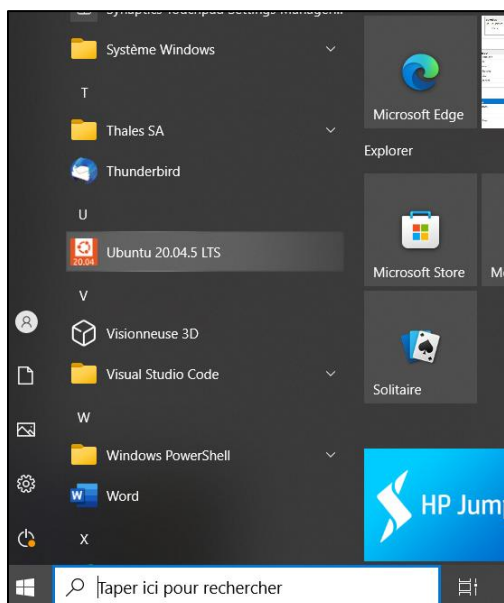


Binômotron : tech stack

Pour prendre les bonnes habitudes dès le premier brief, vous allez travailler dans un environnement LINUX, avec une base de données sur DOCKER. Ce petit document vous explique comment évoluer dans ces technologies.

WSL



Démarrez votre UBUNTU WSL en cliquant sur l'icône dans les applis de WINDOWS.

Dans la console LINUX, créer un répertoire de travail (*iadev-python*, par exemple) en ligne de commande :

```
$ mkdir iadev-python
```

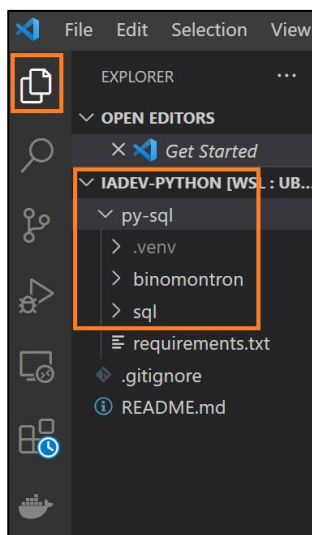
Positionnez-vous dans ce répertoire de travail :

```
$ cd iadev-python
```

Lancez VS CODE :

```
$ code .
```

Structurer le projet

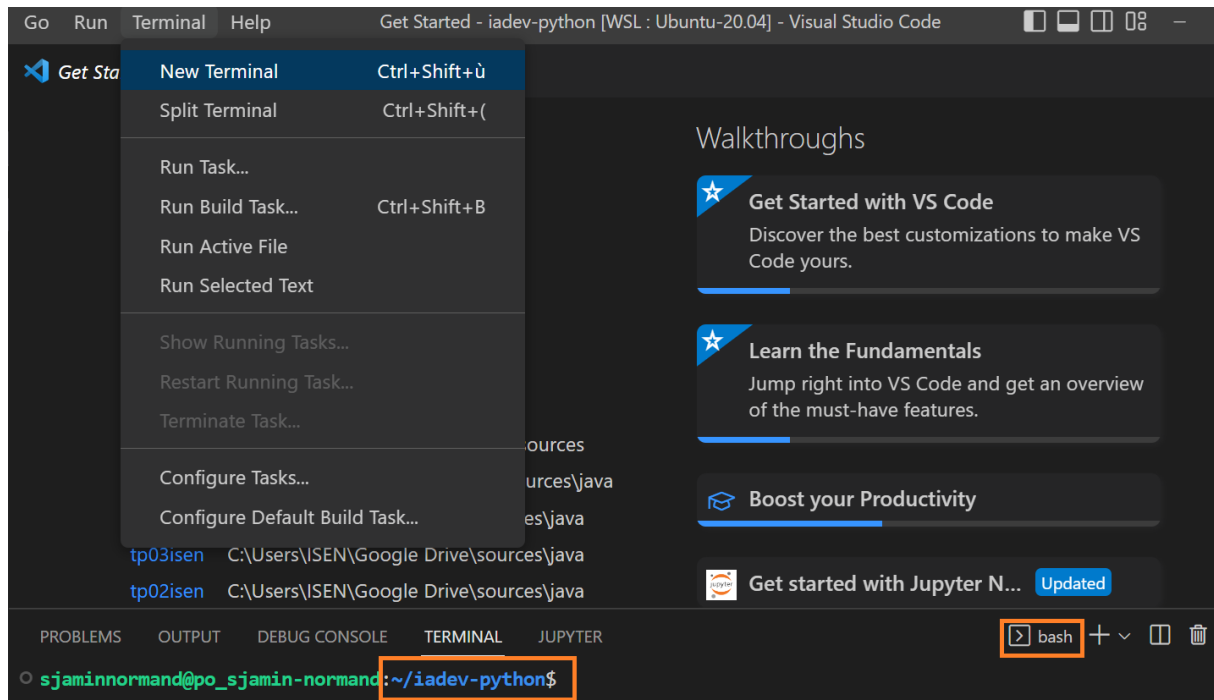


Le répertoire de travail que vous venez de créer vous servira pour tous vos briefs. Pour bien travailler proprement, il est recommandé de créer un répertoire par projet. De plus, il est plus intéressant de placer les ressources en dehors du projet.

Voici un exemple d'architecture créée à partir de VS CODE :

- Le dossier */binomontron* contiendra tous le code Python concernant ce projet,
- Le dossier */sql* contient tous les fichiers utiles aux bases de données.
- Pour simplifier le travail pour les futurs briefs, les dossiers précédents sont dans un répertoire */py-sql*, qui servira pour tous les travaux sur le SQL.

Ouvrez le terminal (menu **Terminal/New Terminal**) pour continuer à travailler en ligne de commande dans VS CODE. Vérifiez bien que le terminal s'est ouvert dans votre répertoire de travail.



SQL sous Docker

Grâce à DOCKER, nous allons pouvoir simuler un serveur de donnée MySQL dans un conteneur. De plus, nous allons pouvoir y accéder grâce à un client graphique, PHPMYADMIN.

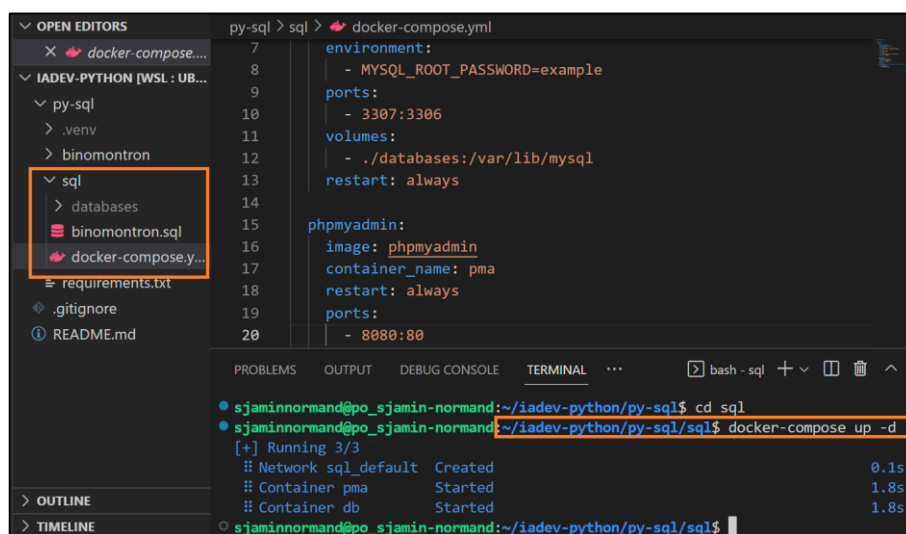
Placez le fichier *docker-compose.yml* dans le répertoire */sql*.

Dans le terminal, placez vous dans le répertoire */sql* grâce à la commande **cd** :

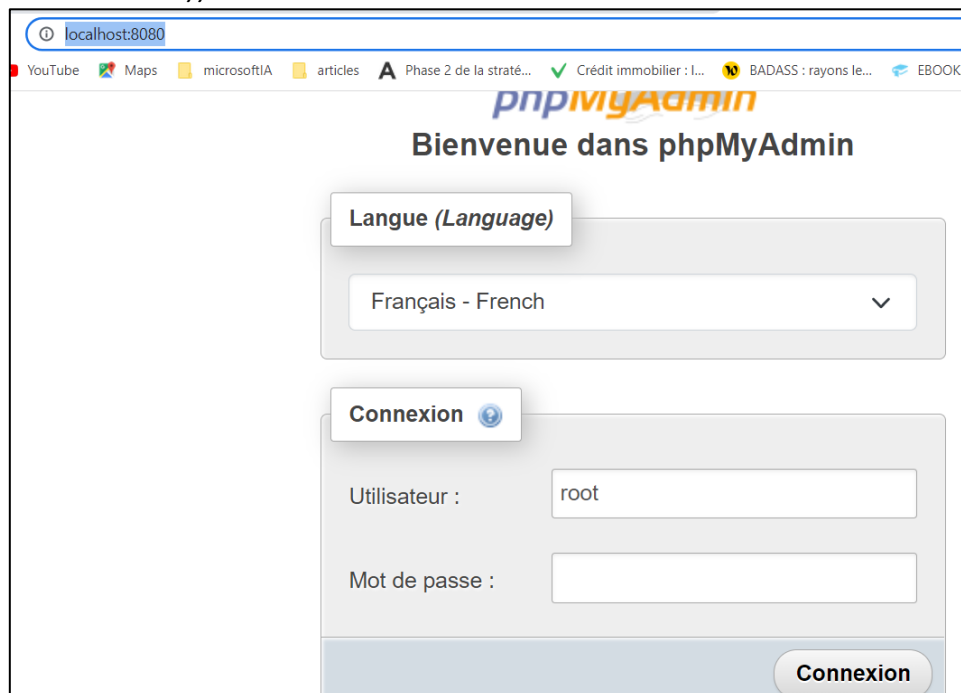
```
$ cd sql
```

Maintenant, vous allez lancer la construction des conteneurs avec la commande suivante :

```
$ docker-compose up -d
```



Quand tout est terminé, que vous avez récupéré le prompt sur la console, ouvrez un navigateur et entrez l'url `//localhost :8080`.



Vous êtes dans PHPMYADMIN, ce qui vous ouvre l'accès à votre base de données MYSQL grâce au login *root* et au mot de passe *exemple*.

Environnement virtuel

Lorsqu'on travaille en Python, il est recommandé de coder son projet au sein d'un environnement virtuel. Ainsi votre projet s'exécute indépendamment des installations de votre machine, avec ses propres bibliothèques et sa version de Python. De plus, il est facile de dupliquer cet environnement sur d'autres postes, pour déployer l'appli, par exemple, ou pour travailler en équipe projet.

Dans le terminal, placez-vous dans répertoire `/py-sql`. Si vous êtes dans un sous-répertoire vous pouvez remonter avec la commande suivante :

```
$ cd ..
```

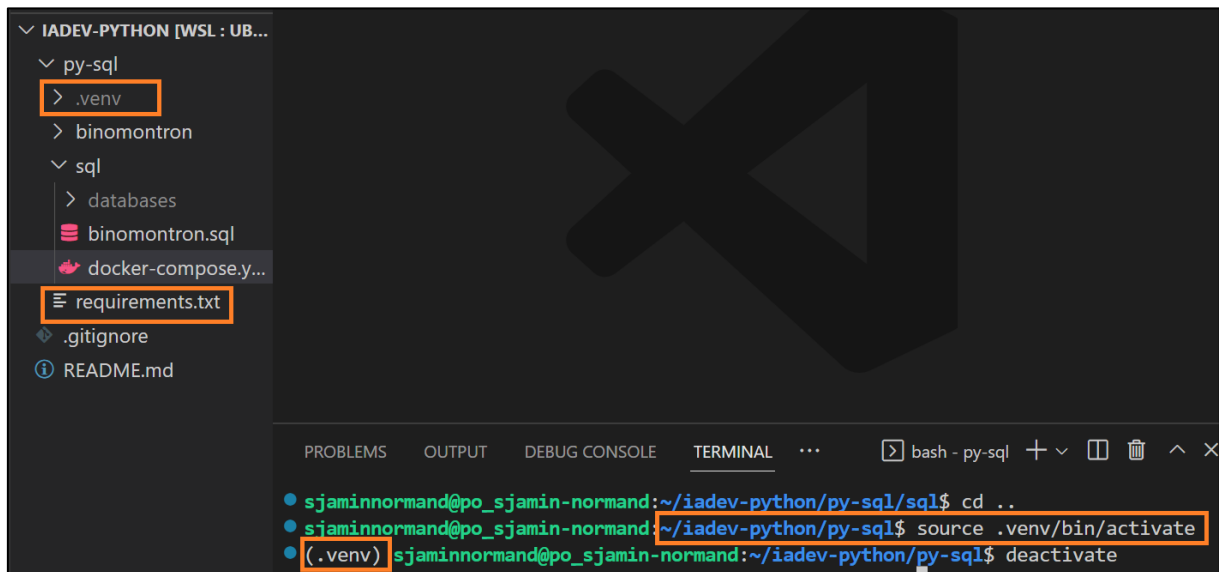
Créez un environnement virtuel :

```
$ python3 -m venv .venv
```

Si tout se passe bien, votre répertoire `/.venv` doit apparaître dans votre répertoire de travail. Vous pouvez maintenant activer l'environnement virtuel :

```
$ source .venv/bin/activate
```

Vous savez que votre environnement est actif s'il apparaît devant votre prompt



Pour ce projet, nous allons devoir faire communiquer PYTHON avec MYSQL. Pour ça, il faut installer un connecteur :

```
$ pip install mysql-connector-python
```

Cette installation c'est faite en suivant la documentation officielle suivante :

<https://dev.mysql.com/doc/connector-python/en/connector-python-installation-binary.html>

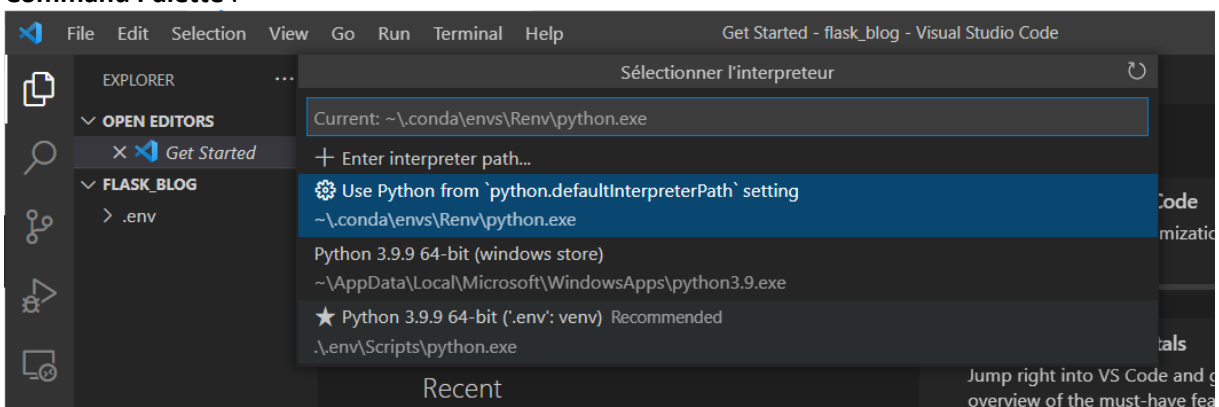
Pour vérifier que le package est bien installé, nous allons les lister :

```
$ pip list
```

Maintenant que tout est ok, nous allons sauvegarder notre environnement dans un fichier requirement.txt, pour être capable de le dupliquer ou de le restaurer :

```
$ pip freeze > requirements.txt
```

Avant de commencer à coder, vérifiez que vous travailler bien avec l'interpréteur de votre environnement. Le `.env` dans notre cas, grâce à la commande **Python : Select Interpreter** dans la **Command Palette** :



Quand vous avez fini de travailler, vous pouvez quitter l'environnement :

```
$ deactivate
```

Trouver les fichiers WSL sous Windows

Lorsque vous travaillez sur WSL, vous êtes toujours sur votre WINDOWS. Tous vos fichiers sont donc sauvegardés sur WINDOWS. Vous pouvez les retrouver sur le lecteur de réseau [\\wsl\\$](#).

Vous retrouverez tous vos travaux sous le chemin suivant : [\\wsl\\$\Ubuntu-20.04\home\\[user\]](#)

