

Accueil > Cours > Débutez avec le framework Django > Installez Django avec pip

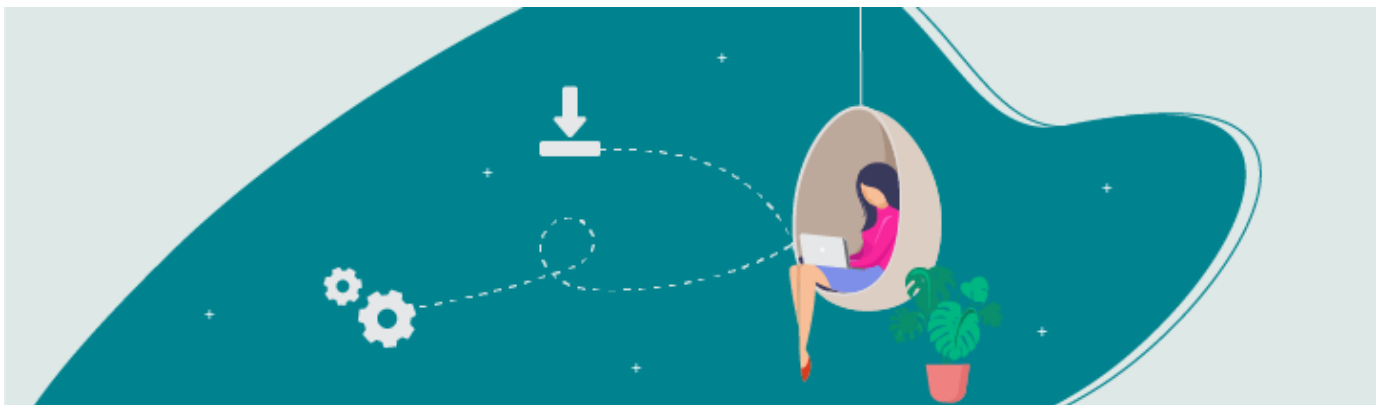
Débutez avec le framework Django

🕒 12 heures 📊 Moyenne

Mis à jour le 26/10/2022



Installez Django avec pip



Pour construire une application web Django, nous avons d'abord besoin de son code source. Comme la plupart des frameworks Python bien connus, Django est disponible pour être installé en tant que package via `pip`.

Reportez-vous à **Tirez le maximum de ce cours**, pour savoir comment trouver le terminal sous différents systèmes d'exploitation, et pour savoir comment suivre ces exemples.

Étape 1 : Commencez par ouvrir un terminal.

text

```
1 ~
2 →
```

Étape 2 : Ensuite, naviguez vers le répertoire où vous conservez généralement vos projets. Si vous ne disposez pas d'un tel répertoire, c'est le moment de le créer ! Celui-ci sera différent selon chaque personne : cela peut par exemple être `~/projets/`, `~/dev/`, `~/code/`, c'est vraiment à vous de voir.

En ce qui me concerne, ce sera :

text

```
1 ~
2 → cd projects/
3 ~/projects
4 →
```

Étape 3 : Une fois que c'est bon, créons un répertoire pour notre première application web Django, puis naviguons dedans :

text

```
1 ~/projects
2 → mkdir django-web-app
3 ~/projects
4 → cd django-web-app
5 ~/projects/django-web-app
```

Si vous suivez ce projet dans Git (et [je vous recommande](#) vivement de le faire !), c'est là que nous initialiserons notre repository.

text

```
1 ~/projects/django-web-app
2 → git init
3 Initialized empty Git repository in /Users/yourname/projects/django-web-app/.git/
```

Étape 4 : Ensuite, nous allons créer un environnement virtuel, afin de garder les packages de notre projet isolés des autres projets.

Python est installé d'office sur les ordinateurs Apple et Linux, néanmoins il est possible qu'il s'agisse de la version 2. Dans ce cas, vous devrez installer la version 3 pour travailler avec Django et indiquer la commande `python3` et non pas juste `python` , si les deux versions sont présentes sur votre ordinateur.

C'est la première fois que nous utilisons la commande `python` . Avant de faire cela, assurez-vous que votre version de Python est l'une de celles prises en charge par la dernière version de Django :

text

```
1 ~/projects/django-web-app
2 → python --version
3 Python 3.9.2
```

text

```
1 ~/projects/django-web-app
2 → python -m venv env
3 ~/projects/django-web-app
4 → ls
5 env
```

En utilisant la commande `ls` , nous pouvons voir que l'environnement virtuel a été créé dans notre répertoire `django-web-app` .

Nous n'incluons pas les paquets installés dans notre repository, ajoutez donc `env/` à votre `.gitignore` si vous utilisez Git.

Étape 5 : Maintenant, activons notre environnement virtuel :

text

```
1 ~/projects/django-web-app
2 → source env/bin/activate
3 (env) ~/projects/django-web-app
```

Le `(env)` au début de notre chemin nous indique que notre environnement virtuel est maintenant activé.

Vous devez toujours avoir votre environnement virtuel activé pendant le développement. Si vous redémarrez votre machine et revenez ensuite au projet, reproduisez l'étape ci-dessus.

Maintenant, lorsque nous exécutons un `pip install` , le package ne sera pas installé dans l'environnement global : à la place, il sera installé dans notre environnement virtuel.

Étape 6 : Et avec ça, utilisons `pip` pour installer Django :

text

```
1 (env) ~/projects/django-web-app
2
3 → pip install django
4 ...
5 Successfully installed django-3.1.7
```

`pip` installera la dernière version disponible de Django, vous pouvez donc voir un numéro de version différent de celui indiqué ci-dessus.

Le package Django est installé, ainsi qu'un certain nombre de **dépendances** : des packages dont Django a besoin pour fonctionner.

Gardons la trace de tous ces packages dans un fichier `requirements.txt`. Il existe un raccourci très pratique pour faire cela :

text

```
1 → pip freeze > requirements.txt
```

Le fichier est créé avec nos dépendances listées :

```
1 # ~/projects/django-web-app/requirements.txt
2
3 asgiref==3.3.1
4 Django==3.1.7
5 pytz==2021.1
6 sqlparse==0.4.1
```

Si vous rencontrez des problèmes avec l'un des éléments de la section ci-dessus, consultez le screencast pour vérifier que votre configuration est correcte.

Une fois Django installé, nous sommes prêts à commencer !

J'ai terminé ce chapitre et je passe au suivant

Et si vous obteniez un diplôme OpenClassrooms ?

- Formations jusqu'à 100 % financées
- Date de début flexible
- Projets professionnalisants
- Mentorat individuel

Trouvez la formation et le financement faits pour vous

[Être orienté](#)[Comparer nos types de formations](#)[Tirez le maximum de ce cours](#)[Configurez un nouveau projet avec
l'utilitaire de ligne de commande de
Django](#)

Les professeurs

Patrick Wampé

Développeur full stack et Data Scientist. Formateur dans plusieurs écoles d'informatique, il a également écrit un livre sur l'IA.

Patrick Heneghan

Software engineer in the UK, coding mostly in Python on backend systems.

Rafiq Hilali

British Software Engineer and Django expert with Lambert Labs. Currently based in BC, Canada.

[OPENCLASSROOMS](#)[OPPORTUNITÉS](#)[AIDE](#)[POUR LES ENTREPRISES](#)[EN PLUS](#)[Français](#)

