

Webentwicklung mit Python und Django

Webanwendungen mit Django entwickeln

01 INTRO

Django bietet einen vollständigen Werkzeugkasten für die Entwicklung datenbankgestützter Webanwendungen mit Python. Routing, Templates, Datenbankzugriff, Formulare, Authentifizierung und Administration sind bereits Bestandteil des Frameworks und lassen sich zu strukturierten Anwendungen verbinden.

Der Kurs zeigt die Entwicklung einer Django-Anwendung vom ersten Projekt bis zu einer vollständigen Webanwendung. Neben den einzelnen Framework-Komponenten stehen die Struktur einer Django-Anwendung, die Trennung von Verantwortlichkeiten, Sicherheit, Tests und die Vorbereitung für den produktiven Betrieb im Mittelpunkt.

DAUER

5 Tage

FORMAT

Präsenz · Live Online · Inhouse

VORKENNTNISSE

Sichere Python-Grundkenntnisse, grundlegende Kenntnisse von HTML und Webtechnologien

02 TRAININGSPROGRAMM

Inhalte

Grundlagen von Webanwendungen

- Client und Server
- HTTP Request und Response
- URLs und HTTP-Methoden
- HTML, CSS und JavaScript im Zusammenspiel
- statische und dynamische Inhalte
- serverseitige Webanwendungen
- Sessions und Cookies
- grundlegende Webarchitektur

Einstieg in Django

- Django installieren
- Projekt und virtuelle Umgebung
- Django-Projekt erzeugen
- Entwicklungsserver
- Aufbau eines Django-Projekts
- Settings
- Apps
- Django-Konventionen
- erste Anwendung

URLs und Views

- URL-Konfiguration
- URL Patterns
- Views
- Request und Response
- funktionsbasierte Views
- Class-Based Views im Überblick
- URL-Parameter
- Redirects
- Fehlerseiten

Templates

- Django Template Language
- Variablen und Ausdrücke
- Bedingungen und Schleifen
- Template-Vererbung
- Includes
- Filter
- eigene Template Tags im Überblick
- wiederverwendbare Layouts
- Trennung von Darstellung und Programmlogik

Models und Datenbanken

- Django ORM
- Models definieren
- Felder und Datentypen
- Migrationen
- Datensätze erzeugen, lesen, ändern und löschen
- QuerySets
- Filterung und Sortierung
- Beziehungen zwischen Models
- One-to-Many
- Many-to-Many
- Datenbankzugriffe sinnvoll strukturieren

Django Admin

- Admin-Site
- Models registrieren
- Listen und Detailansichten konfigurieren
- Filter und Suche
- Formulare anpassen
- eigene Admin-Aktionen
- Berechtigungen
- Admin als internes Werkzeug nutzen

Formulare und Benutzereingaben

- HTML-Formulare
- Django Forms
- ModelForms
- Validierung
- Fehlermeldungen
- Daten speichern
- File Uploads
- CSRF-Schutz
- Post/Redirect/Get
- sichere Verarbeitung von Eingaben

Authentifizierung und Benutzer

- Django Authentication Framework
- Benutzer
- Login und Logout
- Passwörter
- Zugriff auf Views beschränken
- Gruppen und Berechtigungen
- eigene Benutzerprofile
- Custom User Models
- Authentifizierung frühzeitig richtig planen

Sessions und Zustand

- zustandsloses HTTP
- Cookies
- Sessions
- Session-Daten speichern
- Lebensdauer von Sessions
- benutzerspezifischer Zustand
- typische Einsatzgebiete
- Sicherheitsaspekte

Statische Dateien und Medien

- CSS und JavaScript
- Static Files
- Media Files
- Uploads
- Entwicklungs- und Produktionsumgebung
- Dateien sicher bereitstellen
- typische Projektstruktur

Struktur größerer Django-Anwendungen

- Anwendungen in Apps zerlegen
- Verantwortlichkeiten von Apps
- Models, Views und Templates strukturieren
- Business-Logik nicht in Views verstecken
- Services und Hilfskomponenten
- wiederverwendbare Apps
- Konfiguration und Umgebung trennen
- wachsende Projekte beherrschbar halten

Django ORM vertiefen

- komplexere QuerySets
- Q-Objekte
- Aggregationen
- Annotationen
- Joins über Beziehungen
- select_related
- prefetch_related
- N+1-Probleme
- Datenbankzugriffe analysieren und optimieren

Webanwendungen testen

- Django Test Framework
- pytest und Django im Überblick
- Models testen
- Views testen
- Formulare testen
- Test Client
- Authentifizierung testen
- Testdaten
- Datenbanktests
- Integrationstests

Sicherheit

- typische Risiken von Webanwendungen
- Cross-Site Scripting
- CSRF
- SQL Injection
- sichere Templates
- Authentifizierung und Autorisierung
- Secrets und Konfiguration
- Security Middleware
- Django Security Features
- sichere Produktionskonfiguration

APIs in Django

- JSON Responses
- einfache API-Endpunkte
- Django REST Framework im Überblick
- Serialisierung
- REST und klassische Webanwendung kombinieren
- Abgrenzung zu FastAPI
- wann welches Framework sinnvoll ist

Performance und Caching

- typische Performance-Probleme
- Datenbankabfragen reduzieren
- Caching-Grundlagen
- Django Cache Framework
- unterschiedliche Cache Backends
- Seiten und Daten cachen
- Caching gezielt statt pauschal einsetzen

Deployment und Betrieb

- Entwicklungs- und Produktionsbetrieb
- DEBUG
- Konfiguration über Umgebungsvariablen
- WSGI und ASGI
- statische Dateien
- Datenbankmigrationen beim Deployment
- Logging
- Docker im Überblick
- grundlegende Produktionsarchitektur
- Health Checks und Betrieb

Durchgängige Beispielanwendung

- Anforderungen strukturieren
- Datenmodell entwickeln
- Django-Projekt und Apps aufbauen
- Views und Templates entwickeln
- Formulare integrieren
- Benutzer und Berechtigungen
- Datenbankzugriffe
- Tests
- Anwendung schrittweise erweitern
- für den Betrieb vorbereiten

03 DURCHFÜHRUNG

Rahmen und Organisation

Dauer	5 Tage
Durchführung	Präsenz oder Live Online, vorzugsweise als Inhouse-Schulung
Schwerpunkt	Inhouse-Schulungen für Unternehmen und Teams
Sprache	Deutsch, englische Fachbegriffe und Dokumentation
Unterlagen	Kursunterlagen, Beispiele und Übungsaufgaben
Technik	Eigener Rechner mit aktueller Python-Version und Entwicklungsumgebung

Inhouse-Schulungen

Die Schwerpunkte können an vorhandene Kenntnisse, eingesetzte Technologien und konkrete Anforderungen des Teams angepasst werden.

04 ZIELGRUPPE

Für wen ist der Kurs gedacht?

Der Kurs richtet sich an Python-Entwicklerinnen und -Entwickler, die datenbankgestützte Webanwendungen mit Django entwickeln möchten.

Er eignet sich sowohl für Teilnehmer ohne bisherige Django-Erfahrung als auch für Entwickler, die bereits kleinere Django-Anwendungen erstellt haben und das Framework systematischer kennenlernen möchten.

Die Teilnehmer sollten mit folgenden Themen sicher umgehen können:

- grundlegende Python-Syntax
- Funktionen

- Module und Packages
- Klassen und Objekte
- Exceptions
- virtuelle Python-Umgebungen

Webtechnologien

Grundkenntnisse in HTML sowie ein grundlegendes Verständnis von Webanwendungen sind hilfreich. Tiefere JavaScript- oder CSS-Kenntnisse werden nicht vorausgesetzt.

05 LERNZIELE

Was Sie nach dem Kurs können

Nach dem Kurs können die Teilnehmer:

- Django-Projekte und Apps selbstständig aufbauen
- URLs, Views und Templates zu Webanwendungen verbinden
- Datenmodelle mit dem Django ORM entwickeln
- Datenbankmigrationen durchführen
- Formulare und Benutzereingaben sicher verarbeiten
- Django Admin für interne Verwaltungsoberflächen konfigurieren
- Benutzer, Authentifizierung und Berechtigungen integrieren
- Sessions und benutzerspezifischen Zustand verwalten
- größere Django-Anwendungen sinnvoll strukturieren
- Datenbankzugriffe analysieren und optimieren
- Django-Anwendungen automatisiert testen
- wesentliche Sicherheitsmechanismen von Django einsetzen
- Caching und grundlegende Performance-Optimierungen nutzen
- Django-Anwendungen für den produktiven Betrieb vorbereiten

06 FAQ

Häufige Fragen

Muss ich Django bereits kennen?

Nein. Der Kurs beginnt mit dem Aufbau eines Django-Projekts und entwickelt daraus schrittweise eine vollständige Webanwendung.

Muss ich HTML, CSS und JavaScript beherrschen?

Grundkenntnisse in HTML sind hilfreich. CSS und JavaScript werden dort verwendet, wo sie für die Anwendung benötigt werden, sind aber nicht Schwerpunkt des Kurses.

Wird auch das Django ORM behandelt?

Ja. Das ORM ist ein zentraler Bestandteil des Kurses – von einfachen CRUD-Operationen bis zu Beziehungen, komplexeren QuerySets und der Vermeidung unnötiger Datenbankzugriffe.

Wird Django REST Framework behandelt?

Django REST Framework wird eingeordnet und anhand grundlegender Konzepte betrachtet. Der Schwerpunkt des Kurses liegt jedoch auf Django als vollständigem Webframework.

Was ist der Unterschied zu PY-006 „REST APIs mit Python und FastAPI“?

PY-006 konzentriert sich auf die Entwicklung von APIs und Backend-Services mit FastAPI. PY-011 behandelt vollständige serverseitige Webanwendungen mit Django einschließlich Templates, Formularen, Datenbankmodell, Benutzerverwaltung und Administration.

Werden Tests behandelt?

Ja. Models, Views, Formulare und wesentliche Anwendungsabläufe werden automatisiert getestet.

Wird Deployment behandelt?

Ja, soweit es für das Verständnis einer produktionsfähigen Django-Anwendung erforderlich ist. Der Aufbau einer vollständigen Betriebs- oder Cloud-Infrastruktur ist jedoch nicht Schwerpunkt des Kurses.

Kann der Kurs an eine vorhandene Django-Anwendung angepasst werden?

Ja. Bei Inhouse-Schulungen können Architektur, Datenmodell, Authentifizierung und andere Schwerpunkte an vorhandene Anwendungen und Anforderungen angepasst werden.

UC IT Service · Ulrich Cuber · kontakt@uc-it.de

<https://www.uc-it.de/coaching/python-django/>