

# TP1 – Commencer un projet Django

## Introduction, Contexte et Objectifs

Pour cette suite de TP, nous utiliserons des outils importants :

- un environnement Linux
- un outil de versionnage (git, et le serveur gitlab de l'IUT)
- un framework de programmation web serveur python : Django, installé en local
- Un framework css : Bootstrap 5
- Un framework JS : Chart.js pour des graphiques évolués
- Un IDE comme Visual Studio Code

Le framework Django nous permettra de concevoir petit à petit un travail classique déjà fait en PHP/MySQL pur au S3 (avec un sujet différent) pour voir comment gérer de A à Z, sans tout réinventer, un projet web avec des données persistantes.

Le framework css Bootstrap 5 nous permettra de bénéficier d'un travail conséquent et abouti, en css et même en JavaScript. Nous pourrions utiliser du css pur, mais la logique de ce module est de nous orienter vers ces outils.

Le framework JS chart.js nous permettra d'inclure des graphiques offrant des fonctionnalités statistiques utiles à l'administrateur.

Le serveur gitlab de l'IUT nous habituera, si ce n'est pas déjà le cas, à des pratiques nécessaires au travail collaboratif. Même si, en l'occurrence, vous ne collaborerez qu'avec vous-même, puisque les TP sont à faire individuellement.

Nous travaillerons sous Debian 12 car il faut s'habituer à cet environnement. Tous les TP sont conçus pour fonctionner en ligne de commande dans un terminal Linux et les captures d'écran sont cohérentes avec cet environnement.

Avec l'IDE Visual Studio Code, nous aurons accès à un éditeur de texte évolué et qui intégrera également un terminal, dont vous pourrez vous servir à la place du terminal classique.

Dans cette suite de TP, nous construirons un site de commerce en ligne, avec une interface client simple, et une interface administrateur proposant les fonctionnalités classiques (créer, lire, mettre à jour, supprimer), et des fonctionnalités plus évoluées (divers graphiques et résultats statistiques pour résumer l'activité du site). C'est un classique à maîtriser.

## Pour fixer les idées

Vous disposez sur Moodle d'une vidéo de présentation des fonctionnalités essentielles attendues une fois le travail terminé. Ceci peut vous servir de guide pour la suite.

## Le gitLab de l'iut

Nous pouvons utiliser le gitlab de l'IUT (ou toute autre plateforme de dépôts git) pour disposer d'un dépôt de fichiers distant. C'est une habitude essentielle en travail d'équipe. Même quand l'équipe est réduite à une personne.

### 1. Connexion au gitlab de l'iut

- a. allez à [https://git.iut-orsay.fr/users/sign\\_in](https://git.iut-orsay.fr/users/sign_in)
- b. connectez-vous avec login court et password

### 2. Création d'un nouveau projet et clonage

- a. allez dans vos projets et créez un nouveau projet vierge nommé **webpizza**. Vous pouvez le créer en privé. Laissez cochée l'option de création du README.md, c'est un fichier d'accueil de la page gitlab de votre projet, vous pourrez la personnaliser plus tard.
- b. allez dans le projet et récupérez l'url du clonage en https qui doit être du type  
`https://git.iut-orsay.fr/prenom.nom/webpizza.git`

- c. Dans votre architecture locale, créez un nouveau dossier **home/Documents/R5A05/**. Allez dans ce dossier.
- d. Lancez le clonage à cet endroit par la commande **git clone <lien du projet git>**

```
sgagne@2006-122: ~/Documents/IUT/R5A05
Fichier  Édition  Affichage  Recherche  Terminal  Aide
sgagne@2006-122:~$ cd Documents/IUT/R5A05/
sgagne@2006-122:~/Documents/IUT/R5A05$ git clone https://git.iut-orsay.fr/sebastien.gagne/webpizza.git
Clonage dans 'webpizza'...
Username for 'https://git.iut-orsay.fr': sgagne
Password for 'https://sgagne@git.iut-orsay.fr':
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Réception d'objets: 100% (3/3), fait.
sgagne@2006-122:~/Documents/IUT/R5A05$
```

- e. vérifiez qu'il y a maintenant dans **R5A05** un répertoire **webpizza**, avec à l'intérieur un dossier caché **.git**

```
sgagne@2006-122: ~/Documents/IUT/R5A05/webpizza
Fichier  Édition  Affichage  Recherche  Terminal  Aide
sgagne@2006-122:~$ cd Documents/IUT/R5A05/
sgagne@2006-122:~/Documents/IUT/R5A05$ git clone https://git.iut-orsay.fr/sebastien.gagne/webpizza.git
Clonage dans 'webpizza'...
Username for 'https://git.iut-orsay.fr': sgagne
Password for 'https://sgagne@git.iut-orsay.fr':
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Réception d'objets: 100% (3/3), fait.
sgagne@2006-122:~/Documents/IUT/R5A05$ ls
archives  TP_webpizza  webpizza
sgagne@2006-122:~/Documents/IUT/R5A05$ cd webpizza/
sgagne@2006-122:~/Documents/IUT/R5A05/webpizza$ ls -al
total 20
drwxr-xr-x 3 sgagne sgagne 4096 11 juil. 13:52 .
drwxr-xr-x 5 sgagne sgagne 4096 11 juil. 13:51 ..
drwxr-xr-x 8 sgagne sgagne 4096 11 juil. 13:52 .git
-rw-r--r-- 1 sgagne sgagne 6193 11 juil. 13:52 README.md
sgagne@2006-122:~/Documents/IUT/R5A05/webpizza$
```

### 3. Commandes essentielles git

a. où en êtes-vous ?

La commande « **git status** » vous indique si votre copie locale est en retard (ou en avance) sur la copie distante.

b. ajouter des fichiers à suivre

La commande « **git add .** » ajoute tous les fichiers pas encore suivis dans la liste des fichiers à suivre. Tout changement sur ces fichiers sera indiqué.

c. valider les changements sur les fichiers suivis

La commande « **git commit -m "message pour ce commit"** » permet de valider les changements et indique un message de repérage.

d. pousser les changements validés sur la branche distante

La commande « **git push** » met à jour la branche distante.

e. la commande de récupération de la branche distante

La commande « **git pull** » est très importante. Si vous n'êtes pas seul à travailler sur le même projet, il y a de grandes chances que des push aient été effectués par d'autres et si vous vous apprêtez à faire un push sans avoir mis à jour votre version locale avant, il va y avoir des conflits de version qu'il vaut mieux éviter.

f. Vous pourrez aussi avoir besoin des commandes git relatives à la création et à la fusion de branches. A voir au cas où.

### 4. Chronologie classique

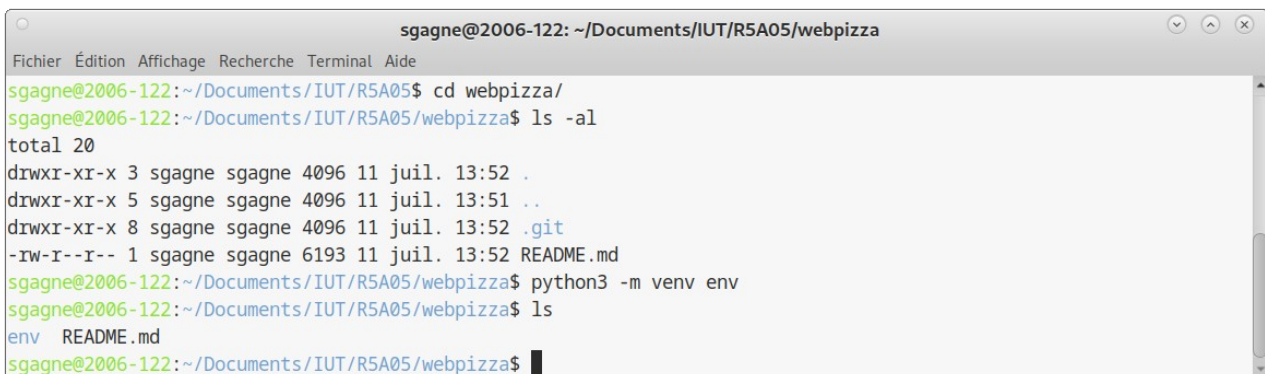
- a. git pull
- b. ... vos changements de code
- c. git status
- d. git add .
- e. git commit -m "mon message personnalisé"
- f. git push
- g. git status par acquis de conscience

Il vaut mieux faire trop de pull/push que pas assez.

## L'outil Django pour le projet webpizza

### 1. Installation de Django

- a. Allez dans le répertoire du projet **webpizza** et lancez la commande **python3 -m venv env**, ce qui crée un environnement virtuel nommé « env » pour Django. Vous le verrez par un **ls** dans le répertoire.



```
sgagne@2006-122: ~/Documents/IUT/R5A05/webpizza
Fichier  Édition  Affichage  Recherche  Terminal  Aide
sgagne@2006-122:~/Documents/IUT/R5A05$ cd webpizza/
sgagne@2006-122:~/Documents/IUT/R5A05/webpizza$ ls -al
total 20
drwxr-xr-x 3 sgagne sgagne 4096 11 juil. 13:52 .
drwxr-xr-x 5 sgagne sgagne 4096 11 juil. 13:51 ..
drwxr-xr-x 8 sgagne sgagne 4096 11 juil. 13:52 .git
-rw-r--r-- 1 sgagne sgagne 6193 11 juil. 13:52 README.md
sgagne@2006-122:~/Documents/IUT/R5A05/webpizza$ python3 -m venv env
sgagne@2006-122:~/Documents/IUT/R5A05/webpizza$ ls
env  README.md
sgagne@2006-122:~/Documents/IUT/R5A05/webpizza$
```

- b. Activer l'environnement virtuel par la commande **source env/bin/activate**



```
sgagne@2006-122: ~/Documents/IUT/R5A05/webpizza
Fichier  Édition  Affichage  Recherche  Terminal  Aide
sgagne@2006-122:~/Documents/IUT/R5A05/webpizza$ ls -al
total 20
drwxr-xr-x 3 sgagne sgagne 4096 11 juil. 13:52 .
drwxr-xr-x 5 sgagne sgagne 4096 11 juil. 13:51 ..
drwxr-xr-x 8 sgagne sgagne 4096 11 juil. 13:52 .git
-rw-r--r-- 1 sgagne sgagne 6193 11 juil. 13:52 README.md
sgagne@2006-122:~/Documents/IUT/R5A05/webpizza$ python3 -m venv env
sgagne@2006-122:~/Documents/IUT/R5A05/webpizza$ ls
env  README.md
sgagne@2006-122:~/Documents/IUT/R5A05/webpizza$ source env/bin/activate
(env) sgagne@2006-122:~/Documents/IUT/R5A05/webpizza$
```

L'activation se voit par (env). L'environnement doit être activé faire tourner le serveur web. Si une désactivation se produit, reprendre la procédure d'activation du b.

- c. Installation de Django dans l'environnement virtuel

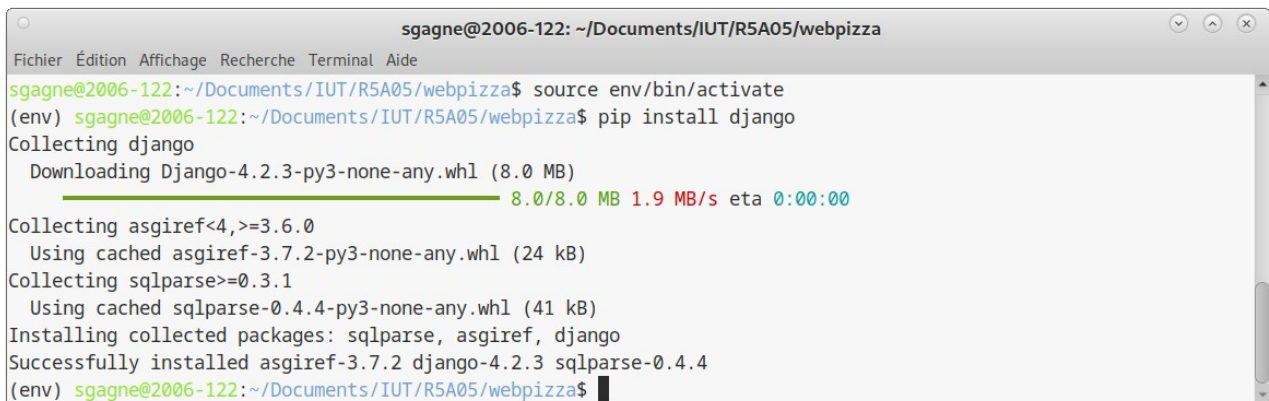
Dans webpizza, en n'oubliant pas d'activer l'environnement virtuel, nous allons installer le framework django pour notre projet. Au préalable, à l'IUT, nous avons besoin de variables pour le proxy :

D'abord, les commandes pour les deux variables :

```
HTTP_PROXY=http://proxy.iut-orsay.fr:3128  
HTTPS_PROXY=http://proxy.iut-oray.fr:3128  
export HTTP_PROXY  
export HTTPS_PROXY
```

Puis la commande d'installation de django :

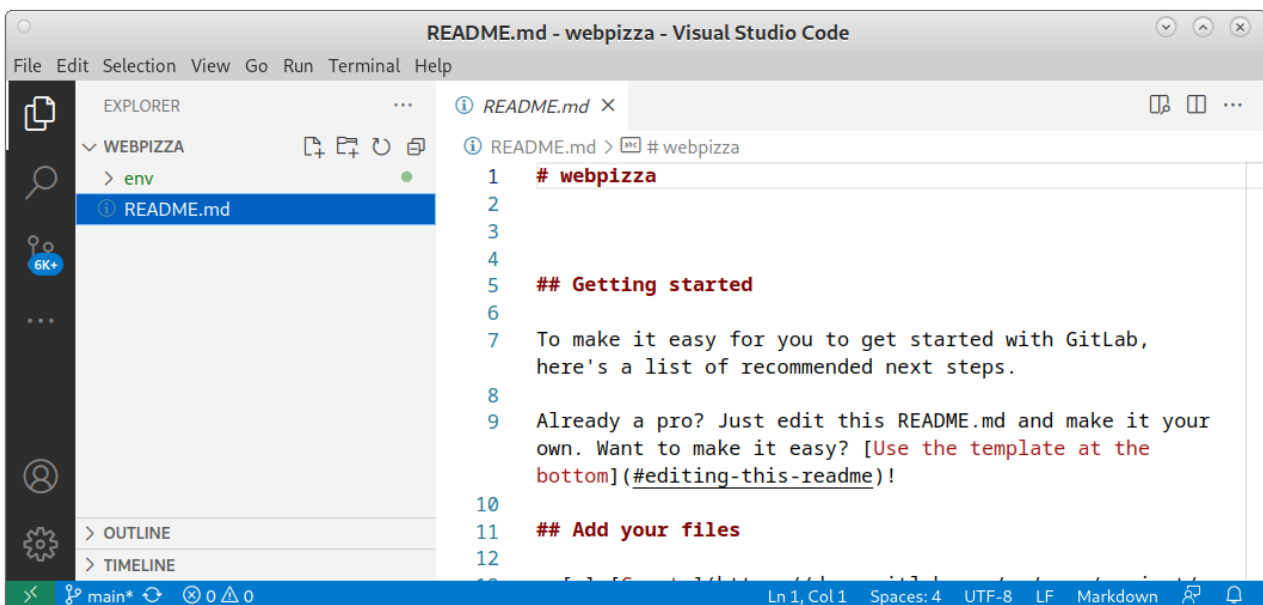
```
pip install django
```



```
sgagne@2006-122: ~/Documents/IUT/R5A05/webpizza  
Fichier Édition Affichage Recherche Terminal Aide  
sgagne@2006-122:~/Documents/IUT/R5A05/webpizza$ source env/bin/activate  
(env) sgagne@2006-122:~/Documents/IUT/R5A05/webpizza$ pip install django  
Collecting django  
  Downloading Django-4.2.3-py3-none-any.whl (8.0 MB)  
      ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.0/8.0 MB 1.9 MB/s eta 0:00:00  
Collecting asgiref<4,>=3.6.0  
  Using cached asgiref-3.7.2-py3-none-any.whl (24 kB)  
Collecting sqlparse>=0.3.1  
  Using cached sqlparse-0.4.4-py3-none-any.whl (41 kB)  
Installing collected packages: sqlparse, asgiref, django  
Successfully installed asgiref-3.7.2 django-4.2.3 sqlparse-0.4.4  
(env) sgagne@2006-122:~/Documents/IUT/R5A05/webpizza$
```

Si vous avez obtenu le message ci-dessus, tout est prêt.  
Sinon, recommencez.

- d. Ouvrez votre IDE VS Code, avec comme répertoire de travail webpizza :



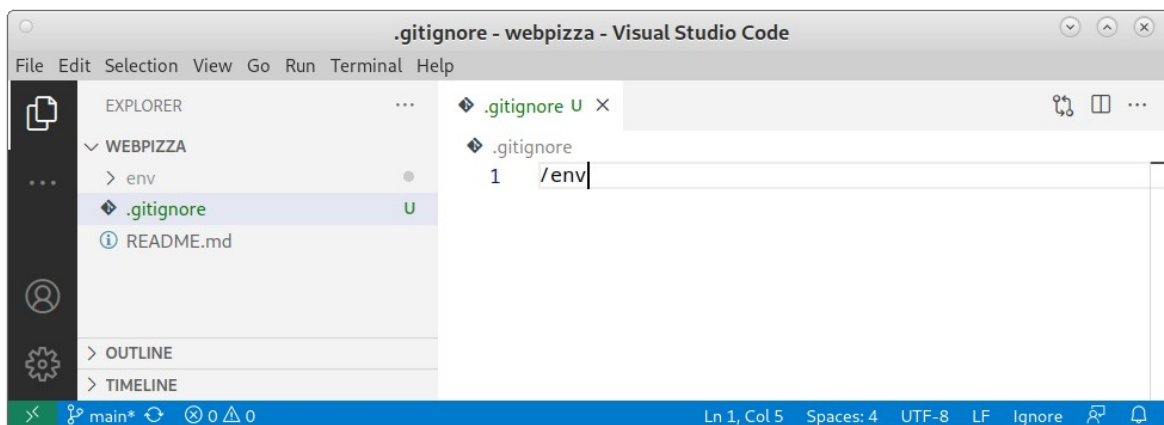
```
README.md - webpizza - Visual Studio Code  
File Edit Selection View Go Run Terminal Help  
EXPLORER  
  WEBPIZZA  
    > env  
    README.md  
6K+  
OUTLINE  
TIMELINE  
main* 0 0 0  
Ln 1, Col 1 Spaces: 4 UTF-8 LF Markdown
```

```
1 # webpizza  
2  
3  
4  
5 ## Getting started  
6  
7 To make it easy for you to get started with GitLab,  
  here's a list of recommended next steps.  
8  
9 Already a pro? Just edit this README.md and make it your  
  own. Want to make it easy? [Use the template at the  
  bottom](#editing-this-readme)!  
10  
11 ## Add your files  
12
```



## e. Le fichier .gitignore

Ce fichier indiquera les répertoires ou fichiers qui doivent être ignorés lors des procédures git. Créez, au niveau de la racine (au même niveau que README.md et env), un fichier **.gitignore** (ce sera un fichier caché) qui contiendra la ligne suivante :

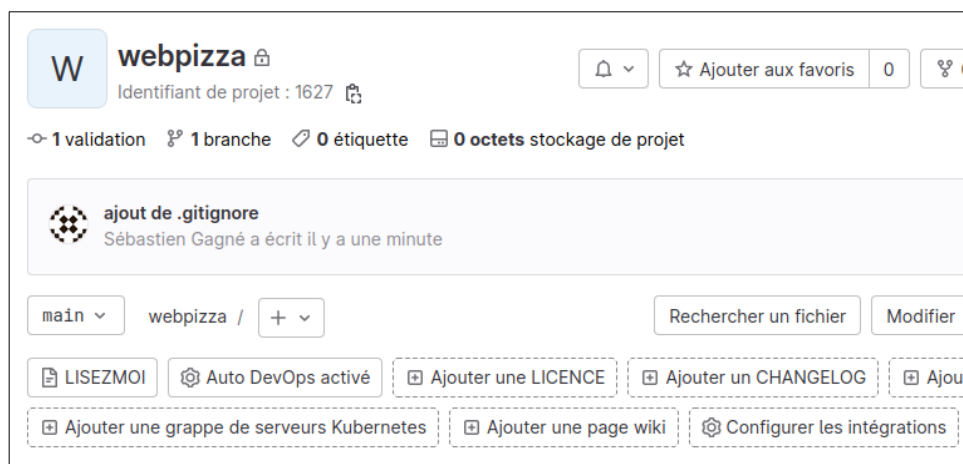


Ce fichier sera certainement mis à jour par la suite. Mettez à jour votre dépôt distant pour ajouter ce fichier .gitignore.

Rappel :

- git status
- git add .
- git commit -m "ajout du fichier .gitignore"
- git push

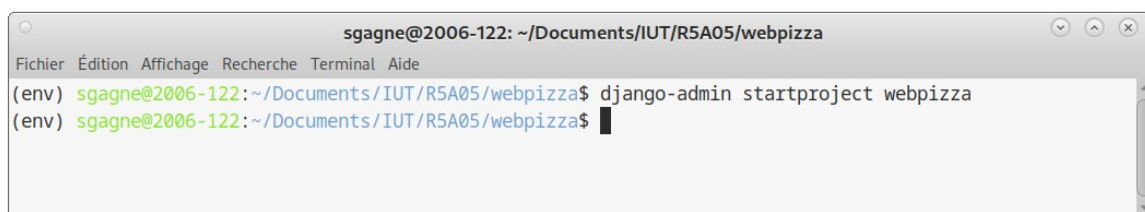
Vous pouvez exécuter ces commandes soit dans un terminal classique, soit dans celui mis à disposition par VS Code.



## Création du projet webpizza

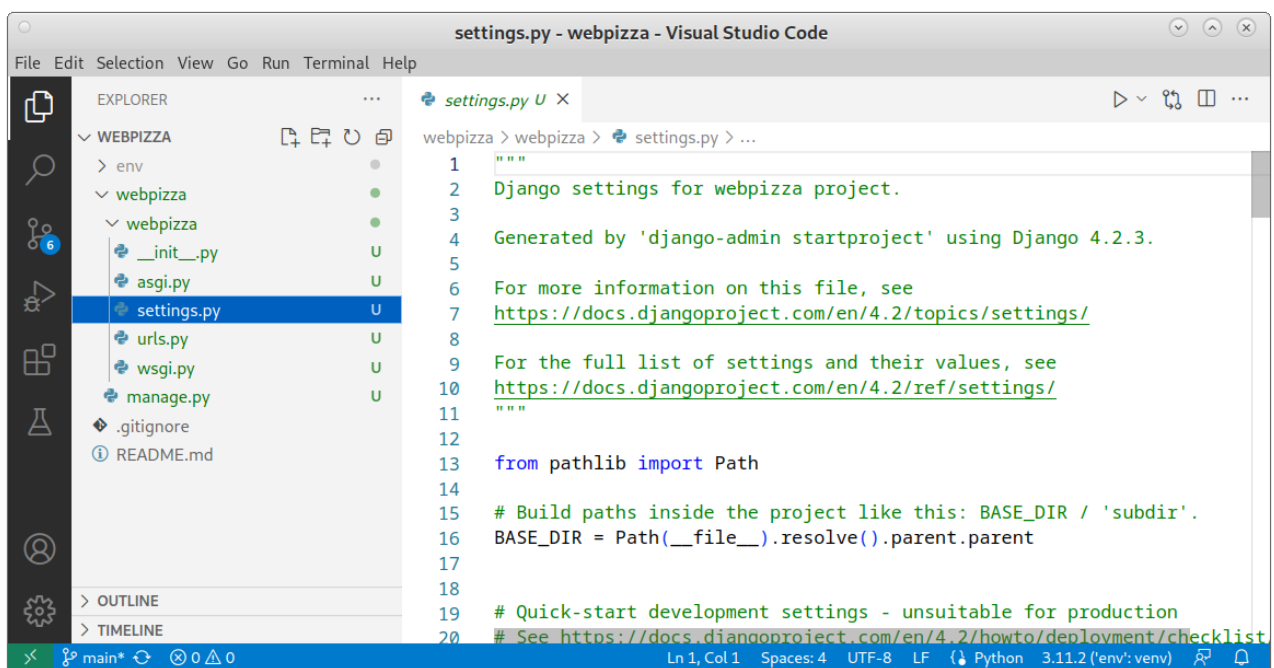
### 1. La commande `django-admin startproject webpizza`

- a. Nous allons créer un projet portant le même nom que le dossier la contenant : webpizza. Ce nom aurait pu être différent.



```
sgagne@2006-122: ~/Documents/IUT/R5A05/webpizza
Fichier Édition Affichage Recherche Terminal Aide
(env) sgagne@2006-122:~/Documents/IUT/R5A05/webpizza$ django-admin startproject webpizza
(env) sgagne@2006-122:~/Documents/IUT/R5A05/webpizza$
```

- b. Cette commande a en fait créé de nombreux fichiers que nous réutiliserons, et qui structurent la base de notre projet. Dans Visual Studio Code :



```
settings.py - webpizza - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
WEBPIZZA
  env
  webpizza
    __init__.py
    asgi.py
    settings.py
    urls.py
    wsgi.py
    manage.py
  .gitignore
  README.md
OUTLINE
TIMELINE
webpizza > webpizza > settings.py > ...
1 """
2 Django settings for webpizza project.
3
4 Generated by 'django-admin startproject' using Django 4.2.3.
5
6 For more information on this file, see
7 https://docs.djangoproject.com/en/4.2/topics/settings/
8
9 For the full list of settings and their values, see
10 https://docs.djangoproject.com/en/4.2/ref/settings/
11 """
12
13 from pathlib import Path
14
15 # Build paths inside the project like this: BASE_DIR / 'subdir'.
16 BASE_DIR = Path(__file__).resolve().parent.parent
17
18
19 # Quick-start development settings - unsuitable for production
20 # See https://docs.djangoproject.com/en/4.2/howto/deployment/checklist
```



## 2. La base de données du projet webpizza

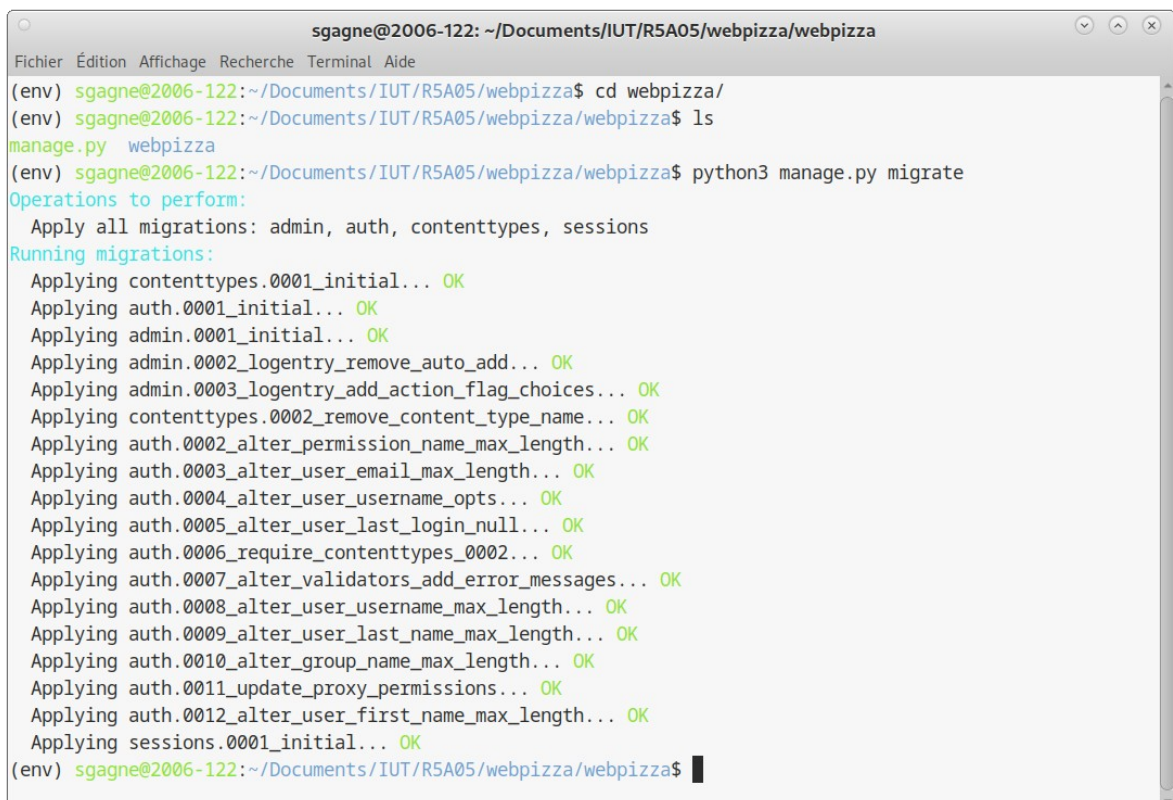
Une base sqlite par défaut est utilisée par Django. On peut configurer sans problème une base de données MySQL, mais à l'IUT les sécurités sont fortes pour pouvoir agir sur les bases MySQL autrement que via PHP qui a reçu les droits nécessaires.

Nous allons donc, en travaillant totalement en local, utiliser une base de données sqlite, avec comme interface graphique sqldbviewer.

### a. La commande `python3 manage.py migrate`

Nous allons utiliser la commande `python3 manage.py migrate` qui va mettre en place la base de données du projet. Elle s'exécute **dans le répertoire webpizza** (celui du projet), **où on trouve aussi le fichier manage.py** qui permet de lancer les commandes en ligne python.

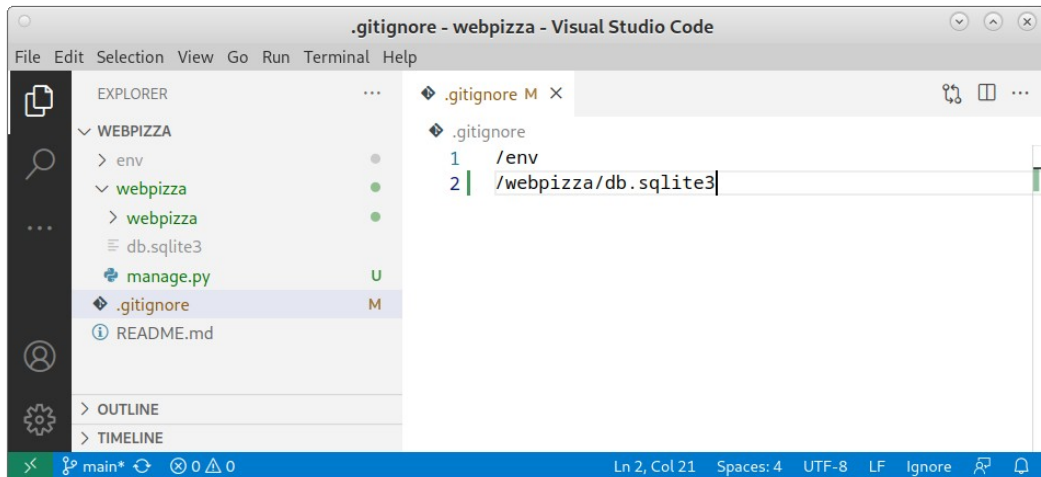
Cette commande ne sert pas qu'à la mise en place, mais de manière générale à permettre les « migrations », qui ne sont que les traductions de nos modèles vers des tables de données. Exécutez cette commande. Le résultat en console doit être similaire à :



```
sgagne@2006-122: ~/Documents/IUT/R5A05/webpizza/webpizza
Fichier Édition Affichage Recherche Terminal Aide
(env) sgagne@2006-122:~/Documents/IUT/R5A05/webpizza$ cd webpizza/
(env) sgagne@2006-122:~/Documents/IUT/R5A05/webpizza/webpizza$ ls
manage.py  webpizza
(env) sgagne@2006-122:~/Documents/IUT/R5A05/webpizza/webpizza$ python3 manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
(env) sgagne@2006-122:~/Documents/IUT/R5A05/webpizza/webpizza$
```

Vous constatez que maintenant dans votre projet il y a un fichier **db.sqlite3**, c'est notre base de données. Pour le moment nous n'y avons pas créé de tables.

Vous pouvez ajouter ce fichier db.sqlite3 à .gitignore, car il n'est pas utile de le suivre dans le dépôt distant.

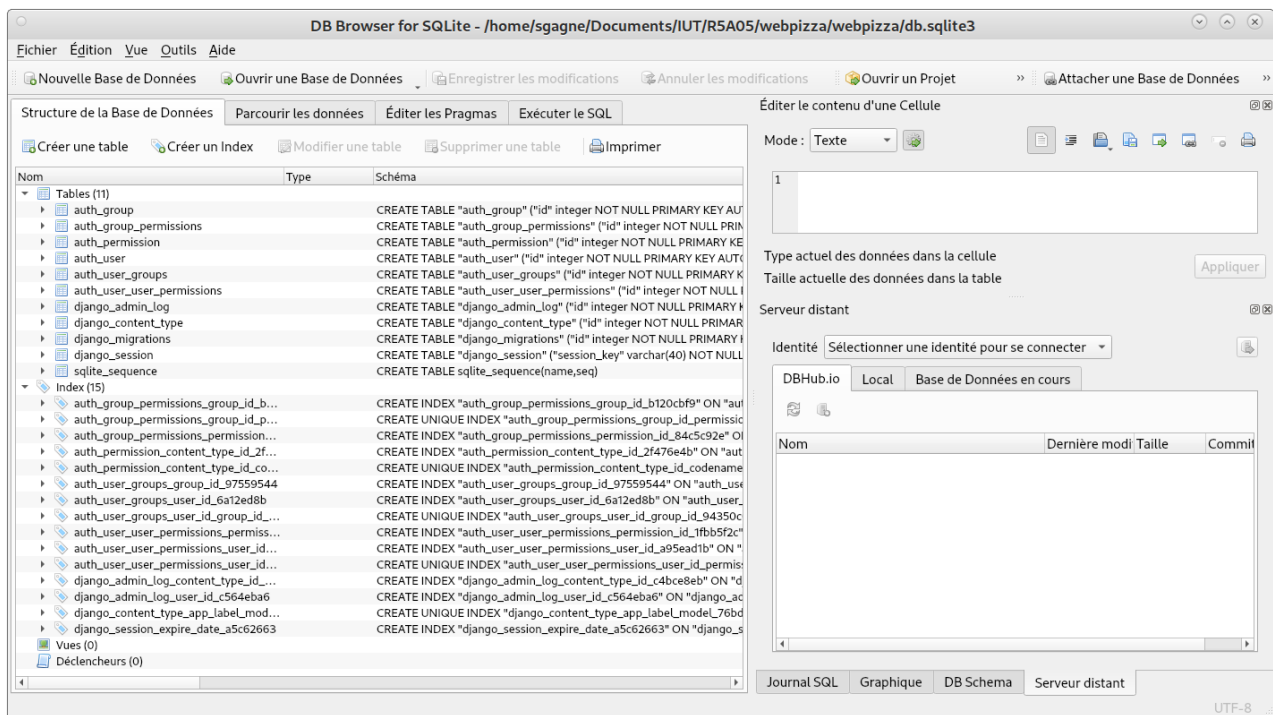


Mettez le dépôt à jour (vos commandes git doivent s'effectuer dans webpizza, pas dans webpizza/webpizza).

### 3. L'outil sqlitebrowser

Cet outil est une interface permettant de consulter/modifier la base de données du projet. Il nous servira probablement pour insérer des jeux de données.

Vérifiez que vous arrivez à ouvrir la base de données du projet avec sqlitebrowser. Vérifiez que dans la liste de tables, il n'y a encore rien concernant le projet, et que pour le moment il n'y a que des tables structurelles, ce qui est normal car nous n'avons pas encore conçu de tables ni enregistré de données.



## Création d'une application dans le projet

Par souci de *modularité*, un projet peut contenir plusieurs « applications ».

Notre première application dans ce projet sera appelée applipizza. C'est dans cette application que l'utilisateur gèrera les pizzas (créer, lire, mettre à jour, supprimer = CRUD).

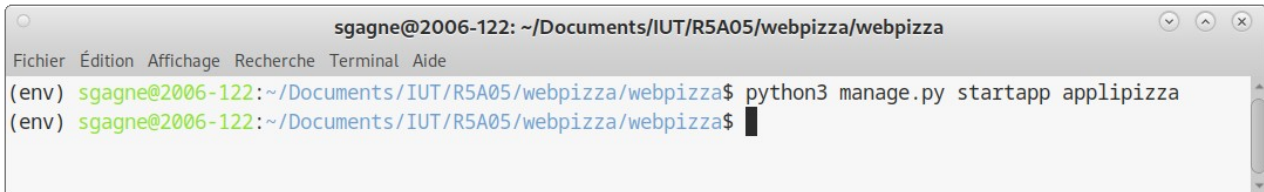
Le nom webpizza est interdit car c'est le nom du projet contenant l'application.

Il pourrait y avoir d'autres applications dans le même projet, et une application d'un projet peut être utilisée dans d'autres projets.

Par exemple, une autre application gèrera les connexions au site. Nous la coderons plus tard.

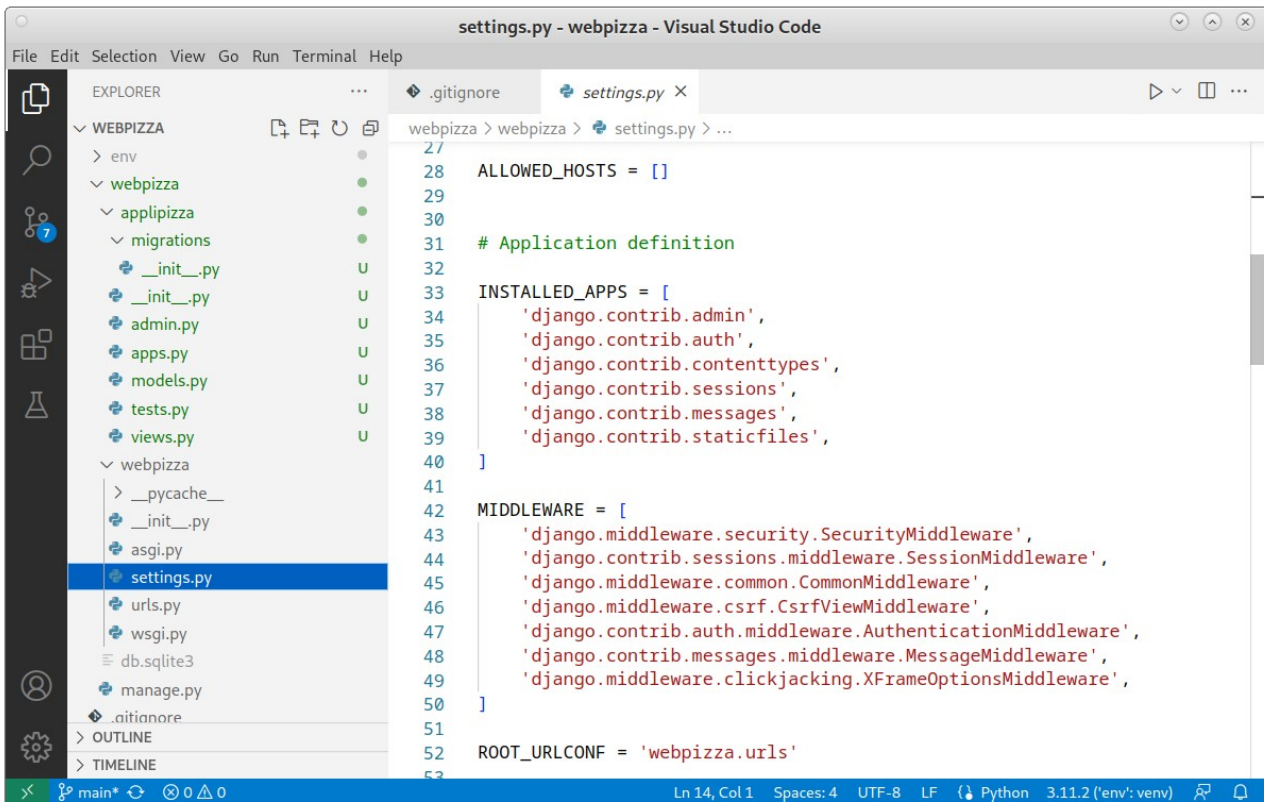
## 1. Création de l'application

- a. Lancez, dans l'environnement virtuel du projet webpizza, la commande **python3 manage.py startapp applipizza**



```
sgagne@2006-122: ~/Documents/IUT/R5A05/webpizza/webpizza
Fichier Édition Affichage Recherche Terminal Aide
(env) sgagne@2006-122:~/Documents/IUT/R5A05/webpizza/webpizza$ python3 manage.py startapp applipizza
(env) sgagne@2006-122:~/Documents/IUT/R5A05/webpizza/webpizza$
```

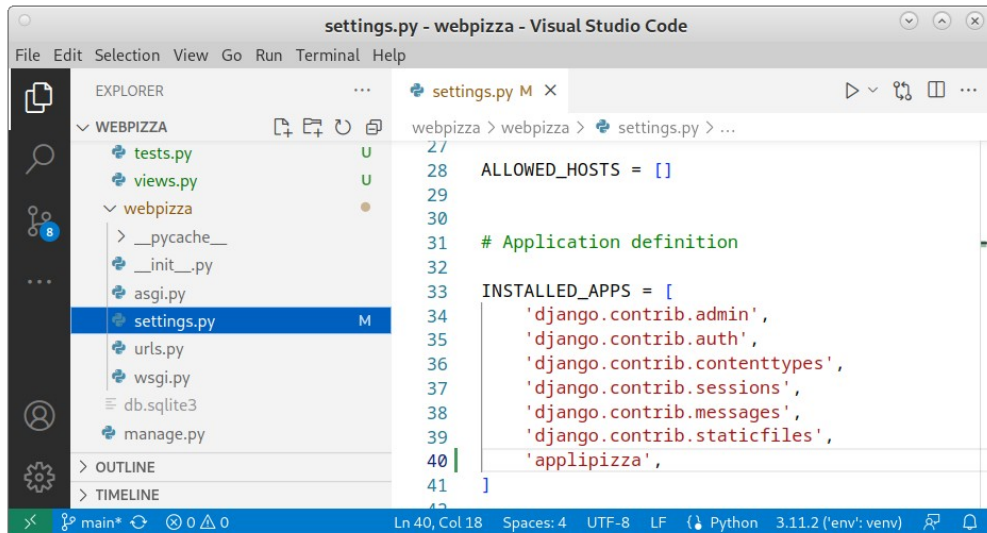
- b. On constate immédiatement l'effet produit par la création de l'application au niveau de l'architecture du projet. Dans webpizza (où on retrouve les fichiers essentiels au projet webpizza, comme settings.py), il y a maintenant un répertoire applipizza.



```
settings.py - webpizza - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
  WEBPIZZA
    > env
    > webpizza
      > migrations
        __init__.py
        __init__.py
        admin.py
        apps.py
        models.py
        tests.py
        views.py
      > webpizza
        > __pycache__
        __init__.py
        asgi.py
        settings.py
        urls.py
        wsgi.py
        db.sqlite3
        manage.py
        .aitanore
    > OUTLINE
    > TIMELINE
.gitignore settings.py
webpizza > webpizza > settings.py > ...
27
28 ALLOWED_HOSTS = []
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40 ]
41
42 MIDDLEWARE = [
43     'django.middleware.security.SecurityMiddleware',
44     'django.contrib.sessions.middleware.SessionMiddleware',
45     'django.middleware.common.CommonMiddleware',
46     'django.middleware.csrf.CsrfViewMiddleware',
47     'django.contrib.auth.middleware.AuthenticationMiddleware',
48     'django.contrib.messages.middleware.MessageMiddleware',
49     'django.middleware.clickjacking.XFrameOptionsMiddleware',
50 ]
51
52 ROOT_URLCONF = 'webpizza.urls'
53
```

c. déclaration de l'application dans les paramètres du projet

Dans le fichiers settings.py (voir capture précédente), il y a la liste python appelée **INSTALLED\_APPS**. Il faut ajouter, en fin de la liste, l'application nouvellement créée :



d. N'oubliez pas de mettre à jour votre dépôt distant, et vérifiez sur la page web du gitlab que le dépôt distant est bien à jour.

## Lancement du serveur web

Nous allons maintenant observer le site créé

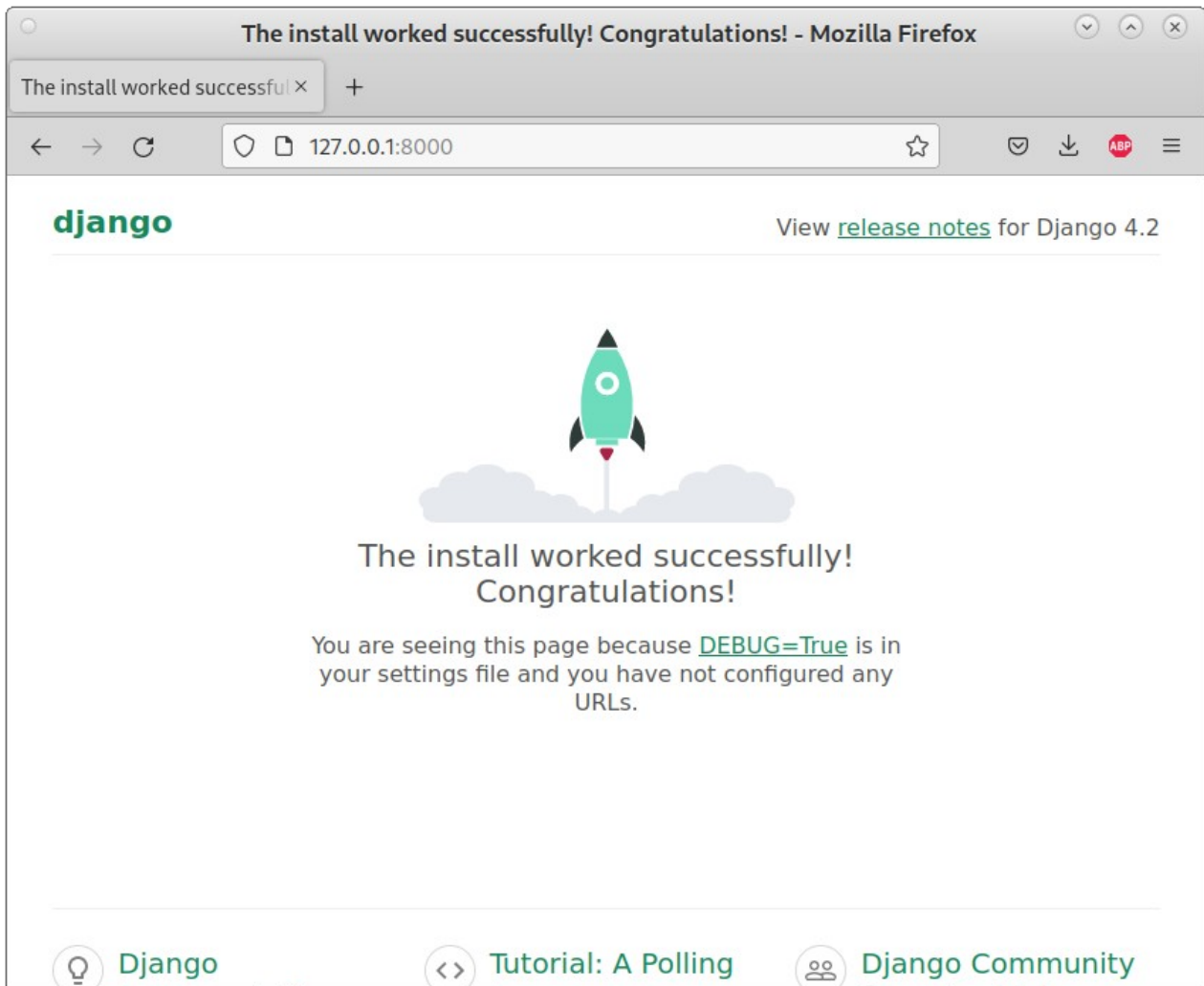
1. La commande **python3 manage.py runserver**

Lancez la commande **python3 manage.py runserver**



2. Dans le navigateur

Aller à l'url <http://127.0.0.1:8000/> pour voir le site actuel



Pour le moment c'est très pauvre et anonyme, mais si vous obtenez ceci, c'est bon signe pour la suite.