

Git



Plan

- Qu'est-ce que git ? qu'est-ce que github ?
- Comment installer git / s'inscrire sur github ?
- Exemple d'utilisation de git
- exercices d'application de certaines commandes
- gestion des branches (schéma)
- Exercices de gestion de branches



Qu'est-ce que Git ?

Définition :

C'est un logiciel, un outil permettant de contrôler la ou les versions d'un projet informatique



Qu'est-ce que GitHub ?

Définition :

Github est un site permettant de stocker dans des "repository" des projets. Il peut communiquer avec git afin de pouvoir ajouter/modifier/supprimer des parties d'un projet. D'autres sites sont similaires comme gitlab par exemple.



Installation de git

Windows :

1ère étape: installer git

--> Allez sur le site officiel de Git : git-scm.com

--> Télécharger la dernière version de Git

--> Installer la comme un logiciel classique

--> /!\ pour linux privilégier les command apt install git-all (distrib debian) ou dnf install git-all (basée RPM)



2ème étape: créer un espace qui stockera l'ensemble des fichiers (Github):

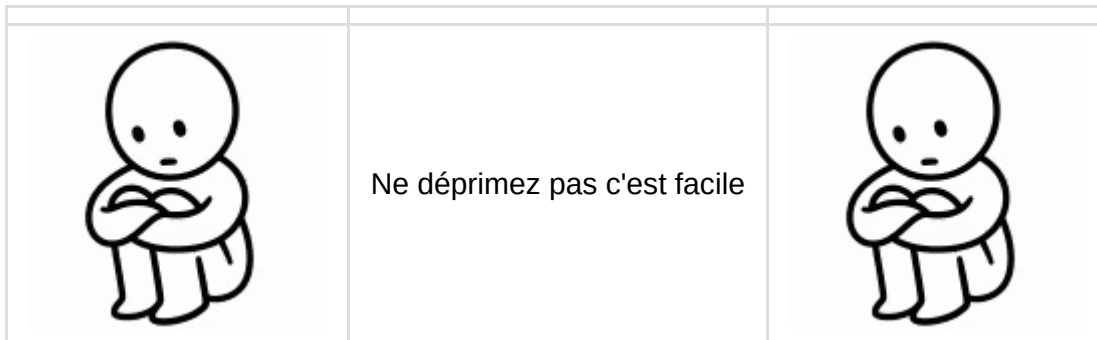
--> Aller sur Github et créer un compte

--> Cliquer sur la flèche pour créer un nouveau repository

--> Vous allez devoir compléter un certain nombre d'information (nom répo, etc...)



L'installation étant terminée il nous reste plus qu'à passer une ligne de commande



3ème étape: commandes --> Récupérer le lien de votre référentiel

--> Lancer votre invite de commande git dans un dossier vide créé à l'avance

--> Effectuer la commande:

```
git clone LienVotreRepo
```



BRAVO ! VOUS AVEZ VOTRE DOSSIER GIT DANS UN PROJET DEV



comment ajouter les modifications sur votre github ?

--> Créer une page web ou un fichier texte

--> Vous allez devoir effectuer 3 commandes:

--> `git add *` pour ajouter les nouveaux fichiers/dossiers créer

--> git commit -m "message" pour écrire un message sur les modifications effectuées
--> git push pour envoyer votre commit vers votre github



Récupérer les modifications de votre github ?

--> git pull pour récupérer le projet modifier de votre github



Git, commande

Commande	Description
git add	ajouter un fichier/dossier a votre git
git rm	supprimer un fichier/dossier de votre git un repository
git commit (-m)	envoyer les changements sur votre git

Commande	Description
git push / pull	envoyer (push) ou récupérer (pull) les modifications d'un repository



Commande	Description
git init	Créer un dépôt Git vide ou réinitialiser un existant
git clone	clone un repository
git status	Afficher l'état de la copie de travail
git checkout	changer de branche



Commande	Description
git branch	afficher la liste des branches
git checkout -b	créer une nouvelle branche
git merge	Fusionner deux ou plusieurs historiques de développement ensemble



exercice d'application de certaines commandes (dev)

Réaliser un site de vente (les produits de votre choix). Vous devrez réaliser 3 commits:

- Le premier sera la création de la page d'accueil (avec seulement l'entête du site).
- Le deuxième sera le remplissage de cette page d'accueil (image, texte, etc...).
- Le dernier sera le footer de la page (lien vers réseaux sociaux, tél ou autres).

Les commits devront être visible sur le github !

Bonus: lorsque vous allez réaliser les trois commits et que vous aurez envoyez les informations sur github, essayer de revenir à un commit précédent (indice: tous les commits ont un identifiant différent).



exercice d'application de certaines commandes (si)

Réaliser un script python (au choix). Vous devrez réaliser 3 commits:

- Le premier sera la création du script (avec un début de script).
- Le deuxième sera la completion de ce script (si vous faites un pendu, finaliser le pendu par exemple, etc...).
- Le dernier sera la création d'un README.txt qui nous expliquera comment lancer le script.

Les commits devront être visible sur le github ! (donc utiliser un git push).

Bonus: lorsque vous allez réaliser les trois commits et que vous aurez envoyez les informations sur github, essayer de revenir à un commit précédent (indice: tous les commits ont un identifiant différent).



La gestion des branches de git

Pour comprendre un peu mieux les branches dans git, je vous propose d'aller sur le site: learngitbranching.js.org et de réaliser les exercices présents sur le site.



exercice (dev)

1- Vous travaillez dans une entreprise. Le chef vous demande de réaliser un projet sur github. Vous serez le chef d'équipe pour ce projet. Afin d'éviter tout bousculement dans le code, vous décidez de réaliser 5 branches distincte: Une branche front pour mathieu, une branche back qui contiendra une branche pour Titouan et Michel, et la branche Master/production qui liera le front et le back. Réaliser ses branches sur votre github puis tester de créer/modifier certains fichier.

2- Si Titouan modifie en premier la branche back, Michel va devoir les récupérer. Mais surprise Titouan a modifié des fichier que Michel avait modifier, la tuille ! il faut donc gérer les conflits. Réaliser la modification des conflits.



exercice (si)

1- Vous travaillez en groupe de deux. Créer un repository et vous travaillerez en collaboration dessus, vous allez devoir réaliser deux scripts:

le premier sera un script permettant de vérifier si le String insérer est un palindrome ou non (fait par l'équipier numéro 1)

le second sera un script permettant la création d'une calculatrice avec les calculs basiques: addition, soustraction, multiplication et division. (fait par l'équipier numéro 2)

2- Après avoir récupérer les modifications de l'autres vous vous rendez compte que vous pouvez rajouter le calcul de puissance. Vous le réalisez donc tous les deux chacun de votre côté (le premier à l'envoyer gagne). Le seconde devra récupérer le fichier modifier sans retirer les

modifications qu'il aura déjà effectuer. Gérer les conflits.



Le problème des blagues sur git ?

C'est que tout le monde n'a pas la même version!



