



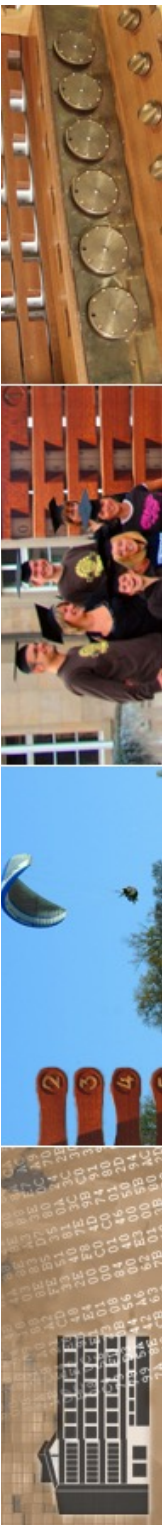
Grundlagen Internet-Technologien

INF3171

Python-Frameworks
DJANGO und andere
Thomas Walter

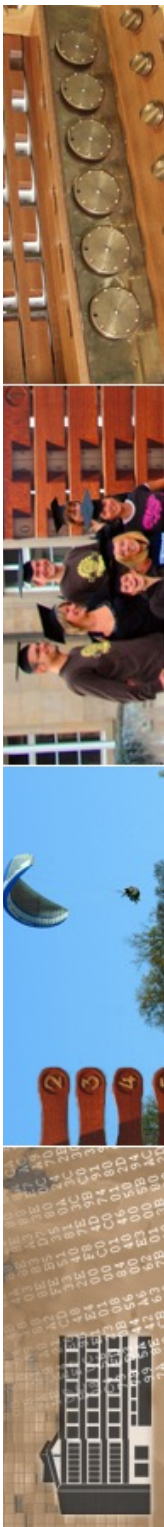
16.07.2026

Version 1.0



Klausur

- Klausur
Do 23.07.2026
08:00 - 10:00
Hörsaal N03 (Hörsaalzentrum Morgenstelle)
- Nachklausur
Fr 09.10.2026
10:00 - 12:00
Hörsaal 1, F119 (Sand)



TOUR DE FRANCE GESAMTWERUNG 2026

2026 ▾

GESAMTWE...

TEAMWER...

PUNKTEWER...

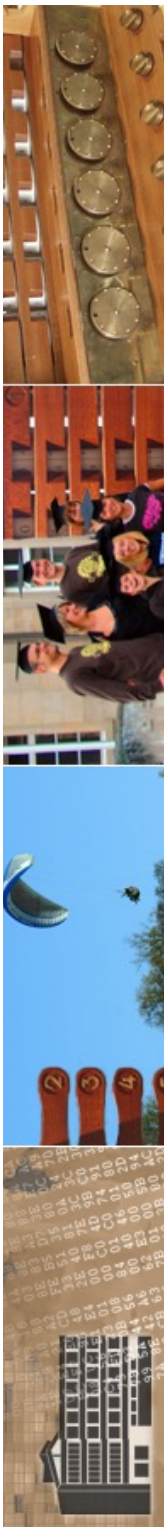
BERGWER...

NACHWUCHSWE...

AKTIVSTER F...

FAHRER		ZEIT
1	 T. Pogacar UAE Team Emirates - XRG	39:25:08
2	 J. Vingegaard Team Visma-Lease a Bike	+3:36
3	 R. Evenepoel Red Bull-Bora-Hansgrohe	+4:06
4	 J. Ayuso Lidl Trek	+4:22
5	 P. Seixas Decathlon CMA CGM	+4:35
6	 F. Lipowitz Red Bull-Bora-Hansgrohe	+4:44
7	 I. Del Toro UAE Team Emirates - XRG	+5:08
8	 M. Skjelmose Lidl Trek	+5:45
9	 L. Martinez Bahrain Victorious	+6:34
10	 T. Pidcock Pinarello - Q36.5 Pro Cycling Team	+11:49
11	 E. Bernal Netcompany INEOS	+12:15



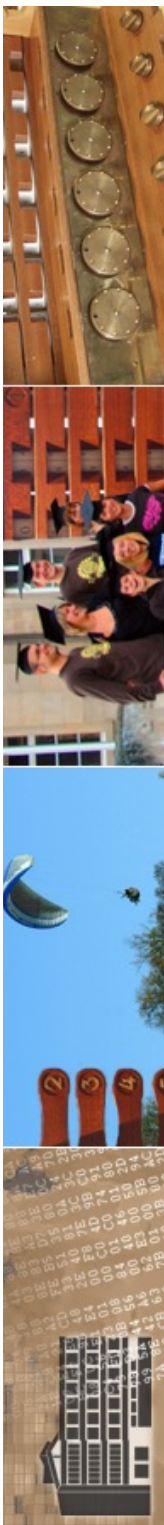


Jan 2026	Jan 2025	Change	Programming Language		Ratings	Change
1	1			Python	22.61%	-0.68%
2	4	▲		C	10.99%	+2.13%
3	3			Java	8.71%	-1.44%
4	2	▼		C++	8.67%	-1.62%
5	5			C#	7.39%	+2.94%
6	6			JavaScript	3.03%	-1.17%
7	9	▲		Visual Basic	2.41%	+0.04%
8	8			SQL	2.27%	-0.14%
9	11	▲		Delphi/Object Pascal	1.98%	+0.19%
10	18	▲▲		R	1.82%	+0.81%
11	32	▲▲		Perl	1.63%	+1.14%
12	10	▼		Fortran	1.61%	-0.42%
13	14	▲		Rust	1.51%	+0.34%
14	15	▲		MATLAB	1.40%	+0.34%
15	13	▼		PHP	1.38%	-0.00%
16	7	▼▼		Go	1.24%	-1.37%



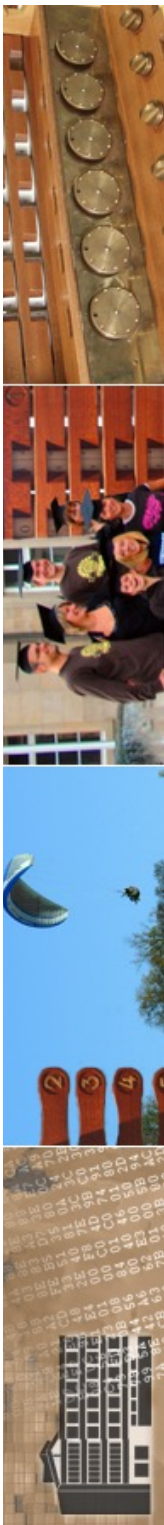
Frameworks

- 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: ZEND Framework, Cake-PHP, Symfony, FLOW3, Laravel
 - Python: Django





Das Framework FastAPI



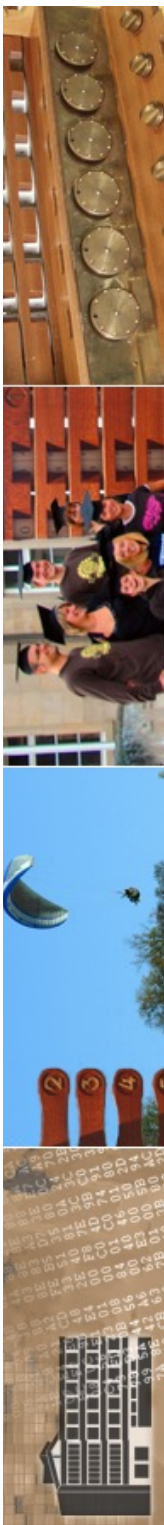
FastAPI

- FastAPI framework, high performance, easy to learn, fast to code, ready for production

FastAPI is a modern, fast (high-performance), web framework for building APIs with Python based on standard Python type hints.

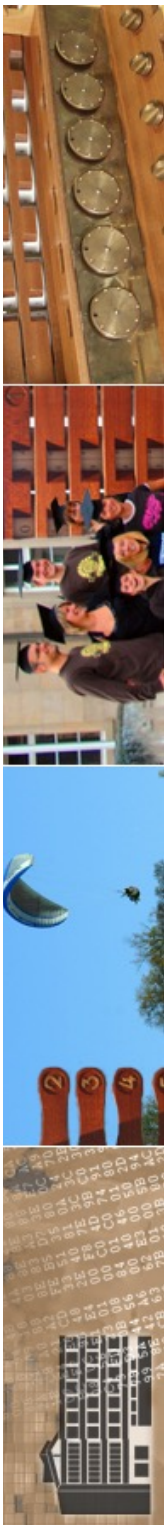
The key features are:

- **Fast:** Very high performance, on par with **NodeJS** and **Go** (thanks to Starlette and Pydantic).
One of the fastest Python frameworks available.
- **Fast to code:** Increase the speed to develop features by about 200% to 300%. *
- **Fewer bugs:** Reduce about 40% of human (developer) induced errors. *
- **Intuitive:** Great editor support. Completion everywhere. Less time debugging.
- **Easy:** Designed to be easy to use and learn. Less time reading docs.
- **Short:** Minimize code duplication. Multiple features from each parameter declaration. Fewer bugs.
- **Robust:** Get production-ready code. With automatic interactive documentation.
- **Standards-based:** Based on (and fully compatible with) the open standards for APIs:
OpenAPI [C] (previously known as Swagger) and JSON Schema [C].



FastAPI

- Veröffentlicht 2018 von Sebastián Ramírez
- wichtiges Ziel: einfaches Erstellen von API-Endpunkten
- darüber hinaus typisches Webframework (mit Einschränkungen)



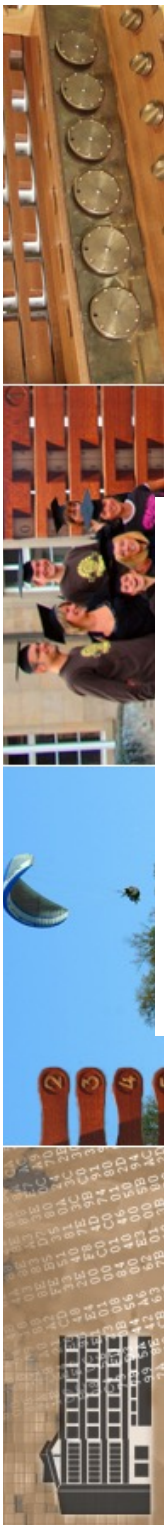


Flask



Flask

Welcome to Flask's documentation. Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications.



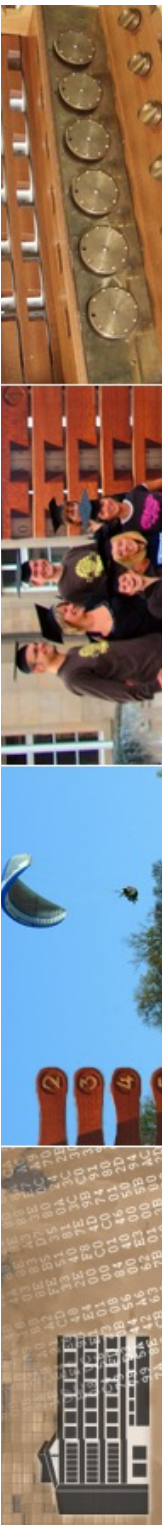
Flask

- schlankes Framework („Micro-Framework“) in Python für Webanwendungen
 - Welcome to Flask’s documentation. Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications.

Install Flask

Within the activated environment, use the following command to install Flask:

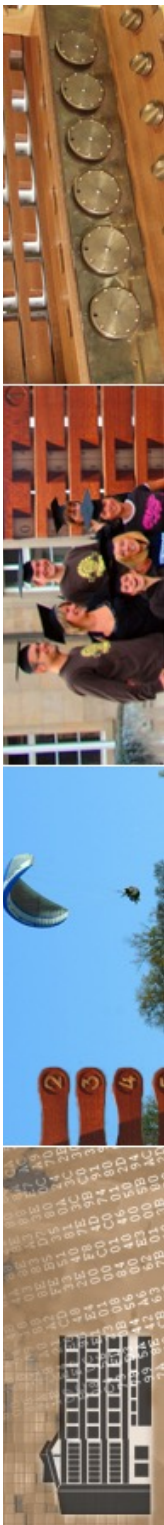
```
$ pip install Flask
```

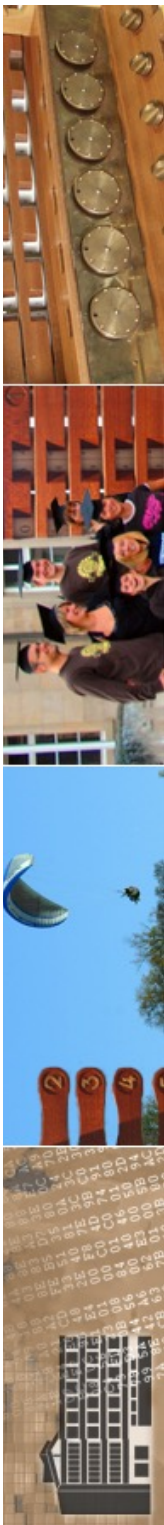




weitere Ansätze

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





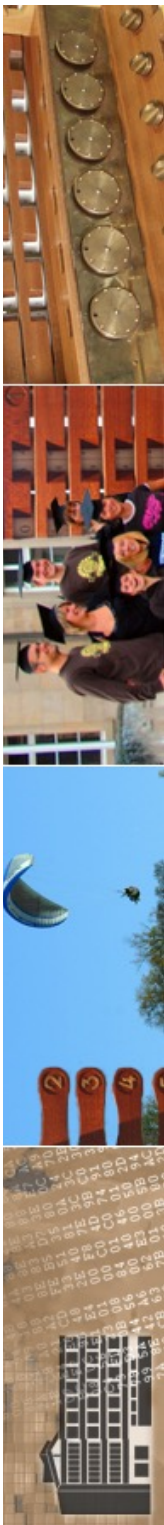
© 2026 Universität
Tübingen





django





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

   9



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

03.12.2019 09:44 Uhr | Developer

Von Rainald Menge-Sonnentag

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 4 wechselt die Zeitzone

Neben dem Wechsel auf die Zeitzones-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)

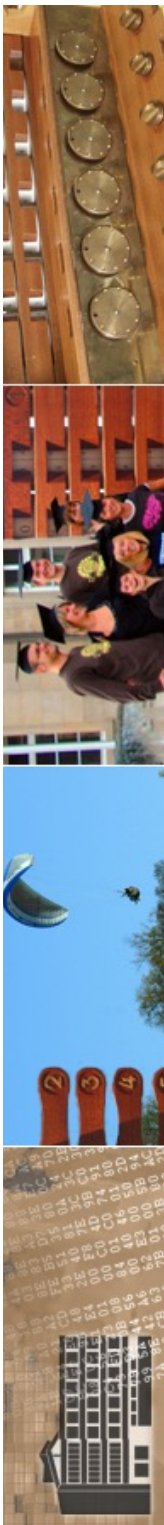
09.12.2021 11:00 Uhr | Developer

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 Zeitzones-Implementierung und beim Rendern von Forms und Formsets.

Frameworks

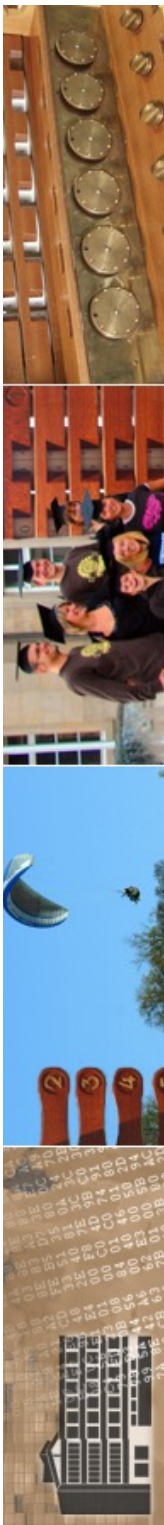
- Framework: wiederverwertbares Softwaresystem mit bereits implementierter, genereller (generischer) Funktionalität
 - Spezialisierung führt zu konkreter Anwendung
 - Framework setzt Architektur um
 - Framework folgt Design Pattern
 - Beispiele serverseitig
 - Ruby: Ruby on Rails
 - PHP: Laminas, Cake-PHP,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





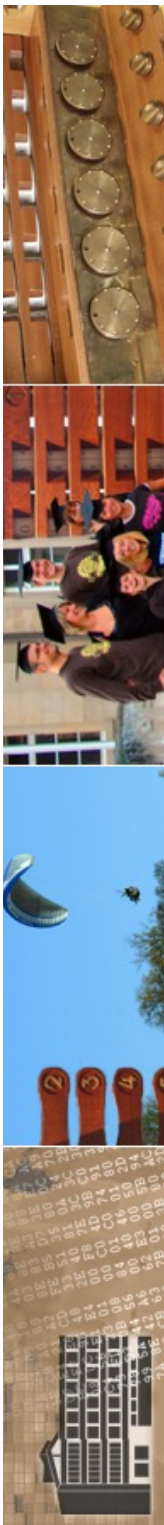
Django makes it easier to build better web apps more quickly and with less code.

Get started with Django



Frameworks – kleiner Überblick

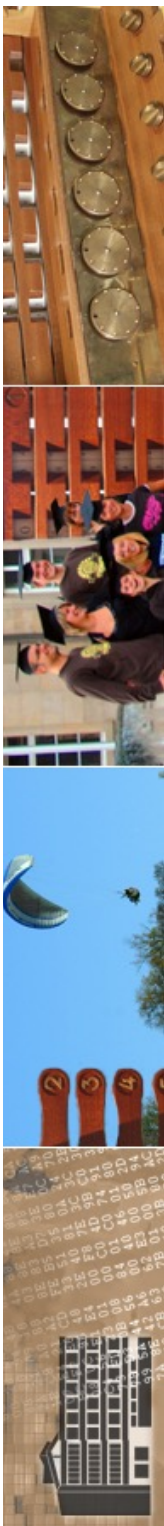
- Überblick
 - Smarty
 - Framework(le)
 - Abspaltung der View
 - eigene Sprache (STL)
 - Laminas
 - Trennung View und Controller
 - PHP
 - jetzt: DJANGO
 - danach: CakePHP und Ruby on Rails





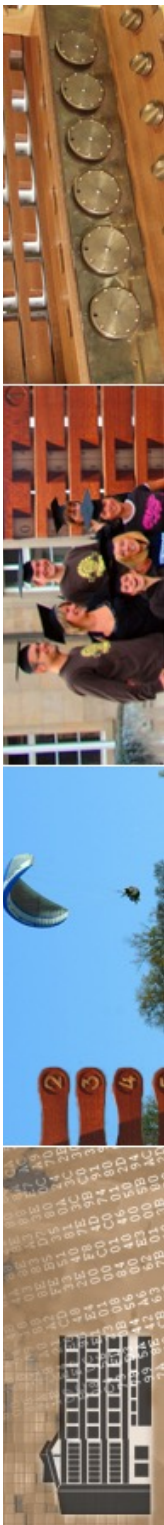
Frameworks (serverseitig) und MVC

	V	C	M
Smarty	X		
Laminas (ZF)	X	X	
R o R	X	X	X
Cake PHP	X	X	X



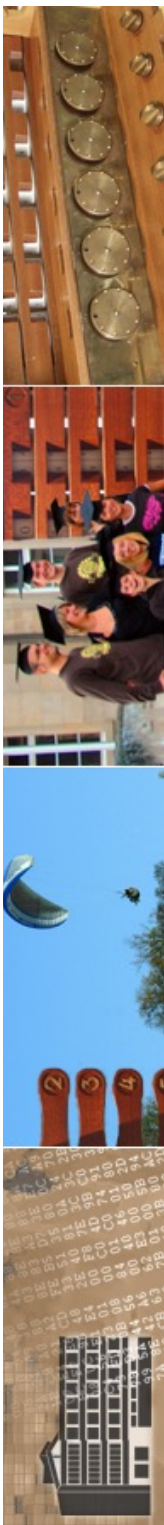
DJANGO

- serverseitiges Framework
- Basis: Python
(vgl. Grundlagen Internet-Technologien)
- Orientierung am MVC
- erstes Release 2005
- aktuell: 6.0.7
 - DJANGO 2/3/4/5/6 benötigen PYTHON3
(1.11 lauffähig mit Python 2.7)



What Python version can I use with Django? 🍀

Django version	Python versions
4.2	3.8, 3.9, 3.10, 3.11, 3.12 (added in 4.2.8)
5.0	3.10, 3.11, 3.12
5.1	3.10, 3.11, 3.12, 3.13 (added in 5.1.3)
5.2	3.10, 3.11, 3.12, 3.13, 3.14 (added in 5.2.8)
6.0	3.12, 3.13, 3.14



Django 6.0.1 release notes 🐘

January 6, 2026

Django 6.0.1 fixes one data loss bug in Django 5.2 and several bugs in 6.0.

Bugfixes 🐘

- Fixed a bug in Django 5.2 where data exceeding **max_length** was silently truncated by **QuerySet.bulk_create()** on PostgreSQL ([#33647](#)).
- Fixed a regression in Django 6.0 where **querystring** mishandled multi-value **QueryDict** keys, both by only preserving the last value and by incorrectly handling **None** values ([#36783](#)).
- Fixed a regression in Django 6.0 that prevented changing the name of a **ManyToManyField** from taking effect when applying migrations ([#36800](#)).
- Fixed a bug where management command colored help (introduced in Python 3.14) ignored the **--no-color** option and the **DJANGO_COLORS** setting ([#36376](#)).
- Fixed a regression in Django 6.0 that caused **bulk_create()** to crash when introspecting the connection on SQLite ([#36818](#)).
- Fixed a visual regression in Django 6.0 for admin form fields grouped under a **<fieldset>** in Safari ([#36807](#)).
- Fixed a crash in Django 6.0 caused by infinite recursion when calling **repr()** on an unevaluated **django.utils.csp.LazyNonce** instance ([#36810](#)).
- Fixed a regression in Django 6.0 where **path()** routes defined using **gettext_lazy()** failed to resolve correctly ([#36796](#)).
- Fixed a regression in Django 6.0 where the **Widget.use_fieldset** attribute of **ClearableFileInput** was flipped from **False** to **True** ([#36829](#)).
- Reverted an undocumented optimization in Django 6.0 that modified permission **name** and **codename** values when renaming models via a migration. This change could affect unrelated **Permission** objects ([#36843](#)) and did not report conflicts ([#36793](#)).



Documentation

Django 6.0.7 release notes ¶

July 7, 2026

Django 6.0.7 fixes three security issues with severity “low” and one bug in 6.0.6.

CVE-2026-48588: Potential exposure of private data via cached **Set-Cookie** response ¶

`UpdateCacheMiddleware` and `cache_page()` avoided caching responses that set a cookie while varying on **Cookie** only when the incoming request contained no cookies at all. When the request already carried an unrelated cookie (such as a language or theme preference cookie), the protection did not apply, allowing a response that sets a session or other sensitive cookie to be stored in Django’s shared cache.

This issue has severity “low” according to the [Django security policy](#).

CVE-2026-53877: Heap buffer over-read in **GDALRaster** ¶

When **GDALRaster** was instantiated with a bytes object representing a raster file, the `vsi_buffer` property could over-read the allocated buffer by approximately 32 bytes. This could result in information disclosure of adjacent heap memory or, in rare cases, a segmentation fault. Only rasters stored in GDAL’s virtual filesystem were affected.

This issue has severity “low” according to the [Django security policy](#).

CVE-2026-53878: Header injection possibility since **DomainNameValidator** accepted newlines in input ¶

DomainNameValidator accepted newlines in domain names. If such values were included in HTTP responses, header injection attacks were possible. Django itself wasn’t vulnerable because **HttpResponse** prohibits newlines in HTTP headers.

The vulnerability only affected uses of **DomainNameValidator** outside Django form fields, as **CharField** strips newlines by default.

This issue has severity “low” according to the [Django security policy](#).

Bugfixes ¶

- Fixed a regression in Django 6.0 where the PBKDF2 and MD5 password hashers raised **UnicodeDecodeError** for **bytes** passwords that were not valid UTF-8. Passwords supplied as **str** or as UTF-8 **bytes** are unaffected ([#37184](#)).

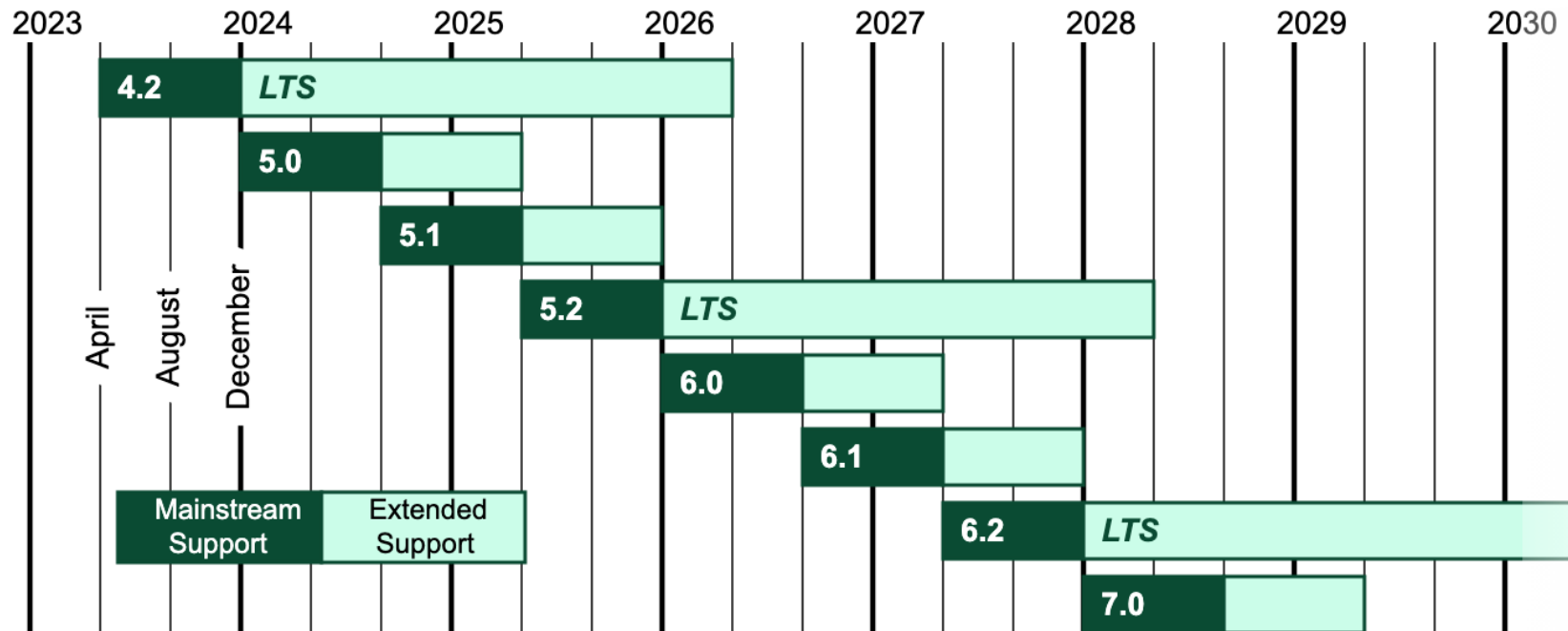
[← Django 6.0.8 release notes](#)

[Django 6.0.6 release notes →](#)





Supported Versions



Release Series	Latest Release	End of mainstream support ¹	End of extended support ²
4.2 LTS	4.2.27	December 4, 2023	April 2026
5.2 LTS	5.2.10	December 3, 2025	April 2028
6.0	6.0.1	August 2026	April 2027





Future Roadmap

Release Series	Release Date	End of mainstream support ¹	End of extended support ²
6.1	August 2026	April 2027	December 2027
6.2 LTS	April 2027	December 2027	April 2030
7.0	December 2027	August 2028	April 2029
7.1	August 2028	April 2029	December 2029



How to get Django

Django is available open-source under the [BSD license](#). We recommend using the latest version of Python. 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 6.0.1. Read the [6.0.1 release notes](#), then install it with [pip](#):

Linux / macOS:

```
python -m pip install Django==6.0.1
```

Windows:

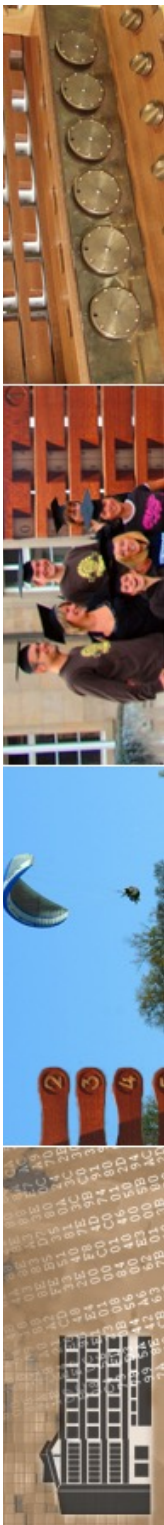
```
py -m pip install Django==6.0.1
```

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.



Python



- aktueller
TIOBE-
Programmer
index

Jan 2026	Jan 2025	Change	Programming Language		Ratings	Change
1	1			Python	22.61%	-0.68%
2	4	▲		C	10.99%	+2.13%
3	3			Java	8.71%	-1.44%
4	2	▼		C++	8.67%	-1.62%
5	5			C#	7.39%	+2.94%
6	6			JavaScript	3.03%	-1.17%
7	9	▲		Visual Basic	2.41%	+0.04%
8	8			SQL	2.27%	-0.14%
9	11	▲		Delphi/Object Pascal	1.98%	+0.19%
10	18	▲▲		R	1.82%	+0.81%
11	32	▲▲		Perl	1.63%	+1.14%
12	10	▼		Fortran	1.61%	-0.42%
13	14	▲		Rust	1.51%	+0.34%
14	15	▲		MATLAB	1.40%	+0.34%
15	13	▼		PHP	1.38%	-0.00%
16	7	▼▼		Go	1.24%	-1.37%





The web framework for
perfectionists with deadlines.

OVERVIEW DOWNLOAD DOCUMENTATION NEWS COMMUNITY CODE ISSUES ABOUT ♥ DONATE ⓘ

Documentation

Search 4.1 documentation



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](#).
- Not found anything? See [FAQ: Getting Help](#) for information on getting support and asking questions to the community.
- Report bugs with Django in our [ticket tracker](#).

Support Django!



Nick Efford 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 4.1 documentation](#)
 - Django documentation

Getting help



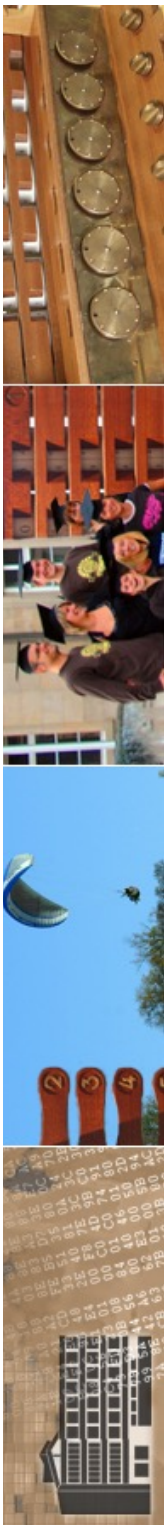
woher der Name?

- ...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 =>
```



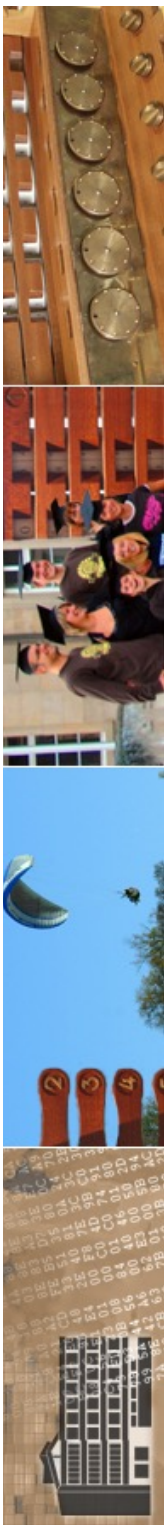


```
[thomas@Vaux => python3  
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 23 2015, 02:52:03)  
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin  
Type "help", "copyright", "credits" or "license" for more information.  
[>>> import django  
□
```





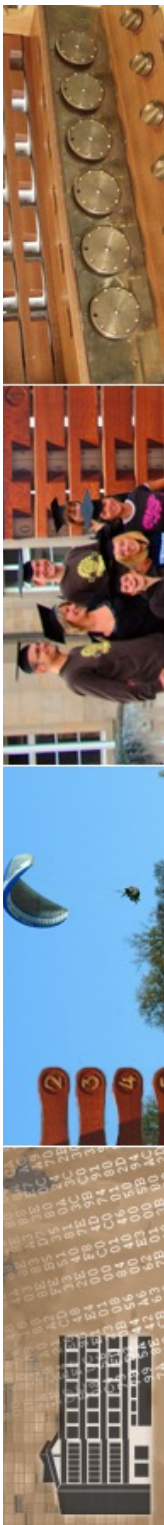
```
[thomas@Vaux => ll
total 32
drwxr-xr-x  21 thomas  admin   714  5 Jan 11:51 ./
drwxrwxr-x  15 root    admin   510  5 Jan 11:51 ../
-rw-r--r--   1 thomas  admin   519  5 Jan 11:51 __init__.py
-rw-r--r--   1 thomas  admin   211  5 Jan 11:51 __main__.py
drwxr-xr-x   5 thomas  admin   170  5 Jan 11:51 __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

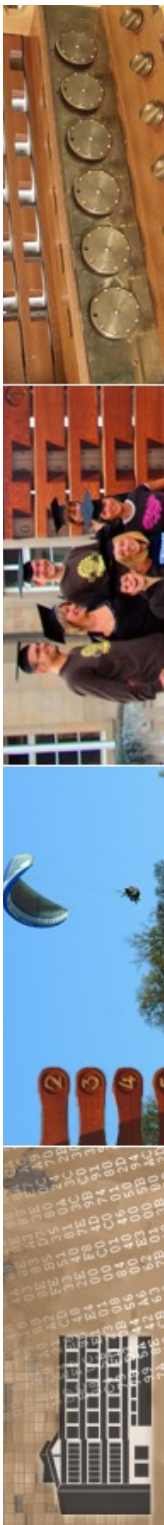
- ein neues DJANGO-Projekt wird erzeugt durch

`django-admin.py startproject`
`<name>`



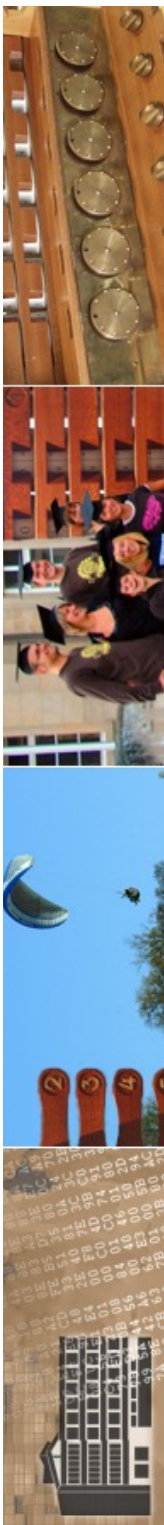
DJANGO und MVC

- 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

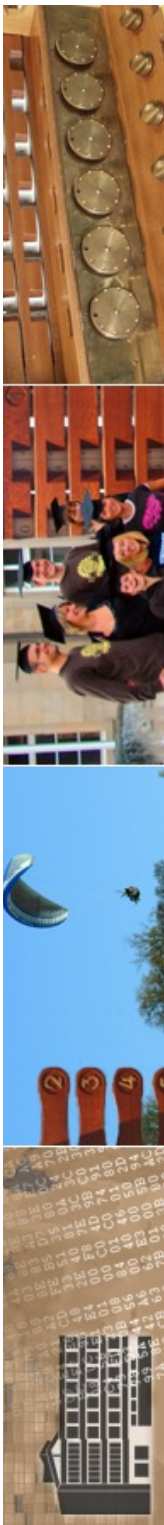
```
1 """
2 Django settings for {{ project_name }} project.
3
4 Generated by 'django-admin startproject' using Django {{ django_version }}.
5
6 For more information on this file, see
7 https://docs.djangoproject.com/en/{{ docs_version }}/topics/settings/
8
9 For the full list of settings and their values, 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))

```

...und nun...

- sind wir inhaltlich fertig mit der Veranstaltung
- ...und freuen uns auf das nächste Semester!

