

# Histoire de GitHub



**2008**

Fondation par Tom Preston-Werner, Chris Wanstrath et PJ Hyett. Lancement officiel le 10 avril.

**2011**

GitHub dépasse le million de dépôts et devient la plus grande forge de code au monde.

**2012**

Levée de fonds de 100M\$ auprès d'Andreessen Horowitz. Lancement de GitHub Enterprise.

**2015**

GitHub atteint 10 millions de dépôts. Lancement de GitHub Desktop 1.0.

**2018**

Acquisition par Microsoft pour 7,5 milliards de dollars. Nat Friedman devient PDG.

**2019-2023**

Lancement de GitHub Actions, Packages, Codespaces et Copilot. Plus de 100M d'utilisateurs.

# Importance de GitHub

## Collaboration

Facilite la collaboration entre développeurs du monde entier, permettant le travail simultané sur les mêmes projets.

## Contrôle de version

Système Git intégré qui permet de suivre les modifications, revenir en arrière et gérer différentes versions du code.

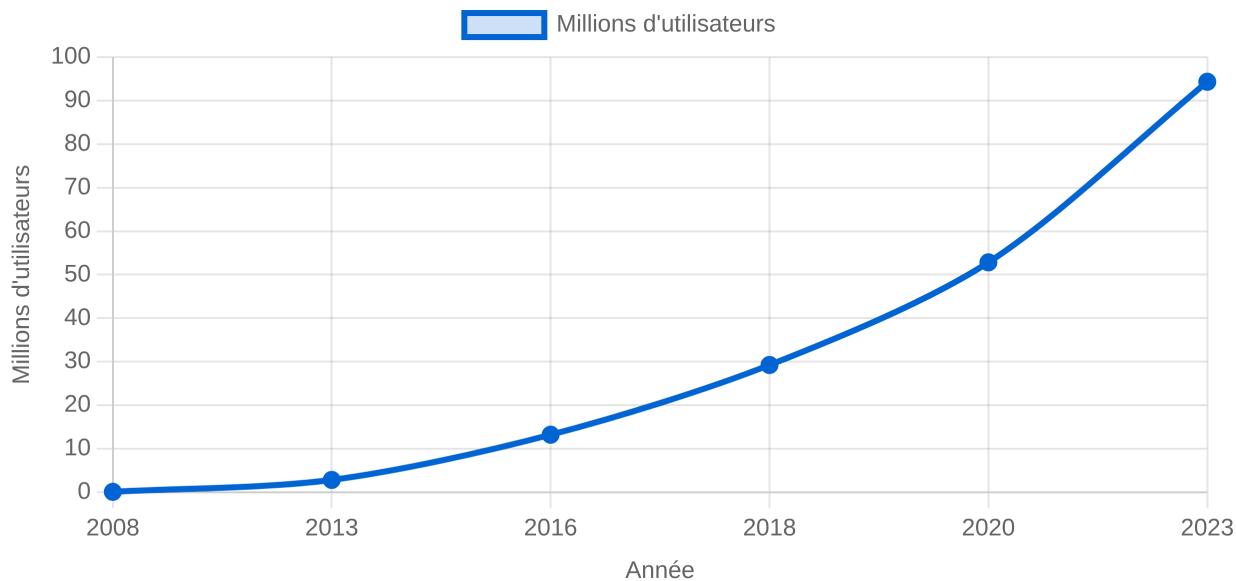
## Communauté open source

Plus grande plateforme d'hébergement de projets open source, favorisant le partage de connaissances et l'innovation.

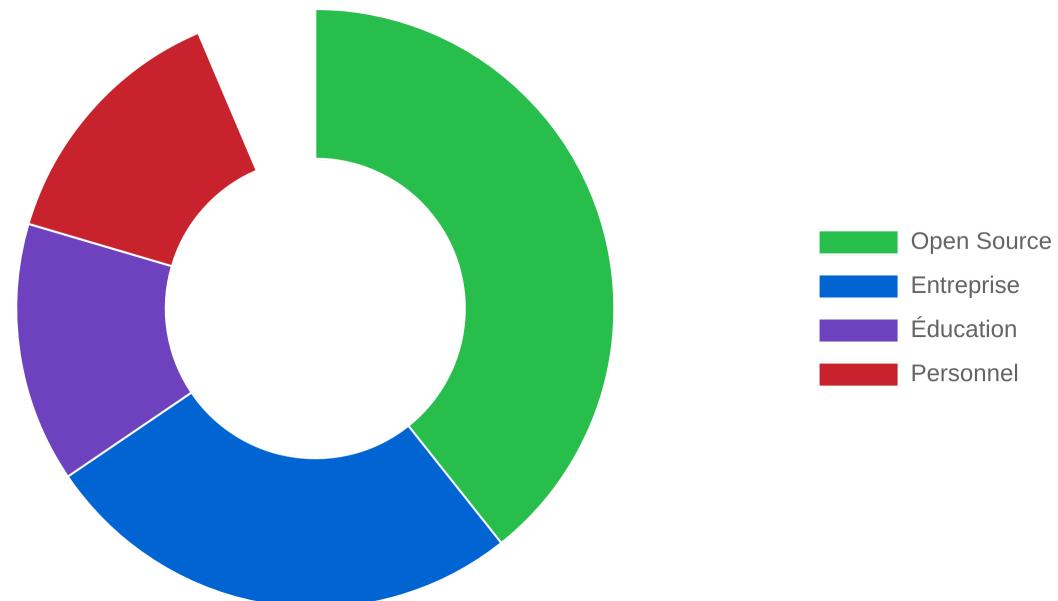
## Visibilité professionnelle

Sert de portfolio pour les développeurs et facilite le recrutement dans l'industrie technologique.

Croissance des utilisateurs GitHub



Répartition des projets sur GitHub



# Caractéristiques principales

## Repositories

Espaces de stockage pour vos projets où tout le code est organisé, documenté et partagé avec d'autres développeurs.

## Pull Requests

Mécanisme permettant de proposer des modifications au code, de les examiner et de les discuter avant de les fusionner.

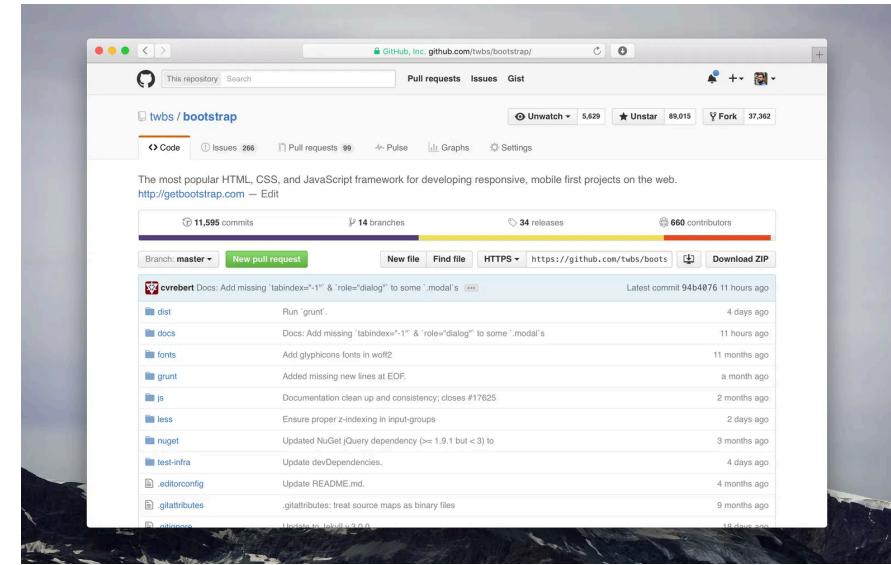
## Issues

Système de suivi des problèmes, des améliorations et des tâches pour gérer efficacement le développement de projets.

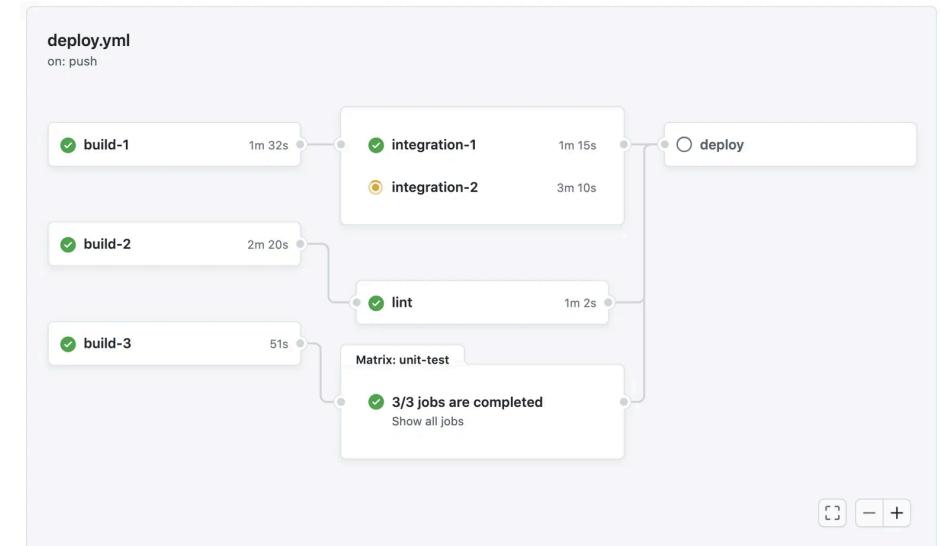
## GitHub Actions

Outil d'automatisation permettant de créer des workflows CI/CD personnalisés directement dans votre repository.

## Repository GitHub



## GitHub Actions Workflow

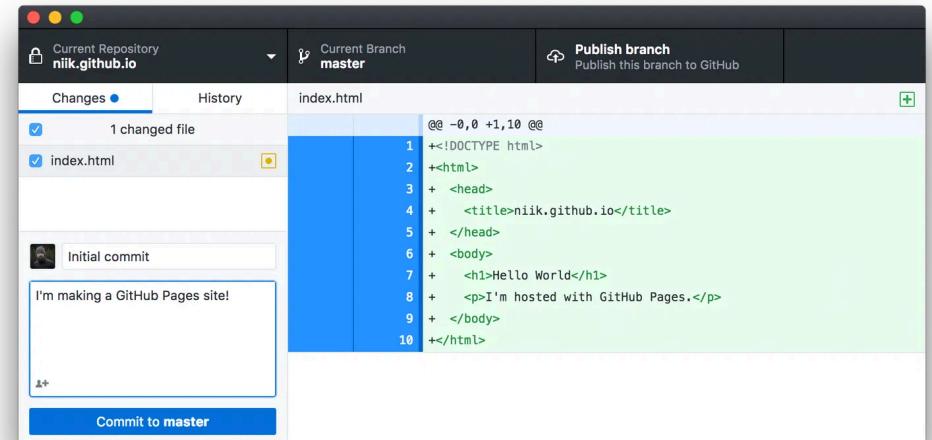


# Déploiement de projets

## GitHub Pages

Service d'hébergement gratuit pour sites web statiques directement depuis un dépôt GitHub.

- 1 Créer un dépôt nommé `username.github.io`
- 2 Pousser le code HTML/CSS/JS dans le dépôt
- 3 Activer GitHub Pages dans les paramètres du dépôt
- 4 Accéder au site via `username.github.io`



A screenshot of the GitHub desktop application interface. The top bar shows "Current Repository: niik.github.io" and "Current Branch: master". A "Publish branch" button is visible. The main area shows a commit history with one entry: "Initial commit" containing the message "I'm making a GitHub Pages site!". Below the commit message is a diff viewer showing the contents of "index.html". The diff shows the following code:

```
@@ -0,0 +1,10 @@
<!DOCTYPE html>
<html>
<head>
<title>niik.github.io</title>
</head>
<body>
<h1>Hello World</h1>
<p>I'm hosted with GitHub Pages.</p>
</body>
</html>
```

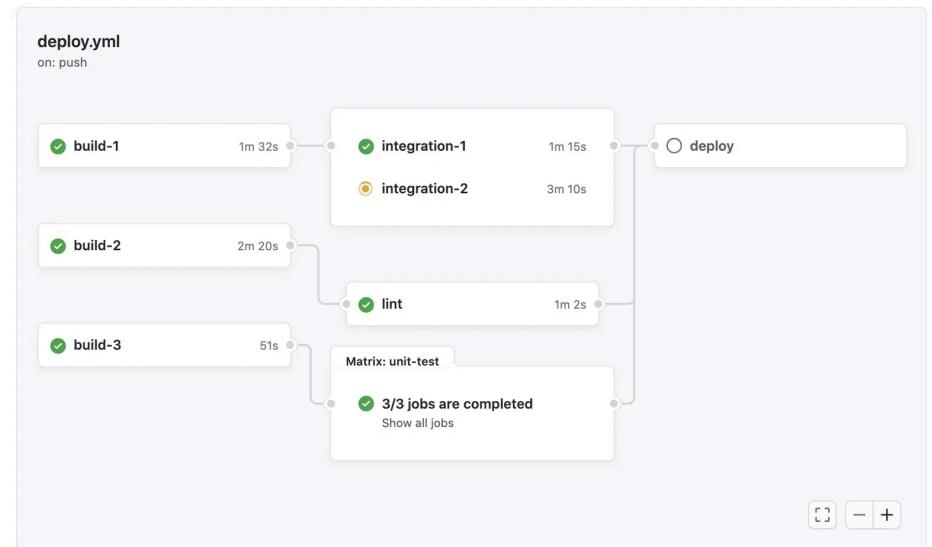
At the bottom of the commit view is a blue "Commit to master" button.

## GitHub Actions

Automatisation du workflow CI/CD pour déployer des applications sur différentes plateformes.

- 1 Créer un fichier workflow YAML dans `.github/workflows/`
- 2 Définir les étapes de build, test et déploiement
- 3 Configurer les secrets pour les clés d'API
- 4 Pousser les modifications pour déclencher le workflow

## GitHub Actions Workflow



# Déploiement manuel de dossier sur GitHub

## 1 Initialiser un dépôt Git local

Naviguez vers votre dossier de projet et initialisez un dépôt Git local.

```
$ cd /chemin/vers/votre/projet  
$ git init
```

## 2 Ajouter les fichiers au suivi

Ajoutez tous les fichiers de votre projet au suivi Git.

```
$ git add .
```

## 3 Créer un commit

Enregistrez les modifications avec un message descriptif.

```
$ git commit -m "Premier commit"
```

## 4 Lier au dépôt distant

Connectez votre dépôt local à un dépôt GitHub distant.

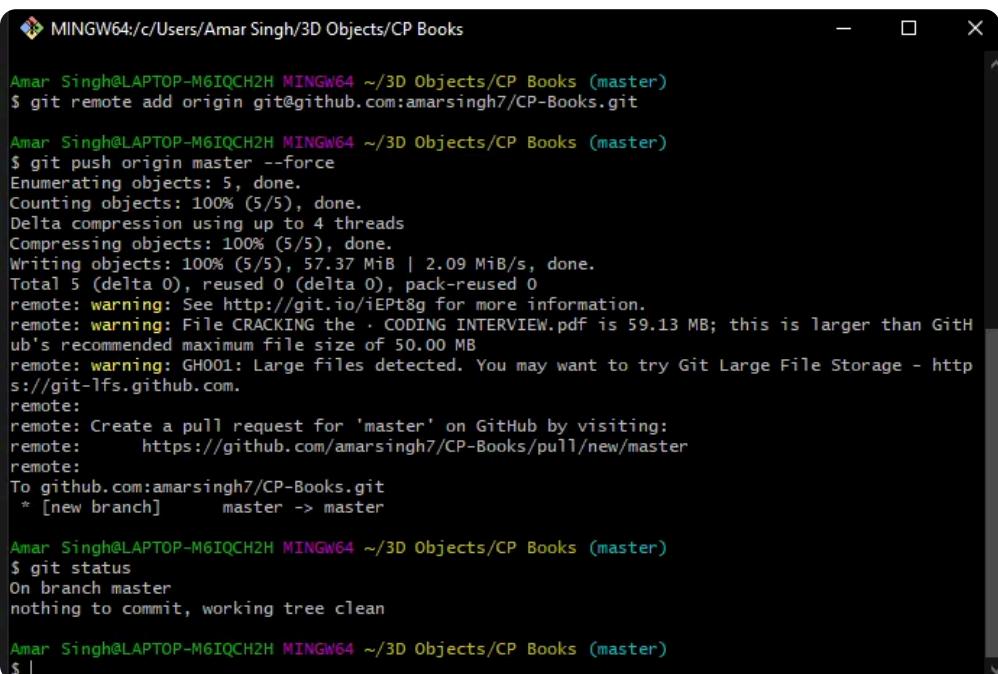
```
$ git remote add origin  
https://github.com/username/repository.git
```

## 5 Pousser les modifications

Envoyez vos modifications vers GitHub.

```
$ git push -u origin main
```

```
1 git:  
2   git add .  
3   git commit -m "$m"  
4   git push -u origin master  
5
```

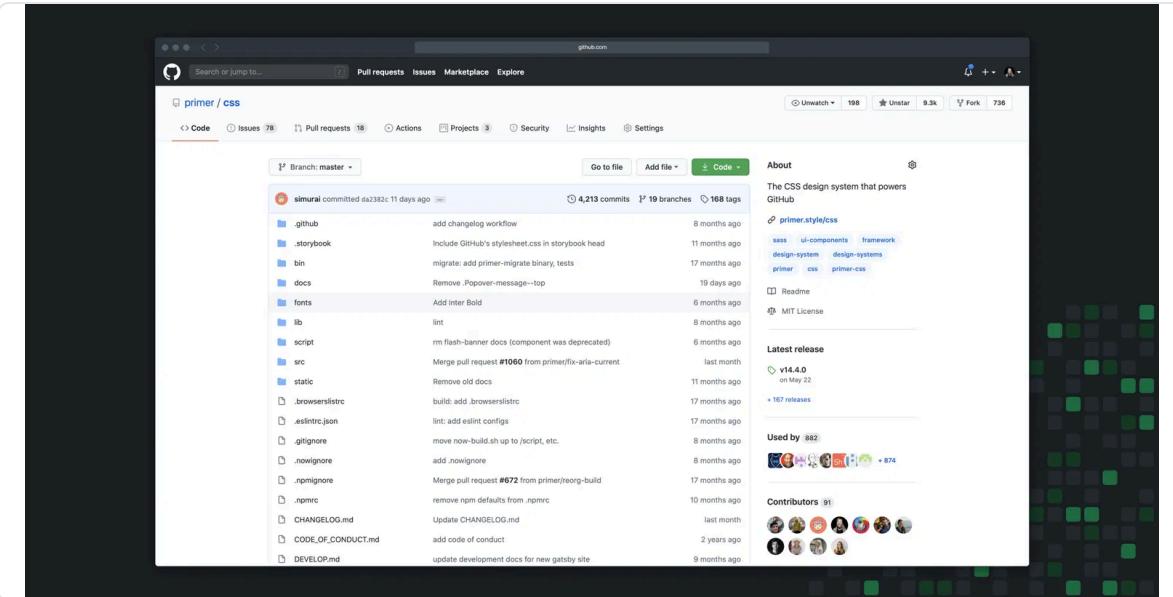


The terminal window shows the following sequence of commands and their output:

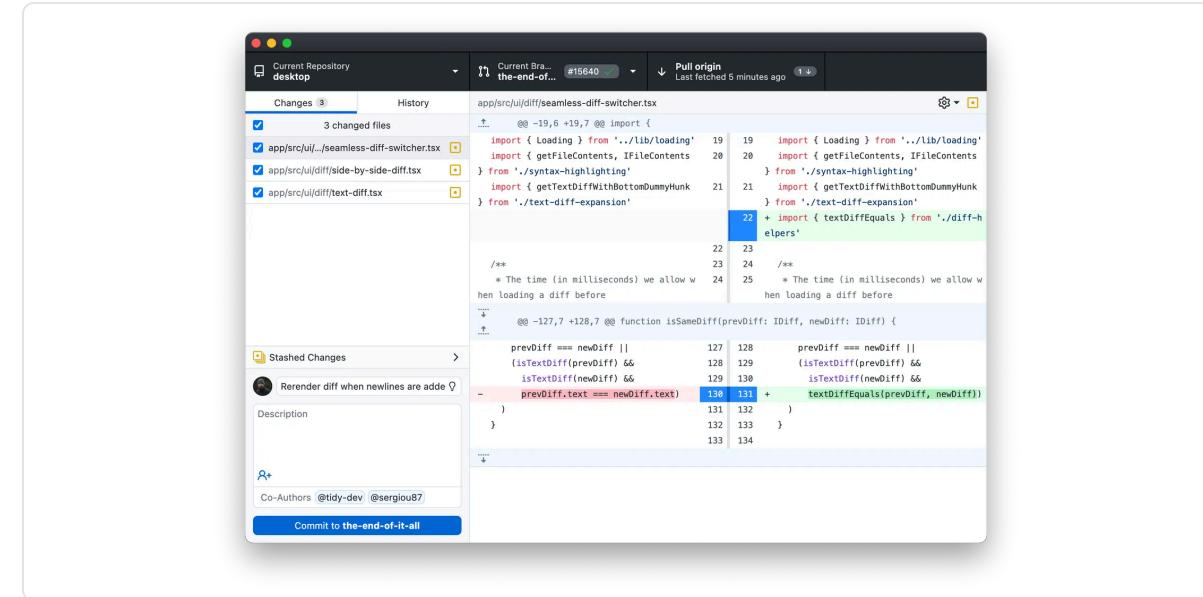
```
MINGW64:/c/Users/Amar Singh/3D Objects/CP Books  
Amar Singh@LAPTOP-M6IQCH2H MINGW64 ~/3D Objects/CP Books (master)  
$ git remote add origin git@github.com:amarsingh7/CP-Books.git  
Amar Singh@LAPTOP-M6IQCH2H MINGW64 ~/3D Objects/CP Books (master)  
$ git push origin master --force  
Enumerating objects: 5, done.  
Counting objects: 100% (5/5), done.  
Delta compression using up to 4 threads  
Compressing objects: 100% (5/5), done.  
Writing objects: 100% (5/5), 57.37 MiB | 2.09 MiB/s, done.  
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0  
remote: warning: See http://git.io/iEPt8g for more information.  
remote: warning: File CRACKING THE . CODING INTERVIEW.pdf is 59.13 MB; this is larger than GitHub's recommended maximum file size of 50.00 MB  
remote: warning: GH001: Large files detected. You may want to try Git Large File Storage - https://git-lfs.github.com.  
remote:  
remote: Create a pull request for 'master' on GitHub by visiting:  
remote:     https://github.com/amarsingh7/CP-Books/pull/new/master  
remote:  
To github.com:amarsingh7/CP-Books.git  
 * [new branch]      master -> master  
  
Amar Singh@LAPTOP-M6IQCH2H MINGW64 ~/3D Objects/CP Books (master)  
$ git status  
On branch master  
nothing to commit, working tree clean  
  
Amar Singh@LAPTOP-M6IQCH2H MINGW64 ~/3D Objects/CP Books (master)  
$
```

# Interface GitHub

## Interface Web



## GitHub Desktop



📁 Explorateur de fichiers

.TODO Projets et tableaux Kanban

🔗 Insights et statistiques

⎇ Gestion des branches

❗ Issues et Pull Requests

⚙️ Paramètres du dépôt

⌚ Historique des modifications

⬆️ Push/Pull simplifiés

⟳ Synchronisation automatique

</> Visualisation des changements

➕ Création de branches

↗️ Ouverture dans éditeur externe

# Configuration de GitHub

## 👤 Configuration du profil

Personnalisez votre profil avec une photo, une bio et vos informations professionnelles pour améliorer votre visibilité.

## 🛡 Paramètres de sécurité

Activez l'authentification à deux facteurs (2FA) et gérez vos clés SSH pour sécuriser votre compte.

## 🔧 Configuration des repositories

Définissez les règles de branches, les modèles d'issues et les workflows d'actions pour standardiser vos projets.

## 🔔 Notifications

Configurez les notifications par email ou web pour rester informé des activités importantes sur vos projets.

## Paramètres du profil GitHub

The screenshot shows the GitHub profile settings interface. At the top, there are tabs for Overview, Repositories, Projects, and Packages, with Overview selected. Below the tabs is a large circular profile picture of the GitHub mascot, Mona Lisa Octocat. To the right of the profile picture is a bio section with the handle @octocato and the text "Hi there! I'm currently working on something cool! I'm currently learning with help from docs.github.com Ask me about GitHub". Below the bio is a "Pinned" section containing three repository cards: "atom" (Forked from atom/atom), "vscode" (Forked from microsoft/vscode), and "JavaScript".

## Paramètres d'un repository GitHub

The screenshot shows a GitHub repository page for "blacklight2000 / test-repo-travis". The page includes sections for "Code", "Issues", "Pull requests", "Wiki", and "Graphs". It displays 3 commits, 2 branches, 0 releases, and 1 contributor. The commits listed are: ".gitignore" (first commit, 13 days ago), ".travis.yml" (first commit, 13 days ago), "lib\_mod.py" (po, 11 hours ago), and "main.py" (first commit, 13 days ago). The repository has 1 star and 0 forks.

# Astuces et bonnes pratiques

## Gestion des branches

- ✓ Utilisez **Git Flow** pour organiser vos branches (main, develop, feature/, release/, hotfix/).

- ✓ Créez des branches spécifiques pour chaque fonctionnalité ou correction.

```
git checkout -b feature/nouvelle-fonctionnalite
```

- ✓ Supprimez les branches après leur fusion pour garder le dépôt propre.

## Commits efficaces

- ✓ Rédigez des messages de commit clairs et descriptifs.

```
git commit -m "Ajoute la validation des formulaires"
```

- ✓ Faites des commits fréquents et de petite taille pour faciliter le suivi des changements.

- ✓ Utilisez des préfixes conventionnels : **feat:**, **fix:**, **docs:**, **style:**, **refactor:**.

## Collaboration

- ✓ Utilisez les **Pull Requests** pour la revue de code avant de fusionner les modifications.

- ✓ Commentez votre code et documentez votre projet avec un README.md complet.

- ✓ Utilisez les **Issues** pour suivre les bugs et les fonctionnalités à développer.

- ✓ Configurez des **templates** pour les issues et pull requests.

## Sécurité et productivité

- ✓ Utilisez un **.gitignore** adapté à votre projet pour éviter de pousser des fichiers sensibles.

- ✓ Configurez des **GitHub Actions** pour automatiser les tests et le déploiement.

- ✓ Utilisez **GitHub CLI** pour interagir avec GitHub depuis votre terminal.

```
gh pr create --title "Nouvelle fonctionnalité" --body "Description"
```

- ✓ Activez **Dependabot** pour maintenir vos dépendances à jour et sécurisées.