

## **Travail Pratique2**

**Cours : Environnement Web 1**

**Date de remise : 6 décembre 2022 (5% de moins pour chaque jour de retard)**

**Pondération : 10%**

**Professeur : Lorry James, Encarnacion**

**Objectifs : Mettre en pratique les notions vues à ce jour.**

# Travail pratique 2

## 1. Modalités

Pondération 10% de la note finale

Échéance de remise: 6 décembre 2022 à 23h PM

Travail à réaliser en individuel. Attention, le professeur peut s'il le juge nécessaire poser des questions à chacun étudiants pour s'assurer qu'ils sont les auteurs authentiques du travail remis.

Le but de ce travail est de vous évaluer sur l'ensemble des contenus abordés dans le cadre du cours et de vos capacités à réfléchir pour trouver une solution. Tout plagiat sera sévèrement sanctionné.

## 2. Directives.

### **Remise**

Une fois le travail terminé remettre votre fichier sur Léa dans l'espace intitulé «TP2 ».

**Il faut faire des captures d'écrans de chaque étape**

## Télécharger et installer Git sur Debian

Ouvrez votre machine virtuelle Debian et installez Git

```
sarah@debian:~$ sudo apt install git
[sudo] password for sarah:
Reading package list... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.30.2-1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

## Configurer votre identité

Commencez par renseigner votre nom et votre adresse e-mail. **C'est une information importante** car vous en aurez besoin pour toutes vos validations dans Git :

```
sarah@debian:~$ git config --global user.name "SarahFergusonPoirier"
sarah@debian:~$ git config --global user.email sarahfergusonpoirier@hotmail.com
```

Pour vérifier que vos paramètres ont bien été pris en compte, et vérifier les autres paramètres, il suffit de passer la commande `git config --list`

```
sarah@debian:~$ git config --list
user.name=SarahFergusonPoirier
user.email=sarahfergusonpoirier@hotmail.com
```

Allez sur [www.github.com](https://github.com) et copier le lien https de votre projet

```
https://github.com/SarahFergusonPoirier/web1.git
```

Maintenant vous devez cloner le projet dans le dossier **Documents**

```
sarah@debian:~$ cd Documents
sarah@debian:~/Documents$ git clone
https://github.com/SarahFergusonPoirier/web1.git
```

Quand vous lancez la commande **ls** que voyez-vous :

```
sarah@debian:~/Documents$ ls
web1
```

Modifier le contenu du fichier index.html en ajoutant un nouveau paragraphe

```
sarah@debian:~/Documents$ cd web1
sarah@debian:~/Documents/web1$ nano index.html
```

Ajouter ce fichier dans git et faites un **commit** des changements effectués

```
sarah@debian:~/Documents/web1$ git add index.html
sarah@debian:~/Documents/web1$ git commit -m "Ajout d un paragraphe dans le
fichier index.html"
[master (root-commit) 841460e] Ajout d un paragraphe dans le fichier index.html
1 file changed, 11 insertions(+)
create mode 100644 web1/index.html
```

Si vous avez une erreur d'authentification. Donner les étapes pour générer un token:

Aucune erreur d'authentification

Copier le token qui a été généré ensuite utilisez le comme mot de passe.

Une fois le commit effectuer lancer la commande **git log**, que voyez-vous? À quoi sert cette commande selon vous?

```
sarah@debian:~/Documents/web1$ git log
commit 3652b7a25800089q9033477e256cde437dc00444 (HEAD -> main)
Author: SarahFergusonPoirier sarahfergusonpoirier@hotmail.com
Date: Thu Dec 1 23:24:02 2022 -0500
    Ajout d un paragraphe dans le fichier index.html
commit 879dff70e0345d0607a7dda33e244bb4e78c97ea (origin/main, origin/HEAD)
Author: SarahFergusonPoirier sarahfergusonpoirier@hotmail.com
Date: Thu Dec 1 22:27:07 2022 -0500
    Ajout des fichiers html et css de base
```

Cela sert à montrer l'historique des commits du répertoire Git.

## Création des branches

Maintenant créer les branches tache1 et tache2 :

```
sarah@debian:~/Documents/web1$ git branch tache1  
sarah@debian:~/Documents/web1$ git branch tache2
```

Passer à la branche tache1 :

```
sarah@debian:~/Documents/web1$ git checkout tache1  
Switched to branch 'tache1'
```

Modifier le fichier .css en changeant la couleur à **green** :

```
sarah@debian:~/Documents/web1$ nano styles.css
```

Ajouter ce fichier dans git et faites un **commit** des changements effectués

```
sarah@debian:~/Documents/web1$ git add styles.css  
sarah@debian:~/Documents/web1$ git commit -m "Modification du fichier styles.css"  
[tache1 7fe5bc1] Modification  
1 file changed, 1 insertions(+)  
create mode 100644 web1/style.css
```

## Fusionner dans Git

Fusionner tache1 dans tache2 :

```
sarah@debian:~/Documents/web1$ git checkout tache2  
Switched to branch 'tache1'  
sarah@debian:~/Documents/web1$ git merge tache1
```

Que contient maintenant le fichier .css :

Les modifications demandées plus tôt, c'est-à-dire de remplacer la couleur rouge par la couleur verte.

Dans **github** que remarquez-vous?

Il n'y a aucun changement, car je n'ai fait que des commits et non des push.

Pouvez-vous simuler un cas de conflit de fusion, et comment le gériez-vous?

Si j'avais modifié le fichier css dans la branche tache2 avant de fusionner les branches tache1 et tache2, en modifiant la couleur pour blue, il y aurait eu conflit de fusion, car les deux changent la couleur de h1 de façon différente.

Pour résoudre le conflit, il faut ouvrir le fichier causant le conflit (dans cet exemple, le fichier css) dans un éditeur de code, trouver la source du conflit identifiée par git et choisir la version finale souhaitée. Pour savoir quel fichier cause le conflit en cas de doute, il faut faire la commande git status. À l'intérieur du dossier, le conflit est identifié entre <<<<<<< HEAD et >>>>>>> Nom de la branche. Les deux modifications sont séparées par =====. Une fois le conflit réglé, il faut faire à nouveau un commit.