

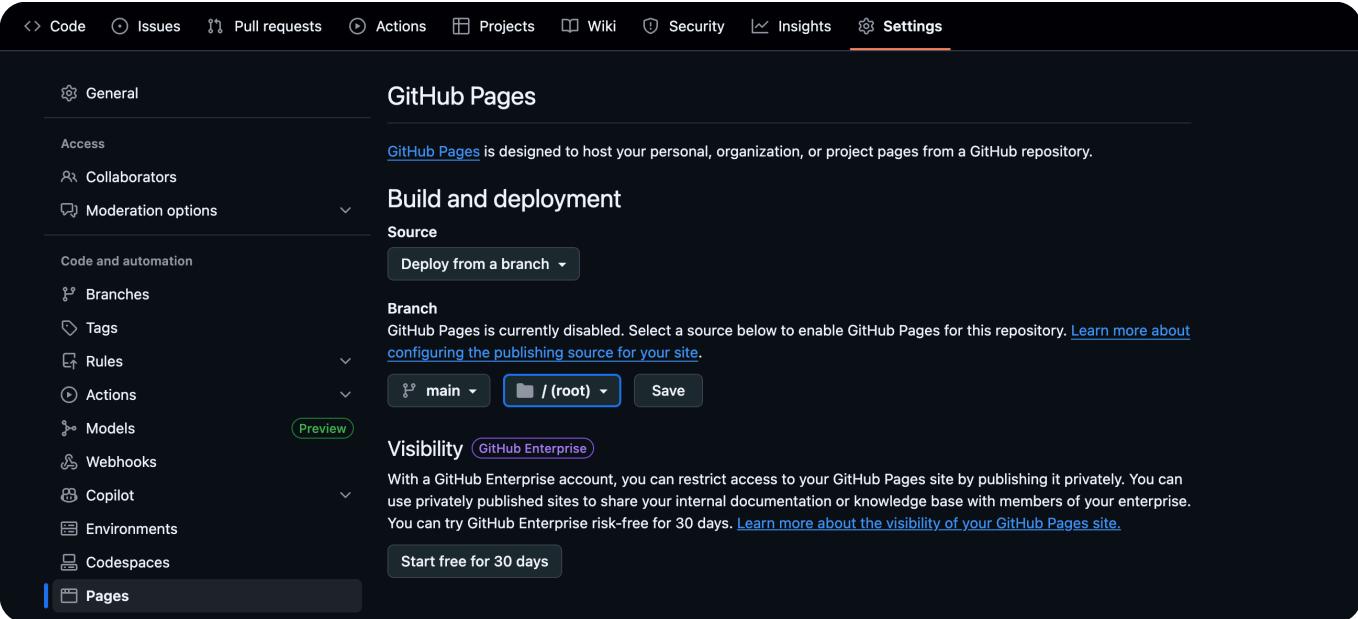
# CRÉATION & DÉPLOIEMENT D'UN SITE STATIQUE

< BOOTSTRAP W-INT-100 />



# CRÉATION & DÉPLOIEMENT D'UN SITE STATIQUE

Pour réaliser votre premier site web en ligne, vous allez créer un **site statique** et l'héberger grâce à **GitHub Pages**.



The screenshot shows the GitHub Pages settings page for a repository. On the left, there's a sidebar with various repository settings like General, Access, Collaborators, and Moderation options. The main area is titled "GitHub Pages" and contains the following sections:

- Build and deployment**:
  - Source**: A dropdown menu set to "Deploy from a branch".
  - Branch**: A note stating "GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository." with a link to "Learn more about configuring the publishing source for your site".
  - Two dropdown menus: "main" and "/ (root)".
  - A "Save" button.
- Visibility**: A note for GitHub Enterprise users about restricting access to their GitHub Pages site.
- A "Start free for 30 days" button.

Dans ce projet, vous apprendrez à :

- ✓ créer un **dépôt GitHub public** ;
- ✓ utiliser **Git** pour versionner et publier votre travail ;
- ✓ héberger un site web statique accessible publiquement en ligne.



Avant de commencer, assurez-vous d'avoir un environnement propre : un dossier clair, un dépôt initialisé et un index HTML minimal.

Une bonne préparation facilite un déploiement sans erreur.

# 1. Pré-requis

Avant de démarrer, vous devez disposer de :

- ✓ un **compte GitHub** fonctionnel ;
- ✓ **Git** installé sur votre ordinateur ;
- ✓ un éditeur de code (VSCode recommandé).

# 2. Créer un dépôt GitHub

Rendez-vous sur : <https://github.com>

1. Cliquez sur **New repository**
2. Donnez un nom à votre dépôt (ex. : `site-statique`)
3. Choisissez la visibilité **Public** □
4. Cochez **Add a README file**
5. Cliquez sur **Create repository**



Votre dépôt est maintenant créé.

## Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).  
Required fields are marked with an asterisk (\*).

### 1 General

Owner \*



dylan-epitech

Repository name \*



klivio

klivio is available.

Great repository names are short and memorable. How about [legendary-dollop](#)?

### Description

0 / 350 characters

### 2 Configuration

#### Choose visibility \*

Choose who can see and commit to this repository



Public

#### Start with a template

Templates pre-configure your repository with files.



No template

#### Add README

READMEs can be used as longer descriptions. [About READMEs](#)



Off

#### Add .gitignore

.gitignore tells git which files not to track. [About ignoring files](#)



No .gitignore

#### Add license

Licenses explain how others can use your code. [About licenses](#)



No license



Create repository

### 3. Cloner votre dépôt en local

Dans votre terminal :

```
git clone https://github.com/votre-nom-utilisateur/site-statique.git  
cd site-statique
```

Vous pouvez maintenant créer votre fichier index.html ainsi que vos dossiers CSS, images, etc.

### 4. Activer GitHub Pages

1. Ouvrez votre dépôt sur GitHub
2. Cliquez sur Settings
3. Dans le menu de gauche, sélectionnez Pages
4. Dans la section Build and deployment, configurez :

```
Source : Deploy from a branch  
Branch : main  
Folder : choisissez le dossier contenant votre index.html  
(ex : /, /html-css, etc.)
```

5. Cliquez sur Save

Après quelques secondes, GitHub génère automatiquement votre site l'url ressemblera à cela :

**<https://votre-nom-utilisateur.github.io/site-statique>**

## 5. Mettre à jour et publier votre site

À chaque modification, exécutez :

```
git add .  
git commit -m "Mise à jour du site"  
git push
```



GitHub Pages se charge automatiquement du déploiement.  
Votre site sera mis à jour en ligne en quelques secondes.



Si votre site ne se met pas à jour :  
vérifiez que index.html se trouve dans le bon dossier ;  
vérifiez que vous poussez sur la branche main ;  
attendez quelques secondes : GitHub Pages n'est pas instantané.

## 6. Résultat attendu

À la fin de ce projet, vous devez obtenir :

1. un dépôt **GitHub public** propre et correctement structuré ;
2. une page web statique en ligne via **GitHub Pages** ;
3. un lien public **fonctionnel** menant à votre site ;
4. votre projet **HTML/CSS** consultable en ligne ;
5. votre projet **HTML/Tailwind** également publié s'il fait partie de l'exercice.



Votre dépôt GitHub devient votre vitrine professionnelle.

Prenez soin de vos commits, de votre structure et de votre présentation : cela comptera dans vos futurs projets.

v 1.0.0



{EPITECH}