
Atelier 1 Démarrer le projet avec GitHub



A un haut niveau, GitHub est un site web et un service de cloud qui aide les développeurs à stocker et à gérer leur code, ainsi qu'à suivre et contrôler les modifications qui lui sont apportées. Pour comprendre exactement ce qu'est GitHub, vous devez connaître deux principes liés :

- Contrôle de version
- Git

Dans cet atelier, nous allons d'abord expliquer ces deux principes. Ensuite, nous en apprendrons plus sur GitHub.

Qu'est-ce que le Contrôle de Version ?

Le contrôle de version aide les développeurs à suivre et à gérer les modifications apportées au code d'un projet logiciel.

Le contrôle de version permet aux développeurs de travailler en toute sécurité à travers les branchements et les fusions.

Avec le branchement, un développeur duplique une partie du code source (appelé le référentiel). Le développeur peut alors apporter des modifications en toute sécurité à cette partie du code sans affecter le reste du projet.

Ensuite, une fois que le développeur a réussi à faire fonctionner correctement sa partie du code, il ou elle peut fusionner ce code avec le code source principal pour le rendre officiel.

Tous ces changements sont ensuite suivis et peuvent être annulés si nécessaire.

Qu'est-ce Que Git ?

Git est un système de contrôle de version open-source spécifique créé par Linus Torvalds en 2005.

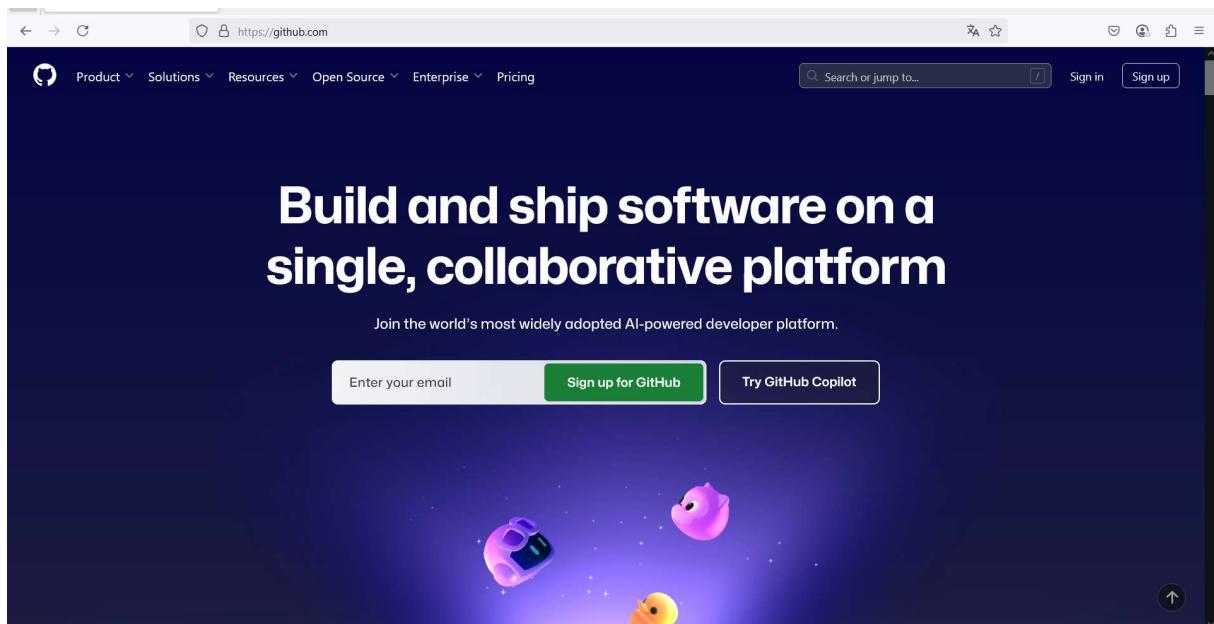
Concrètement, Git est un système de contrôle de version distribué, ce qui signifie que l'ensemble de la base du code et de l'historique est disponible sur l'ordinateur de chaque développeur, ce qui permet des branchements et une fusion facile.

Créez un compte GitHub

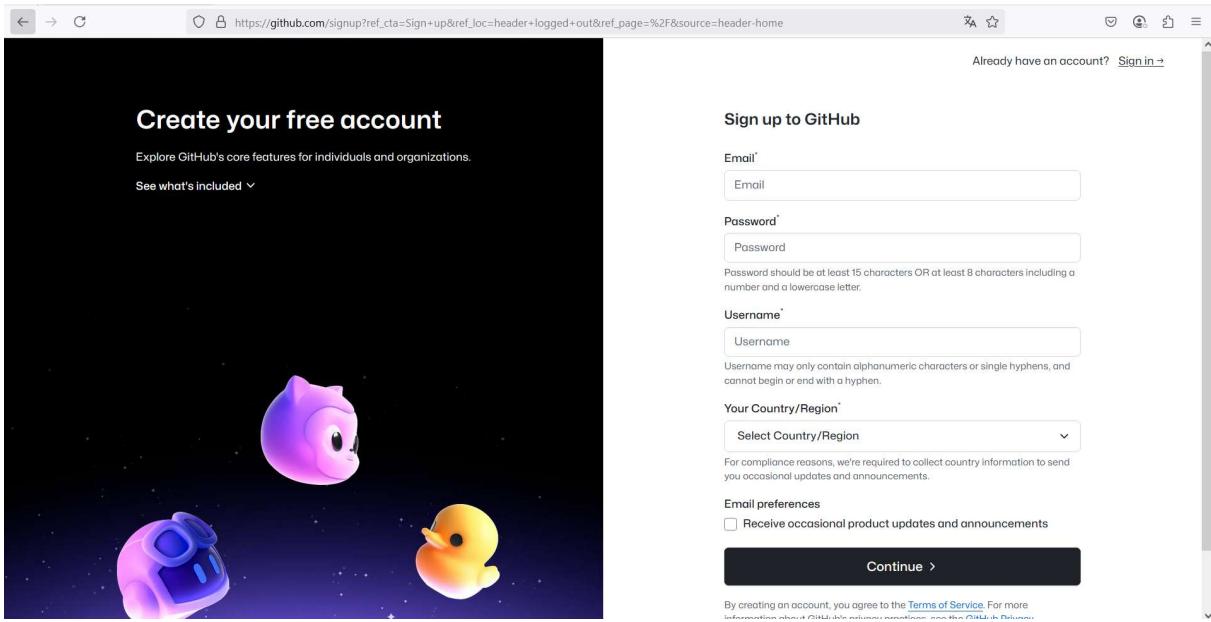
GitHub est un service en ligne qui va vous permettre d'héberger vos dépôts distants.

Pour créer votre compte GitHub, rendez-vous sur la page d'accueil et cliquez sur Sign up.

<https://github.com/>



On vous demandera alors de renseigner un e-mail, un mot de passe et un nom d'utilisateur.



Un code de vérification vous sera envoyé sur votre adresse e-mail afin de confirmer votre identité.

Et voilà, vous êtes à présent inscrit sur GitHub ! Par défaut, GitHub est gratuit. Mais sachez qu'il existe également des offres payantes si vous décidez de passer à la vitesse supérieure.

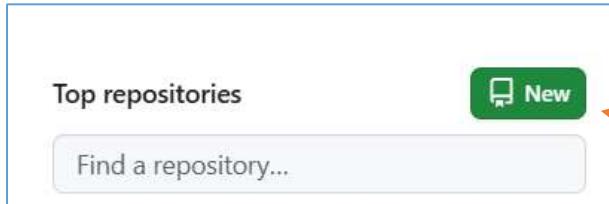
Votre tableau de bord

Vous pouvez consulter votre tableau de bord personnel pour :

- Suivre les problèmes et extraire les demandes sur lesquelles vous travaillez ou que vous suivez.
- Accéder à vos principaux repositories et pages d'équipe.
- Rester à jour sur les activités récentes des organisations et des repositories auxquels vous êtes abonné.

L'interface repository

L'interface Repositories est l'emplacement où vous pourrez créer et retrouver vos dépôts existants.

A screenshot of the GitHub "Create a new repository" form. The URL in the address bar is "github.com/new". The form has a title "Create a new repository". It asks for the "Repository name" which is "myRepoProject". Below that, there's a "Description (optional)" field and a section for choosing between "Public" and "Private". The "Public" option is selected. At the bottom, there's a checkbox for "Add a README file".

Dans le champ Repository name, saisir le nom du projet (sans espace et pas de caractères accentués), la description n'est pas obligatoire, il faut ensuite choisir une portée pour le projet :

Public - tout le monde pourra voir le projet

Privé - contrôler qui peut voir le projet en envoyant une invitation à chaque personne

Une fois terminé, on aura une interface comme ici qui te montre les premiers pas pour publier le projet sur ce dépôt :

The screenshot shows a GitHub repository page for 'myRepoProject'. At the top, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation is a search bar and several quick access buttons. The main content area has two sections: 'Start coding with Codespaces' (with a 'Create a codespace' button) and 'Add collaborators to this repository' (with a 'Invite collaborators' button). A 'Quick setup' section provides instructions for setting up the repository on desktop or via HTTPS/SSH, along with a command-line setup script:

```
echo "# myRepoProject" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/espace2021/myRepoProject.git
```

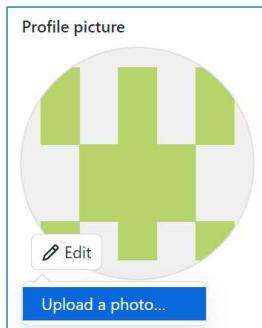
Configuration et paramétrage d'un compte

Accédez aux paramètres de votre compte en utilisant le lien en haut à droite de la fenêtre :

The screenshot shows the GitHub user profile sidebar for 'espace2021'. It includes links for Set status, Your profile, Your repositories, Your Copilot, Your projects, Your stars, Your gists, Your organizations, Your enterprises, and Your sponsors. At the bottom, there are links for Try Enterprise (Free), Feature preview, and Settings, where 'Settings' is highlighted.

The screenshot shows the GitHub Settings page for the user 'espace2021'. The left sidebar is titled 'Public profile' and includes sections for Account, Appearance, Accessibility, Notifications, Access (Billing and plans, Emails, Password and authentication, Sessions, SSH and GPG keys, Organizations, Enterprises, Moderation), and Code, planning, and automation. The main area is titled 'Public profile' and contains fields for Name, Public email (with a note about email privacy), Bio (with a note about @mentioning others), Pronouns (set to 'Don't specify'), and URL. A 'Profile picture' section shows a placeholder image of a green cross on a white background, with an 'Edit' button and a 'Upload a photo...' button.

Ensuite, si vous le souhaitez, vous pouvez remplacer l'avatar généré pour vous par une image de votre choix. Sélectionnez la section « Profile » (profil) (au-dessus de la section « SSH Keys ») et cliquez sur « Upload new picture » (télécharger une nouvelle image).



Github utilise les adresses électroniques pour faire correspondre les commits Git aux utilisateurs. Si vous utilisez plusieurs adresses électroniques dans vos commits et que vous souhaitez que GitHub les relie correctement, vous devez ajouter toutes les adresses que vous avez utilisées dans la section « Emails » (adresses électroniques) de la section d'administration.

The screenshot shows the GitHub Settings sidebar. The 'Emails' section is highlighted with a blue bar at the bottom. Other sections visible in the sidebar include Public profile, Account, Appearance, Accessibility, Notifications, Access (Billing and plans), and Code, planning, and automation.

The screenshot shows the GitHub Settings page for the user 'espace2021'. The left sidebar has sections like Public profile, Account, Appearance, Accessibility, Notifications, Access, Billing and plans, and Emails (which is selected). Under Emails, there are two entries: 'sandrahammamitlili@gmail.com - Primary' and 'sandra_h_t@yahoo.fr'. Each entry has a description and a trash icon. Below the entries is a 'Add email address' section with a text input and an 'Add' button.

L'adresse du haut est vérifiée et définie comme adresse principale, c'est-à-dire que ce sera l'adresse utilisée pour vous envoyer toutes les notifications.

La deuxième adresse est non encore vérifiée, ce qui signifie que vous ne pouvez pas en faire votre adresse principale.

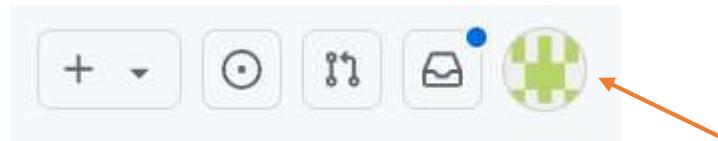
La seconde adresse quand elle est vérifiée peut donc aussi être définie comme adresse principale si on l'échange avec la première.

Si GitHub détecte une de ces adresses dans des messages de validation dans n'importe quel dépôt du site, il les reliera à votre compte utilisateur.

Votre profil

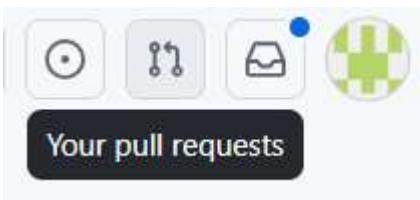
Sur votre profil, vous pourrez éditer vos informations, mais aussi voir le total de vos contributions sur les différents projets.

Les contributions sont toutes les actions sur des repositories que vous allez effectuer. Que ce soient vos repositories, ceux d'autres personnes ou des repositories publics.



The screenshot shows two main sections of a GitHub profile. The top section is a sidebar with a blue header containing the user's icon and name 'espace2021'. Below the header, there are several menu items with icons: Set status (smiley face), Your profile (person), Your repositories (code repository), Your Copilot (robot), Your projects (grid), Your stars (star), Your gists (script), Your organizations (building), Your enterprises (globe), and Your sponsors (heart). The 'Your profile' item is highlighted with a light gray background. The bottom section is the user's profile overview. It features a large green profile picture placeholder. A 'ProTip!' message encourages updating the profile with name, location, and a picture. Below the tip are tabs for Overview (selected), Repositories (0), Projects (0), Stars (0), Followers (0), and Following (0). The 'Popular repositories' section is empty, stating 'You don't have any public repositories yet.' On the left, a 'Set your status' button is visible. At the bottom, it says 'Joined 6 minutes ago' and has an 'Edit' button. To the right, a contribution calendar shows one contribution in March of the previous year. Contribution settings are available via a dropdown arrow.

L'onglet Pull requests



L'onglet Pull requests, quant à lui, permet de faire des demandes de modifications réalisées sur le code.

Les pull requests (ou demandes de pull), vous permettent d'informer les autres utilisateurs des modifications que vous avez appliquées à une branche d'un repository sur GitHub, et que vous voulez fusionner avec le code principal.

The screenshot shows a browser window with the URL github.com/pulls. The page title is "Pull Requests". At the top, there are navigation icons and a search bar containing the query "is:open is:pr author:espace2021 archived:false". Below the search bar, there are tabs for "Created", "Assigned", "Mentioned", and "Review requests", with "Created" being the active tab. It also shows "0 Open" and "0 Closed" pull requests. To the right of the search bar are filters for "Visibility", "Organization", and "Sort". The main content area displays a message: "No results matched your search. You could search [all of GitHub](#) or try an [advanced search](#)". A note at the bottom says "ProTip! `nomilestone` will show everything without a milestone".

© 2025 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact](#) [Manage cookies](#) [Do not share my personal information](#)

La fonctionnalité Explore

Via Explore, vous pourrez trouver de nouveaux projets open source sur lesquels travailler, en parcourant les projets recommandés, en vous connectant à la communauté GitHub et en recherchant des repositories par sujet ou par libellé.

The screenshot shows the GitHub sidebar menu. The menu items are: Home, Issues, Pull requests, Projects, Discussions, Codespaces, Copilot, Explore (which is highlighted with a blue background), and Marketplace.

The screenshot shows the GitHub Explore page. On the left, there are two repository cards: 'unicode-org / icu4x' and 'Azure / azure-rest-api-specs'. Both cards include details like star counts (1.5k and 2.8k respectively), issue counts, pull request counts, and discussion counts. On the right, there is a sidebar titled 'Trending developers' which lists three developers: 'liam' (with a profile picture of a person), 'Azure SDK Bot' (with a Microsoft logo icon), and 'Sebastian Raschka' (with a profile picture of a man). Below the developer list is a link 'See more trending developers >'. The overall interface is light-colored with blue and grey accents.

Installer Git sur l'ordinateur

Visiter le lien :

<https://git-scm.com/downloads>

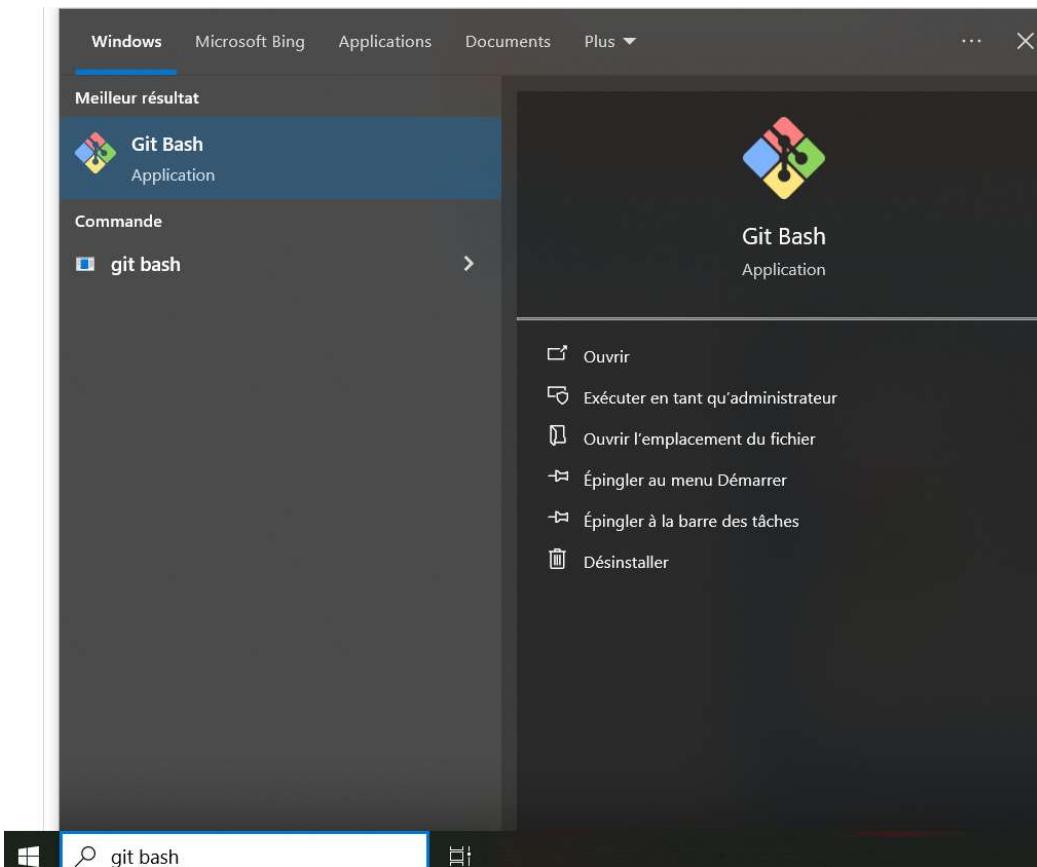
The screenshot shows the 'Downloads' section of the git-scm.com website. It features a large image of a computer monitor displaying a software interface with the text 'Latest source Release 2.49.0' and a 'Download for Windows' button. Below this, there are three main download categories: 'macOS', 'Windows', and 'Linux/Unix'. Each category has a small icon next to its name. To the right of the download sections, there are two smaller boxes: one for 'GUI Clients' and one for 'Logos'. The 'GUI Clients' box contains text about built-in tools and links to 'View GUI Clients →'. The 'Logos' box contains text about various Git logos and a link to 'View Logos →'. The overall design is clean with a white background and a mix of dark and light gray shades.

Choisissez et téléchargez la version de Git qui correspond à votre installation : MacOS, Windows ou Linux/Unix.

Exécutez le fichier que vous venez de télécharger.

Appuyez sur Suivant à chaque fenêtre puis sur Installer. Lors de l'installation, laissez toutes les options par défaut, elles conviennent bien.

Git Bash est l'interface permettant d'utiliser Git en ligne de commande.



Ouvrir

Git Bash se lance.

```
MINGW64:/c/Users/user
user@Sandra MINGW64 ~ $ |
```

Vérifie l'installation avec :

```
git --version
```

Voici un résumé succinct des commandes Git les plus courantes :

Initialisation et configuration

git init : Initialise un nouveau dépôt Git.

git config : Configure Git (ex : nom d'utilisateur, email).

Statut et historique

git status : Affiche l'état des fichiers (modifiés, en attente...).

git log : Affiche l'historique des commits.

Ajout et validation

git add <fichier> : Ajoute un fichier à l'index (staging).

git add . : Ajoute tous les fichiers modifiés.

git commit -m "message" : Enregistre les modifications avec un message.

Synchronisation

git remote add origin <url> : Ajoute un dépôt distant.

git push origin main : Envoie les commits vers la branche main du dépôt distant.

git pull : Récupère les dernières modifications du dépôt distant.

Branches

git branch : Liste les branches locales.

git branch <nom> : Crée une nouvelle branche.

git checkout <branche> : Bascule vers une branche.

git checkout -b <branche> : Crée et bascule vers une nouvelle branche.

git merge <branche> : Fusionne une branche dans l'actuelle.

Divers

git clone <url> : Clone un dépôt distant en local.

git diff : Affiche les différences entre les fichiers.

git reset : Annule les changements (selon les options).

Ajouter un projet déjà créé

On vous demandera pour la première fois, de s'authentifier à GitHub

1. Naviguer et se mettre à l'intérieur du répertoire du projet dans l'invite de commande (cmd) puis taper :

```
git init
```

pour que git initialise ce dossier en projet git. Cela ajoute les trois arbres nécessaires au suivi de version

- le répertoire de travail (la version actuelle): working directory
- l'index : la liste des choses qui sont préparées (la commande pour ajouter au « stage » est « add»)
- le head qui correspond à ce qui a été « commit », c'est à dire vraiment envoyé comme version

2. Puis taper :

```
git add .
```

C'est la commande qui « stage » tous les documents, qui les ajoute à l'index. Pour info, on peut sortir de la pile un document avec la commande git reset HEAD votre Fichier

3. Taper :

```
git commit -m "mon premier commit"
```

qui permet de transférer les fichiers en attente dans l'index, vers le Head. Le message permet de savoir à quoi cela correspond.

On utilise l'option -m pour spécifier le message du commit, avec Git, on ne peut pas faire un commit sans spécifier de message. Il faut donc dans ce message expliquer pourquoi on a fait des changements. Ici on utilise juste le message "mon premier commit"

4. Dans l'invite de commande taper :

```
git remote add origin leLienCopiéDuRepository
```

Exemple

```
git remote add origin https://github.com/espace2021/myRepoProject.git
```

va désigner le répertoire distant comme cible du projet.

5. Taper par la suite :

```
git remote -v
```

pour la vérification que les choses fonctionnent dans les deux sens.

6. Enfin taper

```
git push --set-upstream origin main
```

```

Enumerating objects: 47, done.
Counting objects: 100% (47/47), done.
Delta compression using up to 8 threads
Compressing objects: 100% (44/44), done.
Writing objects: 100% (47/47), 549.21 KiB | 12.20 MiB/s, done.
Total 47 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/espace2021/myRepoProject.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

```

Pour chaque mise à jour du contenu, pour envoyer les modifications locales apportées au dépôt du serveur :

```

git add .
git commit -m "le nom du commit"
git push

```

Le système de branches

Le principal atout de Git est son système de branches.

Les différentes branches correspondent à des copies de votre code principal à un instant T, où vous pourrez tester toutes vos idées les plus folles sans que cela impacte votre code principal.

Une branche est une “copie” d’un projet sur laquelle on opère des modifications de code.

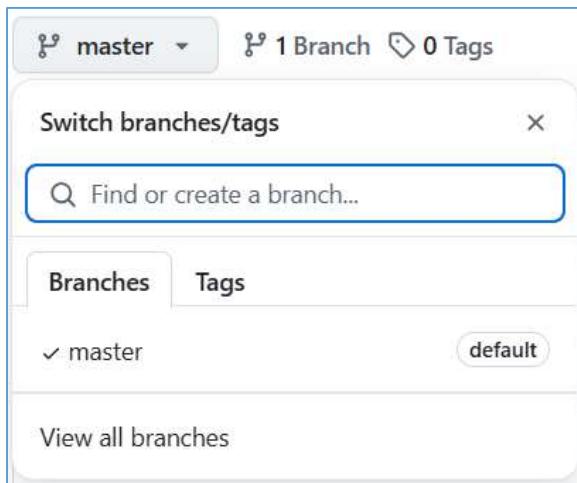
Sous Git, la branche principale est appelée la branche main, ou master.

La branche principale (main ou master) portera l’intégralité des modifications effectuées. Le but n’est donc pas de réaliser les modifications directement sur cette branche, mais de les réaliser sur d’autres branches, et après divers tests, de les intégrer sur la branche principale.

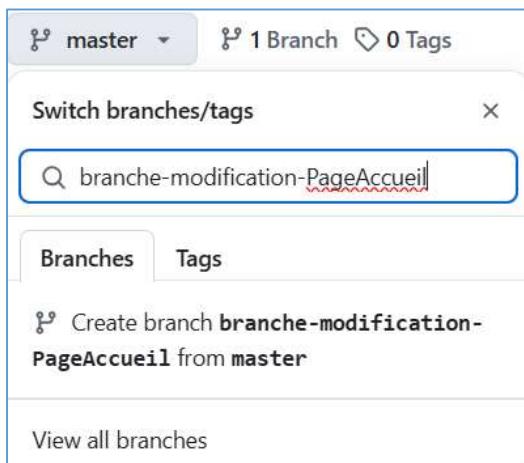
Méthode 1 : Créer une branche depuis l'interface GitHub (web)

Aller sur le dépôt GitHub.

En haut de la liste des fichiers, clique sur le menu déroulant de la branche (souvent nommé main ou master).



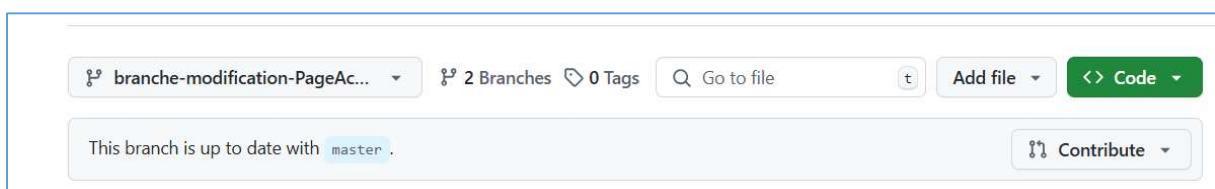
Dans la zone de texte, tape le nom de ta nouvelle branche.



Cliquer sur

Create branch branche-modification-PageAccueil from master

La branche est maintenant créée et tu y es automatiquement positionné.



Méthode 2 : Créer une branche en ligne de commande

Créer une branche

```
git checkout -b nom-ta-branche
```

Pousser la branche sur GitHub

```
git push origin nom-branche
```

On peut ensuite faire les modifications, les valider avec git commit, puis faire un push.