



# Module 2

## Installation de Git

# Contenu du module

- Principes d'installation selon le système d'exploitation
  - Installation sous Linux
  - Installation sous macOS
  - Installation sous Windows
- Validation de l'installation et premières commandes
  - L'aide en ligne de commande
  - Principes et syntaxe des commandes Git
- Configuration post-installation
  - Création de comptes d'utilisateur

# Installation de Git

- Installation sous Linux
  - A partir de paquets existants dans la distribution.
  - A partir des sources.
- Installation sous macOS
  - Avec Homebrew
  - Avec XCode
- Installation sous Windows
  - Avec un installeur

# Installation sous Linux

- A partir des paquets d'une distribution
  - RedHat, CentOS, Fedora
    - `sudo yum install git`
  - Debian, Ubuntu
    - `sudo apt install git`
- A partir des sources
  - Installation des dépendances
    - *make, libssl-dev, libz-dev, gettext, libexpat1-dev, libcurl4-openssl-dev, build-essential*
  - Récupération et décompression du code source de Git
    - <https://github.com/git/git/archive/v2.27.0.tar.gz>
  - Compilation et installation
    - `cd code_git/git-2.27.0`
    - `make prefix=/usr/local all`
    - `make prefix=/usr/local install`

# Installation sous macOS

- Git est fourni avec l'IDE XCode
  - Aucune installation supplémentaire nécessaire.
- *Command Line Tools for XCode* permet d'installer les outils de développement en ligne de commande (dont Git !) sans avoir besoin d'installer tout XCode.
  - *Apple Developer ID* nécessaire !
- Avec Homebrew
  - Télécharger et installer **Homebrew** (Un gestionnaire de paquets pour macOS)
    - <https://brew.sh>
  - Dans un terminal
    - `brew install git`

# Installation sous Windows

- Un installeur graphique est disponible
  - <https://git-scm.com/download/win>
- Il installe :
  - Une intégration à l'explorateur Windows
  - Git Bash, permettant de disposer du shell Bash sous Windows
    - Et donc de certaines commandes Linux.
  - Un gestionnaire de mots de passe
    - Permettant de mémoriser les authentifications aux différents dépôts.

## Git Bash sous Windows 10



```
MINGW64:/c/Users/Etienne/FormationGIT
Etienne@ELANGLET-PC MINGW64 ~
$ pwd
/c/Users/Etienne
Etienne@ELANGLET-PC MINGW64 ~
$ mkdir FormationGIT
Etienne@ELANGLET-PC MINGW64 ~
$ cd FormationGIT/
Etienne@ELANGLET-PC MINGW64 ~/FormationGIT
$ touch README.md
Etienne@ELANGLET-PC MINGW64 ~/FormationGIT
$ ls -l
total 0
-rw-r--r-- 1 Etienne 197121 0 juin 22 14:19 README.md
Etienne@ELANGLET-PC MINGW64 ~/FormationGIT
$ |
```

# La ligne de commande

- Git s'utilise en ligne de commande !
  - Terminal Linux ou macOS
  - Git Bash sous Windows (recommandé) ou une simple Invite de commande
- Syntaxe :
  - `git <commande> [<arguments>]`

```
$ git
usage: git [--version] [--help] [-C <path>] [-c name=value]
         [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
         [-p|--paginate|--no-pager] [--no-replace-objects] [--bare]
         [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
         <command> [<args>]

The most commonly used git commands are:
  add           Add file contents to the index
  bisect        Find by binary search the change that introduced a bug
  branch        List, create, or delete branches
  checkout      Checkout a branch or paths to the working tree
  clone         Clone a repository into a new directory
  commit        Record changes to the repository
  diff          Show changes between commits, commit and working tree, etc
  fetch         Download objects and refs from another repository
  grep          Print lines matching a pattern
  ...

'git help -a' and 'git help -g' lists available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
```

# L'aide

- `git help`
  - Aide sur la syntaxe générale de Git.
- `git help <commande>`
  - Aide sur la syntaxe et l'usage de la commande donnée en argument.
    - Affiche une « man page » dans un navigateur Web.
- `git help tutorial`
  - Un tutoriel précis et concis sur Git.



# Configuration de Git

- Une configuration minimale de Git est nécessaire avant de pouvoir l'utiliser.
- Git dispose de très nombreuses options permettant d'adapter son fonctionnement en fonction des usages.
  - `git help config` permet de connaître les différentes possibilités de configuration de l'outil.
- La configuration peut se faire à trois niveaux :
  - Le niveau système :
    - Les options concernent tous les utilisateurs de la machine.
    - `git config --system ...`
  - Le niveau utilisateur :
    - Les options ne concernent que l'utilisateur qui les définit.
    - `git config --global ...`
  - Le niveau local :
    - Les options ne sont alors valides que pour le dépôt Git sur lequel les options sont définies.
    - `git config --local ...`

# L'identité utilisateur

- L'identité utilisateur permet de savoir qui apporte les modifications aux fichiers du dépôt.
  - Elle doit être définie avant de pouvoir utiliser les commandes Git.
- L'identité est constitué :
  - D'un nom d'utilisateur.
  - D'une adresse email.
- Sur la ligne de commande :
  - `git config --global user.name "Etienne LANGLET"`
  - `git config --global user.email "elanglet@eni.fr"`
- NOTE :
  - L'identité (et autres préférences) est enregistrée dans un fichier nommé `.gitconfig` et stockée dans le répertoire personnel de chaque utilisateur.

# L'éditeur de texte

- Git utilise un éditeur de texte pour, par exemple :
  - Rédiger les commentaires associés à chaque commit.
  - Permettre la résolution des conflits.
  - Et de manière générale, pour la saisie de toute information nécessaire lors de la manipulation du dépôt.
- L'éditeur par défaut est Vi ...
- Une commande git config permettra de choisir un éditeur alternatif est d'enregistrer cette préférence.
  - `git config --global core.editor "<chemin vers l'editeur>"  
[options]"`
- Exemple avec Notepad++ :
  - `git config --global core.editor "'C:\Program Files  
(x86)\Notepad++\notepad++.exe' -multiInst -notabbar -nosession  
-noPlugin"`

# Un proxy HTTP(S)

- Lors d'opérations sur un dépôt distant en HTTP ou HTTPS, il peut être nécessaire de spécifier un serveur proxy à utiliser pour permettre à Git de sortir sur Internet.
- `git config --global http.proxy  
http://user:pass@proxyhost:proxyport`
- `git config --global https.proxy  
https://user:pass@proxyhost:proxyport`

# Les alias

- Certaines commandes Git sont riches de plusieurs options qui peuvent être combinées.
  - `git log --graph --decorate --oneline ...`
- Pour se faciliter la vie, il est possible de créer des alias (des raccourcis) sur ces commandes.
- Les alias sont stockés dans la configuration, ils se créés donc avec la commande `git config`.
- Exemple pour la commande précédente :
  - `git config --global alias.ll "log --graph --decorate --oneline ..."`
- L'alias créé se nomme `ll`
  - `git ll`

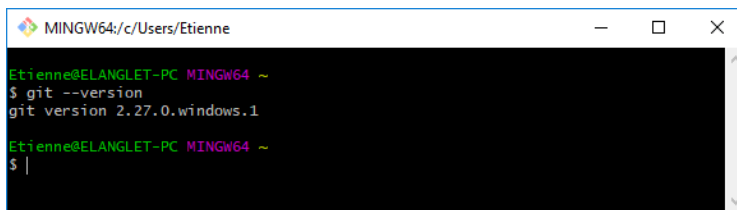
# Travaux Pratiques



[www.eni-service.fr](http://www.eni-service.fr)

# Travaux pratiques

- Exercice 1 : Installation de Git sous Windows
  - Téléchargez la dernière version de Git à l'adresse suivante :
    - <https://git-scm.com/downloads>
  - Lancez l'installation à partir de l'installateur téléchargé.
    - Conservez les options d'installation par défaut.
      - (Les options seront détaillées pendant le corrigé.)
  - Validez l'installation :
    - Lancez **Git Bash** et tapez la commande `git --version`
    - Le résultat doit ressembler à ceci :



```
MINGW64:/c/Users/Etienne
Etienne@ELANGLET-PC MINGW64 ~
$ git --version
git version 2.27.0.windows.1
Etienne@ELANGLET-PC MINGW64 ~
$ |
```

# Travaux pratiques

- Exercice 2 : Configuration de Git
  - Depuis **Git Bash**
  - Configurez votre identité utilisateur Git en spécifiant :
    - Votre nom d'utilisateur.
    - Votre email.
  - Vérifiez que les informations d'identité sont bien enregistrées avec la commande `git config --list`
  - Configurez ensuite l'éditeur par défaut de Git de sorte à utiliser Notepad++
  - Vérifiez la valeur de l'option avec `git config --list`
  - Observez le contenu du fichier `.gitconfig` situé dans votre répertoire personnel (`C:\Users\<nom d'utilisateur>`)



# Fin du module

- Avez-vous des questions

