

## TP 1 - INSTALL

Configurer Notepad++ dans les variables d'environnement : **CMD** notepad++

Installer GIT (Choose notepad++ as editor pendant l'installation)

- C:\Program Files\Git\mingw64\etc, éditer le fichier **gitconfig** et voir la rubrique : core

## TP2 - SETUP

Configurer les identifiants

```
$ git config --global user.name "Your Name"
$ git config --global user.email "your_email@whatever.com"
```

Configurer les préférences supplémentaires

```
$ git config --global core.autocrlf false
```

```
$ git config --global core.safecrlf false
```

## TP3 - CREER UN PROJET

Créer un dossier « bonjour »

```
$ mkdir bonjour
$ cd bonjour
$ echo 'bonjour, monde' >> bonjour.rb
```

Initialiser un repository git dans le dossier bonjour

```
$ git init
```

OUTPUT

Ajouter le programme au repository

```
$ git add bonjour.rb
$ git commit -m "Premier Commit"
```

OUTPUT

## TP4 – VERIFIER LES STATUS

Vérifier le statut du repository

```
$ git status
```

OUTPUT (Explication)

## TP5 – EFFECTUER DES CHANGEMENTS

Changer le programme « bonjour »

Dans le fichier bonjour.rb, **ajoutez** « C'est moi Picsou ! »

Vérifier le statut

```
$ git status
```

OUTPUT (Explication)

### TP6 – STAGING AREA

Informez Git de l'ajout du fichier dans l'index

```
$ git add bonjour.rb  
$ git status
```

OUTPUT (Explication)

### TP7 – STAGING ET COMMITTING

**Explication** de la philosophie des commits

```
$ git add a.rb  
$ git add b.rb  
$ git commit -m "Changes for a and b"  
  
$ git add c.rb  
$ git commit -m "Unrelated change to c"
```

(Explication)

### TP8 – COMMITTING

Faire un commit. Par suite de cette action, commentez votre commit et puis enregistrez et fermez afin de vérifier la fin du processus de commit

```
$ git commit
```

OUTPUT (Explication) du pop-up de commentaire (n'oubliez pas de valider et de fermer)

OUTPUT (Explication) fin du commit

Revérifiez le statut

```
$ git status
```

OUTPUT (Explication)

### TP 9 – GIT TRAITE AVEC LES CHANGEMENTS ET NON LES FICHIERS

Ajouter au fichier bonjour.rb « **si ce n'est pas moi, c'est Riri !** » et ajouter le fichier à l'index

```
$ git add bonjour.rb  
$ git status
```

Rajouter au fichier « **sinon, c'est Fifi** » et ne l'ajoutez pas cette fois-ci à l'index

```
$ git status
```

OUTPUT (Explication)

Commettons et voyons

```
$ git commit -m "Ajout de Riri"  
$ git status
```

OUTPUT Explication

Ajouter le second changement

```
$ git add .  
$ git status
```

OUTPUT Explication

Committer le second changement

```
$ git commit -m "Added a comment"
```

OUTPUT Explication

## TP – 10 HISTORIQUE

Voir l'historique d'un projet

```
$ git log
```

OUTPUT

Historique en une ligne

```
$ git log --pretty=oneline
```

OUTPUT

Historique avancé

```
$ git log --pretty=oneline --max-count=2  
$ git log --pretty=oneline --since='5 minutes ago'  
$ git log --pretty=oneline --until='5 minutes ago'  
$ git log --pretty=oneline --author=<your name>  
$ git log --pretty=oneline --all
```

Historique avec style

```
$ git log --oneline --graph --decorate
```

OUTPUT Explication