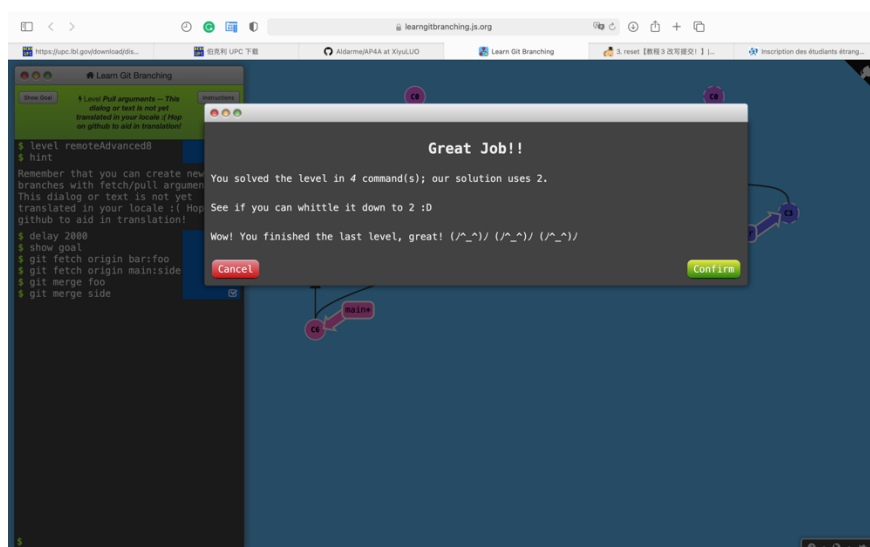
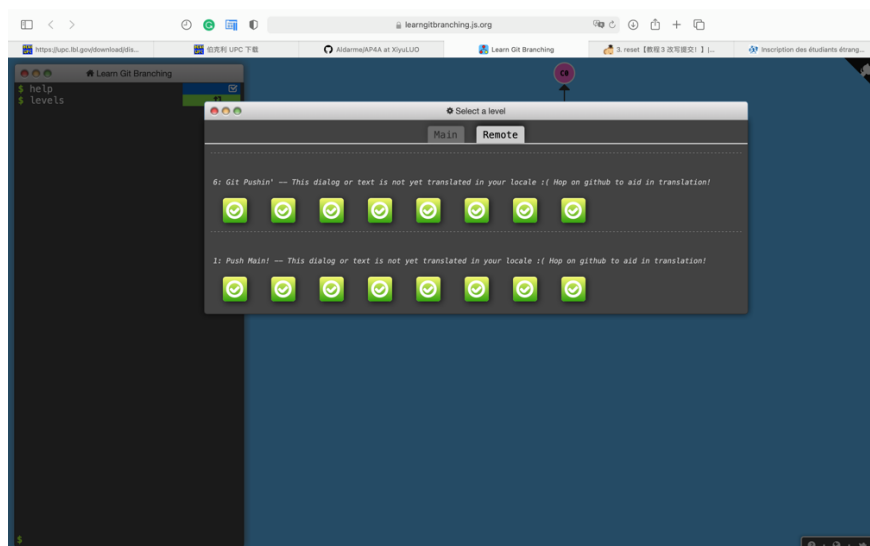
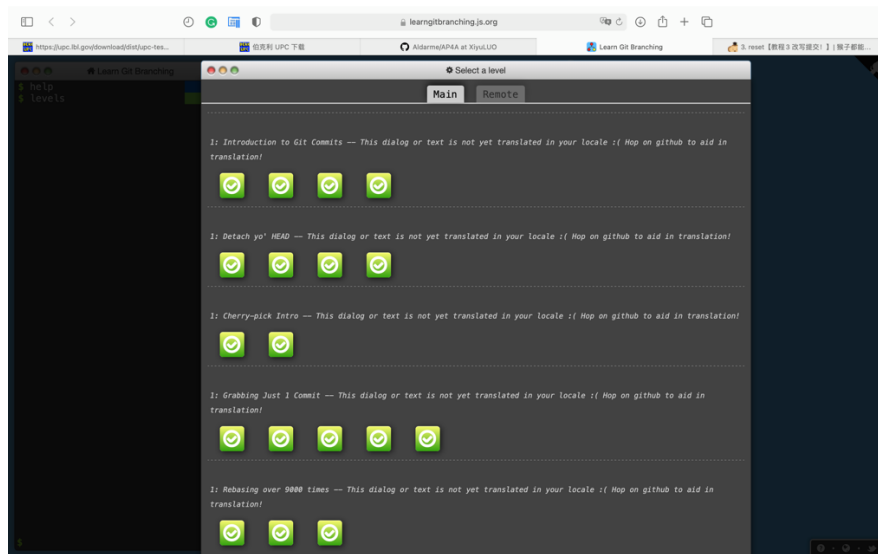


Rapport Commandes Git

1. Le certificat de réussite du tuto git



2. L'explication des commandes git

a) *Git config*

Il nous permet d'obtenir et de définir des variables de configuration. Ces variables peuvent contrôler divers aspects de l'apparence et du fonctionnement de Git. Par exemple, nous pouvons utiliser *git config --global* pour configurer un nom d'utilisateur et un mot de passe.

b) *Git init*

Il nous permet de créer des dépôts git. Cette commande peut être exécutée à la racine d'un projet non-git existant pour initialiser le projet existant en tant que dépôt git, ou elle peut être utilisée pour initialiser un répertoire vide en tant que dépôt git.

c) *Git status*

Il est utilisé pour afficher l'état du répertoire de travail et de la zone de transit. Cette commande montre les changements qui ont été mis à jour, ceux qui ne l'ont pas été, et les fichiers qui n'ont pas été suivis par Git. La commande *git status* ne montre pas les informations qui ont été commises dans l'historique du projet. Pour voir l'historique du projet, utilisez *git log*.

d) *Git add*

Il garde la trace des nouveaux fichiers, ou plutôt ajoute le contenu du répertoire de travail à la zone de transit.

e) *Git push*

Il est utilisé pour télécharger le contenu du référentiel local vers un référentiel distant. C'est l'opposé de *git fetch*, qui importe les commets de la branche distante vers la branche locale, et *git push*, qui commit les changements de la branche locale vers la branche distante.

f) *Git merge*

Il est utilisé pour fusionner une ou plusieurs branches dans la branche que nous avons extraite. Il déplace ensuite le pointeur de la branche courante vers le résultat de la fusion.

g) *Git diff*

Il montre la différence entre un fichier qui a été écrit dans la zone de transit et un fichier qui a été modifié mais pas encore écrit dans la zone de transit.

h) *Git blame*

Il peut retracer l'historique des modifications apportées à un fichier donné. Lorsqu'un membre de l'équipe commit une erreur, il peut être utilisé pour savoir quel membre de l'équipe a modifié une certaine partie du code d'un fichier et trouver le responsable de l'incident.