

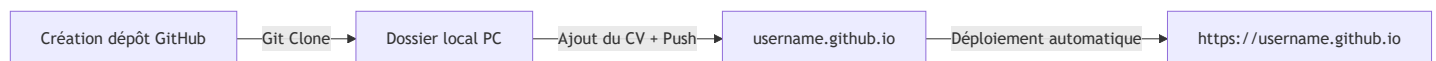
Déployer son CV sur GitHub Pages (Dépôt Principal)

1. Le concept : Le dépôt spécial `username.github.io`

Sur GitHub, chaque utilisateur a droit à un **dépôt principal d'utilisateur**.

- **username.github.io** : L'adresse du site sera directement <https://username.github.io/>

C'est la solution idéale pour héberger votre CV ou votre portfolio principal.



2. Les prérequis

1. Votre identifiant GitHub exact (exemple : `alex-dev`).
2. Vos fichiers HTML/CSS de CV prêts sur votre ordinateur.
3. **Git** installé et configuré sur votre ordinateur.

3. La pratique : Étape par étape

Étape 1 : Créer le dépôt sur GitHub

1. Allez sur [GitHub](#).
2. Cliquez sur le bouton **+** (en haut à droite) puis **New repository**.
3. **Repository name** : Entrez exactement votre nom d'utilisateur suivi de `.github.io`.
 - Exemple : Si votre pseudo est `alex-dev`, nommez le dépôt `alex-dev.github.io`.
4. **Visibility** : Choisissez **Public**.
5. Cochez la case **Add a README file** (facilite le premier clonage).
6. Cliquez sur **Create repository**.

Étape 2 : Cloner le dépôt en local

1. Sur la page de votre nouveau dépôt, cliquez sur le bouton vert **Code**.
2. Copiez l'URL HTTPS (ex:

```
[https://github.com/username/username.github.io.git](https://github.com/username/username.github.io.git)
```

3. Ouvrez votre terminal, placez-vous dans le dossier où vous stockez vos projets, puis tapez :

```
git clone https://github.com/username/username.github.io.git
```

4. Entrez dans le dossier créé :

```
cd username.github.io
```

Étape 3 : Ajouter les fichiers du site et publier

1. Copiez tous les fichiers de votre site (votre `index.html` , le dossier CSS, les images...) à l'intérieur de ce nouveau dossier.
2. Dans votre terminal, enregistrez et envoyez les modifications :

```
git add .  
git commit -m "feat: ajout des fichiers du site"  
git push origin main
```

Étape 4 : Vérifier l'activation de GitHub Pages

Pour ce type de dépôt spécifique, **GitHub Pages s'active automatiquement** dès le premier `push` .

1. Sur GitHub, allez dans **Settings > Pages** sur votre dépôt.
2. Vérifiez que la branche est bien réglée sur `main` et le dossier sur `/ (root)` .
3. Attendez 1 à 2 minutes.
4. Votre site est accessible à l'adresse : <https://username.github.io>

```
[ Push sur main ] —> [ Activation auto ] —> https://username.github.io
```

⚠ Pièges fréquents à éviter

- **Faute dans le nom du dépôt** : Le nom doit correspondre à 100% à votre pseudo GitHub. Une erreur de frappe empêchera la publication automatique.
 - **Fichier principal** : Le fichier d'accueil doit obligatoirement s'appeler `index.html` (tout en minuscules) et se trouver directement dans le dossier, pas dans un sous-dossier.
 - **Mises à jour** : Pour modifier votre CV en ligne par la suite, il vous suffira d'enregistrer vos fichiers, puis de faire `git add .`, `git commit -m "update"` et `git push`.
-

Aller plus loin : Les sous-sites (sites projets)

Les dépôts nommés différemment (comme `mon-cv` ou `projet-1`) sont considérés comme des **projets secondaires**. Si vous activez github-pages sur ces dépôts, ils viennent se placer automatiquement après la barre oblique (/) de votre domaine principal.

Cela vous permet de publier autant de sites web que vous le souhaitez à partir d'un seul compte GitHub. Par exemple :

- <https://username.github.io/> (votre site principal).
- <https://username.github.io/mon-cv/> (votre CV).
- <https://username.github.io/jeu-javascript/> (un projet de jeu web).

Voici le paragraphe à ajouter, à placer idéalement juste après l'**Étape 4 : Vérifier l'activation de GitHub Pages** dans votre cours :

Aller plus loin : Les workflows GitHub Actions pour Pages

En arrière-plan, GitHub utilise un outil d'automatisation appelé **GitHub Actions** pour transformer votre code en site web.

- **Fonctionnement automatique** : Par défaut, GitHub crée son propre workflow invisible pour publier vos fichiers statiques (HTML/CSS/JS).
- **Personnalisation (`.github/workflows/`)** : Si vous souhaitez contrôler précisément le déploiement (par exemple pour compiler du code Sass, tester votre HTML ou utiliser un générateur de site comme Jekyll), vous pouvez créer votre propre fichier de configuration `.github/workflows/pages.yml`.

Ce fichier YAML indique à GitHub les étapes exactes à exécuter à chaque fois que vous faites un `git push` sur la branche `main` :

```
name: Deploy GitHub Pages

on:
  push:
    branches: ["main"]

jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout du code
        uses: actions/checkout@v4

      - name: Configuration de GitHub Pages
        uses: actions/configure-pages@v4

      - name: Envoi des fichiers
        uses: actions/upload-pages-artifact@v3
        with:
          path: '.'

      - name: Déploiement sur GitHub Pages
        uses: actions/deploy-pages@v4
```

En résumé : Pour un CV en HTML/CSS pur, l'option par défaut de GitHub suffit largement. Le fichier YAML devient utile dès que votre projet nécessite des étapes de compilation automatisées avant la mise en ligne.