

Versioning / Travail collaboratif

Partie 0 : Introduction générale



Dr. Hamidou KASSOGUÉ

CFECO
Juin 2018

Logiciels & Plateformes : Git, GitHub, GitKraken



Plan gloabl

- 1 Contexte
- 2 Concepts et fondements de base
 - Généralités sur le versioning
 - Les logiciels et plateformes de versioning
 - Programme et évaluation
- 3 Environnement Git
 - Logiciel Git
 - Installation et configuration

Plan de section

- 1 Contexte
- 2 Concepts et fondements de base
 - Généralités sur le versioning
 - Les logiciels et plateformes de versioning
 - Programme et évaluation
- 3 Environnement Git
 - Logiciel Git
 - Installation et configuration

Questions !

- ❶ Quels IDEs utilisés-vous quand vous développez ?
- ❷ Avez-vous déjà entendu parlé du versioning ? L'utilisez vous ?
- ❸ Comment faites-vous lorsque vous voulez avoir des versions de votre code ?
- ❹ Comment faites-vous pour vous retrouver sur des dizaines de fichiers sources et des milliers de code sources.

Constat

```
14938 private javax.swing.JTextField tf_rc_radius_humidiy;  
14939 private javax.swing.JTextField tf_rc_radius_phys;  
14940 private javax.swing.JTextField tf_rc_radius_state;  
14941 private javax.swing.JTextField tf_rc_radius_temp;  
14942 private javax.swing.JTextField tf_rc_radius_zone;  
14943 private javax.swing.JTextField tf_rc_rain_duration;  
14944 private javax.swing.JTextField tf_rc_rain_intensity;  
14945 private javax.swing.JTextField tf_rc_rain_radius;  
14946 private javax.swing.JTextField tf_rc_rain_total_du;  
14947 private javax.swing.JTextField tf_rc_resistance;  
14948 private javax.swing.JTextField tf_rc_saturation;  
14949 private javax.swing.JTextField tf_rc_size;  
14950 private javax.swing.JTextField tf_rc_solltype;  
14951 private javax.swing.JTextField tf_rc_state_gw;  
14952 private javax.swing.JTextField tf_rc_state_sw;  
14953 private javax.swing.JTextField tf_ref_ew;  
14954 private javax.swing.JTextField tf_ref_gsw;  
14955 private javax.swing.JTextField tf_right_bottom;  
14956 private javax.swing.JTextField tf_te;  
14957 private javax.swing.JTextField tf_test_rain;  
14958 private javax.swing.JTextField tf_ts;  
14959 private javax.swing.JTabbedPane tp_analysis;  
14960 private javax.swing.JTabbedPane tp_monitor;  
14961 private javax.swing.JTabbedPane tp_rc_attributes_editions;  
14962 private javax.swing.JTabbedPane tp_rc_main;  
14963 private javax.swing.JMenu vue;  
14964 // End of variables declaration//GEN-END:variables  
14965 }  
14966
```

- Milliers de **lignes** de code
- Nombreux **fichiers**
- Des **erreurs** qui surviennent
- Des **bugs** qu'on comprend pas
- Retour à un **état antérieur** (version)

Des problèmes

- Qu'est-**ce qui** a été modifié (fichiers et lignes de code), **quand**, par **qui** et **pourquoi** ?

Des problèmes

- Qu'est-**ce qui** a été modifié (fichiers et lignes de code), **quand**, par **qui** et **pourquoi** ?
- Quelles **modifications** furent pour **ajouter** telle **fonctionnalité**, **résoudre** tel **bug** ?

Des problèmes

- Qu'est-ce qui a été modifié (fichiers et lignes de code), quand, par qui et pourquoi ?
- Quelles modifications furent pour ajouter telle fonctionnalité, résoudre tel bug ?
- Quelle est l'historique du code (les dates, les différentes versions, les intervenants) ?

Des problèmes

- Qu'est-**ce qui** a été modifié (fichiers et lignes de code), **quand**, par **qui** et **pourquoi** ?
- Quelles **modifications** furent pour **ajouter** telle **fonctionnalité**, **résoudre** tel **bug** ?
- Quelle est l'**historique** du code (les **dates**, les différentes **versions**, les **intervenants**) ?
- Comment **organiser** le code en **équipe** (toi tu modifie ceci, lui il modifie cela et moi ceci) **sans écraser** les modifications de l'autre ?

Des problèmes

- Qu'est-**ce qui** a été modifié (fichiers et lignes de code), **quand**, par **qui** et **pourquoi** ?
- Quelles **modifications** furent pour **ajouter** telle **fonctionnalité**, **résoudre** tel **bug** ?
- Quelle est l'**historique** du code (les **dates**, les différentes **versions**, les **intervenants**) ?
- Comment **organiser** le code en **équipe** (toi tu modifies ceci, lui il modifie cela et moi ceci) **sans écraser** les modifications de l'autre ?



Versioning

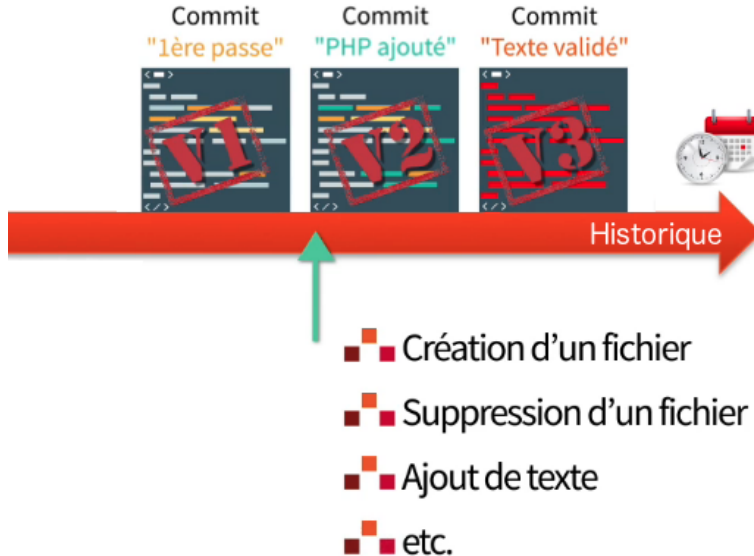
Plan de section

- 1 Contexte
- 2 Concepts et fondements de base
 - Généralités sur le versioning
 - Les logiciels et plateformes de versioning
 - Programme et évaluation
- 3 Environnement Git
 - Logiciel Git
 - Installation et configuration

Plan de sous-section

- 1 Contexte
- 2 Concepts et fondements de base
 - Généralités sur le versioning
 - Les logiciels et plateformes de versioning
 - Programme et évaluation
- 3 Environnement Git
 - Logiciel Git
 - Installation et configuration

Ce qu'est le versioning

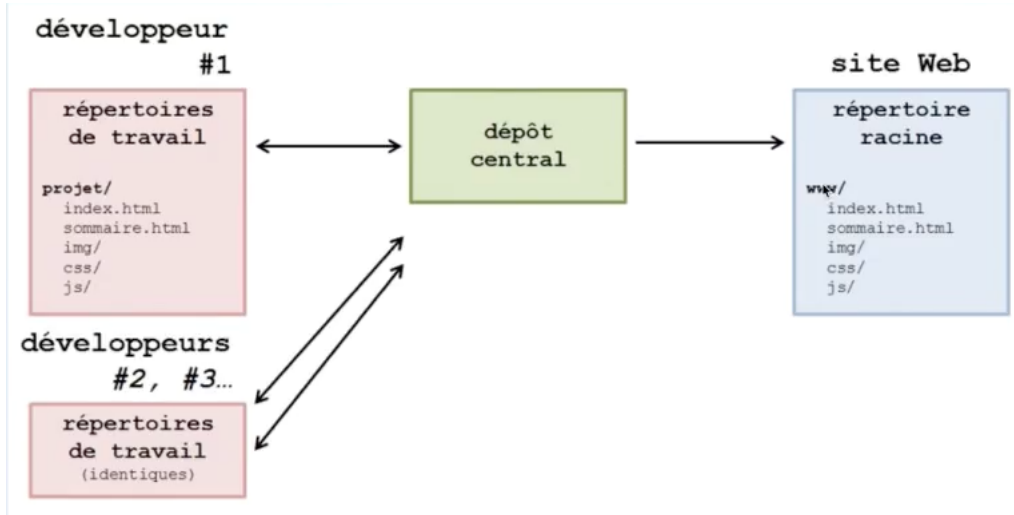


Le versioning dépassé

Nom	^	Date de modification	Taille	Type
▶ PFE v 1.0		15 juillet 2017 à 04:27	595 Ko	Dossier
▶ PFE v 1.1		15 juillet 2017 à 04:27	605 Ko	Dossier
▶ PFE v 1.2		15 juillet 2017 à 04:27	773 Ko	Dossier
▶ PFE v 1.3		15 juillet 2017 à 04:27	773 Ko	Dossier
▶ PFE v 2.0		15 juillet 2017 à 04:27	921 Ko	Dossier
▶ PFE v 2.1		15 juillet 2017 à 04:27	1 Mo	Dossier
▶ PFE v 2.2		15 juillet 2017 à 04:28	1,1 Mo	Dossier
▶ PFE v 2.3		15 juillet 2017 à 04:28	1,1 Mo	Dossier
▶ PFE v 2.4		15 juillet 2017 à 04:28	1,2 Mo	Dossier
▶ PFE v 3.0		15 juillet 2017 à 04:28	1,2 Mo	Dossier
▶ PFE v 3.1		15 juillet 2017 à 04:28	1,2 Mo	Dossier
▶ PFE v 3.2		15 juillet 2017 à 04:28	1,2 Mo	Dossier
▶ PFE v 3.3		15 juillet 2017 à 04:28	1,2 Mo	Dossier
▶ PFE v 3.4		15 juillet 2017 à 04:28	1,7 Mo	Dossier
▶ PFE v 3.5		15 juillet 2017 à 04:28	1,7 Mo	Dossier
▶ PFE v 4.0		15 juillet 2017 à 04:28	2,7 Mo	Dossier
▶ PFE v 4.1		21 juin 2018 à 16:48	1,9 Mo	Dossier

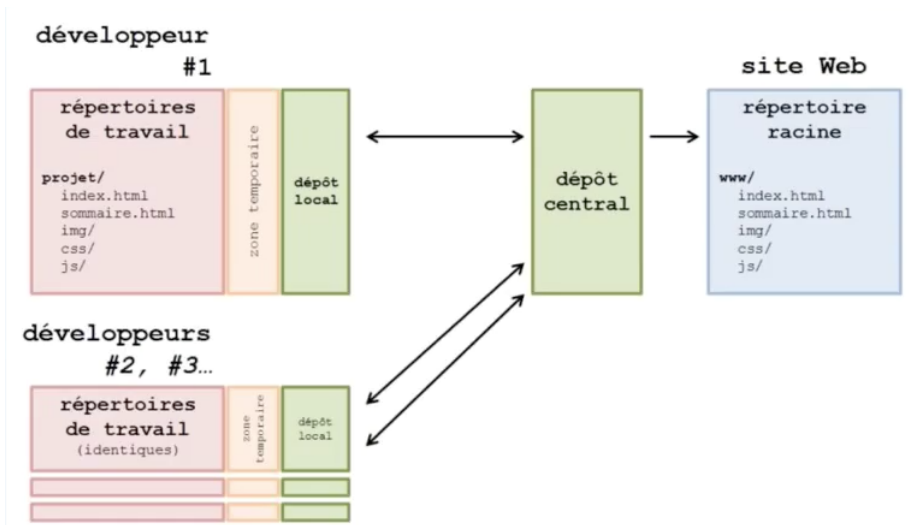
Copier coller N fois tout le dossier racine du code avec des noms différents

Les deux grandes familles de logiciels de versioning



1 : Le versioning centralisé (exemple : CVS, SVN)

Les deux grandes familles de logiciels de versioning



2 : Le versioning décentralisé ou distribué (exemple : Git, Mercurial)

Plan de sous-section

- 1 Contexte
- 2 Concepts et fondements de base
 - Généralités sur le versioning
 - Les logiciels et plateformes de versioning
 - Programme et évaluation
- 3 Environnement Git
 - Logiciel Git
 - Installation et configuration

Logiciels de versioning (centralisés)

CVS

- Un des plus anciens (~1980)
- Fonctionne toujours pour certains projets
- Incapable de suivre les fichiers renommés
==> utiliser SVN son successeur



Logiciels de versioning (centralisés)

CVS

- Un des plus anciens (~1980)
- Fonctionne toujours pour certains projets
- Incapable de suivre les fichiers renommés
==> utiliser SVN son successeur



SVN (Subversion)

- Probablement le plus utilisé actuellement (depuis 2000)
- Simple d'utilisation mais un temps d'adaptation
Utilisation concentrée en ligne de commande
- S'intègre bien à Windows avec Tortoise SVN



Logiciels de versioning (distribués)

Git

- Très puissant et récent (~2004)
Créé par Linux Torwalds
- Plus rapide et gère bien le concept de branches



Logiciels de versioning (distribués)

Git

- Très puissant et récent (~2004)
Créé par Linux Torwalds
- Plus rapide et gère bien le concept de branches



Mercurial

- Plus récent, complet et puissant
Apparu quelques jours après le début de développement de Git
- Très comparable à Git



Plateformes ou serveurs de services versioning

GitHub

- Gratuit pour les projets opensources
- Projets privés illimités pour 3 users (sinon payant)
- Pas de limite d'utilisateurs pour projet opensources



Plateformes ou serveurs de services versioning

GitHub

- Gratuit pour les projets opensources
- Projets privés illimités pour 3 users (sinon payant)
- Pas de limite d'utilisateurs pour projet opensources



Bitbucket

- Pas de limite de projets privés
- 5 utilisateurs maxi par projet (sinon payant)



Plan de sous-section

- 1 Contexte
- 2 Concepts et fondements de base
 - Généralités sur le versioning
 - Les logiciels et plateformes de versioning
 - Programme et évaluation
- 3 Environnement Git
 - Logiciel Git
 - Installation et configuration

Objectifs :

Objectif global :

Appréhender le versioning et le travail collaboratif

Objectifs :

Objectif global :

Appréhender le versioning et le travail collaboratif

Objectifs spécifiques :

- Savoir manipuler Git (en console et graphiquement)
- Savoir cloner/migrer un dépôt distant
- Savoir coder en groupe

Programme

Deux parties à piloter ensemble :

- 1 Prise en main avec Git
- 2 Codage collaboratif

Modalités d'évaluation

On verra comment !

Plan de section

- 1 Contexte
- 2 Concepts et fondements de base
 - Généralités sur le versioning
 - Les logiciels et plateformes de versioning
 - Programme et évaluation
- 3 Environnement Git
 - Logiciel Git
 - Installation et configuration

Plan de sous-section

- 1 Contexte
- 2 Concepts et fondements de base
 - Généralités sur le versioning
 - Les logiciels et plateformes de versioning
 - Programme et évaluation
- 3 Environnement Git
 - Logiciel Git
 - Installation et configuration

Outils nécessaires et compléments

Les nécessaires



Git (logiciel)
git-scm.com



Editeur de texte (client)
Bloc note, TextEdit, IDE...



GitHub ou Bitbucket (serveur)
github.com, bitbucket.org

Outils nécessaires et compléments

Les compléments



Git Large File Storage
<https://git-lfs.github.com/>



Gitignore.io
<https://www.gitignore.io/>

Outils nécessaires et compléments

Les compléments



GitKraken

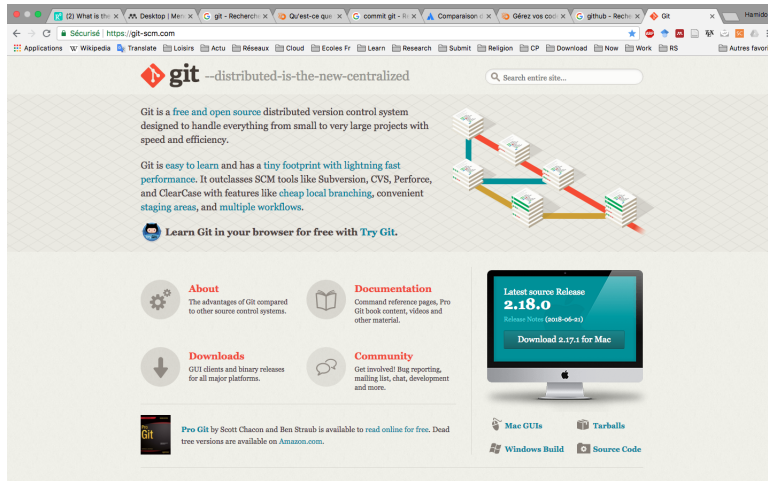
<https://www.gitkraken.com/>



Visual Studio Code

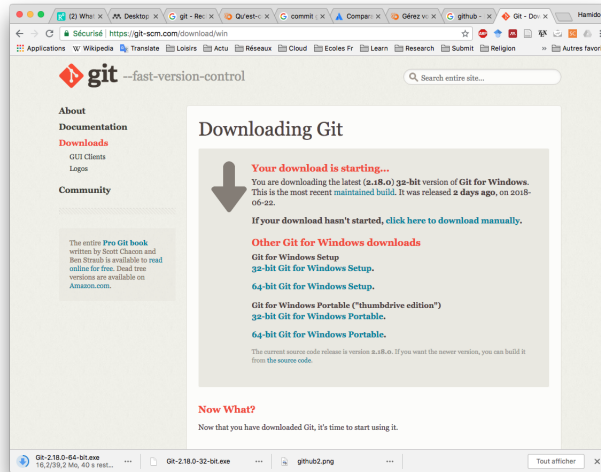
<https://code.visualstudio.com/>

Téléchargement (logiciel)



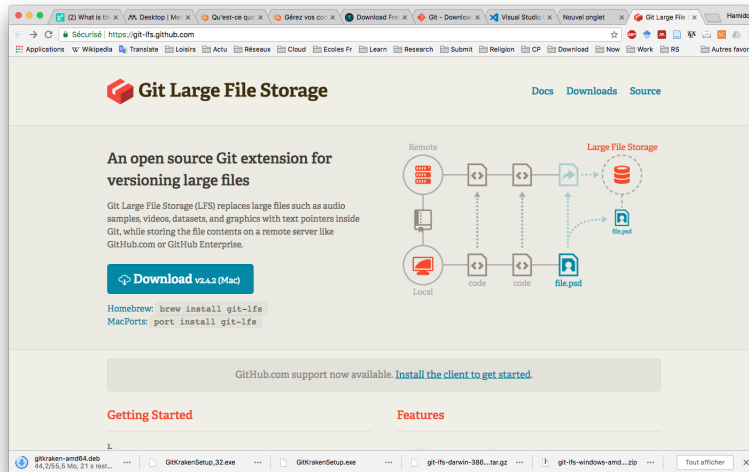
Page d'accueil de téléchargement du logiciel Git

Téléchargement (logiciel)



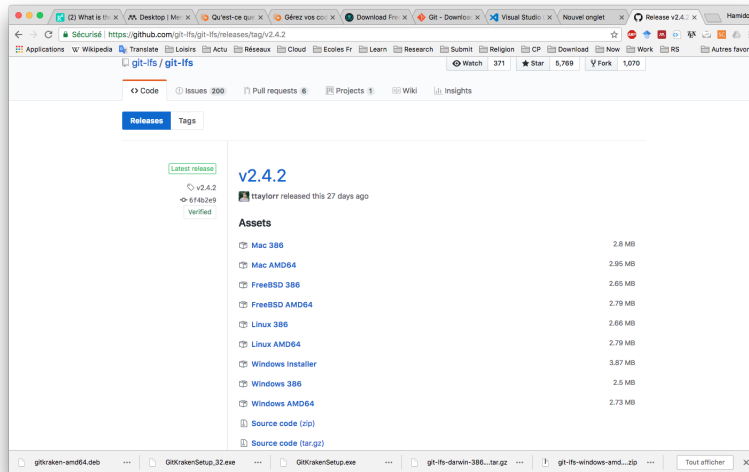
Page de téléchargement du logiciel Git selon l'OS

Téléchargement (complément du logiciel)



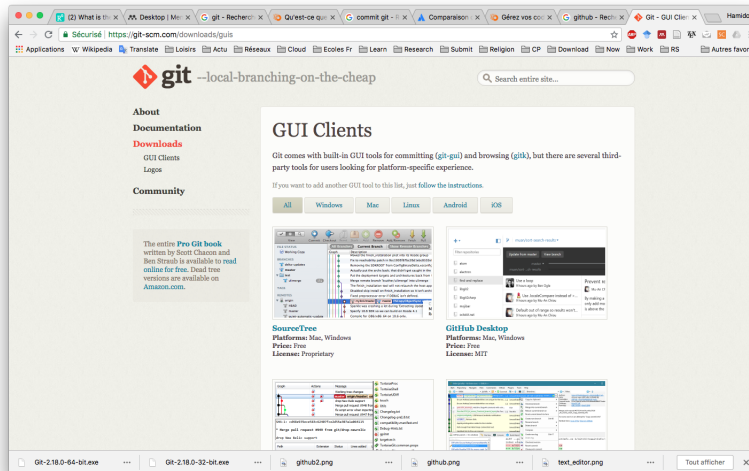
Page d'accueil de téléchargement de Git Large File Storage

Téléchargement (complément du logiciel)



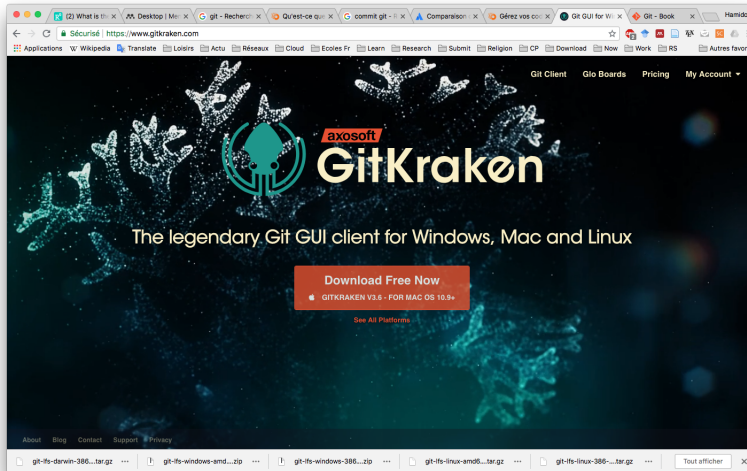
Page de téléchargement de Git Large File Storage selon l'OS

Téléchargement (clients GUI)



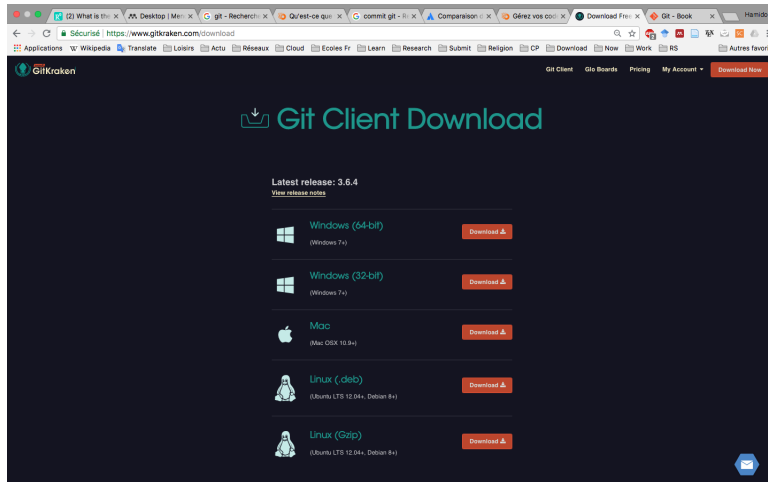
Page d'accueil des différents clients GUI

Téléchargement (clients GUI)



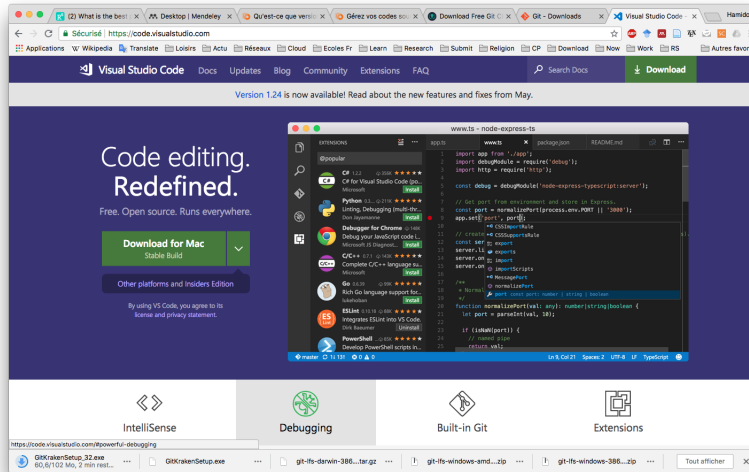
Page d'accueil de téléchargement de GitKraken

Téléchargement (clients GUI)



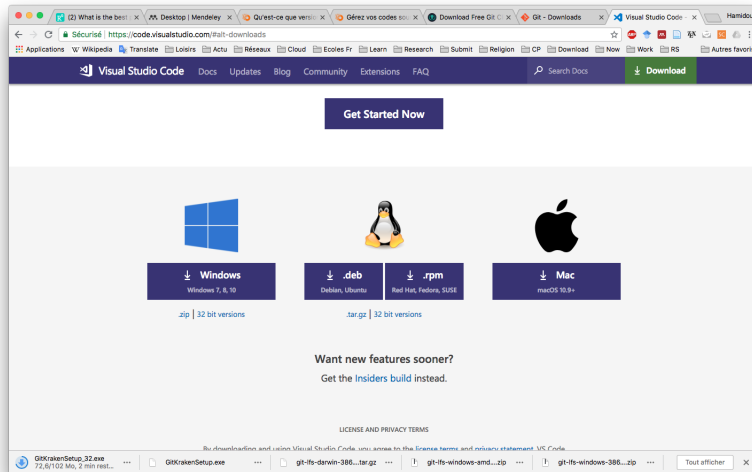
Page de téléchargement de GitKraken selon l'OS

Téléchargement (clients GUI)



Page d'accueil de téléchargement de Visual Studio Code

Téléchargement (clients GUI)



Page de téléchargement de Visual Studio Code selon l'OS

Plan de sous-section

- 1 Contexte
- 2 Concepts et fondements de base
 - Généralités sur le versioning
 - Les logiciels et plateformes de versioning
 - Programme et évaluation
- 3 Environnement Git
 - Logiciel Git
 - Installation et configuration

Installation

L'ordre d'installation est à prendre en compte uniquement pour le logiciel [git](#) et le complément [Git Large File Storage](#).

Installation

L'ordre d'installation est à prendre en compte uniquement pour le logiciel [git](#) et le complément [Git Large File Storage](#).

- **Windows**

Pour chaque fichier d'installation, double clic et valeurs par défaut à chaque étape.

Installation

L'ordre d'installation est à prendre en compte uniquement pour le logiciel [git](#) et le complément [Git Large File Storage](#).

- **Windows**

Pour chaque fichier d'installation, double clic et valeurs par défaut à chaque étape.

- **Mac**

Même cas que Windows.

Il faut le mot de passe et autoriser les applications de sources inconnues de App Store.

Installation

L'ordre d'installation est à prendre en compte uniquement pour le logiciel [git](#) et le complément [Git Large File Storage](#).

- **Windows**

Pour chaque fichier d'installation, double clic et valeurs par défaut à chaque étape.

- **Mac**

Même cas que Windows.

Il faut le mot de passe et autoriser les applications de sources inconnues de App Store.

- **Linux**

Commende : [sudo apt-get install git-core gitk](#).

Mot de passe requis aussi et il faut donc la connexion.

Configuration

Utilisateur

```
1 git config --global user.name "votre_pseudo"  
2 git config --global user.email moi@email.com
```


Configuration

Utilisateur

```
1 git config --global user.name "votre_pseudo"  
2 git config --global user.email moi@email.com
```

Gestion des couleurs

```
1 git config --global color.diff auto  
2 git config --global color.status auto  
3 git config --global color.branch auto
```