

Exercices - Git

Pour ces exercices, en plus de taper dans le terminal vos commandes, veuillez à me rendre cet exercice résolu sous forme numérique éditable: bloc note, document word, etc. Mais pas de pdf...

Donc à chaque étape, donnez les commandes utilisées et le résultat.

Ne faites pas toutes les commandes sans les noter ainsi que leur résultat sinon vous perdrez des informations si vous le faites après.

Exercice n°1 - Creation / indexation / validation

1.1 - Création d'un dépôt

1. Créez un répertoire nommé premierDepot
2. Fais-en un dépôt.
3. Faites un git status pour vérifier que le répertoire est bien devenu un dépôt.

1.2 - Modification (Ajouter de fichiers)

Pour éviter des alertes de sécurité de Chrome, j'ai nommé les fichiers avec l'extension .txt. Nous les renommerons plus tard avec la bonne extension.

1. Dans le répertoire que vous venez de créer, copiez les 3 fichiers suivants:

- [index.txt](#)
- [genius.txt](#)
- [display.txt](#)

2. Faites un git status. (Que vous dit/signale git ?)

1.3 - Indexation

1. Indexez le fichier index.txt avec git.
2. Indexez les autres fichiers avec une seule commande git.
3. Vérifiez avec la bonne commande git si les 2 indexations se sont bien passées.

1.4 - Validation/Commit

Validez les fichiers présents dans la zone d'index avec le message suivant: "Commit initial" (C'est souvent le premier message que l'on donne pour le tout premier commit)

1.5 - Modifications (Renommer des fichiers)

Ici la modification va porter sur le fait que l'on va renommer nos fichiers.

Pour rappel, voilà comment renommer un fichier en ligne de commandes:

A. Windows: ren fichier1 fichier2 (va renommer le fichier1 avec le nom fichier2) B. Mac Os/Linux: mv fichier1 fichier2 (va renommer le fichier1 avec le nom fichier2)

Renommez les fichiers de la manière suivante:

1. index.txt devient index.html
2. genius.txt devient index.js
3. display.txt devient display.js
4. Evidemment vérifiez avec dir (windows) ou ls (mac/linux) s'ils sont bien renommés.

1.6 - Indexation

1. Faites un git status
2. Indexez les fichiers modifiés.
3. Refaites un git status (vous voyez ce qui a changé depuis le point 1 ? Quoi ?)

1.7 - Validation/Commit

1. Faites un git status
2. Validez/Commitez vos fichiers en ajoutant comme message "Commit initial"
3. Faits un git status (vous voyez ce qui a changé depuis le point 1 ? Quoi ?)

1.8 - Affichage de l'historique des commits

Affichez l'historique des commits à l'aide de la commande: ???

1.9 - Modification (de code)

Explication des différents fichiers: - genius.html: il sert à afficher une page qui effectue des calculs. - genius.js: c'est le fichier qui contient nos fonctions javascript de mathématique: add et subtract. - display.js: il y a une fonction display qui permet d'afficher le résultat d'un calcul à l'écran. Vous ne devez pas y toucher !

L'exercice:

1. Editez le fichier genius.js
2. Ajoutez la fonction multiply qui retourne (return) la multiplication des deux nombres donnés en paramètre dans la signature de la fonction multiply.
3. Editez le fichier genius.html
4. Appelez la fonction display pour qu'elle affiche la multiplication de a par b ($a * b$).
5. Inspirez-vous des deux lignes précédentes qui font la soustraction et l'addition de a et b.
6. Faites un git status (vous voyez ce qui a changé depuis le point 1 ? Quoi ?)

1.10 - Indexation

1. Faites un git status
2. Indexez les fichiers modifiés.
3. Refaites un git status (vous voyez ce qui a changé depuis le point 1 ? Quoi ?)

1.11 - Validation

1. Faites un git status
2. Validez/Commitez vos fichiers en ajoutant comme message "Ajout de la fonction multiply et adaptation du fichier genius.html"
3. Faites un git status (vous voyez ce qui a changé depuis le point 1 ? Quoi ?)