

Git

Git est un logiciel très puissant et considéré par beaucoup comme un indispensable.

Le logiciel en lui même permet :

- D'avoir un historique de son code et ainsi pouvoir revenir dans un état précédent du projet;
- Gérer plusieurs versions du même code simultanément (avec les branches);
- Faire des backups de son code sur un/des serveur·s distants;
- Définir des releases dans l'historique (avec des tags)
- Collaborer avec d'autres sur le projet (avec les patches)

Mais quand on utilise une "git forge" (un site pour héberger le code tel que Gitlab, GitHub ou autre) on peut avoir encore d'autres avantages :

- Un moyen de partager facilement le code à tout le monde si on le souhaite
- Un "issue tracker" (c'est un genre de forum pour gérer le développement du projet)
- Un moyen pour les gens de faire des copies du code (appelées "fork"), de les modifier et d'ainsi pouvoir fusionner les deux projets en ajoutant les changements en un clic
- Un moyen d'avoir une page d'accueil du projet avec un README formaté
- Un wiki pour gérer la documentation du projet sans se prendre la tête
- Une vision claire du projet, de son historique, etc. Même pour celles et ceux qui ne connaissent pas Git.

Et avec certaines fonctionnalités de plus haut niveau de ces derniers :

- Un hébergement automatique de certain types de sites (avec Pages)
- Une automatisation des tests, build et/ou déploiement du projet en continu (voir chapitre suivant)

Installation de Git

Sur Linux il suffit d'installer le paquet `git` disponible dans tous les gestionnaires de paquets et parfois même pré-installé dans certaines distributions.

Sous Windows ou macOS, il faut l'installer depuis le site: <https://git-scm.org>

Une fois cela fait, sur Linux ou macOS on peut ouvrir un terminal, et sur Windows on peut faire "clic droit" dans un dossier puis "ouvrir Git Bash".

Ensuite on va donner quelques informations à Git qui vont se retrouver dans l'historique de nos projets.

```
git config --global user.name "Votre nom"
git config --global user.email masuperadresse@mail.com
```

Petit diagramme

Voici un petit diagramme pour vous aidez à vous retrouver dans les différents termes (HEAD, commit, branche, tag, remote, stash, index, etc)

Zoo diagram Gitlab
Image diagram type unknown

Branch diagram Gitlab
Image diagram type unknown

Revision #1

Created 26 April 2023 19:09:18 by SnowCode

Updated 26 April 2023 19:09:18 by SnowCode