

INTRODUCTION DE GIT ET GITHUB

Slides adopted from Meghan Nelson <https://www.slideshare.net/HubSpot/git-101-git-and-github-for-beginners>

APERÇU

1. Installer git et créer un compte Github
2. Qu'est-ce que git?
3. Comment git fonctionne-t-il?
4. Qu'est-ce que GitHub?
5. Exemple court utilisant un git et GitHub

INSTALLER GIT ET CRÉER UN COMPTE GITHUB

Installer git

- **Mac** <https://git-scm.com/download/mac>
- **Windows** <https://git-scm.com/download/win>

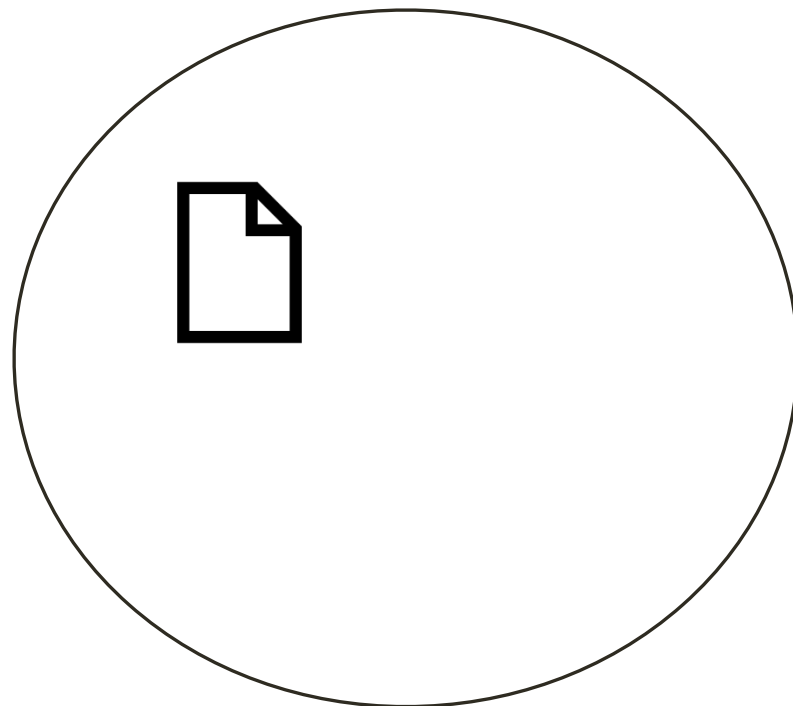
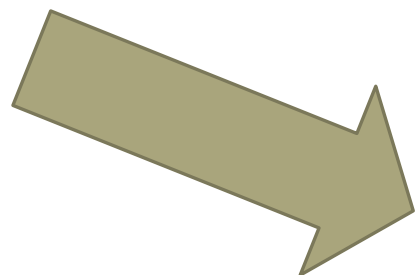
Créer un compte Github

- www.github.com
- Gratuit pour les référentiels (repositories) publics

QU'EST-CE QUE UN GIT?

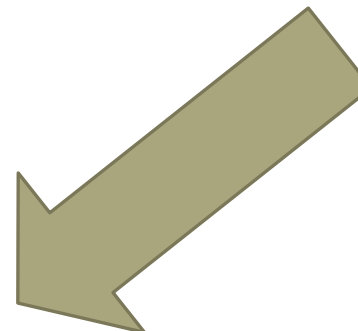
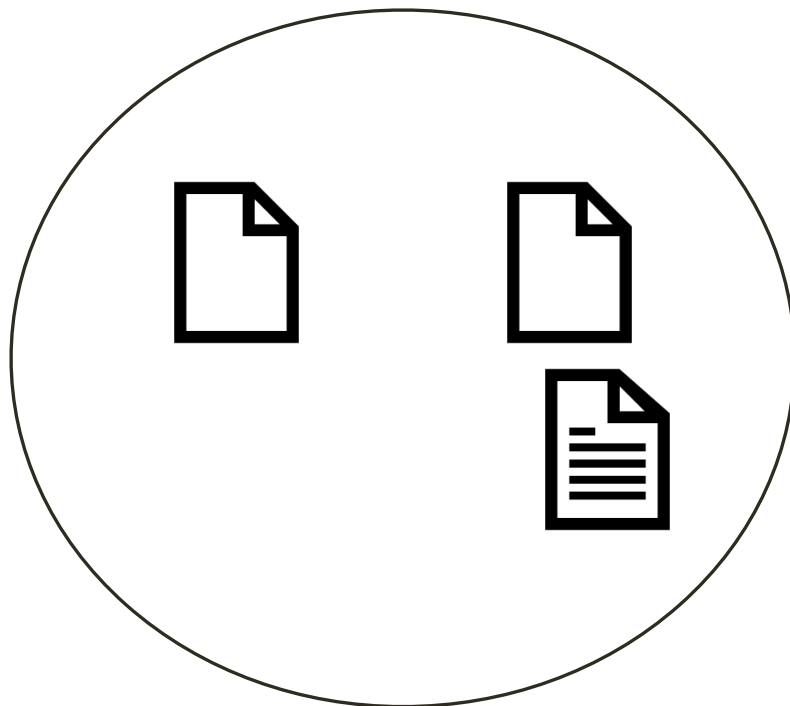
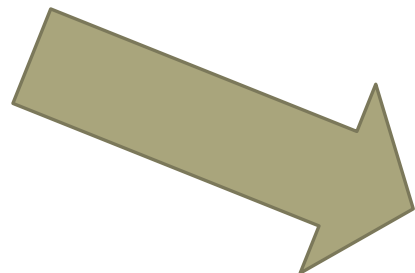


Alice





Alice



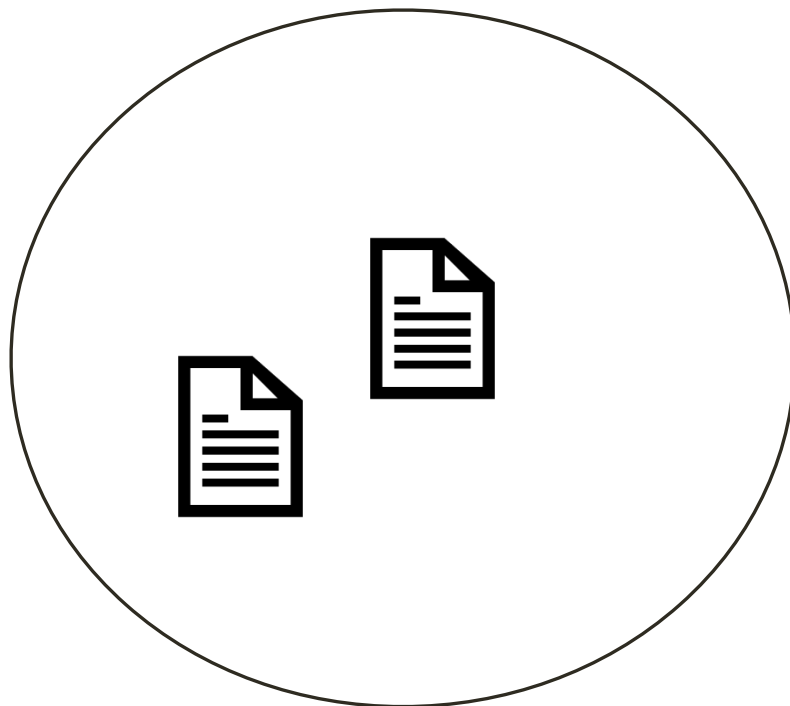
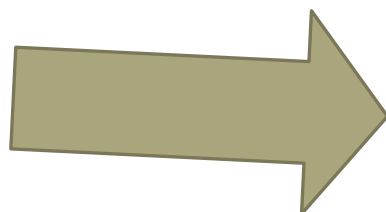
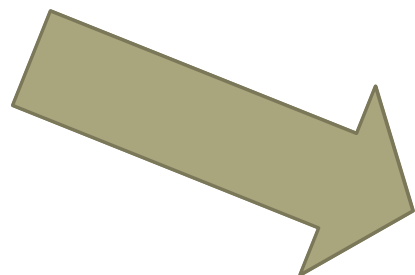
Bob



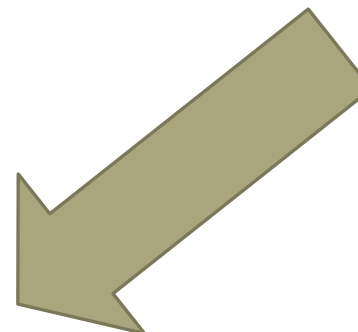
Alice



Joe



Bob





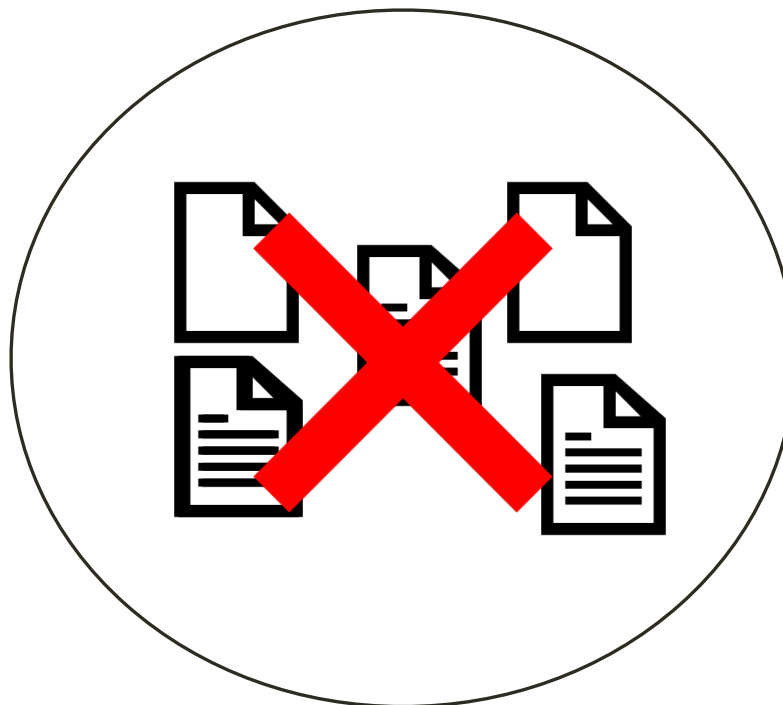
Alice



Bob



Joe



Qui a remplacé les fichiers ? Quand ?

QU'EST-CE QUE LE CONTRÔLE DE VERSION ?

- Un système qui conserve des enregistrements de vos modifications.
- Permet un développement collaboratif.
- Permet de savoir qui a effectué quelles modifications et quand.
- **Vous permet d'annuler toute modification et de revenir à un état précédent.**

QU'EST-CE QU'UN GIT ?

- Contrôle de version distribué
- Les utilisateurs conservent l'intégralité du code et de l'historique sur leurs machines locales
- Les utilisateurs peuvent apporter des modifications sans accès à Internet
- (Sauf pousser et extraire des modifications depuis un serveur distant)

QU'EST-CE QU'UN GIT ?

- Début en 2005.
- Créé par Linus Torvald pour aider au développement du noyau Linux.
- Git n'est pas le seul système de contrôle de version, mais (nous pensons) c'est le meilleur.

COMMENT ÇA FONCTIONNE ?

- Cela peut être compliqué au début, mais il y a quelques concepts clés.
- La terminologie importante de **git** est dans les diapositives suivantes

CONCEPTS CLÉS: **SNAPSHOTS**

- La façon dont **git** garde une trace de l'historique de votre code.
- Enregistre essentiellement à quoi ressemblent tous vos fichiers à un moment donné.
- Vous décidez quand prendre un **snapshot** et de quels fichiers.
- Avoir la possibilité de revenir en arrière pour visiter n'importe quel snapshot.
- Vos snapshots ultérieurs resteront également disponibles.

CONCEPTS CLÉS: COMMIT

- L'acte de créer un **snapshot**
- Peut être un nom ou un verbe
 - “I committed code”
 - "Je viens de faire un nouveau commit"
- Essentiellement, un projet est composé d'un ensemble de commits.

CONCEPTS CLÉS: COMMIT

- Commits contient 3 parties d'information information:
 1. Information sur la façon dont les fichiers ont changé par rapport à précédemment
 2. Une référence au commit qui l'a précédé
 - Appelé le “commit parent”
 3. Un nom de code de hachage
 - Ressemblera à quelque chose comme:
 - fb2d2ec5069fc6776c80b3ad6b7cbde3cade4e

CONCEPTS CLÉS: REPOSITORIES

- Souvent abrégé en « **repo** ».
- Une collection de tous les fichiers et l'historique de ces fichiers
 - Se compose de tous vos commits
 - Lieu où est stocké tout votre travail

CONCEPTS CLÉS: REPOSITORIES

- Peut être sur une machine locale ou sur un serveur distant (GitHub !)
- L'acte de copier un repository depuis un serveur distant est appelé **clonage**
- Le clonage depuis un serveur distant permet aux équipes de travailler ensemble

CONCEPTS CLÉS: REPOSITORIES

- Le processus de téléchargement de commits qui n'existent pas sur votre ordinateur à partir d'un repository distant est appelé **pulling**.
- Le processus d'ajout de vos modifications locales au repository distant est appelé « **pushing** ».

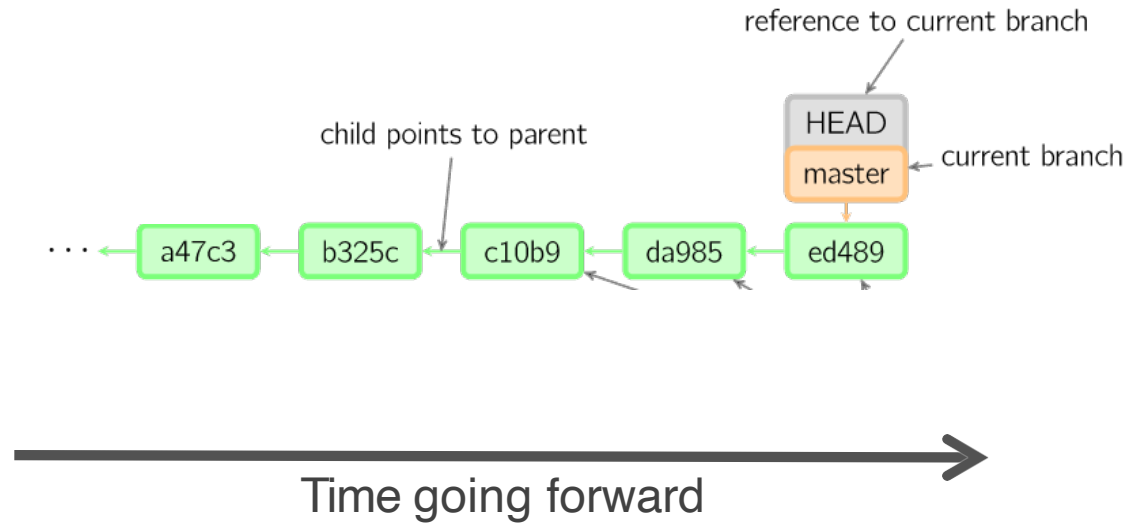
CONCEPTS CLÉS: BRANCHES

- Tous les commits dans git vivent sur une branche.
- Mais il peut y avoir de très nombreuses branches.
- La branche principale d'un projet est appelée la branche maître (master branch)

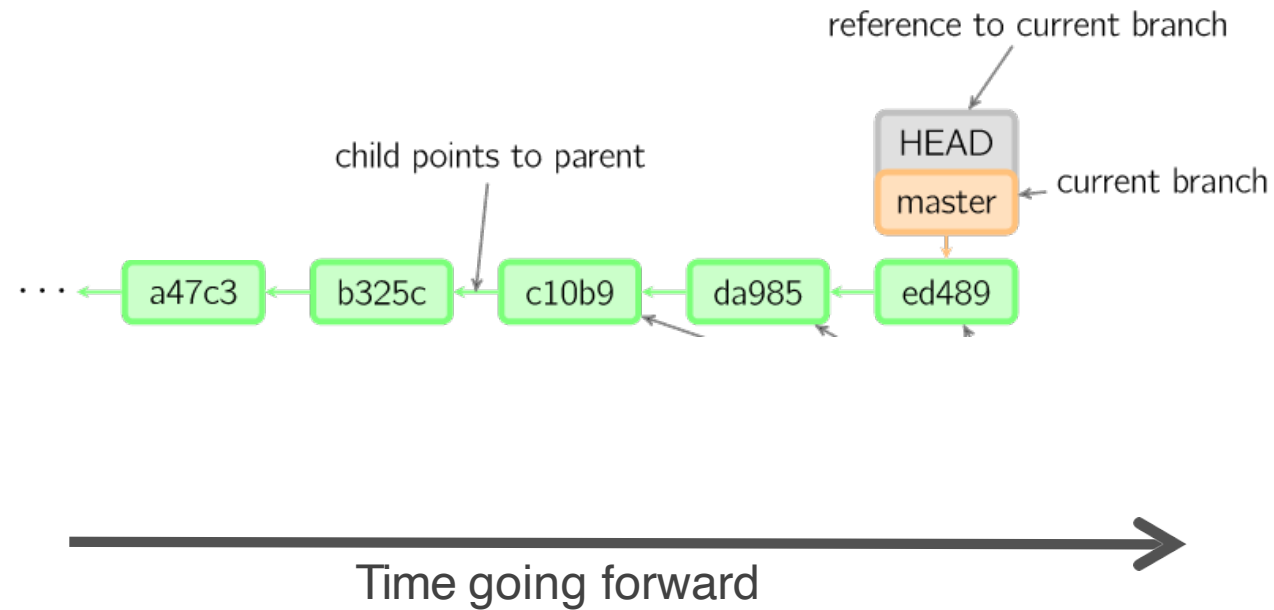
À QUOI RESSEMBLE UN PROJET TYPE ?

- Un tas de commits liés entre eux qui vivent sur une branche, contenus dans un repository.
- Les images suivantes prises et modifiées à partir de :
 - <http://marklodato.github.io/visual-git-guide/index-en.html>
 - Un bon tutorial!

À QUOI RESSEMBLE UN PROJET TYPE ?

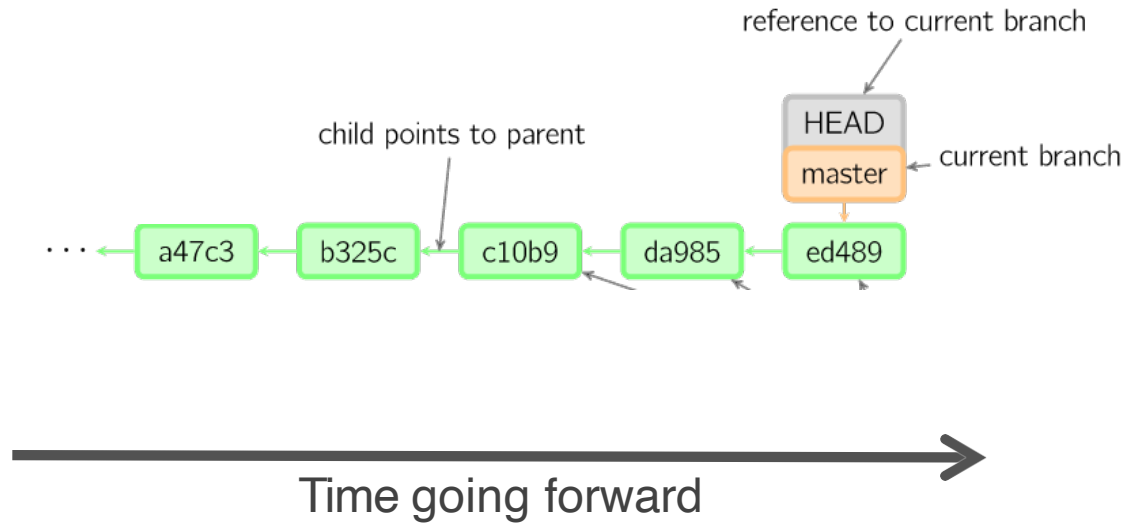


QU'EST-CE QUE HEAD?



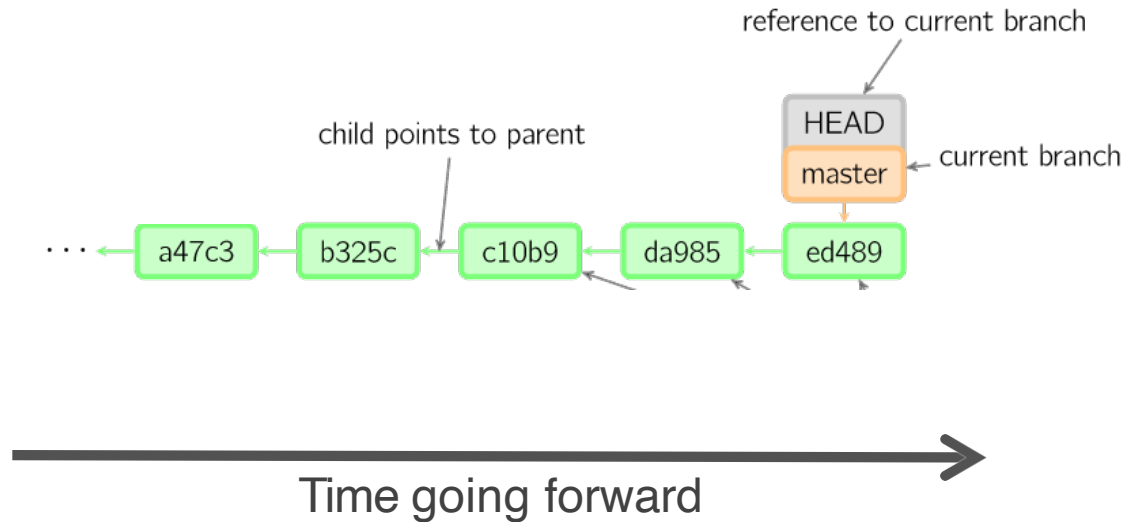
QU'EST-CE QUE HEAD?

Reference au commit le plus récent.



QU'EST-CE QUE MASTER?

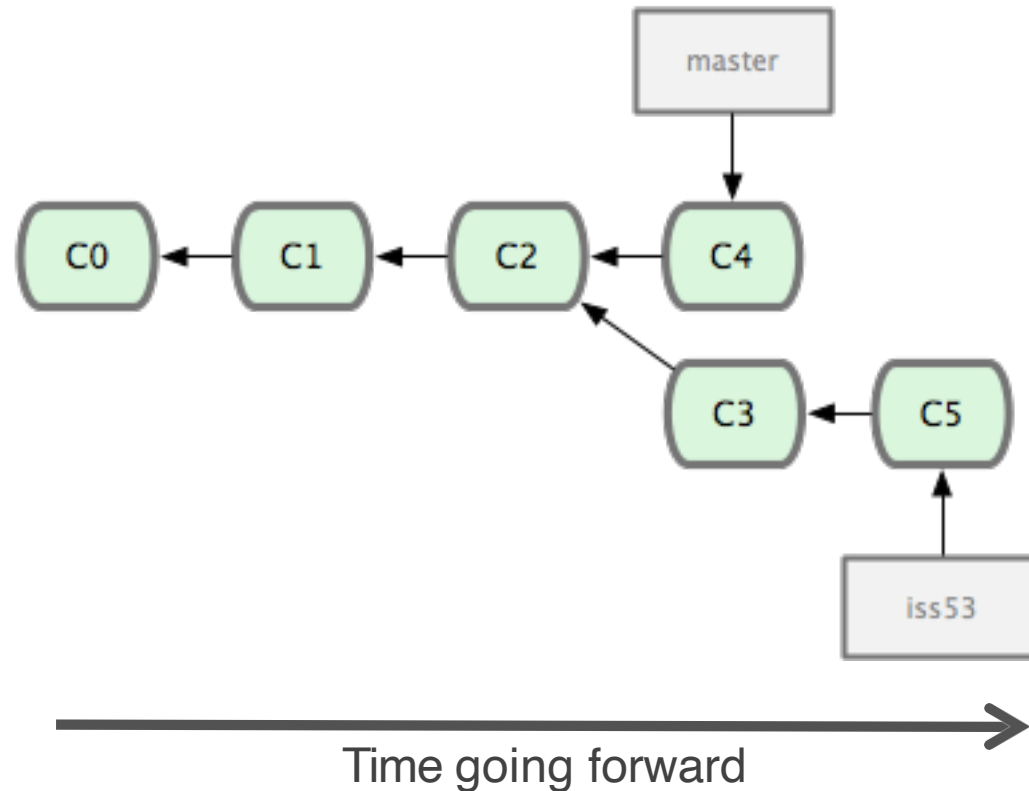
- La branche principale dans votre project
- Il n'est pas nécessaire de l'appeler maître (master), mais c'est presque toujours le cas !



CONCEPTS CLÉS: DÉRIVATION DE LA BRANCHE MAÎTRE

- The start of a branch points to a specific commit.
- When you want to make any changes to your project you make a new branch based on a commit.

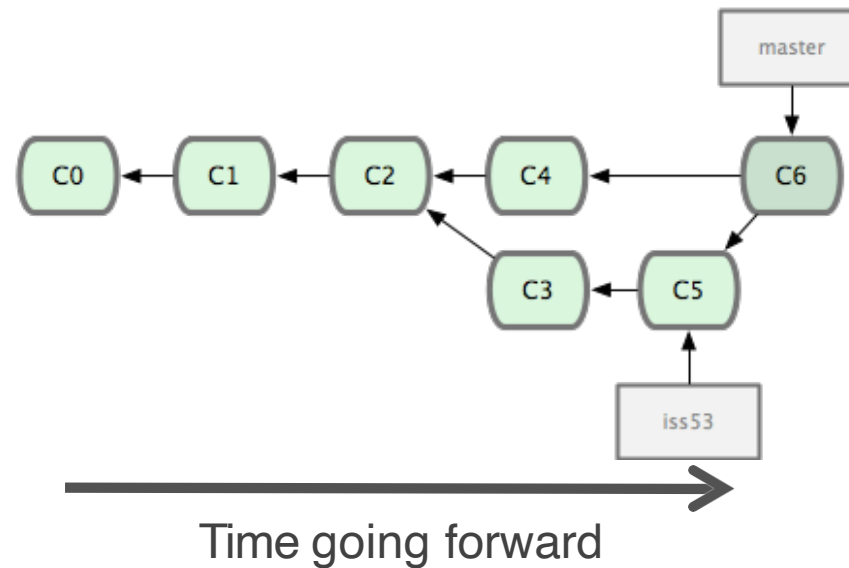
CONCEPTS CLÉS: DÉRIVATION DE LA BRANCHE MAÎTRE



Images de: <https://codingdomain.com/git/merging/>

CONCEPTS CLÉS: FUSIONNER (MERGING)

Une fois que vous avez terminé avec votre fonctionnalité, vous la fusionnez à nouveau dans le maître



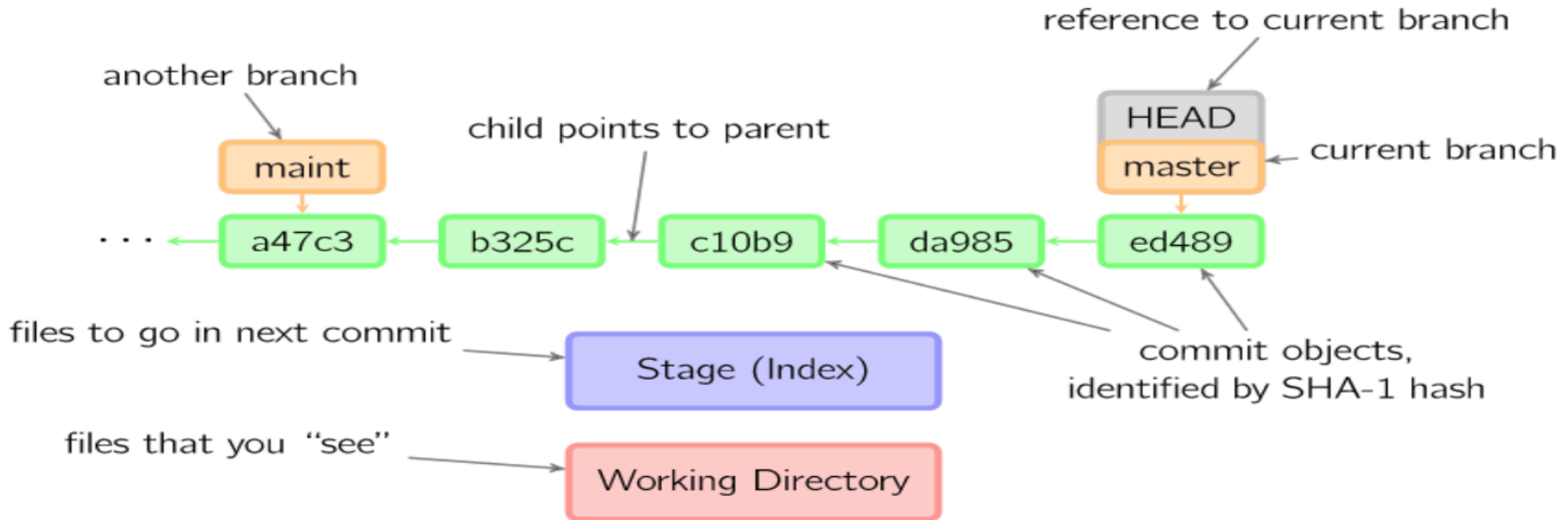
COMMENT FAIRE UN COMMIT ?

- Il existe de nombreux « états » et « emplacements » dans lesquels un fichier peut se trouver.
- Local sur votre ordinateur : le « répertoire de travail :[working directory](#) »
- Lorsqu'un fichier est prêt à être placé dans un commit, vous l'ajoutez à « l'index » ou à la « [staging](#) ».
- Staging est le nouveau terme préféré, mais vous pouvez voir à la fois « index » et « staging » utilisés.

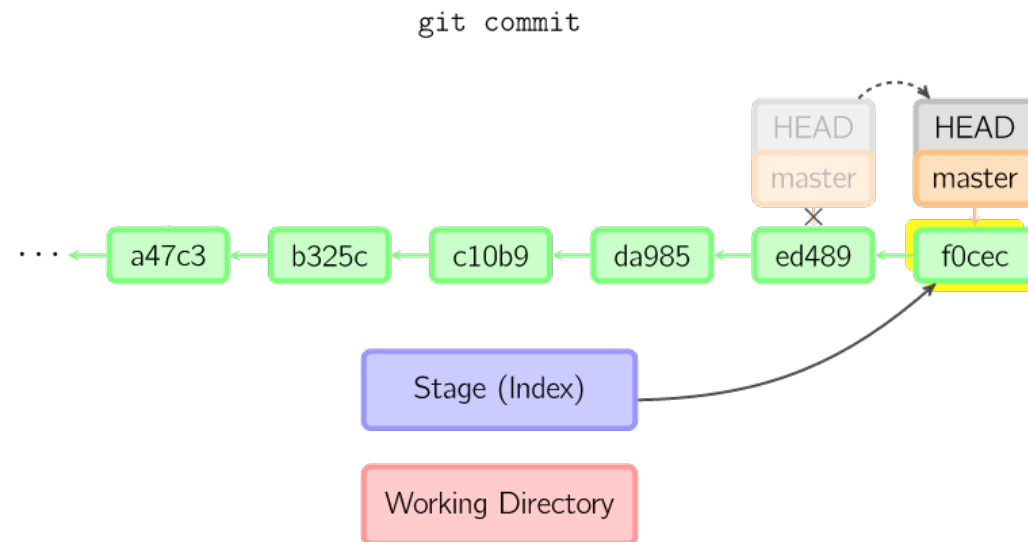
COMMENT FAIRE UN COMMIT ?

- Le processus:
 - Apporter des modifications à un fichier
 - Utilisez la commande `git add` pour placer le fichier dans l'environnement de transfert: `staging environment`
 - Utilisez la commande `'git commit'` pour créer un nouveau commit

COMMENT FAIRE UN COMMIT ?



COMMENT FAIRE UN COMMIT ?



QU'EST-CE QUE GITHUB?

- www.github.com
- Le plus grand service d'hébergement de repository (référentiel) Git basé sur le Web
 - Aka, héberge des « référentiels distants: [remote repositories](#) »
- Permet la collaboration de code avec n'importe qui en ligne
- Ajoute des fonctionnalités supplémentaires en plus de git
 - Interface utilisateur: UI, documentation, suivi des bogues, demandes de fonctionnalités, demandes d'extraction (pull) et bien plus encore !
-

RESSOURCES SUPPLEMENTARIES

- Official git site and tutorial: <https://git-scm.com/>
- GitHub guides: <https://guides.github.com/>
- Command cheatsheet:
 - <https://education.github.com/git-cheat-sheet-education.pdf>
 - <https://github.github.com/training-kit/downloads/github-git-cheat-sheet.pdf>
 - (Arabic) <https://github.github.com/training-kit/downloads/ar/github-git-cheat-sheet/>
- Interactive git tutorial: <https://learngitbranching.js.org/>
- Visual/interactive cheatsheet: <http://ndpsoftware.com/git-cheatsheet.html>
- GitHub Desktop: <https://help.github.com/en/desktop/getting-started-with-github-desktop>