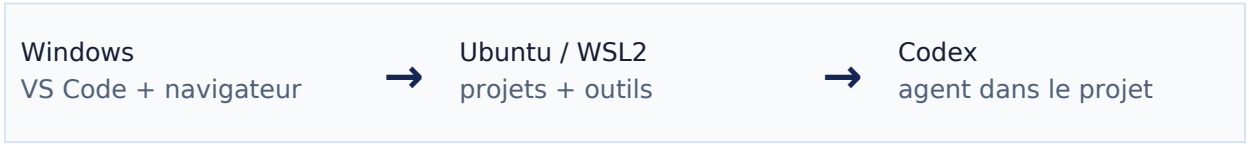
# Installer et utiliser Codex dans VS Code avec Ubuntu

Guide simple - à partir du terminal Ubuntu déjà opérationnel dans WSL2

| | |
|---|---|
| **Périmètre** | Ce document commence une fois Ubuntu accessible. Il ne couvre pas l'installation de Windows, de WSL2 ni d'Ubuntu, et n'inclut pas Cline. |

## Objectif

Obtenir un environnement où l'interface de VS Code fonctionne sous Windows tandis que le projet, le terminal et Codex travaillent dans Ubuntu.




### Avant de commencer

Prévoir : Windows avec WSL2 et Ubuntu fonctionnels, VS Code installé, et un compte ChatGPT compatible avec Codex.

# 1. Accéder à Ubuntu et vérifier WSL2

**1** Ouvrir Ubuntu
Depuis le menu Démarrer de Windows, rechercher Ubuntu puis ouvrir le terminal.

**2** Vérifier le noyau
Dans le terminal Ubuntu, exécuter :

```
uname -a
```

Le résultat doit contenir Linux et généralement microsoft-standard-WSL2. L'invite ressemble à :

```
dnguyen@ordinateur:~$
```

**À retenir** Le symbole ~ représente votre dossier personnel Linux, par exemple `/home/dnguyen`.

## Créer le dossier de projets

```
mkdir -p ~/projets
cd ~/projets
```

Conservez vos projets dans Ubuntu, par exemple `/home/dnguyen/projets`, plutôt que dans `/mnt/c/...`. Les outils de développement y sont généralement plus rapides et les permissions plus prévisibles.

# 2. Connecter VS Code à Ubuntu

**1** Installer l’extension WSL
Dans VS Code, ouvrir Extensions avec Ctrl + Maj + X, rechercher WSL et choisir celle publiée par Microsoft.

**2** Ouvrir une fenêtre WSL
Cliquer sur l’indicateur distant en bas à gauche, puis choisir Connect to WSL ou Connect to WSL using Distro… → Ubuntu.

**3** Contrôler la connexion
En bas à gauche, vérifier la présence de WSL: Ubuntu.

| VS Code - repères visuels |
| --- |
| Barre d'état, en bas à gauche : WSL: Ubuntu |
| Palette de commandes : Ctrl + Maj + P |
| Terminal intégré : Terminal → New Terminal |

## Test dans le terminal intégré

```
uname -a
pwd
```

pwd affiche le dossier courant. Si VS Code est bien connecté, ces commandes s’exécutent dans Ubuntu.

| | |
| --- | --- |
| **Si cela bloque** | Ouvrir Ctrl + Maj + P, lancer Developer: Reload Window, puis reconnecter Ubuntu. |

# 3. Installer l’extension officielle Codex

1. Ouvrir les extensions
   Utiliser Ctrl + Maj + X.

2. Rechercher Codex
   Rechercher Codex - OpenAI’s coding agent et vérifier que l’éditeur est OpenAI.

3. Installer dans Ubuntu
   Cliquer sur Install in WSL: Ubuntu si ce bouton apparaît. Sinon, utiliser Install dans la fenêtre déjà connectée à WSL.

| | |
|---|---|
| **Version stable** | Ne choisissez pas Switch to Pre-Release Version pour une première installation. La préversion donne accès plus tôt à certaines nouveautés, mais peut être moins stable. |

## Ouvrir Codex

Si l’icône Codex n’apparaît pas dans la barre latérale :

```
Ctrl + Maj + P
Codex: Open Codex Sidebar
```

Vous pouvez aussi faire un clic droit sur la barre verticale des icônes et cocher Codex. Si la commande reste absente, rechargez la fenêtre avec Developer: Reload Window.

# 4. Connecter Codex à ChatGPT

**1** Lancer la connexion
Dans le panneau Codex, cliquer sur Sign in with ChatGPT.

**2** S’authentifier
Le navigateur s’ouvre. Se connecter avec le compte ChatGPT abonné, puis autoriser l’accès.

**3** Revenir dans VS Code
Après validation, revenir dans la fenêtre WSL: Ubuntu.

**Important** Choisissez Sign in with ChatGPT, pas Use API Key. Une clé API utilise un compte API et une facturation séparés de l’abonnement ChatGPT.

| Configuration attendue |
| --- |
| Fenêtre VS Code : WSL: Ubuntu |
| Extension : Codex, éditeur OpenAI |
| Authentification : compte ChatGPT |
| Projet : dossier situé sous /home/… |

# 5. Ouvrir un projet et donner une première tâche

Créez ou placez le projet dans Ubuntu. Exemple :

```
mkdir -p /home/dnguyen/projets/nadia-fitness
cd /home/dnguyen/projets/nadia-fitness
```

Dans VS Code, utilisez File → Open Folder… et ouvrez ce dossier. La première consigne peut être volontairement prudente :

| | |
|---|---|
| **Première consigne** | Analyse ce projet et explique-moi sa structure sans modifier les fichiers. |

## Pour garder le contrôle

| | |
|---|---|
| **Consigne conseillée** | Analyse d’abord le projet. Indique-moi les dépendances et les commandes que tu souhaites exécuter, puis demande mon accord avant toute installation. |

## Ce que Codex peut faire dans Ubuntu

| Action | Exemple | Comportement |
|---|---|---|
| Dépendances Node | `npm install` | Crée généralement `node_modules/` dans le projet. |
| Dépendances PHP | `composer install` | Installe les bibliothèques du projet. |
| Environnement Python | `python3 -m venv .venv` | Isole les paquets Python du projet. |
| Paquets système | `sudo apt install …` | Peut nécessiter votre accord et votre mot de passe sudo. |

# 6. Prévisualiser un site statique

Dans votre exemple, Codex a exécuté :

```
cd /home/dnguyen/projets/nadia-fitness
python3 -m http.server 8000
```

Première ligne : cd place le terminal dans le dossier exact du projet.

Deuxième ligne : Python lance un petit serveur HTTP pour ce dossier sur le port 8000. Aucun package n’est installé et le projet n’est pas publié sur Internet.

| | |
|---|---|
| **Adresse locale** | Ouvrir http://localhost:8000 dans le navigateur Windows. |

Le serveur reste actif tant que le processus tourne. Pour l’arrêter proprement, revenir dans le terminal concerné et appuyer sur :

```
Ctrl + C
```

Ce mini-serveur convient aux sites HTML/CSS/JavaScript statiques. Un projet PHP, SPIP, Node/Vite ou autre framework doit utiliser son serveur ou sa commande propre.

# 7. Mémo rapide

| | |
|---|---|
| Vérifier Ubuntu | `uname -a` |
| Ouvrir Codex | `Ctrl + Maj + P → Codex: Open Codex Sidebar` |
| Ouvrir un projet | `File → Open Folder… → /home/dnguyen/projets/...` |
| Demander une analyse | `Analyse ce projet sans modifier les fichiers.` |
| Lancer un site statique | `python3 -m http.server 8000` |
| Ouvrir le site | `http://localhost:8000` |
| Arrêter le serveur | `Ctrl + C` |

## Bonnes pratiques

- Garder les projets sous `/home/votre_utilisateur/projets`.
- Lire les commandes proposées avant de les autoriser.
- Ne jamais copier une clé API, un mot de passe ou un secret dans une conversation.
- Utiliser Git pour pouvoir vérifier et annuler les changements.
- Arrêter les serveurs locaux inutilisés avec `Ctrl + C`.

| | |
|---|---|
| **Vous êtes prêt** | VS Code sert d'interface, Ubuntu fournit l'environnement Linux, et Codex peut analyser, modifier, tester et lancer le projet dans ce même environnement. |