



Cégep de Saint-Hyacinthe
Département d'informatique

Programmation client-serveur

420-2RP-HY

Notes de cours

Gestion de versions de programmes – Pratique 1



Enseignante
Giovana Velarde

Hiver 2022

Table des matières

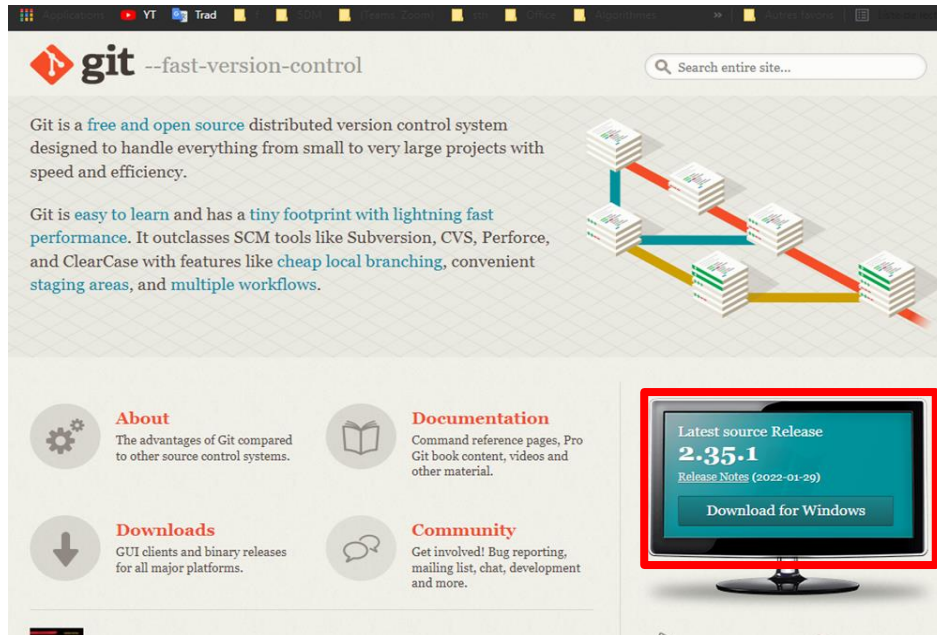
1	LES OUTILS GIT ET GITHUB	3
1.1	INSTALLATION DE GIT.....	3
1.2	CRÉATION D'UNE COMPTE GITHUB.....	5
2	PRÉPARATION DE L'ENVIRONNEMENT.....	6
2.1	CONFIGURER L'ENVIRONNEMENT GIT	6
2.2	CRÉATION D'UN DÉPÔT DISTANT DANS GITHUB	7
2.3	UTILISATION DE GIT DANS VISUAL STUDIO CODE	8
3	CLONER UN DÉPÔT DISTANT DANS VISUAL STUDIO CODE.....	9
4	GESTION DE MODIFICATIONS.....	11
5	TRAVAIL COLLABORATIF.....	17

Gestion de versions – Partie pratique 1

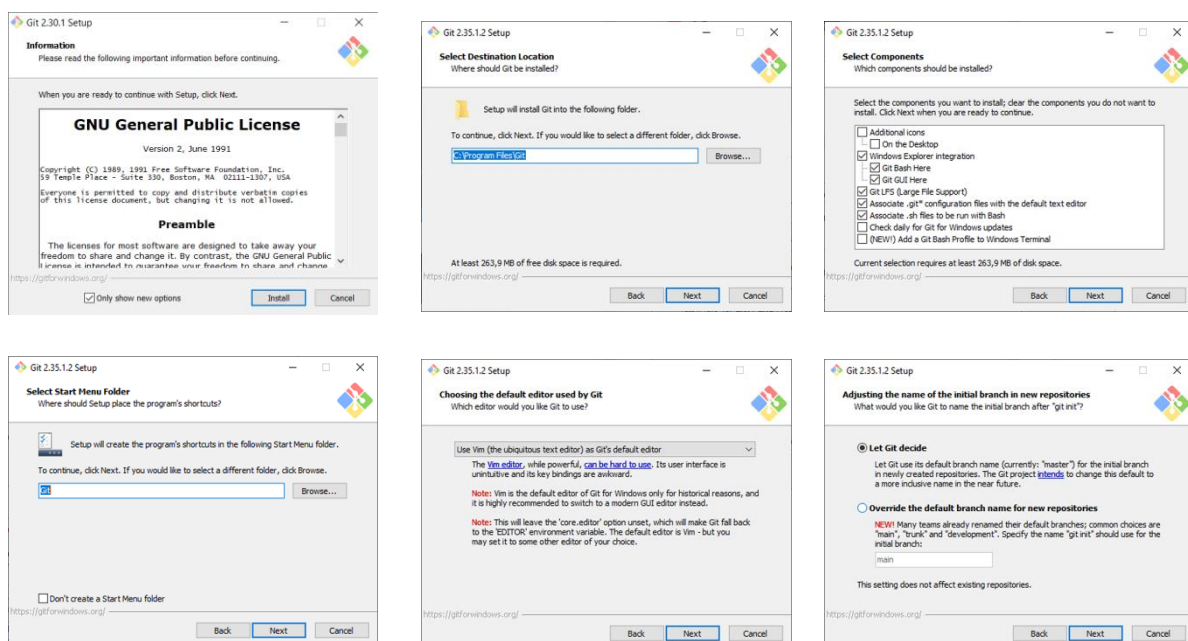
1 Les outils Git et GitHub

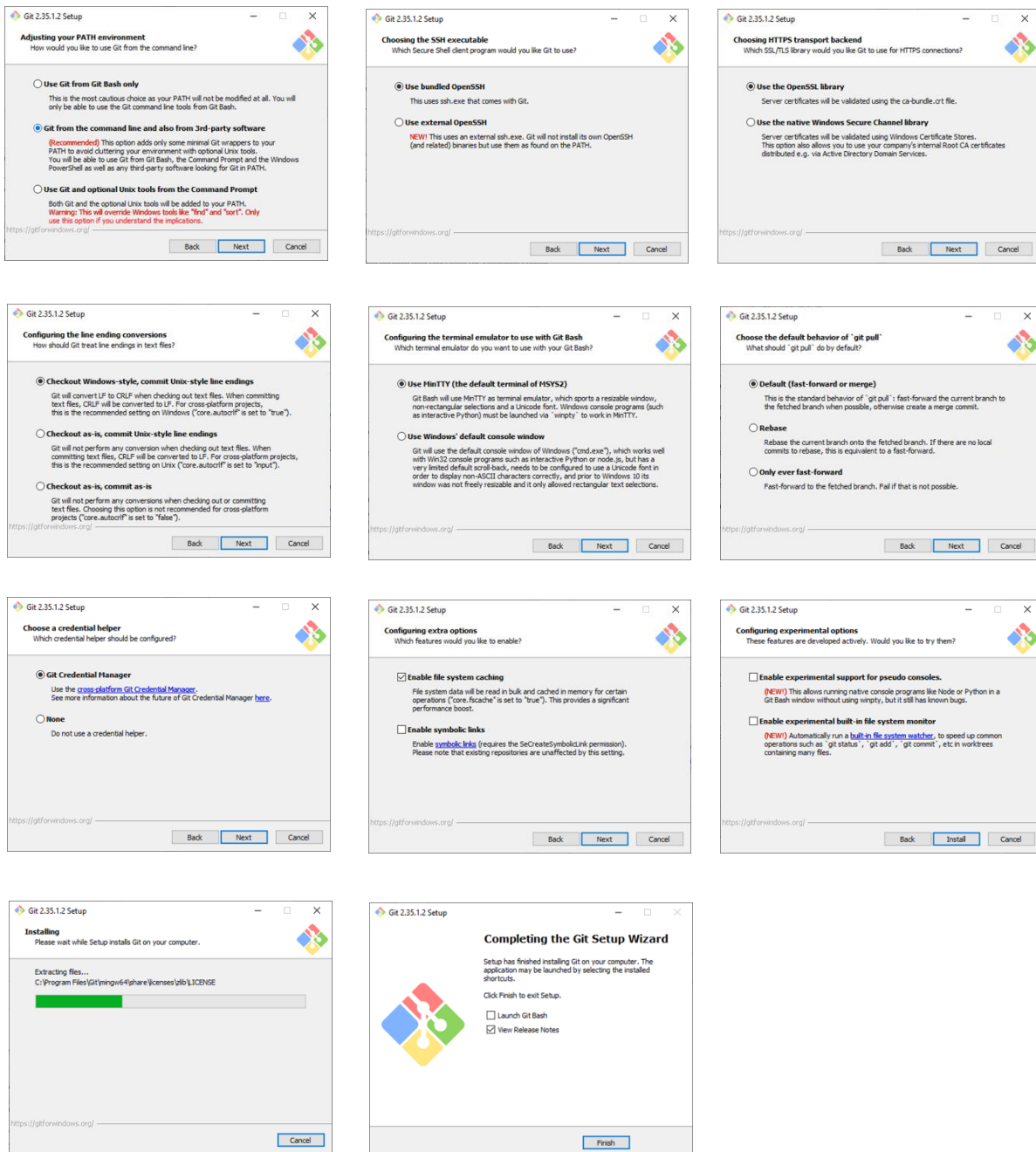
1.1 Installation de git

→ Aller sur le site de Git <https://git-scm.com/> et télécharger la dernière version disponible.



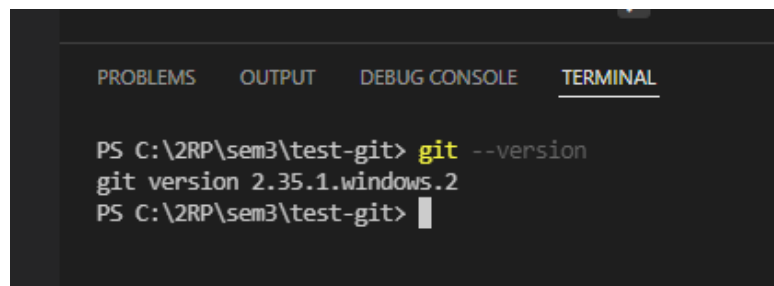
→ Lancer l'installation Git et suivre les instructions (laisser les configurations par défaut)





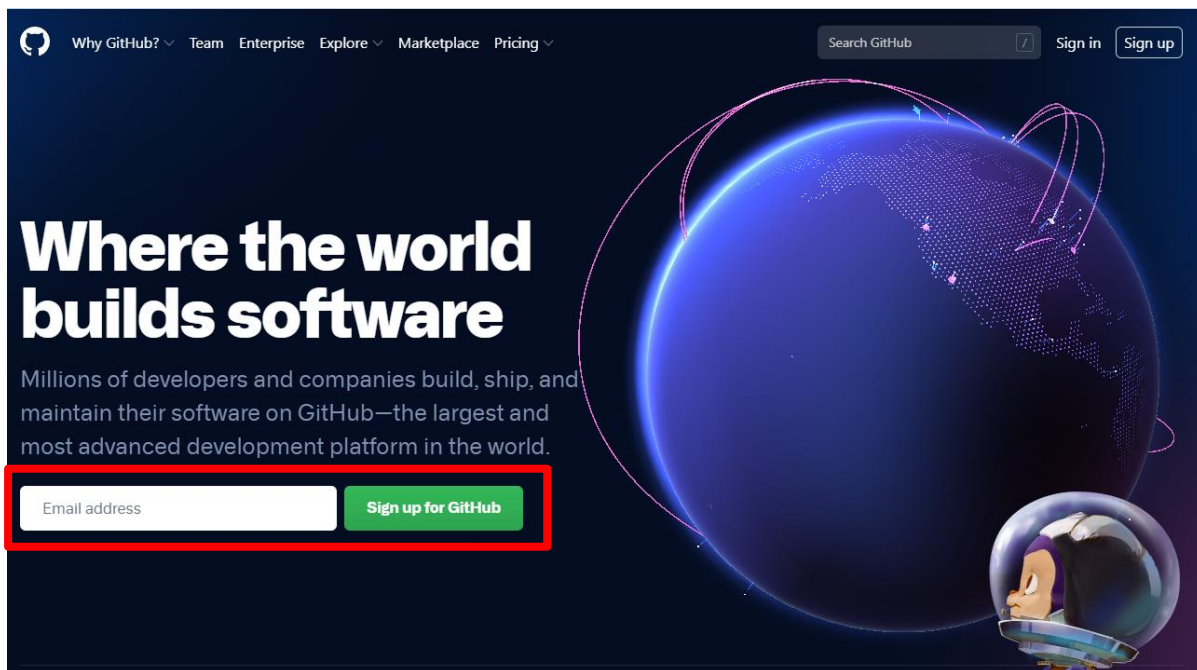
→ Redémarrer Visual Studio Code

→ Vérifier l'installation git :



1.2 Création d'une compte GitHub

→ Aller sur le site de GitHub <https://github.com/> et se créer un compte.



Join GitHub

Create your account

Username *

Email address *

Password *

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Email preferences


☐ Send me occasional product updates, announcements, and offers.

Verify your account

Veuillez résoudre cette énigme pour nous prouver que vous êtes une personne

Vérifier

Create account



Sign in to GitHub

Username or email address

Password [Forgot password?](#)

Sign in

2 Préparation de l'environnement

2.1 Configurer l'environnement Git

La configuration **Git** se fait à la ligne de commande une seule fois par ordinateur.

La commande `git config` permet d'établir les variables de configuration. Ce nom permet à Git de savoir à qui créditer les modifications faites aux projets futurs.

Pour configurer votre nom taper :

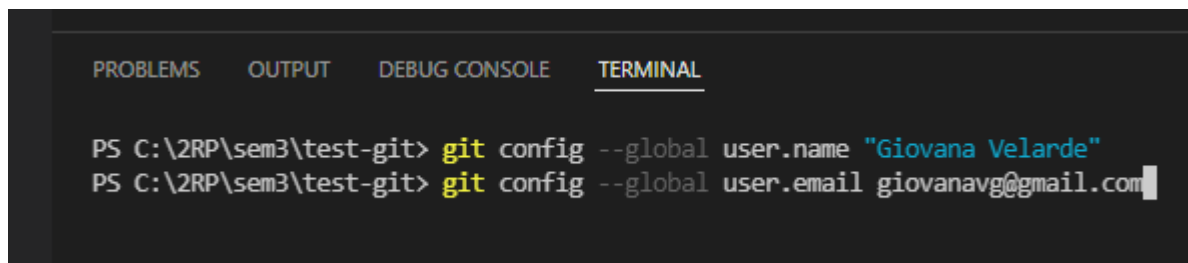
```
$ git config --global user.name "ton nom"
```

Pour configurer votre adresse courriel, taper :

```
$ git config --global user.email "ton adresse courriel de GitHub"
```

S'assurer de taper la même adresse email utilisée pour le compte **GitHub**.


Exemple :



A screenshot of a terminal window with a dark background. At the top, there are tabs labeled 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL', with 'TERMINAL' being the active tab. The terminal shows two commands being executed in a PowerShell prompt (PS C:\2RP\sem3\test-git>). The first command is `git config --global user.name "Giovana Velarde"` and the second is `git config --global user.email giovanavg@gmail.com`. The output of the first command is "Giovana Velarde" and the output of the second is "giovanavg@gmail.com".

Pour vérifier la configuration :

```
$ git config user.name
$ git config user.email
```

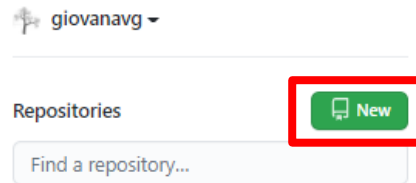


A screenshot of a terminal window with a dark background, similar to the previous one. It shows the output of the verification commands. The first command `git config user.name` returns "Giovana Velarde". The second command `git config user.email` returns "giovanavg@gmail.com".

Les configurations globales sont enregistrés dans le fichier **.gitconfig** du dossier **C:\Users\<username>**. Il est possible de l'ouvrir et le modifier

2.2 Création d'un dépôt distant dans GitHub

→ Dans la section « Repositories », cliquer → New

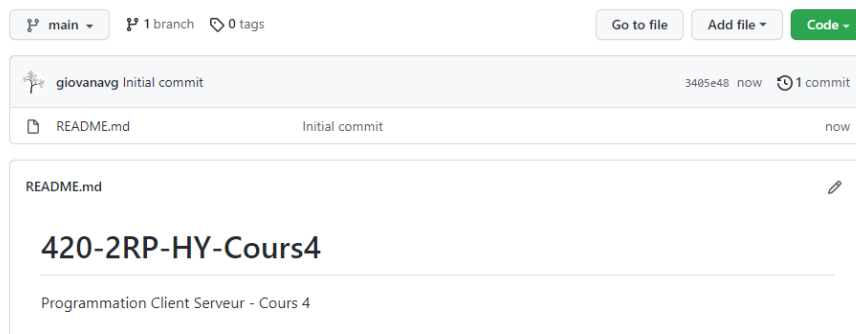


→ Dans « Repository name » tapez le nom du dépôt. Exemple: 420-2RP-HY-Cours4

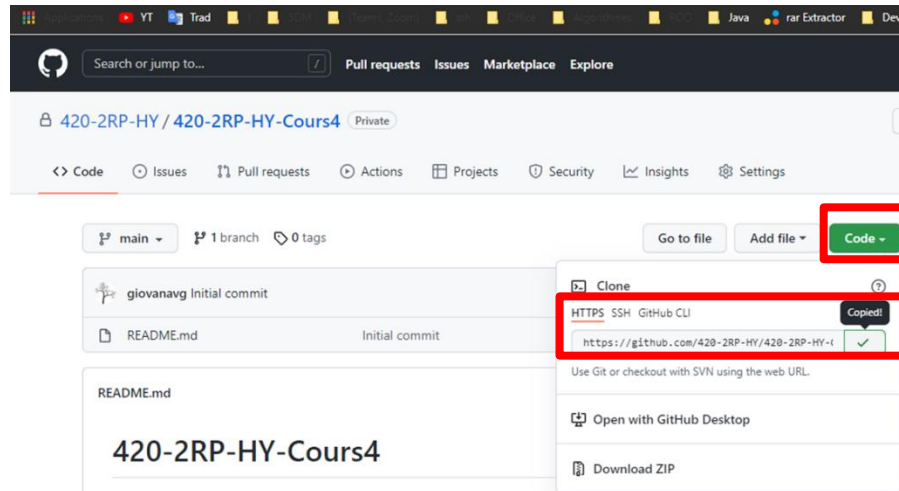
→ Cliquer → Create repository

A screenshot of the GitHub 'Create repository' form. The form includes fields for 'Owner' (set to '420-2RP-HY') and 'Repository name' (set to '420-2RP-HY-Cours4' with a green checkmark). Below these is a 'Description' field with the text 'Programmation Client Serveur - Cours 4'. The 'Public' radio button is selected, but the 'Private' option is highlighted with a red box. Under 'Initialize this repository with:', the 'Add a README file' checkbox is checked and highlighted with a red box. Other options like 'Add .gitignore' and 'Choose a license' are unchecked. At the bottom is a green 'Create repository' button.

Le dépôt créé :

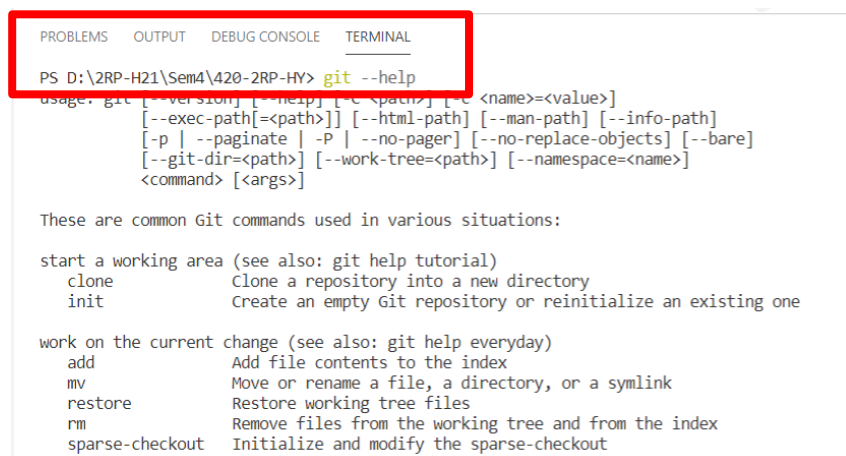
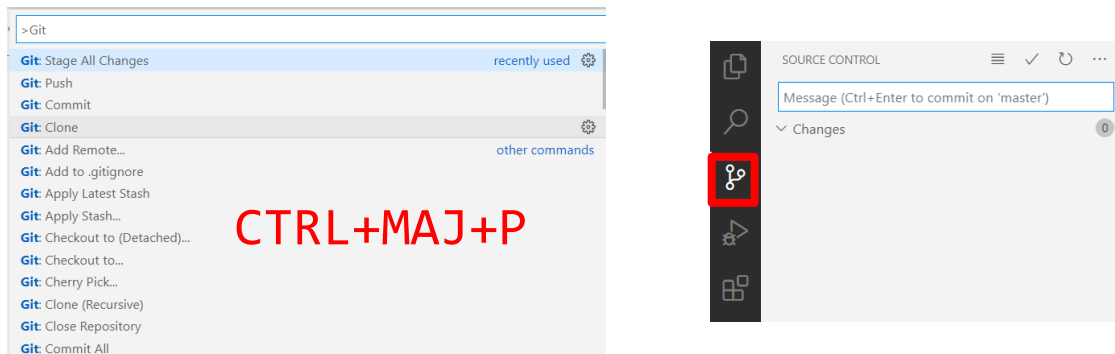


→ Copier l'URL du dépôt distant créé :



2.3 Utilisation de Git dans Visual Studio Code

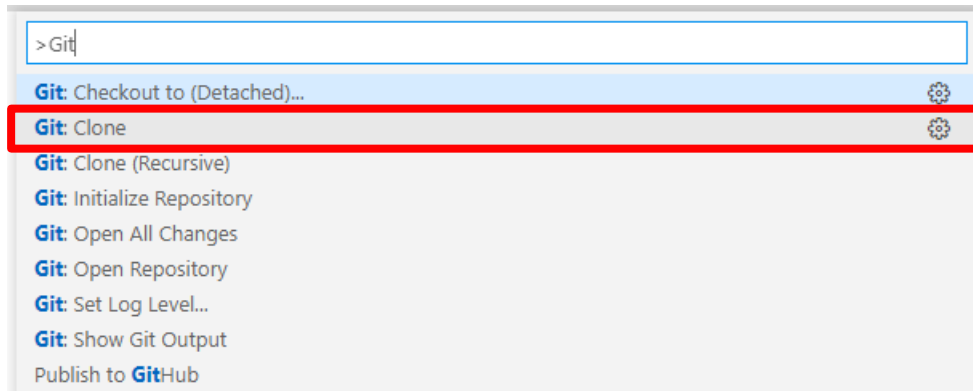
Visual Studio Code intègre des outils de contrôle de version Git. On peut les accéder depuis le panneau de contrôle de versions, le menu de commandes ou le terminal.



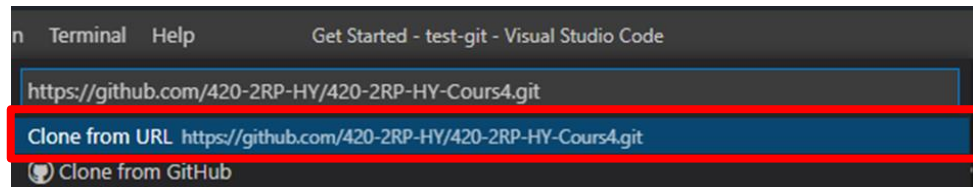
3 Cloner un dépôt distant dans Visual Studio Code

Pour importer un projet GitHub (dépôt distant) et travailler localement (dépôt local) on doit le **cloner**. Cela signifie qu'on récupère tout l'historique et tous les codes source d'un projet. GitHub est un serveur de dépôts Git et en contient une multitude. Pour cloner un dépôt distant, nous avons besoin son adresse URL GitHub.

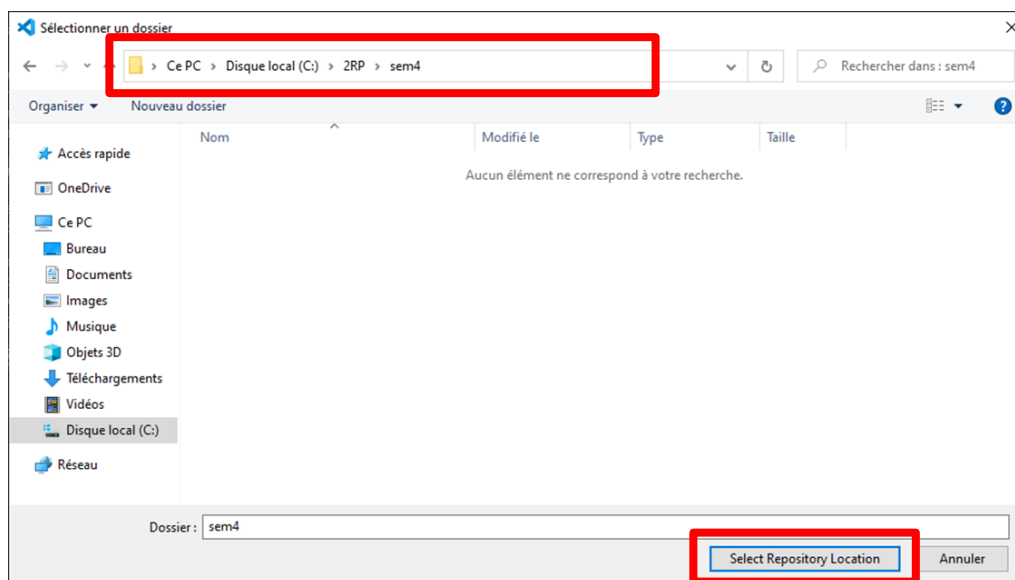
- Dans la palette de commandes (CTRL + MAJ + P)
- Sélectionner « **Git** : Clone »



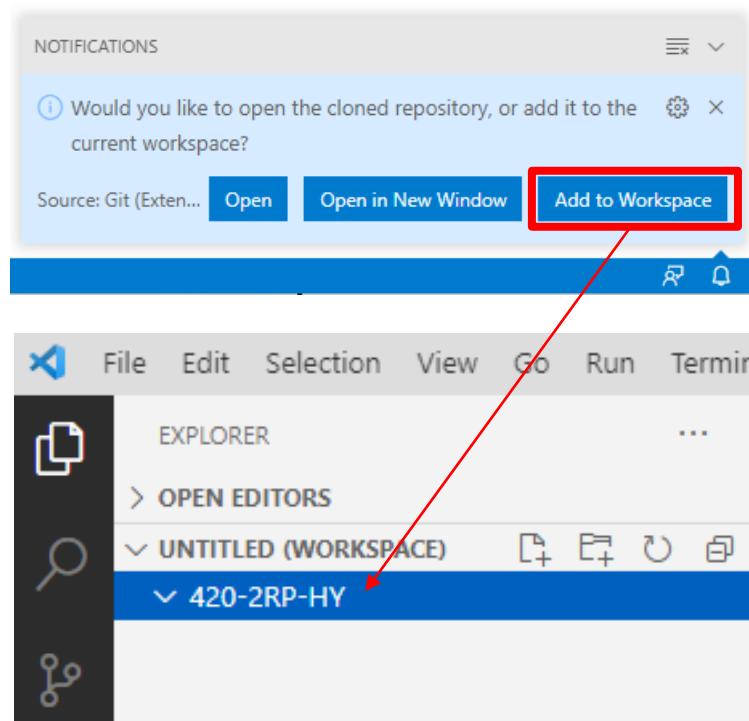
- Coller l'URL du dépôt distant GitHub que on veut cloner
- Cliquer « Clone from URL https://github.com/... »



- Sélectionner le dossier de travail du projet local (dépôt local) où le projet GitHub (dépôt distant) sera cloné.

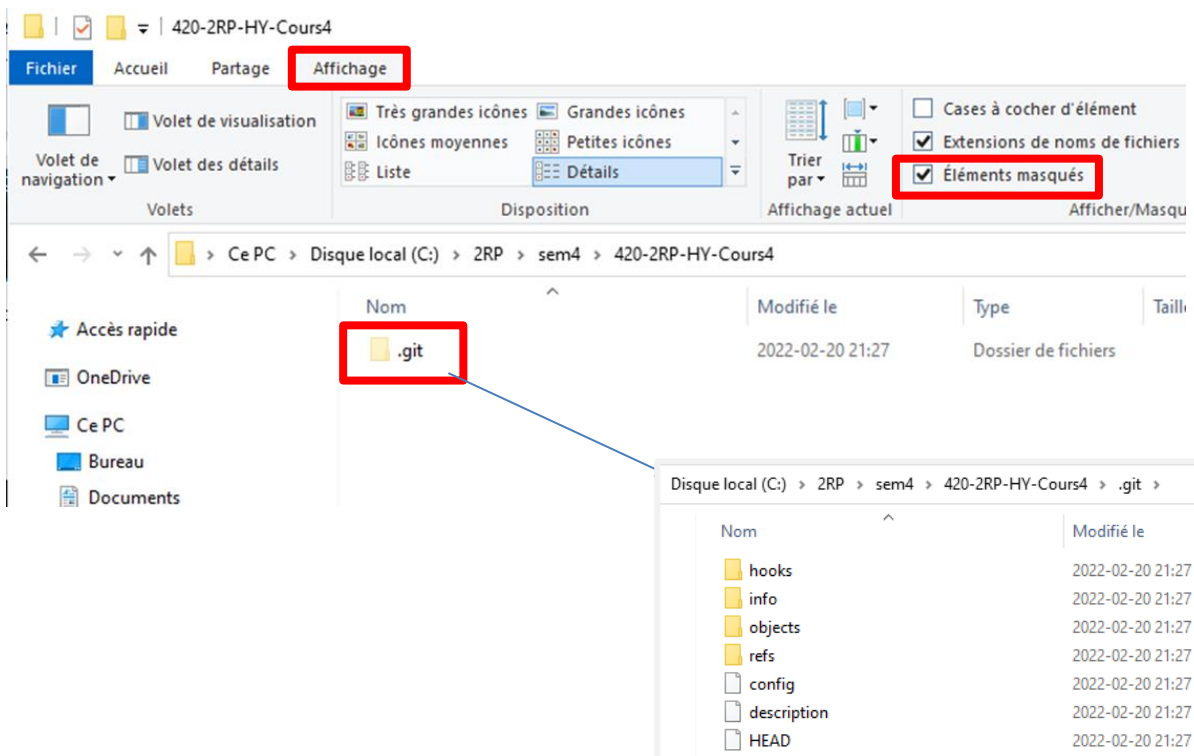


→ Ajouter le dossier du projet local dans VSCode.



Note : Si le clonage ne finit pas (n'arrête pas), lui arrêter (Cancel) et vérifier que le référentiel local ait été créé.

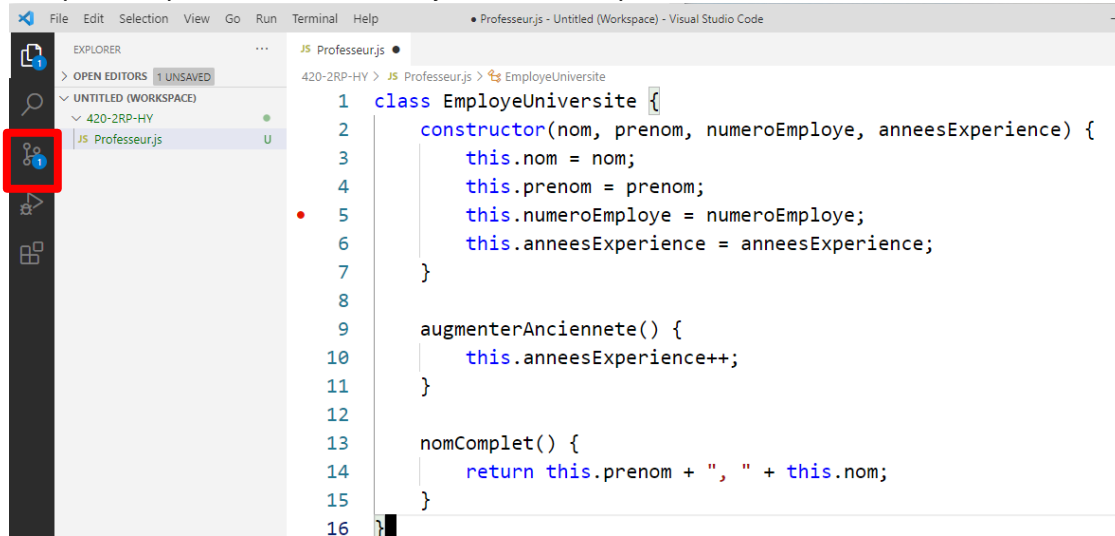
→ Un dossier caché « **.git** » est créé dans le dossier du projet. Ce dossier contient plusieurs fichiers utilisés par Git pour gérer les versions du code. L'**historique** des différentes modifications faites aux fichiers du dépôt y seront enregistrées.



4 Gestion de modifications

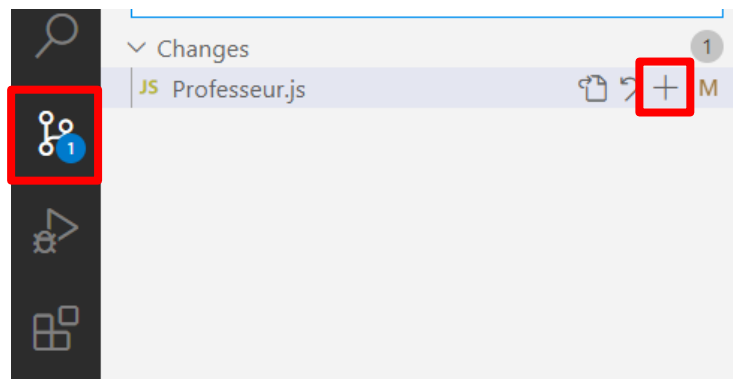
PREMIÈRE COMMIT

→ Dans Visual Studio Code créer un nouveau fichier dans votre projet. Modifier le fichier et l'enregistrer. Le panneau de Gestion de versions notifie les modifications faites dans le dépôt local qui n'ont pas encore été mis à jour dans le dépôt central GitHub.

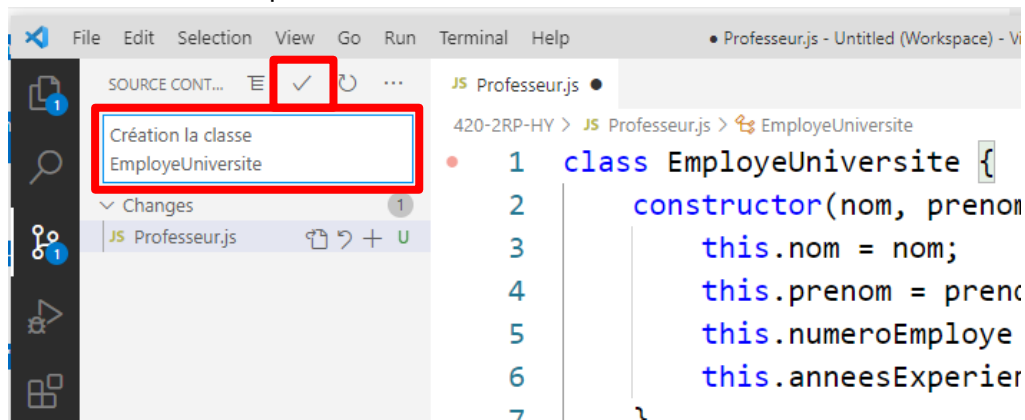


Pour envoyer une modification dans le dépôt central (GitHub) :

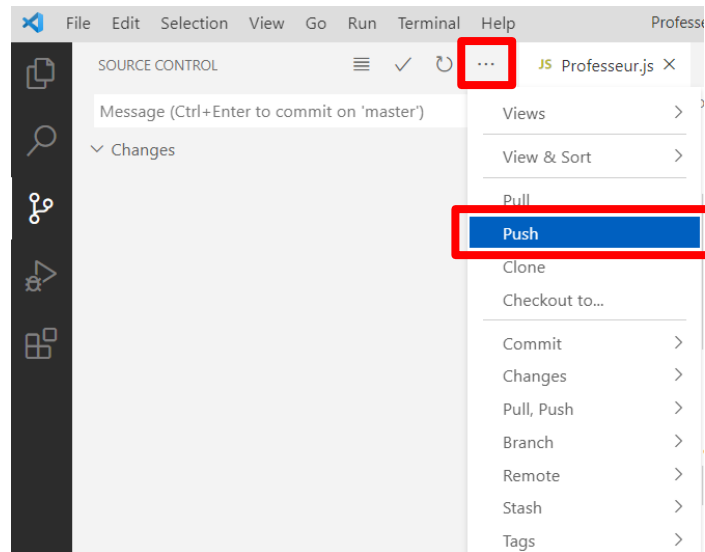
- Ouvrir le panneau de contrôle de la source (Source Control),
- Faire un **stage** (indexation) : Cliquer dans le bouton + « Stage all changes »



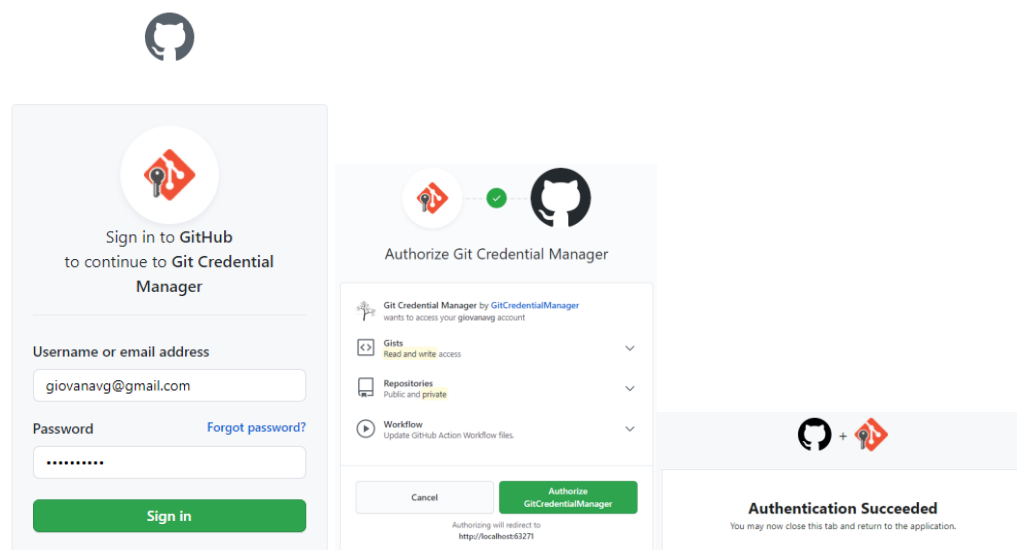
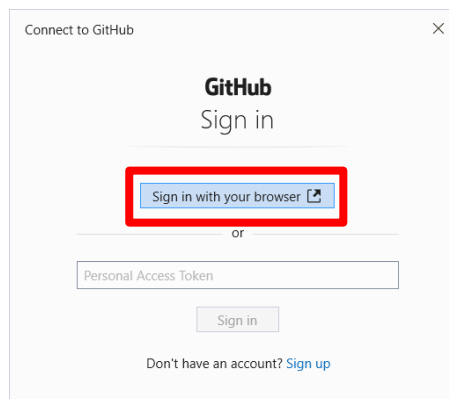
→ Pour faire un **commit**, entrer un **commentaire** correspondant aux modifications apportées dans la boîte en haut et cliquer sur le bouton ✓ « Commit ».



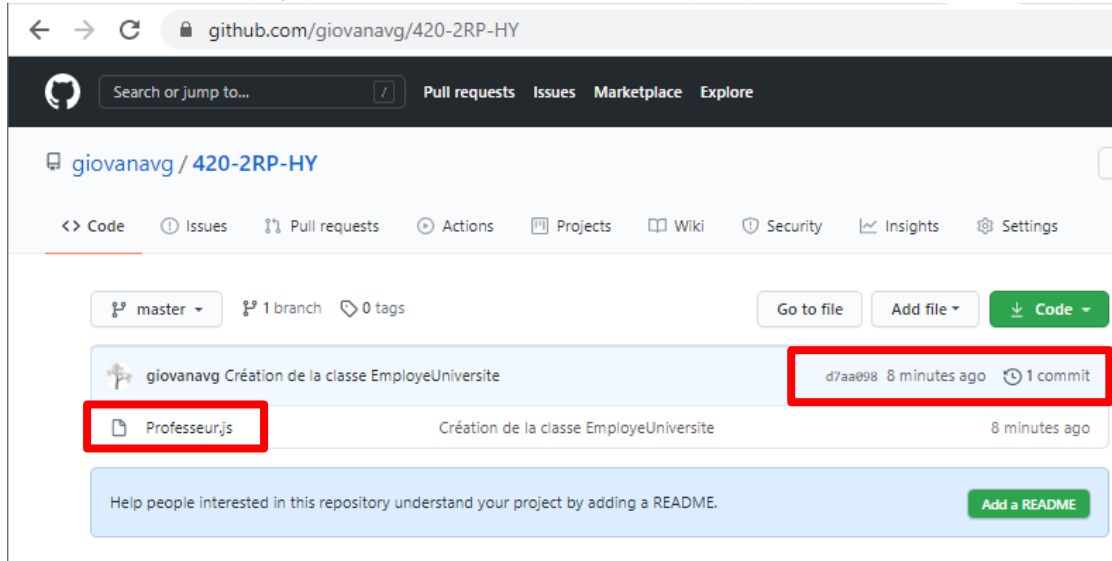
→ Pour pousser les modifications vers le dépôt distant, cliquer le bouton ... (trois points) en haut et à droite, et cliquer « **Push** »



La première fois, une connexion au compte GitHub sera demandée.

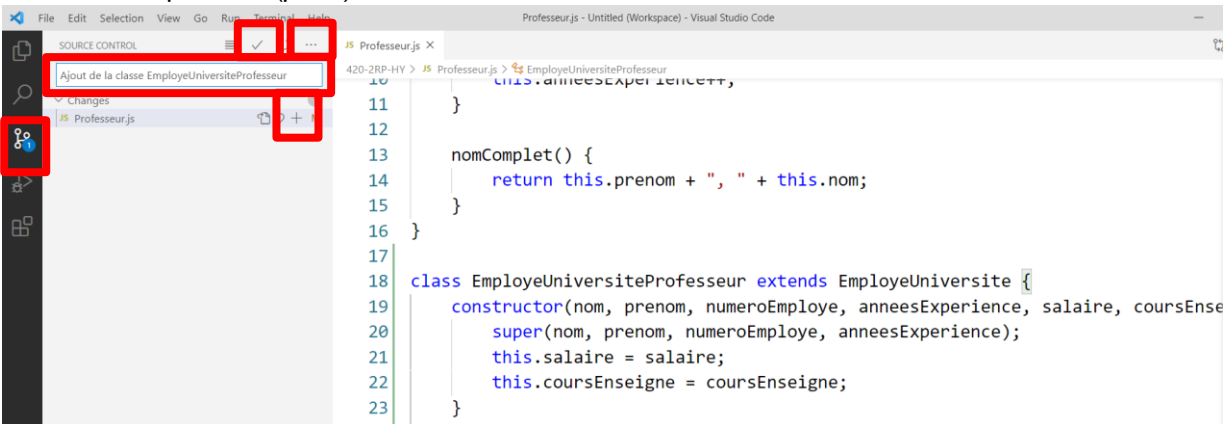


→ Vérifier que les changements faits au dépôt local ont été mis à jour dans le dépôt distant



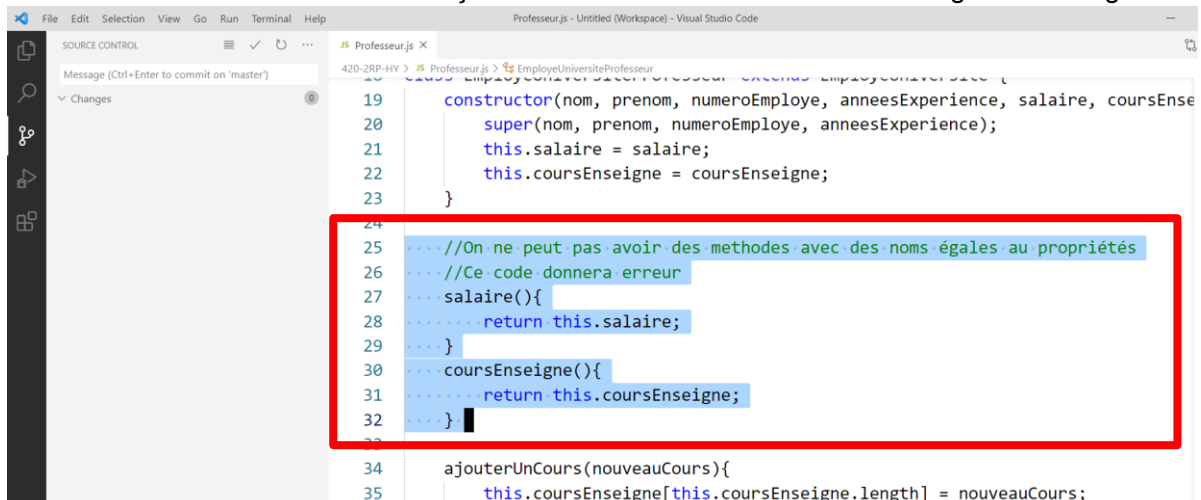
DEUXIÈME COMMIT

- Ajouter la classe « employeUniversiteProfesseur » et enregistrer le fichier.
- Faire un *stage*.
- Ajouter un commentaire et faire un *commit*.
- Finalement pousser (*push*).

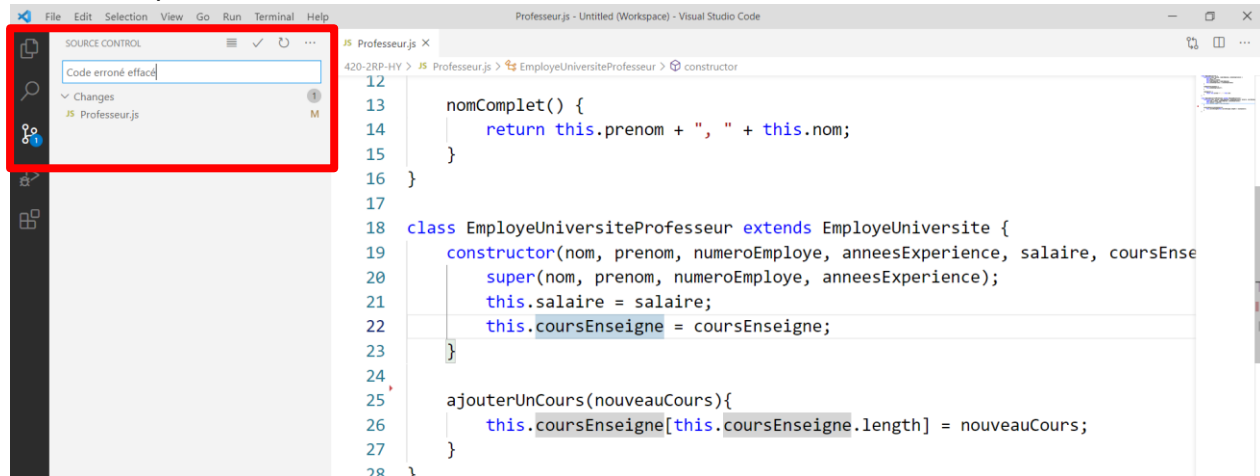


TROISIÈME COMMIT

→ Modifier à nouveau le fichier en effaçant les méthodes salaire et coursEnseigne et l'enregistrer.

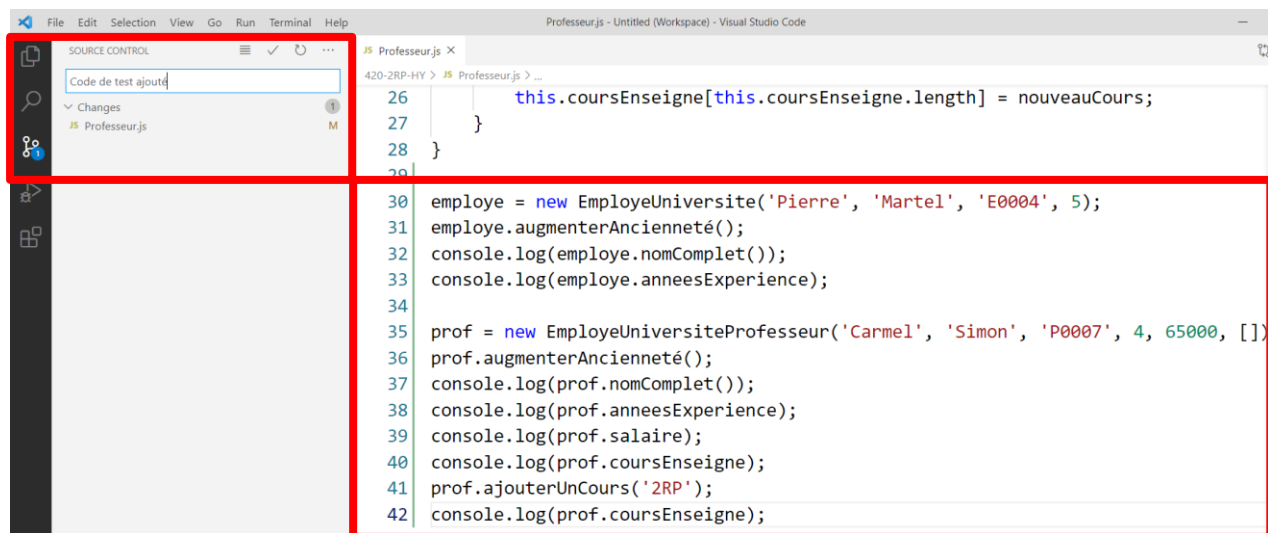


- Faire un *stage*
- Ajouter un commentaire et faire un *commit*.
- Faire un *push*.



QUATRIÈME COMMIT

- Ajouter le code de test.
- Faire toutes les étapes nécessaires pour pousser le commit dans le Serveur.

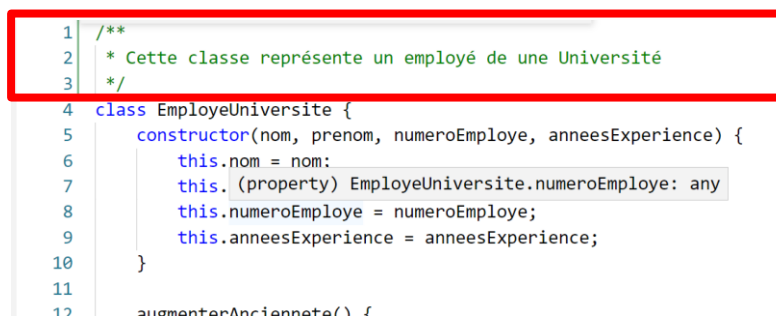


CINQUIÈME COMMIT

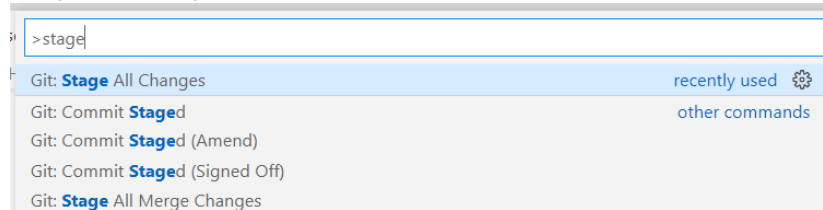
Une autre façon de gérer les modifications est en utilisant la Palette de commandes de Visual Studio Code.

CTRL + MAJ + ENTRÉE

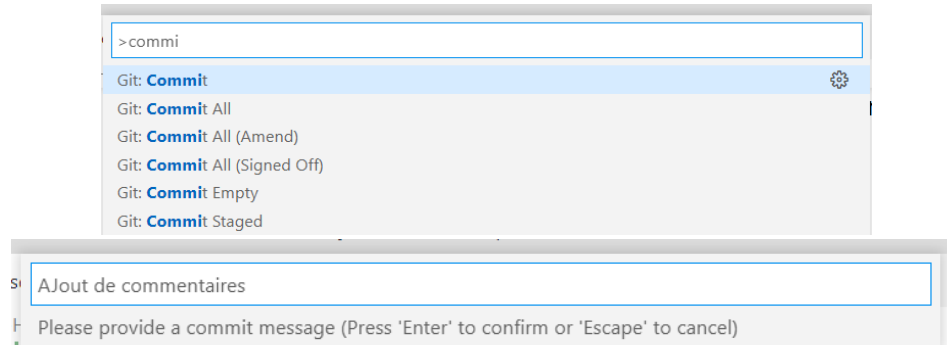
- Ajouter les commentaires de description de chaque classe.



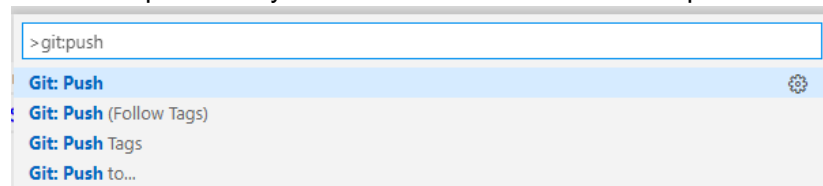
→ Cliquer « Git : Stage all changes »



→ Cliquer « Git :commit » et tapez le commentaire + ENTRÉE



→ Cliquer « Git :Push » pour envoyer vos modifications sur votre dépôt

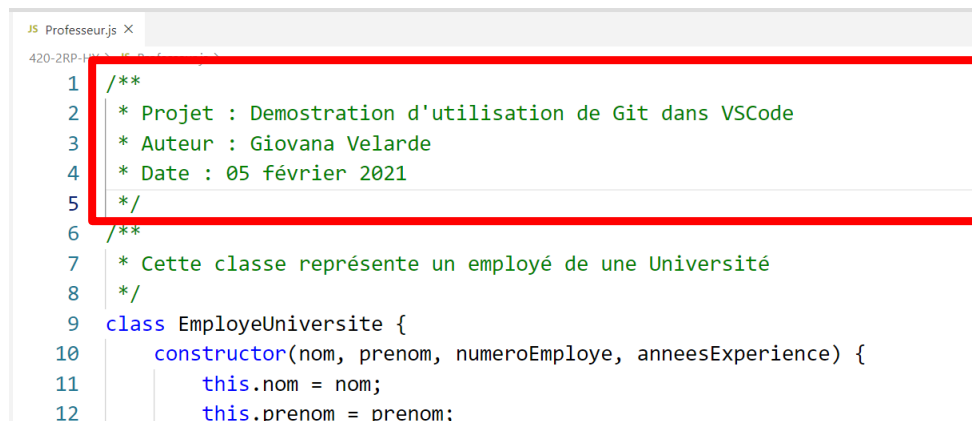


SIXIÈME COMMIT

On peut aussi utiliser les commandes Git dans le terminal de Visual Studio Code (ou dans le terminal Git Bash).



Ajouter la documentation du code.



Pour faire un *stage* dans le terminal, taper :

```
git add .
```

Pour faire un *commit* dans le terminal taper la commande commit avec le commentaire :

```
git commit -m "Ajout de documentation"
```

```
PS D:\2RP-H21\Sem4\420-2RP-HY> git commit -m "Ajout de documentation"
[master 058aec9] Ajout de documentation
1 file changed, 5 insertions(+)
PS D:\2RP-H21\Sem4\420-2RP-HY>
```

Pour faire un *push* vers le dépôt serveur, taper :

```
git push
```

```
PS D:\2RP-H21\Sem4\420-2RP-HY> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 397 bytes | 397.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/giovanavg/420-2RP-HY.git
9eee2da..058aec9 master -> master
PS D:\2RP-H21\Sem4\420-2RP-HY>
```

→ Vérifier dans le dépôt central GitHub que les *commits* faits dans le dépôt local ont été poussés

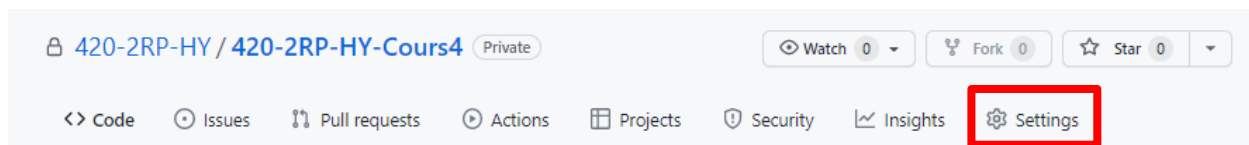
The screenshot shows the GitHub interface for the repository 'giovanavg / 420-2RP-HY'. At the top, there are tabs for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, and Insights. Below these, the repository name and a 'Unwatch' button are visible. The main section shows the commit history for the 'master' branch, which has 1 branch and 0 tags. A red box highlights the latest commit: 'giovanavg Ajout de documentation' (058aec9, 8 minutes ago) with 6 commits. Below this, a list of previous commits is shown, also highlighted by a red box:

- Ajout de documentation (giovanavg committed 6 minutes ago)
- Ajout de commentaires (giovanavg committed 26 minutes ago)
- Code de test ajouté (giovanavg committed 35 minutes ago)
- Code erroné effacé (giovanavg committed 37 minutes ago)
- Ajout de la classe EmployeUniversiteProfesseur (giovanavg committed 41 minutes ago)
- Création de la classe EmployeUniversite (giovanavg committed 1 hour ago)

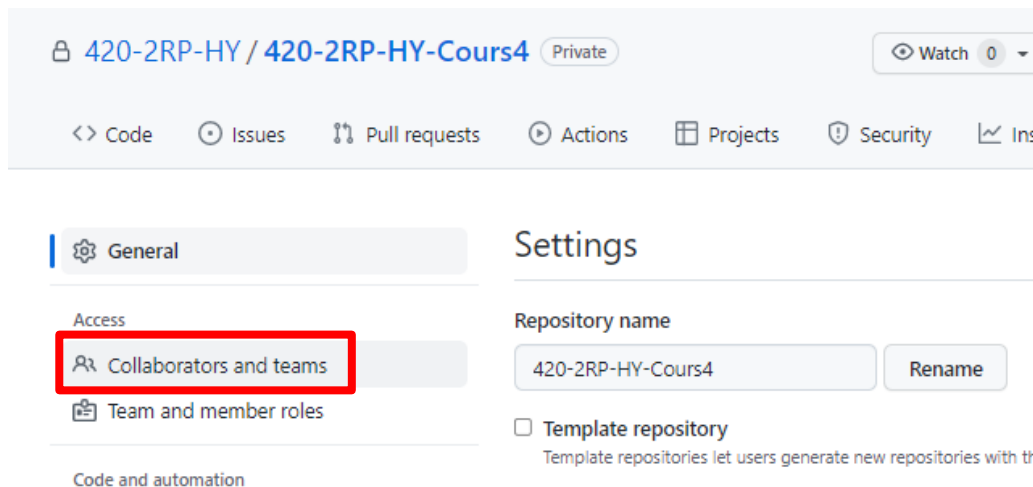
5 Travail collaboratif

Pour ajouter un collaborateur dans notre projet :

→ Dans le dépôt central GitHub, cliquer *Settings*



→ Cliquer *Collaborators and teams*



→ Dans *Manage access*, cliquer *Add people*

Who has access

PRIVATE REPOSITORY

Only those with access to this repository can view it.

[Manage](#)

BASE ROLE

None

No base role set. Only **Owners** and those with direct access can access this repository.

[Set base role](#)

DIRECT ACCESS

0 teams or members have access to this repository. Only **Owners** can contribute to this repository.

Manage access

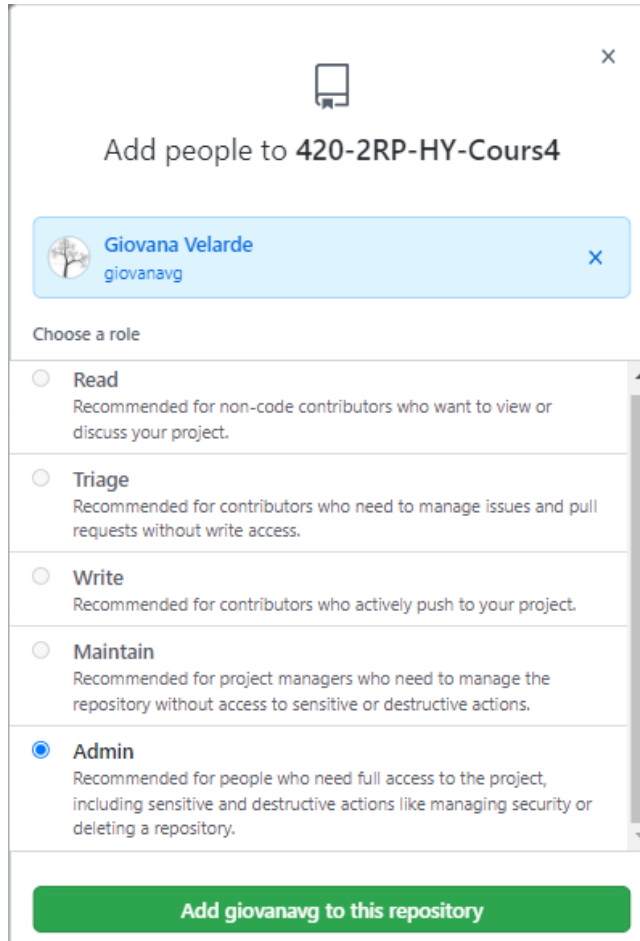
[Create team](#)

You haven't added any teams or people yet

Organization owners can manage individual and team access to the organization's repositories. Team maintainers can also manage a team's repository access. [Learn more about organization access](#)

[Add people](#) [Add teams](#)

→ Assigner au collaborateur le rôle *Admin* ou *Writer*



The screenshot shows a GitHub modal window titled "Add people to 420-2RP-HY-Cours4". At the top, there is a close button (X) and a repository icon. Below the title, a collaborator named "Giovana Velarde" with the username "giovanavg" is listed in a light blue box, also with a close button (X). Underneath, a section titled "Choose a role" contains a list of roles with radio buttons:

- ☐ **Read**
Recommended for non-code contributors who want to view or discuss your project.
- ☐ **Triage**
Recommended for contributors who need to manage issues and pull requests without write access.
- ☐ **Write**
Recommended for contributors who actively push to your project.
- ☐ **Maintain**
Recommended for project managers who need to manage the repository without access to sensitive or destructive actions.
- ☒ **Admin**
Recommended for people who need full access to the project, including sensitive and destructive actions like managing security or deleting a repository.

At the bottom of the modal is a green button labeled "Add giovanavg to this repository".

Le collaborateur doit [cloner le dépôt central dans son poste local](#) pour pouvoir travailler.