

Démarrer avec GitHub

1. 🌟 C'est quoi GitHub ?

GitHub = Un **cloud** pour stocker et partager du code

- Comme Google Drive, mais pour les développeurs
- Sauvegarde automatique de vos projets
- Accessible de n'importe où
- Gratuit pour usage public et privé

2. 📄 Créer un Compte GitHub

2.1. Étape 1 : Inscription

1. Aller sur <https://github.com>
2. Cliquer sur "Sign up" (S'inscrire)
3. Renseigner :
 - Adresse email
 - Mot de passe (fort !)
 - Nom d'utilisateur (visible publiquement)
4. Vérifier l'email
5. Choisir le plan **Free** (gratuit)

2.2. Étape 2 : Profil

- Ajouter une photo de profil
- Remplir votre bio (optionnel)
- Votre profil : <https://github.com/votre-username>

3. 📦 Notion de Repository (Repo)

3.1. Qu'est-ce qu'un Repository ?

Un **repository** = Un **dossier de projet** sur le cloud

- Contient tous vos fichiers de code
- Historique des modifications
- Peut être **public** (visible par tous) ou **privé** (juste vous)

3.2. Analogie

Repository = Dossier Google Drive

- ├─ fichiers de code
- ├─ images
- ├─ documentation
- └─ historique des versions

4. + Créer un Repository

4.1. Via l'interface web GitHub

1. Cliquer sur le bouton vert **New** ou **+** en haut à droite
2. Remplir :
 - **Repository name** : `mon-premier-projet`
 - **Description** : "Mon projet de test" (optionnel)
 - **Public** ou **Private** : choisir la visibilité
 - ☒ Cocher "Add a README file" (recommandé)
 - ☒ Cocher "Add .gitignore" : choisir le template (Python, Node, etc.)
3. Cliquer sur "Create repository"

Résultat : Votre projet est en ligne ! 🚀

5. 📁 Ajouter des Fichiers (Interface Web)

5.1. Méthode 1 : Upload direct


1. Ouvrir votre repository
2. Cliquer sur "Add file" → "Upload files"
3. Glisser-déposer vos fichiers
4. Ajouter un message : "Ajout de mes fichiers"
5. Cliquer sur "Commit changes"

5.2. Méthode 2 : Créer un fichier en ligne

1. Cliquer sur "Add file" → "Create new file"
2. Nommer le fichier : `index.html`
3. Écrire le contenu
4. Cliquer sur "Commit new file"

5.3. Méthode 3 : Éditer un fichier existant

1. Cliquer sur le fichier

2. Cliquer sur l'icône **crayon**  (Edit)
3. Modifier le contenu
4. Cliquer sur "Commit changes"

6. 📄 Télécharger un Repository

6.1. Télécharger le ZIP


1. Ouvrir le repository
2. Cliquer sur le bouton vert "Code"
3. Cliquer sur "Download ZIP"
4. Décompresser sur votre ordinateur

7. 🔄 Synchroniser VSCode avec GitHub

7.1. Pourquoi synchroniser ?


- **Sauvegarder** vos paramètres VSCode (thème, extensions, raccourcis)
- **Retrouver** la même configuration sur un autre ordinateur
- **Partager** votre setup avec d'autres machines

7.2. Étape 1 : Activer la synchronisation

1. **Ouvrir VSCode**
2. Cliquer sur l'icône **engrenage**  en bas à gauche
3. Cliquer sur "Turn on Settings Sync..." (Activer la synchronisation)
4. **Choisir ce qui sera synchronisé :**
 - ☒ Settings (Paramètres)
 - ☒ Keyboard Shortcuts (Raccourcis)
 - ☒ Extensions
 - ☒ User Snippets (Extraits)
 - ☒ UI State (Interface)
5. Cliquer sur "Sign in & Turn on"
6. Choisir "Sign in with GitHub"
7. Autoriser VSCode à accéder à votre compte GitHub

C'est fait ! ☒ Vos paramètres sont maintenant sauvegardés sur GitHub

7.3. Étape 2 : Récupérer sur un autre ordinateur

1. Installer VSCode sur le nouvel ordinateur
2. Ouvrir VSCode
3. Cliquer sur l'icône **engrenage** 
4. Cliquer sur "Turn on Settings Sync..."
5. Cliquer sur "Sign in & Turn on"
6. Se connecter avec le **même compte GitHub**

Magie ! 🪄 Toutes vos configurations se téléchargent automatiquement !

7.4. Vérifier l'état de la synchro

- Icône **cloud** en bas à gauche de VSCode
- Cliquer dessus pour voir l'état
- **Cloud avec ✓** = Tout est synchronisé
- **Cloud avec ⏸** = Synchronisation en cours

8. 🔑 Authentification GitHub

8.1. Token d'accès personnel (si demandé)

1. Aller sur GitHub → **Settings** (Paramètres)
2. **Developer settings** (en bas du menu gauche)
3. **Personal access tokens** → **Tokens (classic)**
4. **Generate new token** (classic)
5. Donner un nom : "VSCode Sync"
6. Cocher **repo** et **user**
7. Cliquer sur **Generate token**
8. **COPIER** le token (il ne sera montré qu'une fois !)
9. Le coller dans VSCode quand demandé

9. 📁 Structure d'un Repository Type

```
mon-projet/
├── README.md          # Description du projet
├── .gitignore         # Fichiers à ignorer
├── src/              # Code source
│   ├── index.html
│   └── style.css
├── images/           # Images
└── docs/             # Documentation
```

10. 🎯 Bonnes Pratiques

10.1. README.md

- **Toujours** créer un README.md
- Expliquer à quoi sert le projet
- Ajouter des instructions d'installation

10.2. Noms de fichiers

- Pas d'espaces → Utiliser `-` ou `_`
-  `mon-projet.js`
-  `mon projet.js`

10.3. Organisation

- Un repository = Un projet
- Créer des dossiers pour organiser
- Mettre un `.gitignore` pour ignorer les fichiers temporaires

11. Public vs Privé

Type	Visibilité	Usage
Public	Tout le monde peut voir	Projets open source, portfolio
Private	Seulement vous (+ collaborateurs)	Projets personnels, professionnels

Astuce : Vous pouvez changer de public à privé (et vice-versa) à tout moment dans Settings

12. À Retenir

1. **GitHub** = Cloud pour code
2. **Repository** = Dossier projet en ligne
3. **Commit** = Sauvegarder une version
4. **Sync VSCode** = Sauvegarder vos paramètres
5. **Gratuit** pour usage personnel

13. Pour Aller Plus Loin

- **GitHub Desktop** : Application pour gérer GitHub sans ligne de commande
- **GitHub Pages** : Héberger un site web gratuitement
- **GitHub Gist** : Partager des snippets de code rapidement