Déposer un fichier en 12 étapes facilement sur Github, avec git et Gnu/Linux!

Prérequis : avoir installé git sur sa machine.
Sur une Debian ou une dérivée comme Ubuntu :
>>sudo apt install git ou >>sudo apt install git-all si on veut toutes les options, dans le doute, toujours préférer la première solution).
Sur une RedHat ou une dérivée comme Fedora :
>>sudo apt install git ou >>sudo apt install git-all si on veut toutes les options, dans le doute, toujours préférer la première solution).
Sur une Arch Linux ou une dérivée comme Manjaro :
>>sudo pacman -S git ou >>yay -S git (pour la version améliorée par la communauté)

#Les mots en majuscules seront des mots à personnaliser.
#Les >> indiquent une commande à taper (en bleu) à leur droite.
#La couleur rouge indique que ce paragraphe concerne plutôt des utilisateurs habitués, et qu'il vaut mieux se référer à la documentation principale, plus complète.

1) Créer un compte sur le site Github

(sautez cette étape si vous avez déjà un compte).

2) Créer un répertoire dans lequel placer son ou ses fichiers git

>> mkdir MONREPERTOIRE
(au besoin accorder les droits nécessaires à ce répertoire :
>> sudo chmod u+rwx MONREPERTOIRE
>>sudo chown MONPSEUDO MONREPERTOIRE)
Voilà, votre répertoire est créé et fonctionnel!

3) se déplacer dans le répertoire créé

>> cd MONREPERTOIRE
exemple de renvoi de terminal : [MONREPERTOIRE]$

4) si on veut rajouter un fichier README ou Lisez-moi :

echo "# MONREPERTOIRE" >> README.md ou
echo "# MONREPERTOIRE" >> LISEZ-MOI.md

5) placer votre logiciel ou votre dossier, votre ou vos fichiers dans le Dossier MONREPERTOIRE

(Attention : ne pas toucher au répertoire caché .git créé par git init)

6) Indiquer à Git que MONREPERTOIRE doit être considéré comme un dossier à partager sur Github

>> [MONREPERTOIRE]$ git init

Voilà, on a terminé la première partie, on a réussi à mettre le dossier qu'on veut partager sur Github sur un espace dédié (et on a fait comprendre au passage à Git que c'est bien ce dossier dont il est question), qu'il va falloir maintenant envoyer sur Github.
(En profiter pour rajouter : git add README.md ou git add LISEZ-MOI.md)

7) Se connecter sur le site Github, puis cliquer sur l'icône (+) en haut à droite de la page d'accueil puis sélectionner l'option New Repository (qui signifie « Créer un nouveau dépôt »)

Sur la page qui s'affiche, dans le champ « Repository name », entrez un nom pour votre dépôt.
Attention : il faut créer le même nom que celui qui est sur votre ordinateur, pour que les deux dossiers soient fusionnés ensuite. Ainsi ici MONREPERTOIRE aura le même nom que MONDEPOT...

Cliquez sur "Create Repository", c'est à dire : "Créer un dépôt".
(Vous pouvez aussi décrire ce que contiendra votre dépôt dans le petit encart prévu à cet effet si vous en avez envie).

8) Maintenant on va mettre en ligne notre dépôt :

En face de "HTTPS" , on a une ligne qui ressemble à ça :
https://github.com/JEANCLAUDE/MONDEPOT/.git
On a donc l'adresse de notre dépôt en ligne sur github. Il suffit donc maintenant de lui adjoindre : git remote add origin, pour faire comprendre à la machine où doit se situer notre dépot en ligne .
remote = déplacer
add = ajouter
Origin = premier dépôt, dépôt original
Ce qui donne :
>> git remote add origin https://github.com/JEANCLAUDE/MONDEPOT/.git

9) On indique maintenant à git d'accepter de prendre tous les fichiers qu'on va lui envoyer

>> git add --all
(attention il y a deux traits avant all)

10) On utilise maintenant la commande commit pour réaliser une « photo instantannée » du paquet mis en ligne avec toutes les modifications faites à cet instant, ainsi toutes les modifications faites avant cette opération seront enregistrées :

>> git commit -m « first commit »
La commande git commit sans option, ouvre un éditeur de texte et demande à écrire un message de commit, pour éviter d'ouvrir cet éditeur, par exemple en cas d'utilisation d'une machine virtuelle ou d'une machine minimale en ligne de commandes par exemple, il vaudra mieux utiliser l'option -m.
l'option `-m` renonce à l'invite d'éditeur de texte au profit d'un message contextuel. On peut choisir le message qu'on veut.
L'option -a inclut uniquement les fichiers choisis (ceux qui ont été ajoutés grâce à la commande git add à un moment de l'historique).
On peut ajouter les deux options :
>> git commit -am MESSAGE
`Cette combinaison crée immédiatement un commit de tous les changements stagés et insère un message de commit contextuel.`
On pourra utiliser aussi :
>>git commit --amend
Au lieu de créer un nouveau commit, le dernier commit sera modifié et les derniers changements seront ajoutés au dernier commit.
(Attention : cette commande ouvre un éditeur de texte pour modifier le commit en question).

Il faut maintenant créer la branche « main » :
>> git branch -M main

11) Il reste maintenant à "pousser" le Repertoire dans le dossier Originel, qui aura pour premier chapitre le nom : "main" qui a remplacé l'ancien nom : "Master"

Ayons cette méthode mnémotechnique : on pousse sur ton origine et ça accouche d'un « main ». (Et à chaque fois qu'on voudra ajouter une

amélioration au dossier, il faudra "pousser" à nouveau pour "accoucher" d'un nouveau dossier en arborescence).

>>git push -u origin main

Voilà, le plus dur est fait, on a publié notre dossier sur Github avec tout ce qu'il faut. Le prompteur vous demande de vous identifier (compte de Github, avec le mot de passe).
Il reste maintenant à

12) Créer une branche pour créer une page sur notre dépôt, sur le site :

cliquer sur Create branch : gh-pages (oui le nom de notre branche s'appelle gh-pages, c'est laid, mais on s'en fout, si si).

NB : si jamais vous voulez ajouter des choses :
git add --all
git commit -m 'NOMDEMONNOUVEAUCOMMIT'
git push

Voilà, à ce stade vous avez normalement réussi à mettre votre dépôt en ligne. Si vous avez des erreurs, n'hésitez pas à en parler, et à mentionner des correctifs pour faire évoluer la documentation.