

GIT ET LA GESTION DE VERSIONS 1/3

Auteur : Sébastien Inion

Gestion de versions centralisée et décentralisée

- Gestion de versions décentralisée

- Git
- Mercurial
- Bazaar



Bazaar

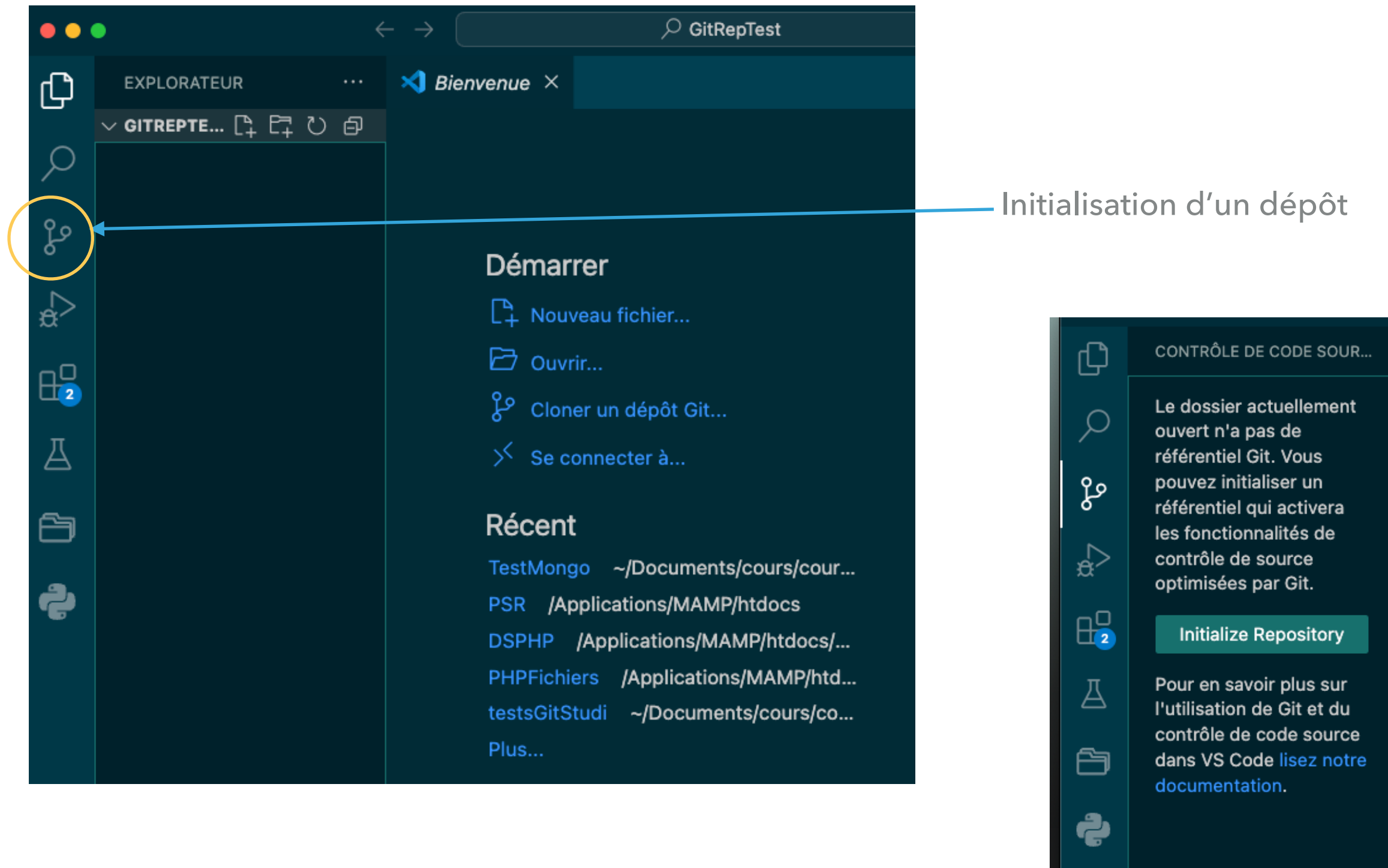
GIT

- *Git est **désynchronisée** (tout est sur votre ordinateur en local : fichiers et historique)
- *Un moment vous pouvez avoir besoin de vous synchroniser et cela se fera avec un serveur : dépôt (ex : Github)
- *Sur ces serveurs on trouve des projets, ou dépôts ou repository...

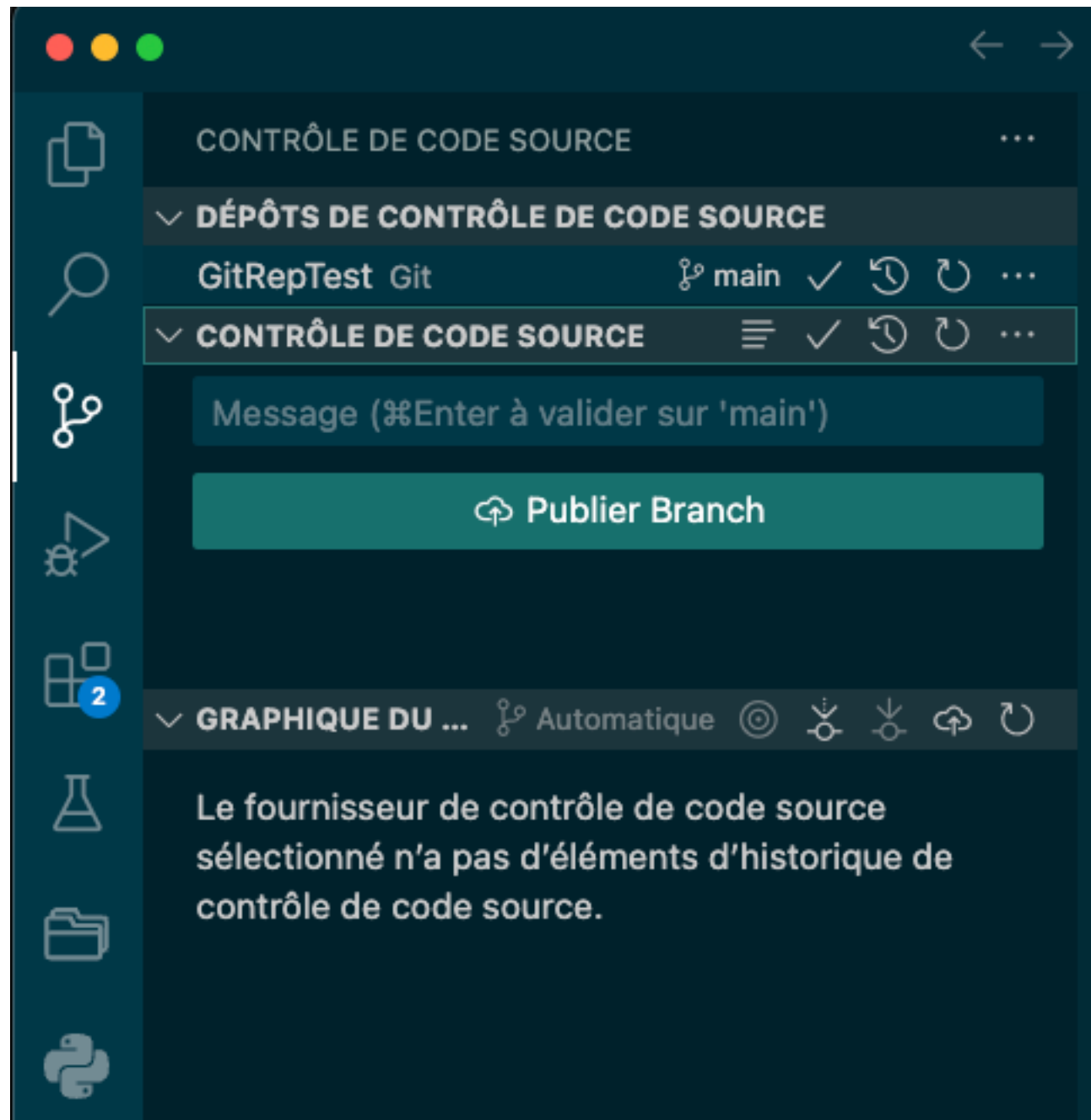


GIT ET LA GESTION DE VERSIONS

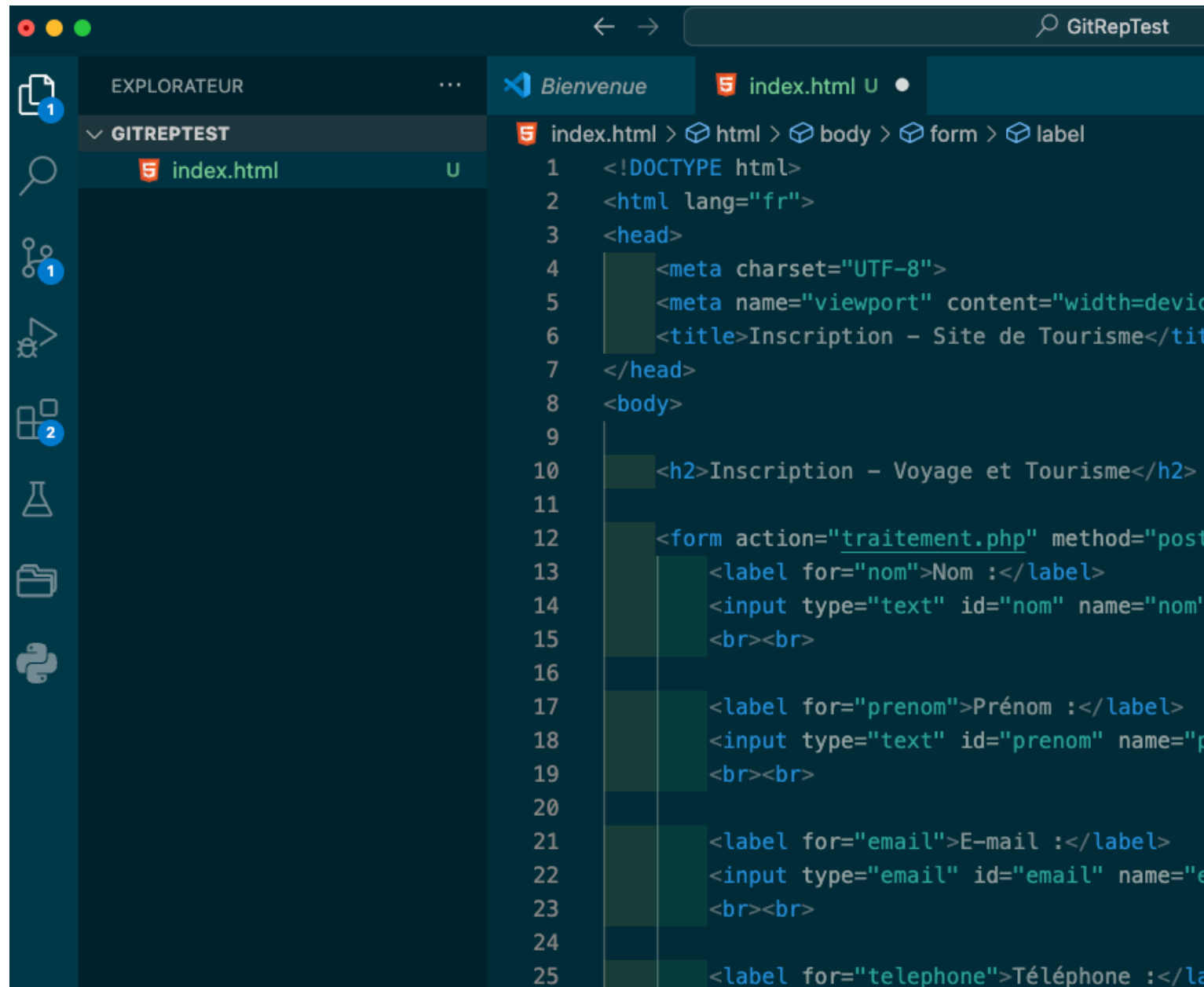
Sur VSC



Sur VSC



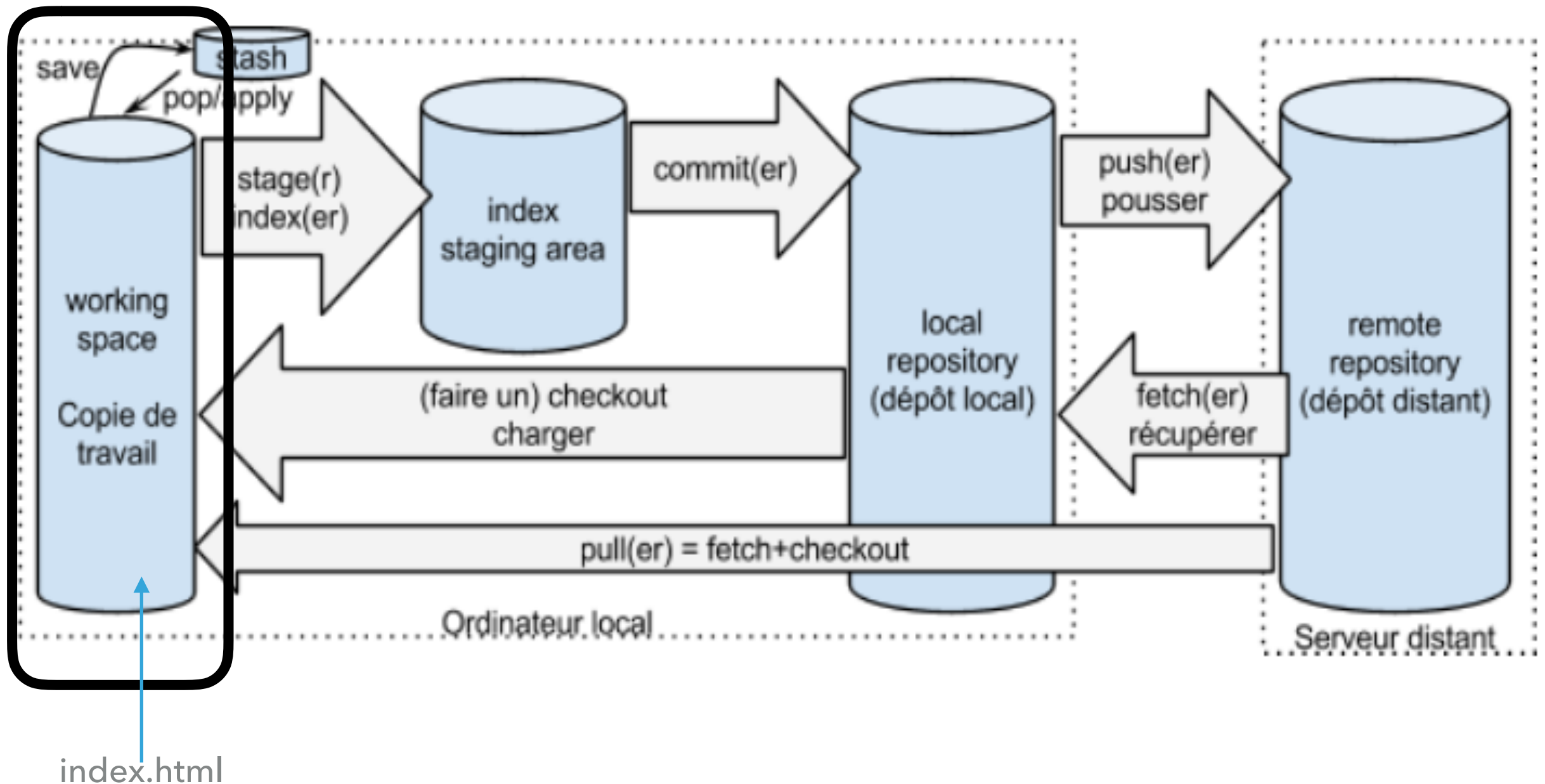
Sur VSC - Ajout d'un fichier



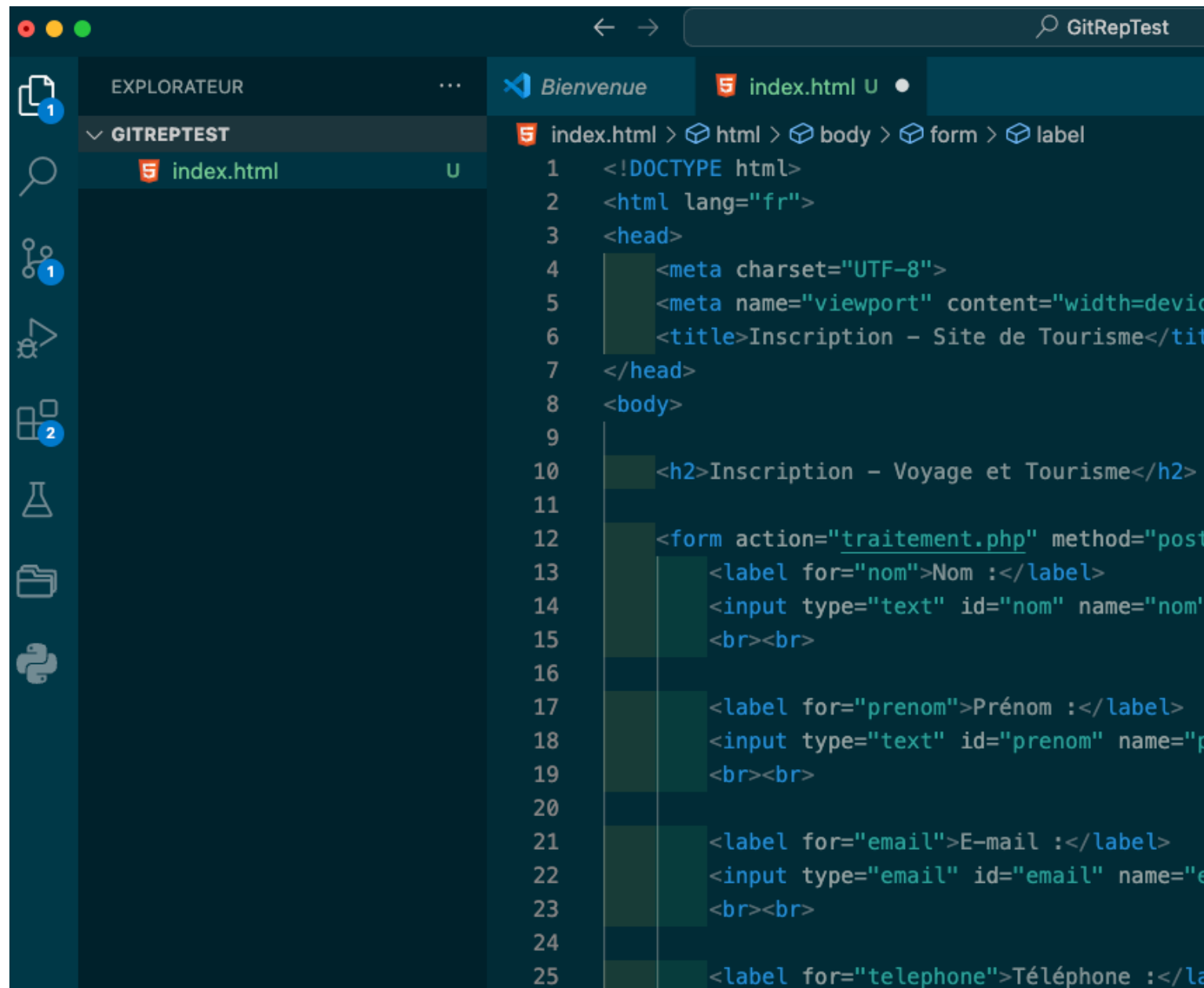
```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <title>Inscription - Site de Tourisme</title>
7 </head>
8 <body>
9
10  <h2>Inscription - Voyage et Tourisme</h2>
11
12  <form action="traitement.php" method="post">
13    <label for="nom">Nom :</label>
14    <input type="text" id="nom" name="nom">
15    <br><br>
16
17    <label for="prenom">Prénom :</label>
18    <input type="text" id="prenom" name="prenom">
19    <br><br>
20
21    <label for="email">E-mail :</label>
22    <input type="email" id="email" name="email">
23    <br><br>
24
25    <label for="telephone">Téléphone :</label>
```

Vocabulaire spécifique à Git

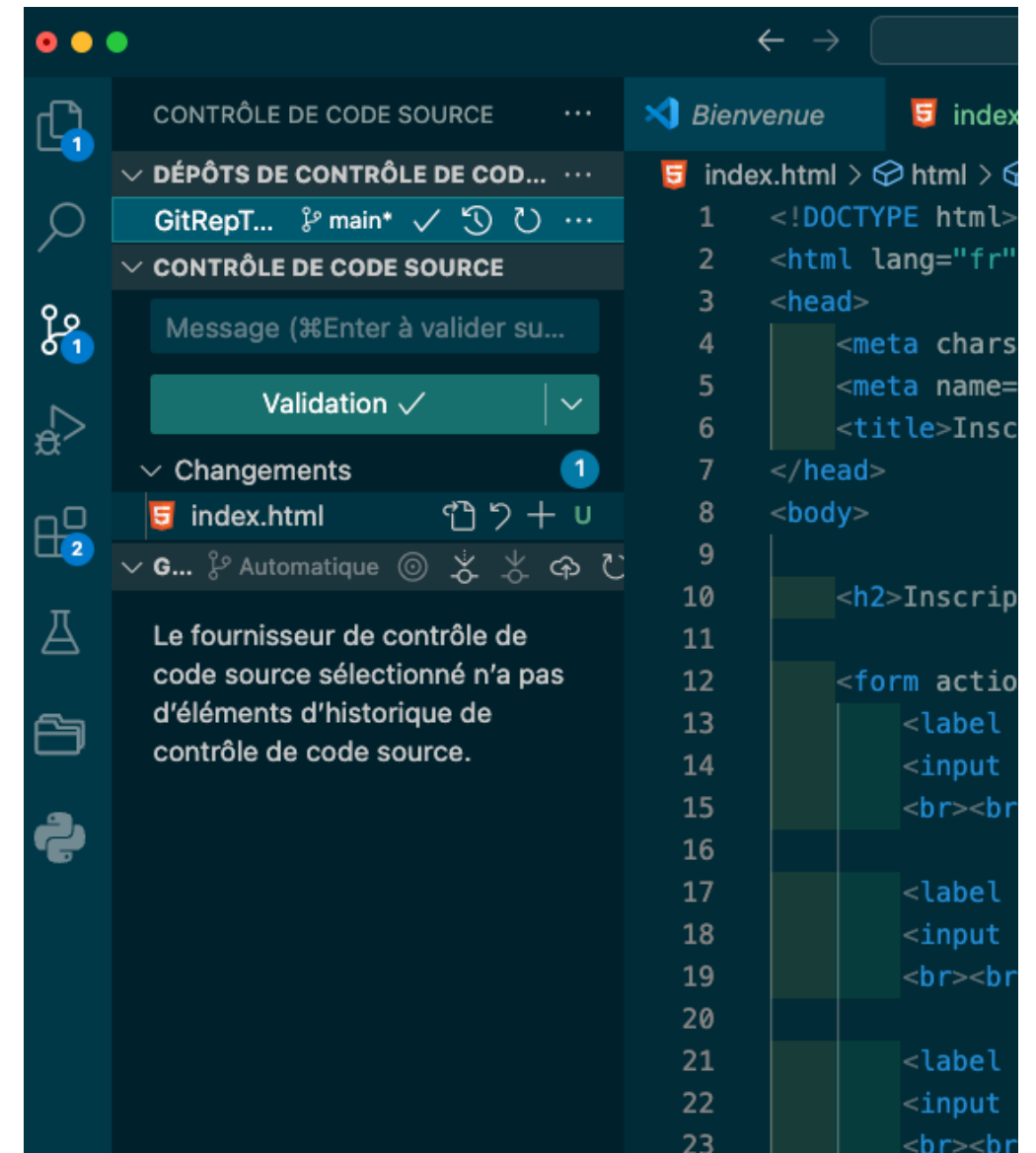
Vocabulaire spécifique à GIT



Sur VSC - Commit



```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <title>Inscription - Site de Tourisme</title>
7 </head>
8 <body>
9
10  <h2>Inscription - Voyage et Tourisme</h2>
11
12  <form action="traitement.php" method="post">
13    <label for="nom">Nom :</label>
14    <input type="text" id="nom" name="nom">
15    <br><br>
16
17    <label for="prenom">Prénom :</label>
18    <input type="text" id="prenom" name="prenom">
19    <br><br>
20
21    <label for="email">E-mail :</label>
22    <input type="email" id="email" name="email">
23    <br><br>
24
25    <label for="telephone">Téléphone :</label>
```



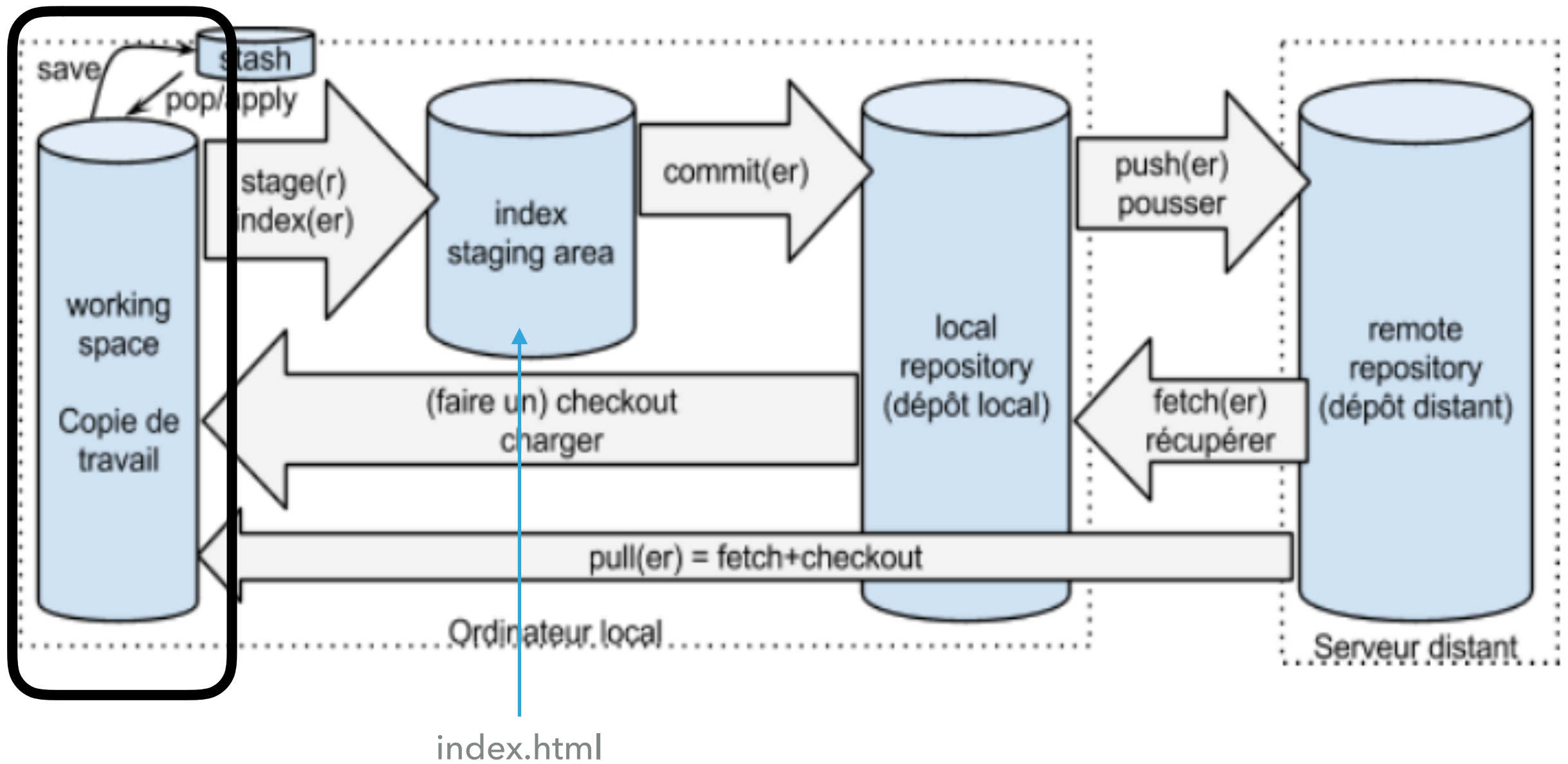
```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <title>Inscription - Site de Tourisme</title>
7 </head>
8 <body>
9
10  <h2>Inscription - Voyage et Tourisme</h2>
11
12  <form action="traitement.php" method="post">
13    <label for="nom">Nom :</label>
14    <input type="text" id="nom" name="nom">
15    <br><br>
16
17    <label for="prenom">Prénom :</label>
18    <input type="text" id="prenom" name="prenom">
19    <br><br>
20
21    <label for="email">E-mail :</label>
22    <input type="email" id="email" name="email">
23    <br><br>
24
25    <label for="telephone">Téléphone :</label>
```

On peut **commiter** car on a fini la partie HTML de mon formulaire.

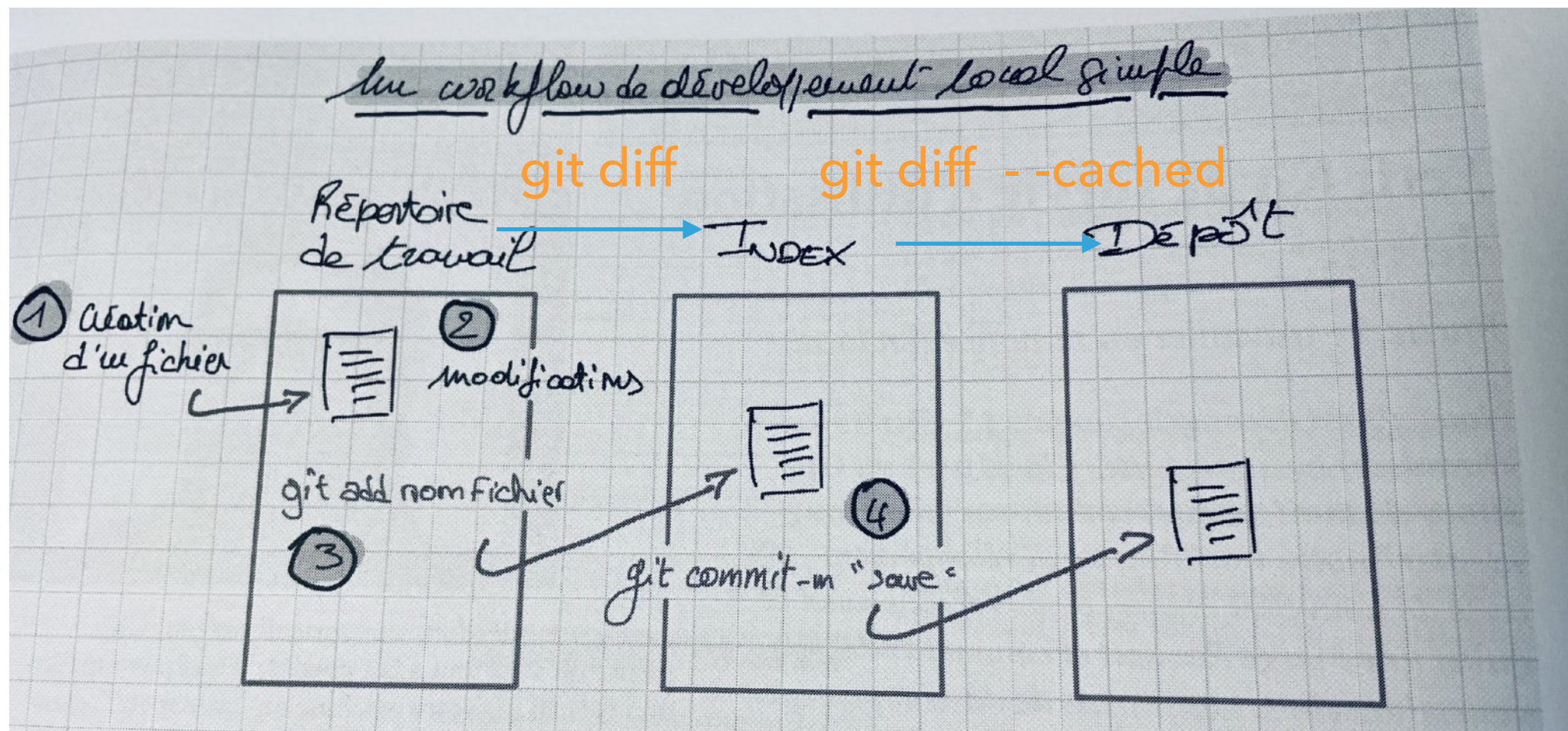
Rappel : Un commit n'est pas une sauvegarde mais une validation d'une étape dans la réalisation de votre projet !

Vocabulaire spécifique à Git

Vocabulaire spécifique à GIT



EN LOCAL



git status : a faire régulièrement pour savoir où vous en êtes.

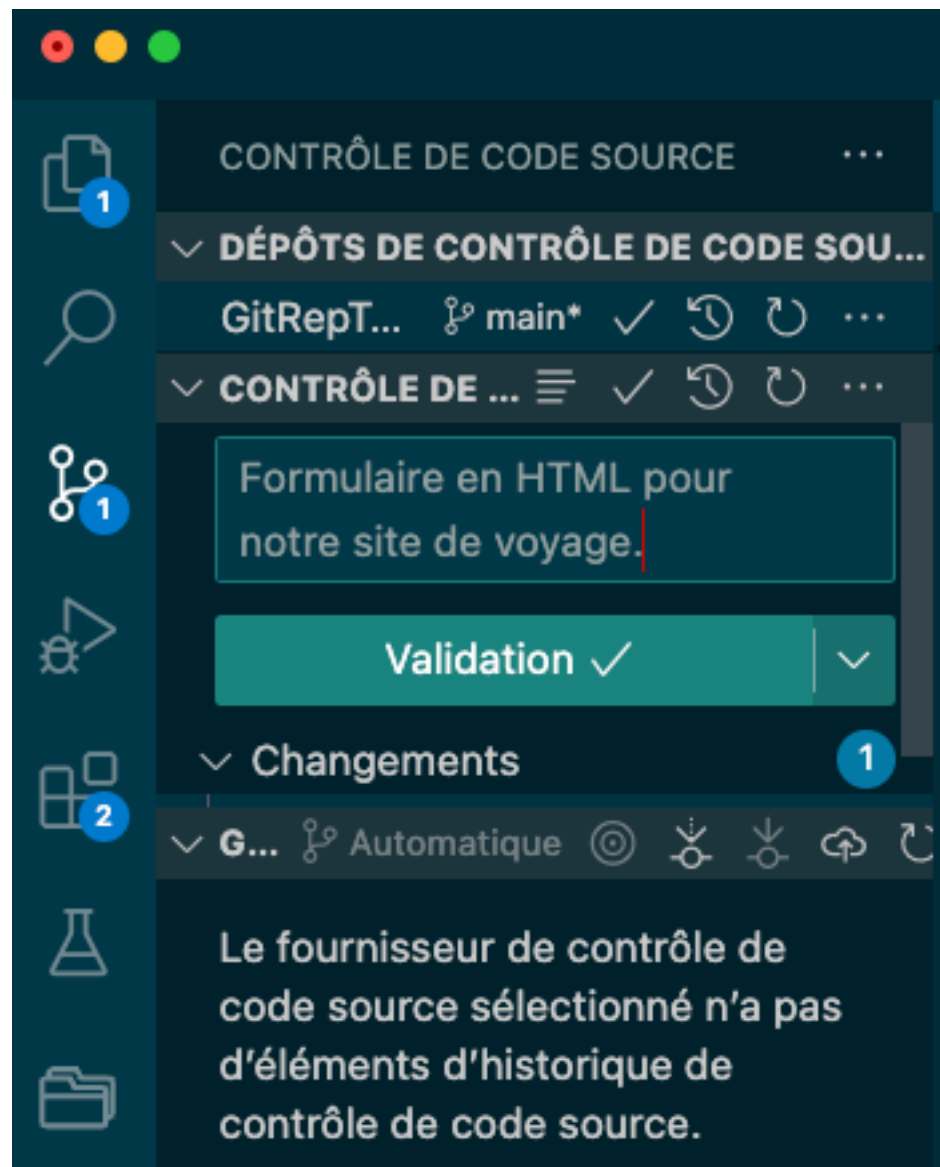
Étape 1 : Création ou modification d'un fichier dans le répertoire de travail.

Étape 2 : Ajout des modifications au staging area (ou index) avec `git add nom_fichier`.

Étape 3 : Enregistrement des modifications dans le dépôt local avec `git commit -m "message"`.

Étape 4 : Les modifications sont désormais sauvegardées dans l'historique du dépôt Git.

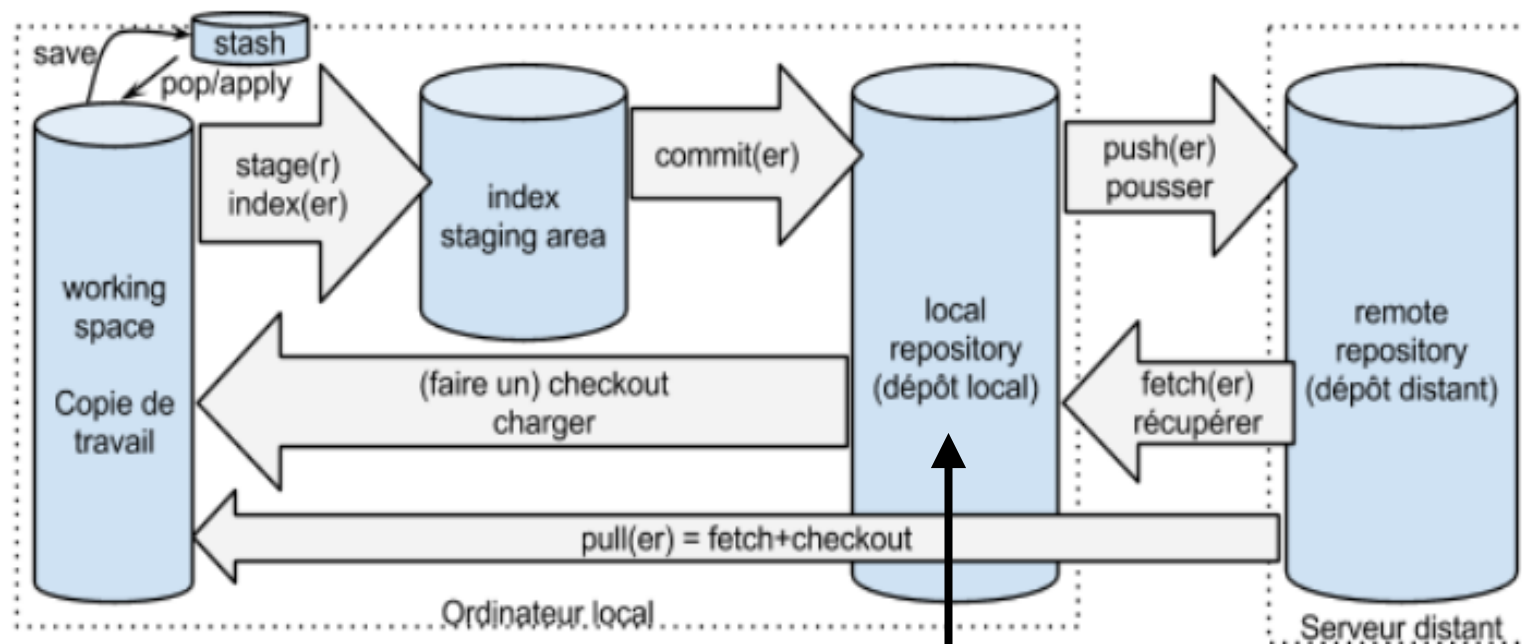
Sur VSC - On commit



Il est important de bien remplir ce champ message !

Ajouter un fichier à l'Index

Vocabulaire spécifique à GIT



index.html

On teste notre projet

Inscription - Voyage et Tourisme

Nom :

Prénom :

E-mail :

Téléphone :

Destination préférée : ▼

Pourquoi souhaitez-vous voyager ?

☐ S'abonner à notre newsletter

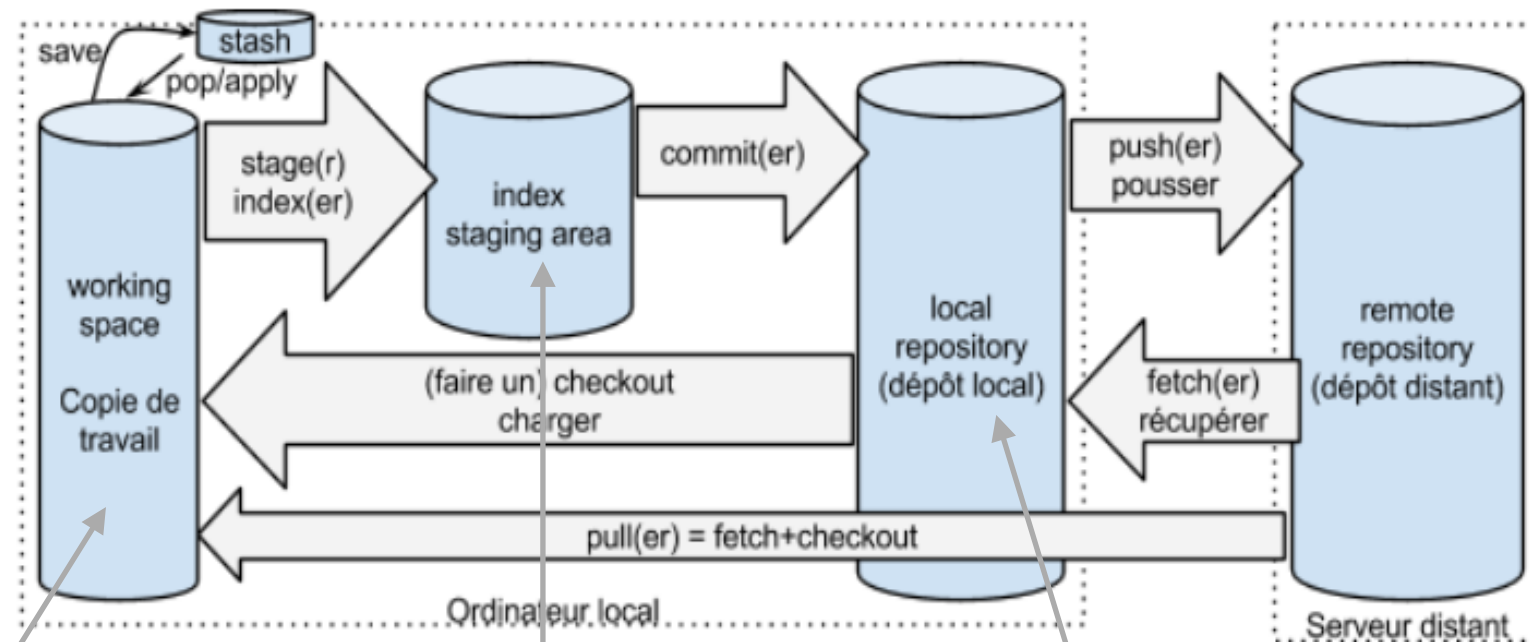
arbres

votre dépôt local est composé de trois "arbres" gérés par git. le premier est votre **espace de travail** qui contient réellement vos fichiers. le second est un **Index** qui joue un rôle d'espace de transit pour vos fichiers et enfin **HEAD** qui pointe vers la dernière validation que vous ayez fait.



Lister les fichiers

Vocabulaire spécifique à GIT



ls

git ls-files —stage

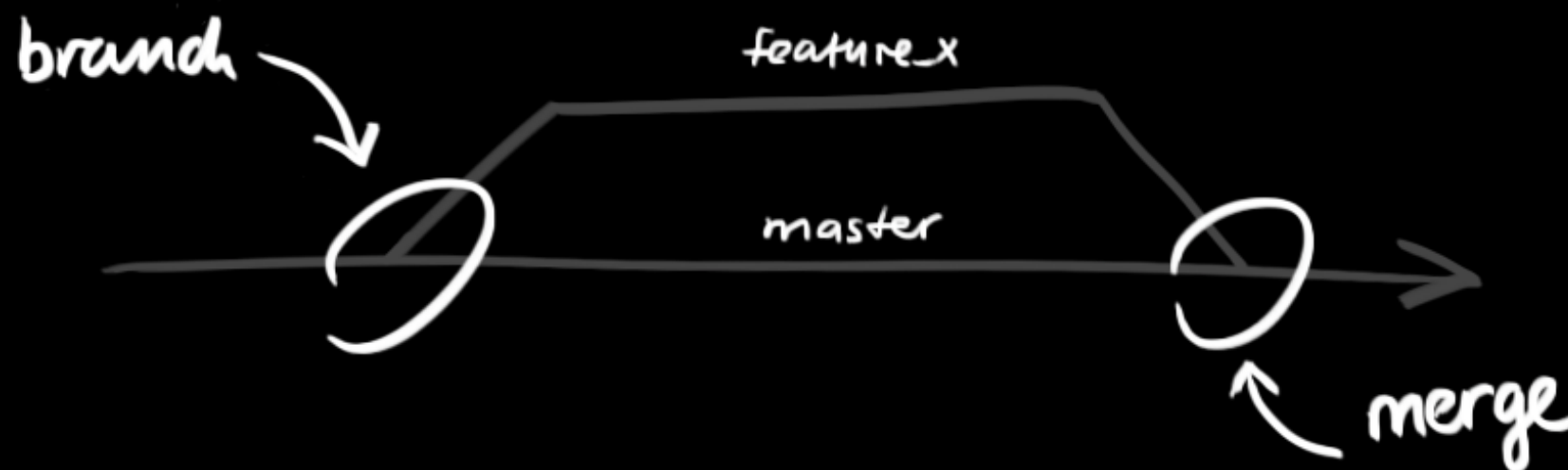
git ls-tree -r HEAD

LE FLUX DE TRAVAIL

Branches

branches

Les branches sont utilisées pour développer des fonctionnalités isolées des autres. La branche *master* est la branche par défaut quand vous créez un dépôt. Utilisez les autres branches pour le développement et fusionnez ensuite à la branche principale quand vous avez fini.



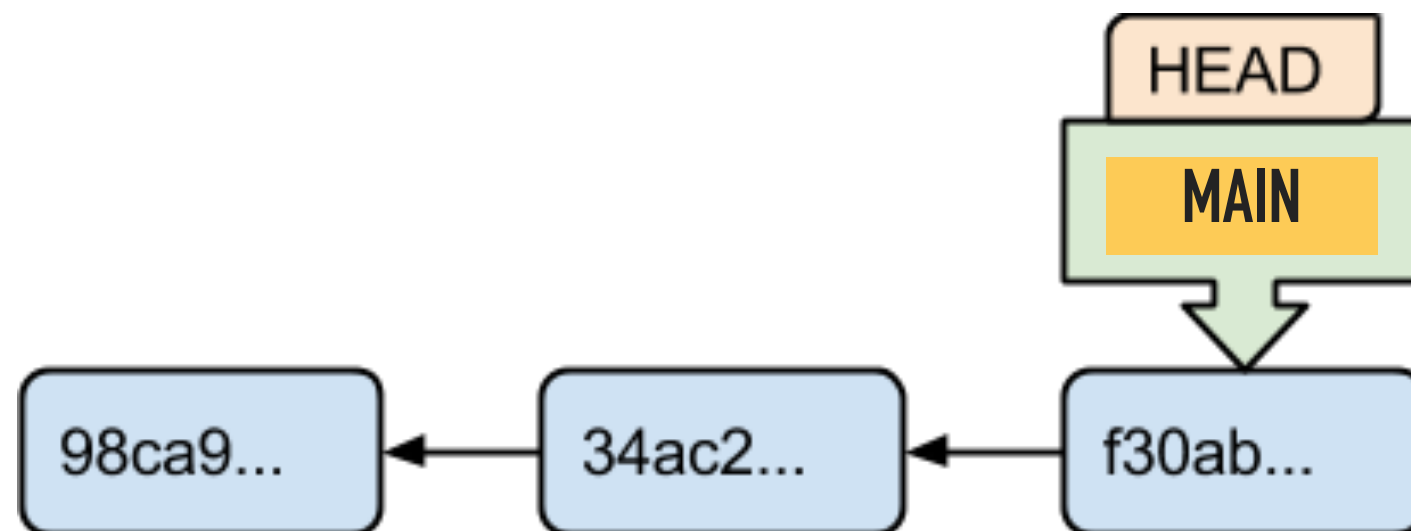
Git et les branches

Première branche : **main**
Une branche pointe sur un commit



Git et les branches

Un pointeur spécial nommé HEAD pointe vers la branche courante de travail !

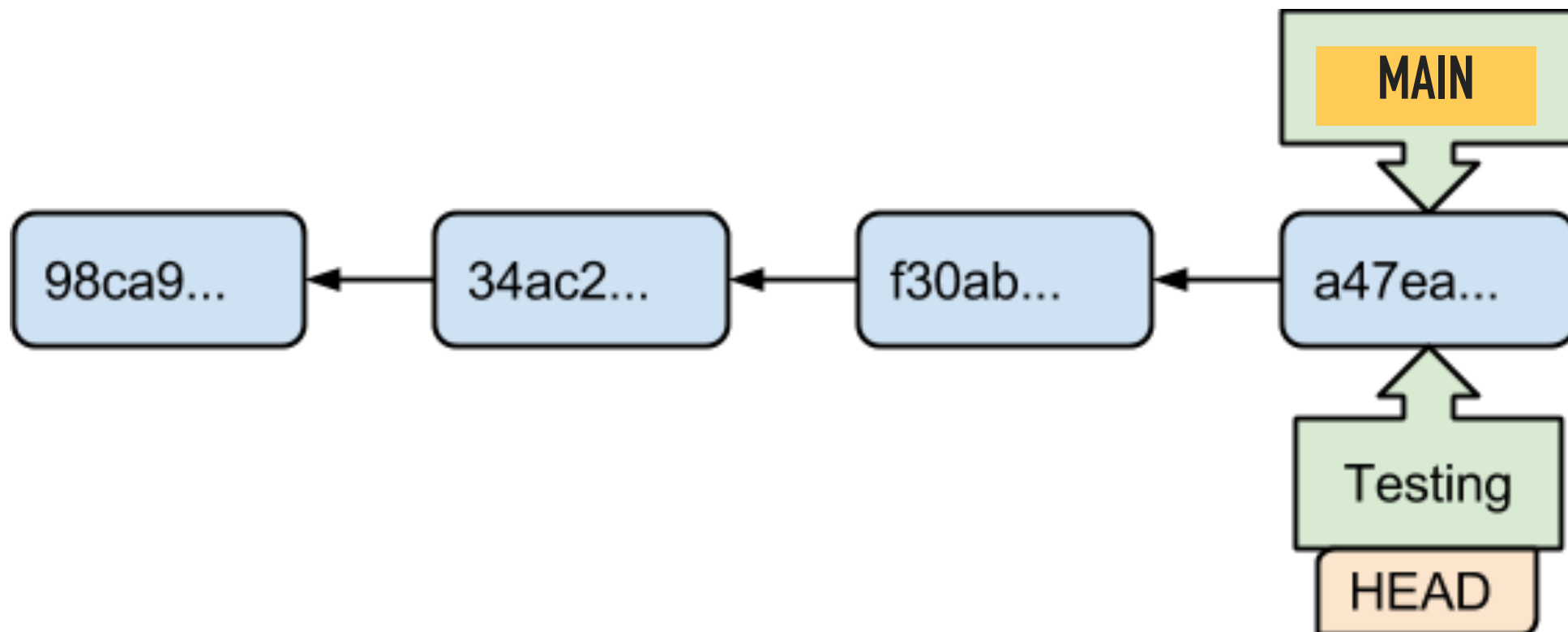


Git et les branches

Création d'une nouvelle branche

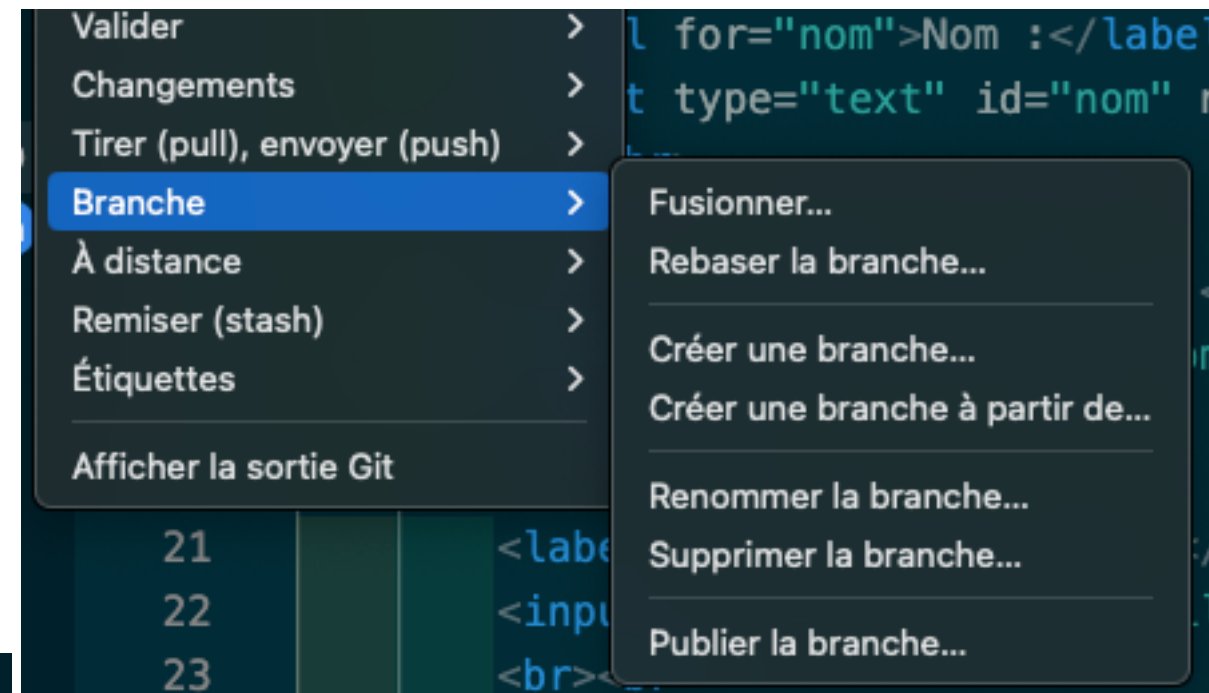
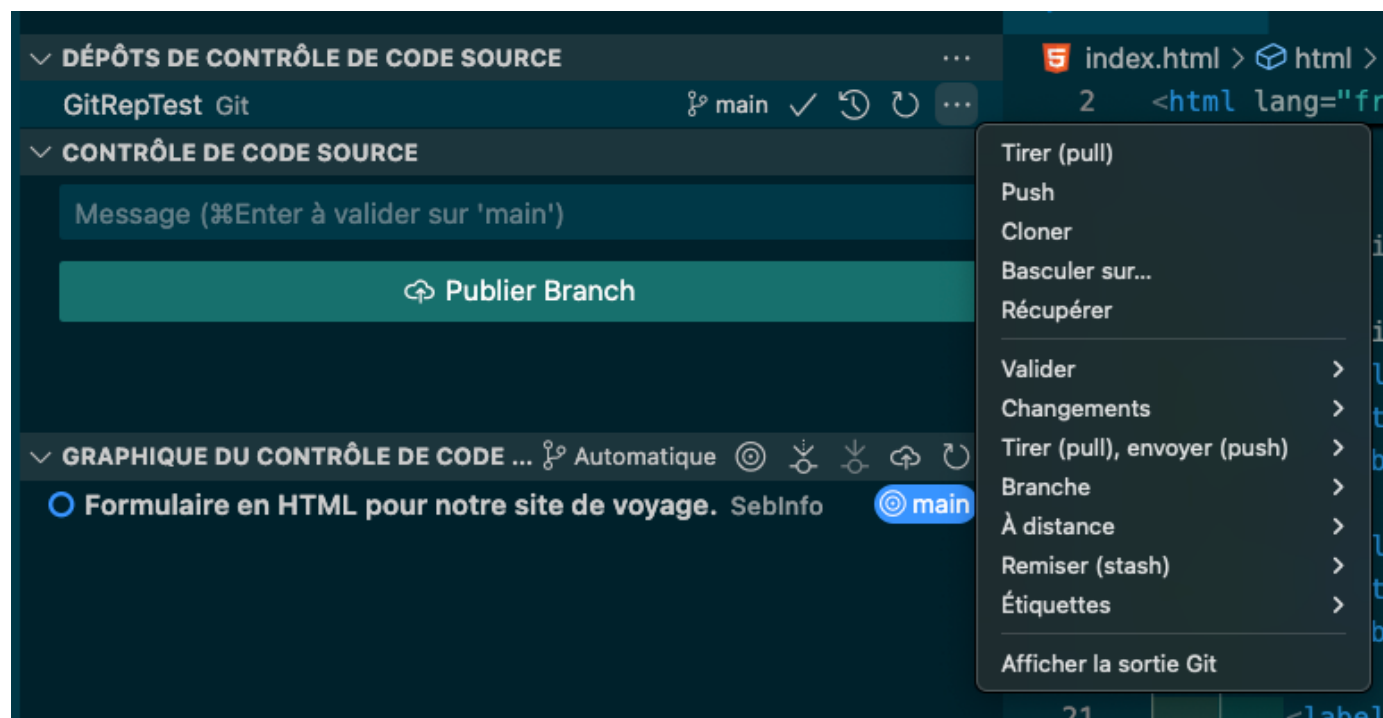
\$git branch Testing

\$git checkout Testing

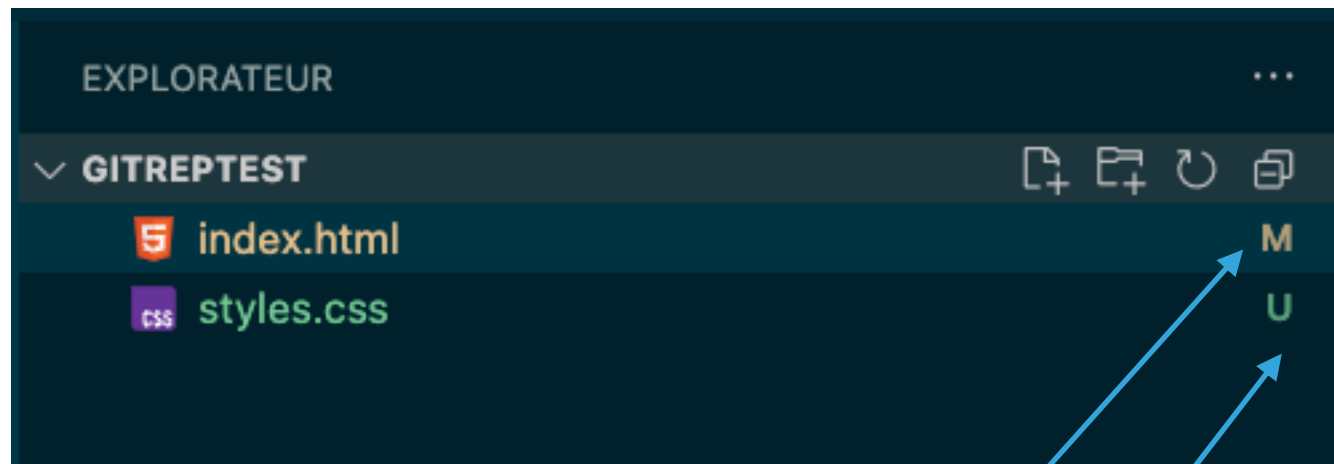


Les branches

On va créer une branche pour tester notre projet en ajoutant du CSS



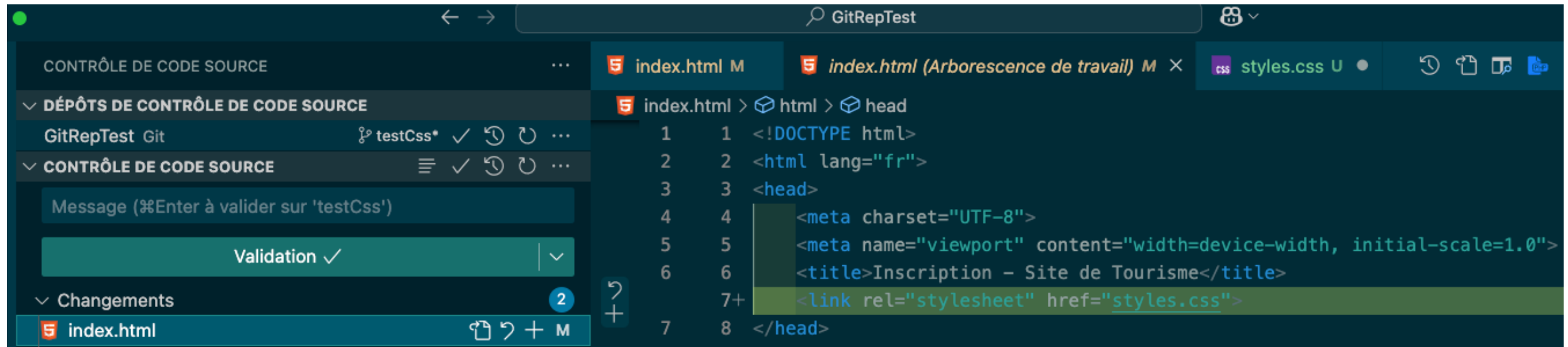
Les branches



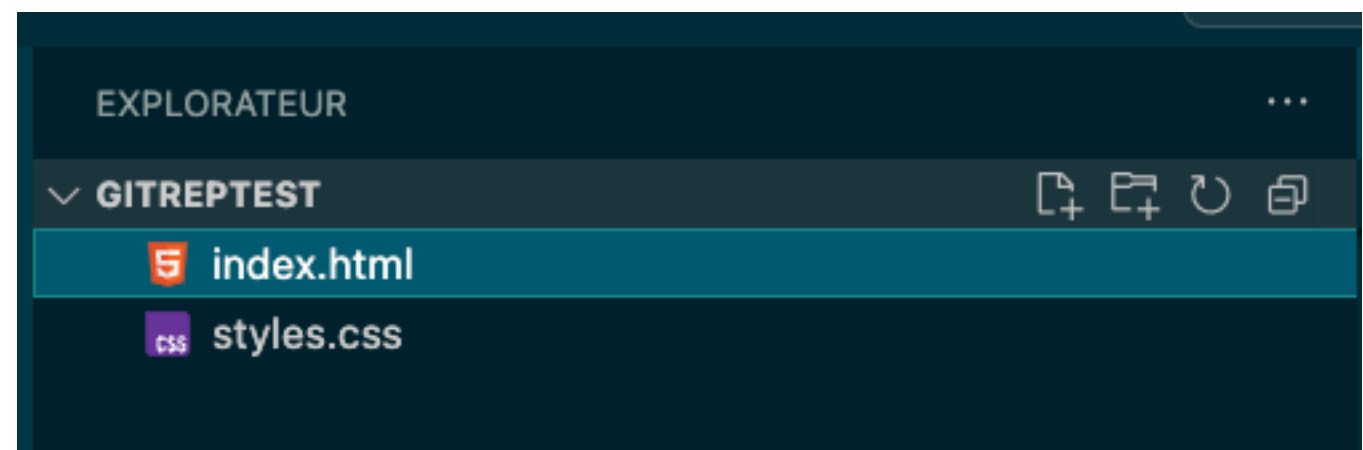
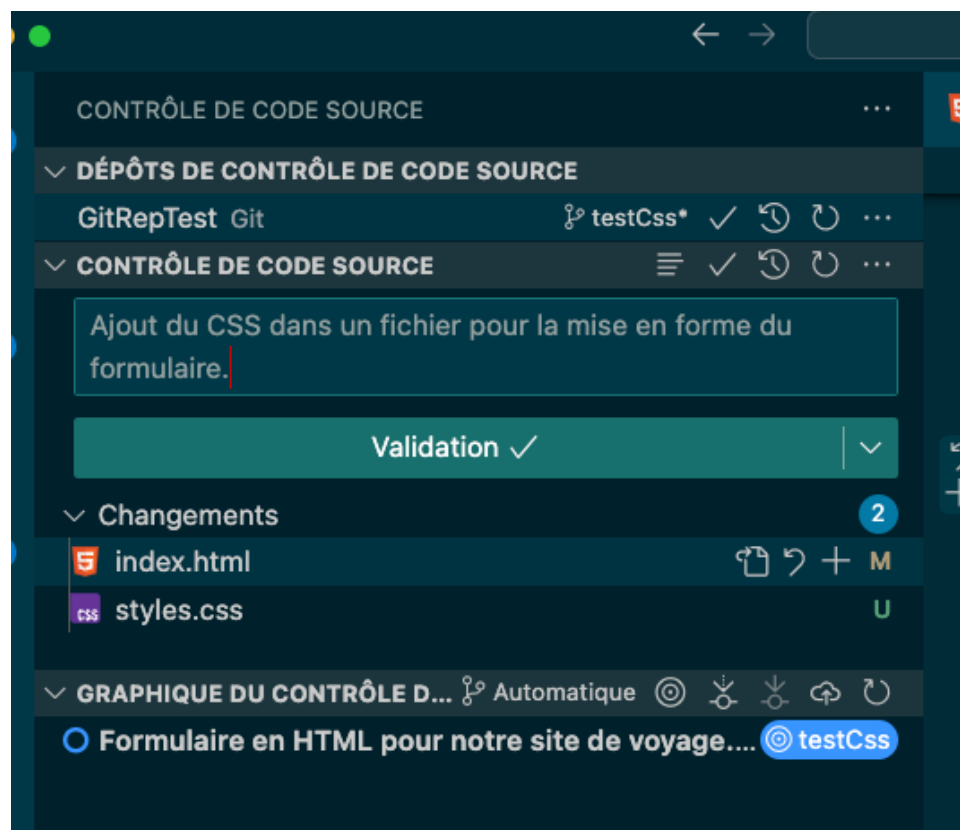
index.html (M - Modified) → Ce fichier a été modifié, mais les changements n'ont pas encore été committés.

styles.css (U - Untracked) → Ce fichier est nouveau et non suivi par Git, ce qui signifie qu'il n'a jamais été ajouté à l'index.

Les branches



On commit dans cette nouvelle branche



On test la branche testCss

Inscription - Voyage et Tourisme

Nom :

Prénom :

E-mail :

Téléphone :

Destination préférée :

Paris, France



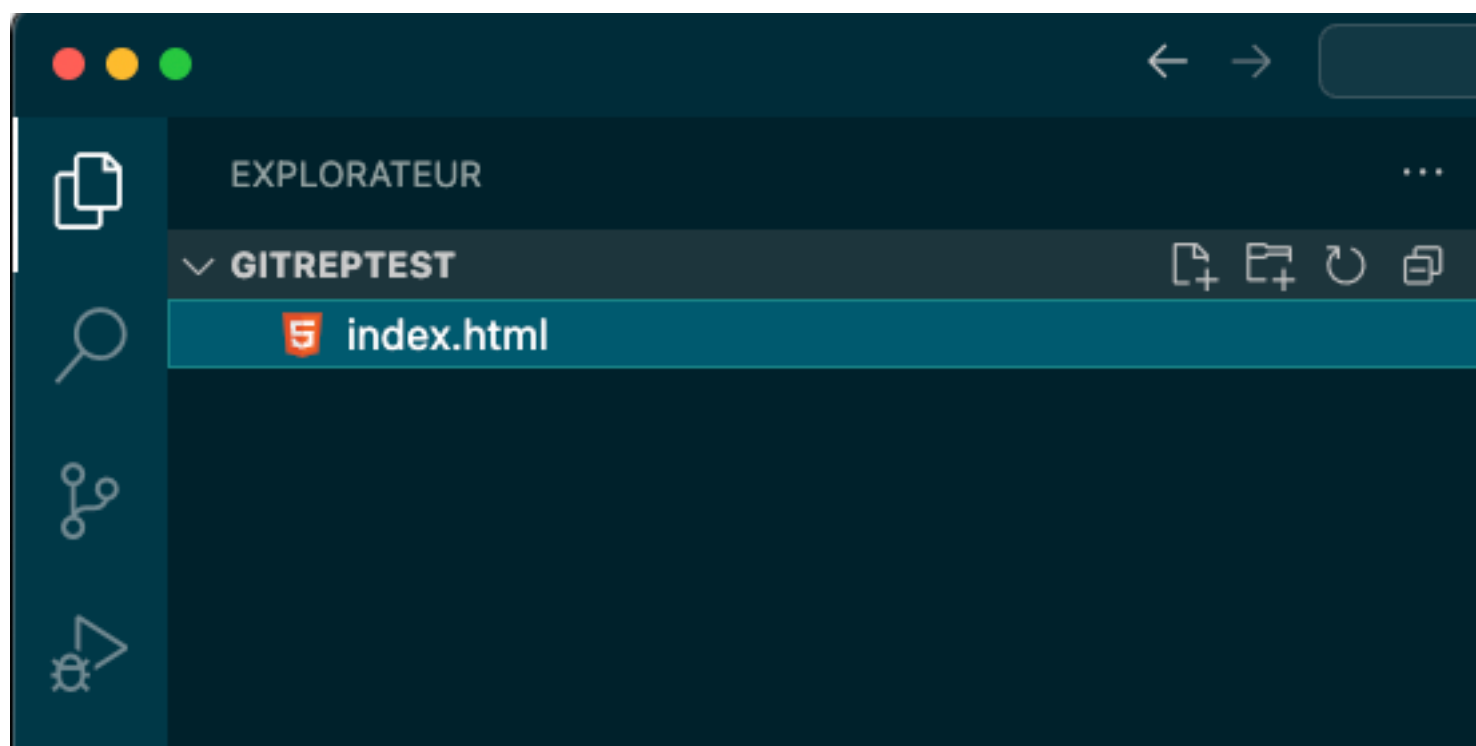
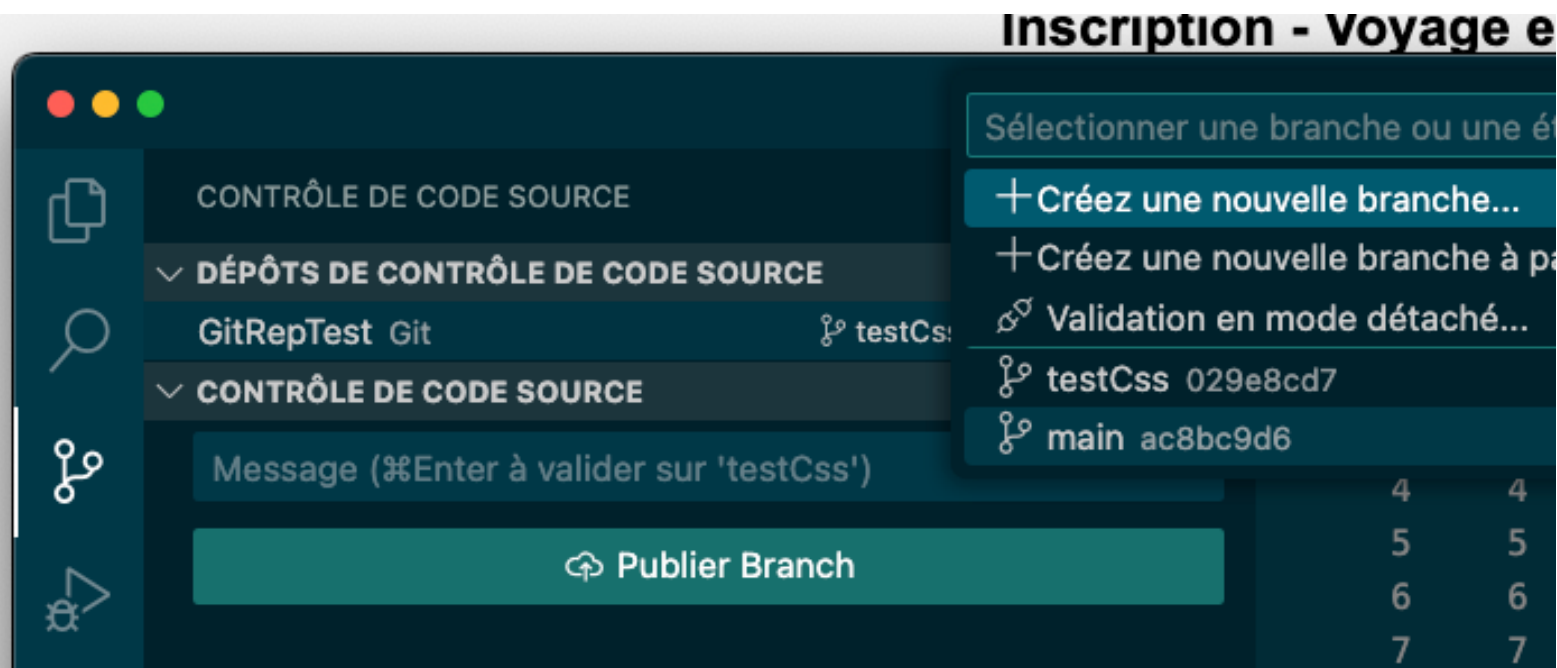
Pourquoi souhaitez-vous voyager ?

☐

S'abonner à notre newsletter

S'inscrire

On repasse sur la branche main



Inscription - Voyage et Tourisme

Nom :

Prénom :

E-mail :

Téléphone :

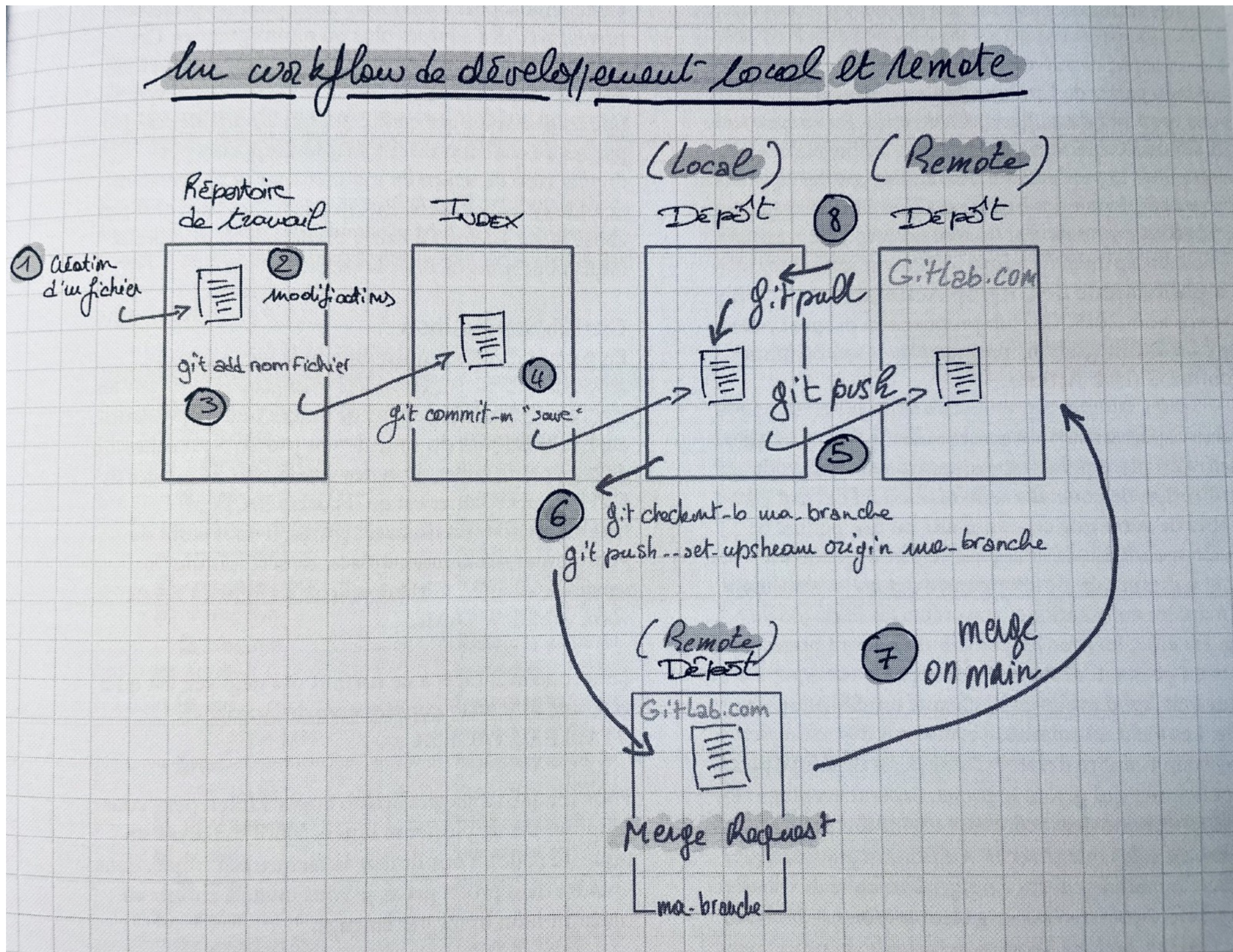
Destination préférée :

Pourquoi souhaitez-vous voyager ?

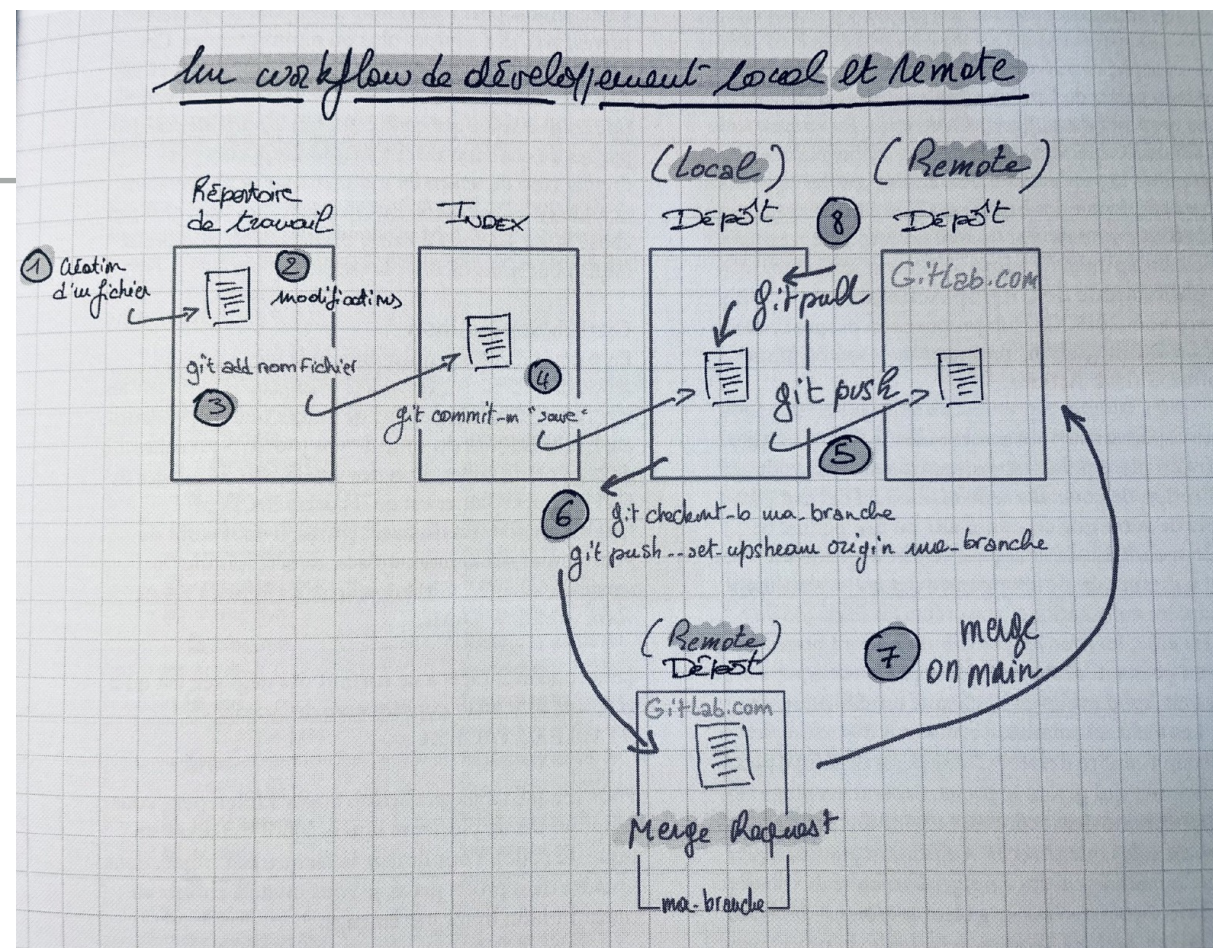
☐ S'abonner à notre newsletter

PASSER PAR UN DÉPÔT

FLUX DE TRAVAIL



FLUX DE TRAVAIL



Étape 1 : Création ou modification d'un fichier dans le répertoire de travail.

Étape 2 : Ajout des modifications au staging area (ou index) avec la commande `git add nom_fichier`.

Étape 3 : Enregistrement des modifications dans le dépôt local avec `git commit -m "message"`.

Étape 4 : Création d'une branche locale avec `git checkout -b ma_branche` et association au dépôt distant avec `git push --set-upstream origin ma_branche`.

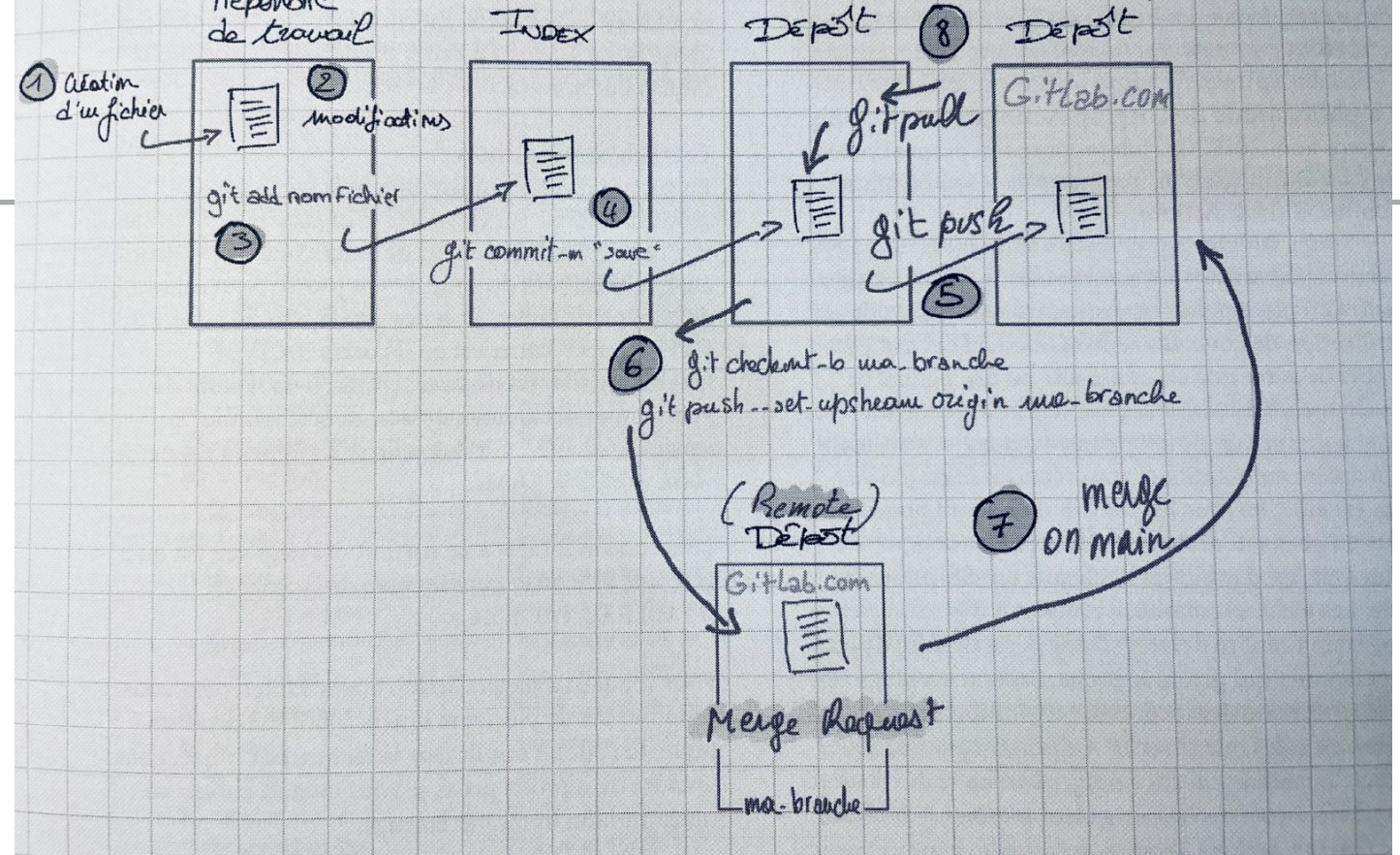
Étape 5 : Envoi des modifications sur le dépôt distant (ex. GitLab ou GitHub) avec `git push`. Après avoir créé une **nouvelle branche** et fait des modifications dessus, il faut l'envoyer sur le dépôt distant (ex: **GitHub/****GitLab**) pour que d'autres développeurs puissent la voir et l'examiner.

Étape 6 : Création d'une merge request dans l'interface du dépôt distant pour proposer la fusion des changements.

Étape 7 : Validation et fusion de la branche sur la branche principale (ex. main).

Étape 8 : permet de récupérer les dernières modifications du dépôt distant vers le dépôt local.

FLUX DE TRAVAIL



L'étape 5 est importante car elle permet d'envoyer les commits **du dépôt local vers le dépôt distant** (ex: GitHub/ GitLab).

Rendre les modifications **accessibles aux autres développeurs**.

Prépare le terrain pour une Merge Request (Pull Request).

L'étape 6 : Demander une revue de code avant d'intégrer la branche dans `main`.

Permettre aux autres développeurs de vérifier et commenter le code.

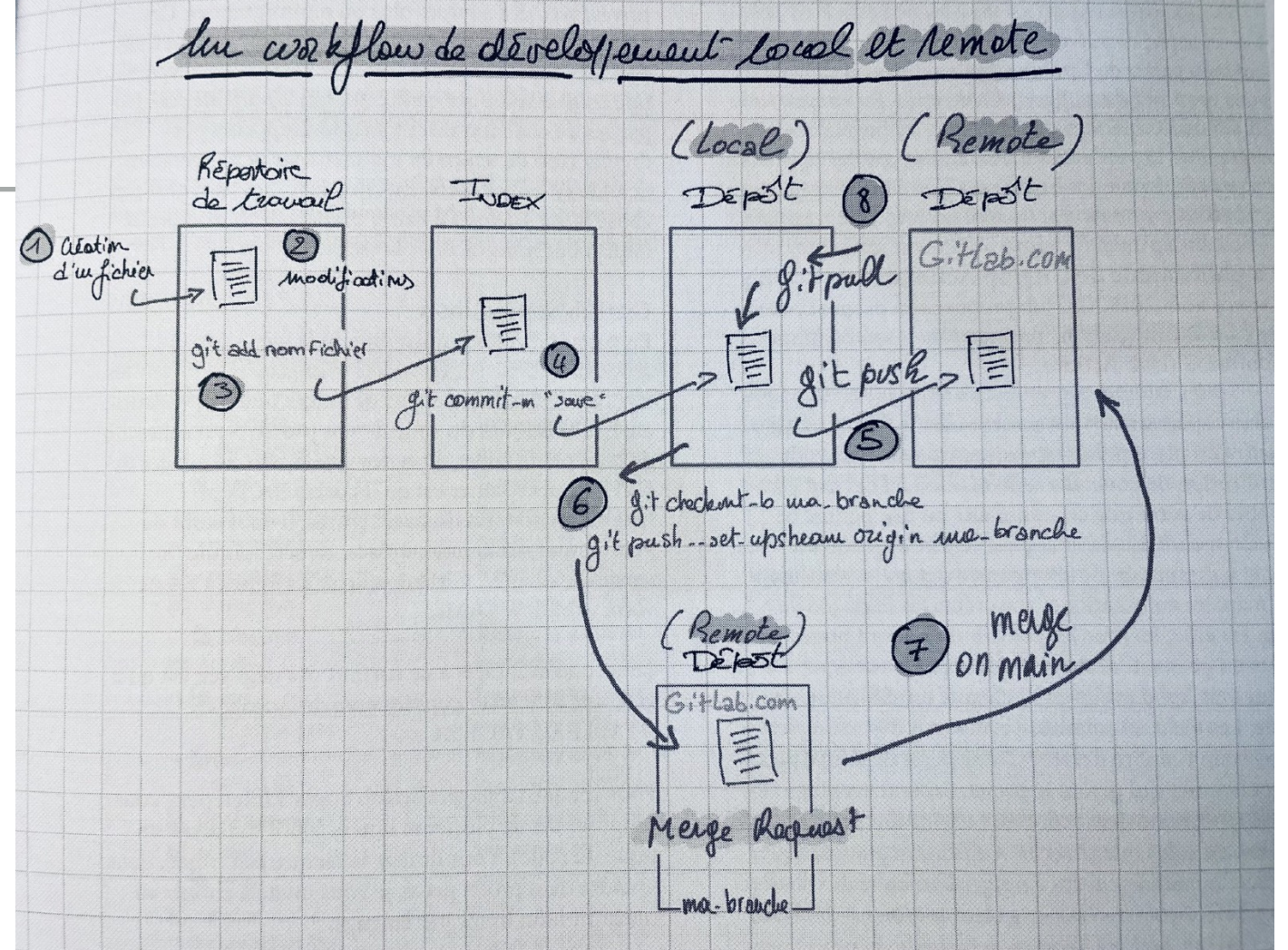
Éviter d'introduire des erreurs ou bugs dans `main`.

Comment faire sur GitHub/GitLab ?

1. Aller sur **GitHub/GitLab**.
2. Sélectionner la branche `ma-branch`.
3. Cliquer sur **"New Merge Request" (GitLab) / "New Pull Request" (GitHub)**.
4. Comparer avec `main` et ajouter un **titre + description**.
5. Assigner des **reviewers** pour examiner le code.

Le code est **prêt à être fusionné dans `main`**, mais attend une validation.

FLUX DE TRAVAIL



L'étape 7 : Fusion de la branche avec main (merge)

📌 Rôle :

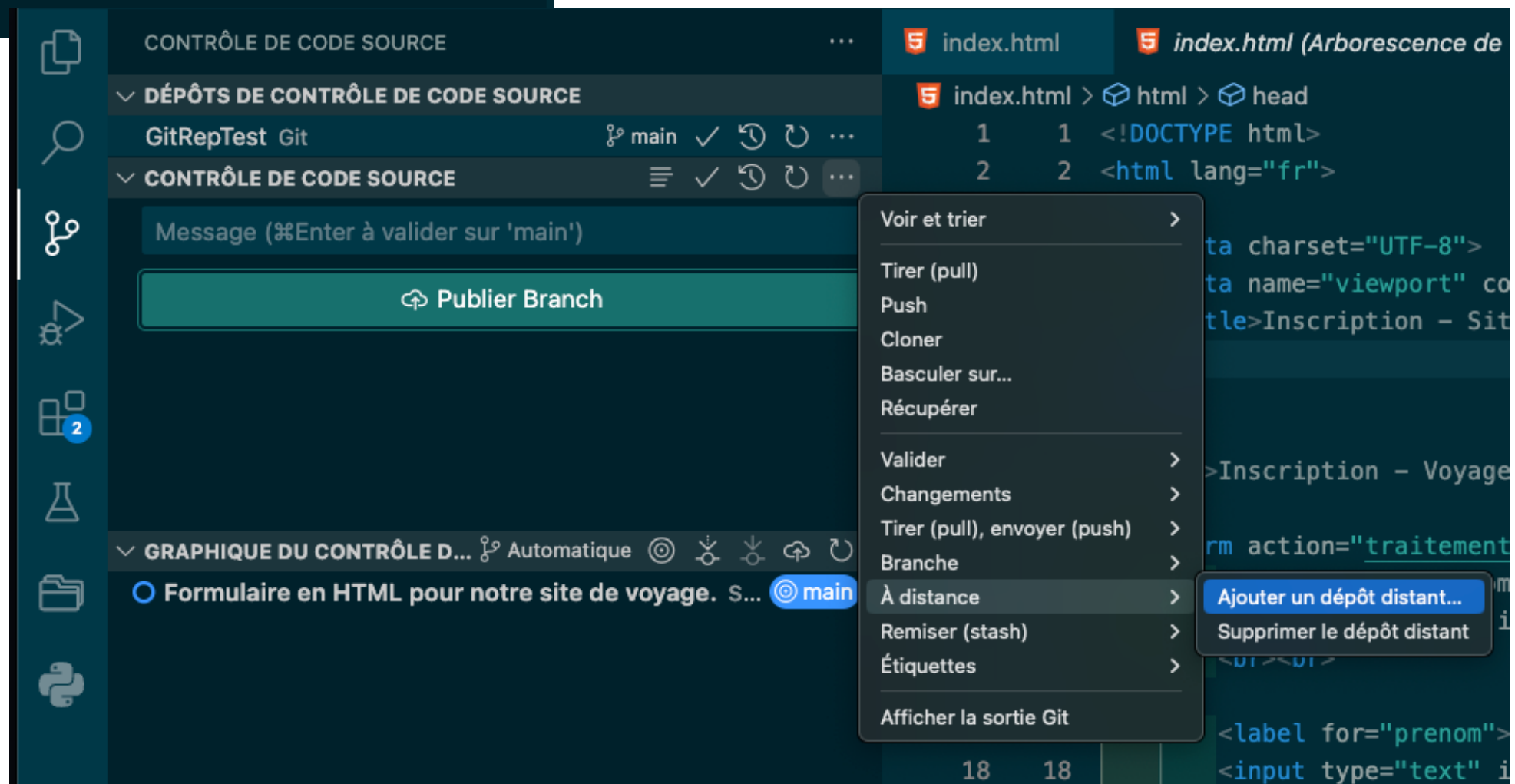
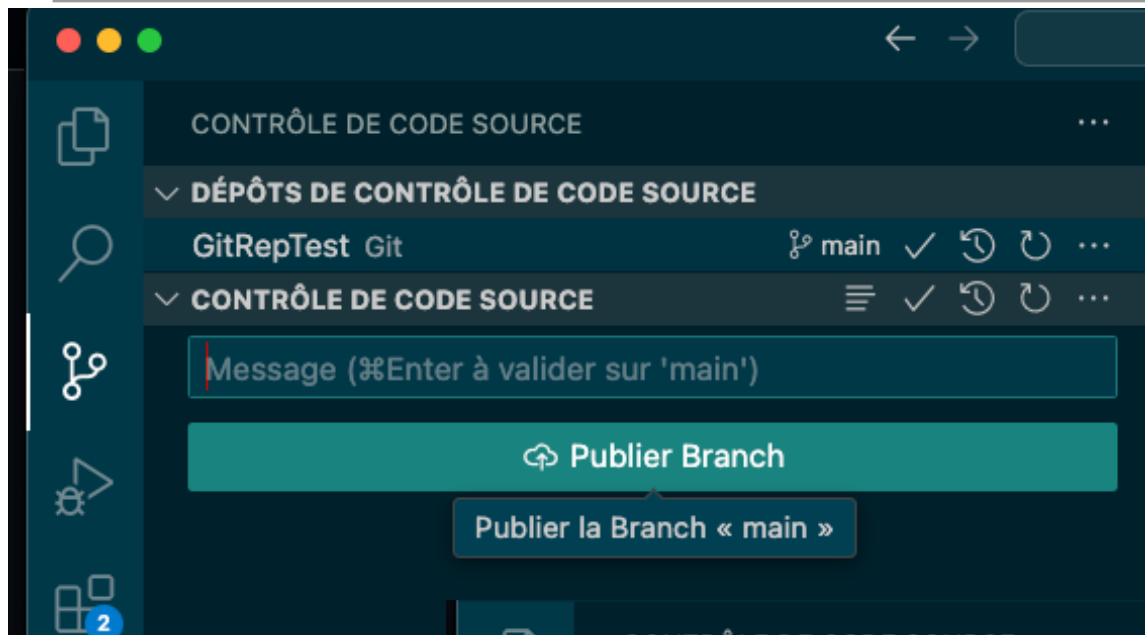
- Intégrer les modifications dans **main** une fois validées.
- Finaliser la mise à jour du code.

L'étape 8 : Mise à jour du dépôt local avec `git pull`

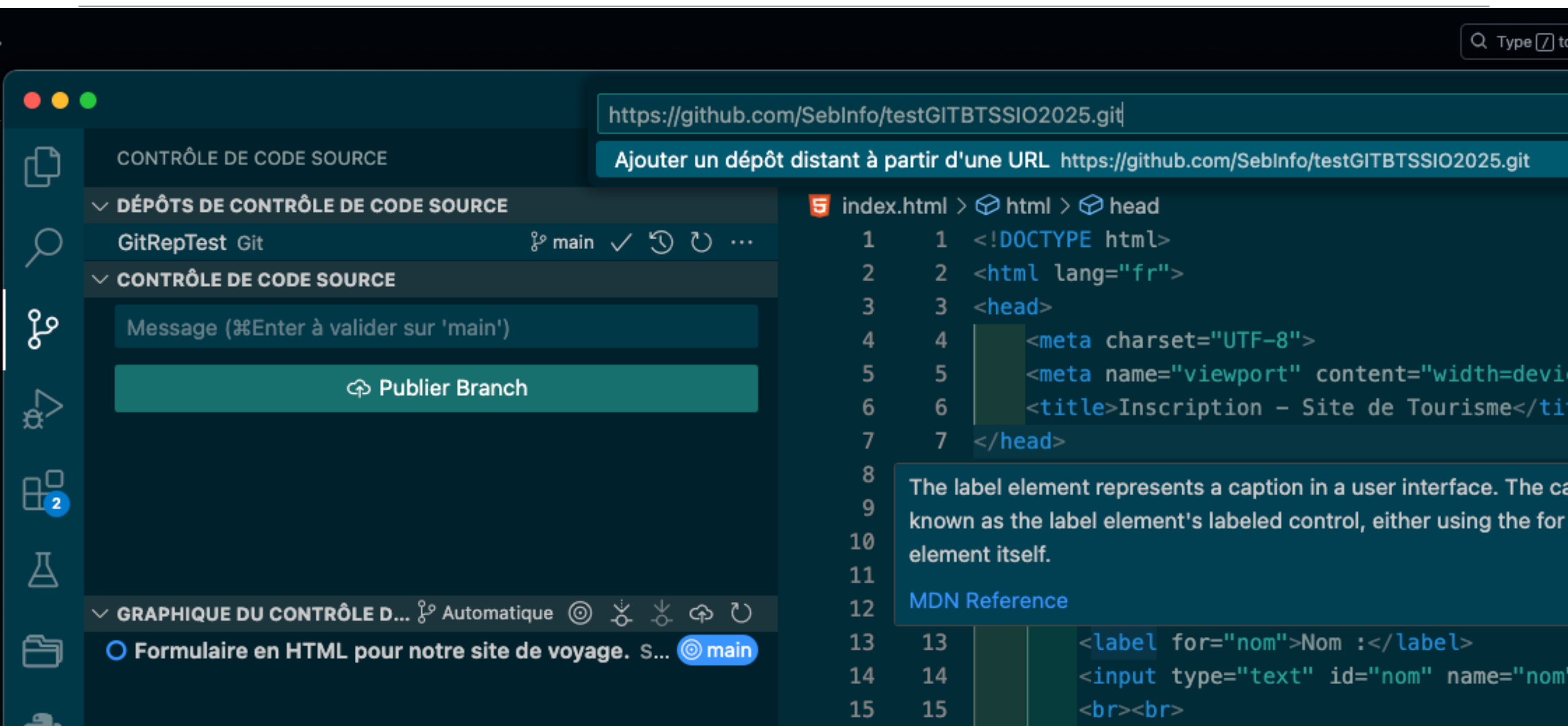
📌 Rôle :

- Synchroniser le dépôt local avec les dernières modifications de **main**.
- Éviter les conflits lors du prochain développement.

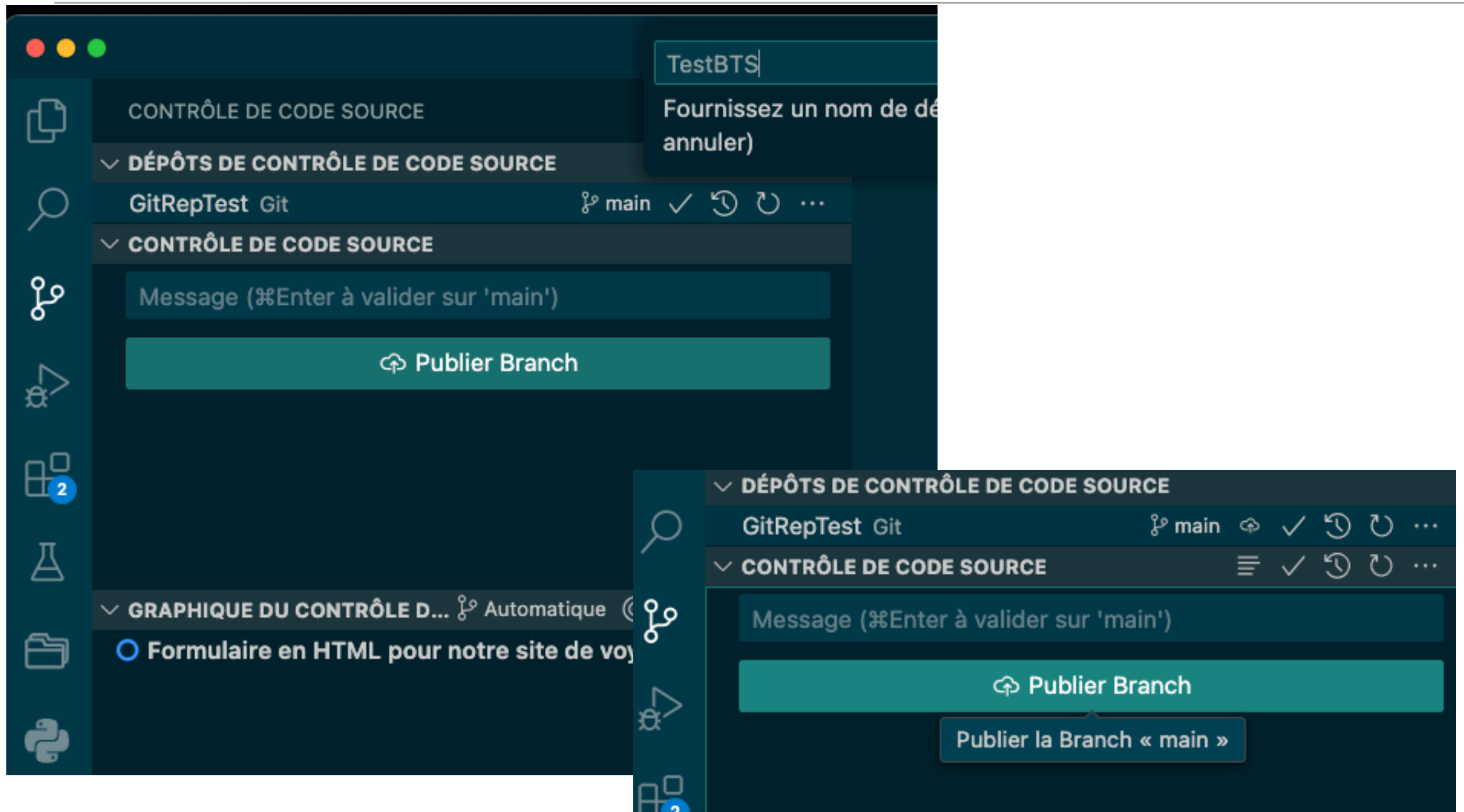
Ajouter un dépôt distant



On envoie la branche main sur le dépôt

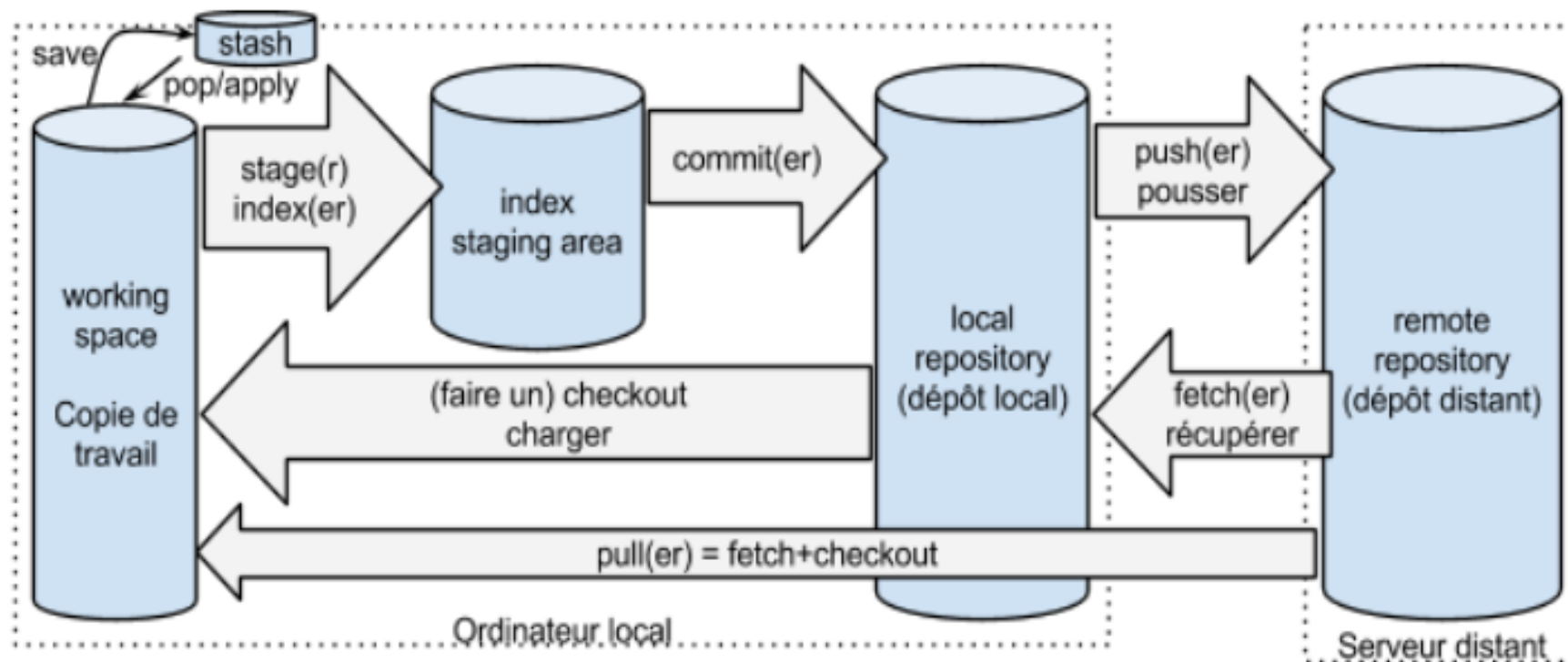


On envoie la branche main sur le dépôt

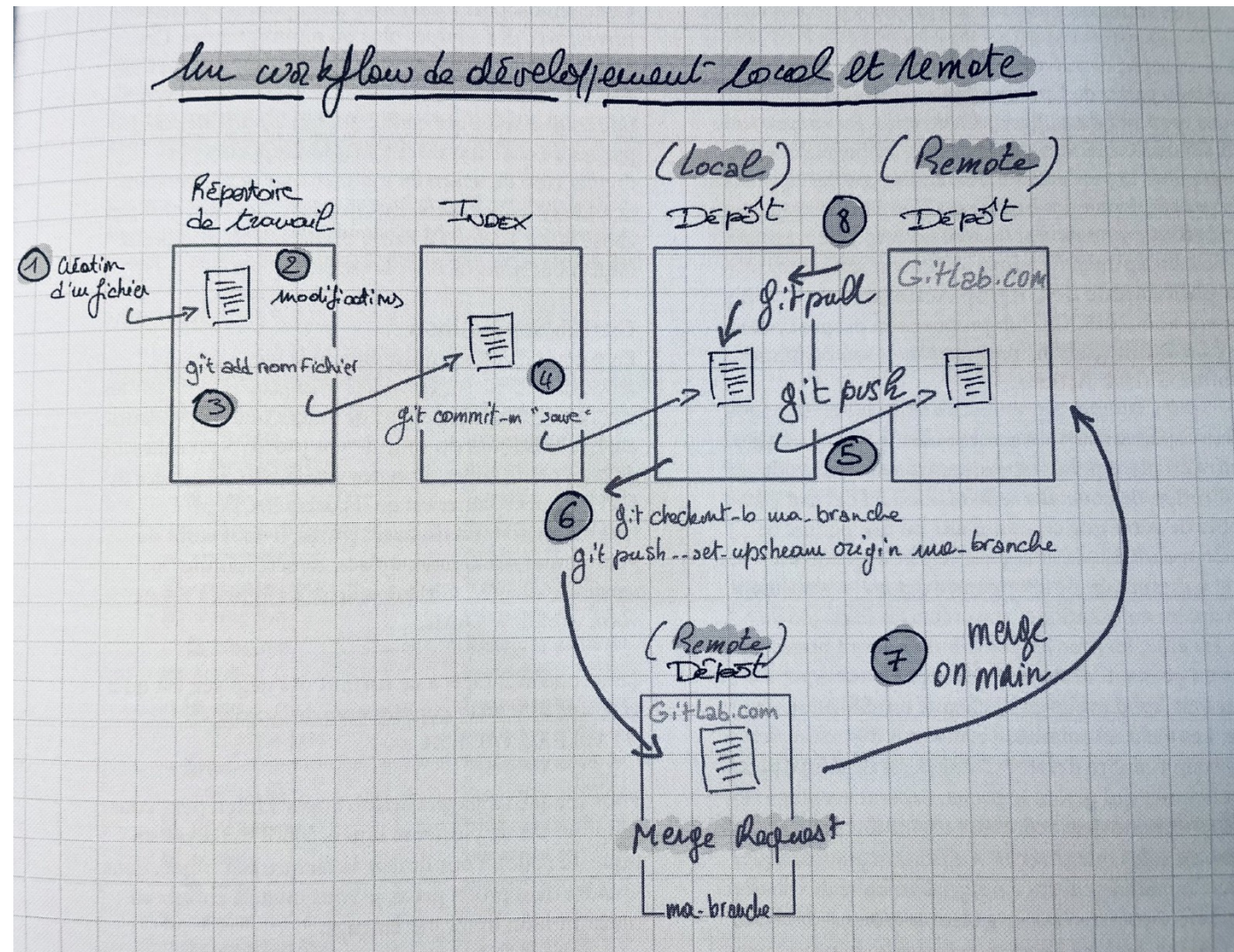


On envoie la branche avec le push

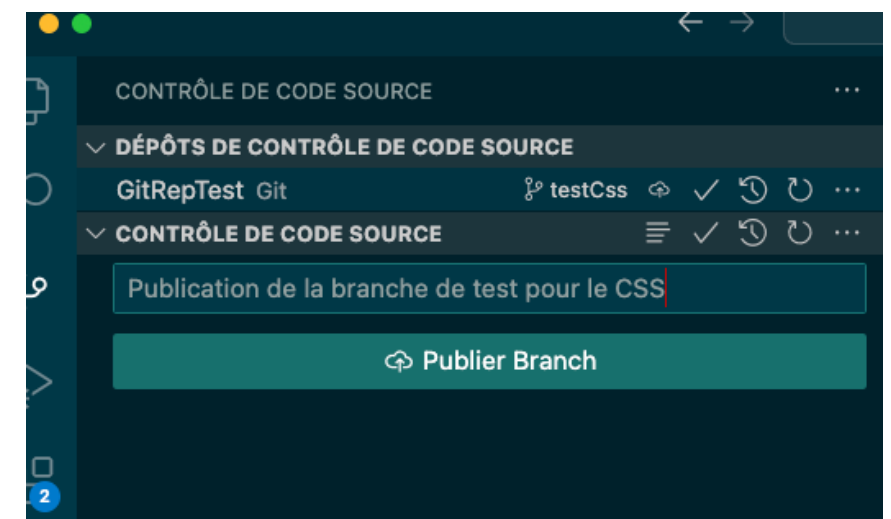
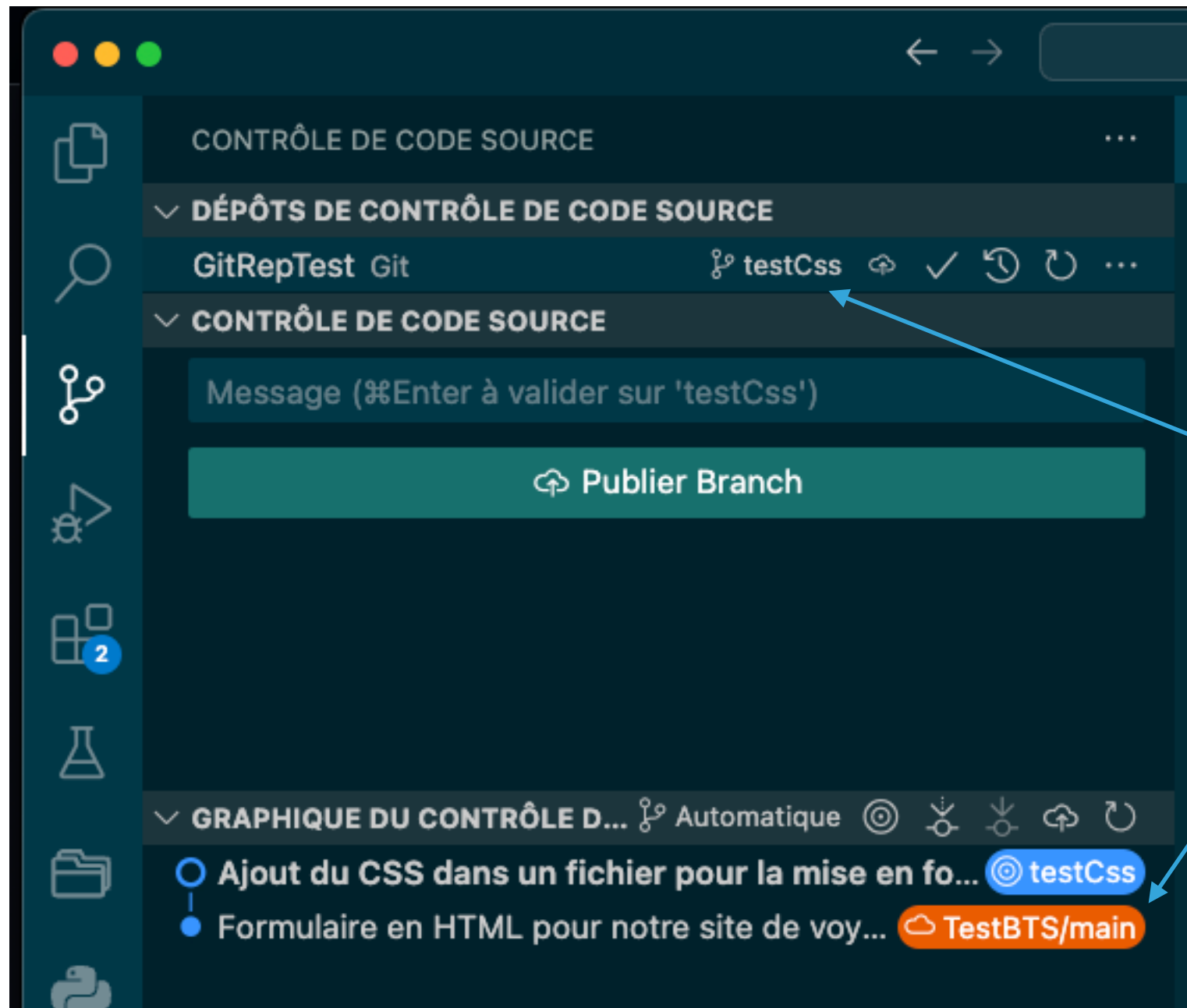
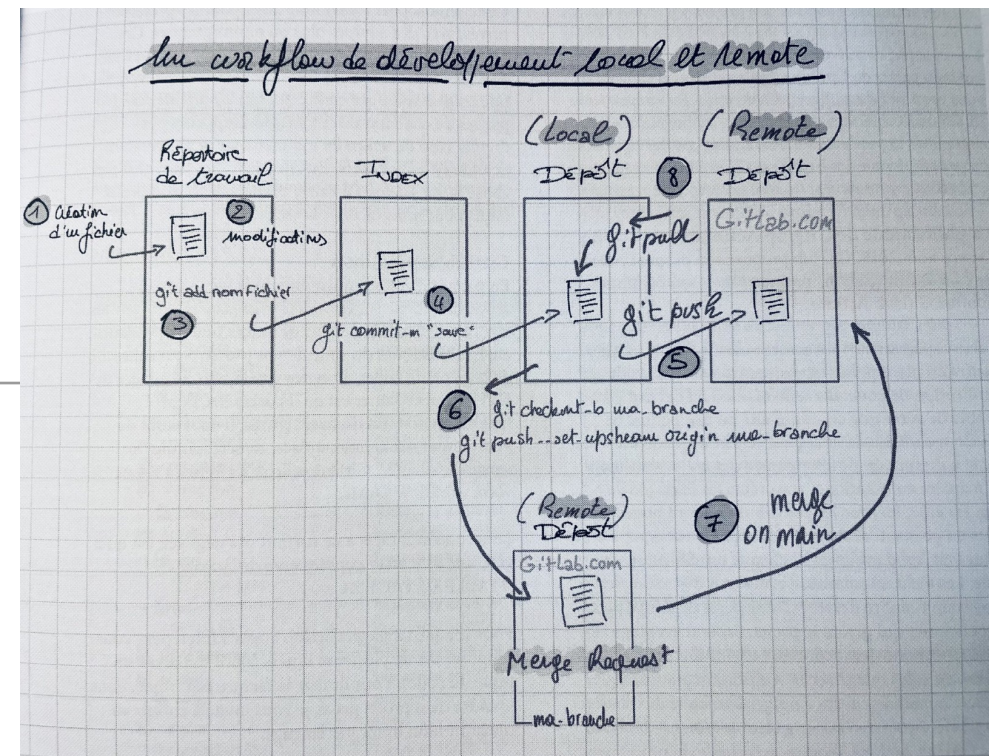
Vocabulaire spécifique à GIT

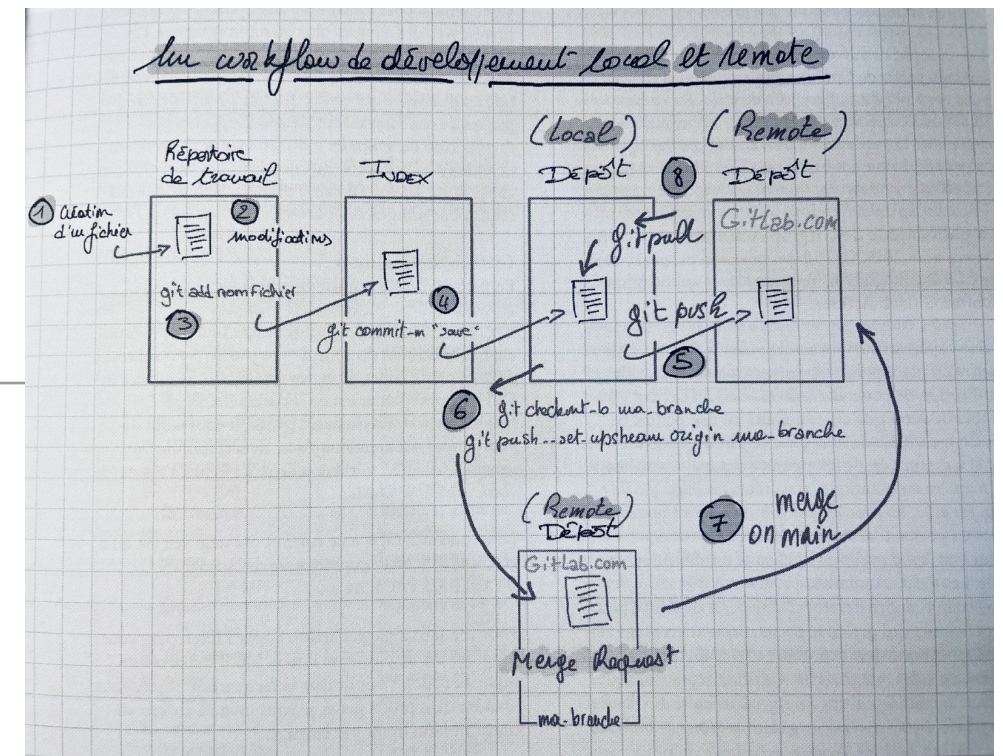
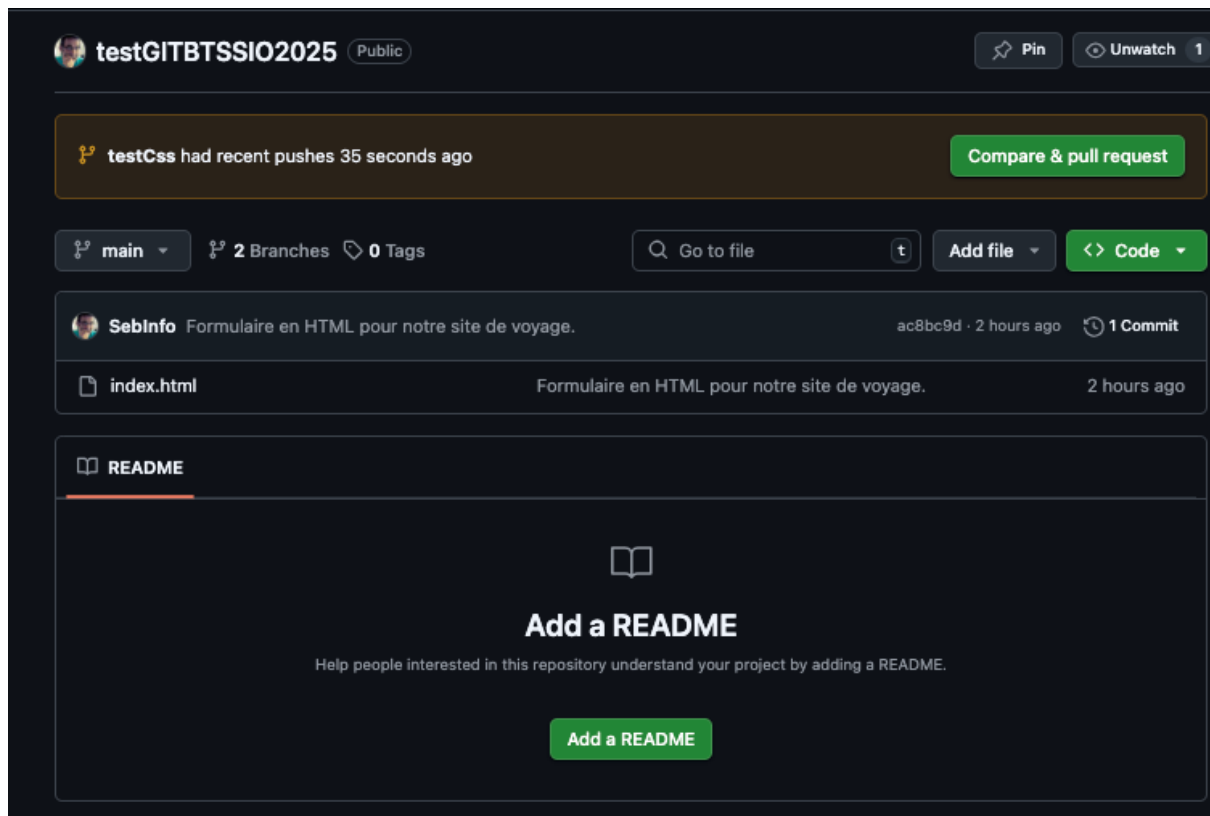


On va maintenant envoyer la branche testCss



On va repartir de l'étape 5 avec notre branche **testCss** que nous allons pusher aussi !

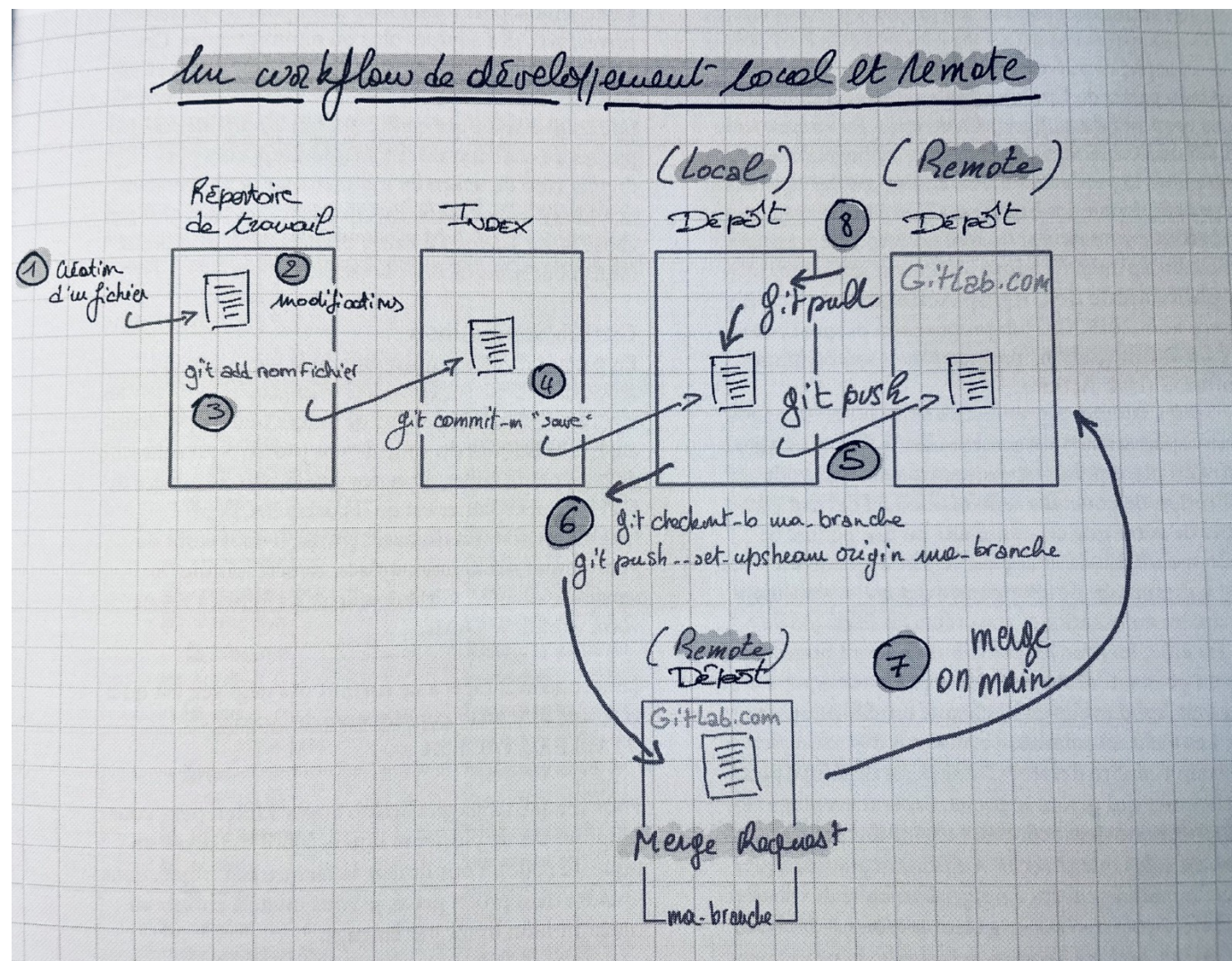




Les **Pull Requests (PR)** et **Merge Requests (MR)** sont deux concepts **quasiment identiques** utilisés dans Git pour **demande la fusion d'une branche dans une autre**, généralement de **feature-branch** → **main**.

- ◆ **Pull Request (PR)** → Terme utilisé sur **GitHub** et **Bitbucket**.
- ◆ **Merge Request (MR)** → Terme utilisé sur **GitLab**.

On passe à l'étape 6 !

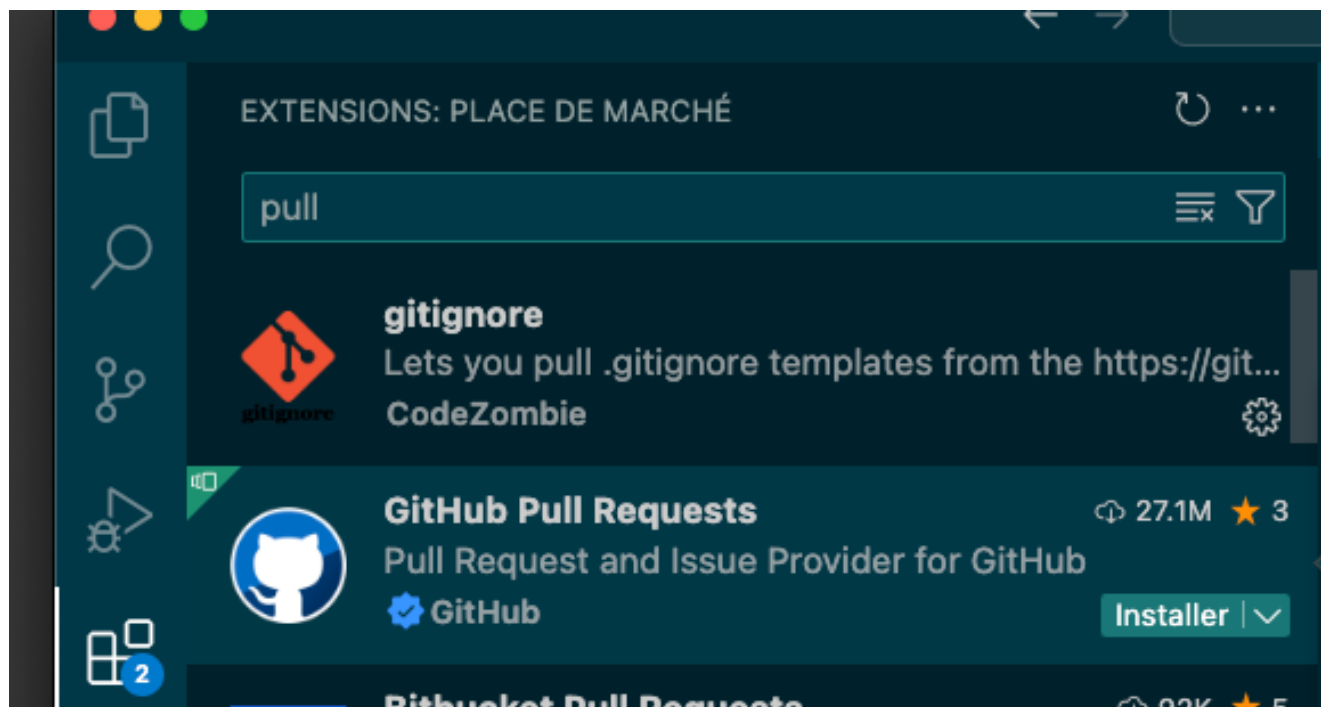


L'étape suivante avec le Merge Request ou Pull Request puisqu'on est sur Github !

On passe à l'étape 6 !

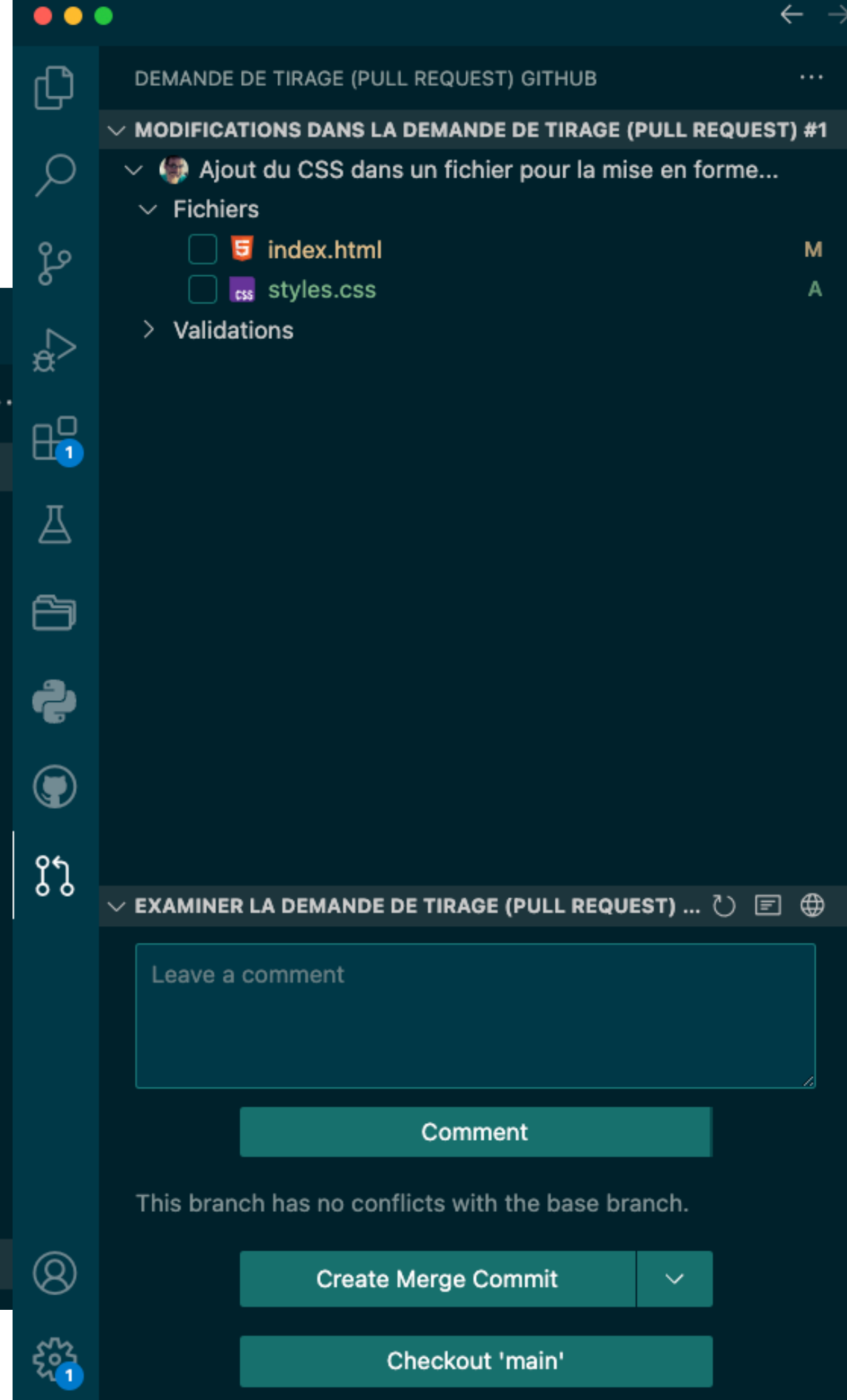
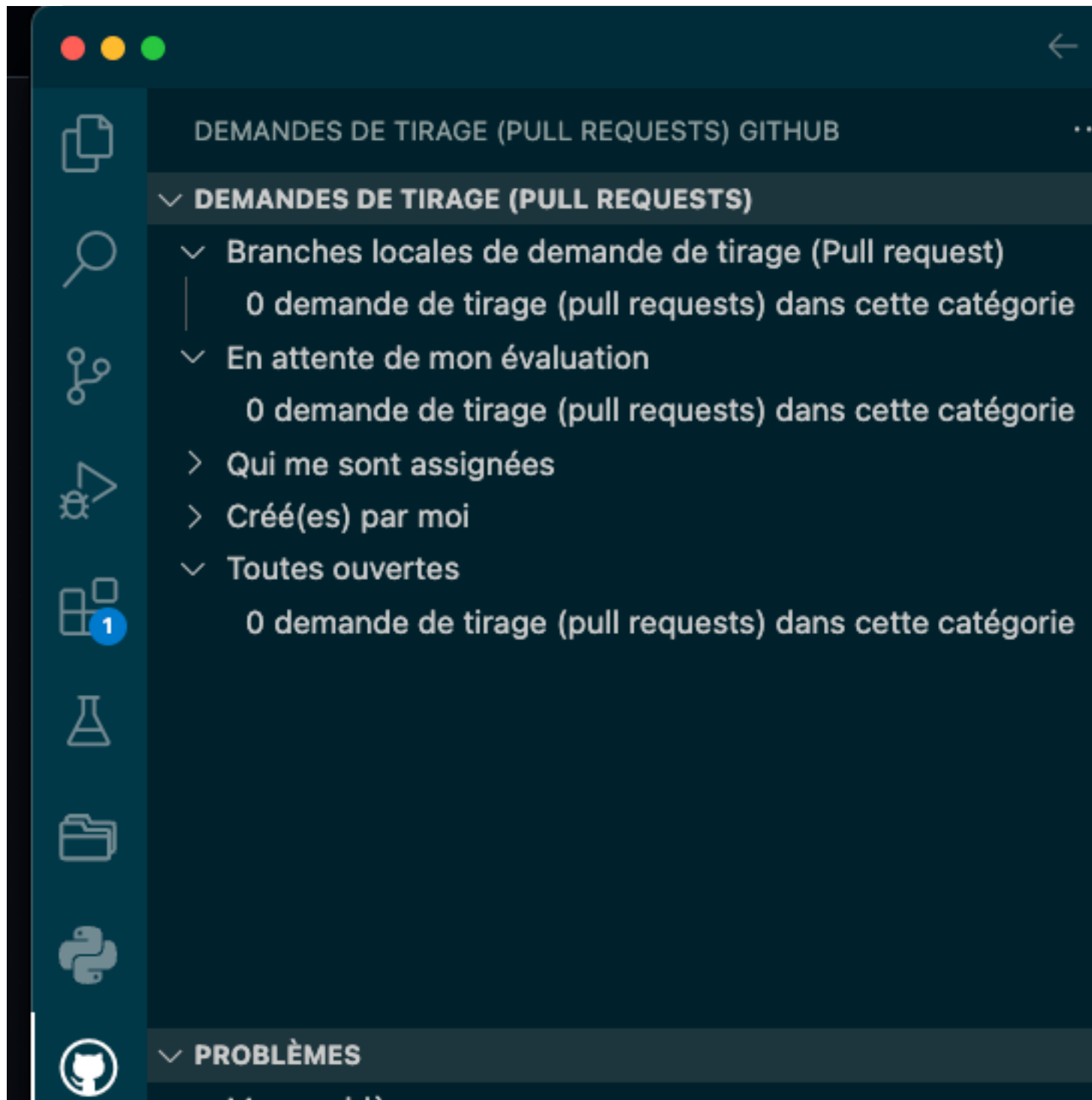
L'étape 6 : On peut le faire depuis Github. Mais si on veut le faire depuis VSC c'est possible !

Il faut installer l'extension "GitHub Pull Requests and Issues"

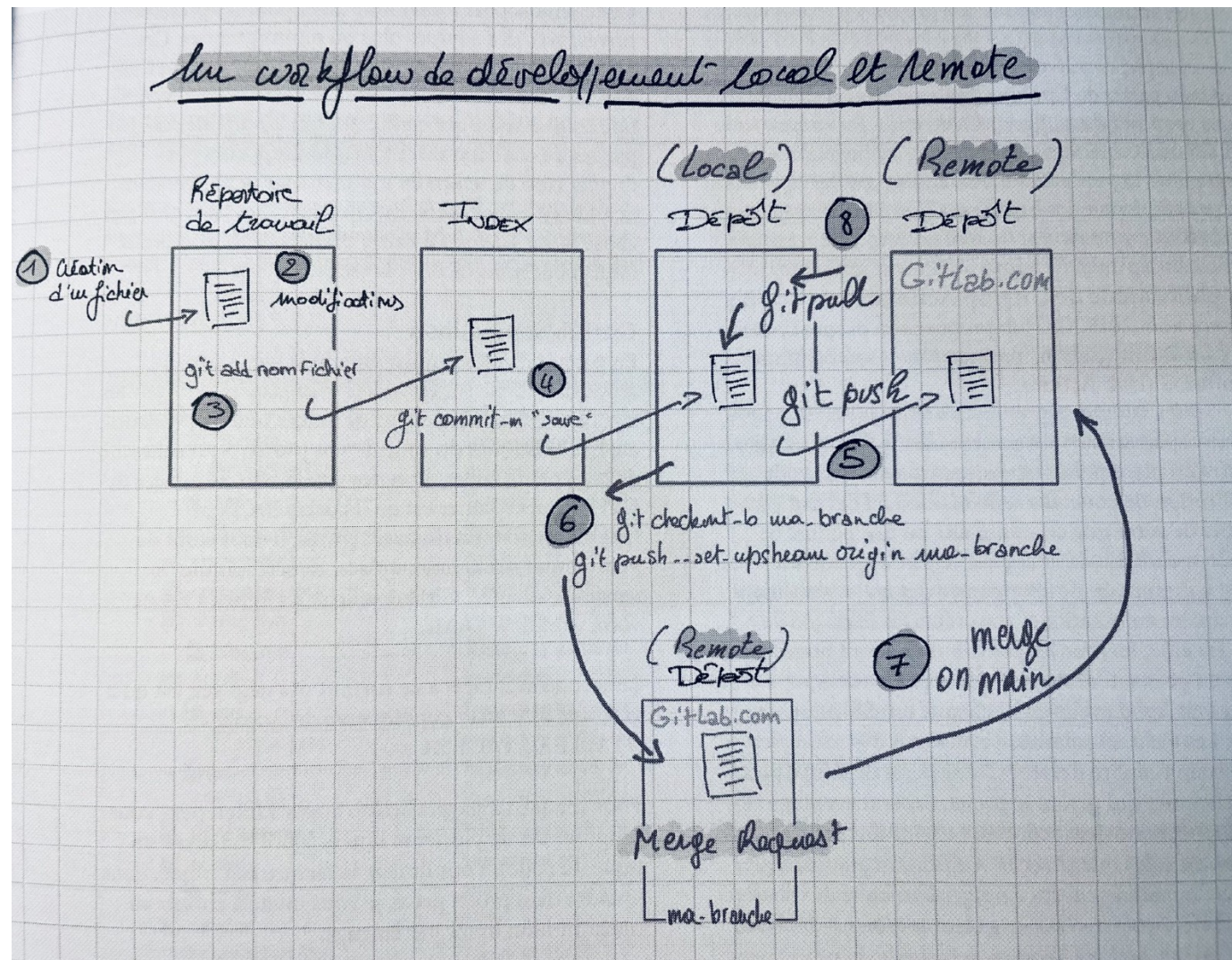


J'ai dû renommer mon host pour la compatibilité
git remote rename TestBTS origin

On passe à l'étape 6 !



On passe à l'étape 7 !



On passe à l'étape 6 !

The screenshot shows the Visual Studio Code interface with a pull request merged. The left sidebar displays the 'DEMANDE DE TIRAGE (PULL REQUEST) GITHUB' section, showing the title 'Ajout du CSS dans un fichier pour la mise en forme...', the files 'index.html' and 'styles.css', and a 'Validations' section. The main editor area shows the 'Ajout du CSS dans un' title and a 'Merged' status. The right sidebar shows the 'CONSOLE DE DÉBOGAGE' (Debug Console) with a filter input and a message from 'SebInfo' stating 'J'aimerais intégrer le CSS pour ce formulaire'. The bottom status bar indicates 'Checkout 'main' without deleting branch'.

DEMANDE DE TIRAGE (PULL REQUEST) GITHUB

MODIFICATIONS DANS LA DEMANDE DE TIRAGE (PULL REQUEST) #1

Ajout du CSS dans un fichier pour la mise en forme...

Fichiers

- index.html
- styles.css

Validations

EXAMINER LA DEMANDE DE TIRAGE (PULL REQUEST) ...

Leave a comment

Comment

Pull request successfully merged.

Create New Pull Request...

Delete branch...

Checkout 'main' without deleting branch

index.html

Pull Request #1

Ajout du CSS dans un

Merged

SebInfo merged changes

Checked Out

Checkout 'main'

Refresh

SebInfo commented 4 minutes ago

J'aimerais intégrer le CSS pour ce formulaire

PROBLÈMES

SORTIE

TERMINAL

PORTS

CONSOLE DE DÉBOGAGE

Filtre (exemple : text, !exclude, \escape)

On a bien une fusion puisque ici main contient le css

The screenshot shows a GitHub repository interface. At the top, the repository name is 'testGITBTSSIO2025' with a 'Public' badge. Navigation buttons include 'Pin', 'Unwatch', and a dropdown with '1'. Below this, a branch selector shows 'main' with a dropdown arrow, followed by '2 Branches' and '0 Tags'. A search bar 'Go to file' and buttons 'Add file' and 'Code' are present. The main content area displays a merge pull request by 'SebInfo' from 'SebInfo/testCss' with commit hash 'b81bd9d' and '3 Commits' made '1 minute ago'. Below the merge, two file changes are listed: 'index.html' and 'styles.css', both with the description 'Ajout du CSS dans un fichier pour la mise en forme du for...' and a timestamp of '2 hours ago'. At the bottom, a 'README' section is highlighted with an orange underline, featuring an 'Add a README' button and a message: 'Help people interested in this repository understand your project by adding a README.'

testGITBTSSIO2025 Public

main 2 Branches 0 Tags

Go to file Add file Code

SebInfo Merge pull request #1 from SebInfo/testCss b81bd9d · 1 minute ago 3 Commits

index.html Ajout du CSS dans un fichier pour la mise en forme du for... 2 hours ago

styles.css Ajout du CSS dans un fichier pour la mise en forme du for... 2 hours ago

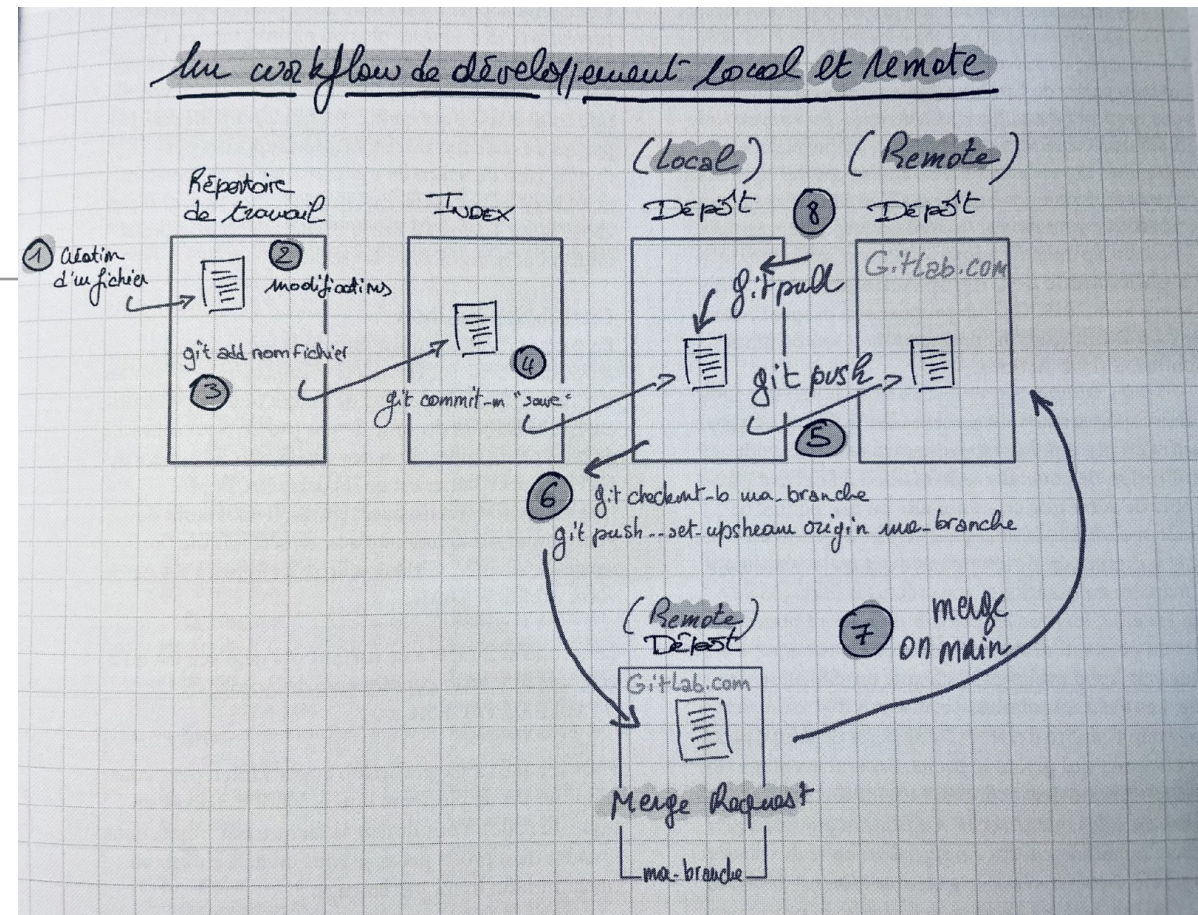
README

Add a README

Help people interested in this repository understand your project by adding a README.

Add a README

On passe à l'étape 8

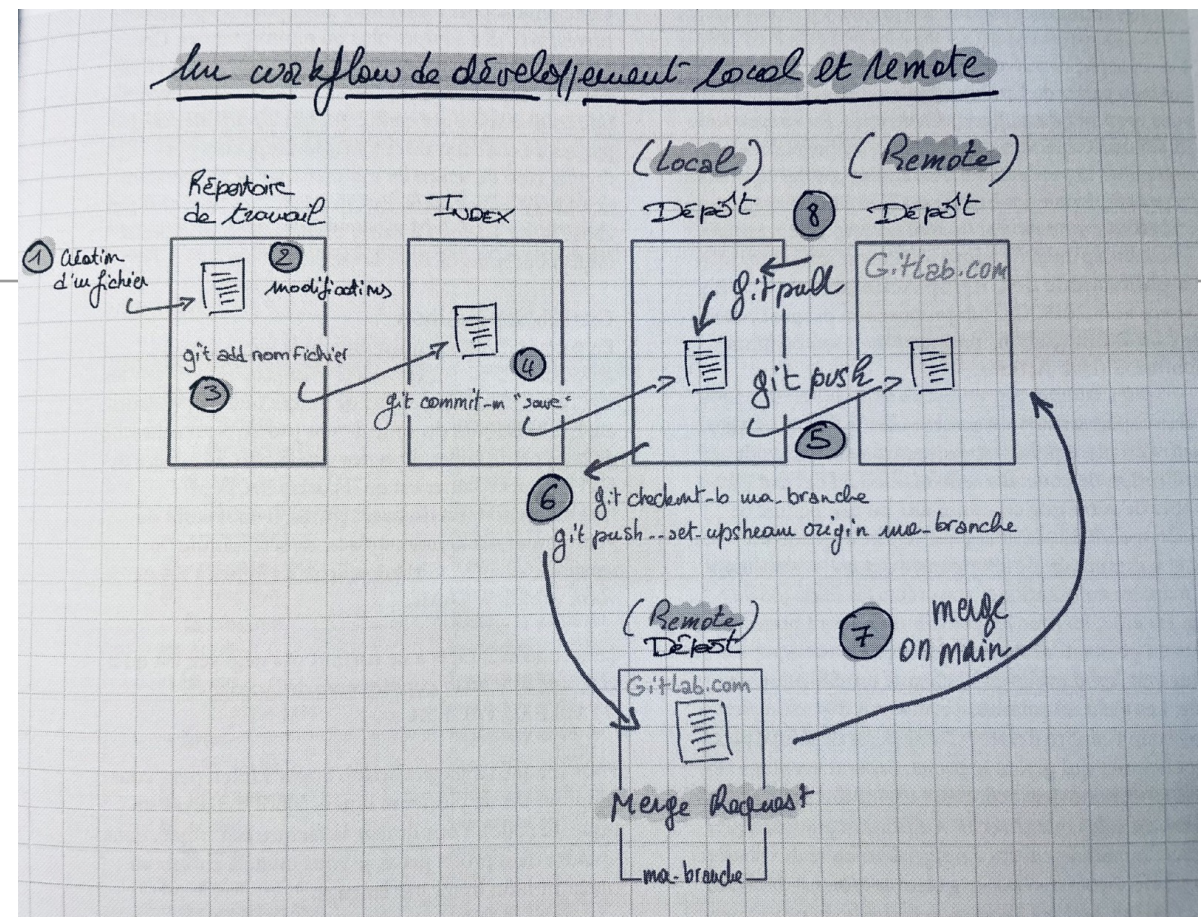


Etape 8 : Permet de récupérer les dernières mises à jour du dépôt distant (`main`) vers ton dépôt local. Important cette étape 8 car sinon le main local et distant sera pas le même.

Pourquoi `git pull` est important ?

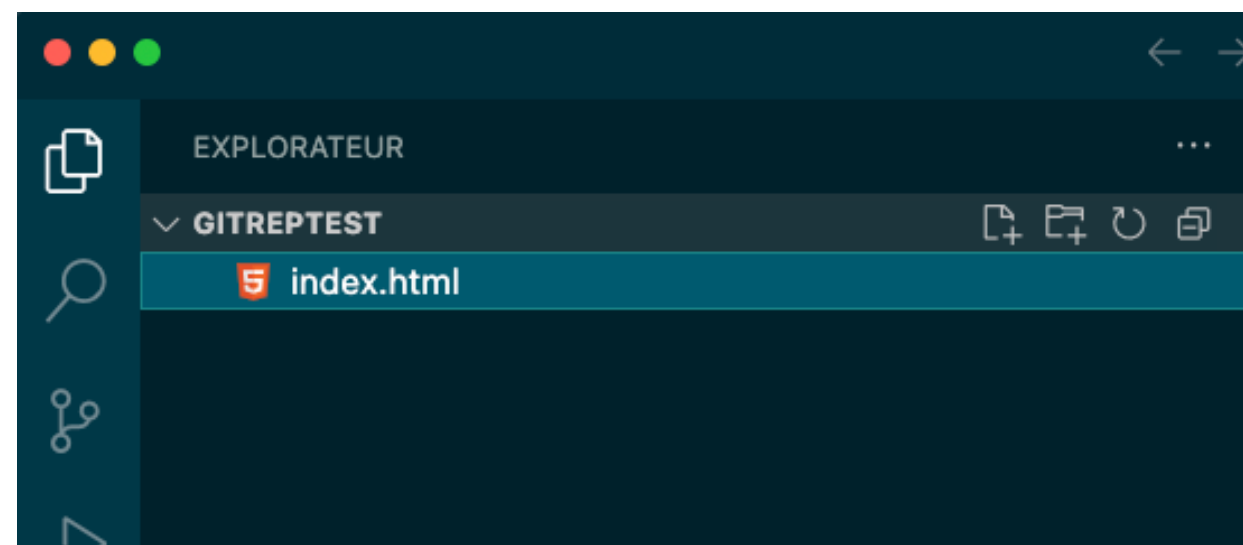
- ✓ Synchronise ton code local avec GitHub.
- ✓ Évite les conflits en s'assurant que tu as bien la dernière version de `main`.
- ✓ Permet de travailler sur du code à jour avant d'ajouter de nouvelles modifications.

On passe à l'étape 8



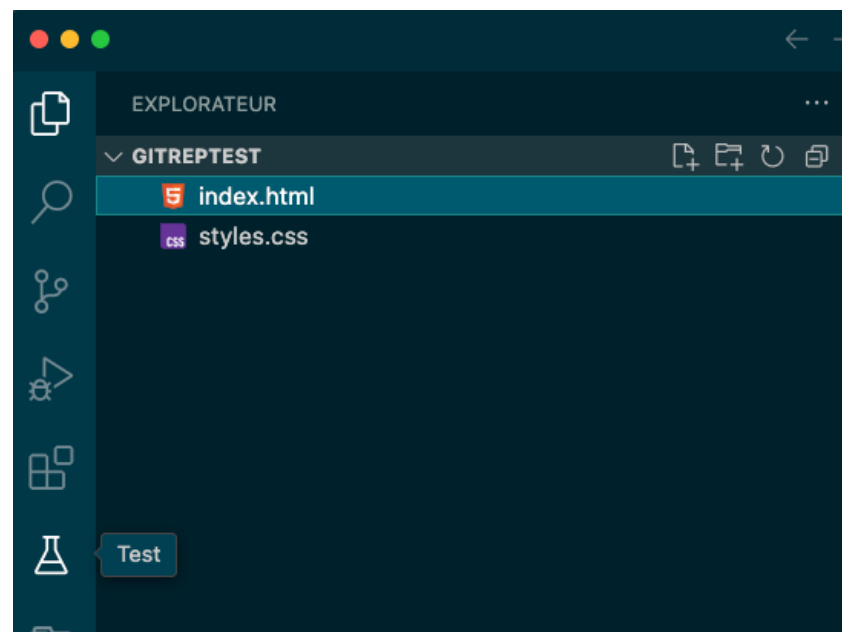
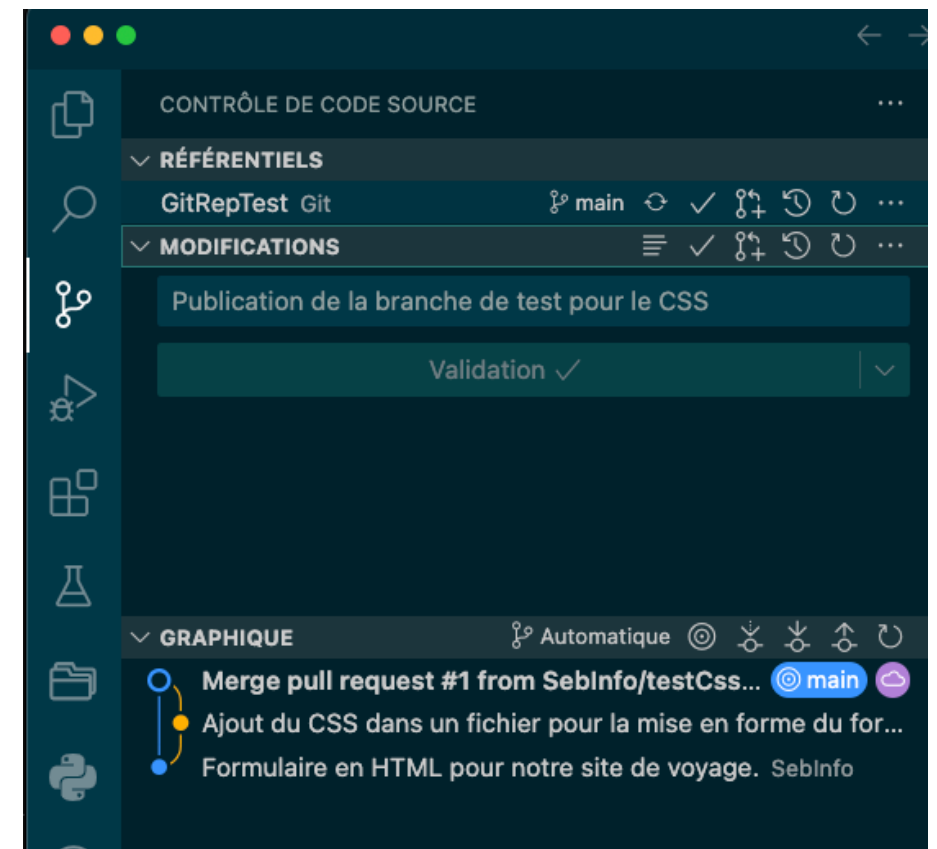
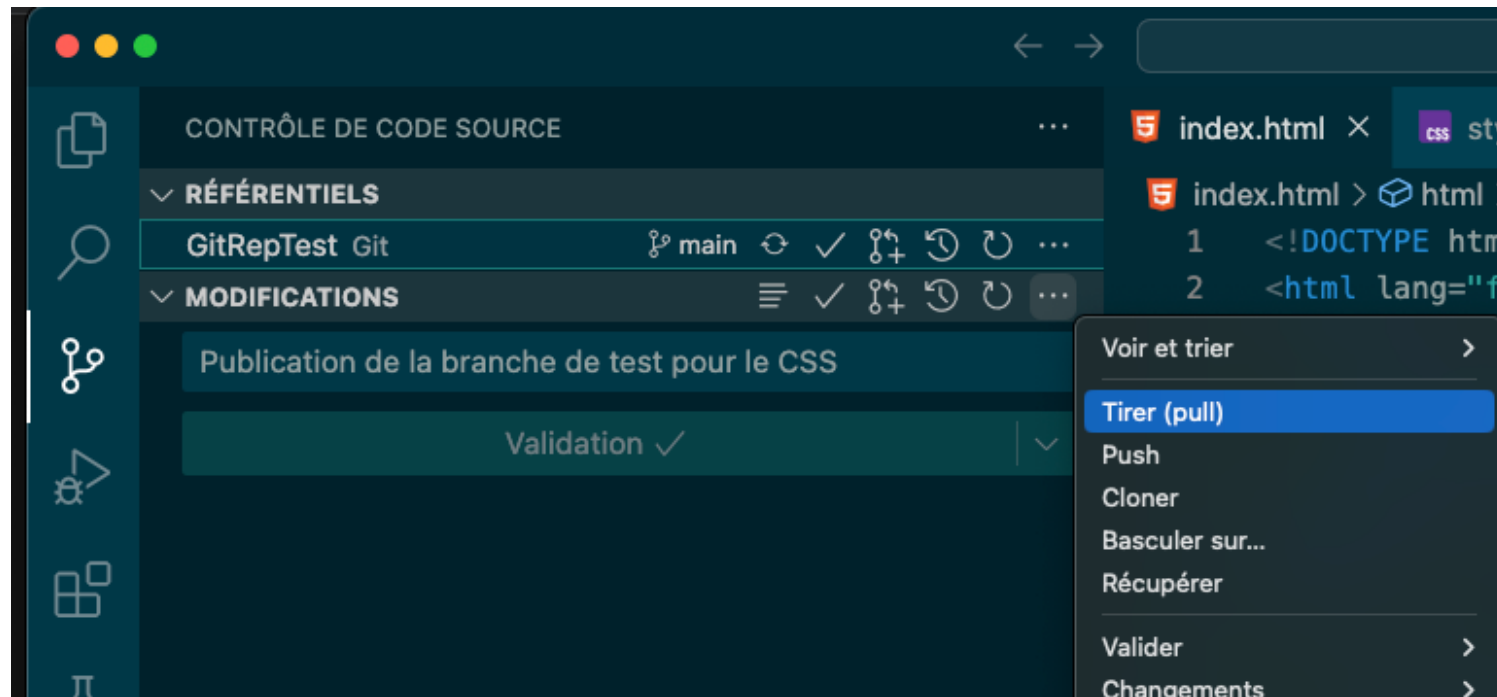
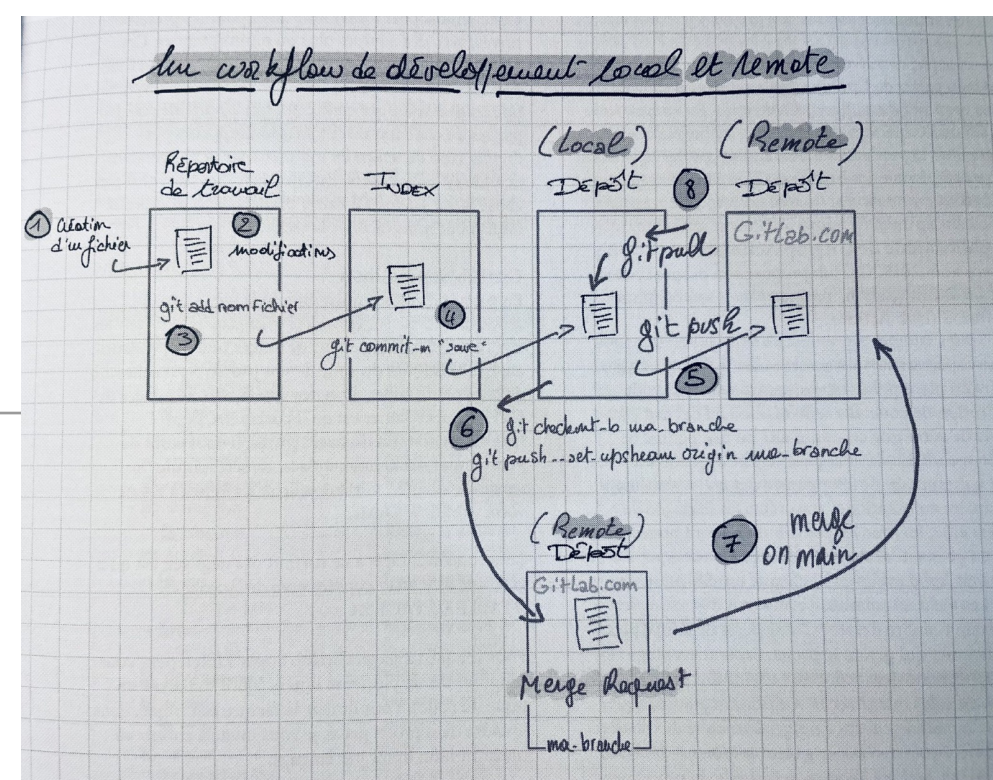
Sur VSC placer vous sur la branche main

Pas encore le CSS



On passe à l'étape 8

Puis on fait le git pull



Git et les bonnes pratiques

- Commiter des modifications en relation les unes des autres
-> deux aspects = deux commit
- Tester le code avant de comiter
- Utiliser les branches
- Commiter souvent
- Ecrire de bonne description dans les commit
- Ne pas commiter un travail à moitié fait
- Git n'est pas un système de Sauvegarde/Backup



**MERCI POUR
VOTRE ATTENTION**