Prise en main de *VsCode*, compilation/exécution d'un programme C

Sébastien Jean

IUT de Valence
Département Informatique

v1.0, 17 septembre 2025

IUT A Valence
Université Grenoble Alpes

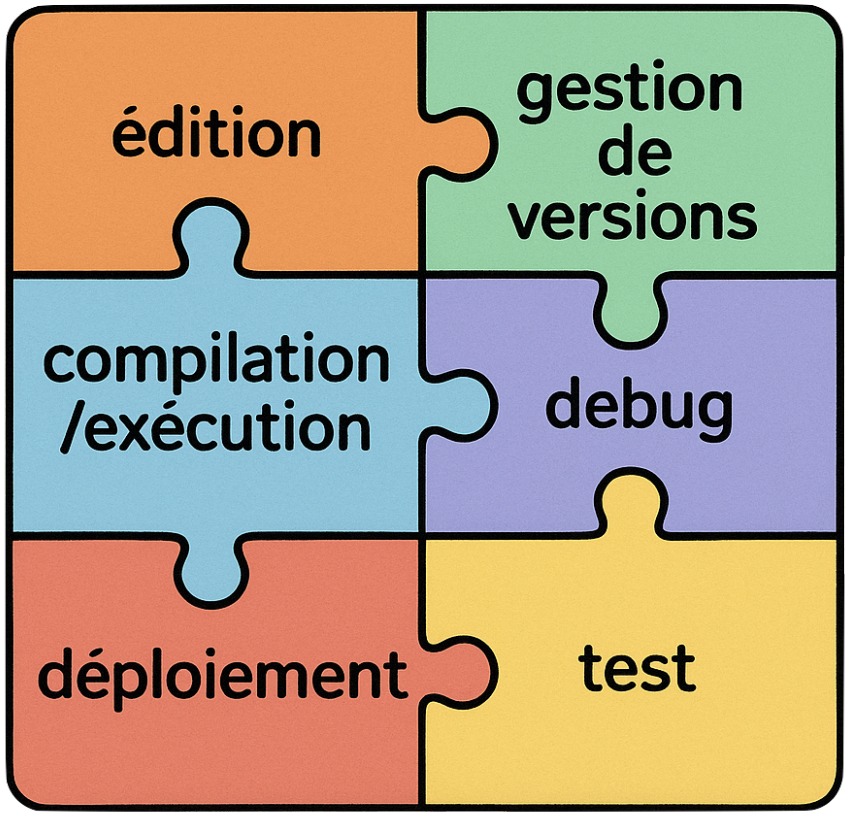
IDE

- Qu'est ce qu'un environnement de développement intégré (**IDE**) ?

IUT Valence
Université Grenoble Alpes

IDE

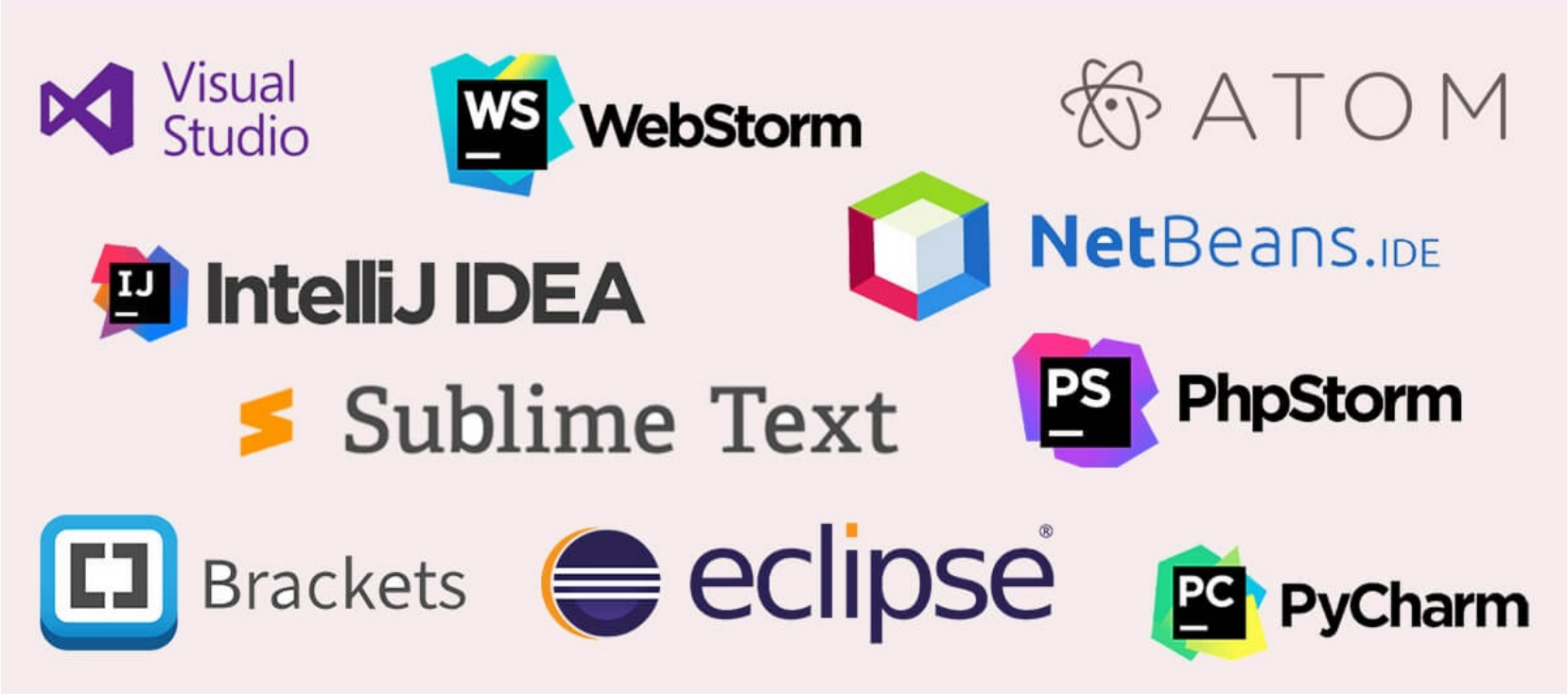
- Un IDE est un atelier qui rassemble les principaux outils permettant la production de logiciel



- Un IDE est configurable et extensible

IUT Valence
Université Grenoble Alpes

Quelques IDE ou simples



IUT Valence
Université Grenoble Alpes

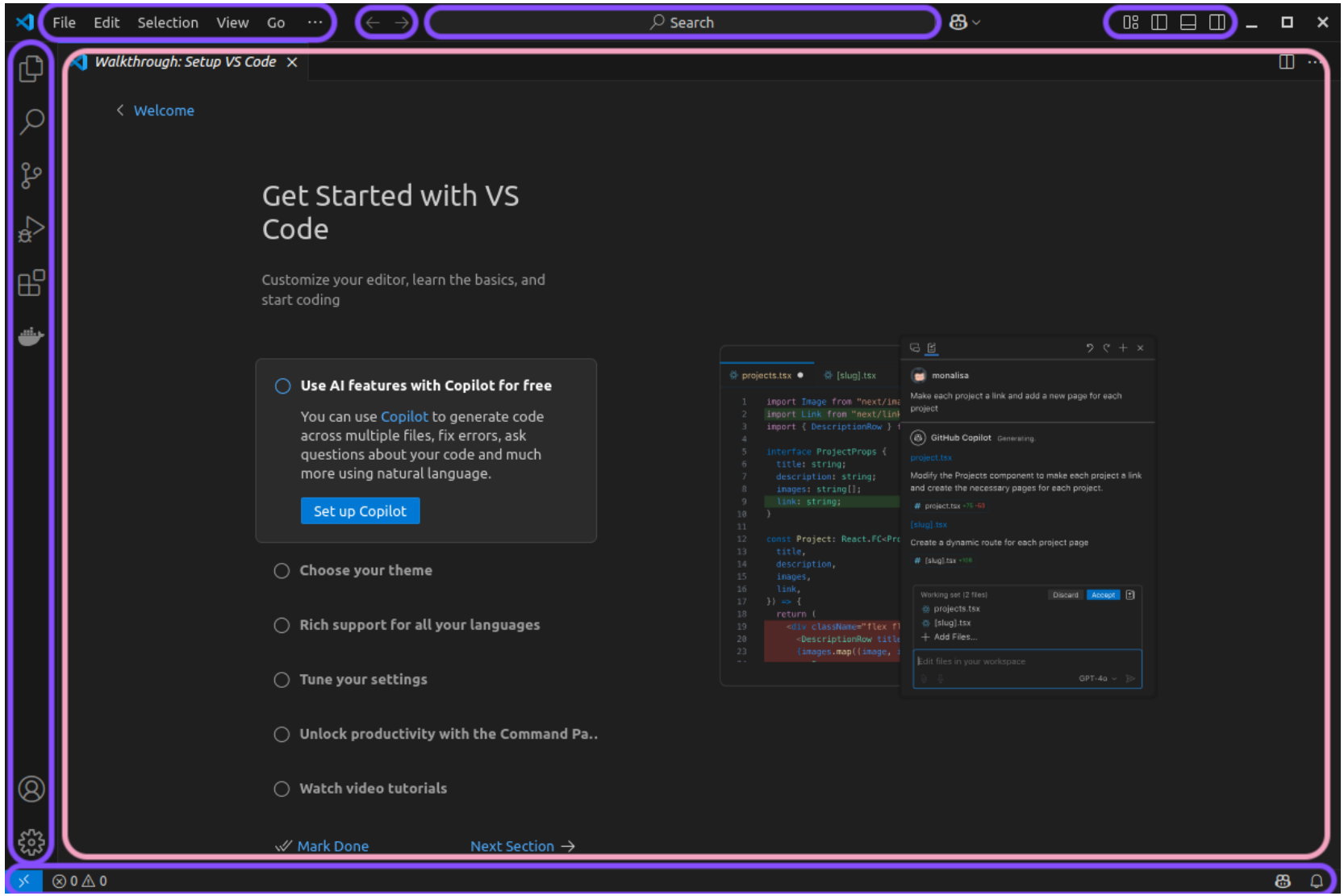
VsCode

- **Exécuter** *VsCode*

IUT Valence
Université Grenoble Alpes

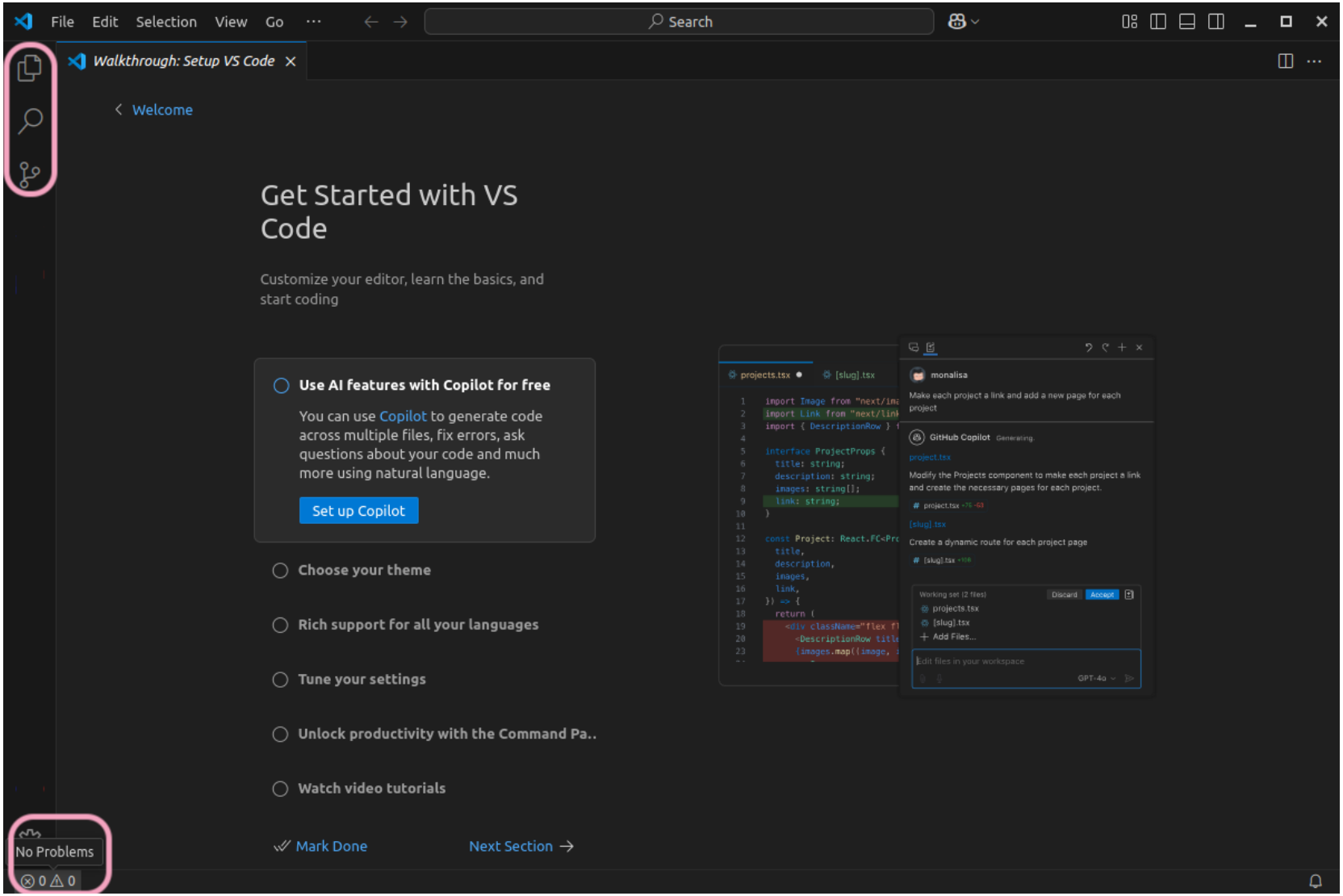
Exécuter *VsCode* pour la première fois

- Tour d'horizon des différentes zones de l'interface

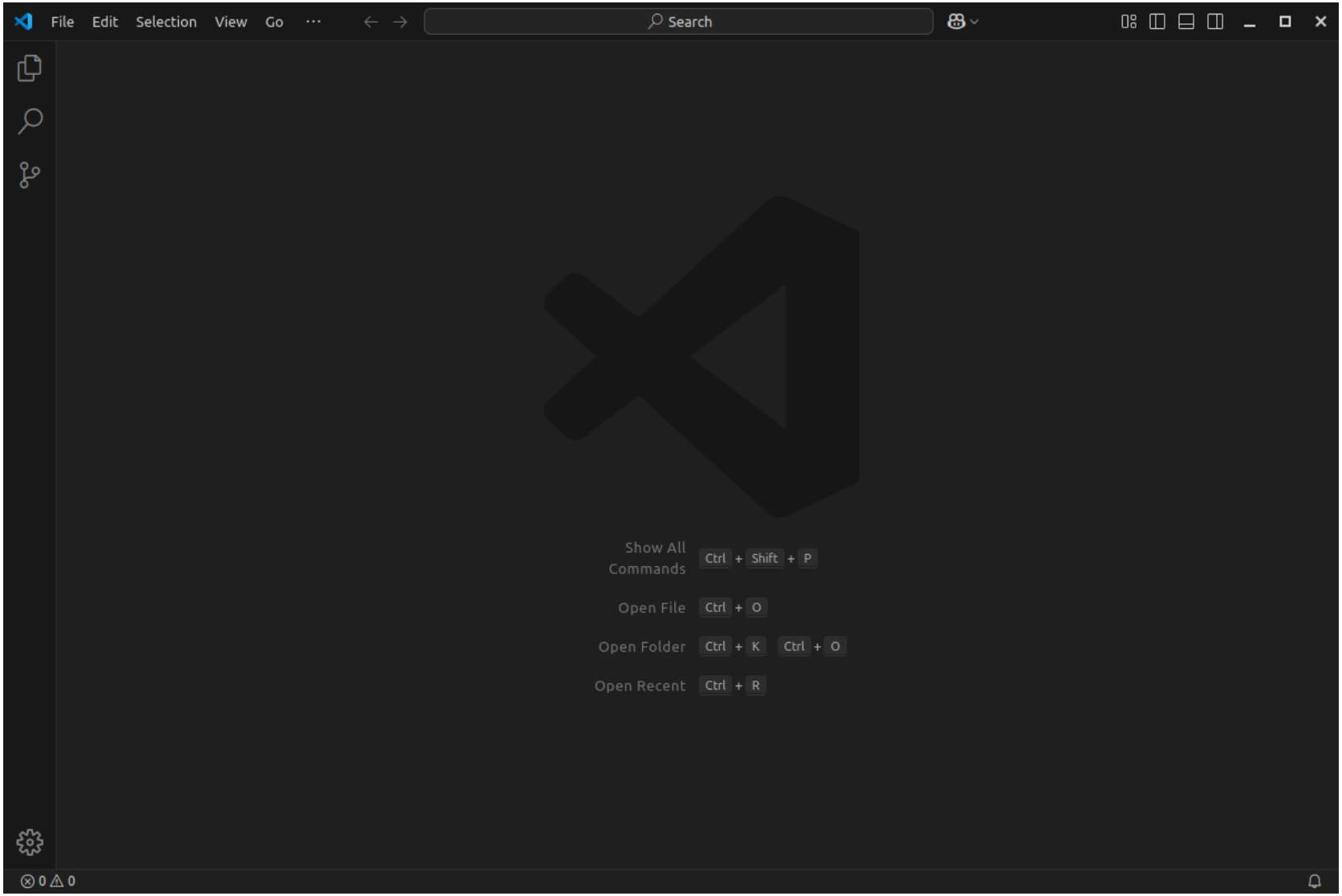


Simplifier le menu latéral

- Se débarrasser des fonctions *inutiles* pour le moment

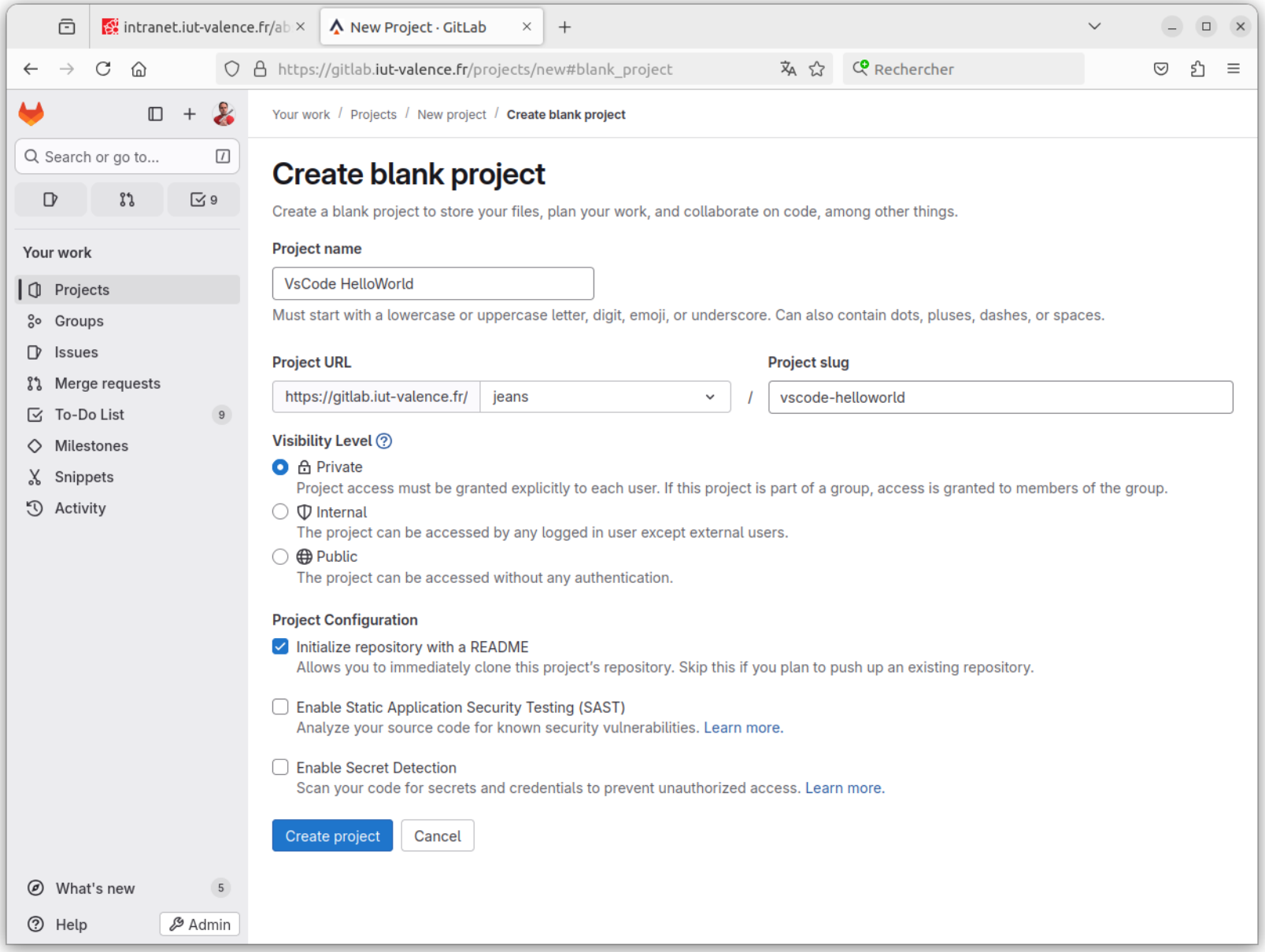


Etre prêt à coder

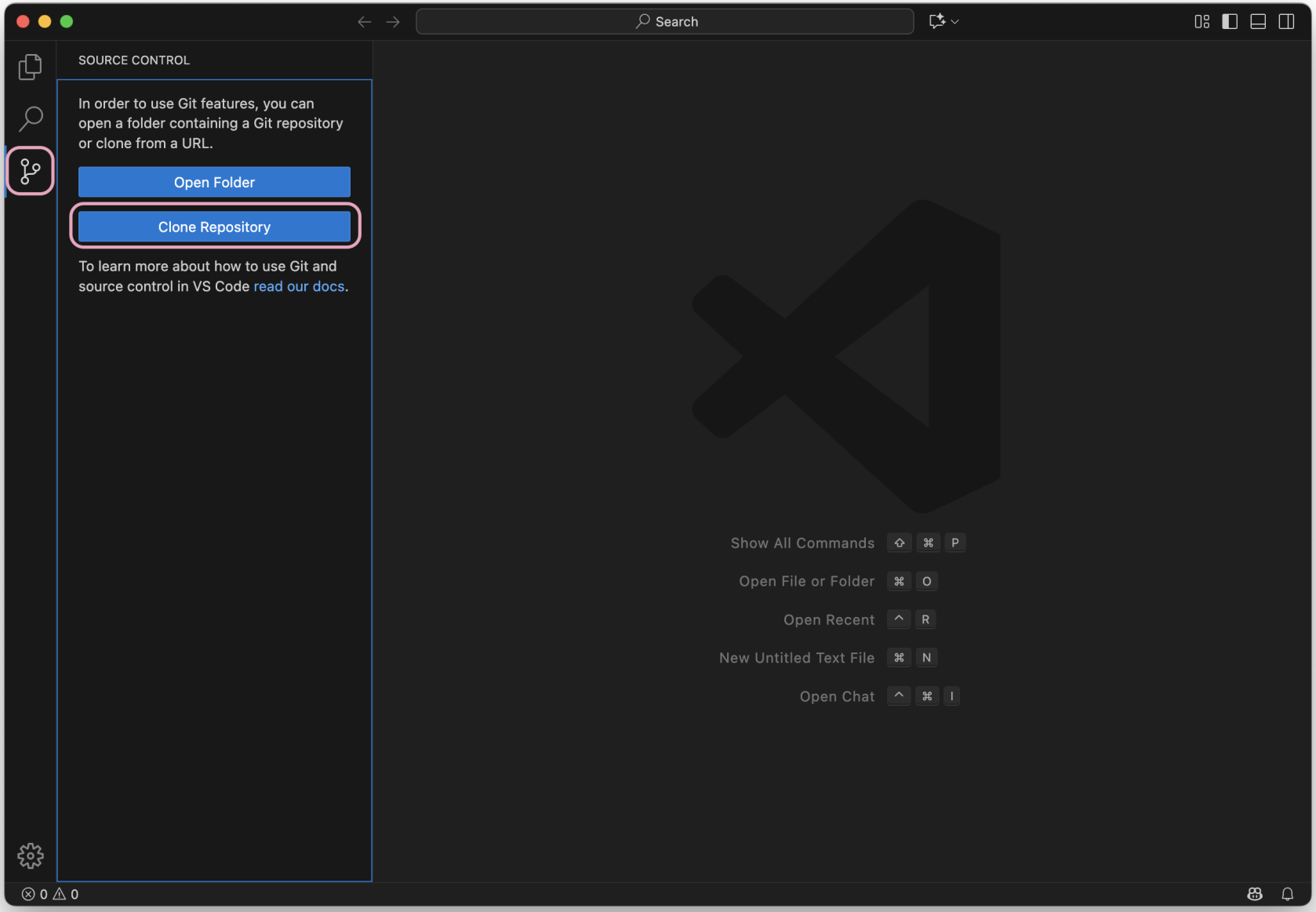


IUT Valence
Université Grenoble Alpes

Créer d'un dépôt gitlab

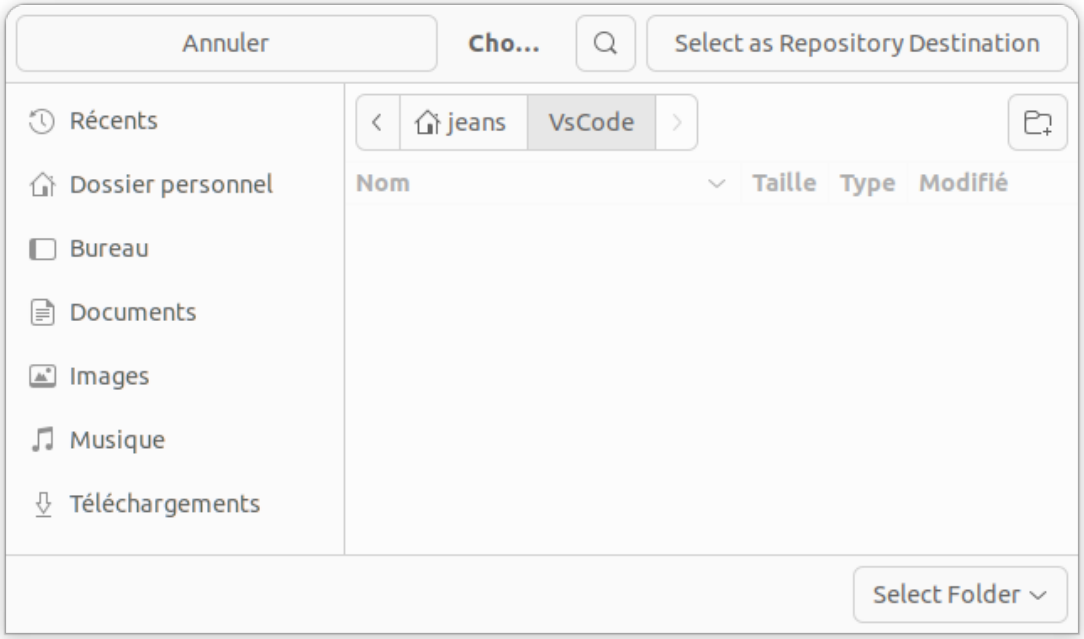


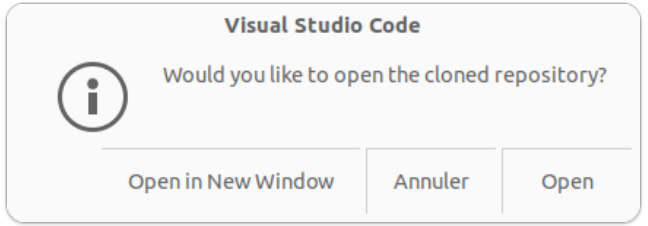
Cloner un dépôt *GitLab* depuis *VsCode*



Cloner un dépôt *GitLab* depuis *VsCode* (fin)

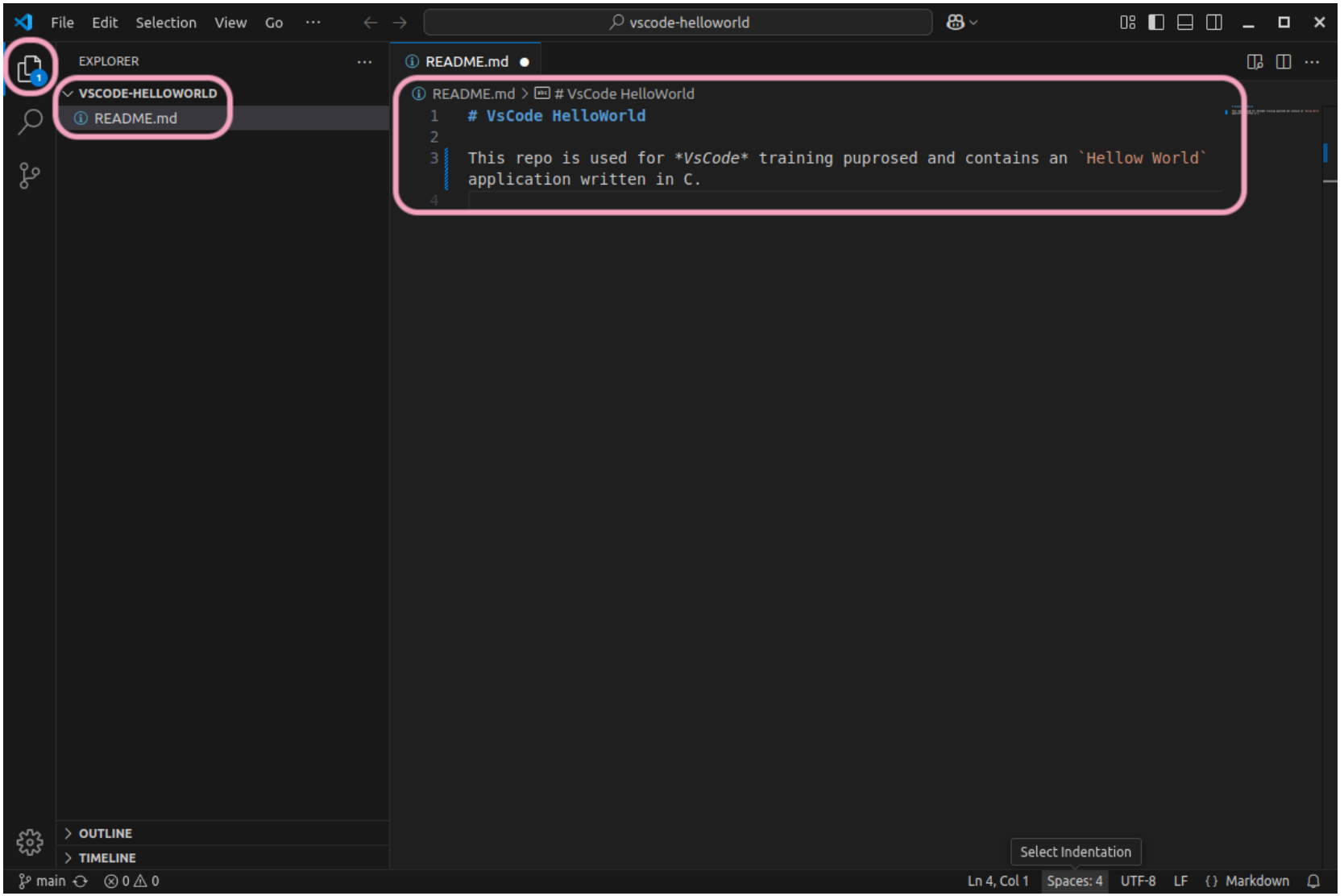
- Créer un répertoire d'accueil pour le clone
- Ouvrir le dossier dans *VsCode*





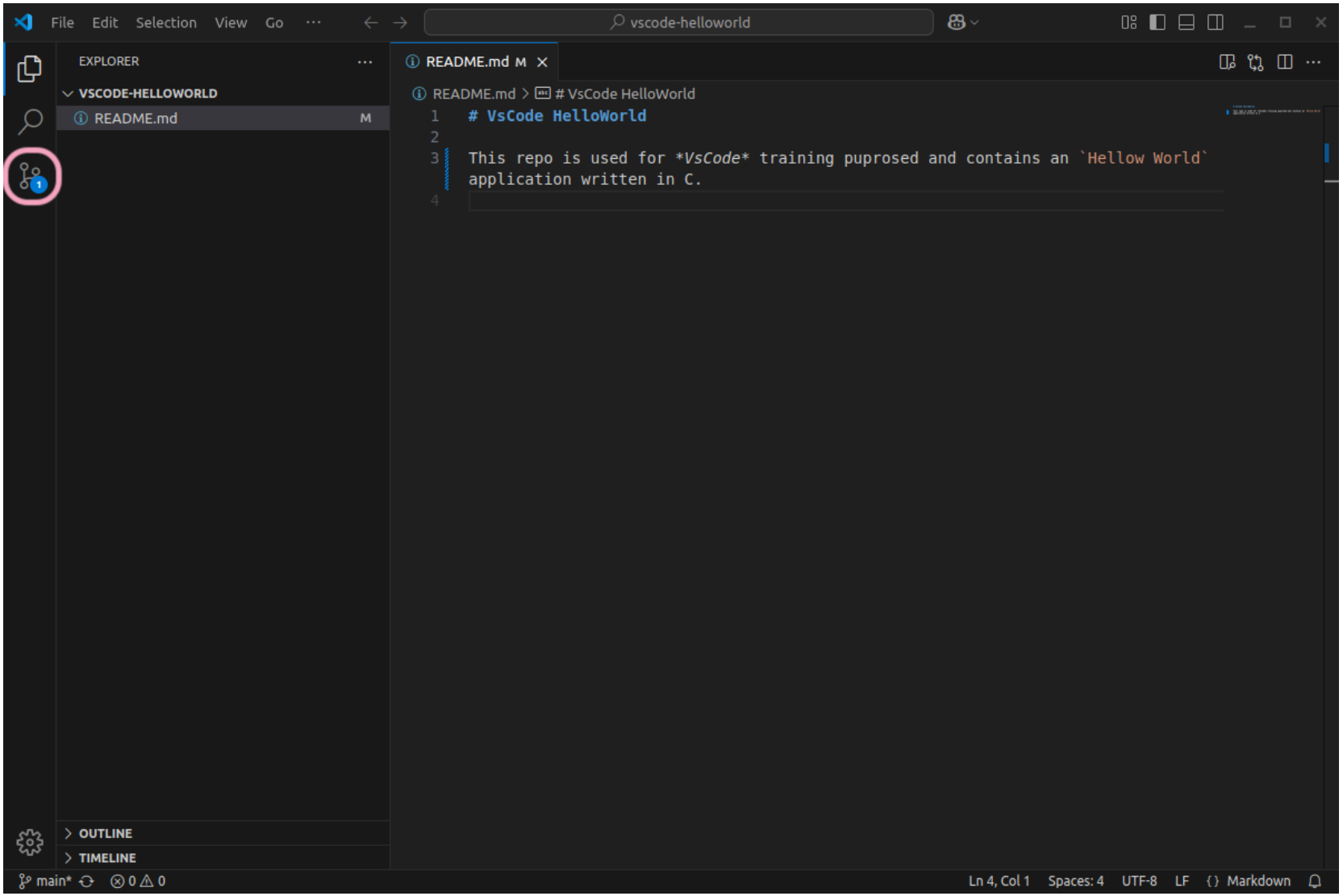
Editer (sans sauvegarder) le fichier README

- N.B. : le fichier n'est pas automatiquement sauvegardé



Sauvegarder le fichier `README`

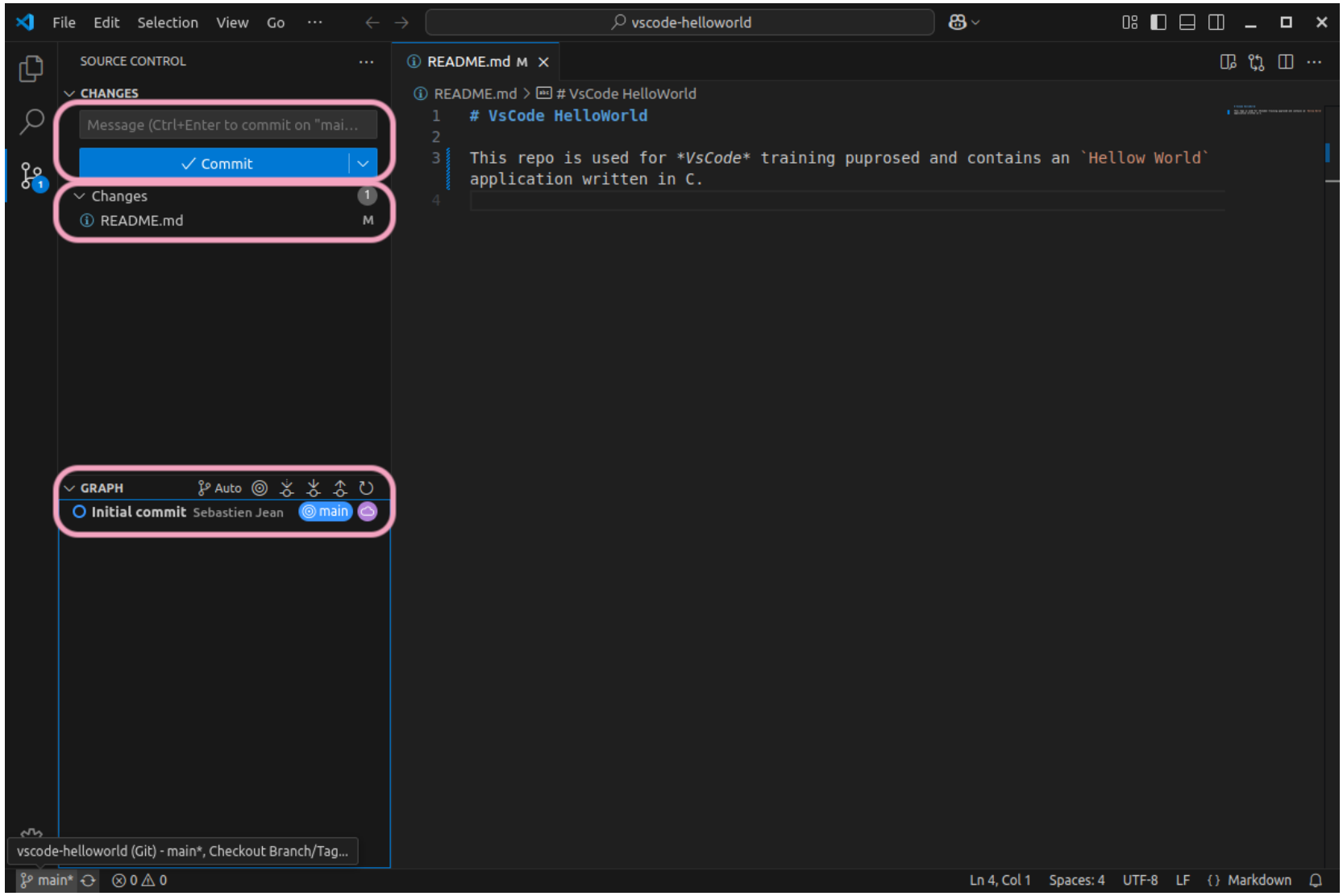
- L'apparition du fichier est observée par *git*



IUT Valence
Université Grenoble Alpes

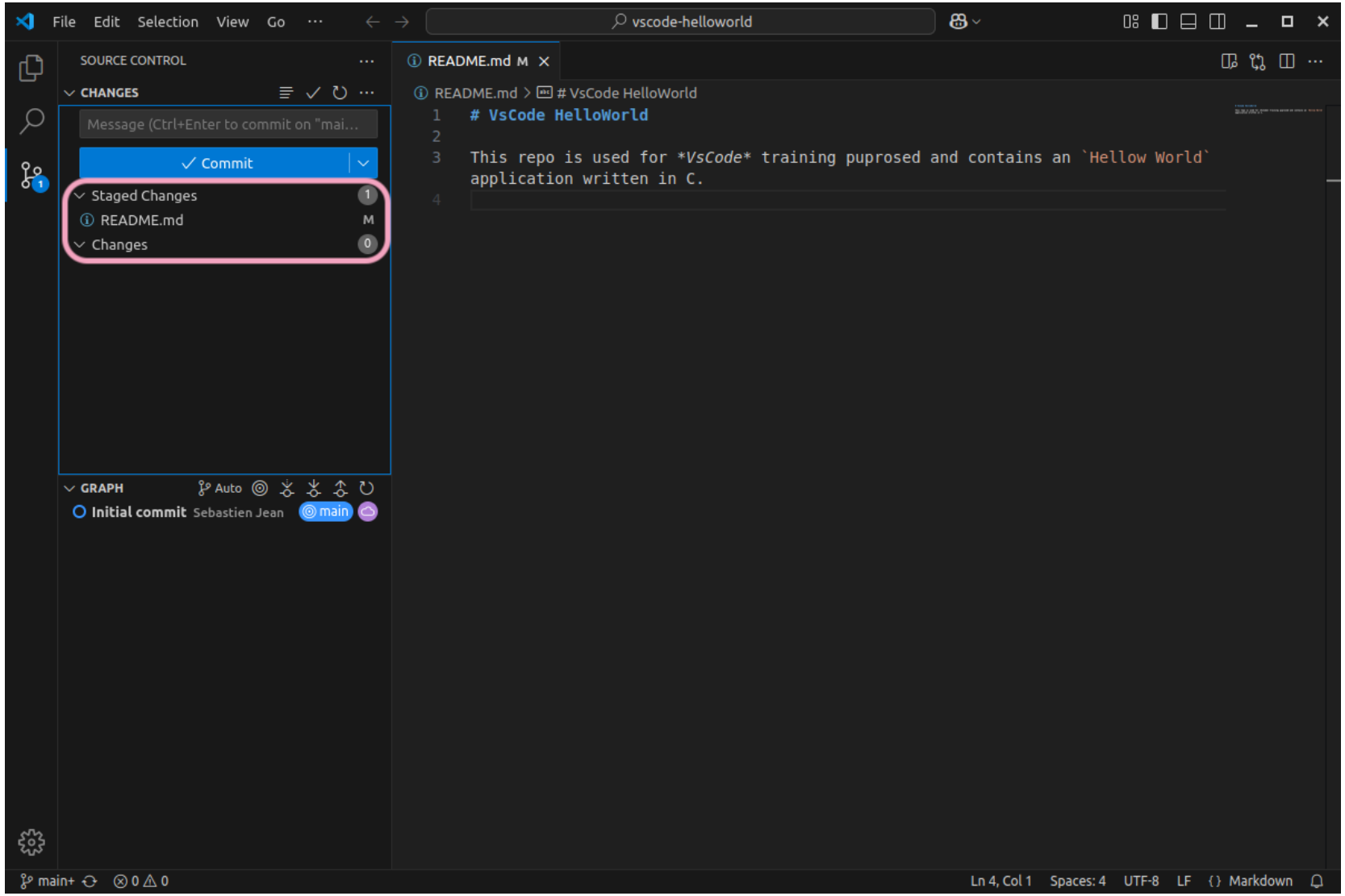
Vue "gestion de versions"

- Tour d'horizon (et liens avec les commandes en ligne)



IUT Valence
Université Grenoble Alpes

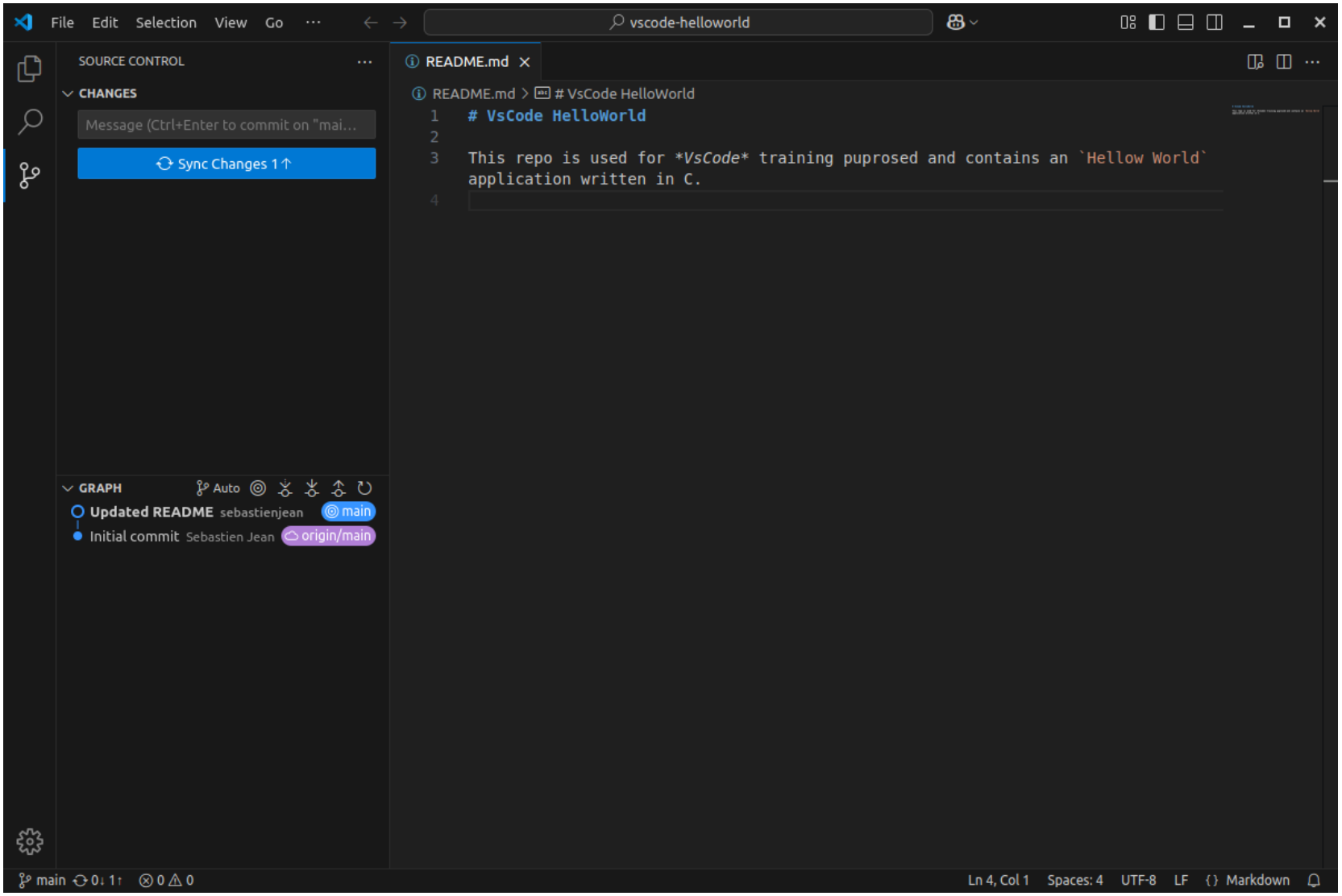
Préparer la prochaine version (*unstaged* → *staged*)



- Quelle est la commande *git* correspondante ?

IUT Valence Université Grenoble Alpes

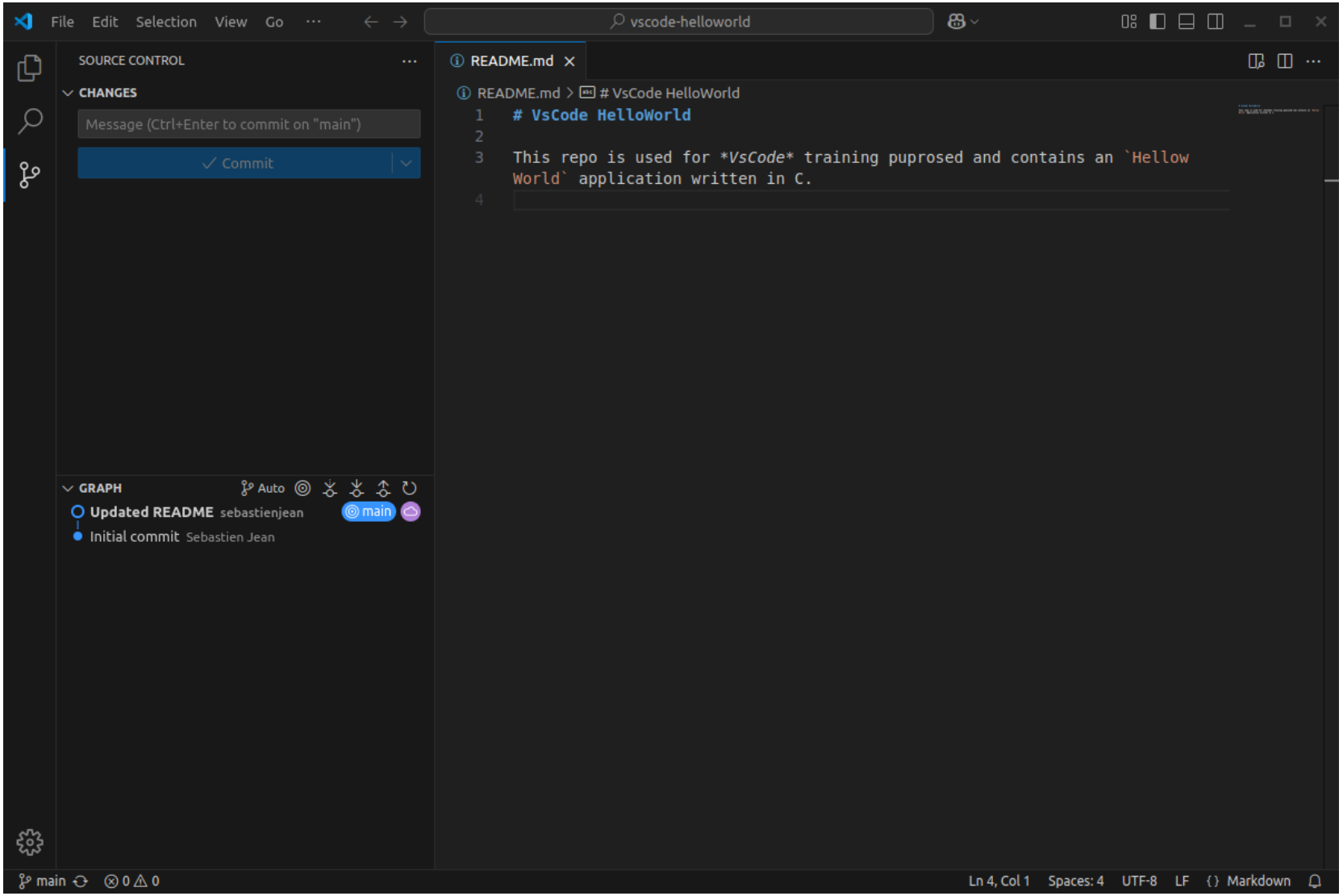
Produire une nouvelle version (*commit*)



- Sans surprise le commit est toujours local

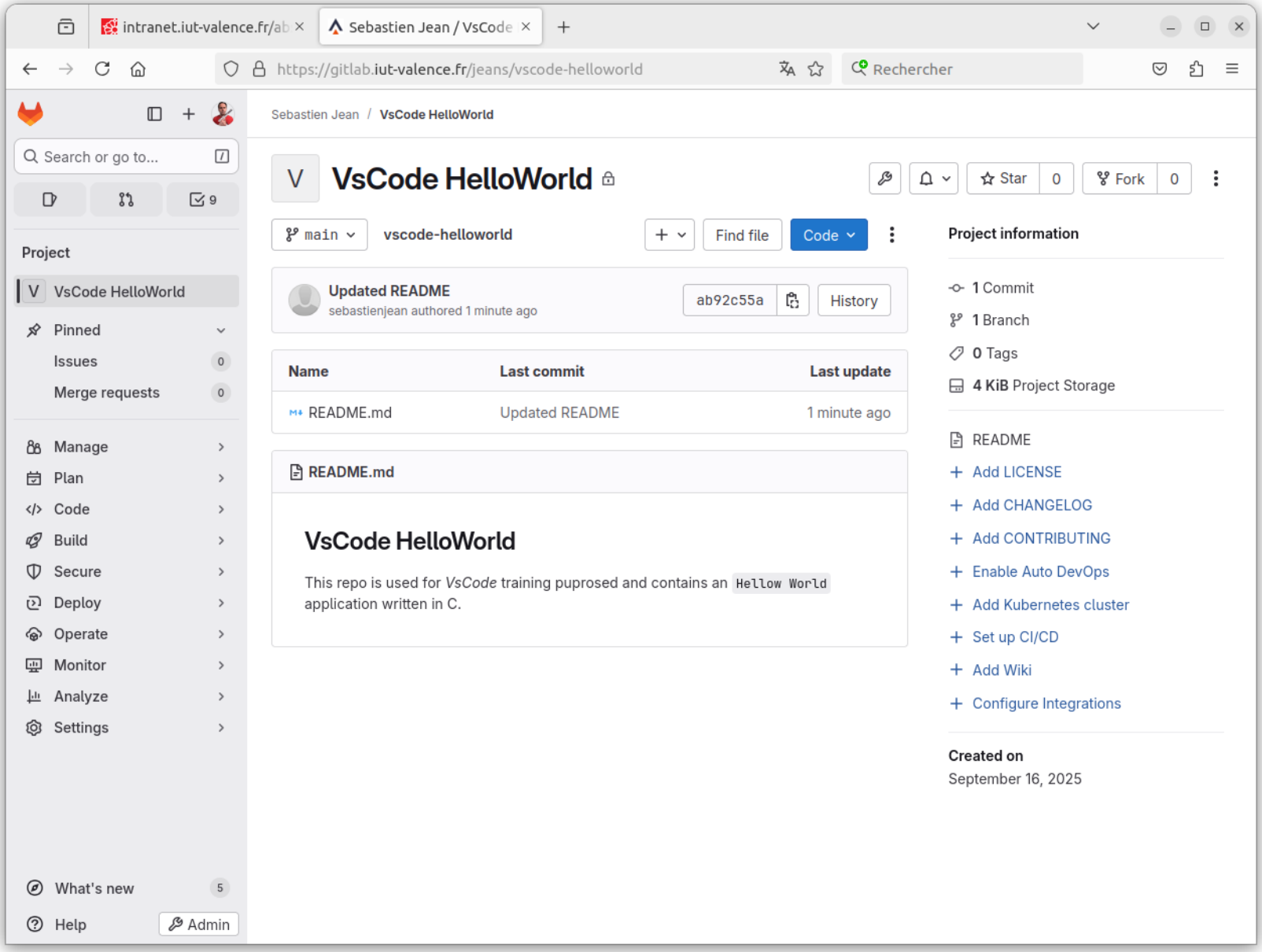
IUT Valence
Université Grenoble Alpes

Pousser le *commit* sur le serveur

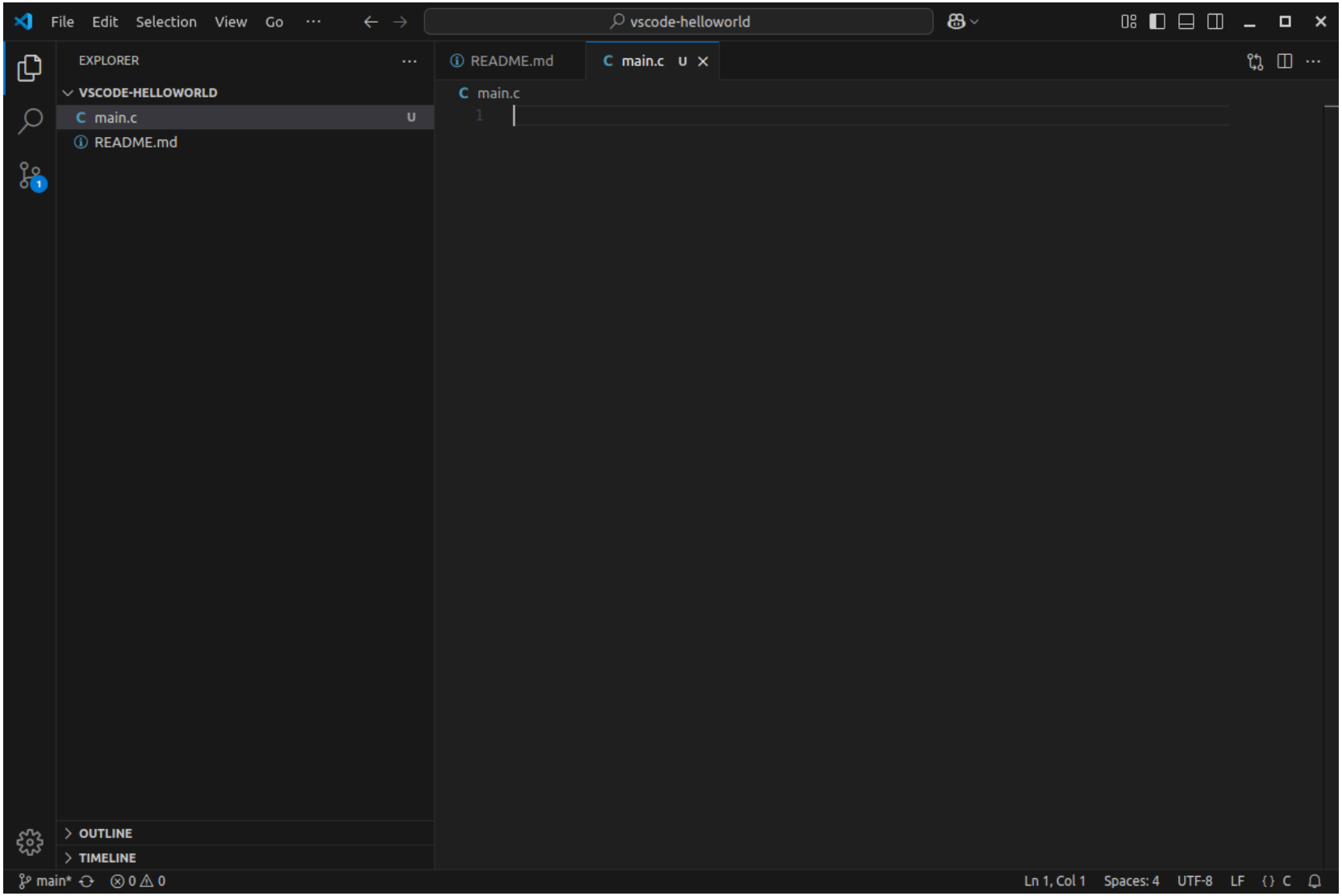


IUT Valence
Université Grenoble Alpes

Pousser le *commit* sur le serveur (fin)



Créer un fichier .c



IUT Valence
Université Grenoble Alpes

Hello world !, V0

Code C

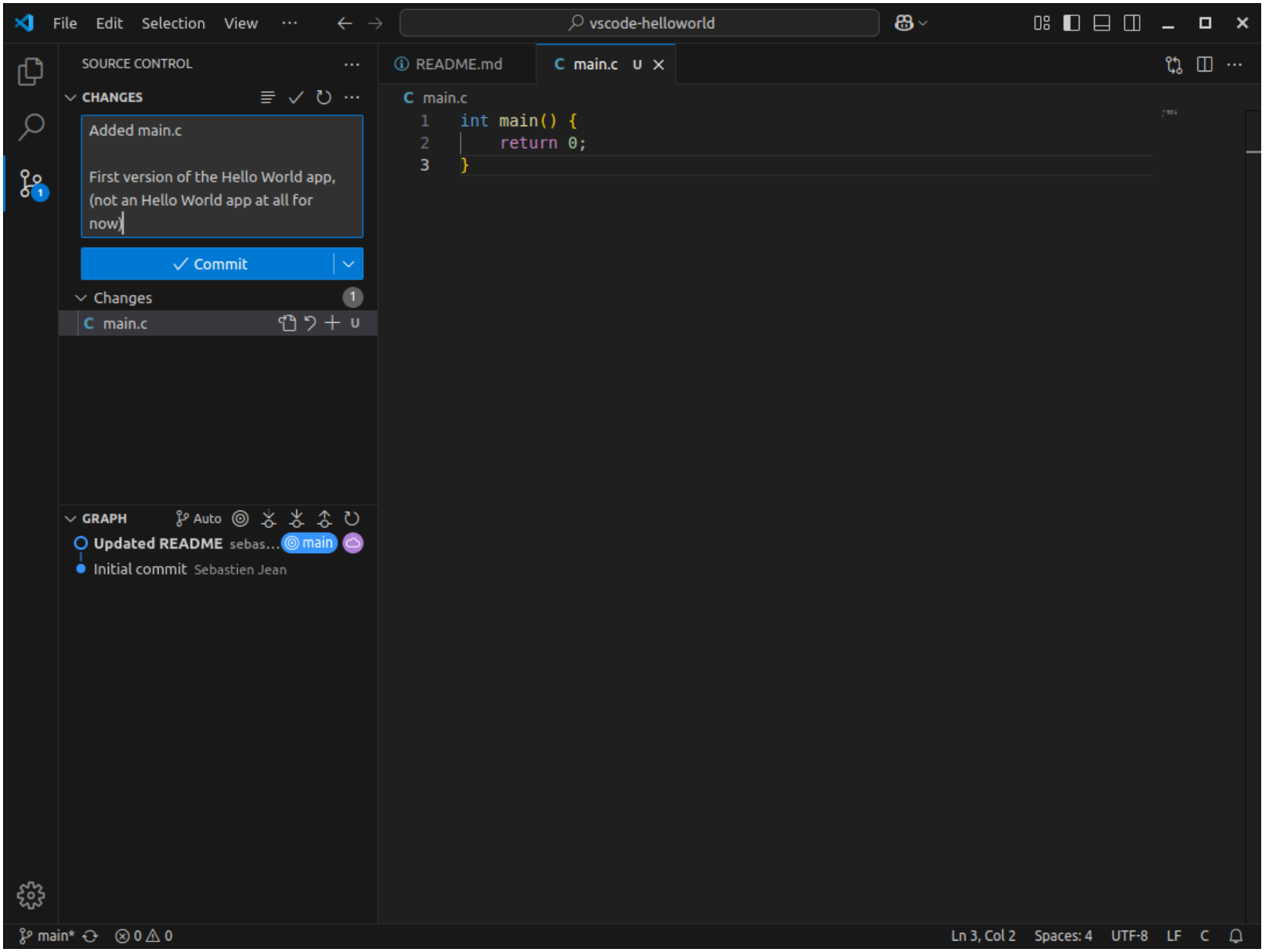
```c
int main() {

  return 0;
}
```

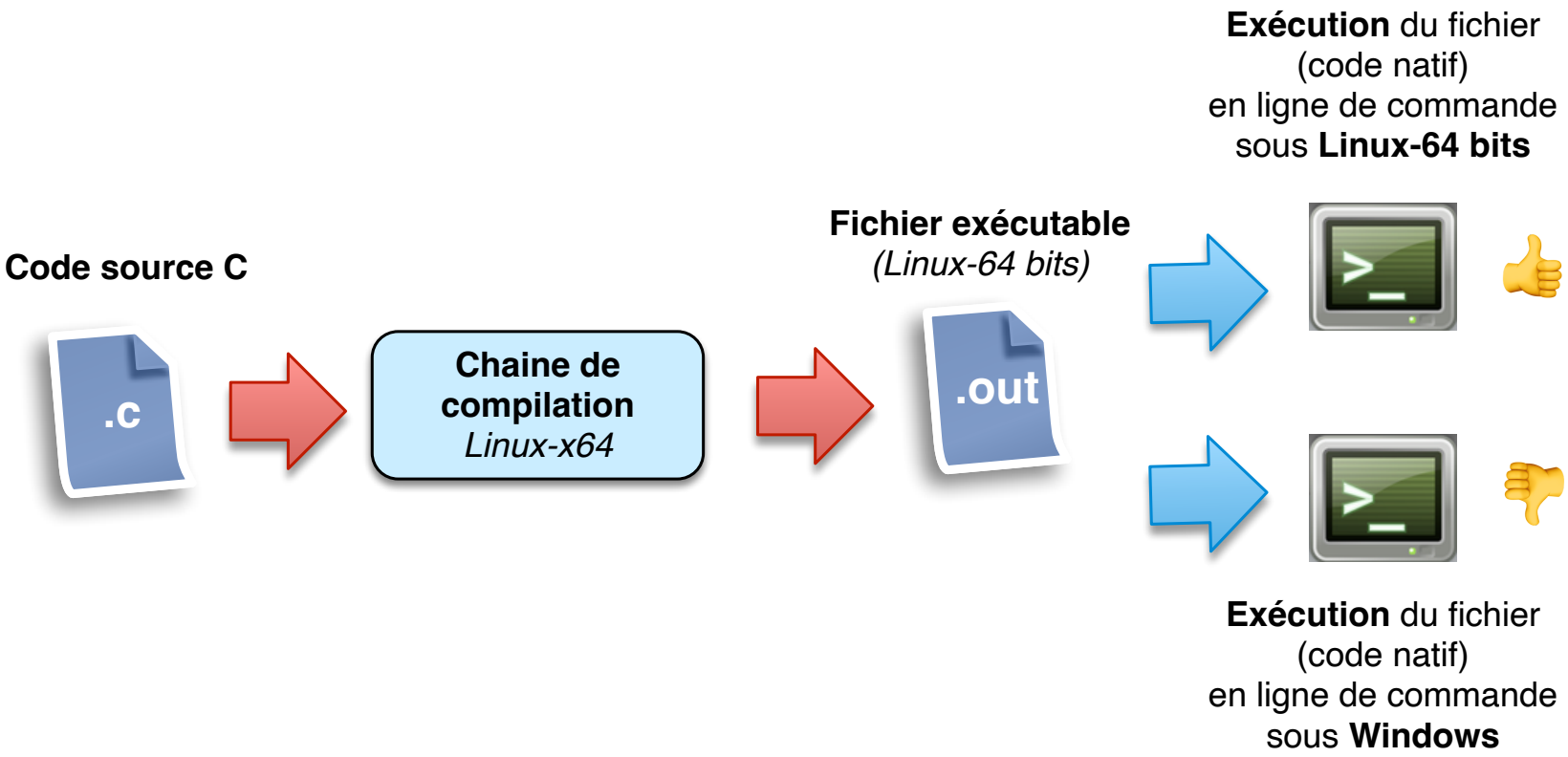
- N.B. : Le programme est destiné à être exécuté depuis un terminal
- La valeur retournée par la fonction `main` est un **statut de terminaison**
  - $0 \rightarrow$ normal,
  - $\neq 0 \rightarrow$ anormal,

IUT Valence Université Grenoble Alpes

Hello world !, V0 : commit



Compiler et exécuter un programme

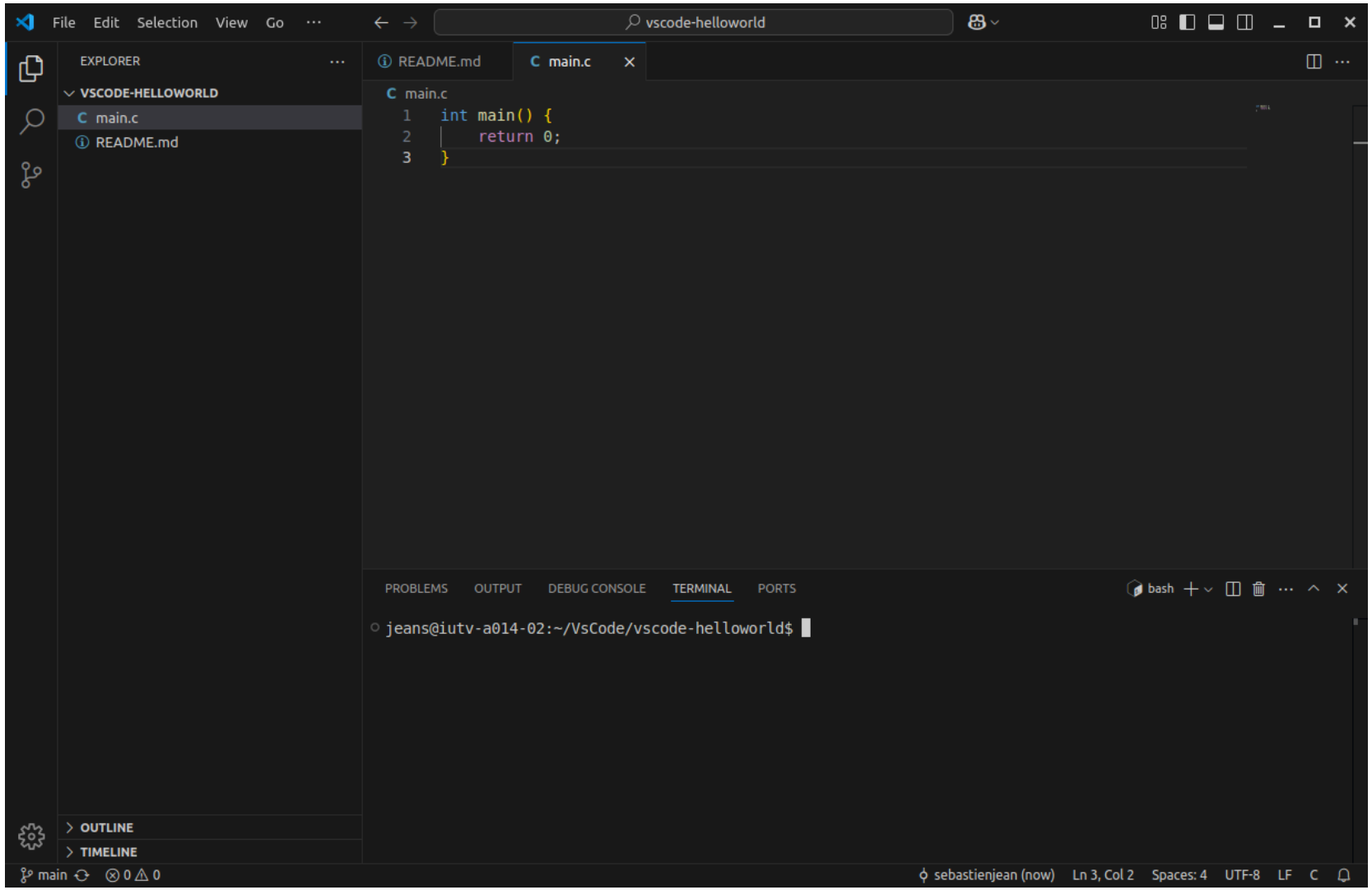


- N.B. : version **très** simplifiée
  - La **chaine de compilation** est une suite d'outils comprenant le compilateur mais pas que ...
  - Le processus de "compilation" comporte un certain nombre d'étapes intermédiaires

IUT Valence Université Grenoble Alpes

Vue *Terminal* de *VsCode*

- Terminal intégré (dans le répertoire ouvert)



Compiler et exécuter le programme

- Depuis le terminal, **compiler** le fichier `main.c` à l'aide de `gcc`
- **Observer** le contenu détaillé (`ls -al`) du répertoire
- **Caractériser** `a.out`
- **Exécuter** `a.out`

IUT Valence
Université Grenoble Alpes

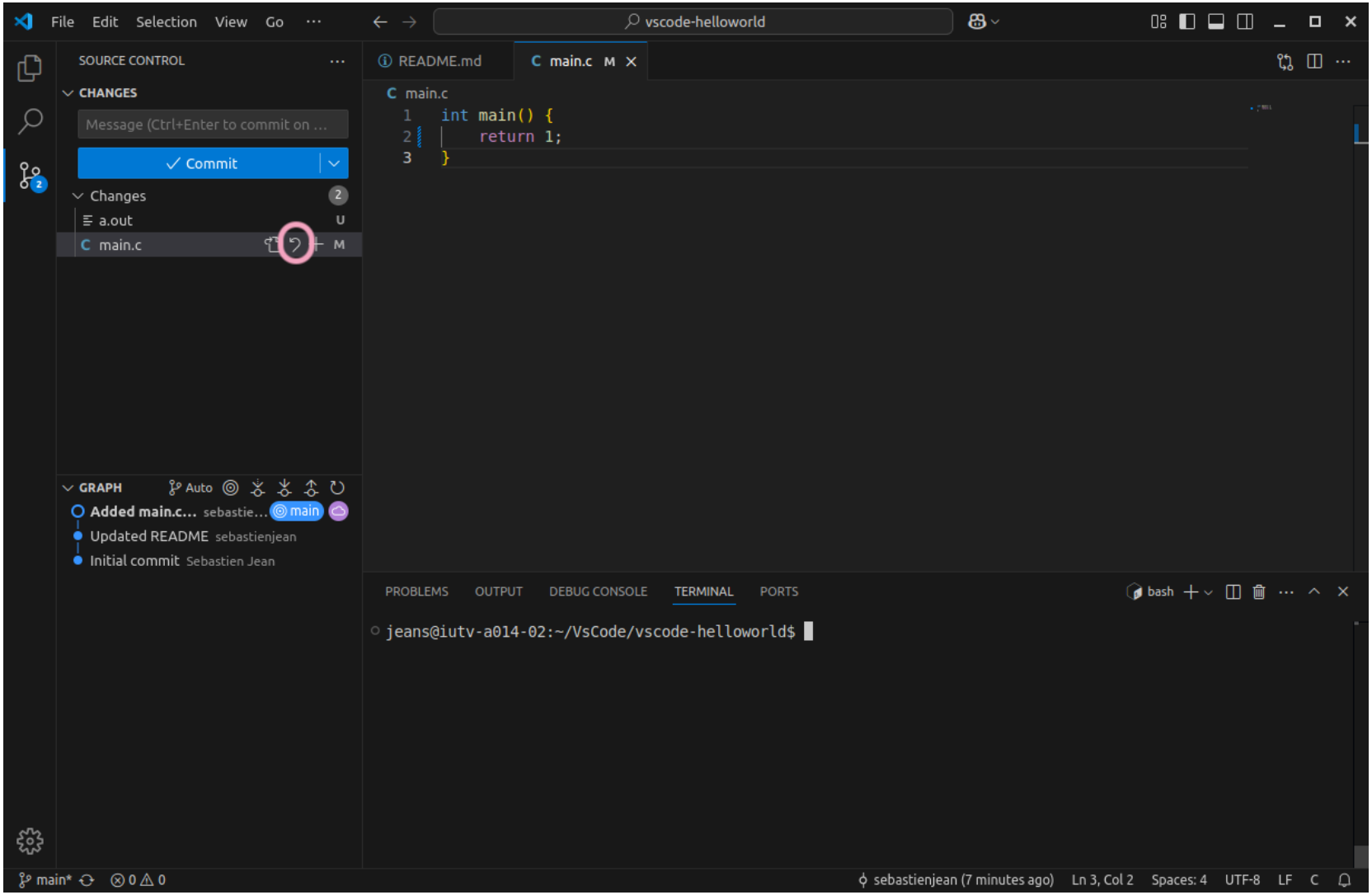
Compiler et exécuter le programme (fin)

- N.B. : dans le *shell Linux* (`bash`) la variable d'environnement `$?` contient le statut de terminaison de la dernière commande exécutée

- **Afficher** le statut de terminaison de l'exécutable `a.out`
- **Modifier** le code source pour changer le statut de terminaison
- **Recompiler** `main.c`
- **Afficher** le statut de terminaison de l'exécutable `a.out`

IUT Valence Université Grenoble Alpes

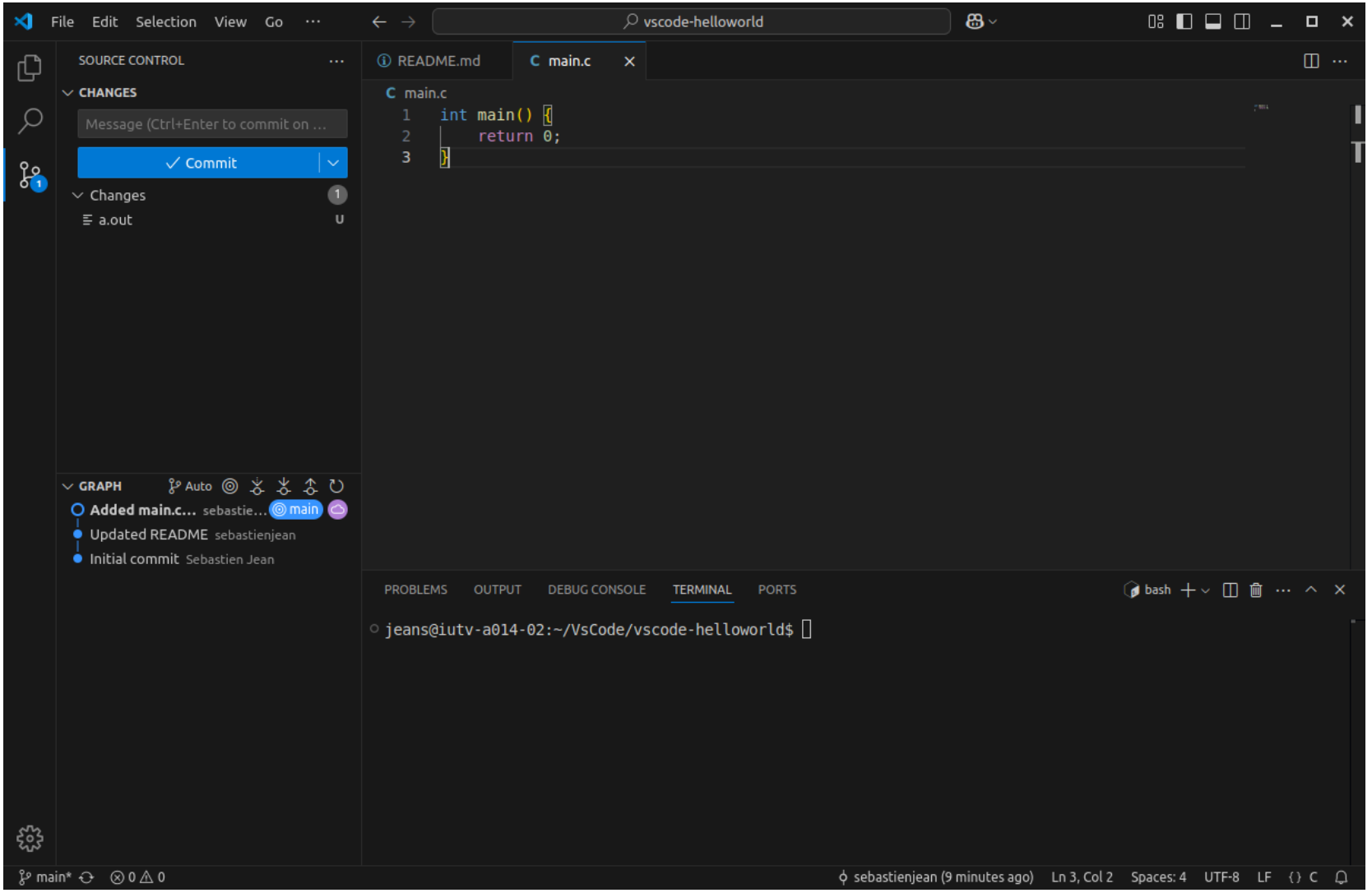
Annuler les modifications du fichier



- Quelle est la commande *git* correspondante ?

IUT Valence Université Grenoble Alpes

Annulation les modifications du fichier (fin)



IUT Valence
Université Grenoble Alpes

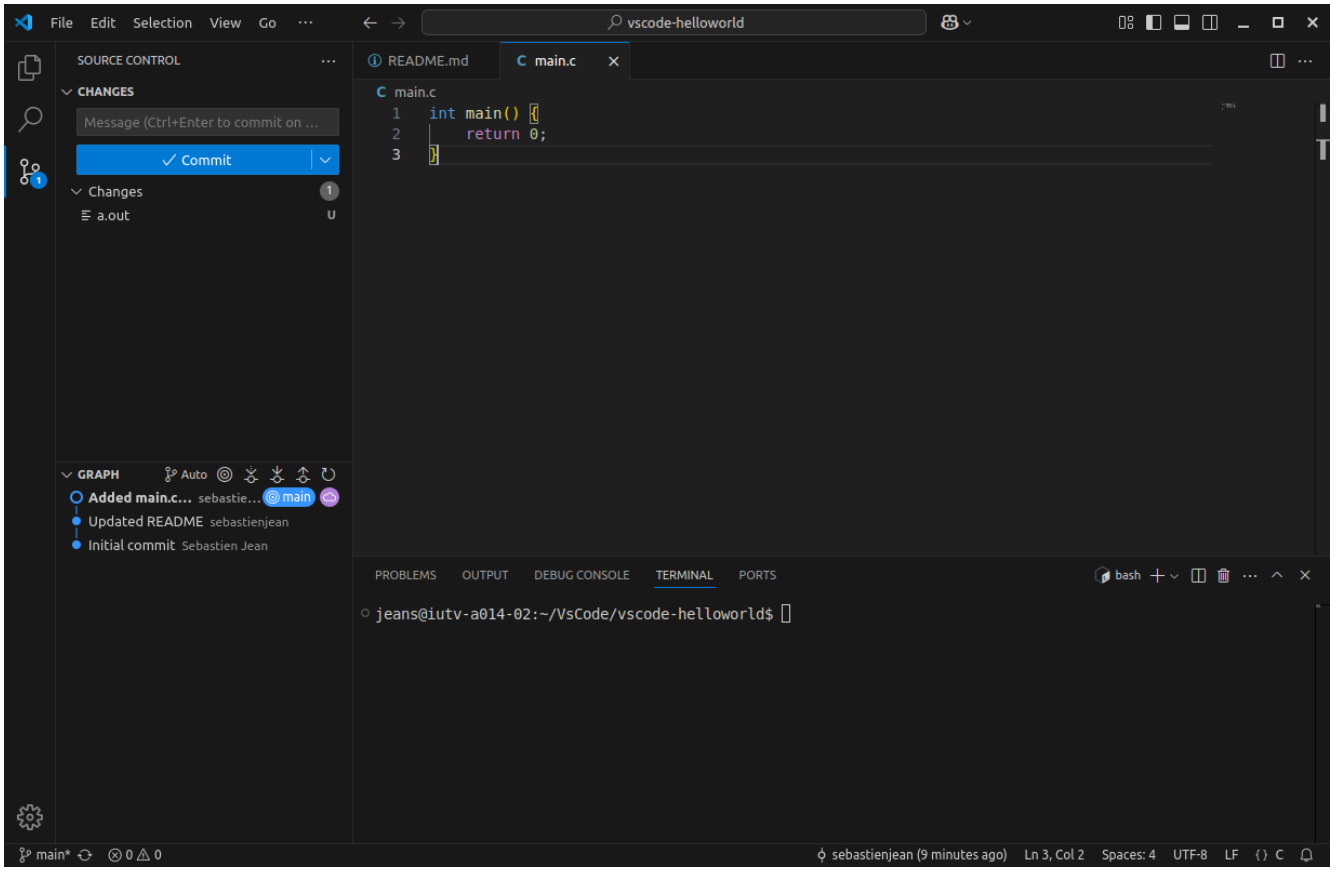
Ignorer le suivi d'un fichier

- N.B. : On peut ajouter dans n'importe quel répertoire du dépôt un fichier `.gitignore` qui spécifie les **ressources (fichiers, répertoires) à ignorer** dans ce répertoire et en dessous (il est possible d'utiliser des *jokers* pour spécifier des ensembles de ressources)

- En utilisant les menus contextuels, **Ignorer** le fichier `a.out`
- **Observer** l'état des changements
- Question subsidiaire : pourquoi ignorer ce fichier ?

IUT Valence
Université Grenoble Alpes

Ignorer le suivi d'un fichier (fin)



- **Effectuer un commit** du fichier `.gitignore`

IUT Valence
Université Grenoble Alpes

Hello world !, V1

Code C, V1

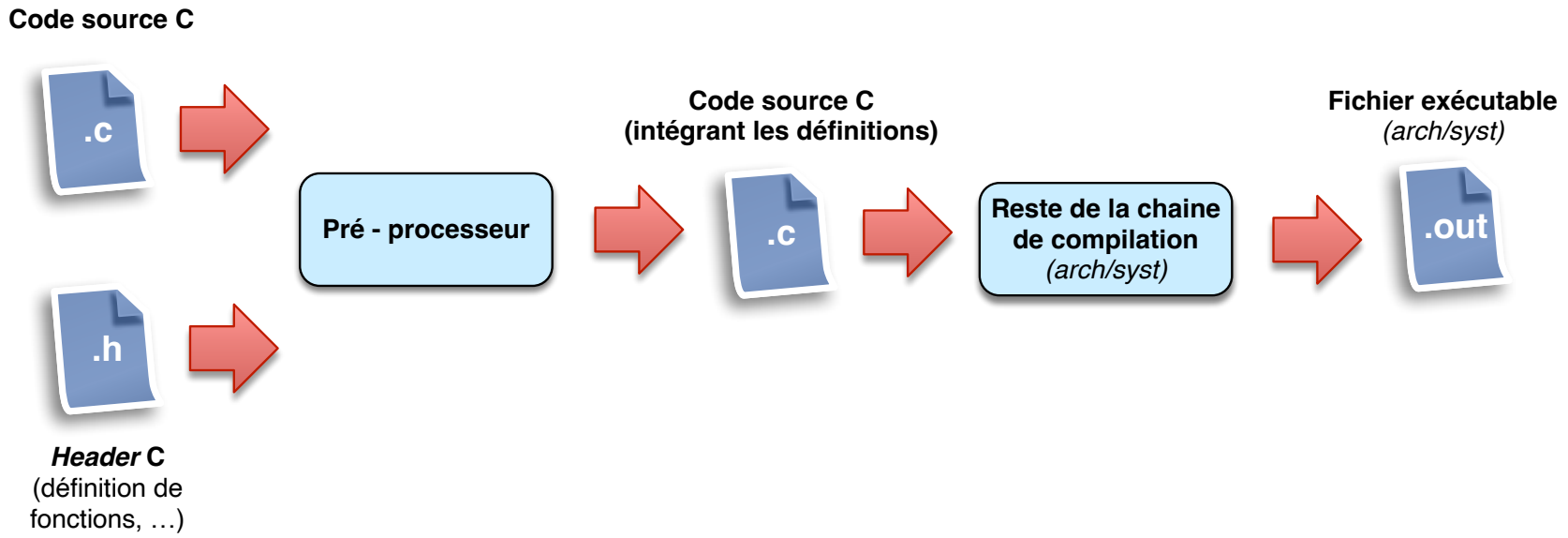
```c
#include <stdio.h>

int main() {
   printf("Hello World!\n");
   return 0;
}
```

- N.B. : la fonction `printf` est une **fonction prédéfinie de la librairie standard C** (*libC*)
  - Elle affiche sur la sortie standard la chaine de caractères passée en paramètres (sans passer à la ligne)
  - Le caractère spécial $\backslash n$ représente un **saut de ligne**

IUT Valence Université Grenoble Alpes

Compiler et exécuter un programme (suite)



- N.B. : version **toujours très simplifiée**
  - Un fichier `.c` faisant référence à une fonction externe doit inclure le fichier de définition `.h` contenant sa signature
  - Le **pré-processeur** résout notamment les inclusions de fichiers `.h`
  - A la sortie du pré-processeur, le fichier `.c` résulte de la fusion du fichier `.c` original et des fichiers `.h` inclus

IUT Valence Université Grenoble Alpes

Compiler et exécuter le programme

- Depuis le terminal, **compiler** le fichier `main.c` à l'aide de `gcc`
  - Utiliser l'option `-o` pour que l'exécutable soit nommé `main` et non plus `a.out`
- **Exécuter** `a.out`, **observer** la sortie sur le terminal
- **Ignorer** le fichier `main`, **effectuer un commit** et **pousser le commit** sur le serveur

IUT Valence
Université Grenoble Alpes

Compléments sur l'utilisation de *VsCode*

- Palette de commandes
- Chercher/remplacer
- Multi-curseurs
- ...

IUT Valence
Université Grenoble Alpes