

Travaux Pratiques : Enoncés

330-TP02_Git_Create

Objectif : Création d'un dépôt local, utiliser les commandes status, add, commit, tag, rm, mv

Techniques abordées :

- Git en ligne de commande

Dans ce TP, nous travaillerons sur un dépôt local.
Ce dépôt contiendra quelques glossaires sur le html et le css.

1. Création d'un nouveau dépôt local sur votre ordinateur.

Le dépôt local s'appellera « glossaires » et sera placé à la racine de votre bureau.

Initialiser ce dépôt :

Soit

- en créant le répertoire par l'explorateur de Windows
- en vous plaçant dedans
- en tenant la touche majuscule faire un clic droit : ouvrir une fenêtre de commande ici
- en ligne de commande, entrer git init

Soit

- ouvrez le bureau par l'explorateur
- en tenant la touche majuscule faire un clic droit : ouvrir une fenêtre de commande ici
- en ligne de commande, entrer git init glossaires
- en ligne de commande, entre cd glossaires

Vérifier le statut de votre dépôt avec la commande « git status »

```
C:\Users\.....\Desktop\glossaires>git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

2. Créer un fichier texte html.txt

Créer un simple fichier texte avec quelques définitions sur le html

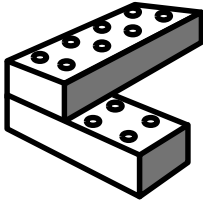
Exemple :

DOCTYPE

sert à spécifier la version du Html

HEAD

partie de la page Html où seront mises les instructions pour le navigateur



Travaux Pratiques : Enoncés

330-TP02_Git_Create

- a. Vérifier le statut de votre dépôt : vos modifications (ajout du fichier `html.txt`) n'est pas un fichier suivi (untracked)

```
C:\Users\...\Desktop\glossaires>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
      html.txt

nothing added to commit but untracked files present (use "git add" to track)
```

- b. Préparer le fichier `html.txt` à être suivi par la commande « `git add html.txt` »
- c. Vérifier le statut de votre dépôt : maintenant git vous informe qu'il y a un nouveau (new) fichier prêt à être commité

```
C:\Users\...\Desktop\glossaires>git status
On branch master

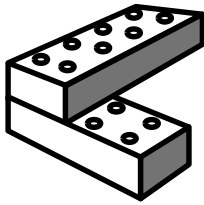
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
      new file:   html.txt
```

- d. Afficher la différence entre la zone d'index (staged area) et le dernier historique de la zone du dépôt (repository) qui est vide car on n'a pas encore fait de commit, par la commande : `git diff --cached` ou `git diff --staged`

```
C:\Users\...\Desktop\glossaires>git diff --staged
diff --git a/html.txt b/html.txt
new file mode 100644
index 0000000..174dabd
--- /dev/null
+++ b/html.txt
@@ -0,0 +1,5 @@
+DOCTYPE
+    sert <C3><A0> sp<C3><A9> cifier la version du Html
+
+HEAD
+    partie de la page Html o<C3><B9> seront mises les instructions pour le n
+avigateur
```

git vous montre ce qui a été ajouté dans le fichier (+) ou retiré (-) : ici 5 ligne ont été ajoutées.



Travaux Pratiques : Enoncés 330-TP02_Git_Create

- e. Commiter vos modifications par la commande :

Git commit -m "ajout du fichier de glossaire sur le html : html.txt"

```
C:\Users\...\Desktop\glossaires>git commit -m "ajout du fichier de glossaire
sur le html : html.txt"
[master (root-commit) d1af56c] ajout du fichier de glossaire sur le html : html.
txt
1 file changed, 5 insertions(+)
create mode 100644 html.txt
```

- f. Vérifier le statut de votre dépôt : maintenant git vous informe que tout est
clean

```
C:\Users\...\Desktop\glossaires>git status
On branch master
nothing to commit, working tree clean
```

- g. Essayer à présent la commande git log pour afficher la liste des
changements effectués dans le dépôt :
combien y en a-t-il ?
quel est le numéro (le sha1) du dernier commit ?

```
C:\Users\...\Desktop\glossaires>git log
commit d1af56ce817e630c26720fe60b8a5f6d28f0bb90 (HEAD -> master)
Author: didier Bonneau <didier.bonneau.vlb@gmail.com>
Date: Tue Sep 22 22:54:49 2020 +0200

    ajout du fichier de glossaire sur le html : html.txt
```

3. Ajout et modification de fichiers

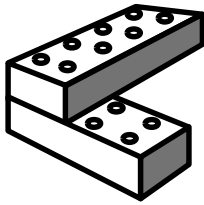
Ajouter un fichier « css.txt » avec quelques définitions et/ou modifier le fichier
« html.txt » en lui ajoutant quelques lignes

- Afficher la différence entre la zone de travail (working) et la zone d'index (staged) par la commande : « git diff » avant le « git add »
- Afficher la différence entre la zone d'index et la zone de repo par la commande : « git diff -cached » après avoir fait un « git add »

4. Revenir en arrière

Vous voulez changer d'avis entre les différents états.

- Faites une modification d'un ou plusieurs glossaires, ajoutez-la à l'index avec « git add »
- Vérifiez cet ajout avec « git status », mais ne la commitiez pas.



Travaux Pratiques : Enoncés

330-TP02_Git_Create

```
C:\Users\...\Desktop\glossaires>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   css.txt
```

- c. Exécutez « git restore --staged » sur le nom de fichier (ou les noms de fichiers) que vous avez préparés pour le commit.
Ancienne commande « git rm --cached » nom du fichier
- d. vérifiez avec git status le résultat.

Le ou les fichiers ne sont plus suivis et ne sont plus dans la « staged area » (index).

```
C:\Users\...\Desktop\glossaires>git restore --staged css.txt

C:\Users\...\Desktop\glossaires>git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        css.txt

nothing added to commit but untracked files present (use "git add" to track)
```

- e. Exécutez maintenant un « git restore » sur le nom de fichier :
Git va remettre le fichier de la zone de travail tel qu'il est dans le dépôt du dernier commit.
Vos modifications sont perdues.
Vérifiez en le rouvrant avec votre éditeur.