Etat initial : le répertoire "projet1" est déjà présent sur le disque de la machine (turing) où sera hébergé le dépôt d'origine.

MACHINE SOURCE (ORIGINE)
On initialise un dépôt git dans ce répertoire existant (mettre l'option --bare si on veut pouvoir faire des push dessus ; ici pour l'exemple on ne le met pas) :
> git init projet1
Remarquer qu'un nouveau répertoire caché nommé ".git" est apparu dans projet1

Voir ce qui est actuellement suivi par git :
> cd projet1
> git status

Rien n'est suivi pour le moment.
Indiquer qu'on veut que les deux fichiers présents dans le répertoire projet1 doivent être suivis :
> git add p1_f1
> git add p1_f2

Voir ce qui est actuellement suivi par git :
> git status

Ici les deux fichiers sont indexés, c'est-à-dire marqués comme à suivre par git, mais pas encore sauvés
Il faut maintenant valider cet ajout et le sauver dans l'historique :
> git commit -m "ajout des deux premiers fichiers"
(Bien prendre soin de mettre un commentaire explicite)

Vérifier le statut courant :
> git status
Tous les fichiers suivis sont sauvés.

MACHINE DISTANTE
Sur une machine distante, on va récupérer une copie du dépôt :
> git clone essert@turing.u-strasbg.fr:/adhome/e/es/essert/Demos/outils_de_dev/test_git/projet1
On voit qu'un nouveau répertoire projet1 est créé, les 2 fichiers sont copiés dedans, et un répertoire caché ".git" est créé aussi dans projet1.

MACHINE SOURCE (ORIGINE)
Sur la machine d'origine (turing), faire une modif dans le fichier p1_f1.
Montrer que git a remarqué que ce fichier a été modifié :
> git status
Indiquer que l'on veut que ces modifications soient sauvées dans l'historique
> git add p1_f1
Montrer que git sait maintenant qu'au prochain commit il faudra envoyer ce fichier (indexé) :
> git status

Puis valider l'envoi de ces modifs
> git commit -m "petite modif dans le fichier p1_f1"
Refaire une autre petite modif sur le fichier et montrer que les deux commandes précédentes peuvent être faites en une seule fois :
> git commit -a -m "petite modif"

MACHINE DISTANTE
Maintenant aller sur la machine distante, et récupérer les informations sur les modifications qui ont été faites sur le dépôt sur la machine d'origine :
> git fetch

On est passé de cet état :
```
---rev0
    ^
  origin/master
    ^
   master
```

A cet état :
```
---rev0------rev1
             ^
           origin/master
    ^
   master
```

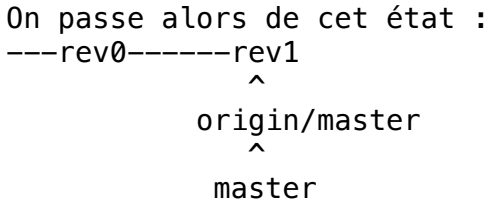
On fusionne alors dans le dépôt de la machine distante les modifications venant de la machine d'origine :
> git pull

Etant donné que sur la machine distante on n'avait pas encore fait de modif, cela revient juste à avancer le pointeur master (fast-forward).
On passe alors de cet état :
```
---rev0------rev1
             ^
           origin/master
    ^
   master
```

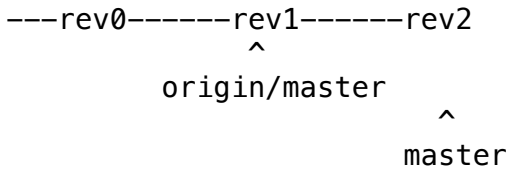
A cet état :
```
---rev0------rev1
             ^
           origin/master
              ^
            master
```

Maintenant modifions un fichier sur la machine distante, puis sauvons les modifs dans l'historique (du dépôt de la machine distante) :
> git commit -a -m "modifs depuis la machine distante"
> git status

On passe alors de cet état :




A cet état :




On veut maintenant envoyer ces modifs sur la machine source, on essaie :
> git push
Mais git refuse car en gros on n'a pas les droits.
Il faut alors faire plutôt un pull depuis la machine source, et là ça fonctionne.
En général, on fonctionne plutôt par pull requests. Lorsque l'on veut envoyer sa nouvelle version à la terre entière, plutôt que de l'imposer (push) on demande si les autres sont d'accord pour la récupérer (pull request).

Sur GitHub, le dépôt est en mode "bare" qui autorise les push, mais il vaut mieux fonctionner par pull request.

GITHUB
=> Demo sur GitHub avec le repository : https://github.com/essert/TechDevDemo.git
Se créer un compte sur GitHub.
Y cloner ce repository, puis le cloner chez soi.
Faire des modifs, puis des commit/push/pull.