Formation Git (1 jour)

Accéder à la formation : https://jolicampus.com/formations/git

Société JoliCode, 18 Avenue Parmentier 75011 Paris / hello@jolicampus.com / +33 1 43 57 39 11

JoliCampus

Objectif

Vous retrouvez dans cette formation notre retour d'expérience, ainsi qu'une présentation de stratégies pertinentes pour l'utilisation de Git au quotidien : modes de gestion des branches, gestion des conflits, stratégie de merge dans le cadre de gros projets avec de multiples intervenants. À l'issue de cette formation, qui est condensée sur une journée, les participants maîtriseront :

- la gestion des versions avec Git
- l'utilisation avancée des branches avec les feature branches
- les workflows avancés et Git-flow
- l'utilisation de sous-modules et subtrees

Pré-requis

- Utilisation rudimentaire de la ligne de commande

Plan

Introduction à Git et gestion de version

Dans cette section, vous apprendrez pourquoi Git s'est imposé comme l'outil incontournable pour la gestion de version et comment il se distingue des autres systèmes de versioning (VCS). Nous aborderons également son installation et sa configuration pour un environnement de travail optimal.

- Qu'est-ce que Git ?
- Gestion décentralisée et différences avec les autres VCS
- Installation et configuration de GitCréation et organisation d'un dépôt Git
- Dépôt local vs dépôt distant (GitHub, GitLab, Bitbucket)
- Premier projet : initialisation d'un dépôt et premiers commits

Commits, branches et gestion des versions

Ici, vous apprendrez à versionner efficacement votre code en utilisant les fonctionnalités essentielles de Git, comme les commits, les branches et les pull requests. Nous explorerons également les workflows basés sur les branches pour organiser le travail en équipe.

- Gestion des commits : créer, éditer et annuler un commit
- Création et gestion de branches
- Push, pull et fetch : synchroniser avec des dépôts distants
- Pull requests et collaboration avec GitHub/Bitbucket
- Fusionner les branches : fast-forward et merges
- Utilisation avancée des branches : feature branching, hotfixes

Collaboration et résolution des conflits

Durée de la formation :

1 jour

Tarif :

600 € HT / personne

Public

- Développeurs et développeuse
- Chefs et cheffes de projet technique

Pré-requis

- Utilisation rudimentaire de la ligne de commande

Qualiopi processus certifié RÉPUBLIQUE FRANÇAISE

Formation certifiée Qualiopi : Eligible aux financements publics et mutualisés.

Demander un devis

Travailler en équipe avec Git requiert de savoir gérer les conflits et résoudre les divergences dans le code. Cette section se concentre sur les meilleures pratiques pour collaborer efficacement tout en maintenant un historique propre et lisible.

JoliCampus

- Gestion des conflits : identifier et résoudre les conflits de merge
- Rebase vs merge : choisir la bonne approche
- Utiliser Git stash pour gérer les modifications temporaires
- Manipuler l'historique : amender des commits, réécrire l'historique (rebase interactif)
- Rappels sur la gestion des accès et permissions sur les dépôts distants

Workflows avancés et Git-flow

Pour aller plus loin avec Git, nous aborderons ici les workflows avancés, comme Git-flow, et comment structurer vos branches pour une gestion efficace des versions dans le cadre de projets complexes. Nous verrons également comment intégrer des outils de CI/CD pour automatiser les processus.

- Introduction à Git-flow : branches master, develop, feature, release, hotfix
- Choisir un workflow adapté : GitHub Flow, GitLab Flow, Git-flow
- Gestion de versions avec Git tags et releases
- Automatisation avec les hooks Git
- Intégration continue (CI) et déploiement continu (CD) avec Git

Bonnes pratiques, performances et outils

Pour conclure la formation, vous découvrirez les bonnes pratiques pour optimiser l'utilisation de Git au quotidien, ainsi que des outils tiers pour améliorer votre productivité. Nous aborderons aussi la gestion des performances avec des projets de grande envergure.

- Bonnes pratiques Git : commit messages, petites modifications, clean code
- Optimisation des performances Git pour les grands projets (gc, pack)
- Utilisation de sous-modules et subtrees
- Outils complémentaires : GitKraken, Sourcetree, GitLens pour VS Code
- Sécurisation des dépôts : .gitignore, git-crypt, GitHub Actions

Note

Nous pouvons adapter la formation à vos besoins, en mettant l'accent sur un aspect plus particulier du développement. Avant chaque formation, nos formateurs contactent les participants afin de déterminer vos objectifs et savoir s'il faut mettre l'accent sur un sujet particulier. C'est le bon moment, pour vous, d'échanger avec le formateur pour lui permettre de mieux cerner vos objectifs et les aspects que vous aimeriez voir approfondis.