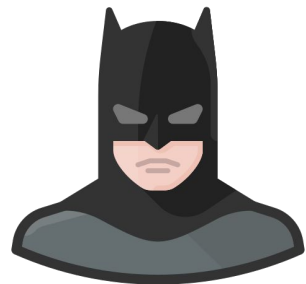


GIT



SAMPAIO Nicolas



- email : nicolas.sampaio@lapiscine.pro
- compte github : [SAMPAIO1748](#)
- développeur web :



Symfony



Définition :

Un logiciel de gestion de versions (ou VCS en anglais, pour version control system) est un logiciel qui permet de stocker un ensemble de fichiers en conservant la chronologie de toutes les modifications qui ont été effectuées dessus.

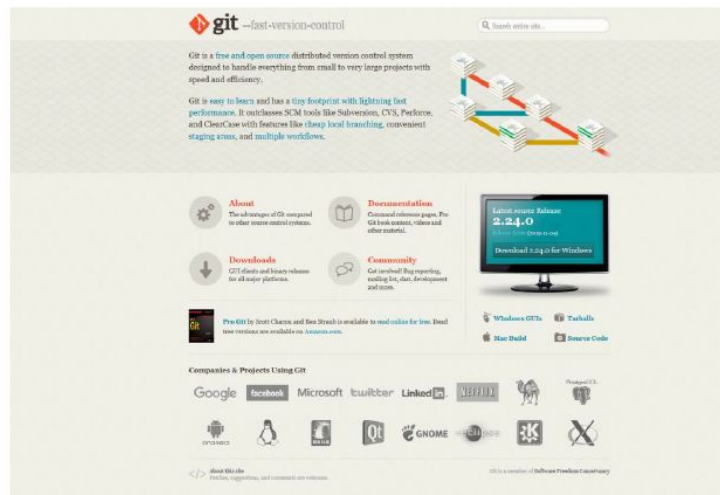
Il permet notamment de retrouver les différentes versions d'un lot de fichiers connexes.

Installation et environnement

Installation

Pour utiliser GIT vous aurez besoin :

1. GIT installé en local (<https://git-scm.com>)
2. Un serveur de stockage (peut être en local ou vers une plateforme)

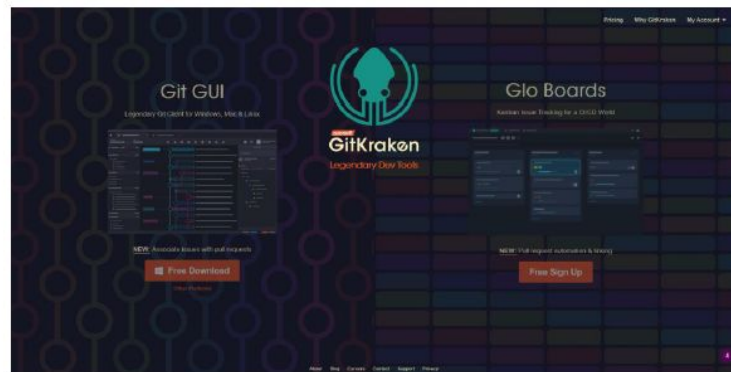
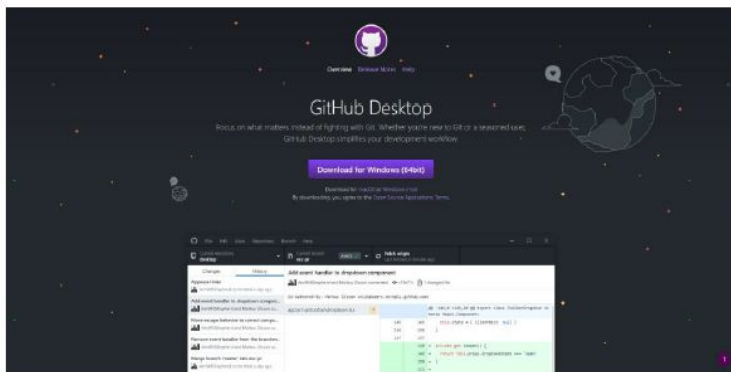


Installation et environnement

Installation

Pour utiliser GIT vous aurez besoin :

3. logiciels

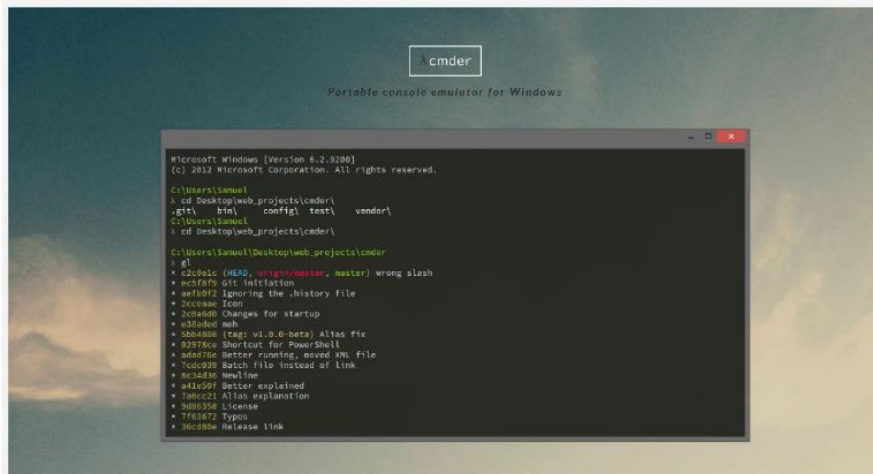


Installation et environnement

Installation

Pour utiliser GIT vous aurez besoin :

3. Logiciels (bonus)



The screenshot shows a Windows command prompt window titled "cmdr" with the subtitle "Portable console emulator for Windows". The window displays the output of a command, showing a list of commit messages from a Git repository. The messages are listed with their commit hashes and descriptions.

```
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\Samuel> cd Desktop\web_projects\cmdr\
.git\ bin\ config\ test\ vendor\
C:\Users\Samuel> cd Desktop\web_projects\cmdr\
C:\Users\Samuel\Desktop\web_projects\cmdr>
* gl
* c2c6c1 (HEAD, origin/master, master) wrong slash
* ac5f0f9 Git Initiation
* aeef0f2 Ignoring the .history file
* 2cc9ae Icon
* 2cc9d06 Changes for startup
* e35dded meh
* 5b0d0e (tag: v1.0.0-beta) Alias fix
* 02978cc Shortcut for PowerShell
* add76e Better running, moved XML file
* 76cd939 Batch file instead of link
* 6c34d36 NewLine
* 441c50f Better explained
* 76cc211 Alias explanation
* 5d86358 License
* 7761672 Typos
* 3acc006 Release link
```

Commandes

Les commandes à connaître

git init : Initialise un nouveau dépôt Git.

git config : lance la procédure de configuration de Git.

git help : Affiche une liste de commande utile.

git status : Vérifie le statut de votre repository. Affiche les fichiers du repo, les modifications, la branche courante.

git add : Permet de prendre en compte les fichiers dans votre futur commit.

git commit : Permet de sauvegarder une image de l'instant T de votre code afin de l'envoyer plus tard vers le serveur. (git commit -m "Le message qui illustre vos modifs").

Commandes

Les commandes à connaître

git branch : Permet de créer une nouvelle « branche » afin de travailler sur une version parallèle (git branch branche_de_jerome).

git checkout : Permet de changer de branche (git checkout branche_de_jerome).

git merge : Permet de fusionner deux branches (git merge branche_de_jerome) va fusionner la branche_de_jerome avec la branche principale (master).

git push : Permet d'envoyer vos modifications vers le serveur.

git pull : Permet de récupérer la dernière version sur votre ordinateur d'une branche à jour.


git clone : Permet de récupérer un projet depuis un serveur et de le cloner sur son ordinateur.

Processus

Processus et cycle de fonctionnement

Étape 1 : Créer un repo sur son serveur

Étape 2 : Créer le repo sur son ordinateur



```
echo "# cours-git" >> README.md
```

```
git init
```

```
git add README.md
```

```
git commit -m "first commit"
```

```
git remote add origin
```

```
https://github.com/SAMPAIO1748/cours git.git
```

```
git push -u origin master
```

Processus

Processus et cycle de fonctionnement

1

```
C:\Users\jerom  
λ cd C:\laragon\www
```

On se déplace dans notre dossier

2

```
C:\laragon\www  
λ mkdir cours-git
```

On crée un dossier pour notre projet

3

```
C:\laragon\www  
λ cd cours-git
```

On se déplace dans notre projet

4

```
C:\laragon\www\cours-git  
λ git init  
Initialized empty Git repository in C:/laragon/www/cours-git/.git/
```

On initialise le projet

Processus

Processus et cycle de fonctionnement

5

```
C:\laragon\www\cours-git (master -> origin)  
λ echo "# cours-git" >> README.md
```

On crée un fichier readme.md

6

```
C:\laragon\www\cours-git (master -> origin)  
λ git add README.md
```

On ajoute notre fichier à notre GIT

7

```
C:\laragon\www\cours-git (master -> origin)  
λ git commit -m "first commit"  
[master (root-commit) 97ab463] first commit  
1 file changed, 1 insertion(+)  
create mode 100644 README.md
```

On commit (commente) notre modification

Processus

Processus et cycle de fonctionnement

8

```
C:\laragon\www\cours-git (master -> origin)  
λ git remote add origin https://SAMPAIO1748/cours-git.git
```

On connecte le projet du serveur avec celui de notre ordinateur

9

```
C:\laragon\www\cours-git (master -> origin)  
λ git push -u origin master  
Enumerating objects: 3, done.  
Counting objects: 100% (3/3), done.  
Writing objects: 100% (3/3), 240 bytes | 48.00 KiB/s, done.  
Total 3 (delta 0), reused 0 (delta 0)  
To https://github.com/jeromediaferia/cours-git.git  
* [new branch]      master -> master  
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

On envoie nos fichiers vers le serveur