FONDAMENTAUX : HTML, CSS, JS, PYTHON

Git et GitHub : gestion de versions pour débutants

Apprenez à versionner votre code, collaborer sur GitHub et travailler en équipe avec Git. Formation essentielle avant tout parcours de développement.

DURÉE	TARIF HT	NIVEAU	LANGUE	GROUPE	FORMAT
2j (14h)	**490.00 €**	**Débutant**	**Français**	**3-8**	**Formation**

1 PUBLIC VISÉ

- Développeurs débutants souhaitant structurer et sécuriser leur travail dès le départ
- Étudiants en développement web ou logiciel
- Personnes en reconversion souhaitant adopter les pratiques professionnelles du secteur
- Toute personne qui code seule ou en équipe et qui perd du temps à gérer ses fichiers manuellement

2 PRÉREQUIS

- Savoir utiliser un ordinateur et un terminal de base
- Aucune connaissance en programmation requise
- Avoir suivi la formation Configurer son environnement de développeur web est un plus

3 OBJECTIFS PÉDAGOGIQUES

- Comprendre le fonctionnement et l'utilité d'un système de contrôle de version
- Installer et configurer Git sur son poste de travail
- Créer et gérer des dépôts locaux et distants
- Enregistrer l'historique de son travail de manière claire et structurée
- Travailler avec les branches pour isoler ses développements
- Collaborer efficacement via GitHub : fork, pull request, revue de code
- Résoudre des conflits de fusion sereinement

4 PROGRAMME DÉTAILLÉ

Jour 1 (7h) – Git en local : maîtriser l'essentiel

Module 1 – Introduction au contrôle de version (1h30)

Pourquoi versionner son code ? (45min)

- Les problèmes du travail sans versioning : fichiers perdus, écrasements, "version_finale_v3_VRAIMENT_FINALE"
- Ce que Git apporte : historique, retour arrière, travail en parallèle
- Git vs autres systèmes de versioning : SVN, Mercurial
- Les grands concepts : dépôt, commit, branche, fusion

Installation et configuration (45min)

- Installer Git sur Windows, macOS et Linux
- Configurer son identité : nom, email, éditeur par défaut
- Vérifier l'installation et comprendre la configuration globale
- Présentation de l'interface : terminal Git vs interfaces graphiques (VS Code, GitHub Desktop)

Module 2 – Les fondamentaux de Git (3h)

Les trois zones de Git (45min)

- Le répertoire de travail : là où on modifie les fichiers
- La zone de transit : préparer ce qu'on va enregistrer
- Le dépôt local : l'historique permanent
- Comprendre le cycle de vie d'un fichier dans Git

Enregistrer son travail (1h15)

- Initialiser un dépôt dans un projet existant
- Suivre l'état de son dépôt à tout moment
- Ajouter des fichiers à la zone de transit : tout, un fichier, une partie
- Valider ses modifications avec un message de commit clair
- Bonnes pratiques de commit : fréquence, messages, atomicité

Consulter et naviguer dans l'historique (1h)

- Afficher l'historique des commits : résumé et détail
- Comparer deux versions d'un fichier
- Revenir à un état précédent sans perdre l'historique
- Annuler la dernière modification non validée
- Ignorer des fichiers : le fichier .gitignore

Module 3 – Les branches (2h30)

Comprendre les branches (45min)

- Qu'est-ce qu'une branche ? La métaphore de l'arbre
- La branche principale : convention et bonnes pratiques
- Pourquoi travailler en branches ? Isolation, parallélisme, sécurité

Créer et naviguer entre les branches (45min)

- Créer une branche pour une nouvelle fonctionnalité
- Basculer entre les branches
- Lister et supprimer des branches
- Cas pratique : développer une fonctionnalité en isolation

Fusionner et rebaser (1h)

- Fusionner une branche dans une autre : fusion directe et fusion avec commit
- Le rebase : réécrire l'historique pour le garder propre
- Quand utiliser la fusion, quand utiliser le rebase ?
- Cas pratique : intégrer une fonctionnalité terminée dans la branche principale

Jour 2 (7h) – GitHub, collaboration et cas réels

Module 4 – GitHub et les dépôts distants (2h)

Prendre en main GitHub (45min)

- Créer un compte et configurer son profil
- Créer un dépôt distant : public vs privé
- Comprendre l'interface GitHub : code, issues, pull requests, actions
- Authentification sécurisée : clé SSH ou token personnel

Synchroniser son travail (1h15)

- Relier un dépôt local à GitHub
- Envoyer ses commits vers GitHub
- Récupérer les modifications d'un dépôt distant
- Cloner un projet existant depuis GitHub
- Cas pratique : publier son projet local sur GitHub et le partager

Module 5 – Collaboration et travail en équipe (3h)

Contribuer à un projet existant (1h)

- Faire un fork : prendre une copie indépendante d'un projet

Cloner son fork et travailler en local

- Synchroniser son fork avec le projet d'origine
- Cas pratique : forker un dépôt de démonstration et y apporter une modification

Les pull requests (1h)

- Qu'est-ce qu'une pull request ? Le mécanisme de validation du code
- Créer une pull request depuis une branche
- Décrire clairement ses modifications : titre, description, captures d'écran
- Participer à une revue de code : commenter, suggérer, approuver
- Fusionner une pull request et nettoyer les branches

Résolution de conflits (1h)

- Pourquoi les conflits apparaissent-ils ?
- Identifier les fichiers en conflit
- Lire et comprendre les marqueurs de conflit
- Résoudre un conflit manuellement et valider la résolution
- Cas pratique : simulation d'un conflit réel entre deux développeurs

Module 6 – Bonnes pratiques et workflows professionnels (2h)

Workflows Git courants (1h)

- Git Flow : branches de fonctionnalité, release, hotfix
- GitHub Flow : plus simple, adapté aux petites équipes
- Trunk-based development : pour les équipes avancées
- Choisir le workflow adapté à son équipe et son projet

Atelier de synthèse (1h)

- Simulation d'un projet collaboratif complet : deux rôles (mainteneur / contributeur)
- Créer une branche, développer, ouvrir une pull request, relire, fusionner
- Résolution d'un conflit en conditions réelles
- Retour formateur et bonnes pratiques à retenir

5 COMPÉTENCES VISÉES

- Initialiser un dépôt Git et gérer son historique au quotidien
- Créer, naviguer et fusionner des branches de manière autonome
- Publier et synchroniser son travail sur GitHub
- Contribuer à un projet collaboratif via les pull requests
- Résoudre un conflit de fusion sans perdre de données

6 MODALITÉS PÉDAGOGIQUES

Formation délivrée en présentiel ou distanciel (visioconférence). Le formateur alterne entre méthode démonstrative (manipulation Git en direct, commentée pas à pas), méthode interrogative et méthode active (chaque apprenant manipule Git sur son propre poste tout au long de la formation). L'accent est mis sur la pratique réelle et la compréhension des mécanismes plutôt que sur la mémorisation des commandes.

7 MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Support de cours numérique mis à disposition des apprenants
- Fiche de référence des commandes Git essentielles
- Dépôt GitHub de démonstration fourni par le formateur
- Pour le distanciel : visioconférence (Microsoft Teams ou équivalent), partage d'écran, chat en direct
- Accès à la plateforme pédagogique LaPolaris (supports, ressources, émargement)

8 MODALITÉS D'ÉVALUATION

- En cours de formation : manipulations Git en direct à chaque module, cas pratiques corrigés
- En fin de formation : atelier de synthèse — simulation complète d'un workflow collaboratif
- Questionnaire d'auto-évaluation des acquis en fin de parcours

9 CRITÈRES D'ÉVALUATION

- Capacité à initialiser un dépôt et gérer son historique de manière autonome
- Maîtrise des branches : création, navigation, fusion et résolution de conflits
- Publication et synchronisation d'un projet sur GitHub sans aide
- Participation active à un workflow collaboratif via pull request
- Clarté et pertinence des messages de commit produits

10 MODALITÉS DE VALIDATION

Attestation de fin de formation délivrée à l'issue du parcours, conditionnée à une assiduité d'au moins 80 % et à la réalisation de l'atelier de synthèse final. L'attestation précise les objectifs atteints et les compétences acquises.

11 SUIVI ET ACCOMPAGNEMENT

- Feuilles d'émargement signées par demi-journée (présentiel) ou émargement numérique (distanciel)
- Traçabilité des activités pédagogiques réalisées
- Attestation d'assiduité délivrée en fin de formation
- Suivi individuel via la vérification du dépôt GitHub de chaque apprenant en fin de parcours

12 CONDITIONS D'ACCÈS

Formation accessible sur inscription directe, sans prérequis administratif particulier. Le financement peut être pris en charge par l'employeur dans le cadre d'un plan de développement des compétences, ou en autofinancement avec possibilité de paiement en plusieurs fois.

13 DÉLAIS D'ACCÈS

Inscription possible jusqu'à **5 jours ouvrés** avant le début de la session. Pour toute demande urgente, nous contacter directement.

ACCESSIBILITÉ · HANDICAP

Nos formations sont accessibles aux personnes en situation de handicap. Pour toute situation nécessitant un aménagement (matériel, temporel ou pédagogique), nous vous invitons à nous contacter avant l'inscription afin d'étudier les adaptations possibles. Référent handicap : contact@lapolaris.fr

LaPolaris

TÉL. +33762584798

EMAIL contact@lapolaris.fr

WEB lapolaris.fr