

Django est un framework web "full-stack" de haut niveau, surnommé le "Framework des perfectionnistes avec des délais" car il permet de construire des applications web robustes, sécurisées et maintenables très rapidement.

Instagram, Pinterest, National Geographic, YouTube, NASA website, Dropbox, Google, Uber etc. sont des entreprises qui utilisent Django pour une partie ou l'intégralité de leur infrastructure, ce qui témoigne de sa fiabilité et de son évolutivité.

- La maîtrise de Django est une compétence très recherchée par les employeurs pour les postes de développeur backend ou full-stack.
- Connaissant Django vous avez acquis la base parfaite pour ajouter des API RESTful robustes via Django Rest Framework (DRF).

Les 4 raisons pour se former à Django :

1. L'Écosystème Python

Python est l'un des langages de programmation les plus populaires, polyvalents et faciles à apprendre. Django est construit en ce langage.

Se former à Django c'est :

- Bénéficier de la lisibilité et de la simplicité du code Python, ce qui rend le développement plus agréable et les projets plus faciles à maintenir.
- Pouvoir utiliser les compétences Python acquises pour d'autres domaines (Data Science, Machine Learning, Automatisation, etc).

2. Sécurité Intégrée

Réputé pour son accent mis sur la sécurité, Django gère pour vous une grande partie des vulnérabilités courantes du web, ce qui est un atout majeur :

- Protection contre les menaces : Il intègre des mécanismes pour se prémunir contre les attaques XSS (Cross-Site Scripting),
- CSRF (Cross-Site Request Forgery), et les injections SQL (grâce à l'ORM).
- Système d'Authentification : Un système complet de gestion des utilisateurs, des sessions et des permissions est fourni dès l'installation, sécurisant l'accès aux données.

3. Batteries Included" (Tout-en-un)

Django est un framework complet. Contrairement aux frameworks minimalistes, il fournit l'essentiel pour construire une application web moderne sans avoir à chercher et intégrer des dizaines de bibliothèques tierces :

- ORM (Object-Relational Mapping) : Il permet d'interagir avec la base de données en utilisant du code Python (au lieu d'écrire du SQL complexe), ce qui simplifie la gestion des données.
- Interface d'Administration Automatique : Il génère automatiquement une interface d'administration professionnelle pour gérer le contenu et les données de l'application, un énorme gain de temps.
- Système de Templating : Un système puissant pour générer le code HTML de manière dynamique.

4. Développement rapide

Django fournit des conventions et des outils pour structurer les projets de manière logique (architecture MVT : Modèle-Vue-Template), ce qui facilite la collaboration et la maintenance.

Il gère de nombreuses tâches répétitives (URL routing, formulaires, etc.), vous permettant de vous concentrer sur la logique métier unique de votre application.

Contenu de la formation Développement Web avec Django



Public concerné	Développeurs web, Chefs de projet.
Objectifs pédagogiques	Savoir interfacer Django avec une base de données Développer des vues dynamiques Développer des templates Gérer les formulaires Django Mettre en œuvre les principales fonctions avancées
Pré requis	Connaissances des bases du langage Python et technologies Web (HTML, CSS, JavaScript...)
Durée	4 jours
Prix HT	2400 €
Référence	DJDEV

Contenu du cours

1 Introduction

Échafaudage d'un projet et d'une application Django

Modèle

Vue

Template

Introduction à HTTP

Traitement d'une requête

Exercice

Préparation de l'environnement. Installation de Python, de Django avec le serveur web, d'un SGBDR (PostgreSQL, MySQL, ou autre) et d'un IDE. Création d'un nouveau projet.

Création de l'application Bookr de la formation

2 L'accès aux données avec Django

Le modèle ORM Django.

Les modèles Django : Model, Field, Table, Column, Primary Key...

Mapping des associations : OneToMany, ManyToOne, ManyToMany, OneToOne.

Stratégies de mapping de l'héritage : Meta Class, OneToOneField, ManyToManyField...

Accès aux données avec les QuerySets.

Exercice

Opérations CRUD de la base de données de Django

3 Mappage d'URL, vues et modèles

Vues basées sur les fonctions

Vues basées sur les classes

Configuration d'URL

Templates

Héritage de template

Exercice

Développer des vues dynamiques HTML avec Django.

4. Introduction à Django Admin

Création d'un compte superutilisateur

Opérations CRUD avec l'application Django Admin (Create, Retriew, Update, Delete)

Utilisateurs et groupes

Exercice

Gestion d'utilisateurs et de groupes via l'application Admin

CRUD via l'application Admin

5 Gestion des formulaires

Les avantages des formulaires Django.

Création des formulaires.

Les différents types de champs.

Validation des données d'un formulaire.

Gestion et personnalisation des messages d'erreurs.

Génération des formulaires à partir des modèles.

Exercice

Développer les formulaires associés aux opérations CRUD du l'application Bookr.

6. Validation avancée des formulaires et modèles de formulaires

Validation et nettoyage des champs personnalisés

Validateurs personnalisés

Méthodes de nettoyage

Validation multi-champs

La classe ModelForm

Exercice

Création de formulaire avancée avec l'appli Bookr

7. Sessions et authentification

Implémentation des vues et des modèles d'authentification

Stockage des mots de passe dans Django

La page de profil et `Objet request.user`

Décorateurs d'authentification et redirection

Enrichissement des modèles avec des données d'authentification

Le moteur de session

Avez-vous besoin de signaler le contenu des cookies ?

Stockage des données dans les sessions

Exercice

Ajout d'une page de profil

Ajout de décorateurs d'authentification aux vues, activation/désactivation des liens de connexion et de déconnexion dans le modèle de base

TRAVAUX PRATIQUES

L'alternance d'exposés théoriques et de mises en application immédiates à travers de nombreux travaux pratiques donnent une pédagogie efficace.

Les méthodes et bonnes pratiques pour le développement de site et application web avec Django.

Les retours d'expérience de spécialistes de Django.