours ago). Below the history, the 'README.md' file is open, showing the title 'Introduction'.


Commit	Author	Message	Time
b21e47f	veronika98stephanie	First Push	20 seconds ago
b21e47f	veronika98stephanie	Initial commit	19 seconds ago
b21e47f	veronika98stephanie	Initial commit	9 hours ago

Introduction




Editing and re-push data

At first tetsing.py contents is
only 1 line of python language
comment

Branch: master ▾ [Introduction](#) / [Testing.py](#) Find file Copy path

 **veronika98stephanie** First Push b21e47f 4 minutes ago

0 contributors

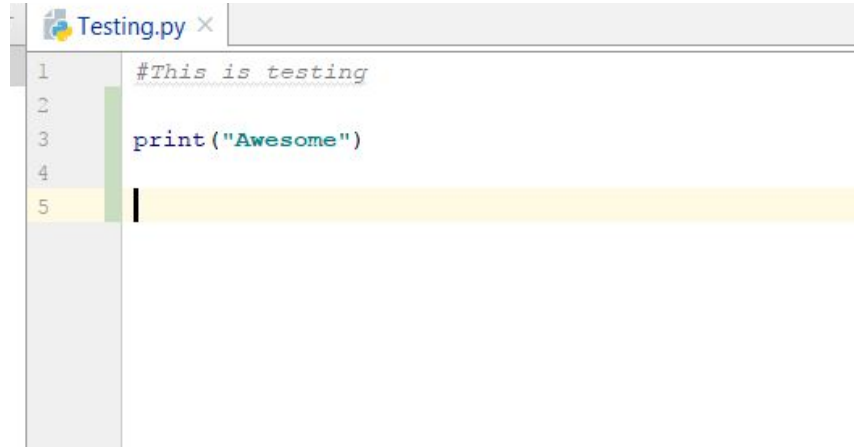
1 lines (1 sloc) | 16 Bytes Raw Blame History   

1 #This is testing



Editing and re-push data

Then, you made a change on your code from the cloned folder. In order to re-push your edited code from windows cmd or mac terminal, follow several steps bellow.



```
Testing.py x
1  #This is testing
2
3  print("Awesome")
4
5  |
```


Editing and re-push data

First, open your windows
cmd or mac terminal, then go
to your local repository
(folder that contains the file
that you have edited)

Then, input:

git add .

git commit -m "Your message"

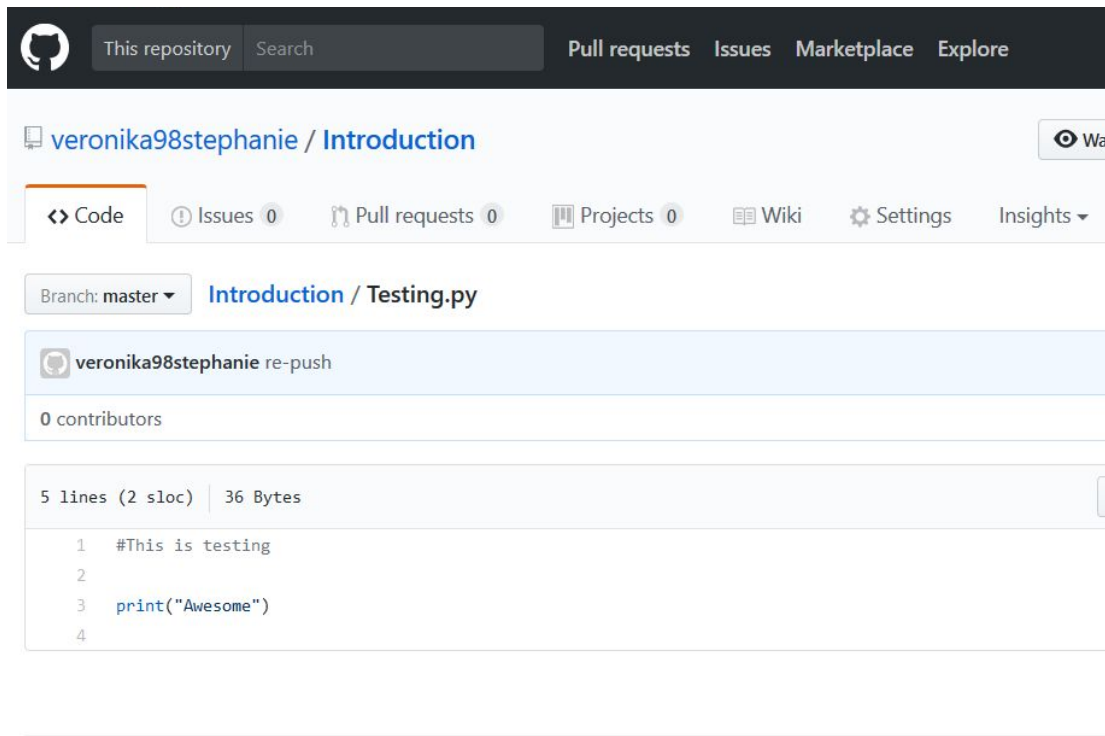
git push

With this you are done

```
user@Veros-ASUS MINGW64 /e/github/introduction (master)
$ git add .
warning: LF will be replaced by CRLF in .idea/workspace.xml.
The file will have its original line endings in your working directory.
user@Veros-ASUS MINGW64 /e/github/introduction (master)
$ git commit -m "re-push"
[master 716b808] re-push
2 files changed, 28 insertions(+), 21 deletions(-)
user@Veros-ASUS MINGW64 /e/github/introduction (master)
$ git push
Counting objects: 5, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 776 bytes | 0 bytes/s, done.
Total 5 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/veronika98stephanie/Introduction.git
```

Editing and re-push data

Now you can see the changes on your github



The screenshot displays the GitHub interface for the repository 'veronika98stephanie / Introduction'. The top navigation bar includes links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the repository name, there are tabs for 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Settings', and 'Insights'. The 'Code' tab is selected, showing the file 'Introduction / Testing.py' on the 'master' branch. A commit by 'veronika98stephanie' is shown as a 're-push'. The file statistics indicate it is 5 lines (2 sloc) and 36 Bytes. The code content is as follows:

```
1 #This is testing
2
3 print("Awesome")
4
```

Useful Resources:

- https://www.youtube.com/playlist?list=PL4cUxeGkcC9goXbgTDQ0n_4TBzOO0ocPR
- <https://git-scm.com/book/en/v2>
- <https://www.youtube.com/playlist?list=PL6gx4Cwl9DGDV6SnbINIVUd0o2xT4JbMu>
- <https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf>