



Grundlagen der Web-Entwicklung

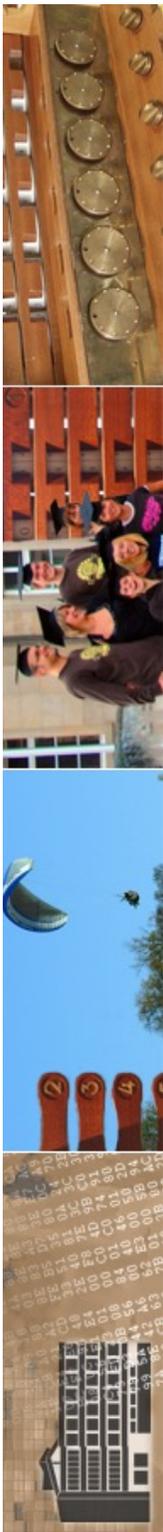
INF3172

Das Python-Framework DJANGO

Thomas Walter

20.01.2022

Version 1.0





Webframework: Django 3 bietet asynchrone Verarbeitung

Das Python-Webframework Django spricht in Version 3.0 mit MariaDB und bringt asynchrone Verarbeitung dank ASGI.

Lesezeit: 2 Min.  In Pocket speichern

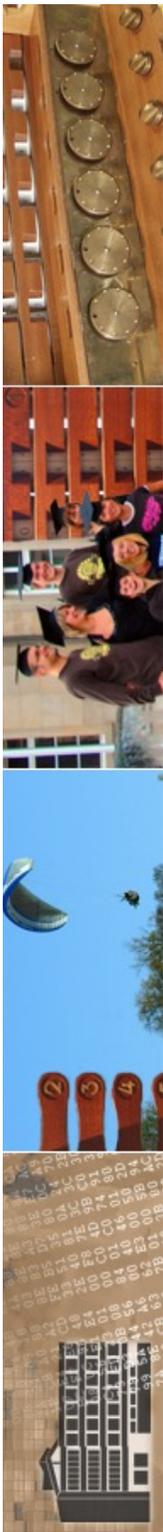


Jazz-Gitarrist Django Reinhardt, Namensgeber für das Django-Framework (Bild: Royal Opera House Covent Garden, Flickr)

03.12.2019 09:44 Uhr | Developer

Von Rainald Menge-Sonntag

Das in Python entwickelte Webframework Django ist in Version 3.0 erschienen. Die wichtigsten Neuerungen sind die Einführung asynchroner Funktionen und eine Anbindung an die Open-Source-Datenbank MariaDB. Außerdem können Entwickler für Text-, ganzzahlige und andere Felder Enumerations als Auswahl für die Inhalte definieren. Version 3.0 ist im Gegensatz zum Vorgänger Django 2.2 kein LTS-Release (Long-term Support).



Webframework: Django-Nutzer sind updatefreudig und setzen auf Bootstrap



Eine internationale Umfrage unter mehr als 7000 Django-Entwicklern zeigt, dass sie in neuen Projekten meist das aktuelle stabile Release verwenden.

Lesezeit: 3 Min.  In Pocket speichern



(Bild: Shutterstock/chanpipat)

04.01.2022 16:13 Uhr | Developer

Von Maika Möbus

Die Django Software Foundation hat in Zusammenarbeit mit dem Unternehmen JetBrains den "Django Developers Survey 2021" durchgeführt. Mehr als 7000 Personen nahmen im August 2021 an der Umfrage zu dem in Python geschriebenen Open-Source-Webframework Django teil.





Webframework: Django 4 wechselt die Zeitzone

Neben dem Wechsel auf die Zeitzonen-Implementierung der Standard-Library von Python bringt die neue Hauptversion eine Anbindung an Redis als Cache.

Lesezeit: 3 Min.  In Pocket speichern



(Bild: William P. Gottlieb Collection des Library of Congress, gemeinfrei)

09.12.2021 11:00 Uhr | Developer

Von Rainald Menge-Sonnentag

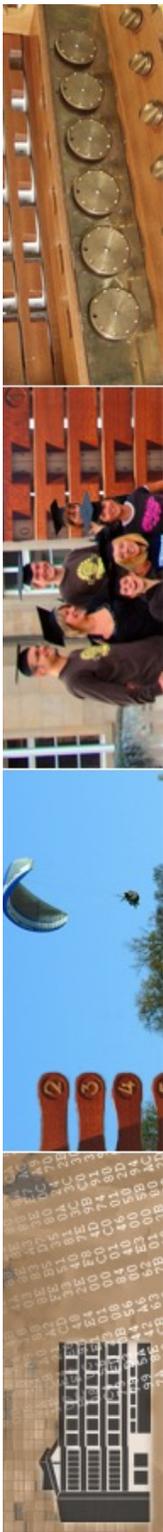
Das in Python entwickelte Open-Source-Webframework Django ist in Version 4.0 erschienen. Das Release hat ein neues Backend zum Caching mit Redis an Bord. Außerdem gibt es Änderungen bei der Zeitzonen-Implementierung und beim Rendern von Forms und Formsets.





Frameworks

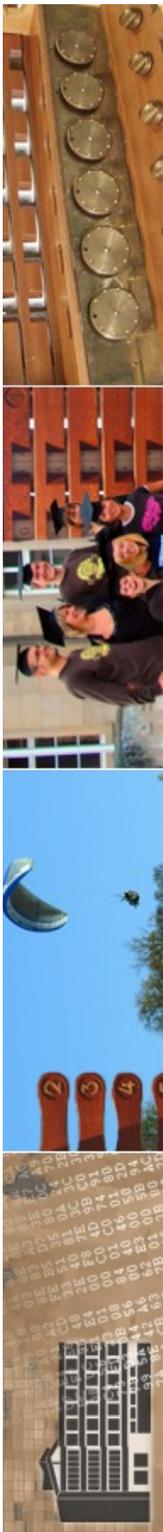
- Framework: wiederverwertbares Softwaresystem mit bereits implementierter, genereller (generischer) Funktionalität
 - Spezialisierung führt zu konkreter Anwendung
 - Framework setzt Architektur um
 - Framework folgt Design Pattern
 - Beispiele
 - Ruby: Ruby on Rails
 - PHP: Laminas, Cake-PHP, Symfony, FLOW3
 - Python: Django





wo wir stehen

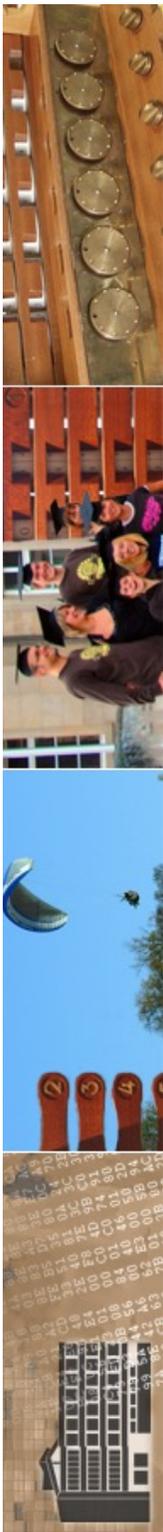
- Überblick
 - Smarty
 - Framework(le)
 - Abspaltung der View
 - eigene Sprache (STL)
 - Laminas
 - Trennung View und Controller
 - PHP
 - jetzt: DJANGO
 - danach: CakePHP und Ruby on Rails

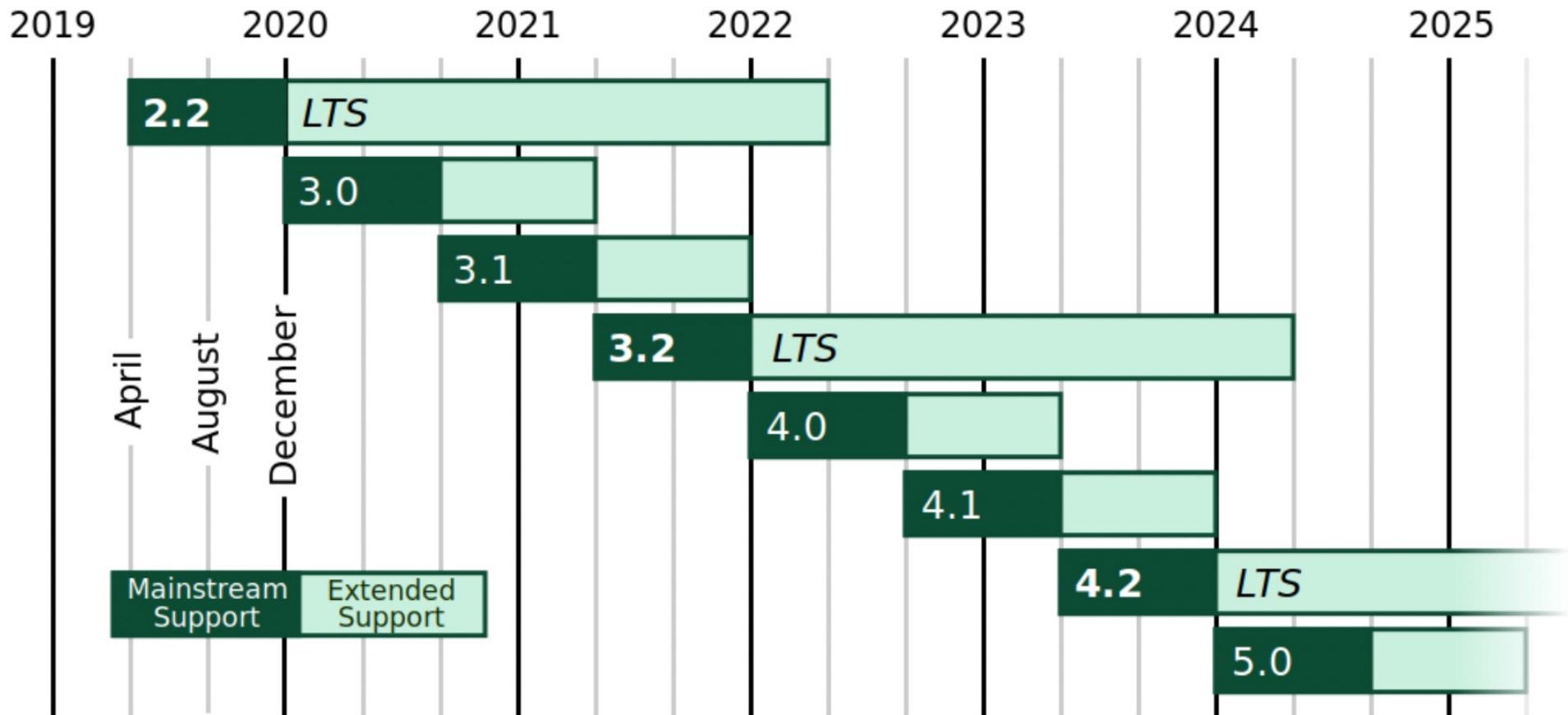




DJANGO

- serverseitiges Framework
- Basis: Python
(vgl. Grundlagen Internet-Technologien)
- Orientierung am MVC
- erstes Release 2005
- aktuell: 4.0.1
 - DJANGO 2/3 benötigen PYTHON3
(1.11 lauffähig mit Python 2.7)





| Release Series | Latest Release | End of mainstream support ¹ | End of extended support ² |
|----------------|----------------|--|--------------------------------------|
| 4.0 | 4.0.1 | August 2022 | April 2023 |
| 3.2 LTS | 3.2.11 | December 2021 | April 2024 |
| 2.2 LTS | 2.2.26 | December 2, 2019 | April 2022 |





How to get Django

Django is available open-source under the [BSD license](#). We recommend using the latest version of Python 3. The last version to support Python 2.7 is Django 1.11 LTS. See [the FAQ](#) for the Python versions supported by each version of Django. Here's how to get it:

Option 1: Get the latest official version

The latest official version is 4.0.1. Read the [4.0.1 release notes](#), then install it with `pip`:

```
pip install Django==4.0.1
```

Option 2: Get the latest development version

The latest and greatest Django version is the one that's in our Git repository (our revision-control system). This is only for experienced users who want to try incoming changes and help identify bugs before an official release. Get it using this shell command, which requires [Git](#):

```
git clone https://github.com/django/django.git
```

You can also download a [gzipped tarball](#) of the development version. This archive is updated every time we commit code.

After you get it

See the [installation guide](#) for further instructions. Make sure you read the documentation that corresponds to the version of Django you've just installed.

And be sure to sign up for the [django-users mailing list](#), where other Django users and the Django developers themselves all hang out to help each other.



Support Django!



Jeremias Kangas donated to the Django Software Foundation to support Django development. Donate today!

For the impatient:

- Latest release: [Django-4.0.1.tar.gz](#)
Checksums: [Django-4.0.1.checksum.txt](#)
Release notes: [Online documentation](#)

Which version is better?

We improve Django almost every day and are pretty good keeping the code stable. Thus, using the latest development is a safe and easy way to get access to new features as they are added. If you choose to follow the development version, keep in mind that there will occasionally be backwards-incompatible changes. You'll want to pay close attention to the commit messages watching [Django on GitHub](#) or subscribing to [django-updates](#).

If you're just looking for a stable deployment target and are waiting for the next release, you'll want to stick with the latest official release (which will always include detailed notes on the changes you'll need to make while upgrading).

Previous releases

- Django 3.2.11 (LTS): [Django-3.2.11.tar.gz](#)
Checksums: [Django-3.2.11.checksum.txt](#)
Release notes: [Online documentation](#)
- Django 2.2.26 (LTS): [Django-2.2.26.tar.gz](#)
Checksums: [Django-2.2.26.checksum.txt](#)
Release notes: [Online documentation](#)

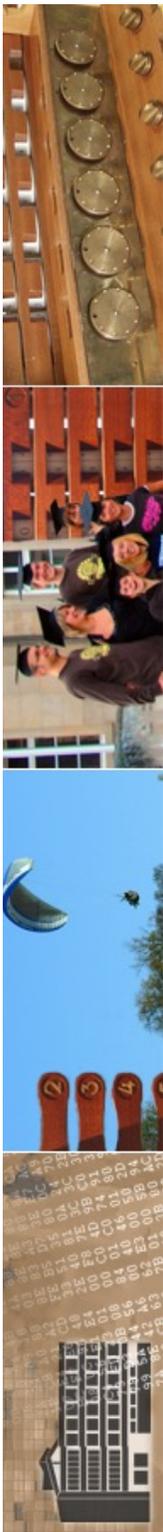




Python

- aktueller
TIOBE-
Programmer
index

| | Jan 2022 | Jan 2021 | Change | Programming Language | Ratings | Change |
|----|----------|----------|--------|--|---------|--------|
| 4 | 3 | | ↑ |  Python | 13.58% | +1.86% |
| 1 | 1 | | ↓ |  C | 12.44% | -4.94% |
| 2 | | | ↓ |  Java | 10.66% | -1.30% |
| 4 | 4 | | |  C++ | 8.29% | +0.73% |
| 5 | 5 | | |  C# | 5.68% | +1.73% |
| 6 | 6 | | |  Visual Basic | 4.74% | +0.90% |
| 7 | 7 | | |  JavaScript | 2.09% | -0.11% |
| 8 | 11 | | ↑ |  Assembly language | 1.85% | +0.21% |
| 9 | 12 | | ↑ |  SQL | 1.80% | +0.19% |
| 10 | 13 | | ↑ |  Swift | 1.41% | -0.02% |
| 11 | 8 | | ↓ |  PHP | 1.40% | -0.60% |
| 12 | 9 | | ↓ |  R | 1.25% | -0.65% |
| 13 | 14 | | ↑ |  Go | 1.04% | -0.37% |





Django documentation

Everything you need to know about Django.

First steps

Are you new to Django or to programming? This is the place to start!

- **From scratch:** [Overview](#) | [Installation](#)
- **Tutorial:** [Part 1: Requests and responses](#) | [Part 2: Models and the admin site](#) | [Part 3: Views and templates](#) | [Part 4: Forms and generic views](#) | [Part 5: Testing](#) | [Part 6: Static files](#) | [Part 7: Customizing the admin site](#)
- **Advanced Tutorials:** [How to write reusable apps](#) | [Writing your first patch for Django](#)

Getting help

Having trouble? We'd like to help!

- Try the [FAQ](#) – it's got answers to many common questions.
- Looking for specific information? Try the [Index](#), [Module Index](#) or the [detailed table of contents](#).
- Search for information in the archives of the [django-users](#) mailing list, or [post a question](#).
- Ask a question in the [#django IRC channel](#).
- Report bugs with Django in our [ticket tracker](#).

Support Django!



Intevation GmbH donated to the Django Software Foundation to support Django development. Donate today!

Browse

- Prev: [Django documentation contents](#)
- Next: [Getting started](#)
- [Table of contents](#)
- [General Index](#)
- [Python Module Index](#)

You are here:

- [Django 3.0 documentation](#)
 - Django documentation

Getting help



woher der Name?

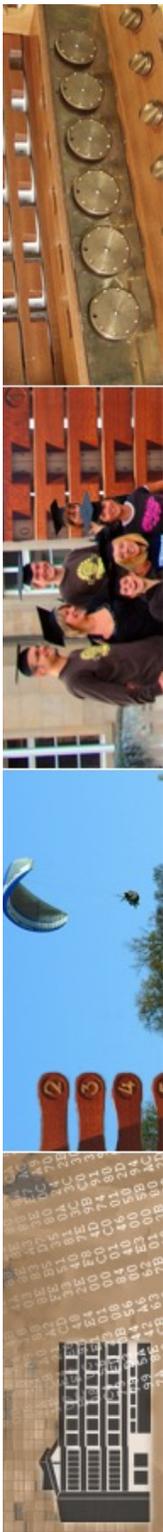
- ...nach Django Reinhardt (Jazz-Gitarist)





Installation

- einfachste Methode:
mittels PYTHON-Repository-Funktion **pip**
 - ggf. erst aktuelles pip installieren
<https://pip.pypa.io/en/latest/>
 - Alternative aus Paket-Installationen wie arp-get

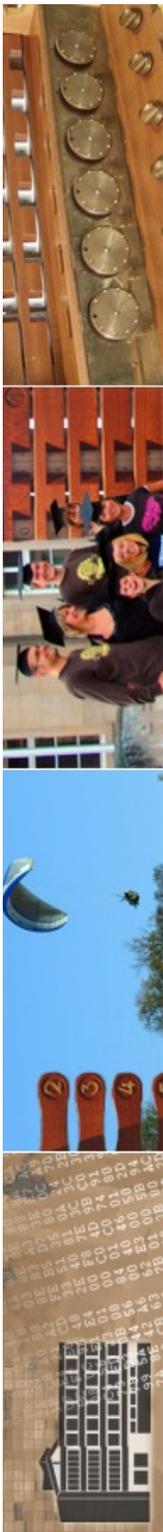




```
[thomas@vpn0245 => python3 ./get-pip.py
Collecting pip
  Downloading pip-7.1.2-py2.py3-none-any.whl (1.1MB)
    100% |#####| 1.1MB 405kB/s
Collecting wheel
  Downloading wheel-0.26.0-py2.py3-none-any.whl (63kB)
    100% |#####| 65kB 2.6MB/s
Installing collected packages: pip, wheel
  Found existing installation: pip 6.0.8
  Uninstalling pip-6.0.8:
    Successfully uninstalled pip-6.0.8
  Successfully installed pip-7.1.2 wheel-0.26.0
thomas@vpn0245 => █
```

```
thomas@guest-8d46b5c9 => pip install --upgrade pip
Collecting pip
  Downloading pip-9.0.1-py2.py3-none-any.whl (1.3MB)
    100% |#####| 1.3MB 274kB/s
Installing collected packages: pip
  Found existing installation: pip 7.1.2
  Uninstalling pip-7.1.2:
    Successfully uninstalled pip-7.1.2
  Successfully installed pip-9.0.1
thomas@guest-8d46b5c9 =>
```





```
[thomas@Vaux => python3  
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 23 2015, 02:52:03)  
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin  
Type "help", "copyright", "credits" or "license" for more information.  
[>>> import django  
    ]
```





```

[thomas@Vaux => ll
total 32
drwxr-xr-x  21 thomas  admin   714   5 Jan 11:51 ./
drwxrwxr-x  15 root     admin   510   5 Jan 11:51 ../
-rw-r--r--   1 thomas  admin   519   5 Jan 11:51 __init__.py
-rw-r--r--   1 thomas  admin   211   5 Jan 11:51 __main__.py
drwxr-xr-x   5 thomas  admin   170   5 Jan 11:51 __pycache__/
drwxr-xr-x   6 thomas  admin   204   5 Jan 11:51 apps/
drwxr-xr-x   4 thomas  admin   136   5 Jan 11:51 bin/
drwxr-xr-x   9 thomas  admin   306   5 Jan 11:51 conf/
drwxr-xr-x  20 thomas  admin   680   5 Jan 11:51 contrib/
drwxr-xr-x  20 thomas  admin   680   5 Jan 11:51 core/
drwxr-xr-x   9 thomas  admin   306   5 Jan 11:51 db/
drwxr-xr-x   8 thomas  admin   272   5 Jan 11:51 dispatch/
drwxr-xr-x  12 thomas  admin   408   5 Jan 11:51 forms/
drwxr-xr-x   9 thomas  admin   306   5 Jan 11:51 http/
drwxr-xr-x  12 thomas  admin   408   5 Jan 11:51 middleware/
-rw-r--r--   1 thomas  admin  7957   5 Jan 11:51 shortcuts.py
drwxr-xr-x  19 thomas  admin   646   5 Jan 11:51 template/
drwxr-xr-x  10 thomas  admin   340   5 Jan 11:51 templatetags/
drwxr-xr-x  10 thomas  admin   340   5 Jan 11:51 test/
drwxr-xr-x  49 thomas  admin  1666   5 Jan 11:51 utils/
drwxr-xr-x  11 thomas  admin   374   5 Jan 11:51 views/
[thomas@Vaux =>

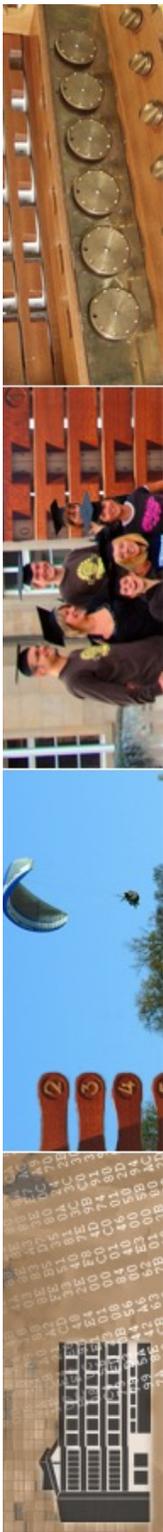
```





Erzeugen eines Projektes

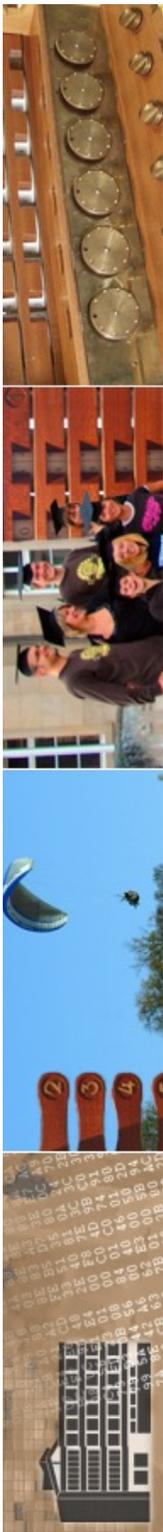
- ein neues DJANGO-Projekt wird erzeugt durch
`django-admin.py startproject`
`<name>`





DJANGO und MVC

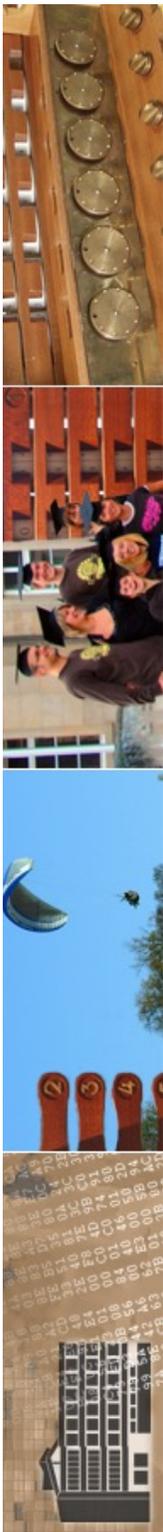
- wie viele anderen Frameworks setzt DJANGO MVC um
 - eigentlich MVP: Model-View-Presenter, das aus MVC hervorgegangen
<http://www.wildcrest.com/Potel/Portfolio/mvp.pdf>
 - DJANGO *generiert Modell direkt aus korrespondierenden Datenbanktabellen*
 - genauer bei Ruby on Rails und CakePHP
 - MVC in Verzeichnisstruktur angelegt





zentrale Dateien

- zentrale Dateien in einer DJANGO-APP
 - settings.py
allgemeine Einstellungen für das ganze Projekt
Datenbankkonnektivität
 - urls.py
Weiterleitung der Requests zu Controllern
(Dispatcher)
 - models.py
Korrespondenz zu Datenbanktabellen
 - views.py
Controller





Datenbankkonfiguration

- zentral in der Datei `settings.py`

```

2
3 DEBUG = True
4 TEMPLATE_DEBUG = DEBUG
5
6 ADMINS = (
7     'Thomas Walter', 'thomas.walter@uni-tuebingen.de'
8 )
9
10 MANAGERS = ADMINS
11
12 DATABASE_ENGINE = 'mysql'           # 'postgresql_psycopg2', 'postgresql', 'mysql', 'sqlite3' or
13 DATABASE_NAME = 'webkompendium'     # Or path to database file if using sqlite3.
14 DATABASE_USER = 'thomas'           # Not used with sqlite3.
15 DATABASE_PASSWORD = ''             # Not used with sqlite3.
16 DATABASE_HOST = ''                 # Set to empty string for localhost. Not used with sqlite3.
17 DATABASE_PORT = ''                 # Set to empty string for default. Not used with sqlite3.
18

```



settings.py

```

1  """
2  Django settings for {{ project_name }} project.
3
4  Generated by 'django-admin startproject' using Django {{ django_version }}.
5
6  For more information on this file, see
7  https://docs.djangoproject.com/en/{{ docs_version }}/topics/settings/
8
9  For the full list of settings and their values, see
10 https://docs.djangoproject.com/en/{{ docs_version }}/ref/settings/
11 """
12
13 import os
14
15 # Build paths inside the project like this: os.path.join(BASE_DIR, ...)
16 BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
17
18
19 # Quick-start development settings - unsuitable for production
20 # See https://docs.djangoproject.com/en/{{ docs_version }}/howto/deployment/checklist/
21
22 # SECURITY WARNING: keep the secret key used in production secret!
23 SECRET_KEY = '{{ secret_key }}'
24

```



View

- View in DJANGO:

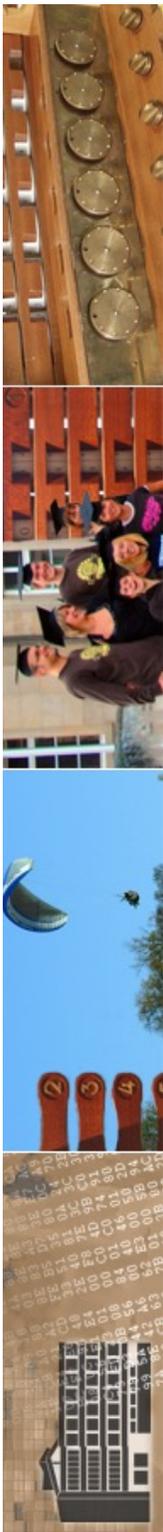
HTML mit integriertem Python-Code

- Auswertung serverseitig
- verwendet „DJANGO template language“ (Python)

- Syntax:

```
{% python_Code %}
```

```
{{ Variable }}
```



```

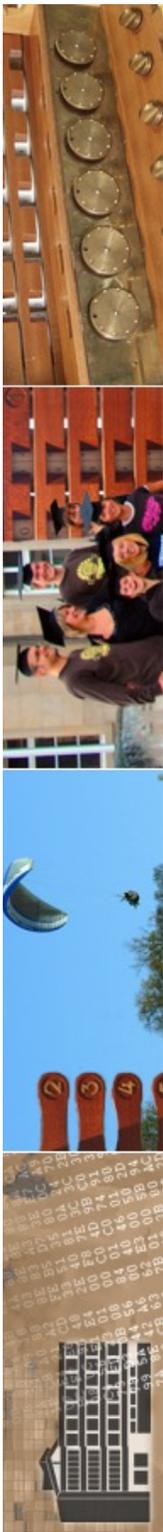
1 from django.template import Context, loader
2 from buecher.buecheranwendung.models import Buecherdjango
3 from django.http import HttpResponse
4
5 def index(request):
6     t = loader.get_template('buecheranwendung/index.html')
7     c = Context({'message' : '<h2>Hello django</h2>'})
8     return HttpResponse(t.render(c))
9
10 def info(request):
11     info = "<B>Kompendium der Web-Programmierung</B>\
12           <BR>Anwendung des django-Frameworks"
13     return HttpResponse(info)
14
15 def buecher(request):
16     #buecher_liste = Buecherdjango.objects.all().order_by('jahr')
17     #output = ""
18     #for p in buecher_liste:
19     #    output += str(p)+"<BR>"
20     #return HttpResponse(output)
21     t = loader.get_template('buecheranwendung/buecher.html')
22     buecher_liste = Buecherdjango.objects.all().order_by('-jahr')
23     c = Context({'liste' : buecher_liste})
24     return HttpResponse(t.render(c))

```



weitere Ansätze

- es gibt neben DJANGO weitere Python-basierte Web-Frameworks
- wichtig: ZOPE



The World of Zope

Documentation

Community

Developer Information

Welcome to Zope

Zope is a free and open source web application server written in the object-oriented programming language [Python](#). Since its release in 1998, Zope continues to grow into many distinct applications, frameworks, libraries and tools. [The World of Zope](#) highlights the most important components.

Zope community projects are hosted in the [Zope Foundation organization](#) on GitHub.

Documentation

An overview over the most important documentation resources.

- [Documentation](#)

Community

The Zope community is one of the largest and most professional open-source communities worldwide.

- [Community](#)
- [Developer Information](#)

Next ➔

...und nun...

- kennen wir ein modernes, typisches Framework mit seiner Arbeitsweise als konkretes Architekturbeispiel kennen gelernt
- weitere, noch komplexere Beispiele zu Frameworks: RoR und CakePHP

