



Sommaire

Chapitre 1 – GIT.....	5
1.1 - Présentation.....	6
1.1.1 - Git : les fonctionnalités.....	6
1.1.2 - Clients Git.....	7
1.1.2.1 - Clients console.....	7
1.1.2.2 - Clients graphiques.....	7
1.1.3 - Github, le dépôt distant et public.....	7
1.1.4 - Bibliographie - webographie.....	8
1.1.5 - Divers.....	8
1.2 - Téléchargement et installation sous Linux.....	9
1.3 - Configuration.....	10
1.4 - Créer un dépôt.....	14
1.4.1 - Le projet.....	14
1.4.2 - La description du projet : readme.txt.....	15
1.4.3 - Le wiki.....	16
1.5 - Cloner un dépôt.....	17
1.6 - Ajout et validation des modifications.....	18
1.7 - Annulation d'un commit et autres.....	19
1.7.1 - Annulation du message de modification.....	19
1.7.2 - Annulation dans le tracé des validations.....	20
1.7.3 - Annulation dans le tracé des validations et dans le fichier source....	21
1.7.4 - Annulation d'un add.....	22
1.7.5 - Annulation de toutes les modifications d'un fichier.....	22
1.8 - Communication client-serveur.....	23
1.8.1 - Git sur le « serveur ».....	23
1.8.1.1 - Le protocole local.....	24
1.8.1.2 - Le protocole HTTP.....	25
1.8.2 - Mettre à jour la copie locale.....	26
1.8.3 - Mettre à jour le serveur.....	26
1.8.4 - Annuler une mise à jour serveur.....	26
1.8.5 - Autres.....	26
1.9 - Travailler avec les branches.....	27
1.9.1 - Créer une branche.....	27
1.9.2 - Fusionner une branche.....	27
Chapitre 2 - ANNEXES.....	28
2.1 - Installer GIT sous Windows.....	29
2.1.1 - Téléchargement.....	29

2.1.2 - Installation.....	29
2.1.3 - Après l'installation.....	37
2.1.4 - Connaître la version de Git installée.....	38
2.1.5 - La console	39
2.1.6 - Raccourcis bureau.....	39
2.1.7 - Additionnal informations.....	40
2.2 - Clients git graphiques.....	41
2.2.1 - Clients Git for Windows.....	41
2.2.2 - Autres clients.....	41
2.3 - Git et NetBeans.....	42
2.4 - Historique des gestionnaires de versions.....	43
2.4.1 - Gestion locale : via RCS.....	43
2.4.2 - Gestion centralisée.....	44
2.4.3 - Gestion décentralisée.....	45
2.5 - GitHub, service d'hébergement.....	46
2.5.1 - Présentation.....	46
2.5.2 - Mes comptes GitHub.....	47
2.5.2.1 - Paramètres de mes comptes.....	47
2.5.2.2 - Création d'un compte GitHub.....	48
2.5.3 - Nouveau dépôt.....	49
2.5.4 - Nouveau projet.....	52
2.5.5 - wiki.....	53
2.5.6 - Code / Upload.....	54
2.5.7 - Quelques manipulations.....	58
2.5.7.1 - Création d'un fichier.....	58
2.5.7.2 - Suppression d'un fichier.....	58
2.5.7.3 - Création d'un dossier.....	58
2.5.7.4 - Suppression d'un dossier.....	58
2.6 - GitHubDesktop.....	59
2.6.1 - Présentation.....	59
2.6.2 - IHM.....	60
2.6.3 - Clones créés :.....	61
2.7 - Git Bash.....	63
2.8 - Résumé.....	64
2.8.1 - Git, GitBash, GitHub, GitHubDesktop.....	64
2.8.2 - Démarche.....	64
2.8.3 - Résumé des principales commandes Git.....	65
Chapitre 3 - EXEMPLES.....	66
3.1 - Projet QCM 2019.....	67
3.1.1 - Contexte.....	67
3.1.2 - Étape 1.....	68

3.1.3 - Étape 2.....	72
3.1.4 - Un autre développeur doit travailler avec moi sur le projet.....	74
3.1.5 - Divers.....	75

CHAPITRE 1 – GIT

1.1 - PRÉSENTATION

Pour travailler avec Git il faut 3 éléments :

- ✓ le logiciel Git (le gestionnaire de projets décentralisé),
- ✓ un dépôt (un serveur local ou un serveur distant),
- ✓ un client.

1.1.1 - Git : les fonctionnalités

Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, créateur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2.

Il est disponible pour Linux, Windows, Mac et Solaris.

Il permet de stocker différentes versions du code source.

Il permet un travail collaboratif en facilitant la synchronisation entre différents développeurs.

Il permet de savoir qui est à l'origine d'une modification.

Le dépôt, puisque local et autonome, est toujours accessible et extrêmement rapide, même lorsque l'utilisateur est déconnecté du réseau. Git ne stocke pas les fichiers, mais les différences entre les fichiers : ce principe lui permet de disposer d'un des mécanismes de fusion (merge) des plus efficaces et léger.

Contrôle total sur le dépôt : on ne dépend plus d'une entité externe.

Un système de branchage extrêmement efficace.

1.1.2 - Clients Git

1.1.2.1 - Clients console

git ...

Cf sections 1.3 et suivantes.

1.1.2.2 - Clients graphiques

Cf section 2.2 (GitHub Desktop pour Windows par exemple).

1.1.3 - Github, le dépôt distant et public

Github est un service d'hébergement de projets gérés avec le langage Git.
Cf la section 2.5 pour la présentation et la création d'un compte.

1.1.4 - Bibliographie - webographie

<http://git-scm.com/book/fr/D%C3%A9marrage-rapide>

<http://doc.ubuntu-fr.org/git>

<http://fr.openclassrooms.com/informatique/cours/gerez-vos-codes-source-avec-git/installer-et-configurer-git>

<https://github.com/msysgit/msysgit/wiki>

1.1.5 - Divers

<https://github.com/>

1.2 - TÉLÉCHARGEMENT ET INSTALLATION SOUS LINUX

Téléchargez et installez GIT et une interface graphique pour les logs.

```
| sudo apt-get install git-core gitk
```

ou

```
| sudo apt-get install git
```

Connaître la version de Git installée :

```
| git --version
```

1.3 - CONFIGURATION

La syntaxe pour modifier un élément de configuration est la suivante :

```
git config --global paragraphe.option valeur
```

|

Vous devez créer un user et renseigner son e-mail.

Le nom du user.

```
git config --global user.name "nom"
```

```
| git config --global user.name "p"
```

L'e-mail du user.

```
git config --global user.email email@fournisseur
```

```
| git config --global user.email pascal.buquet@laposte.net
```

L'éditeur (facultatif, par défaut c'est nano sous Ubuntu).

```
git config --global core.editor editeur
```

```
| git config --global core.editor vim
```

Note : cette commande ne sera pas exécutée, nano sera utilisé.

Vous pouvez, c'est facultatif, créer des raccourcis pour la console de GIT.

```
git config --global alias.co checkout
git config --global alias.br branch
git config --global alias.st status
git config --global alias.a add
git config --global alias.ci commit
```

Vous pouvez, c'est facultatif, paramétrer les couleurs de la console de GIT.

```
git config --global color.diff auto
git config --global color.status auto
git config --global color.branch auto
```

git.odt

Afficher la configuration

```
git config --list
```

Afficher la valeur d'un paramètre

```
git config paramètre
```

Il est aussi possible de passer par l'édition du fichier de configuration (c'est un fichier caché) nommé .gitconfig situé dans votre dossier personnel.

```
[color]
    diff = auto
    status = auto
    branch = auto
[user]
    email = pascal.buguet@laposte.net
    name = p
[alias]
    ci = commit
    co = checkout
    st = status
    br = branch
    a = add
[core]
    editor = vim
```

Note : la ligne editor = vim ne sera pas créée, car nous utiliserons l'éditeur texte standard d'Ubuntu qui est nano.

Sous Windows il est dans c:\Utilisateurs\pasca\ et contient :

```
[filter "lfs"]
    clean = git-lfs clean -- %f
    smudge = git-lfs smudge -- %f
    process = git-lfs filter-process
    required = true
[user]
    name = pascalbuguet
    email = pascal.buguet@laposte.net
```

1.4 - CRÉER UN DÉPÔT

1.4.1 - Le projet

Vous devez créer un nouveau dépôt ou clonez un dépôt existant (cf paragraphe suivant).

Pour cela créez un nouveau dossier dans votre arborescence et lancez la commande suivante :

```
git init
```

```
mkdir git_depot_1  
cd git_depot_1  
git init
```

Résultat de la dernière commande :

```
Dépôt Git vide initialisé dans /home/pascal/git_depot_1/.git/
```

Dans le dossier ainsi créé un dossier caché .git a été créé, c'est dans ce dossier que Git stockera les différentes révisions et informations du projet. Cela crée le squelette d'une arborescence git.

Note : si vous voulez créer votre dépôt git dans un dossier de projet existant exécutez les commandes suivantes :

```
cd /projet_existant  
git init
```

et ajoutez ensuite les fichiers au dépôt, par exemple :

```
git add README.TXT  
git add *.java  
git commit -m 'Version initiale du projet'
```

1.4.2 - La description du projet : readme.txt

Two common ways to document a project are **README** files and **wikis**: README files are a quick and simple way for other users to learn more about your work.

Wikis on GitHub help you present in-depth information about your project in a useful way.

<https://guides.github.com/features/wikis/>

Project name: Your project's name is the first thing people will see upon scrolling down to your README, and is included upon creation of your README file.

Description: A description of your project follows. A good description is clear, short, and to the point. Describe the importance of your project, and what it does.

Table of Contents: Optionally, include a table of contents in order to allow other people to quickly navigate especially long or detailed READMEs.

Installation: Installation is the next section in an effective README. Tell other users how to install your project locally. Optionally, include a gif to make the process even more clear for other people.

Usage: The next section is usage, in which you instruct other people on how to use your project after they've installed it. This would also be a good place to include screenshots of your project in action.

Contributing: Larger projects often have sections on contributing to their project, in which contribution instructions are outlined. Sometimes, this is a separate file. If you have specific contribution preferences, explain them so that other developers know how to best contribute to your work. To learn more about how to help others contribute, check out the guide for setting guidelines for repository contributors.

Credits: Include a section for credits in order to highlight and link to the authors of your project.

1.4.3 - *Le wiki*

Possibilité de créer un WIKI du projet.

1.5 - CLONER UN DÉPÔT

Ceci permet de copier un dépôt déjà existant sur un autre serveur.

```
protocole://uri
```

```
git://uri
```

Placez-vous dans un dossier

```
mkdir git_depot_1  
cd git_depot_1  
git://github.com/schacon/grit.git
```

```
git://github.com/pascalbuguet/depot1
```

La dernière commande clone le dépôt situé à l'URI mentionné via le protocole git.

1.6 - AJOUT ET VALIDATION DES MODIFICATIONS

Affichage de l'état des changements

```
git status
```

Visualisation des différentes versions

```
git diff
```

Ajout d'un fichier dans le dépôt en attente de validation

```
git add nom_du_fichier_valider [autre_fichier]
```

Validation

```
git commit -a
```

ou

```
$git commit fichier1 [fichier2]
```

Note : à la suite d'un commit le fichier COMMIT_EDITMSG est ouvert dans nano. Vous devez saisir un commentaire à la première ligne du fichier.

Le commit est local.

1.7 - ANNULATION D'UN COMMIT ET AUTRES

Il existe de nombreuses possibilités pour annuler un commit :

1. annuler un commentaire,
2. annuler une validation,
3. annuler une validation et la modification dans le fichier source.

Et aussi :

1. annuler un add,
2. annuler une modification.

1.7.1 - Annulation du message de modification

```
git commit --amend
```

ceci ouvre le fichier avec nano pour que vous puissiez modifier le message du fichier COMMIT_EDITMSG.

1.7.2 - Annulation dans le tracé des validations

Annulation depuis le dernier commit

```
git reset HEAD
```

Annulation du dernier commit

```
git reset HEAD^
```

Annulation de l'avant-dernier commit

```
git reset HEAD^^
```

idem

```
git reset HEAD~2
```

Annulation jusqu'à un commit référencé

```
git reset valeur_hexa
```

1.7.3 - Annulation dans le tracé des validations et dans le fichier source

Annulation depuis le dernier commit

```
git reset --hard HEAD
```

Annulation du dernier commit

```
git reset --hard HEAD^
```

etc.

1.7.4 - Annulation d'un add

```
git reset HEAD -- fichier_a_retirer
```

Note : il y a bien un espace avant et après les --.

1.7.5 - Annulation de toutes les modifications d'un fichier

Annulation des modifications dans un fichier avant un commit

```
git checkout nom_du_fichier
```

1.8 - COMMUNICATION CLIENT-SERVEUR

1.8.1 - *Git sur le « serveur »*

<http://git-scm.com/book/fr/Git-sur-le-serveur-Protocoles>

Il existe 4 protocoles pour Git :

1. Le protocole local,
2. Le protocole HTTP/S,
3. Le protocole Git,
4. Le protocole SSH.

1.8.1.1 - Le protocole local

Nous allons créer un « dépôt-serveur ».
Pour cela nous créons un dossier nommé `git_depot_distant` et nous clonons le dépôt déjà existant.

```
| mkdir git_depot_distant  
| cd git_depot_distant
```

```
| git clone /home/pascal/git_depot_1
```

`git pull file:///home/pascal/git_depot_distant/git_depot_1`

ou

```
| git clone --bare /home/pascal/git_depot_1 git_depot_distant
```

L'option `--bare` : copie seulement les infos du projet.

1.8.1.2 - Le protocole HTTP

git.odt

1.8.2 - Mettre à jour la copie locale

```
git pull
```

1.8.3 - Mettre à jour le serveur

```
git push
```

1.8.4 - Annuler une mise à jour serveur

Pour annuler une mise-à-jour que vous avez faite sur le serveur.

```
git revert id_en_hexa
```

1.8.5 - Autres

```
log
```

```
git log -p
```

1.9 - TRAVAILLER AVEC LES BRANCHES

1.9.1 - Créer une branche

1.9.2 - Fusionner une branche

CHAPITRE 2 - ANNEXES

2.1 - INSTALLER GIT SOUS WINDOWS

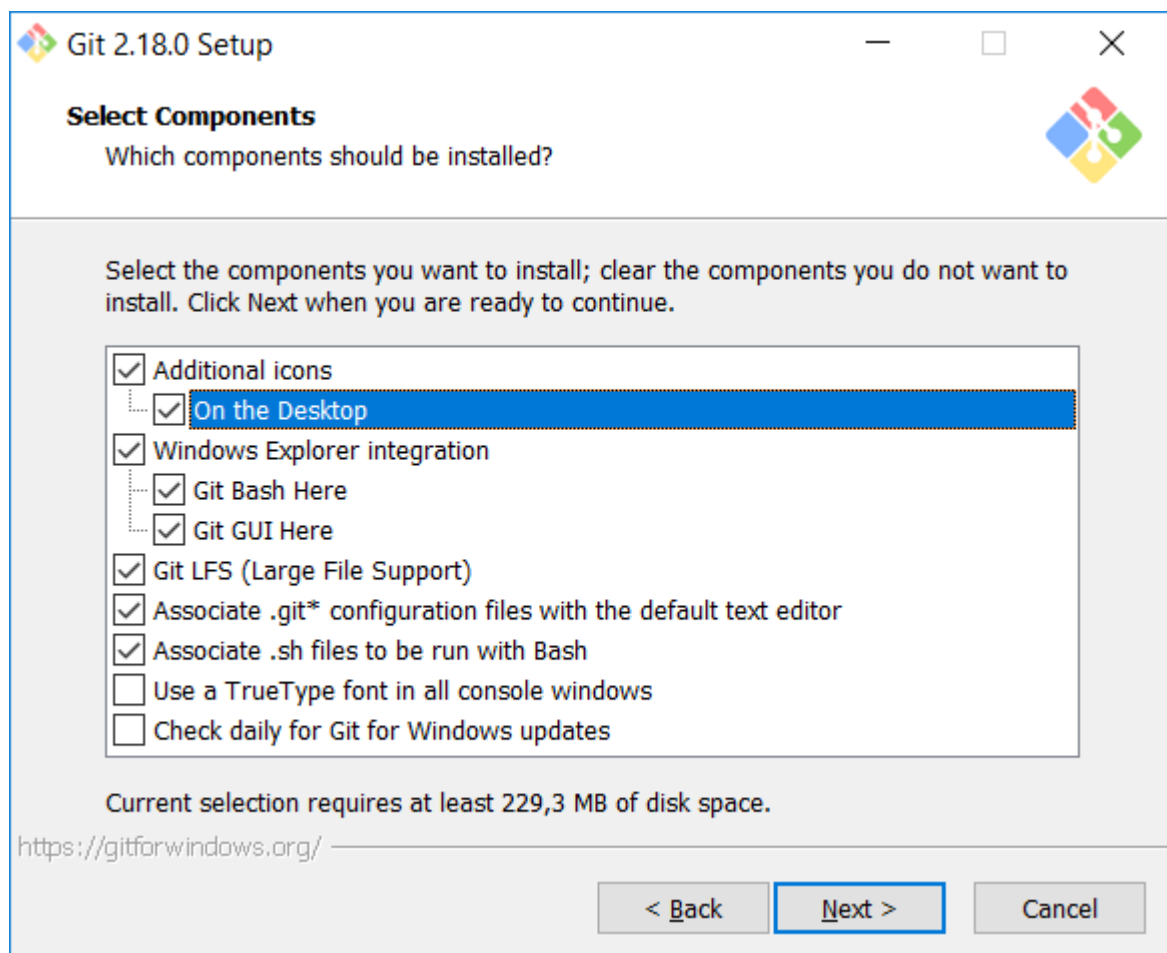
2.1.1 - Téléchargement

<https://gitforwindows.org/>

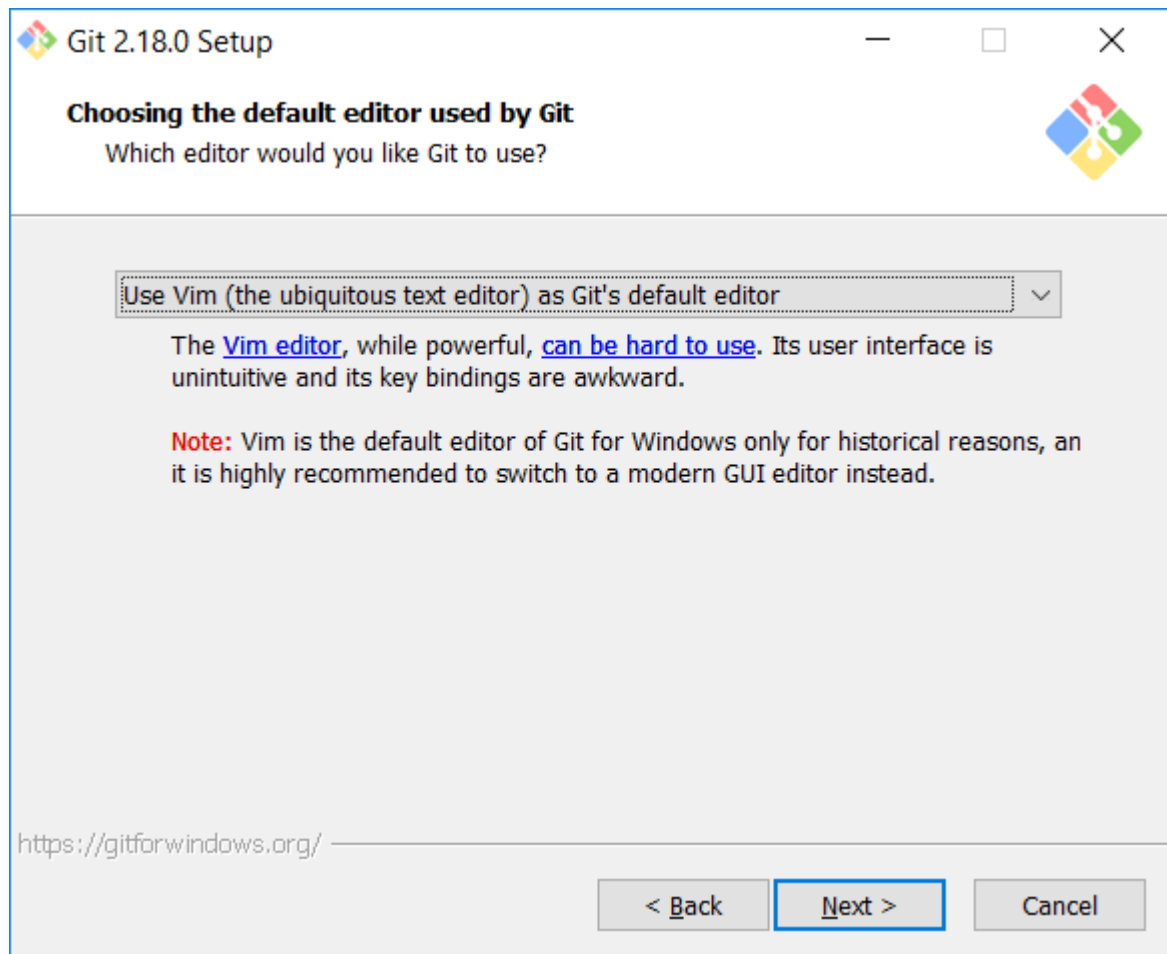
Git-2.18.0-64-bit.exe au 1^{er} juillet 2018.

2.1.2 - Installation

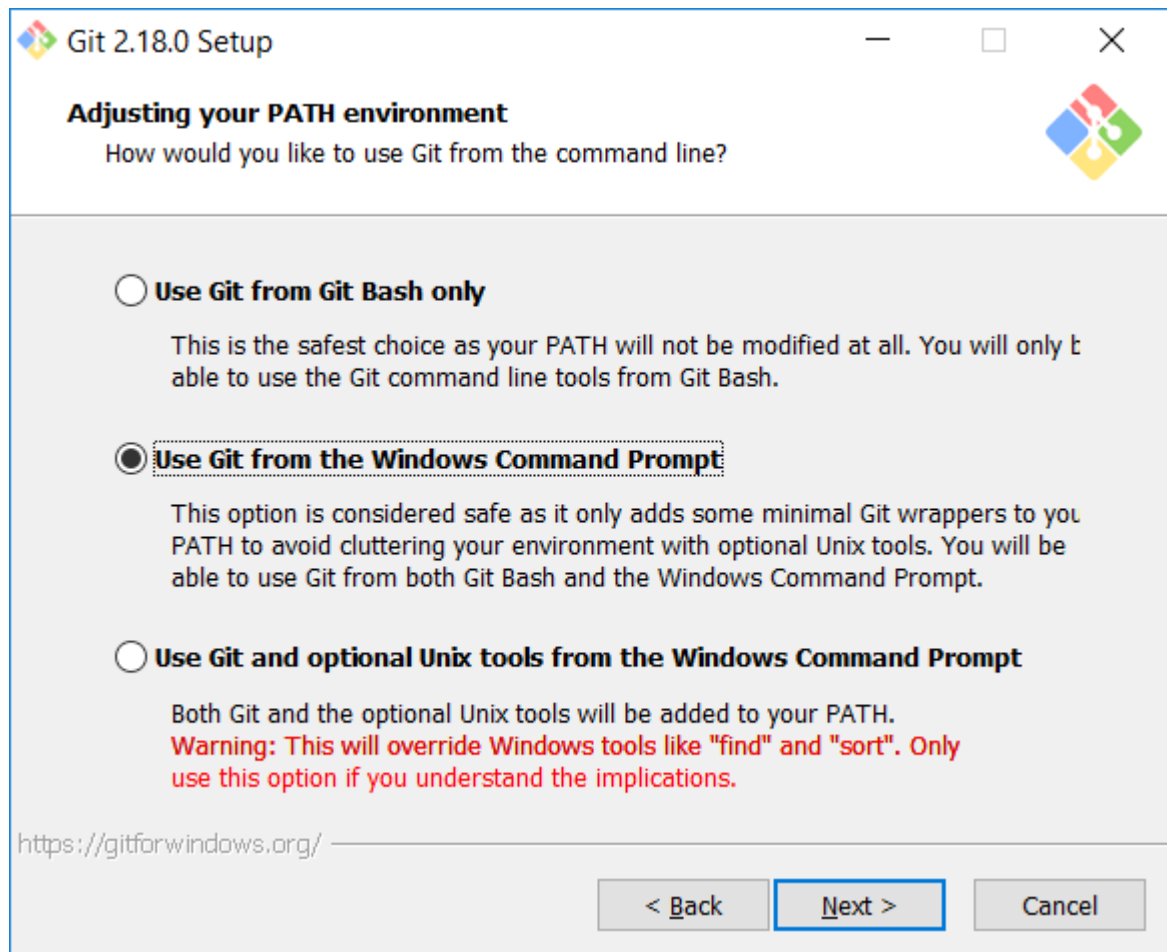
Composants :



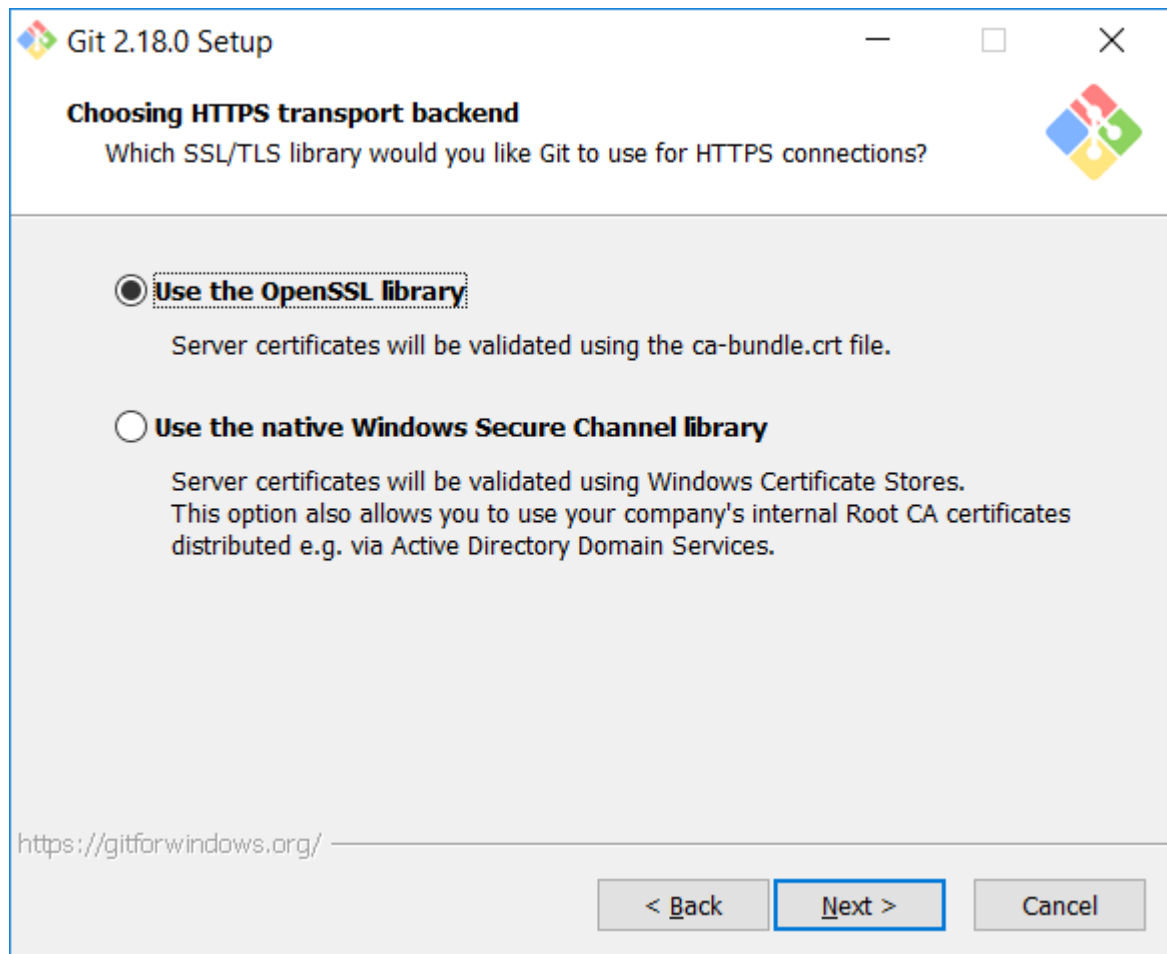
Éditeurs : vim, Notepad++, Visual Studio Code, ...



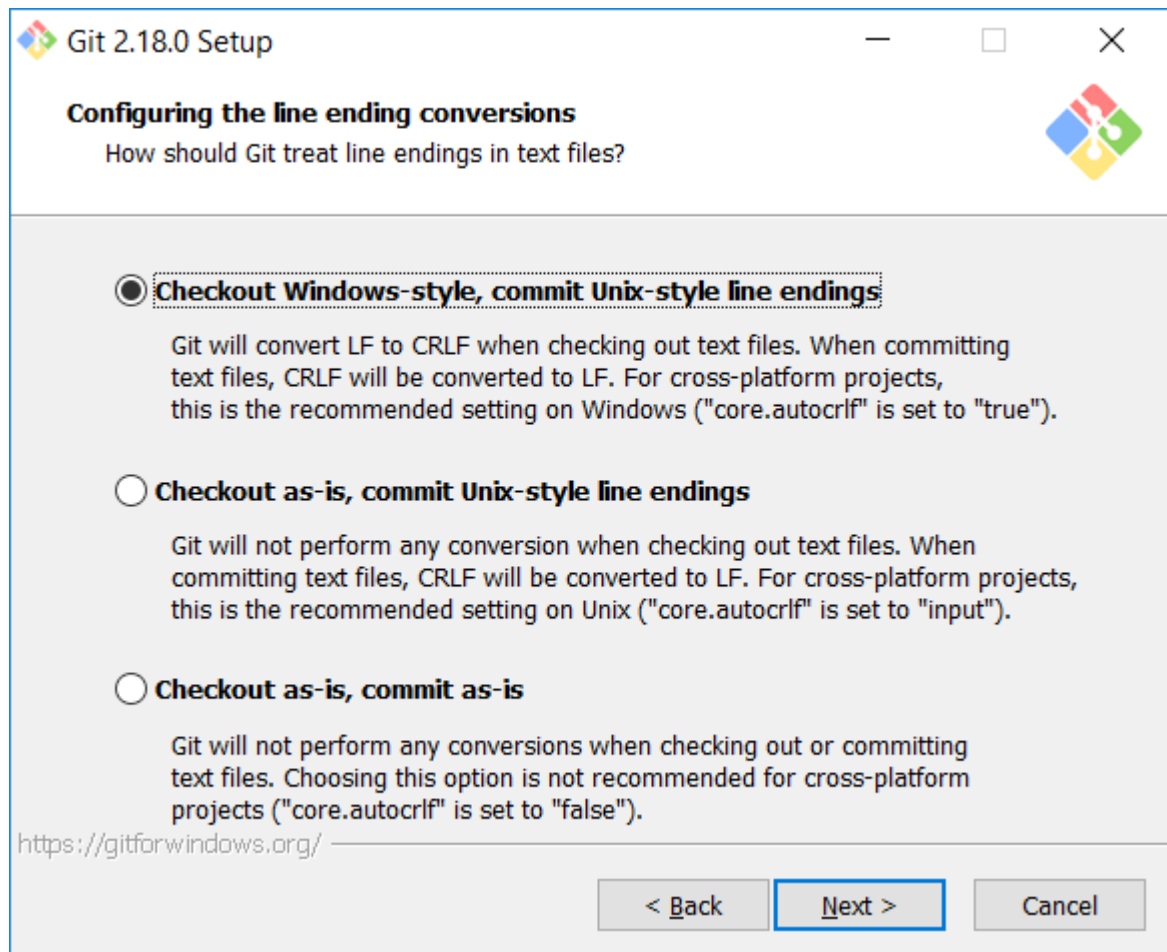
PATH : Using Git from Windows Console ...



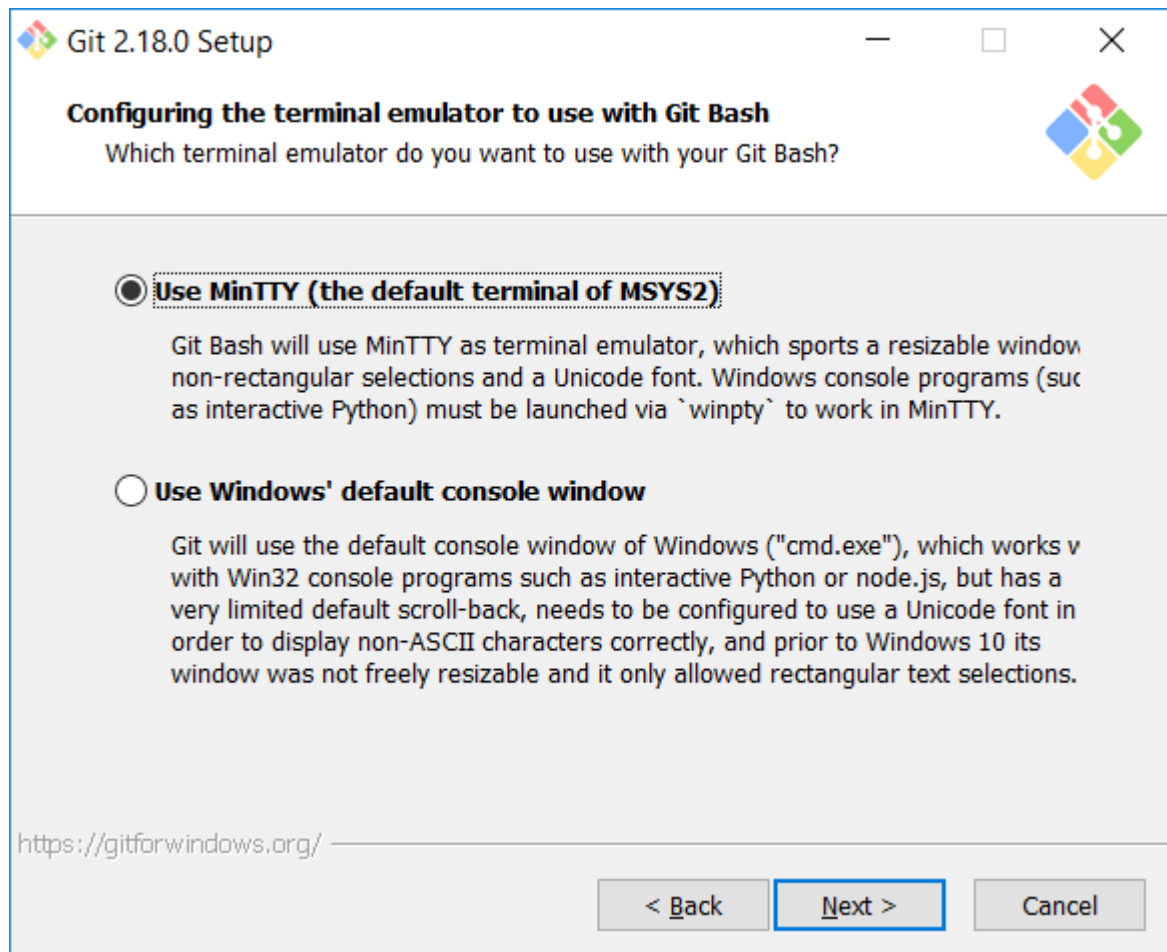
HTTPS transport : Use OpenSSL



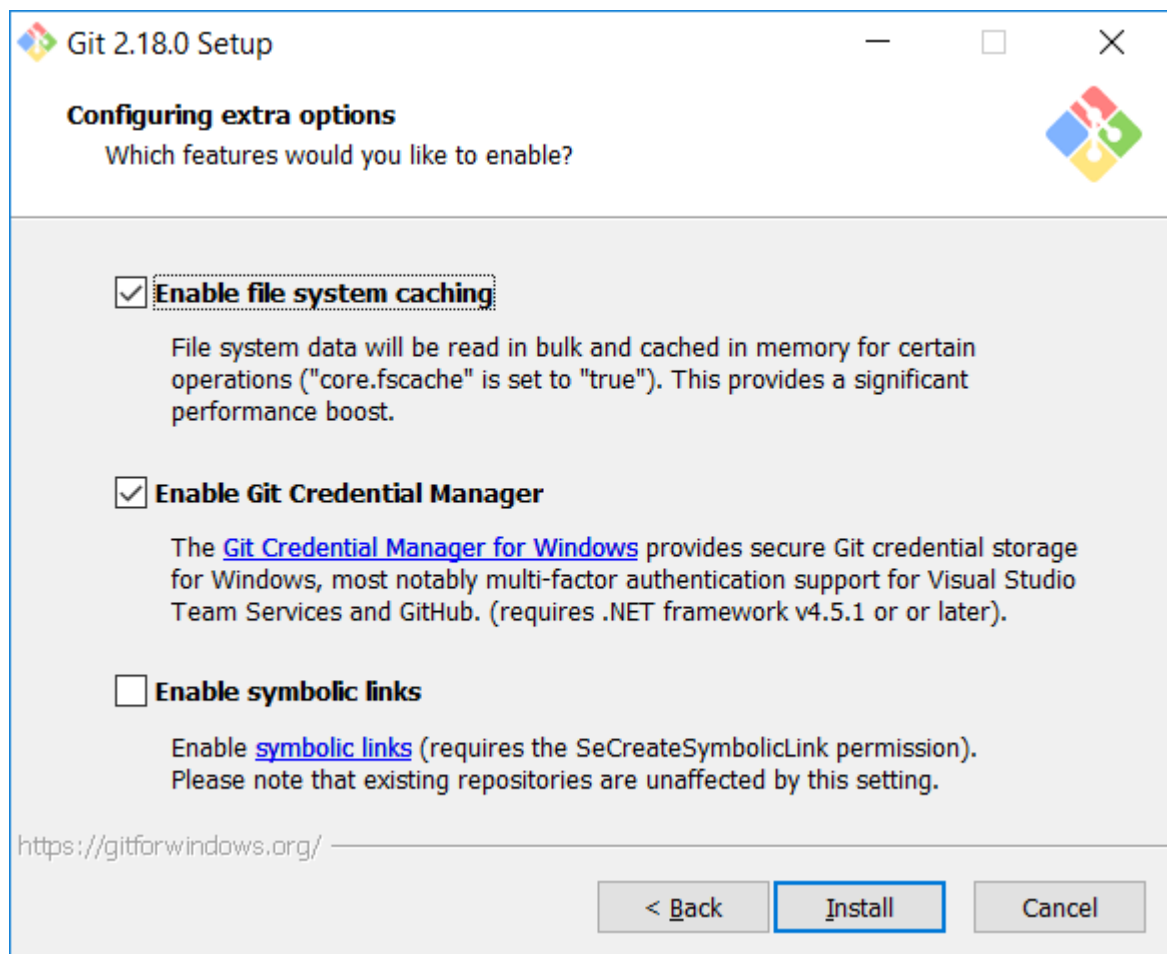
Line ending : Windows Style ...



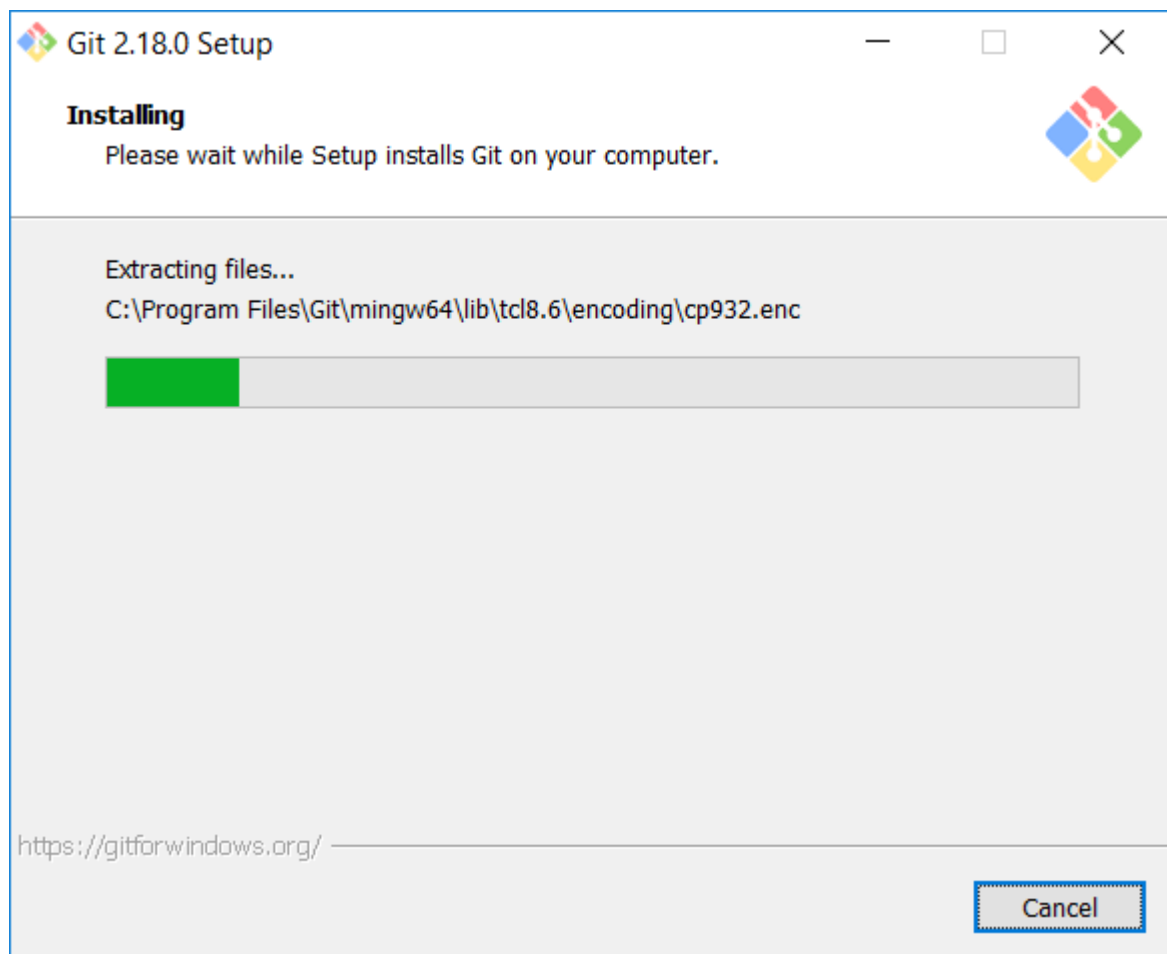
Terminal Emulator : Use Windows console ?



Extra options ?



Installation ...



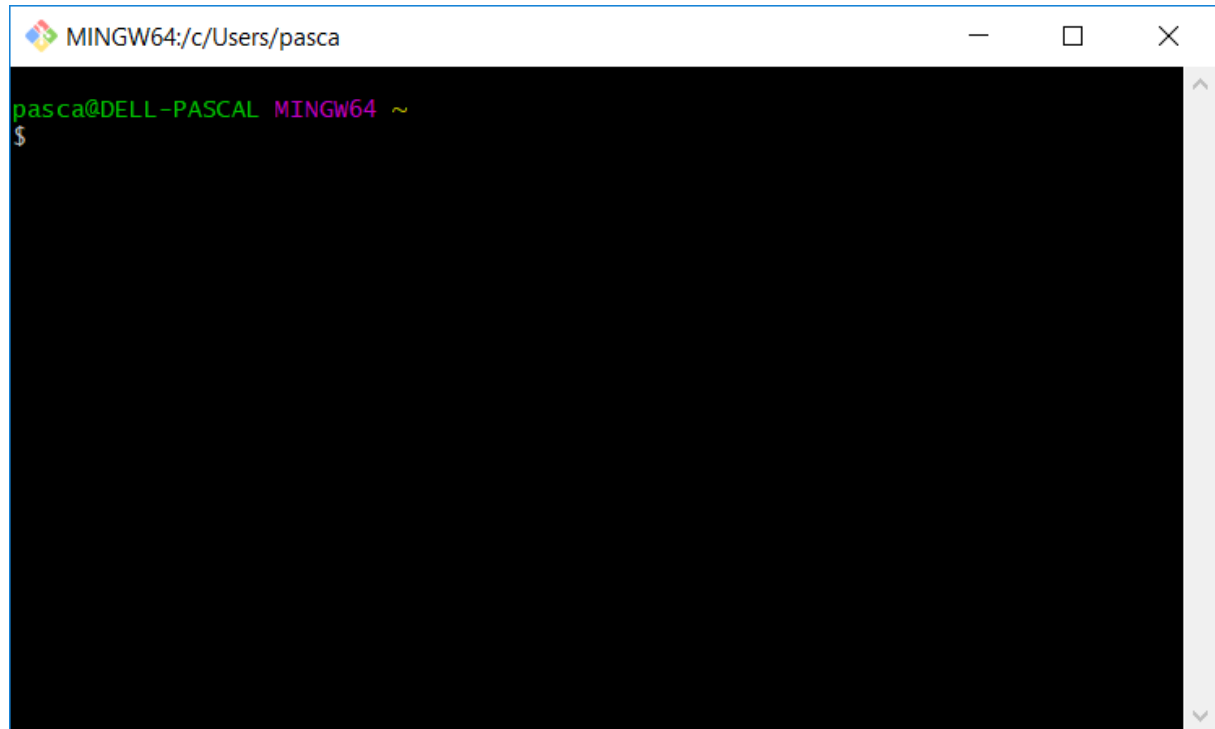
2.1.3 - Après l'installation

```
| git config --global user.email "pascal.buguet@laposte.net"  
| git config --global user.name "Pascal"
```

2.1.4 - Connaître la version de Git installée

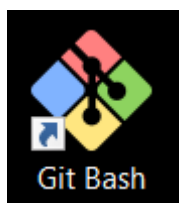
| `git --version`

2.1.5 - La console ...



2.1.6 - Raccourcis bureau

"C:\Program Files\Git\git-bash.exe" -cd-to-home
Démarré dans %HOMEDRIVE%%HOMEPATH%



et/ou



2.1.7 - Additionnal informations

<file:///C:/Program%20Files/Git/ReleaseNotes.html>

Cela installe **msys** (un système d'émulation des commandes Unix sous Windows) et Git simultanément.

Cf aussi TortoiseGIT, l'interface graphique pour Windows.

Comme c'est une émulation c'est un peu lent et il est recommandé d'utiliser Mercurial qui possède une belle interface graphique.

2.2 - CLIENTS GIT GRAPHIQUES

2.2.1 - Clients Git for Windows

<https://git-scm.com/download/gui/windows>

Choix au 1^{er} juillet 2018 : **GitHubDesktop**.

<https://desktop.github.com/>

Téléchargez GitHubDesktopSetup.exe.
Installez-le.

2.2.2 - Autres clients

<http://git-scm.com/downloads/guis>

Le tableau ci-dessous est issu des informations de l'URL ci-dessus,
d'autres produits payants présentés existent.

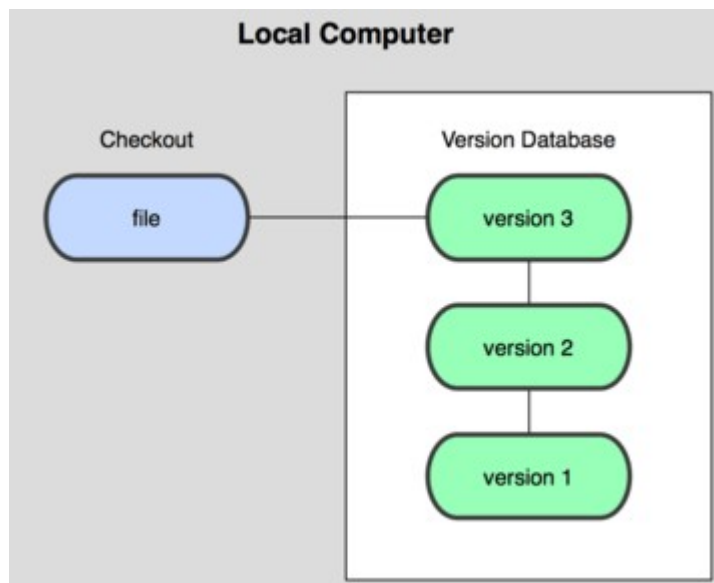
Produit	Système	Prix
GitHub for Windows https://windows.github.com/	Windows	Free
Git Extensions https://code.google.com/p/gitextensions/	Windows	Free
GitHub for Mac https://mac.github.com/	iOS	Free
Giggle https://wiki.gnome.org/Apps/giggle/	Linux	Free
Gitg https://wiki.gnome.org/Apps/Gitg/	Linux	Free
Git-cola http://git-cola.github.io/	Linux, Mac, Windows	Free
GitEye http://www.collab.net/products/giteye	Linux, Mac, Windows	Free

2.3 - GIT ET NETBEANS

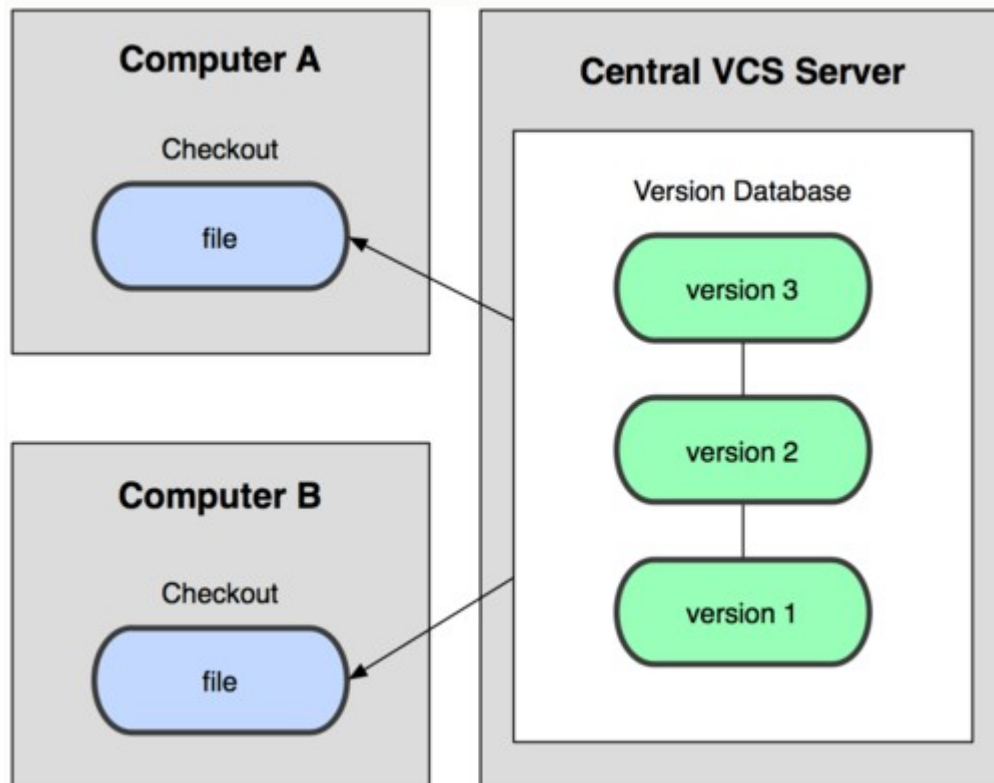
2.4 - HISTORIQUE DES GESTIONNAIRES DE VERSIONS

Source : <http://git-scm.com/book/fr/D%C3%A9marrage-rapide-%C3%80-propos-de-la-gestion-de-version>

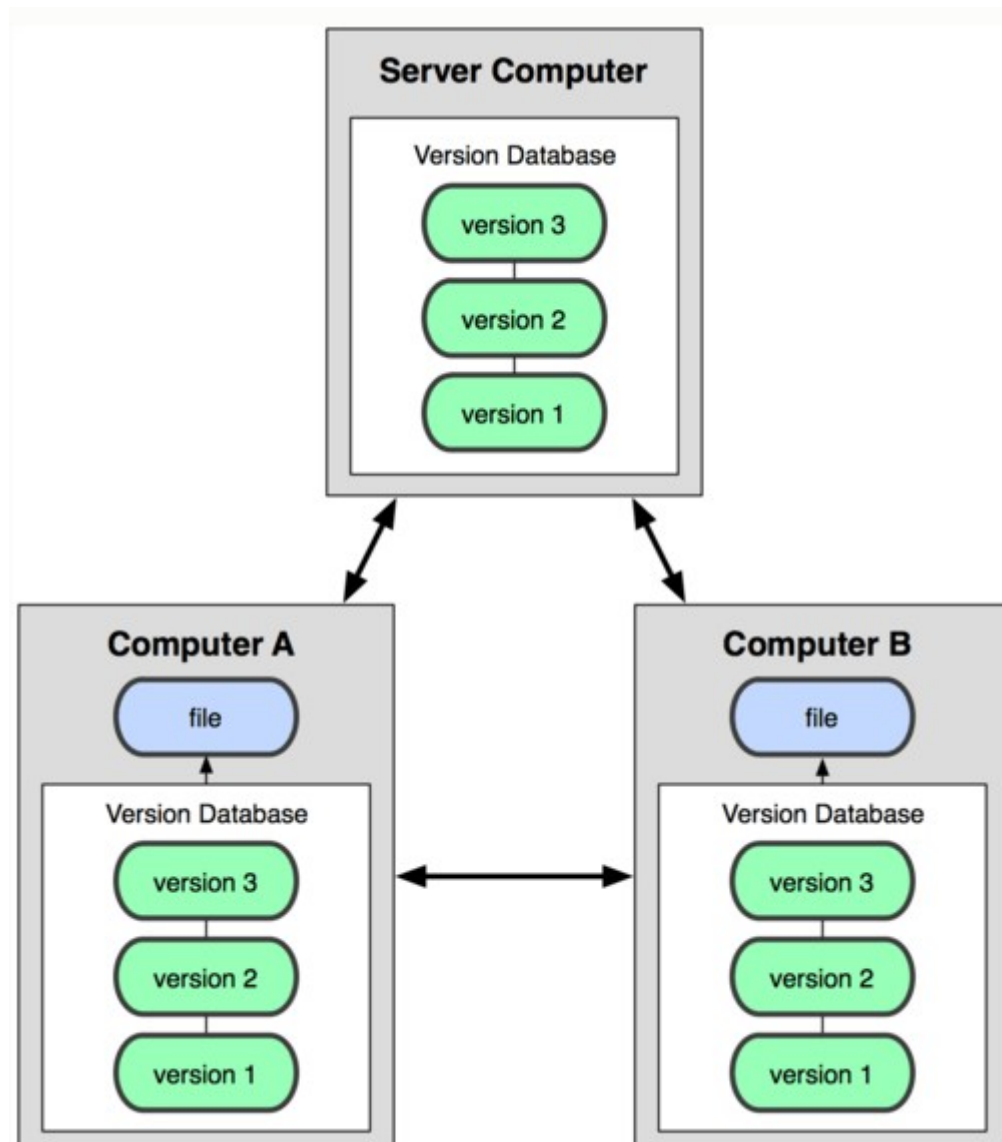
2.4.1 - Gestion locale : via RCS



2.4.2 - Gestion centralisée



2.4.3 - Gestion décentralisée



2.5 - GITHUB, SERVICE D'HÉBERGEMENT

2.5.1 - Présentation

Issue de Wikipedia et autres.

« GitHub (exploité sous le nom de GitHub, Inc.) est un **service web d'hébergement** et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git. Ce site est développé en Ruby on Rails et Erlang par Chris Wanstrath, PJ Hyett et Tom Preston-Werner. GitHub propose des comptes professionnels payants, ainsi que des comptes gratuits pour les projets de logiciels libres.

Le nom GitHub est composé du mot « git » faisant référence à un système de contrôle de version open-source et le mot « hub » faisant référence au réseau social bâti autour du système Git.

GitHub est centré vers l'aspect social du développement. En plus d'offrir l'hébergement de projets avec Git, le site offre de nombreuses fonctionnalités habituellement retrouvées sur les réseaux sociaux comme les flux, la possibilité de suivre des personnes ou des projets ainsi que des graphes de réseaux pour les dépôts. GitHub offre aussi la possibilité de créer un wiki et une page web pour chaque dépôt. Le site offre aussi un logiciel de suivi de problèmes.

Alors que le système traditionnel open source amène chaque contributeur à télécharger les sources du projet et à proposer ensuite ses modifications à l'équipe du projet, GitHub repose sur le principe du fork par défaut : toute personne « forkant » le projet devient publiquement de facto le leader de son projet portant le même nom que l'original. »

GitHub appartient à Microsoft.

Statistiques (début 2019) :

31 millions d'utilisateurs.
100 millions de projets.

git.odt

2.5.2 - Mes comptes GitHub

2.5.2.1 - Paramètres de mes comptes

[https://github.com/pascalbuguet/
pascalbuguet/Gri...1](https://github.com/pascalbuguet/pascalbuguet/Gri...1)
pascal.buguet@laposte.net

Autre compte pour tests à « plusieurs ».
[https://github.com/pb-2020/
pb-2020/Gri...1](https://github.com/pb-2020/pb-2020/Gri...1)
borisespion@gmail.com

2.5.2.2 - Création d'un compte GitHub

Allez à <https://github.com/>

Cliquez sur « Sign up » (s'inscrire)

Join GitHub

Create your account

Username *

Email address *

Password *

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter.
[Learn more.](#)

Email preferences

☐ Send me occasional product updates, announcements, and offers.


Verify your account

Verify ... Résoudre le puzzle ...

2.5.3 - Nouveau dépôt

Vous êtes dans la page d'accueil de votre GitHub,
Sélectionnez l'item « Repositories »,
Cliquez sur le bouton « New ».

Overview **Repositories 4** Projects 0 Stars 0 Followers 0 Following 0

Type: All ▾ Language: All ▾  **New**


<https://github.com/new>

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

Repository name

 **pascalbuguet ▾** /

Great repository names are short and memorable. Need inspiration? How about **potential-tribble**.

Description (optional)

☒  **Public**
Anyone can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None ▾** | Add a license: **None ▾** 

Create a new repository

A repository contains all the files for your project, including the revision history.



Owner

Repository name

 **pascalbuguet** ▾ / 

Great repository names are short and memorable. Need inspiration? How about **potential-tribble**.

Description (optional)

- ☒  **Public**
Anyone can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾ | Add a license: **None** ▾ 

Create repository

Note : private est payant (≥ 7 \$ par mois).

Après validation ...

pascalbuguet / depot1

Unwatch

1

Star

0

Fork

0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Settings

Insights

Première expérience

Edit

Add topics

1 commit

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

pascalbuguet committed on GitHub Initial commit

Latest commit c53f9b6 just now

README.md

Initial commit

just now

README.md

depot1

Première expérience

2.5.4 - Nouveau projet

pascalbuguet / depot1

Unwatch 1

Star 0

Fork 0

<> Code

! Issues 0

🔗 Pull requests 0

📁 Projects 1

📖 Wiki

⚙ Settings

🔍 Insights ▾

New Project

pascalbuguet / depot1

<> Code

! Issues 0

🔗 Pull requests 0

📁 Projects 0


Create a new project

Name

Description


Save project

2.5.5 - wiki

 [pascalbuguet](#) / [depot1](#)

[Unwatch](#) [1](#) [★ Star](#) [0](#) [Fork](#)

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 1](#) [Wiki](#) [Settings](#) [Insights](#)



Welcome to the depot1 wiki!

Wikis provide a place in your repository to lay out the roadmap of your project, show the current status, and document software better, together.

[Create the first page](#)

2.5.6 - Code / Upload

<> Code

! Issues 0

🔗 Pull requests 0

📁 Projects 0

📖 Wiki

⚙ Settings

🔍 Insights ▾

Première expérience

Edit

[Add topics](#)

📁 1 commit

🌿 1 branch

📦 0 releases

👤 1 contributor

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾

pascalbuguet committed on **GitHub** Initial commit

Latest commit c53f9b6 14 minutes ago

[README.md](#)

Initial commit

14 minutes ago

README.md

depot1

Première expérience

pascalbuguet / depot1

Unwatch 1

<> Code

Issues 0

Pull requests 0


Projects 1

Wiki

Settings


Insights

depot1 /



Drag files here to add them to your repository

Or [choose your files](#)



Commit changes

Add files via upload

 /5MinutesDoucheCompriseDesktop/build.xml



 /5MinutesDoucheCompriseDesktop/dist/5MinutesDoucheCompriseDesktop.jar



Commit changes

Upload1|

Add an optional extended description...

- ☒  Commit directly to the `master` branch.
- ☐  Create a **new branch** for this commit and start a pull request. [Learn](#)


Commit changes

Cancel

Commit ...

git.odt

Le résultat du commit ...

 **pascalbuguet / depot1** Unwatch


<> Code Issues 0 Pull requests 0 Projects 1 Wiki Settings Insights


Première expérience


[Add topics](#)


2 commits 1 branch 0 releases

Branch: master ▼ New pull request Create new file Upload files

 **pascalbuguet** committed on **GitHub** Upload1 Late

 [5MinutesDoucheCompriseDesktop](#) Upload1

 [README.md](#) Initial commit

 **README.md**

Page 57 sur 75

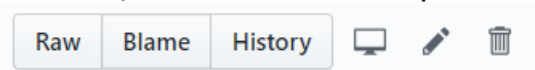
2.5.7 - Quelques manipulations

2.5.7.1 - Création d'un fichier

Bouton « Create » new file » ... un éditeur est affiché, ... à la fin il faut cliquer sur le bouton « Commit ».

2.5.7.2 - Suppression d'un fichier

Il faut cliquer sur le nom du fichier (lien), le fichier est ouvert dans l'éditeur, il faut ensuite cliquer sur le bouton supprimer



et enfin cliquer sur le bouton Commit.

2.5.7.3 - Création d'un dossier

Il faut créer un nouveau fichier en le « préfixant » par un nom de dossier. Un exemple pour créer le dossier /libs/ :

qcm_web_2019 / libs / or cancel

Puis un commit.

2.5.7.4 - Suppression d'un dossier

```
git rm -r one-of-the-directories
git commit -m "Remove duplicated directory"
git push origin <your-git-branch> (typically 'master', but not always)
```

ou (mais ce n'est pas terrible!)

Supprimer tous les fichiers du dossier !!!

2.6 - GITHUBDESKTOP

2.6.1 - Présentation

<https://desktop.github.com/>

Note : New in 2019 : Extend your GitHub workflow beyond your browser with GitHub Desktop, completely redesigned with Electron. Get a unified cross-platform experience that's completely open source and ready to customize. (TODO in future!).

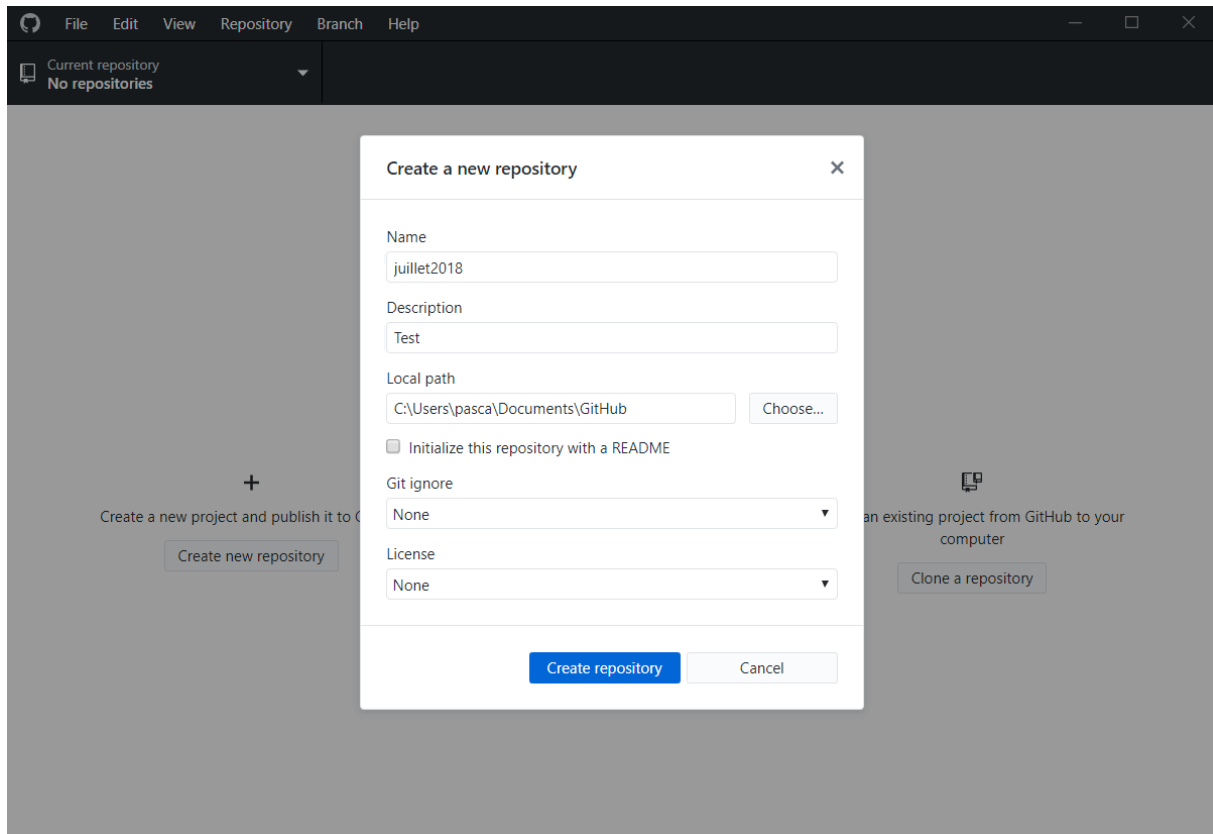


Téléchargez-le et installez-le.

git.odt

2.6.2 - IHM

C:\Users\pasca\AppData\Local\GitHubDesktop\GitHubDesktop.exe



Repository ?

C:\Users\pasca\Documents\GitHub\juillet2018>

Contient 1 fichier nommé Fichier1.txt

Provient d'où ? D'un clonage ?

2.6.3 - Clones créés :

Sur Lenovo II Windows :

| \pascal\supports\git\projets_git

Sur Desktop kviteko :

|

Items de menu :

File

- New Repository
- Add local Repository
- Clone Repository
- Options
- Exit

Edit

- Undo
- Redo
- Cut
- Copy
- Paste
- Select All

View

- Change
- History
- Repository list
- Branches list
- Toogle full screen
- Zoom ...
- Toogle developper tools

Repository

- Push
- Pull
- Remove
- View on Github
- Open in command prompt
- Show in explorer
- Open in Visual Studio Code
- Repository Settings

Branch

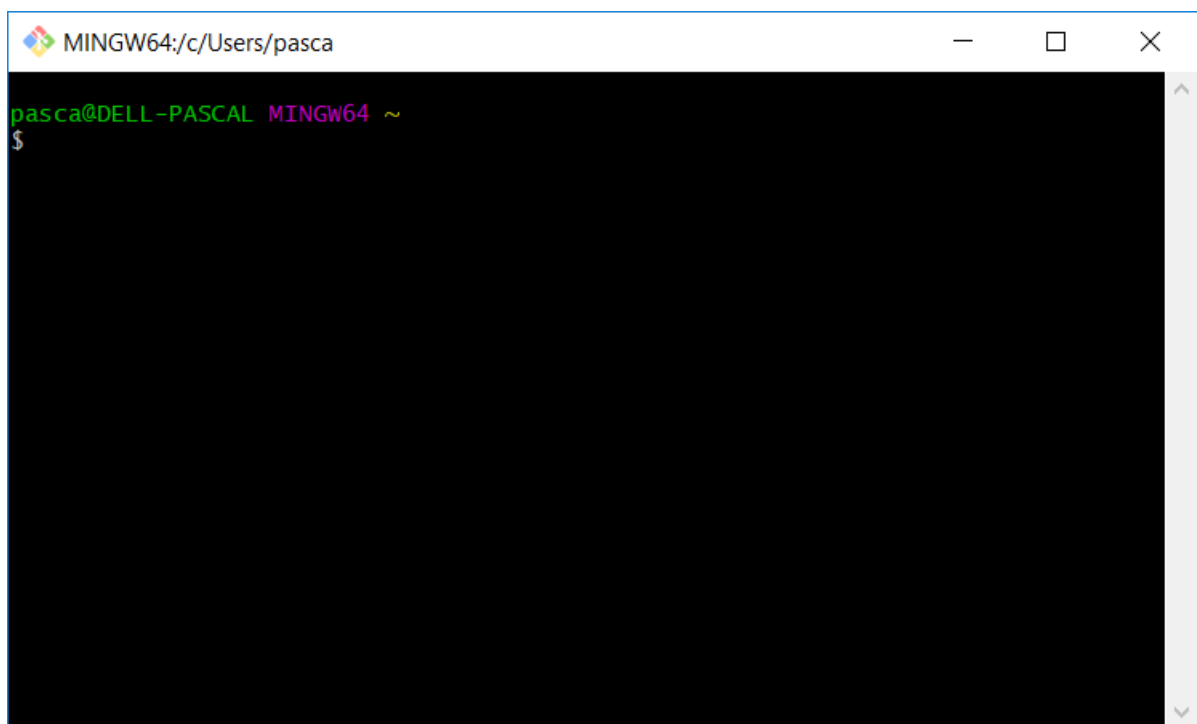
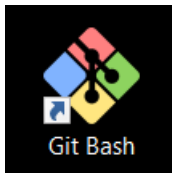
- New branch
- Rename
- Delete
- Update from default branch
- Compare to branch
- Merge into current branch
- Compare on github
- Create new pull request

Help

...

2.7 - GIT BASH

Outil pour Git en mode console avec une émulation Linux.
Il est installé lorsque l'on installe Git.



```
pwd  
/c/Users/pasca
```

Les commandes de base sont celles de Linux : `pwd`, `ls`, `cd`, `cat`, `mkdir`, `rmdir`, ...

Les principales commandes Git : cf.2.8.3.

2.8 - RÉSUMÉ

2.8.1 - *Git, GitBash, GitHub, GitHubDesktop*

Logiciel	Description	Comparaison SQL
Git	Un logiciel, un langage, de gestion de versions	SQL
GitHub	Un « serveur » de dépôts. GitHub (exploité sous le nom de GitHub, Inc.) est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git.	BD MySQL
Git Bash	Console Git en émulation Linux	mysql> client console
GitHub Desktop	Outil graphique pour une gestion de projet avec Git	MySQLQueryBrowser Client lourd

2.8.2 - *Démarche*

Création d'un projet vide avec Git.
On se positionne sur le poste client, dans un dossier quelconque.

2.8.3 - Résumé des principales commandes Git

Objectif	Commande
Connaître la version de Git installée	git --version
Créer un dépôt	git init
Ajouter un ou plusieurs fichiers au dépôt	git add fichier
Cloner un dépôt	git://URIDuDepot
Connaître l'état d'un dépôt	git status
Connaître les différences entre le dépôt et l'état du projet ?	git diff
Valider définitivement les modifications	git commit
Créer une branche ou basculer entre branches	git checkout
Ajouter une nouvelle version d'une ressource	git push
Récupérer une nouvelle version d'une ressource	git pull
Fusionner une branche	git merge
Connaître la configuration du dépôt	git config

cf aussi :

<https://www.hostinger.fr/tutoriels/commandes-git/>

CHAPITRE 3 - EXEMPLES

3.1 - PROJET QCM 2019

3.1.1 - Contexte

Après la création d'un projet (c'est donc la version 1.0.0) je veux le « déposer » sur mon GitHub !

Le projet a été créé sous NetBeans dans
c:/xampp/htdocs/QCM_web_2019

Mon GitHub :

<https://github.com/pascalbuguet/>

User : pascalbuguet

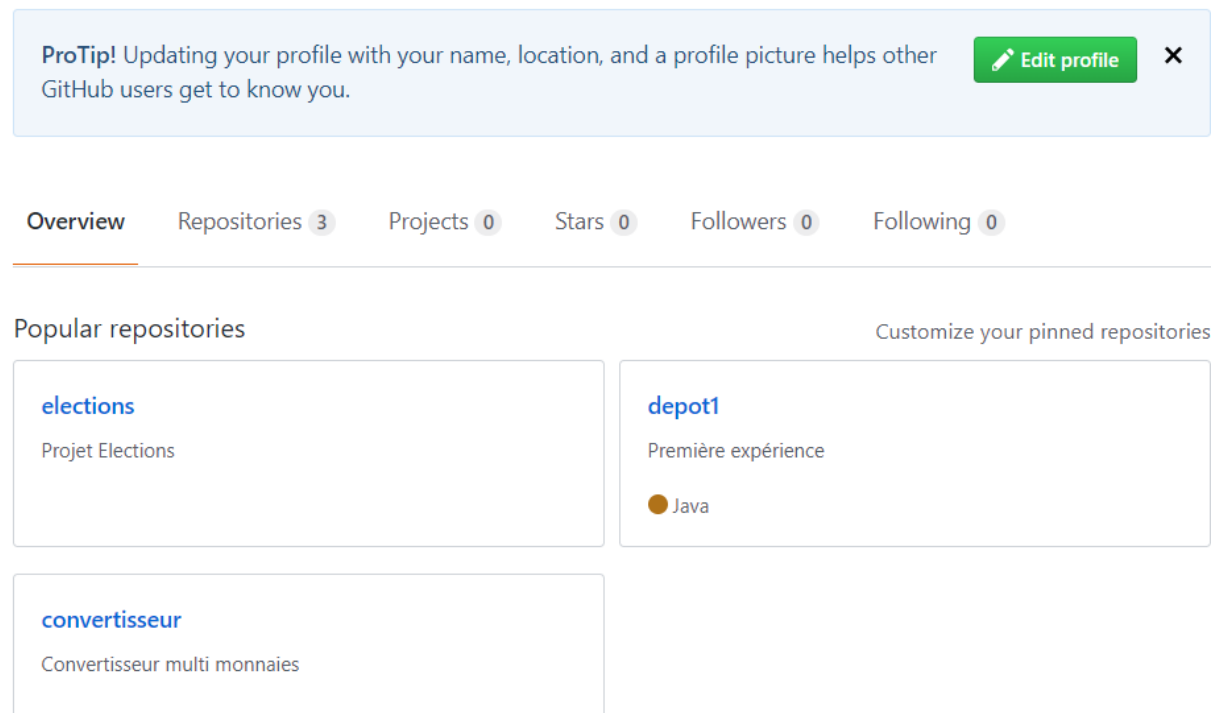
PWD : Gr... 1

E-mail : pascal.buguet@laposte.net

3.1.2 - Étape 1

Je vais donc sur mon Github et je crée un nouveau projet nommé « qcm_web_2019 ».

J'atterris sur cet écran :




Je clique sur l'item de menu « Repository »
Je clique sur le bouton « New ».

Je saisis le nom du projet : « qcm_web_2019 ».

Create a new repository

A repository contains all project files, including the revision history.

Owner

 pascalbuguet ▾

Repository name *

/ qcm_web_2019 ✓

Great repository names are short and memorable. Need inspiration? How about **curly-lamp**?

Description (optional)

Le projet QCM version web en PHP



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

☐ Initialize this repository with a README

Description (optional)

Le projet QCM version web en PHP



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.



Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip

Add .gitignore: None ▾

Add a license: None ▾



Create repository

Je clique sur « Create repository ».

Le résultat : le dépôt est créé, quasiment vide ...

The screenshot shows the GitHub interface for a repository named 'qcm_web_2019' by user 'pascalbuguet'. At the top, there are navigation tabs: 'Code' (selected), 'Issues' (0), 'Pull requests' (0), 'Projects' (0), 'Wiki', 'Insights', and 'Settings'. To the right of the repository name, there are buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). Below the navigation bar, the repository description is 'Le projet QCM version web en PHP' with an 'Edit' button. A 'Manage topics' link is also present. A summary bar shows '1 commit', '1 branch', '0 releases', and '1 contributor'. Below this, there are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and a green 'Clone or download' button. The commit history section shows a single commit by 'pascalbuguet' titled 'Initial commit' with the latest commit hash 'd6b0ad9' and the time 'just now'. Below the commit history, the file list shows 'README.md' as the only file, also with the 'Initial commit' and 'just now' timestamp. The content of the 'README.md' file is displayed below, showing the text 'acm web 2019'.

pascalbuguet / qcm_web_2019

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Le projet QCM version web en PHP Edit

Manage topics

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

pascalbuguet Initial commit Latest commit d6b0ad9 just now

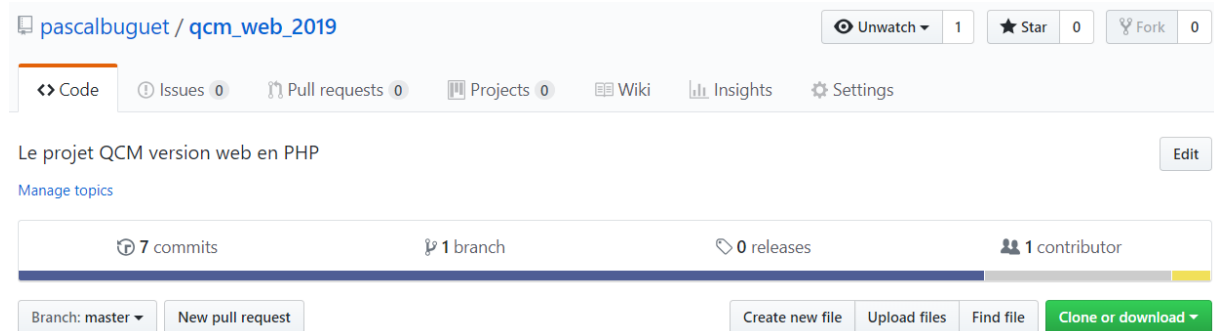
README.md Initial commit just now

README.md

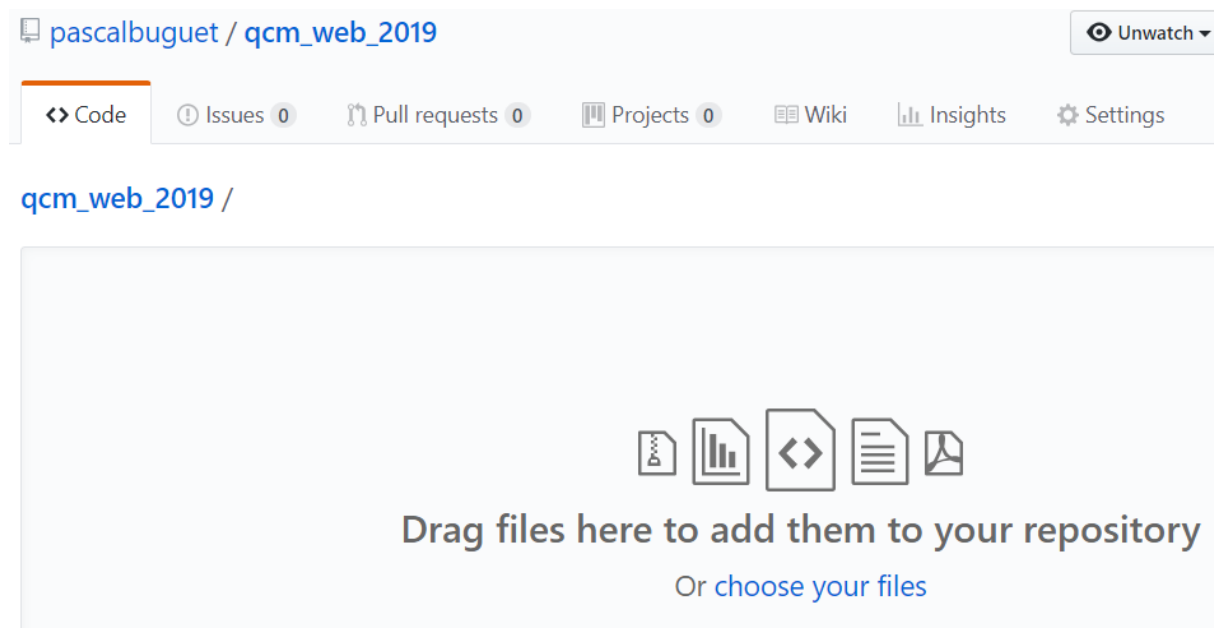
acm web 2019

3.1.3 - Étape 2

Je vais ajouter le projet PHP, les codes ...

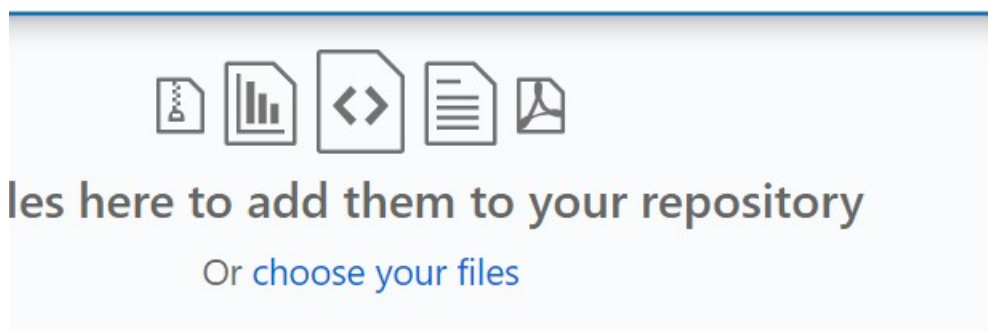


Je clique sur le bouton « Upload files ».



Je bascule dans l'explorateur Windows

css	17/02/2019 07:03
daos	28/02/2019 05:05
documentation	27/02/2019 19:55
essais	21/02/2019 06:04
images	17/02/2019 06:08
js	18/02/2019 05:16
libs	28/02/2019 05:05
nbproject	17/02/2019 05:46



Je « Drag and Drop » un dossier.

3.1.4 - Un autre développeur doit travailler avec moi sur le projet

Il possède un compte GitHub (pb-2020).
Il travaille sur une autre machine.
Il clone le projet ?
Il modifie le projet (il ajoute un fichier).
Il valide les modifications.

3.1.5 - Divers