S 1.01

DEV

TP - GitHub

1 Principe général de GitHub

Lors de vos TP de R1.01, vous avez appris à utiliser l'outil GIT en créant un dépôt **local** (sur votre ordinateur), permettant de conserver un historique des modifications réalisées sur votre projet.

GitHub est une plateforme en ligne permettant de créer des dépôts GIT **distants**. En plus de conserver l'historique des modifications, cette plateforme permet notamment de :

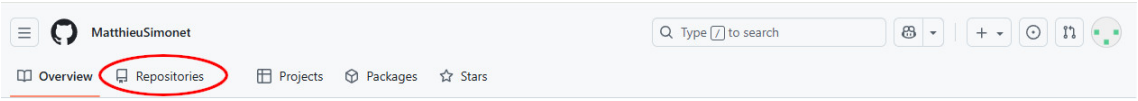
- **Sauvegarder votre code en ligne :** vous pouvez ainsi travailler depuis n'importe quel ordinateur sans transfert manuel de fichiers.
- **Rendre votre code accessible :** plusieurs personnes peuvent collaborer simultanément sur un même projet.

En résumé, GitHub permet de travailler à plusieurs sur un même code depuis n'importe quelle machine connectée à Internet, sans avoir à gérer manuellement le transfert ou la fusion des fichiers.

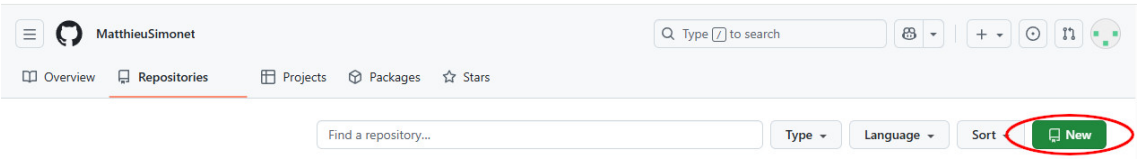
Dans ce TP, nous découvrirons les fonctionnalités de base de GitHub utiles pour votre SAE.

2 Création d'un dépôt en ligne

Nous allons commencer par créer un dépôt en ligne (repository). Rendez-vous sur le site **GitHub.com** et connectez-vous. Cliquez sur l'onglet *Repositories*, situé en haut à gauche, pour accéder à la liste de vos dépôts (probablement vide pour l'instant).



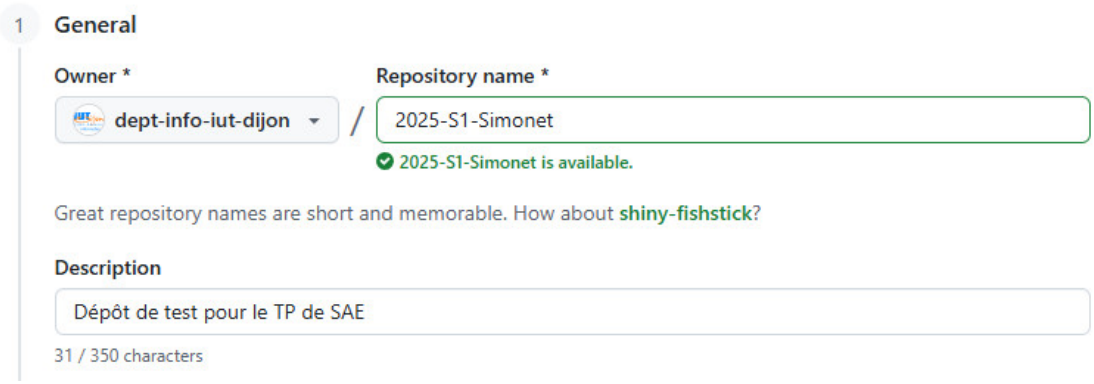
Cliquez ensuite sur le bouton vert *New*, en haut à droite, pour ouvrir l'écran de création d'un dépôt.



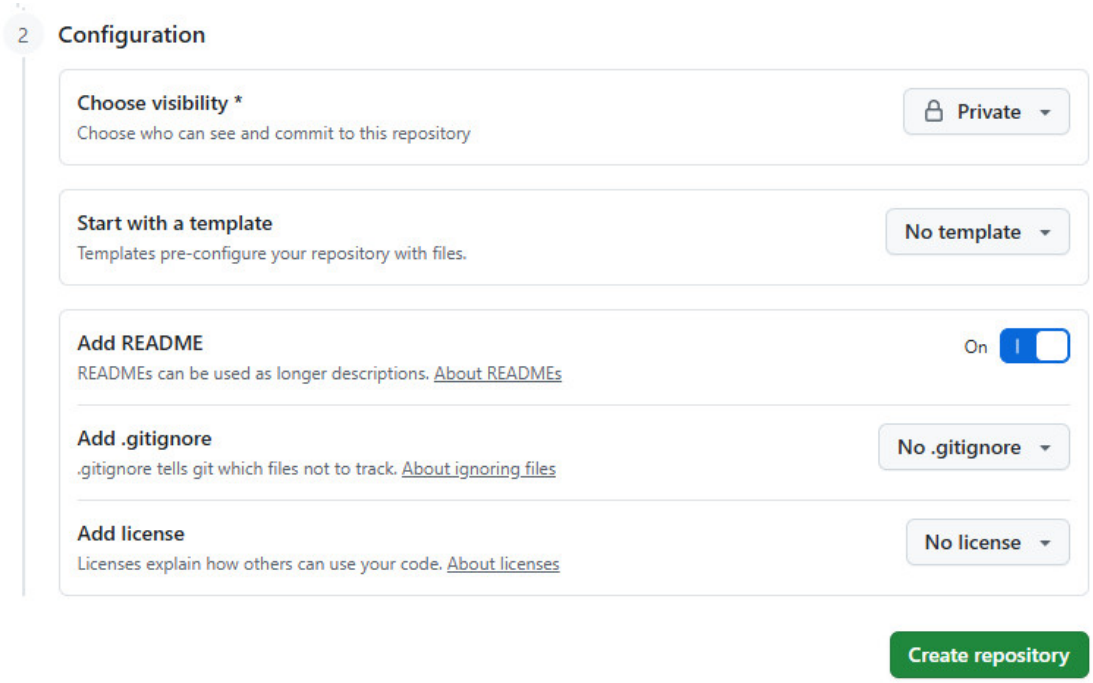
Dans cet écran, vous devez configurer votre dépôt. La première étape consiste à choisir un propriétaire (owner) et un nom pour votre dépôt.

Important : pour tous les projets réalisés dans le cadre de votre formation à l'IUT, vous devez IMPÉRATIVEMENT choisir *dept-info-iut-dijon* comme propriétaire ! Pour vos projets personnels, vous choisirez votre propre compte.

Choisissez donc *dept-info-iut-dijon* comme propriétaire et nommez votre dépôt *2025-S1-XXXX*, en remplaçant XXXX par votre nom. Vous pouvez également renseigner une courte description.

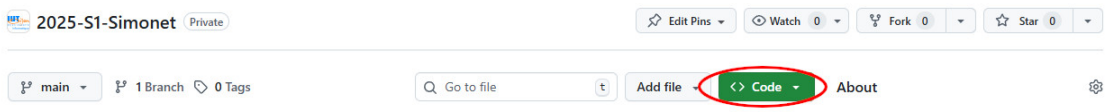


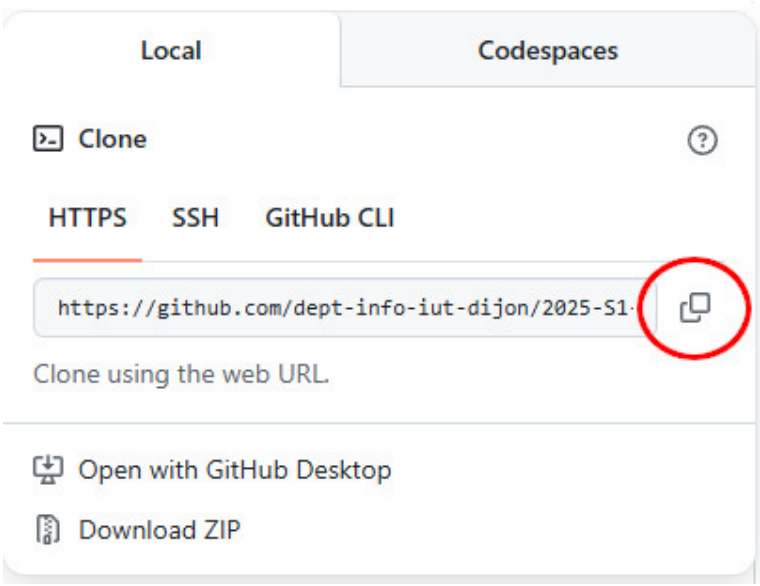
Dans la seconde partie de l'écran, configurez votre dépôt comme indiqué ci-dessous :



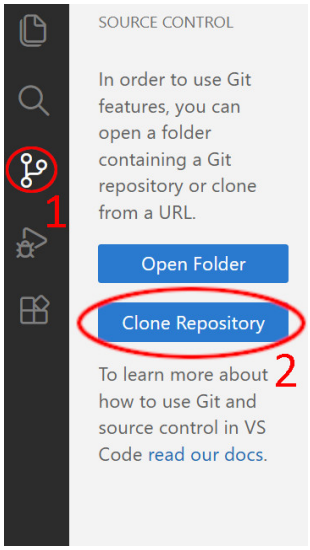
- **Visibility :** définit qui peut voir votre dépôt. En privé, seuls les membres invités et les enseignants y ont accès.
- **Template :** permet de créer un dépôt à partir d'un modèle (non utilisé ici).
- **README :** ajoute un fichier texte destiné à présenter le projet.
- **.gitignore :** liste les fichiers à exclure du dépôt. Nous n'utiliserons pas les modèles proposés : un fichier personnalisé sera ajouté ensuite.
- **License :** permet de choisir une licence open source. Inutile dans un cadre universitaire, mais utile pour des projets personnels publics.

Une fois le dépôt correctement paramétré, cliquez sur *Create repository*. Votre dépôt est alors créé et vous accédez à sa page principale. Nous y reviendrons plus tard. Pour l'instant, récupérez l'adresse de votre dépôt : cliquez sur le bouton vert *Code*, puis sur l'icône *Copier*.





3 Relier Visual Studio Code au dépôt en ligne



Gardez la page GitHub ouverte : elle servira plus tard. Ouvrez Visual Studio Code et assurez-vous qu'aucun dossier n'est actuellement ouvert (fermez-le si nécessaire).

Ouvrez l'extension *Source Control* en cliquant sur son icône, puis cliquez sur *Clone Repository*. Collez l'adresse de votre dépôt et validez. Choisissez un dossier pour stocker le projet, puis acceptez de l'ouvrir. Un dépôt Git **local** est alors automatiquement créé et lié à votre dépôt distant.

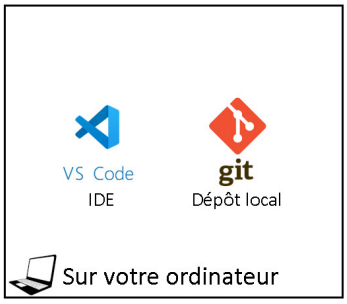
Une fois ce dossier ouvert dans VS Code (sur cette machine ou une autre), la liaison au dépôt GitHub sera directement reconnue : inutile de refaire la procédure.

4 Premier commit

4.1 Fonctionnement de GitHub

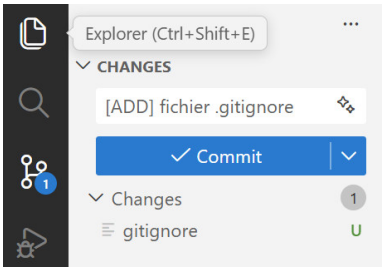
Pour bien utiliser GitHub, il faut comprendre que vous interagirez avec **deux dépôts distincts** :

- un dépôt local, présent sur votre machine (dans le dossier *.git*),
- un dépôt distant, hébergé sur GitHub.



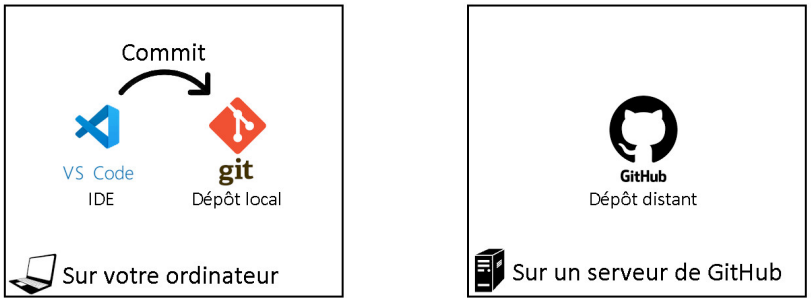
4.2 Ajout du .gitignore

Nous allons ajouter un premier fichier : le fichier `.gitignore`, qui évite d'envoyer en ligne des fichiers inutiles (configuration, temporaires...). Récupérez le fichier `.gitignore` fourni dans le dossier du TP sur le commun, puis copiez-le dans le dossier de votre projet.



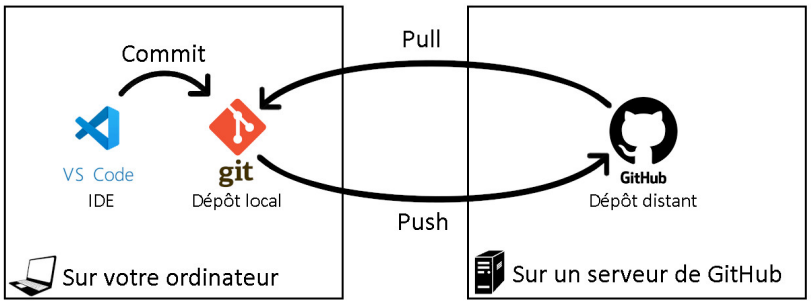
Dans le menu *Source Control* de VS Code, vous verrez apparaître un changement. Entrez alors le message de commit suivant : `[ADD] fichier .gitignore` puis cliquez sur *Commit*.

Dans votre navigateur, rechargez la page de votre dépôt. Le fichier `.gitignore` n'apparaît pas encore : c'est normal ! Le commit a été fait **localement**, mais le dépôt local n'a pas encore été synchronisé avec le dépôt distant.



Pour synchroniser les deux dépôts, cliquez sur *Sync Changes*. Cette action réalise automatiquement :

- **Pull :** les dernières modifications distantes sont récupérées,
- **Merge :** elles sont fusionnées avec votre travail local,
- **Push :** votre version locale mise à jour est envoyée sur GitHub.



Rechargez alors de nouveau la page de votre dépôt : le fichier `.gitignore` apparaît désormais. Vous devriez aussi voir que la page indique *2 commits*.

Note : il n'est pas nécessaire de synchroniser après chaque commit. Plusieurs commits peuvent être synchronisés d'un coup.

4.3 Conseils sur les commits

Quelques bonnes pratiques :

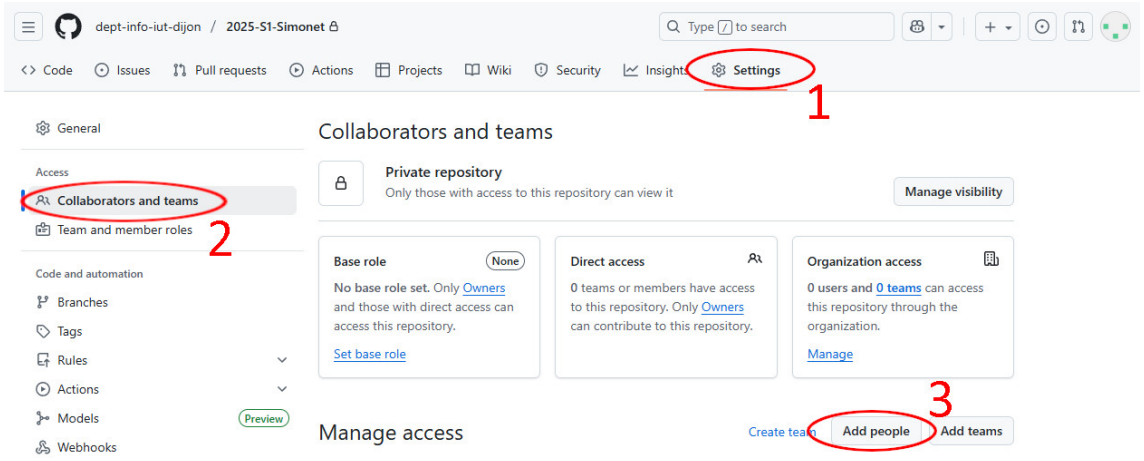
- Pour tout nouveau dépôt, commencez **toujours** par un commit contenant uniquement le `.gitignore`.
- Faites des commits fréquents (par exemple après chaque fonction ou procédure).
- Ne faites jamais de commit si votre code ne fonctionne pas.
- Rédigez des messages précis et normalisés, par exemple : *[Ajout] Unité personnage.pas, ajout de la fonction attaquer*.

5 Travailler à plusieurs

Nous allons maintenant voir comment collaborer sur un même dépôt. Travaillez au moins en binôme. Un étudiant sera le *propriétaire* du dépôt ; les autres seront *collaborateurs*.

5.1 Inviter un collaborateur

Le propriétaire doit inviter les collaborateurs. Pour cela, rendez-vous sur la page du dépôt, cliquez sur *Settings*, puis *Collaborators and teams*, et enfin sur *Add people*.



Entrez le login GitHub du collaborateur, sélectionnez le rôle *Admin*, puis validez. Le collaborateur reçoit une invitation (mail + GitHub). Une fois acceptée, il a accès au dépôt.

5.2 Cloner le dépôt

Le collaborateur peut alors cloner le dépôt sur son ordinateur, en suivant la procédure décrite en section 3, à partir de l'adresse du dépôt du propriétaire.

5.3 Première collaboration

Testons le fonctionnement :

- Un premier étudiant crée un projet FreePascal dans le dossier lié au dépôt distant et modifie le programme principal pour afficher *Bonjour !*. Il fait un commit puis une synchronisation.
- Le deuxième étudiant récupère ensuite la modification via *Source Control* → … → *Pull*.

Note : un `pull` est automatiquement effectué lors d'une synchronisation.

5.4 Conflits

Il arrive que Git ne puisse pas fusionner automatiquement les versions du projet : c'est un **conflit**. Heureusement, VS Code propose un outil simple pour les résoudre.

Pour provoquer volontairement un conflit :

- Le premier étudiant modifie le message en *Bonjour Alice !*, puis fait un commit (sans synchroniser).
- Le deuxième modifie le message en *Bonjour Bob !*, puis fait un commit (sans synchroniser).
- Le premier synchronise.
- Le deuxième synchronise.

Un conflit apparaît chez le deuxième étudiant. Un bouton *Resolve in Merge Editor* apparaît : cliquez dessus. VS Code affiche alors les deux versions à fusionner.

Les boutons disponibles :

- **Accept Current :** conserve la version locale (Bonjour Bob !)
- **Accept Incoming :** conserve la version distante (Bonjour Alice !)
- **Accept Combination :** garde les deux versions à la suite

Choisissez l'option pertinente, puis cliquez sur *Complete Merge*. Le commit proposé est automatiquement rempli. Validez-le puis synchronisez.

6 Et pour finir

6.1 Suppression du dépôt

Maintenant que vous avez pris en main GitHub, supprimez le dépôt créé pour ce TP. Pour cela, rendez-vous sur la page du dépôt, onglet *Settings*. Tout en bas se trouve la *Danger Zone*. Cliquez sur *Delete this repository* et finalisez la suppression.

6.2 Création du dépôt de la SAE

Créez maintenant le dépôt pour votre SAE. En groupe :

- le dépôt doit s'appeler *2025-S1-SAE1.01-LeNomDeVotreEquipe*,
- il doit être privé,
- le propriétaire doit être *dept-info-iut-dijon*.

Puis invitez les autres membres du groupe.