

## Programmation et projet encadré - L7TI005

Pages GitHub

---

Yoann Dupont [yoann.dupont@sorbonne-nouvelle.fr](mailto:yoann.dupont@sorbonne-nouvelle.fr)

Pierre Magistry [pierre.magistry@inalco.fr](mailto:pierre.magistry@inalco.fr)

2024-2025

Université Sorbonne-Nouvelle  
INALCO  
Université Paris-Nanterre

# Les pages GitHub

Les pages GitHub<sup>1</sup> (ou gh-pages) permettent de créer et déployer un site web lié à un dépôt git.

C'est une solution simple pour créer des sites statiques lorsqu'on en a besoin<sup>2</sup>.

Dans le cadre de ce cours, nous allons voir comment utiliser les gh-pages pour créer un site pour le projet de groupe.

---

<sup>1</sup>Documentation officielle : <https://pages.github.com>

<sup>2</sup>Il est également possible de créer sa page perso en page GitHub.

## Avant de commencer

Les gh-pages ont une part de magie<sup>3</sup> pour les rendre simples à utiliser. Plusieurs façons de faire existent, on vous montrera la plus cohérente avec l'esprit du module.

Nous allons procéder de la façon suivante :

- exemple de résultat attendu illustré sur un dépôt bidon ;
- exemple de création déroulé sur votre dépôt personnel ;
- potentiellement : librairie utilisable pour agencer facilement votre site.

---

<sup>3</sup>Définition : [https://en.wikipedia.org/wiki/Magic\\_\(programming\)](https://en.wikipedia.org/wiki/Magic_(programming))

## Démonstration

On retourne sur le dépôt git-along<sup>4</sup>

---

<sup>4</sup>vous pourrez le cacher après la démo si vous voulez, mais en attendant, il doit être public.

# Créer un site pour son dépôt I

Il n'y a besoin que de deux éléments pour créer un site à partir d'un dépôt GitHub :

- un fichier index à la racine du dépôt, ici : `index.html`<sup>5</sup> ;
- dire à GitHub de transformer un dépôt en site.

---

<sup>5</sup>On peut créer un site en markdown, mais ça demande plus d'étapes... et de magie.

# Créer un site pour son dépôt I

Il n'y a besoin que de deux éléments pour créer un site à partir d'un dépôt GitHub :

- un fichier `index.html` à la racine du dépôt, ici : `index.html`<sup>5</sup> ;
- dire à GitHub de transformer un dépôt en site.

Un fichier `index.html` est disponible sur le git.

- téléchargez-le ;
- modifier le titre dans le `head` ;
- poussez-le sur votre propre dépôt

---

<sup>5</sup>On peut créer un site en markdown, mais ça demande plus d'étapes... et de magie.

# Créer un site pour son dépôt I

Il n'y a besoin que de deux éléments pour créer un site à partir d'un dépôt GitHub :

- un fichier `index` à la racine du dépôt, ici : `index.html`<sup>5</sup> ;
- dire à GitHub de transformer un dépôt en site.

Un fichier `index.html` est disponible sur le git.

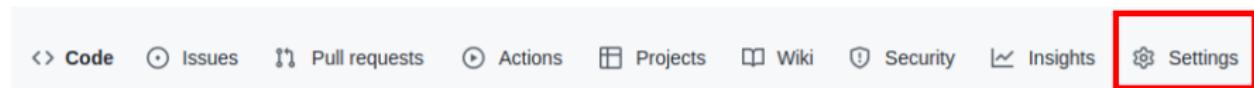
- téléchargez-le ;
- modifier le titre dans le `head` ;
- pushez-le sur votre propre dépôt

Nous allons maintenant dire à GitHub de déployer le site depuis le dépôt.

---

<sup>5</sup>On peut créer un site en markdown, mais ça demande plus d'étapes... et de magie.

## Créer un site pour son dépôt II



Aller sur le dépôt sur GitHub puis cliquer dans l'onglet *Settings* à droite.

# Créer un site pour son dépôt III

## General

Access

Collaborators

Moderation options



Code and automation

Branches

Tags

Actions



Webhooks

Environments

Pages

Sur la barre de navigation à gauche, cliquer sur la section Pages (en bas sur l'image).

# Créer un site pour son dépôt IV

## Build and deployment

### Source

Deploy from a branch ▾

- Section *Build And Deployment*
- Source → *Deploy from a branch*
- Branch → *main*

### Branch

GitHub Pages is currently disabled. Select a source below

main ▾

/ (root) ▾

Save

# C'est fini !

Après avoir sauvegardé, il faudra attendre 1 à 3 minutes et le site sera disponible à l'adresse : <https://nomutilisateur.github.io/nomdepot>.

Il faudra également attendre ce temps après chaque modification faite au site avant d'en voir les effets.

Maintenant, il ne reste plus qu'à créer un site pour le projet et le remplir !

# C'est fini !

Après avoir sauvegardé, il faudra attendre 1 à 3 minutes et le site sera disponible à l'adresse : <https://nomutilisateur.github.io/nomdepot>.

Il faudra également attendre ce temps après chaque modification faite au site avant d'en voir les effets.

Maintenant, il ne reste plus qu'à créer un site pour le projet et le remplir !

## Le nom de la page web : exemple

le dépôt <https://github.com/YoannDupont/git-along>  
donne la page <https://YoannDupont.github.io/git-along>

(C'est pour l'exemple, ce dépôt sera supprimé à un moment.)

## Prévisualiser le résultat avant publication

Vous pouvez prévisualiser le résultat sur un navigateur en ouvrant tout simplement le fichier `index.html` dans un navigateur. Cette solution fonctionne dans notre cas car nous faisons un site web statique.

## Prévisualiser le résultat avant publication

Vous pouvez prévisualiser le résultat sur un navigateur en ouvrant tout simplement le fichier `index.html` dans un navigateur. Cette solution fonctionne dans notre cas car nous faisons un site web statique.

À présent, à chaque push, le site sera mis à jour !

## Exercice : finaliser le mini-projet

Pour finaliser le mini-projet le but est de construire deux pages HTML et de les publier sur GitHub en tant que pages. Les étapes à suivre sont :

1. créer une page d'accueil `index.html` qui utilise le style Bulma avec :
  - une présentation rapide du mini-projet
  - le lien vers la page HTML qui contient le tableau de résultat

## Exercice : finaliser le mini-projet

Pour finaliser le mini-projet le but est de construire deux pages HTML et de les publier sur GitHub en tant que pages. Les étapes à suivre sont :

1. créer une page d'accueil `index.html` qui utilise le style Bulma avec :
  - une présentation rapide du mini-projet
  - le lien vers la page HTML qui contient le tableau de résultat
2. le tableau sera à créer d'abord sans style particulier, il faudra déployer la page GitHub une première fois pour vérifier que le lien fonctionne via la page GitHub du miniprojet

## Exercice : finaliser le mini-projet

Pour finaliser le mini-projet le but est de construire deux pages HTML et de les publier sur GitHub en tant que pages. Les étapes à suivre sont :

1. créer une page d'accueil `index.html` qui utilise le style Bulma avec :
  - une présentation rapide du mini-projet
  - le lien vers la page HTML qui contient le tableau de résultat
2. le tableau sera à créer d'abord sans style particulier, il faudra déployer la page GitHub une première fois pour vérifier que le lien fonctionne via la page GitHub du miniprojet
3. une fois qu'on accède bien à la page HTML qui contient le tableau, on modifie son CSS directement jusqu'à obtenir un résultat plus élégant

## Exercice : finaliser le mini-projet

Pour finaliser le mini-projet le but est de construire deux pages HTML et de les publier sur GitHub en tant que pages. Les étapes à suivre sont :

1. créer une page d'accueil `index.html` qui utilise le style Bulma avec :
  - une présentation rapide du mini-projet
  - le lien vers la page HTML qui contient le tableau de résultat
2. le tableau sera à créer d'abord sans style particulier, il faudra déployer la page GitHub une première fois pour vérifier que le lien fonctionne via la page GitHub du miniprojet
3. une fois qu'on accède bien à la page HTML qui contient le tableau, on modifie son CSS directement jusqu'à obtenir un résultat plus élégant
4. une fois le style finalisé, mettre à jour le script de façon pour générer directement la page avec le style Bulma que vous lui avez créé

## Exercice : finaliser le mini-projet

Pour finaliser le mini-projet le but est de construire deux pages HTML et de les publier sur GitHub en tant que pages. Les étapes à suivre sont :

1. créer une page d'accueil `index.html` qui utilise le style Bulma avec :
  - une présentation rapide du mini-projet
  - le lien vers la page HTML qui contient le tableau de résultat
2. le tableau sera à créer d'abord sans style particulier, il faudra déployer la page GitHub une première fois pour vérifier que le lien fonctionne via la page GitHub du miniprojet
3. une fois qu'on accède bien à la page HTML qui contient le tableau, on modifie son CSS directement jusqu'à obtenir un résultat plus élégant
4. une fois le style finalisé, mettre à jour le script de façon pour générer directement la page avec le style Bulma que vous lui avez créé

On déploiera le site (= push) à chaque étape jusqu'à obtenir un résultat fonctionnel. Une fois le tout terminé, on crée le tag `miniprojet-3`.

À partir de la semaine prochaine, nous passerons en travail de groupe exclusivement.

Pensez à avoir un dépôt git pour votre groupe et à inviter les autres membres pour qu'ils/elles puissent faire des commits et push sur le dépôt.