Introdution

Git est un utilitaire en ligne de commande qui permet de gérer l'historique et les différentes version d'une base de code ou de tout autre forme de document. GitHub est une plateforme qui permet à tout le monde d’utiliser Git de partout.

1. Qu’est-ce que Git ?
2. Un outil de « gestion de conf » = gérer historique et versions (des travaux parallèles peuvent cohabiter).
3. Travail en équipe avec Git = branches, fusions de branches, fork (duplication de dépôts) , relecture par des pairs, etc.
4. Lien avec la gestion de projet = Schéma des branches (il existe un schéma d’utilisation des branches recommandés, avec branche master, branches d’intégration, branches de maintenance, branches d’évolution, et parfois branches de correction de bugs). Lien avec les outils de « gestion de tickets ». Release => Git tag.
5. GitHub: les fonctionnalités du site web
6. Git sur le net. Différentes API
7. L’intégration continue = Github est nativement branché (branchable) sur des outils d’intégration continue (des programmes qui vont, automatiquement, à la moindre modif de code, compiler / exécuter des tests unitaires ou fonctionnels / exécuter l’appli elle-même, etc. et qui prévient tout de suite si le moindre problème est détecté).
8. Une plateforme qui aide le développement et l’open source
9. Business model (faire payer les dépôts privés pour permettre à tous d’accéder aux dépôts publics)
10. Le travail en équipe devient un travail collaboratif = Des individus du monde entier ne se connaissant pas peuvent et vont travailler ensemble. Ex : une librairie est développée par A. B veut l’utiliser mais détecte un bug. A n’a pas le temps de le corriger suffisamment vite pour B. B va forker la librairie, corriger le bug, déployer cette lib améliorer en production. A pourra récupérer, lorsqu’il le souhaite la modif de B.
11. Exemples de grands projets hébergés sur GitHub et anecdotes (à trouver !)