Scénario de l'atelier « GIT »

1. Identification des étapes

- Installation de Git
- Configuration de Git
- Préparation du projet de test
- Les commits (validations)
- Les branches et fusions
- Clonage d'un projet existant
- Pousser votre code vers Github

2. Déroulement du scénario

2.1. Installation de Git

- Sous Linux:

$ apt-get install git	**(Pour Debian et Ubuntu)**
$ yum install git	**(Pour le système Fedora)**

- Sous Windows:

  Téléchargez simplement le fichier exe d'installateur depuis le lien :

  https://gitforwindows.org/ ou https://gitscm.com/download/win

- Sous Mac:

```
$ brew install git
```

Ou en choisissant une version parmi la liste
https://sourceforge.net/projects/git-osx-installer/files/

2.2. Configuration de Git

- Configurer Git globalement

$ git config --global user.name <username>

$ git config --global user.email <email>

Exemple: $ git config --global user.name Adam

$ git config –global user.email Adam@gmail.com

2.3. Préparation du projet de test

- Créer un nouveau dossier gitTest et y accéder

```
$ mkdir gitTest
$ cd gitTest
```

- Initialiser le nouveau répertoire en tant que répertoire git

```
$ git init.
```

- Afficher le statut actuel du répertoire

```
$ git status
```

- Création d'un fichier de test

```
$ nano hello.txt
```

- Ecrire dans le fichier les 3 lignes suivantes :

```
Hello Bejaia
Hello Algiers
Hello Oran
```

- Sauvegarder et quitter (Ctrl+x puis Y et taper sur « entrer »)

2.4. Les commits (validations)

- Afficher le statut actuel du répertoire

```
$ git status
```

- Ajouter hello.txt au contrôle de version

```
$ git add hello.txt
```

- Afficher le statut actuel du répertoire

```
$ git status
```

- Valider les modifications (commit) dans le répertoire actuel (-m indique à git de sauvegarder un « message de validation » avec ce commit)

```
$ git commit -m 'Add hello.txt'
```

- Afficher le statut actuel du répertoire

```
$ git status
```

- Afficher le journal des commits (une sorte d'historique)

```
$ git log
```

- Supprimer la dernière ligne du fichier « Hello Oran » (Ouvrir le fichier avec la commande **$** nano hello.txt)
- Afficher le statut actuel du répertoire

```
$ git status
```

- Afficher les modifications

```
$ git diff
```

- Commiter les changements (-a indique qu'on ajoute ce fichier à Git

```
$ git commit -a -m 'Remove Hello Oran'
```

- Dans la ligne 2 modifier « Algiers » par « Blida »
- Afficher les modifications effectuées

```
$ git diff
```

- Commiter les changements

```
$ git commit -a -m 'Replacing Algiers by Blida'
```

- Ajouter une troisième ligne « Hello Biskra » puis afficher les changements

```
$ git diff
```

- Commiter les changements

```
$ git commit -a -m 'Adding hello biskra'
```

- Afficher l'ensemble des commits

```
$ git log
```

2.5. Branchement et fusion (Branching and merging)

- Création d'une nouvelle branche «BejaiaBranch»

```
$ git branch BejaiaBranch
```

- Se positionner à la branche « BejaiaBranch »

```
$ git checkout BejaiaBranch
```

- Afficher le statut du répertoire dans la branche « BejaiaBranch »

```
$ git status
```

- Lister toutes les branches de votre répertoire

```
$ git branch
```

- Ajouter au fichier hello.txt la ligne « Hello Cap Carbon » puis afficher les changements

```
$ git diff
```

- Commiter les changements

```
$ git commit -a -m 'Adding hello Cap Carbon'
```

- Lister l'ensemble des commits

```
$ git log
```

- Créer un nouveau fichier de test

```
$ nano branch-example.txt
Ecrire dans le fichier : « Hello Akbou » puis sauvegarder et quitter
```

- Ajouter branch-example.txt à Git

```
$ git add branch-example.txt
```

- Commiter les modifications dans le répertoire actuel

```
$ git commit -m 'branch-example.txt'
```

- Lister l'ensemble des commits

```
$ git log
```

- Revenir à la branche principale

```
$ git checkout master
```

- Fusionner les modifications de la branche "branch-example" dans la branche actuelle

```
$ git merge BejaiaBranch
```

2.6. Cloner un projet existant Github

- Sur GitHub, accédez à la page principale du projet
- Sous le nom du projet, cliquez sur Cloner ou télécharger.
- Dans la section Cloner avec HTTP, copier l'URL de clone du projet.
- Se positionner dans le répertoire qui contiendra le projet
- Cloner le projet avec l'URL copié

```
$ git clone <url>
```

Remarque : Prendre comme exemple le projet suivant : https://github.com/macagua/example.java.helloworld

2.7. Pousser (Push) votre code sur Github

- Créer un compte sur (https://github.com/login)
- Créer un nouveau dépôt (en haut à gauche) ayant me même nom que votre projet en local **gitTest**
- Cliquer sur 'clone or download' et copier l'adresse 'Clone with HTTPS'
- Ajoutez l'URL du répertoire distant sur lequel votre projet local sera poussé.

```
$ git remote add origin <https://github.com/username/Repo.Git>
```

- Vérifier l'URL distante

```
$ git remote -v
```

- Pousser les changements de votre projet local vers le dépôt Git
-

```
$ git push origin master
```