

# INTRODUCTION À JEKYLL

< BOOTSTRAP />



# INTRODUCTION À JEKYLL

## Héberger un site web statique avec Jekyll & GitHub Pages

Ce bootstrap vous fournit les concepts clés et les orientations nécessaires pour publier un site web statique généré avec **Jekyll**, rédigé en partie en **Markdown**, et déployé automatiquement à l'aide de **GitHub Pages**.

Ce document n'est pas un tutoriel complet.  
Vous êtes censé explorer la documentation, expérimenter et faire vos propres choix techniques.

### Prérequis

Avant de commencer, assurez-vous d'avoir :

- ✓ un compte GitHub
- ✓ des connaissances de base en Git
- ✓ de la curiosité

Aucune connaissance backend n'est requise pour ce projet.

## Sites statiques & GitHub Pages

**GitHub Pages** permet d'héberger gratuitement des sites web statiques.

Un site web statique peut être :

- ✓ constitué de fichiers HTML / CSS / JS simples
- ✓ ou généré automatiquement à partir de fichiers sources (Markdown, templates, etc.)

GitHub Pages prend en charge **Jekyll** nativement.

Cela signifie que GitHub peut :

- ✓ lire votre projet Jekyll
- ✓ générer le HTML final
- ✓ le publier automatiquement

## Markdown & Jekyll

### Markdown

---

Markdown est un langage de balisage léger utilisé pour écrire du contenu.

```
# Titre
## Sous-titre

- élément de liste
- autre élément

[lien](https://example.com)
```

Markdown est :

- ✓ lisible
- ✓ facile à maintenir
- ✓ couramment utilisé dans la documentation et les sites statiques

## Jekyll

---

Jekyll est un générateur de sites statiques.

Il vous permet de :

- ✓ rédiger du contenu en Markdown
- ✓ organiser des pages et des collections
- ✓ utiliser des layouts et des templates
- ✓ générer un site web complet

Un projet Jekyll typique contient :

- ✓ \_config.yml
- ✓ index.md ou index.html
- ✓ des dossiers tels que \_posts, \_projects, \_layouts, assets, etc.

Vous êtes censé explorer la manière dont les pages sont créées, comment la navigation fonctionne et comment le contenu est structuré.

## GitHub Pages + Jekyll

Lorsque GitHub Pages détecte un projet Jekyll :

- ✓ il construit le site automatiquement
- ✓ il convertit le Markdown en HTML
- ✓ il applique le thème et les layouts choisis

Vous pouvez configurer ce comportement à l'aide de :

- ✓ \_config.yml
- ✓ les paramètres de GitHub Pages
- ✓ éventuellement GitHub Actions

## Rédiger du contenu avec Markdown

Au lieu d'écrire tout en HTML, vous allez :

- ✓ écrire les pages à l'aide de fichiers .md
- ✓ structurer le contenu avec des titres
- ✓ utiliser des images, des liens et des listes
- ✓ séparer le contenu de la présentation

Exemples de pages en Markdown :

- ✓ page d'accueil (index.md)
- ✓ page « à propos »
- ✓ pages de projets

La manière dont vous organisez votre contenu fait partie du projet.

## Logique de déploiement (vue d'ensemble)

Votre workflow devrait ressembler approximativement à ceci :

1. Vous modifiez des fichiers Markdown ou de configuration
2. Vous committez et poussez vos changements
3. GitHub construit le site automatiquement
4. Le site mis à jour est publié en ligne

Vous n'êtes pas censé téléverser manuellement des fichiers HTML.

Comprendre ce pipeline est plus important que mémoriser des commandes.

## Ce que vous devez explorer par vous-même

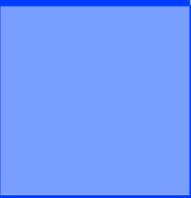
Ce bootstrap omet volontairement certains détails.

Vous êtes encouragé à rechercher :

- ✓ comment Jekyll gère les pages et les collections
- ✓ comment les menus de navigation sont construits
- ✓ comment personnaliser un thème
- ✓ comment GitHub Pages déclenche un build
- ✓ comment le Markdown est rendu
- ✓ comment la configuration impacte le site final

## Documentation utile

- ✓ **GitHub Pages** — <https://docs.github.com/en/pages>
- ✓ **Documentation Jekyll** — <https://jekyllrb.com/docs/>
- ✓ **GitHub Pages + Jekyll** — <https://docs.github.com/en/pages/setting-up-a-github-pages-site-with-jekyll>
- ✓ **Syntaxe Markdown** — <https://www.markdownguide.org/basic-syntax/>



{EPITECH}