Upload a file in 12 easy steps on Github, with git and Gnu / Linux!

Prerequisite: having installed git on your machine.
On a Debian or a derivative like Ubuntu:
>> sudo apt install git or >> sudo apt install git-all if you want all the options, if in doubt always prefer the first solution).
On a RedHat or a derivative like Fedora:
>> sudo dnf install git or >> sudo dnf install git-all if you want all the options, if in doubt always prefer the first solution).
On an Arch Linux or a derivative like Manjaro:
>> sudo pacman -S git or >> yay -S git (for the community enhanced version)

#Uppercase words will be words to personalize.
#The >> indicate a command to type (in blue) to their right.
#The red color indicates that this paragraph is more indicated for regular users, and that it is better to refer to the main, more complete documentation.

1) Create an account on the Github site
(skip this step if you already have an account).

2) Create a directory in which to put your git file (s)

```
>> mkdir MYDIRECTORY
```

(if necessary grant the necessary rights to this directory:

```
>> sudo chmod u + rwx MYDIRECTORY
>> sudo chown MYNICKNAME MYDIRECTORY
```

Here you are, your directory is created and functional!

3) move to the created directory

```
>> cd MYDIRECTORY
```

example of terminal forwarding: [MYDIRECTORY] $

4) if you want to add a README file:

```
echo "# MYDIRECTORY" >> README.md
```

5) place your software, your folder, or your file (s) in the MYDIRECTORY folder

(Warning: do not touch the hidden .git directory created by git init)

6) Tell Git that MYDIRECTORY must be considered as a folder to be shared on Github

```
>> [MYDIRECTORY] $ git init
```

Here we are, we finished the first part, we managed to put the folder we want to share on Github on a dedicated space (and we made it clear in passing to Git that it is indeed this folder in question), that we will now have to send to Github. (Take the opportunity to add: git add README.md or git add LISEZ-MOI.md)

7) Connect to the Github site, then click on the (+) icon at the top right of the home page then select the New Repository option (which means "Create a new repository")

On the page that appears, in the “Repository name” field, enter a name for your repository.
Warning: you must create the same name as the one on your computer, so that the two folders are then merged. So here MYDIRECTORY will have the same name as MYREPOSITORY…

Click on "Create Repository", ie: "Create a repository".
(You can also describe what your deposit will contain in the small insert provided for this purpose if you wish).

8) Now we will put our repository online:

In front of "HTTPS", we have a line that looks like this:
https://github.com/JEANCLAUDE/MONDEPOT.git
We therefore have the address of our online repository on github. So now we just need to add: git remote add origin, to make the machine understand where our online repository should be.
remote = move
add = add
Origin = first deposit, original deposit
Which gives :

```
>> git remote add origin https://github.com/JEANCLAUDE/MONDEPOT.git
```

9) We now tell git to accept to take all the files we are going to send in

```
>> git add --all
```

(be careful there are two dashes before all)

10) We now use the commit command to make a "snapshot" of the uploaded package with all the modifications made at that moment, so all the modifications made before this operation will be saved:

```
>> git commit -m "first commit"
```

The git commit command without options, opens a text editor and asks to write a commit message, to avoid opening this editor, for example when using a virtual machine or a minimal online machine of commands for example, it would be better to use the -m option.
the -m option forgoes the text editor prompt in favor of a pop-up message. You can choose the message you want.
The -a option includes only the selected files (those that were added with the git add command at some point in history).
We can add the two options:

```
>> git commit -am MESSAGE
```

This combination immediately creates a commit of all staged changes and inserts a contextual commit message.
We can also use:

```
>> git commit --amend
```

Instead of creating a new commit, the last commit will be modified and the latest changes will be added to the last commit.
(Warning: this command opens a text editor to modify the commit in question).

We must now create the “main” branch:

```
>> git branch -M main
```

11) It now remains to "push" the Directory into the Original folder, which will have for first chapter the name: "main" which replaced the old name: "Master"

Let's have this mnemonic method: we push on your origin and it gives birth with a "hand". (And each time you want to add an improvement to the file, you will have to "push" again to "give birth" to a new tree folder).

```
>> git push -u origin main
```

Here we are, the hard part is done, we published our file on Github with everything we need. The teleprompter asks you to identify yourself (Github account, with the password).
It now remains to

12) Create a branch to create a page on our repository, on the site:

click on Create branch: NAMEOFYOURBRANCH
If you want to download a specific branch on your pc, your can do it with :

```
git clone –branch NAMEOFTHEBRANCH URLONGETDEB.git
```

Here we are! At this point you normally have succeeded in putting your deposit online. I

##

NB: if you ever want to add things:

```
git add --all
git commit -m 'MyNewCOMMIT'
git push
```

NB2: if you ever want to add things from another pc than the one you usually use: create a Folder identical to MYREPERTOIRE of the first pc, then make a

```
git init
```

put the modified files in MYDIRECTORY of the new pc

```
git add --all

git remote add URLOFYOURGITHUBFILE.git

git commit -m "YOURCOMMITNAME"

git push –set-upstream URLOFYOURGITHUBFILEURL.git NEWNAME
```

(other_than_hand)
(by example : git push –set-upstream URLOFYOURGITHUBFILEURL.git Master)