



GitHub

Une plateforme de
développement collaboratif

Réalisé par :

Asmae ABDELOUHAB
Ghizlane BENNANI

Encadré par:

Prof.Khalid NAFIL

Plan

1. Qu'est ce que GitHub ?
2. Historique de GitHub
3. Pourquoi GitHub ?
4. Concepts de GitHub
5. Atelier pour mieux comprendre



GitHub en tant que plateforme de développement collaboratif

- GitHub est une plateforme web qui a été créée pour permettre aux développeurs de travailler ensemble de manière plus efficace et transparente. Il s'agit d'un service d'hébergement de projets basé sur le système de contrôle de version Git.

- En utilisant GitHub, les développeurs peuvent stocker, gérer et collaborer sur des projets de développement logiciel.
- Chaque projet est hébergé dans un espace de stockage en ligne appelé "dépôt" (repository) qui contient le code source, l'historique des modifications, les problèmes, les notes et bien d'autres éléments essentiels pour la gestion de projets.

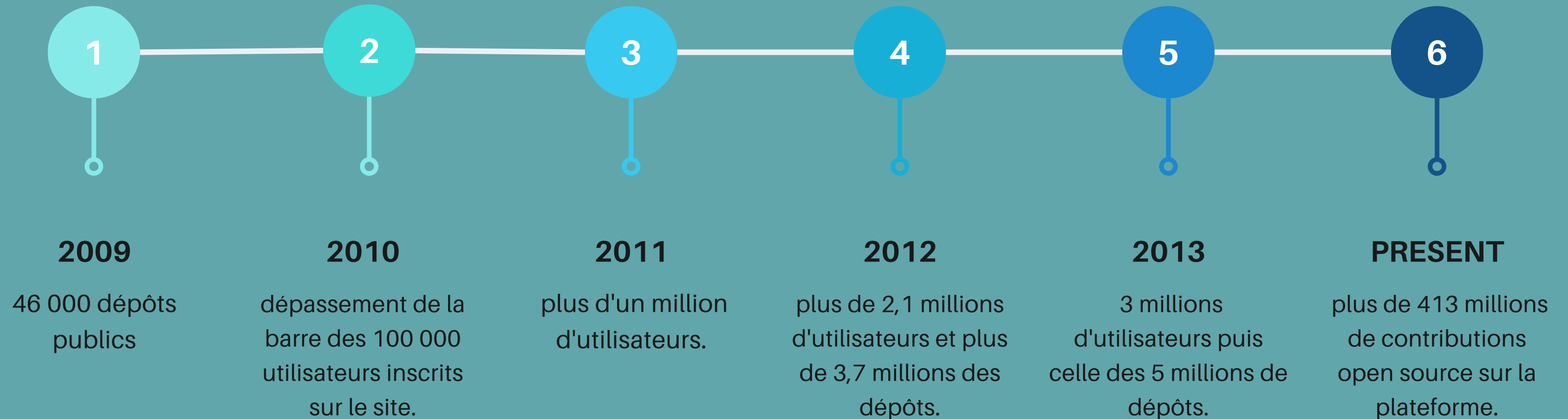
la popularité de GitHub dans la communauté de développement

- Environ 100 millions de développeurs à travers le monde utilisent GitHub.
- En 2022, il y a eu plus de 413 millions de contributions open source sur la plateforme.
- En 2022, les développeurs ont utilisé près de 500 langages pour créer des logiciels sur GitHub
- La popularité de GitHub ne se limite pas aux projets open source. De nombreuses entreprises et équipes de développement privées utilisent également GitHub pour gérer leurs projets internes, renforçant ainsi sa position en tant que plateforme de développement collaborative de premier plan.

la popularité de GitHub dans la communauté de développement

- GitHub offre des fonctionnalités de suivi des problèmes, ce qui facilite la gestion des bogues, des améliorations et des tâches de développement.
- De nombreux projets open source, des petits projets personnels aux grandes initiatives, sont hébergés sur GitHub, ce qui en fait une ressource inestimable pour les développeurs qui souhaitent collaborer et contribuer à des projets de toute envergure.

Historique de github



Historique de github

- Le 24 février 2009, les membres de l'équipe GitHub ont annoncé que durant la première année, le site avait accumulé 46 000 dépôts publics, dont 17 000 dans le seul mois précédent.
- Le 5 juillet 2009, le GitHub Blog a annoncé le dépassement de la barre des 100 000 utilisateurs inscrits sur le site.
- En juillet 2010, GitHub a annoncé qu'il accueille un million de dépôts. En avril 2011, GitHub a annoncé qu'il accueille deux millions de dépôts.
- Le 21 septembre 2011 GitHub a annoncé qu'il avait atteint plus d'un million d'utilisateurs.

Historique de github

- Le 13 septembre 2012, sur sa page d'accueil, GitHub a annoncé qu'il avait plus de 2,1 millions d'utilisateurs et plus de 3,7 millions des dépôts.
- Le 19 décembre 2012, GitHub a annoncé qu'il avait plus de 2,8 millions d'utilisateurs et plus de 4,6 millions de dépôts.
- Le 16 janvier 2013, GitHub a annoncé qu'il avait passé la barre des 3 millions d'utilisateurs puis celle des 5 millions de dépôts.
- Le 10 avril 2013, GitHub a annoncé qu'il avait 3,5 millions d'utilisateurs et accueillait désormais plus de 6 millions de dépôts.
- Le 23 décembre 2013, GitHub a annoncé qu'il avait atteint 10 millions de dépôts hébergés.

Pourquoi GitHub ?

problèmes
résolus

AVANTAGES



PROBLÈMES RÉSOLUS PAR GITHUB

Collaboration en Équipe Facilitée : GitHub permet à des équipes de développeurs de travailler ensemble de manière efficace, qu'elles soient situées localement ou à travers le monde. Il simplifie la collaboration en offrant des outils de partage, de revue de code, et de suivi des modifications.

Contrôle de Version Avancé : GitHub repose sur le système de contrôle de version Git, qui permet de suivre les modifications apportées au code source de manière détaillée. Cela résout le problème des versions du code, des conflits et des pertes de données.

PROBLÈMES RÉSOLUS PAR GITHUB

Hébergement d'Open Source : GitHub a révolutionné l'hébergement de projets open source en offrant un espace de stockage gratuit pour les projets publics. Cela a encouragé la collaboration ouverte et la contribution de la communauté.

Suivi des Problèmes (Issue Tracking) : Les fonctionnalités de suivi des problèmes de GitHub permettent de signaler, suivre et résoudre des bogues, des demandes d'amélioration et des tâches de développement de manière organisée.

PROBLÈMES RÉSOLUS PAR GITHUB

Révision de Code Simplifiée : La revue de code est essentielle pour la qualité du code. GitHub offre des outils qui facilitent la revue de code, les commentaires et les suggestions pour améliorer le code source.

Sécurité et Confidentialité : GitHub propose des fonctionnalités de sécurité, notamment l'authentification à deux facteurs, le chiffrement, et la gestion fine des autorisations pour protéger le code et les données.

Avantages

Contrôle de Version : GitHub utilise Git, un système de gestion de version, qui permet de suivre les modifications apportées aux fichiers au fil du temps. Cela facilite le suivi des versions, la gestion des conflits et le retour en arrière en cas de besoin.

Collaboration Facilitée :

- GitHub facilite la collaboration entre les membres d'une équipe. Plusieurs personnes peuvent travailler sur le même projet simultanément.
- Les fonctionnalités telles que les pull requêtes permettent d'examiner, discuter et fusionner facilement les contributions.

Avantages

VISIBILITÉ ET ACCESSIBILITÉ :

- Les dépôts publics sur GitHub offrent une visibilité mondiale à votre projet, ce qui peut attirer des contributeurs et des utilisateurs potentiels.
- La plateforme est accessible via une interface web, facilitant ainsi la collaboration à distance.

GESTION DES PROBLÈMES (ISSUES) : GITHUB PROPOSE UN SYSTÈME INTÉGRÉ DE GESTION DES PROBLÈMES QUI PERMET DE SIGNALER, DISCUTER ET RÉSOUDRE LES PROBLÈMES RENCONTRÉS DANS UN PROJET.

Avantages

Documentation Simple : La fonctionnalité wiki et la possibilité d'ajouter des fichiers README.md rendent la documentation du projet facile à créer, à maintenir et à partager.

Déploiement Facilité : Les services intégrés, tels que GitHub Actions, facilitent le déploiement automatique de votre application après chaque modification réussie.

Concepts de github

- ↳ Les Dépôts (Repositories)
- ↳ Clonage
- ↳ Fork (fourche)
- ↳ Commits et Messages
- ↳ Collaboration avec les Branches
- ↳ Merge (Fusion)
- ↳ Collaborateurs
- ↳ Pull Requests (Demandes de Tirage)

Dépôts (Repositories)

- Un dépôt est un espace de stockage où vous pouvez stocker les fichiers de votre projet, ainsi que l'historique des modifications apportées à ces fichiers.
- Sur GitHub, un dépôt peut être public, ce qui signifie que tout le monde peut le voir, ou privé, limitant l'accès aux personnes autorisées.

Clonage

- Cloner un dépôt signifie copier l'intégralité du dépôt, y compris l'historique des modifications, sur votre propre machine locale.
- Cela vous permet de travailler sur une copie locale du projet sans affecter le dépôt principal, et vous pouvez ensuite pousser vos modifications une fois que vous êtes prêt.

Fork (fourche)

- Un fork est une copie d'un dépôt sur votre propre compte GitHub. Cela crée essentiellement une version indépendante du projet.
- Les forks sont couramment utilisés pour contribuer à des projets open source. Vous pouvez apporter des modifications à votre fork et proposer ces modifications au dépôt principal via une demande de tirage (pull request).

Commits et Messages

- Un commit est une modification enregistrée dans l'historique d'un dépôt. Cela représente une unité de changement dans votre projet.
- Lorsque vous apportez des modifications à votre projet, vous effectuez un commit pour enregistrer ces modifications. Chaque commit est accompagné d'un message décrivant les changements apportés.
- Les messages de commit sont importants car ils permettent à vous-même et aux autres contributeurs de comprendre rapidement les modifications apportées à chaque étape du projet.

Collaboration avec les branches

Les branches dans un système de contrôle de version permettent aux développeurs de travailler sur des fonctionnalités ou des corrections de bugs de manière isolée, sans affecter la branche principale du projet (généralement appelée main ou master).

La collaboration avec les branches signifie que plusieurs développeurs peuvent travailler simultanément sur différentes fonctionnalités ou problèmes, chacun dans sa propre branche. Cela facilite la parallélisation du travail et la gestion des modifications indépendantes.

MERGE

Une fois qu'un développeur a terminé son travail dans une branche, les changements doivent être fusionnés avec la branche principale ou toute autre branche cible pertinente. C'est ce qu'on appelle une opération de "merge" ou fusion.

COLLABORATEURS

les collaborateurs sont les individus qui contribuent au projet. Chacun peut avoir des droits d'écriture sur le dépôt (par exemple, push) ou simplement des droits de lecture (pull).

Les collaborateurs travaillent généralement sur des branches distinctes et utilisent des mécanismes tels que les pull requests pour coordonner leurs contributions avec le reste de l'équipe.

DEMANDES DE TIRAGE

Elle permet à un contributeur de proposer des changements (commits) à une branche spécifique du dépôt.

Une fois qu'une pull request est ouverte, d'autres membres de l'équipe peuvent examiner les modifications, poser des questions, suggérer des modifications supplémentaires, et finalement approuver la pull request pour la fusion.

Une fois la pull request approuvée, les modifications peuvent être fusionnées dans la branche cible, souvent la branche principale du projet.

merci