

TP Versioning - git

Objectifs pédagogiques

- *Prise en main d'un outil de versioning*
- *Utilisation d'une plateforme collaborative de versioning*

Dans le cadre de ce TP, nous travaillerons sur l'un des outils de versioning les plus couramment utilisés, git. Pour cela, nous utiliserons la plateforme collaborative *Github*. GitHub est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git.

TRAVAIL A REALISER

1. Vérifier le statut actuel de votre dépôt (git status).
2. Ajouter à votre répertoire de travail local le dossier PARTIEL, contenant les fichiers :

- `classes.php`
- `traitement.php`

(Le dossier se trouve sur pronote)

3. Que réalise la commande suivante ?

git status -s --untracked-files=no

4. Mettre à jour votre dépôt distant, en utilisant comme message de commit "Ajout programmes partiel"
5. Qu'affiche chacune des commandes suivantes ? Quel est son intérêt ?

**git log --graph --pretty=short
gitk**

6. Procéder aux modifications suivantes dans votre répertoire de travail :

- a. *Fichier : classes.php*

Modifier la fonction **afficher()** de la classe **Personne** comme suit :

```
private function afficher() {  
    $this->num . '<br/>';  
    $this->nom . '<br/>';  
    $this->prenom . '<br/>';  
    $this->entreprise->nom . '<br/>';  
}
```

- b. *Fichier : traitement.php*

Apporter les modifications suivantes :

```
$developpeur = array(new Developpeur(550,"Le Flock",  
    "Louis-Marie", "lmleflock@test.net","0606060606",2500,  
    "Java"),  
    new Developpeur(650,"Poirie", "Ethan", "ach@test.net",  
    "0707070707",2750, "C++")) ;
```

7. Vérifier les modifications avec la commande **git status**.

8. Vous avez au final décidé de renoncer aux dernières modifications, quelles commandes exécuter ?
9. Vérifier que la désindexation a bien eu lieu, avec la commande **git status**.
10. Donner la commande qui permet de lister les différents "commits" effectués.
11. Choisir un **CommitId** (numéro commit), autre que le dernier et lancer la commande :

git checkout CommitId

12. Décrire les changements ayant eu lieu suite à l'exécution de la commande précédente.
13. Qu'affiche la commande **git status** ?
14. Le dépôt local n'est plus synchronisé avec le dépôt distant suite au git checkout. Lancer la commande suivante :

git checkout main

15. Quel résultat obtient-on ?
16. A partir d'un autre répertoire sur votre disque, lancer la commande suivante :

git clone url_depot_distant

17. Décrire le résultat obtenu ? Quel est l'intérêt de cette opération ?

LA COLLABORATION

On va s'intéresser dans cette partie à l'utilisation de la collaboration dans le cadre d'un même projet.

18. Envoyez une invitation à un binôme de votre choix.
19. Demandez lui de réaliser la modification suivante sur le fichier **traitement.php**

```
$developpeur = array(  new Developpeur(550, "Le Flock", "Louis-Marie",
"lmleflock@test.net", "0606060606", 2500, "Java"),
new Developpeur(650, "Chellal", "Ali", "ach@test.net", "0707070707", 2750,
"C++"),
new Developpeur(151,"Katsiev", "Amir", "amir@test.net", "0606060606",2750,
"Kotlin")) ;
$manager = array(new Manager(500, "RIPAULT", "Alexis", "alexis@mana.net",
"0606060606",4000, "DSI")) ;
```
20. Visualiser les modifications effectuées par le développeur invité :
21. Quelles conséquences sur votre dépôt local et distant ?
22. Afficher le niveau d'implication de ce développeur.
23. Demander lui de vous inviter sur son dépôt.
24. Comment devez-vous procéder pour intégrer une nouvelle personne dans votre projet ?
25. Testez cette opération et donnez un visuel de son dépôt distant.