



mise en place d'un environnement de développement

Note importante : Ce document est conçu comme un guide de référence, mais n'est pas exhaustif. La capacité à rechercher des informations complémentaires et à résoudre des problèmes de manière autonome est une compétence essentielle pour tout développeur. Si vous rencontrez des difficultés ou si certaines étapes ne fonctionnent pas comme prévu, n'hésitez pas à consulter d'autres ressources (documentation officielle, tutoriels YouTube, forums spécialisés comme Stack Overflow). Apprendre à trouver l'information manquante et à déboguer par vous-même est probablement la compétence la plus précieuse que vous développerez dans votre parcours professionnel.

Installation de l'éditeur de code VS Code

Installation de VS Code

- Téléchargez VS Code depuis le [site officiel](#)
- Suivez les instructions d'installation selon votre système d'exploitation
- Lancez VS Code après l'installation

Présentation de VS Code

- **Interface utilisateur** : zones d'éditeur, explorateur de fichiers, terminal intégré
- **Fonctionnalités principales** :
 - Intellisense (auto-complétion de code)
 - Débogage intégré
 - Contrôle de source Git intégré
 - Terminal intégré
 - Personnalisation avancée
- **Raccourcis clavier** : augmentent votre productivité

Extensions utiles pour VS Code

- Live Server : un serveur local avec rechargement automatique, permet de lire une page web sur votre ordinateur
- Prettier code formatter, formatage automatique basé sur des règles, mise en page
- ESLint, définir des conventions et des règles de code
- Live Share : permet la collaboration en temps réel

Installation de Git

Sur Windows

1. Téléchargez l'installateur depuis git-scm.com
2. Exécutez le fichier .exe téléchargé
3. Suivez les instructions de l'assistant d'installation
4. Vérifiez l'installation en ouvrant Git Bash et en tapant `git --version`

Sur macOS

1. **Méthode Homebrew** (recommandée):

```
brew install git
```

2. **Méthode alternative**: Téléchargez l'installateur depuis git-scm.com

3. Vérifiez l'installation avec `git --version` dans le Terminal

Sur Linux

- **Debian/Ubuntu:**

```
sudo apt updatesudo apt install git
```

- **Fedora:**

```
sudo dnf install git
```

- **Arch Linux:**

```
sudo pacman -S git
```

Vérification de l'installation Git

- Ouvrez un terminal (ou Git Bash sur Windows)
- Exécutez la commande `git --version`
- Vous devriez voir s'afficher la version installée (ex: `git version 2.40.1`)

Inscription sur GitHub

1. Création de compte

- Accédez à github.com
- Cliquez sur "Sign up" en haut à droite
- Remplissez le formulaire avec votre email, mot de passe et nom d'utilisateur
- Suivez les étapes de vérification

2. Configuration de votre profil

- Ajoutez une photo de profil
- Complétez votre bio
- Ajoutez vos informations de contact professionnel

3. Premiers pas avec GitHub

- Créez votre premier dépôt
- Explorez des projets populaires
- Suivez des développeurs qui vous intéressent

Commandes Git essentielles

Configuration initiale de Git

```
git config --global user.name "Votre Nom"  
git config --global user.email "votre.email@exemple.com"
```

Différence entre `git init`, `git clone` et `git remote`

`git init`

- **Usage :** `git init`
- **Fonction :** Initialise un nouveau dépôt Git vide dans le répertoire courant
- **Cas d'utilisation :** Lorsque vous commencez un nouveau projet à partir de zéro
- **Exemple :**

```
mkdir mon-projetcd mon-projetgit init
```

`git clone`

- **Usage :** `git clone <url>`
- **Fonction :** Télécharge une copie complète d'un dépôt distant existant (avec tout l'historique)
- **Cas d'utilisation :** Lorsque vous voulez travailler sur un projet existant
- **Exemple :**

```
git clone https://github.com/utilisateur/depot.git
```

`git remote`

- **Usage :** `git remote add <nom> <url>`

- **Fonction** : Connecte votre dépôt local à un dépôt distant
- **Cas d'utilisation** : Après avoir initialisé un dépôt local avec `git init`, vous souhaitez le connecter à GitHub
- **Exemple** :

```
git remote add origin https://github.com/utilisateur/depot.git
```

Autres commandes Git essentielles

git add

- **Usage** : `git add <fichier>` ou `git add .` (pour tous les fichiers)
- **Fonction** : Prépare les fichiers modifiés pour le commit
- **Exemple** :

```
git add index.htmlgit add .
```

git commit

- **Usage** : `git commit -m "message"`
- **Fonction** : Enregistre les modifications préparées dans l'historique du dépôt
- **Exemple** :

```
git commit -m "Ajout de la page d'accueil"
```

git push

- **Usage** : `git push <remote> <branche>`
- **Fonction** : Envoie les commits locaux vers le dépôt distant
- **Exemple** :

```
git push origin main
```

git pull

- **Usage** : `git pull <remote> <branche>`

- **Fonction** : Récupère les modifications du dépôt distant et les fusionne avec votre branche locale
- **Exemple** :

```
git pull origin main
```

git status

- **Usage** : `git status`
- **Fonction** : Affiche l'état des fichiers (modifiés, ajoutés, supprimés)
- **Exemple** :

```
git status
```

git log

- **Usage** : `git log`
- **Fonction** : Affiche l'historique des commits
- **Exemple** :

```
git loggit log --oneline
```

Flux de travail Git typique

1. Cloner un dépôt ou en initialiser un nouveau dossier en local
2. Effectuer des modifications dans les fichiers
3. Ajouter les fichiers modifiés à l'index avec `git add`
4. Valider les modifications avec `git commit`
5. Synchroniser avec le dépôt distant avec `git push`
6. Récupérer les modifications des collaborateurs avec `git pull`

Protection intellectuelle : Ce cours est intégralement protégé par le droit d'auteur conformément au Code de la propriété intellectuelle français. Toute reproduction, diffusion

ou utilisation non autorisée est strictement interdite.
L'intelligence artificielle a été utilisée uniquement comme outil d'assistance pour la correction orthographique et la reformulation de certaines phrases, sous ma supervision et validation. Le contenu pédagogique et intellectuel reste entièrement ma création originale et personnelle.