

## Links

**Projektseite** [www.djangoproject.com](http://www.djangoproject.com)

**Deutsche Seite** [www.django-de.org](http://www.django-de.org)

**E-Book** [www.djangobook.com](http://www.djangobook.com)

## Was ist Django?

Django ist ein in Python geschriebenes Framework, das die schnelle Entwicklung von **Web-Applikationen** ermöglicht. Dabei wird besonders viel Wert auf sauberen Code und die **Wiederverwendbarkeit** einzelner Komponenten gelegt.

Django wird als Open Source Software unter einer **BSD-Lizenz** verteilt. Django bietet umfangreiche Unterstützung für mehrsprachige Anwendungen und verschiedene Datenbanken.

## Verantwortlicher Herausgeber

**Deutscher Django-Verein e.V.**

Erich-Zeigner-Allee 52

04229 Leipzig

Der Verein wird vertreten durch den Vorstand  
Markus Zapke-Gründemann, Martin Mahner  
und Florian Schlachter.

**Vereinsregister** Amtsgericht Leipzig

**Registernummer** VR 4814

Python Web Framework  
**django**

## Kontakt

**Fon** +49 341 39 29 92 90

**Fax** +49 32 12 100 74 39

**kontakt@django-de.org**

**Bürozeiten** Montag bis Freitag 16 bis 18 Uhr

Deutscher Django-Verein e.V.

## Open Source

Django ist Open Source (BSD-Lizenz). Man kann als Entwickler also einerseits in den **Quellcode** schauen, um die Funktionalität zu verstehen, andererseits auch an der Weiterentwicklung von Django teilnehmen. Die Benutzung von Django ist **kostenlos**.

## Dokumentation

Django besitzt eine hervorragende (englische) Dokumentation mit vielen nützlichen Beispielen. Eine **deutsche Version** ist in Arbeit.

## Freundliche Community

Sollten einmal Probleme auftreten, bei denen auch die Dokumentation nicht mehr weiterhilft, kann man sich über die **Mailingliste** (django-users) oder den **IRC-Channel** (#django auf Freenode) Rat von der Django-Community holen.

## Django Foundation

Die Entwicklung von Django wird durch die in den USA ansässige Django Foundation beobachtet und gesteuert. Damit sind sowohl die **Weiterentwicklung** der Software als auch der Schutz vor einem Missbrauch des Namens sichergestellt.

## Don't Repeat Yourself

Django ist nach dem DRY-Prinzip programmiert und organisiert. Dieses besagt, dass jede Wissenseinheit nur eine **eindeutige** und **schlüssig** positionierte Repräsentation im Code haben darf und Redundanz vermieden wird. Nach diesem Prinzip sollten auch die mit Django entwickelten Projekte erstellt werden.

## Mtv – ein Mvc-Muster

Django trennt – wie durch das MVC-Muster vorgegeben – Datenrepräsentation (Model), Darstellung (View) und Logik (Controller) voneinander.

## Eingebautes Admintool

Das automatisch erzeugte Admin-Interface hilft bei der Erstellung und Bearbeitung der Daten. Es kann bei Bedarf **leicht angepasst** und erweitert werden.



## Elegantes URL-Design

Jede URL kann durch den Entwickler nahezu beliebig **selbst bestimmt** werden. Die URL wird beim Aufruf durch einen regulären Ausdruck mit dem entsprechenden Inhalt verknüpft.

## Object-Relational-Mapper

Der Object-Relational-Mapper (ORM) ermöglicht einen einfachen und schnellen Zugriff auf die Datenbank. Es werden **verschiedene Datenbanken** unterstützt.

```
from django.db import models

class Person(models.Model):
    first_name =
models.CharField(max_length=30)
    last_name =
models.CharField(max_length=30)
```

## Template-Engine

Die Templates werden in einer eigenen Template-sprache geschrieben. Sie bringt bereits viele nützliche Helfer zum Erstellen der Ausgabe mit. Die **umfangreiche Tag-Bibliothek** kann leicht um eigene Funktionen erweitert werden.

```
<p>Dear {{ person_name }},</p>

<p>Here are the items you've
ordered:</p>
<ul>
    {% for item in item_list %}
        <li>{{ item }}</li>
    {% endfor %}
</ul>

{% if ordered_warranty %}
    <p>Your warranty information will
be included in the packaging.</p>
{% endif %}
```