**Classe:**

**1**

**ère**

**année**

**MA 24**

**Semestre:**

2

ème

**Date**

:

6

février

2020

**Appui programmation**

**Sujets :**

**Git**



# OBJECTIFS

* Etre capable d'utiliser un repository.
* Etre capable de travailler en groupe avec des branches
* Etre capable de récupérer des anciennes versions

## Git : bases

Toutes les commandes seront faites sur bash.

Pour chaque commande, notez ce que vous avez tapé dans les zones prévues à cet effet. Il se peut que dans certains cas vous ayez plus d’une commande à écrire.

# CREATION DU REPOSITORY

## Configuration de git

Configurez git avec votre nom, votre email et la coloration syntaxique.

|  |
| --- |
| $ git config --global user.name "[Jan]"  $ git config --global user.email "[Jan.blatter@cpnv.ch]"  $ git config --global color.ui auto |

Écrivez la commande pour générer des alias dans le fichier de configuration de git. (vous pouvez en créer d’autres si vous le souhaitez)

* git checkout 
* git status 
* git commit 

* git branch 

## Création d’un repository distant

Créez un repository sur github.

Cloner le sur un répertoire vous avez en local (créez-en un si besoin). Ce sera votre répertoire de travail pour votre projet.

|  |
| --- |
|  |
|  |

Quelle commande écririez-vous si vous deviez initialiser un repository en local ?

|  |
| --- |
| git-init |

|  |
| --- |
|  |

# COMMITS

## 1er commit

Créer un readme qui résume en 2 lignes le fonctionnement d’une application que vous souhaiteriez développer.

Faites votre premier commit

|  |
| --- |
| $ git add .  $ git status  $ git commit –m «  Ajout du fichier read »  $ git push |

Modifiez le fichier que vous venez de créer et mettez-y la date de création et l’auteur. Avant de faire le commit, afficher les différences de contenu avec le dernier commit

|  |  |
| --- | --- |
| Faites votre premier commit   |  | | --- | | $ git add .  $ git status  $ git commit –m «  Ajout de la date de création et auteur»  $ git push | |

## Autres commits et suivi

Modifiez le fichier readme et notez-y à présent comment l’installer et faites un commit avec comme message « ajout instructions installation »

|  |
| --- |
| $ git add .  $ git status  $ git commit –m «ajout instructions installation »  $ git push |

Affichez les différents commits

|  |
| --- |
|  |
|  |

Créez un nouveau fichier credits.txt où vous noterez dans l’ordre alphabétique les 2 développeurs du projet. Faites votre commit et l’ajout avec comme message « ajout du fichier credits.txt

|  |
| --- |
|  |

Revenez au commit « ajout de la liste des jetons et de leur couleur »

|  |
| --- |
|  |

Comment feriez-vous faire pour ne pas stocker les fichiers dll/exe dans vos commits ?

|  |
| --- |
|  |

# BRANCHES

## Création de branche

Créez une branche par élève :

* 1 pour les fonctionnalités que l’élève1 développera
* 1 pour les fonctionnalités que l’élève2 développera

|  |
| --- |
|  |

Placez-vous sur la branche vous correspondant

|  |
| --- |
|  |

## Merge de branche

Modifiez chacun votre fichier en y mettant un commentaire avec votre nom et la date du jour puis faites un commit avec comme message « *votre nom* et la date » puis faites un merge sur le master

|  |
| --- |
|  |