



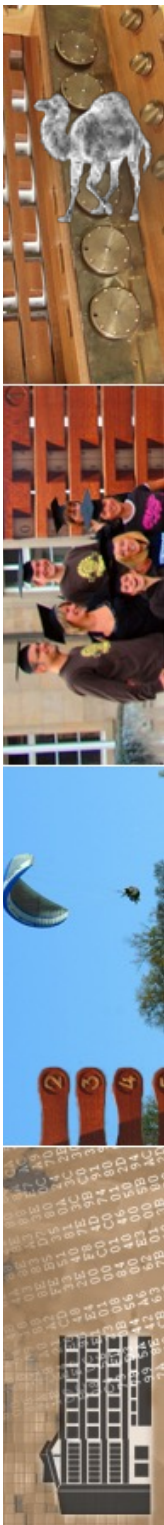
Grundlagen Internet-Technologien

INF3171

Serverseitige Web-Programmierung
mit Python

Version 1.0

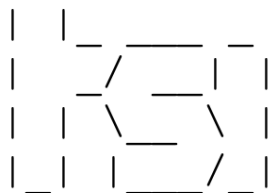
11.06.2026



Aktuelles

- Sommerfest der Informatik

findet am 26.06.2025 ab 17h statt!!!!!!



DIES IST EINE VON DER
FACHSCHAFT INFORMATIK
GENEHMIGTE RUNDMAIL!

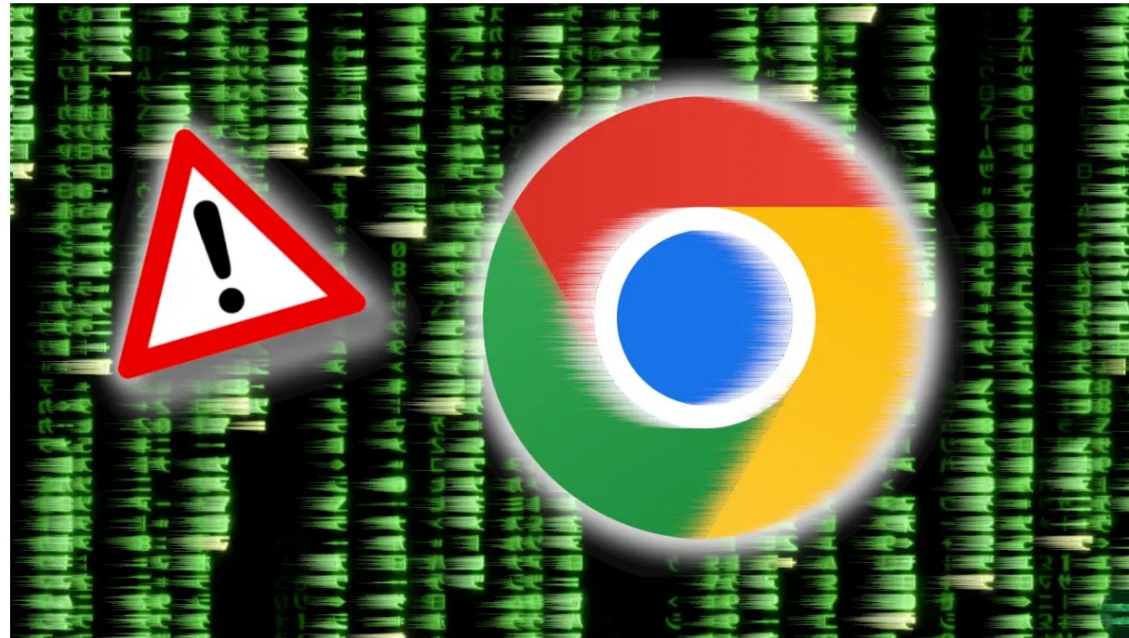
- - - - - English Version below - - - - -

Hallo zusammen,

auch in diesem Sommersemester veranstalten die Fachschaften Informatik und Kognitionswissenschaft ihr traditionelles Sommerfest. Dazu möchten wir euch, liebe Mitstudis, recht herzlich einladen! Kommt am 26. Juni ab 17 Uhr vorbei und verbringt mit uns einen entspannten Sommerabend.

Jetzt aktualisieren: Chrome-Update schließt angegriffene Sicherheitslücke

Das Update für Google Chrome sollten Nutzerinnen und Nutzer rasch installieren, da eine der geschlossenen Lücken angegriffen wird.



(Bild: heise medien)

09.06.2026, 08:01 Uhr | Lesezeit: 2 Min. | Security

Von Dirk Knop

Google hat Updates für den Webbrowser Chrome veröffentlicht. Wieder schließen die Entwickler eine hohe Anzahl an Sicherheitslücken, 74 Stück dieses Mal. Für eine davon kursiert bereits ein Exploit.



GIRO D'ITALIA GESAMTWERTUNG 2026

2026 ▼

GESAMTW...

TEAMWE...

PUNKTEWE...

BERGWE...

NACHWUCHS...

SP...

AKTIVSTER ...

FAHRER

ZEIT

1	  J. Vingegaard Team Visma-Lease a Bike	83:22:51
2	  F. Gall Decathlon CMA CGM	+5:22
3	  J. Hindley Red Bull-Bora-Hansgrohe	+6:25
4	  T. Arensman Netcompany INEOS	+7:02
5	  D. Gee-West Lidl Trek	+7:56
6	  A. Eulálio Bahrain Victorious	+9:39
7	  M. Storer Tudor Pro Cycling Team	+10:13



am 04.07. ist es soweit

STRECKE 2026

- ...die 115. Tour de France startet!
- Finale am 27.07. in Paris
- 21 Etappen

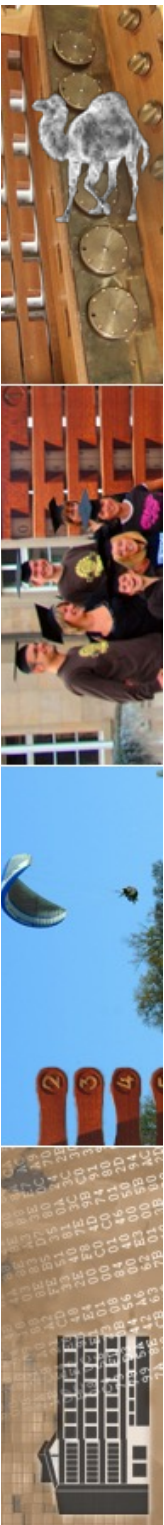


LÉGENDE / THE KEY

	Grand Départ Race Start		Arrivée finale Race finish
	Ville ou site départ Start town or site		Étape en ligne Stage
	Ville ou site arrivée Finish town or site		C.L.M. individuel Individual time trial
	Repos Rest day		C.L.M. par équipe Team time trial

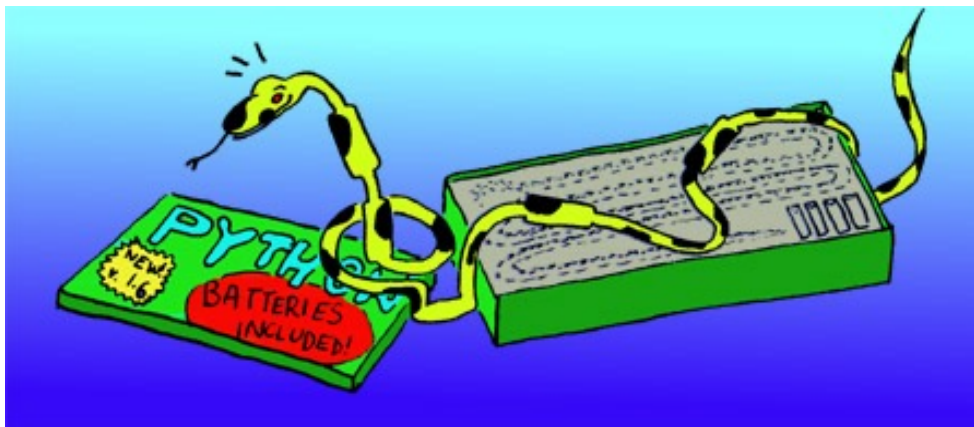
Evaluation

- bitte noch teilnehmen



Python Module

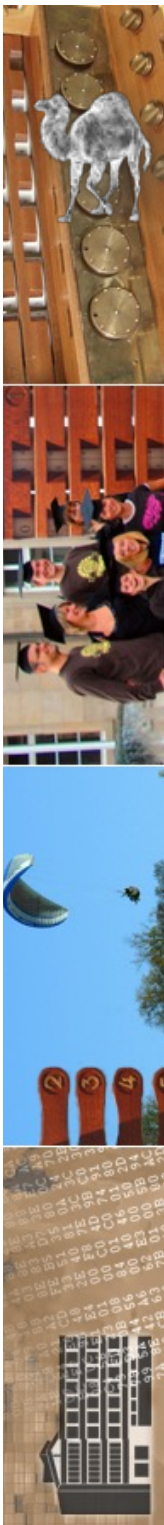
- zum Python-»Paket« gehören zahlreiche Module, die verschiedenste Funktionalitäten bereitstellen
- Module werden durch die Umgebungsvariable **PYTHONPATH** lokalisiert
 - Module liegen (typischerweise) in *vorcompilierter Form* als **.pyc** vor



noch mehr: Paketmanager

- darüber hinaus:
Erweiterung von Programmiersprachen über
Paketmanager
- Python: pip (Python installs packages)
(bzw. pip3)

<https://pip.pypa.io/en/stable/>





pip documentation v26.1.2

Search

- Getting Started
- Installation
- User Guide
- Topic Guides
- Reference
- Commands
- PROJECT
- Development
- UX Research & Design
- Changelog
- Code of Conduct
- GitHub

pip

pip is the [package installer for Python](#). You can use it to install packages from the [Python Package Index](#) and other indexes.

If you want to learn about how to use pip, check out the following resources:

- [Getting Started](#)
- [Python Packaging User Guide](#)

If you find bugs, need help, or want to talk to the developers, use our mailing lists or chat rooms:

- [GitHub Issues](#)
- [Discourse channel](#)
- [User IRC](#)
- [Development IRC](#)

If you find any security issues, please report to security@python.org

Privacy Notice

Pip does not collect any telemetry, however, it will send non-identifying (Python version, OS, etc.) to any remote indices used, who may choose to collect it. Please consult Pypi's privacy policy for their data collection and retention practices.

Build and run apps in over 115 regions with MongoDB Atlas, the database for every enterprise.

Read the Docs stable

Versions

latest stable

On Read the Docs

Project Home Builds

Search

Search docs

Addons documentation — Hosted by Read the Docs

Ads by EthicalAds

Close Ad



Getting Started - pip doc

pip.pypa.io/en/stable/getting-started/

pip documentation v26.1.2

Search

Getting Started

Installation

User Guide

Topic Guides

Reference

Commands

PROJECT

Development

UX Research & Design

Changelog

Code of Conduct

GitHub

Getting Started

To get started with using pip, you should [install Python](#) on your system.

Ensure you have a working pip

As a first step, you should check that you have a working Python with pip installed. This can be done by running the following commands and making sure that the output looks similar.

Linux MacOS Windows

```
$ python --version
Python 3.N.N
$ python -m pip --version
pip X.Y.Z from .. (python 3.N.N)
```

If that worked, congratulations! You have a working pip in your environment.

If you got output that does not look like the sample above, please read the [Installation](#) page. It provides guidance on how to install pip within a Python environment that doesn't have it.

Common tasks

Install a package

Linux MacOS Windows

[stable](#)

Simplify infrastructure with MongoDB Atlas, the leading modern database

Ads by EthicalAds

Close Ad





pip show - pip document

pip.pypa.io/en/stable/cli/pip_show/

pip documentation v26.1.2

Search

Getting Started

Installation

User Guide

Topic Guides

Reference

Commands

- pip
- pip install
- pip uninstall
- pip inspect
- pip list
- pip show**
- pip freeze
- pip check
- pip lock
- pip download

pip show

Usage

Unix/macOS Windows

```
python -m pip show [options] <package> ...
```

Description

Show information about one or more installed packages.

The output is in RFC-compliant mail header format.

Options

-f, --files

Show the full list of installed files for each package.

(environment variable: `PIP_FILES`)

Examples

- Show information about a package:

Develop and launch modern apps with MongoDB Atlas, a resilient data platform.

Ads by EthicalAds [Close Ad](#)



gitproject — -bash — 128x31

```
(.venv) thomas@PetitMouton =>
(.venv) thomas@PetitMouton => pip install Flask
Collecting Flask
  Downloading flask-3.1.3-py3-none-any.whl.metadata (3.2 kB)
Collecting blinker>=1.9.0 (from Flask)
  Downloading blinker-1.9.0-py3-none-any.whl.metadata (1.6 kB)
Collecting click>=8.1.3 (from Flask)
  Downloading click-8.4.1-py3-none-any.whl.metadata (2.6 kB)
Collecting itsdangerous>=2.2.0 (from Flask)
  Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting jinja2>=3.1.2 (from Flask)
  Downloading jinja2-3.1.6-py3-none-any.whl.metadata (2.9 kB)
Collecting markupsafe>=2.1.1 (from Flask)
  Downloading markupsafe-3.0.3-cp314-cp314-macosx_11_0_arm64.whl.metadata (2.7 kB)
Collecting werkzeug>=3.1.0 (from Flask)
  Downloading werkzeug-3.1.8-py3-none-any.whl.metadata (4.0 kB)
Download flask-3.1.3-py3-none-any.whl (103 kB)
Download blinker-1.9.0-py3-none-any.whl (8.5 kB)
Download click-8.4.1-py3-none-any.whl (116 kB)
Download itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Download jinja2-3.1.6-py3-none-any.whl (134 kB)
Download markupsafe-3.0.3-cp314-cp314-macosx_11_0_arm64.whl (12 kB)
Download werkzeug-3.1.8-py3-none-any.whl (226 kB)
Installing collected packages: markupsafe, itsdangerous, click, blinker, werkzeug, jinja2, Flask
Successfully installed Flask-3.1.3 blinker-1.9.0 click-8.4.1 itsdangerous-2.2.0 jinja2-3.1.6 markupsafe-3.0.3 werkzeug-3.1.8

[notice] A new release of pip is available: 26.1.1 -> 26.1.2
[notice] To update, run: pip install --upgrade pip
(.venv) thomas@PetitMouton =>
(.venv) thomas@PetitMouton =>
(.venv) thomas@PetitMouton =>
```


Beispiel: jupyter Notebook

Jupyter Notebook

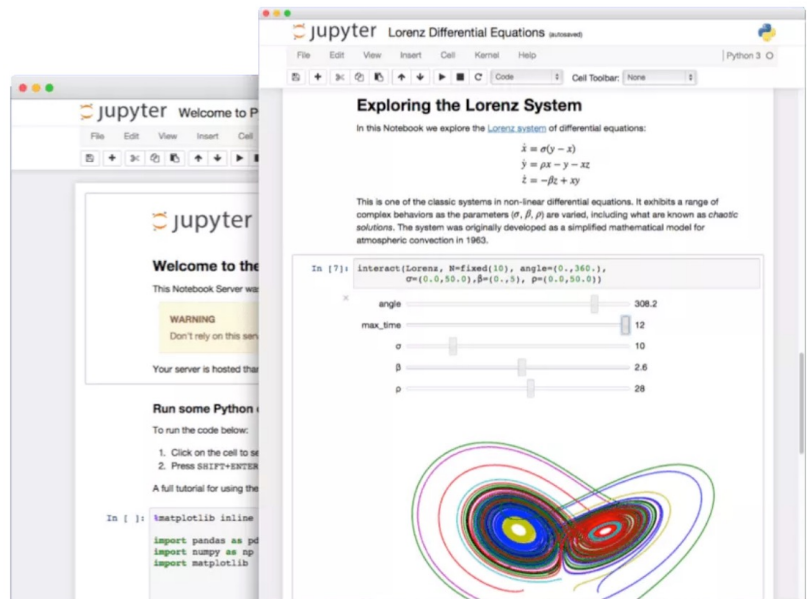
Install the classic Jupyter Notebook with:

```
pip install notebook
```

To run the notebook:

```
jupyter notebook
```





Jupyter Notebook: The Classic Notebook Interface

The Jupyter Notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience.

Try it in your browser

Install the Notebook



Language of choice

Jupyter supports over 40 programming languages, including Python, R, Julia, and Scala.



Share notebooks

Notebooks can be shared with others using email, Dropbox, GitHub and the [Jupyter Notebook Viewer](#).



Interactive output

Your code can produce rich, interactive output: HTML, images, videos, LaTeX, and custom MIME types.



Big data integration

Leverage big data tools, such as Apache Spark, from Python, R, and Scala. Explore that same data with pandas, scikit-learn, ggplot2, and TensorFlow.



```
thomas@PetitMouton =>
thomas@PetitMouton => pip3 install notebook
Collecting notebook
  Downloading notebook-6.5.4-py3-none-any.whl (529 kB)
    _____ 529.8/529.8 KB 2.5 MB/s eta 0:00:00
Collecting jupyter-core>=4.6.1
  Downloading jupyter_core-5.3.0-py3-none-any.whl (93 kB)
    _____ 93.2/93.2 KB 3.4 MB/s eta 0:00:00
Collecting nbformat
  Downloading nbformat-5.9.0-py3-none-any.whl (77 kB)
    _____ 77.6/77.6 KB 3.8 MB/s eta 0:00:00
Collecting traitlets>=4.2.1
  Downloading traitlets-5.9.0-py3-none-any.whl (117 kB)
    _____ 117.4/117.4 KB 3.6 MB/s eta 0:00:00
Collecting terminado>=0.8.3
  Downloading terminado-0.17.1-py3-none-any.whl (17 kB)
Collecting ipython-genutils
  Downloading ipython_genutils-0.2.0-py2.py3-none-any.whl (26 kB)
Collecting jupyter-client>=5.3.4
  Downloading jupyter_client-8.2.0-py3-none-any.whl (103 kB)
    _____ 103.2/103.2 KB 3.4 MB/s eta 0:00:00
Collecting pyzmq>=17
  Downloading pyzmq-25.1.0-cp310-cp310-macosx_10_15_universal2.whl (1.8 MB)
    _____ 1.8/1.8 MB 4.9 MB/s eta 0:00:00
Collecting jinja2
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
    _____ 133.1/133.1 KB 6.7 MB/s eta 0:00:00
Collecting ipykernel
  Downloading ipykernel-6.23.1-py3-none-any.whl (152 kB)
    _____ 152.2/152.2 KB 6.0 MB/s eta 0:00:00
Collecting nbconvert>=5
```

Installing collected packages: webencodings, wcwidth, pure-eval, ptyprocess, pickleshare, mistune, ipython-genutils, fastjsonschema, executing, backcall, appnope, websocket-client, webcolors, uri-template, traitlets, tornado, tinycss2, soupsieve, sniffio, six, Send2Trash, rfc3986-validator, pyzmq, pyyaml, python-json-logger, pypersistent, pygments, pyparser, psutil, prompt-toolkit, prometheus-client, platformdirs, pexpect, parso, pandocfilters, packaging, overrides, nest-asyncio, markupsafe, jupyterlab-pygments, jsonpointer, idna, fqdn, exceptiongroup, defusedxml, decorator, debugpy, attrs, terminado, rfc3339-validator, python-dateutil, matplotlib-inline, jupyter-core, jsonschema, jinja2, jedi, comm, cffi, bleach, beautifulsoup4, asttokens, anyio, stack-data, nbformat, jupyter-server-terminals, jupyter-client, arrow, argon2-cffi-bindings, nbclient, isoduration, ipython, argon2-cffi, nbconvert, ipykernel, jupyter-events, jupyter-server, notebook-shim, nbclassic, notebook. Successfully installed Send2Trash-1.8.2 anyio-3.7.0 appnope-0.1.3 argon2-cffi-21.3.0 argon2-cffi-bindings-21.2.0 arrow-1.2.3 asttokens-2.2.1 attrs-23.1.0 backcall-0.2.0 beautifulsoup4-4.12.2 bleach-6.0.0 cffi-1.15.1 comm-0.1.3 debugpy-1.6.7 decorator-5.1.1 defusedxml-0.7.1 exceptiongroup-1.1.1 executing-1.2.0 fastjsonschema-2.19.1 fqdn-1.5.1 idna-3.4 ipykernel-6.23.1 ipython-8.14.0 ipython-genutils-0.2.0 isoduration-20.11.0 jedi-0.18.2 jinja2-3.1.2 jsonpointer-2.3 jsonschema-4.17.3 jupyter-client-8.2.0 jupyter-core-5.3.0 jupyter-events-0.6.3 jupyter-server-2.6.0 jupyter-server-terminals-0.4.4 jupyterlab-pygments-0.2.2 markupsafe-2.1.3 matplotlib-inline-0.1.6 mistune-2.0.5 nbclassic-1.0.0 nbclient-0.8.0 nbconvert-7.4.0 nbformat-5.9.0 nest-asyncio-1.5.6 notebook-6.5.4 notebook-shim-0.2.3 overrides-7.3.1 packaging-21.3 pandocfilters-1.5.0 parso-0.8.3 pexpect-4.8.0 pickleshare-0.7.5 platformdirs-3.5.3 prometheus-client-0.17.0 prompt-toolkit-3.0.38 psutil-5.9.5 ptyprocess-0.7.0 pure-eval-0.2.2 pyparser-2.21 pygments-2.15.1 pypersistent-0.19.3 python-dateutil-2.8.2 python-json-logger-2.0.7 pyyaml-6.0 pyzmq-25.1.0 rfc3339-validator-0.1.4 rfc3986-validator-0.1.1 six-1.16.0 sniffio-1.3.0 soupsieve-2.4.1 stack-data-0.6.2 terminado-0.17.1 tinycss2-1.2.1 tornado-6.3.2 traitlets-5.9.0 uri-template-1.2.0 wcwidth-0.2.6 webcolors-1.13 webencodings-0.5.1 websocket-client-1.5.3

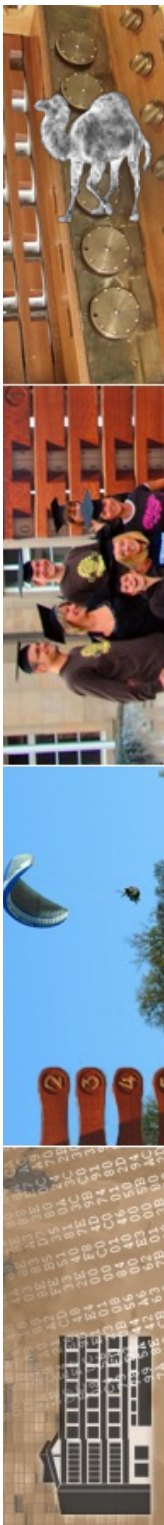


```
Collecting argon2-cffi
  Downloading argon2_cffi-21.3.0-py3-none-any.whl (14 kB)
Collecting nest-asyncio>=1.5
```



Importieren von Modulen

- wir unterscheiden
 - **`import modulname`**
 - Importiert das Modul <modulname>
 - erweitert Namensraum entsprechend (modulname dann bekannt)
 - **`from modulname import obj1, obj2, obj3`**
 - importiert nur die Klassen obj1, obj2 und obj3 vom Modul modulname
 - Namensraum wird nur um diese Klassen erweitert

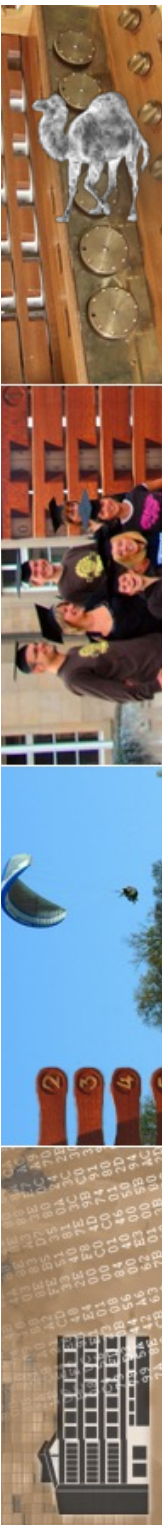


Dokumentation zu den Modulen

- unter
 - <https://docs.python.org/3/py-modindex.html>

gibt es Informationen zu allen Modulen des Python-Systems

- Hinweis: umfangreich!





pip documentation v26.1 Python Module Index

docs.python.org/3/py-modindex.html

Python » English 3.14.5 3.14.5 Documentation » Python Module Index Quick search Go modules index Theme Auto

Python Module Index

[_](#) [a](#) [b](#) [c](#) [d](#) [e](#) [f](#) [g](#) [h](#) [i](#) [j](#) [k](#) [l](#) [m](#) [n](#) [o](#) [p](#) [q](#) [r](#) [s](#) [t](#) [u](#) [v](#) [w](#) [x](#) [z](#)

_

[_future_](#) Future statement definitions

[_main_](#) The environment where top-level code is run. Covers command-line interfaces, import-time behavior, and ``\_\_name\_\_ == '\_\_main\_\_'``.

[_thread](#) Low-level threading API.

[_tkinter](#) A binary module that contains the low-level interface to Tcl/Tk.

a

[abc](#) Abstract base classes according to :pep: '3119'.

[aifc](#) **Deprecated:** Removed in 3.13.

[annotationlib](#) Functionality for introspecting annotations

[argparse](#) Command-line option and argument parsing library.

[array](#) Space efficient arrays of uniformly typed numeric values.

[ast](#) Abstract Syntax Tree classes and manipulation.

[asynchat](#) **Deprecated:** Removed in 3.12.

[asyncio](#) Asynchronous I/O.

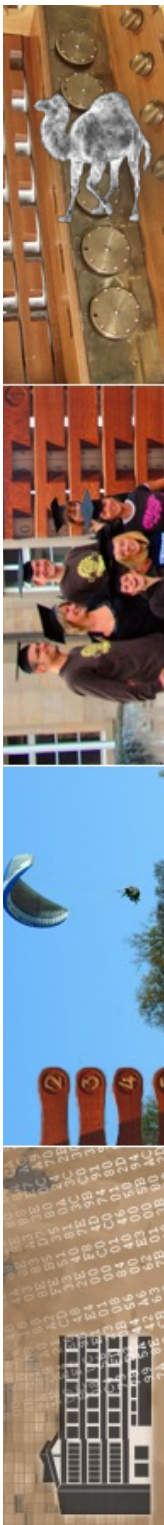
[asyncore](#) **Deprecated:** Removed in 3.12.

[atexit](#) Register and execute cleanup functions.

[audioop](#) **Deprecated:** Removed in 3.13.

Standard-Module

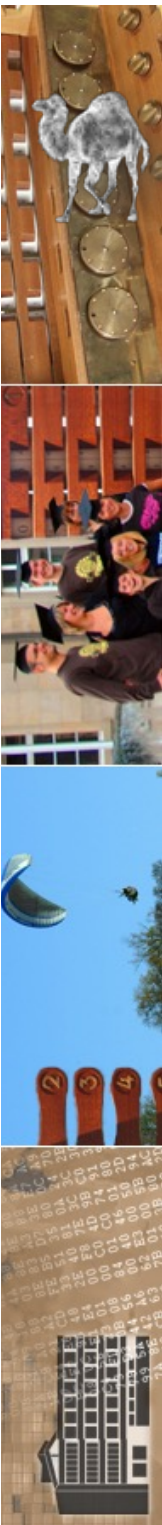
- Modul `string`
 - stellt grundlegende Funktionalität für Zeichenketten zur Verfügung
 - wesentliche Teile sind inzwischen in den Kernsprachumfang von Python integriert
 - ab Python 3.0 entfallen Funktionen wie `split` im Modul und gehören direkt zu Python

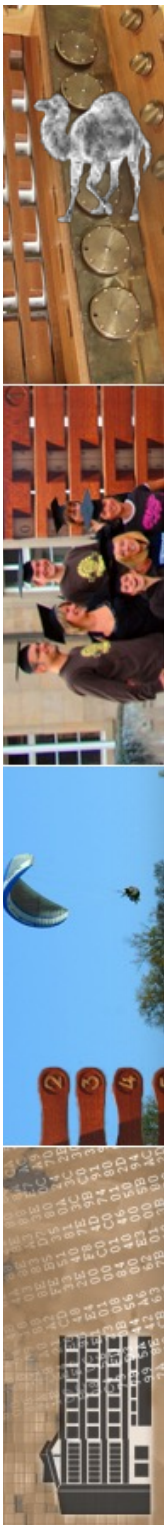


© 2026 Universität
Tübingen

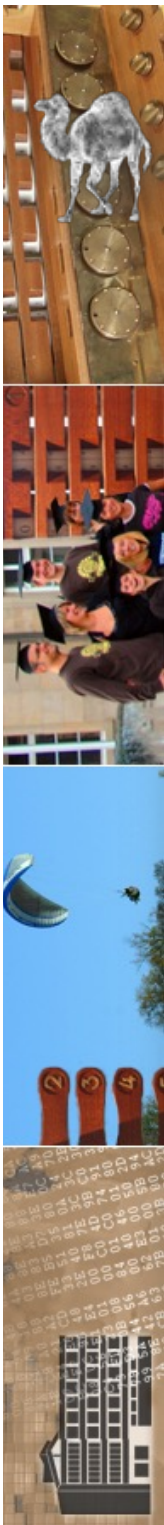
Modul `math`

- Mathematische Funktionalität
 - Attribute:
 - Konstante π
 - Konstante e
 - Methoden: zahlreiche mathematische Funktionen
 - `sqrt(x)`
 - `pow(x, y)`
 - `sin(x)`, `sinh(x)`, `cos(x)`, `cosh(x)`, ...
 - Modul `cmath` für *komplexe* Funktionen





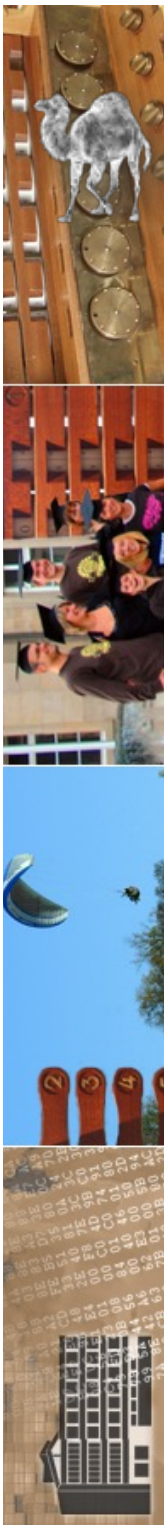
© 2026 Universität
Tübingen

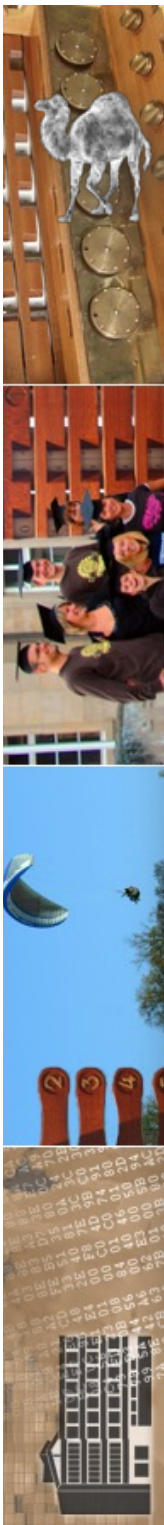


© 2026 Universität
Tübingen

Modul `time`

- Python-Modul für Zeitformatierungen
- Der Nullpunkt ist Unix-typisch der 1.1.1970, 0h
- die Methoden
 - `gmtime()`
 - `localtime()`
- geben typische Liste mit Zeitinformationen zurück
 - (Beispiel in Verbindung mit CGI)





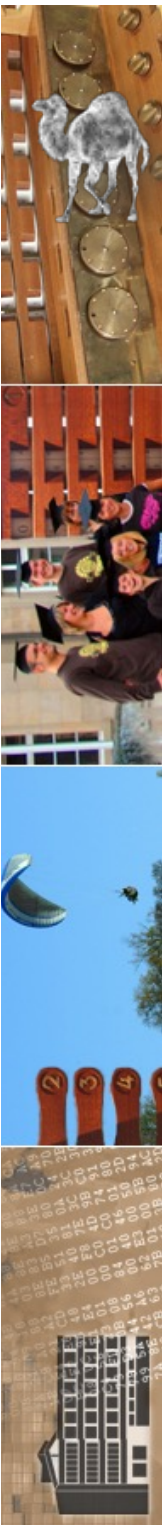
© 2026 Universität
Tübingen

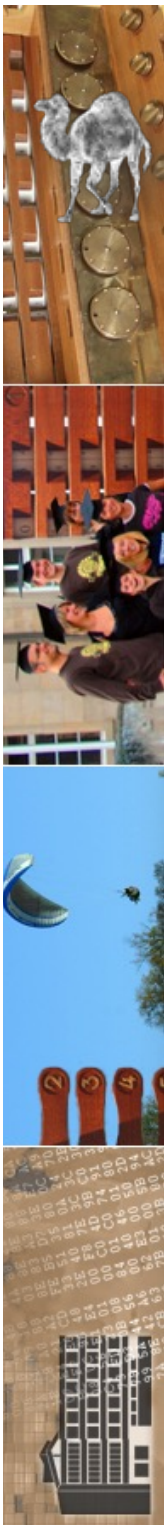
gmtime () und localtime ()

- | Index | Field | Values |
|-------|-----------------------|--------------------------|
| 0 | year | (for example, 1993) |
| 1 | month | range [1,12] |
| 2 | day | range [1,31] |
| 3 | hour | range [0,23] |
| 4 | minute | range [0,59] |
| 5 | second | range [0,61] |
| 6 | weekday | range [0,6], Monday is 0 |
| 7 | Julian day | range [1,366] |
| 8 | daylight savings flag | 0, 1 or -1 |

Modul os

- Modul, um (u.a.) Multitasking zu ermöglichen
- os -- *Miscellaneous* operating system interfaces
- “This module provides a portable way of using operating system dependent functionality.”
 - fork()
 - Fork a child process. Return 0 in the child, the child's process id in the parent. Availability: Unix.

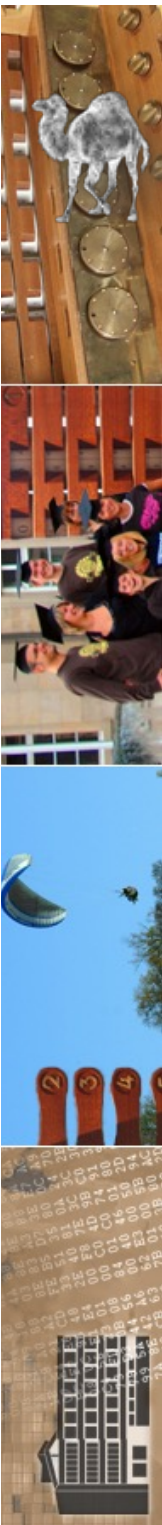




© 2026 Universität
Tübingen

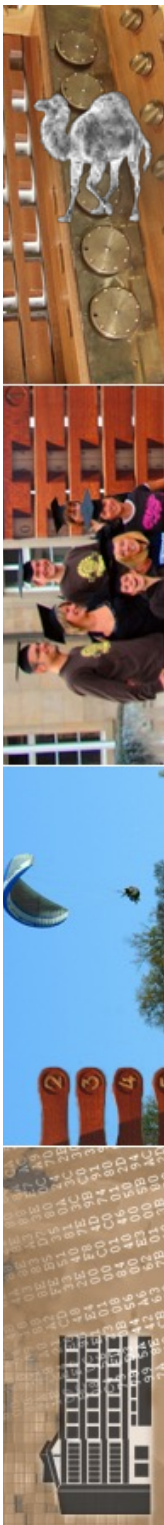
Situation

- bisher: Python ist einfache und nützliche (Systemadministration) Programmiersprache
- nun: wir nutzen Python als **CGI-Sprache**
- Fragen:
 - keine direkt ausführbaren Codes mit Python
 - was ist alles zu tun für CGI mit Python
 - **Formularverarbeitung** mit Python
 - **Generation von HTML** mit Python



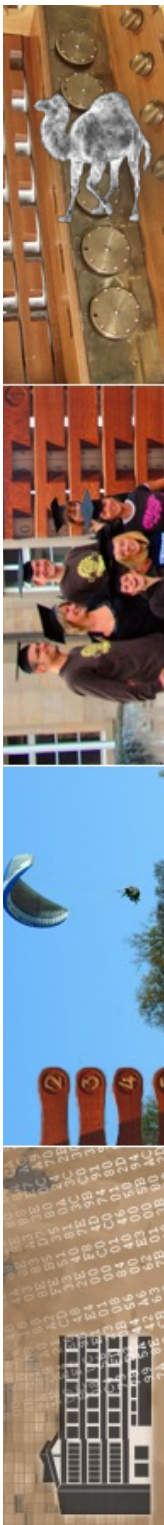
Python und das Internet

- Python ist - ähnlich wie Java - eine *für das Internet optimierte* Programmiersprache
- schon zum Standard-Umfang gehört eine sehr große Bibliothek nützlicher Klassen - ähnlich wie das Paket `java.net`
- dabei u.a. die Klasse `cgi` (vergleichbar ist etwa das PERL-Modul `CGI.pm`)



ganz typisch

- typische Aufgabe:
 - Formular (HTML-<FORM>) wird ausgefüllt und an ein CGI-Script gesendet (mit HTTP-GET oder HTTP-POST)
 - Formular besteht aus mehreren <INPUT>-Feldern
 - CGI-Script muss die einzelnen Formular-Felder auswerten
 - möglich mit CGI-Umgebungsvariablen **QUERY_STRING** - aber nicht sinnvoll





Previous topic

20.13. `xml.parsers.expat`
— Fast XML parsing using
Expat

Next topic

21.1. `webbrowser` —
Convenient Web-browser
controller

This Page

[Report a Bug](#)
[Show Source](#)

21. Internet Protocols and Support

The modules described in this chapter implement Internet protocols and support for related technology. They are all implemented in Python. Most of these modules require the presence of the system-dependent module `socket`, which is currently supported on most popular platforms. Here is an overview:

- 21.1. `webbrowser` — Convenient Web-browser controller
 - 21.1.1. Browser Controller Objects
- 21.2. `cgi` — Common Gateway Interface support
 - 21.2.1. Introduction
 - 21.2.2. Using the `cgi` module
 - 21.2.3. Higher Level Interface
 - 21.2.4. Functions
 - 21.2.5. Caring about security
 - 21.2.6. Installing your CGI script on a Unix system
 - 21.2.7. Testing your CGI script
 - 21.2.8. Debugging CGI scripts
 - 21.2.9. Common problems and solutions
- 21.3. `cgitb` — Traceback manager for CGI scripts
- 21.4. `wsgiref` — WSGI Utilities and Reference Implementation
 - 21.4.1. `wsgiref.util` - WSGI environment utilities
 - 21.4.2. `wsgiref.headers` - WSGI response header tools
 - 21.4.3. `wsgiref.simple_server` - a simple WSGI HTTP server
 - 21.4.4. `wsgiref.validate` — WSGI conformance checker
 - 21.4.5. `wsgiref.handlers` - server/gateway base classes
 - 21.4.6. Examples
- 21.5. `urllib` — URL handling modules
- 21.6. `urllib.request` — Extensible library for opening URLs
 - 21.6.1. Request Objects
 - 21.6.2. OpenerDirector Objects
 - 21.6.3. BaseHandler Objects
 - 21.6.4. HTTPRedirectHandler Objects

cgi Essentials (!!!)

cgi-Programm muss **ausführbar** sein

Code liegt im Unterordner cgi-bin

Ausgabe beginnt mit:

```
Content-type: text/html  
\n
```

Zugriff: URL beginnt mit „cgi-bin“



<
>
↺
⌵
|
⚠
134.2.6.146/~zrvwa01/cgi-bin/python/cgi_3.py
|
🔍
📷
✅
➤
❤
|
📶
📶
≡

Internal Server Error

The server encountered an internal error or misconfiguration and was unable to complete your request.

Please contact the server administrator at `webmaster@localhost` to inform them of the time this error occurred, and the actions you performed just before this error.

More information about this error may be available in the server error log.

Apache/2.4.25 (Debian) Server at 134.2.6.146 Port 80



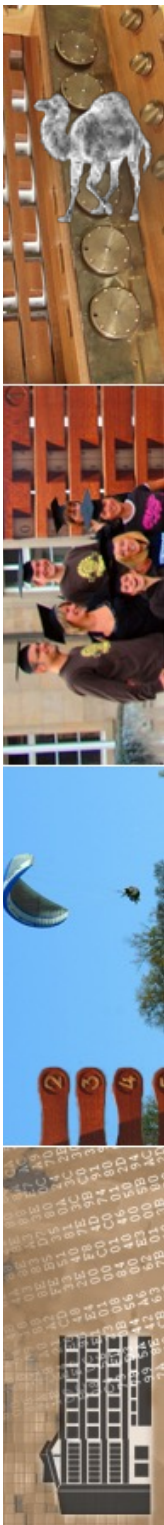
Wie wird Python ausführbar?

- ...durch die Shebang-Zeile:

erste (!!!) Zeile im Scriptprogramm ist ein besonderer Kommentar:

```
#!<path_to_python>/python3
```

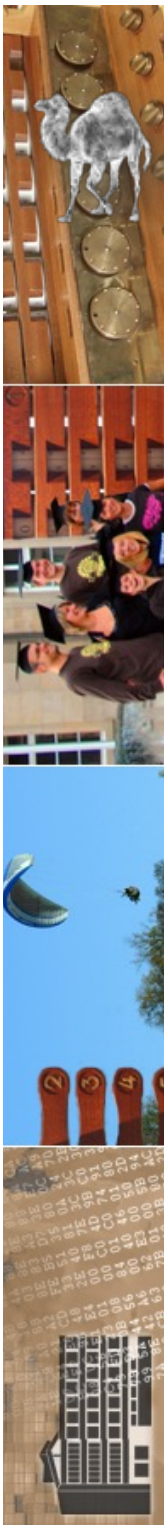
- zusätzlich auf Unix x-bit setzen:
`chmod +x <pythonscript>`
 - und schon ist's ausführbar!
 - gilt für *alle Scriptsprachen*





HelloWorld-CGI in Python

- wir können direkt ein erstes Python-HelloWorld schreiben
- dafür brauchen wir lediglich die richtige *Shebang-Zeile*
- nützlich: typisches **Here-Document**
(`''' Text '''`)





hellocgi.py — /Users/thomas/Documents/Eberhardina/InformationsDienste/Lehre/source/python

math.py x HTTPclient.py x hellocgi.py x time.py x

```
1  #!/usr/bin/python3
2  #
3  # Grundlagen Internet-Technologien
4  #
5  # Formularverarbeitung mit dem Python-Modul CGI
6
7  print (''Content-type: text/html
8
9  <!DOCTYPE html>
10 <HTML>
11 <HEAD>
12     <TITLE>HelloWorld-CGI in python</TITLE>
13     <link rel="stylesheet" type="text/css" href="./css/webkompendium.css">
14 </HEAD>
15 <BODY>
16     <BR /><CENTER><H3>
17     Hello World<BR />GIT likes Python
18     </H3></CENTER><BR />
19 </BODY>
20 </HTML>'' )
```

hellocgi.py 1:1 LF ISO 8859-15 Python 1 update





```
zrvwa01@infodienste =>
zrvwa01@infodienste => more hellocgi.py
#!/usr/bin/python3
#
# Grundlagen Internet-Technologien
#
# CGI mit Python

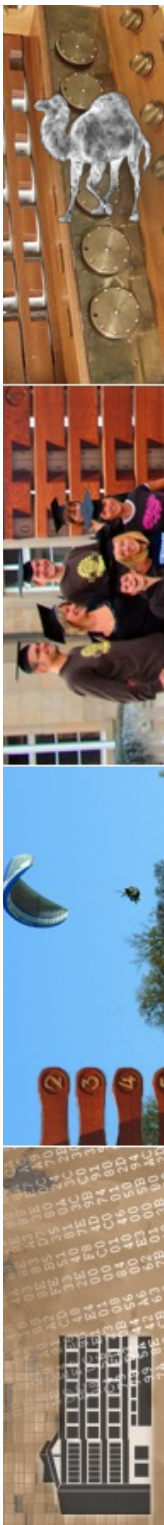
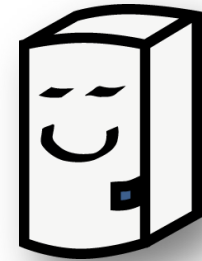
# here-Dokument in Python
print ('''Content-type: text/html

<!DOCTYPE html>
<HTML>
<HEAD>
    <TITLE>HelloWorld-CGI in python</TITLE>
    <link rel="stylesheet" type="text/css" href="./css/webkompendium.css">
</HEAD>
<BODY>
    <BR /><CENTER><H3>
    Hello World<BR />GIT likes Python
    </H3></CENTER><BR />
</BODY>
</HTML>''')
zrvwa01@infodienste =>
```



nächste Schritte

- kopieren nach cgi-bin
- x-bit setzen (Unix/Linux)
- Testen des Scripts **lokal** ("ohne Web")
 - *Script muss auch ohne Web fehlerfrei laufen!*
 - ohne explizit den PYTHON-Interpreter zu starten
- Aufruf aus Client-Browser





```
[zrvwa01@infodienste =>
[zrvwa01@infodienste =>
[zrvwa01@infodienste => ll hellocgi.py
-rwxr-xr-x 1 zrvwa01 443 29. Mai 13:34 hellocgi.py
[zrvwa01@infodienste =>
[zrvwa01@infodienste =>
```





```
zrvwa01@infodienste =>
zrvwa01@infodienste => ./hellocgi.py
Content-type: text/html

<!DOCTYPE html>
<HTML>
<HEAD>
  <TITLE>HelloWorld-CGI in python</TITLE>
  <link rel="stylesheet" type="text/css" href="./css/webkompendium.css">
</HEAD>
<BODY>
  <BR /><CENTER><H3>
  Hello World<BR />GIT likes Python
  </H3></CENTER><BR />
</BODY>
</HTML>
zrvwa01@infodienste =>
```

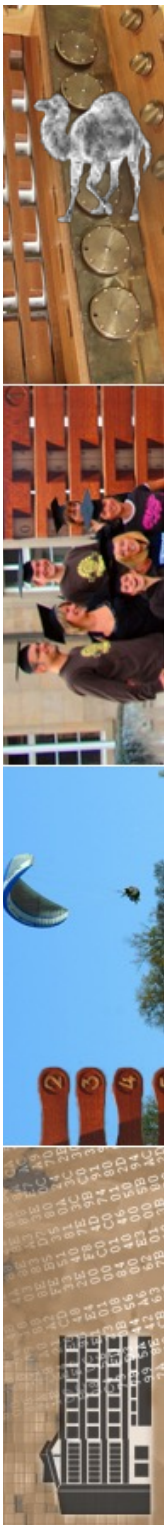


Ausführen im Web

- über die Default-cgi-URL

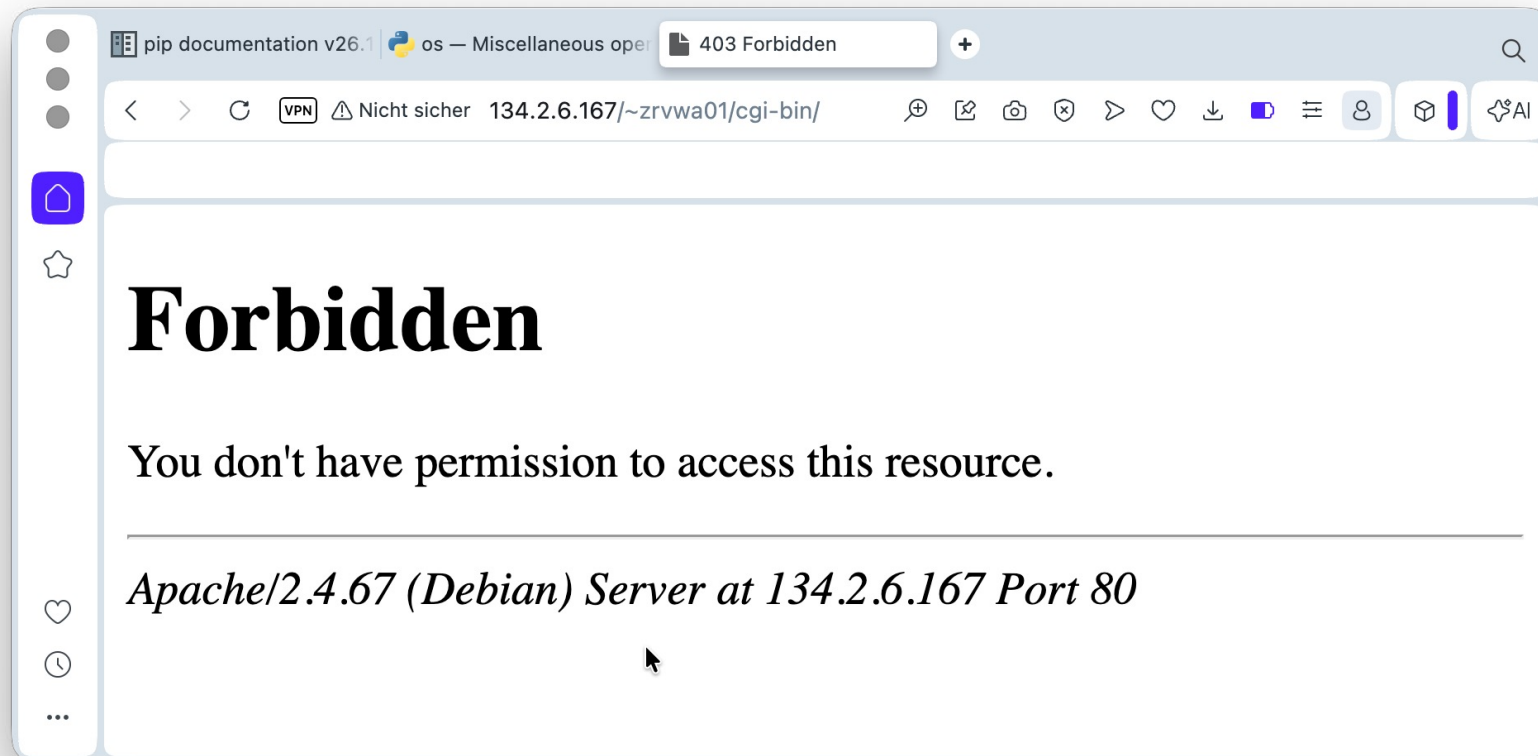
http://.../cgi-bin/name_des_scripts.py

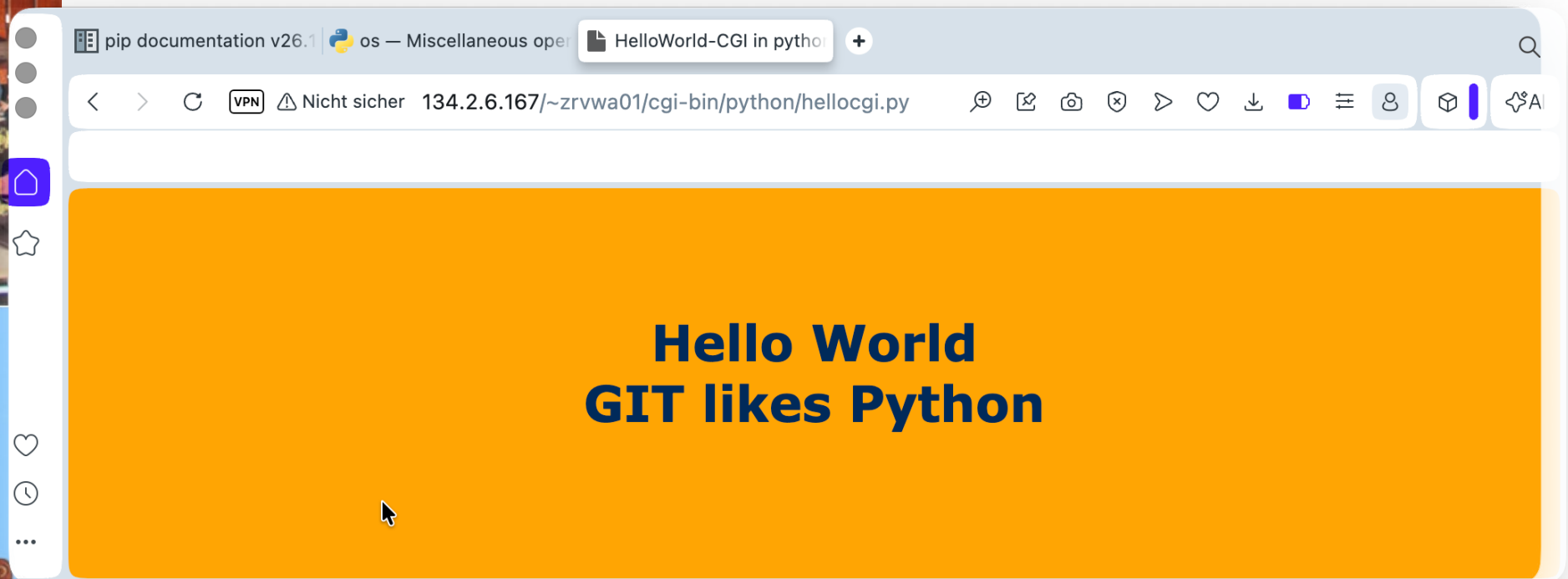
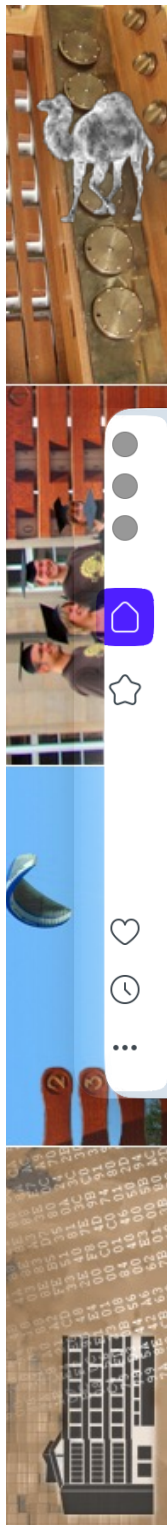
(Default wegen der Default-Apache-Config
ScriptAlias cgi-bin)





Security



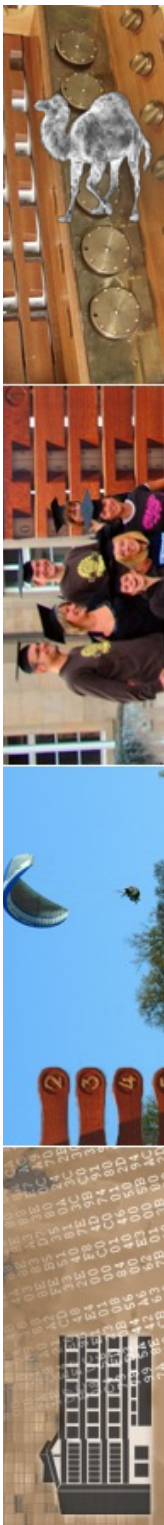


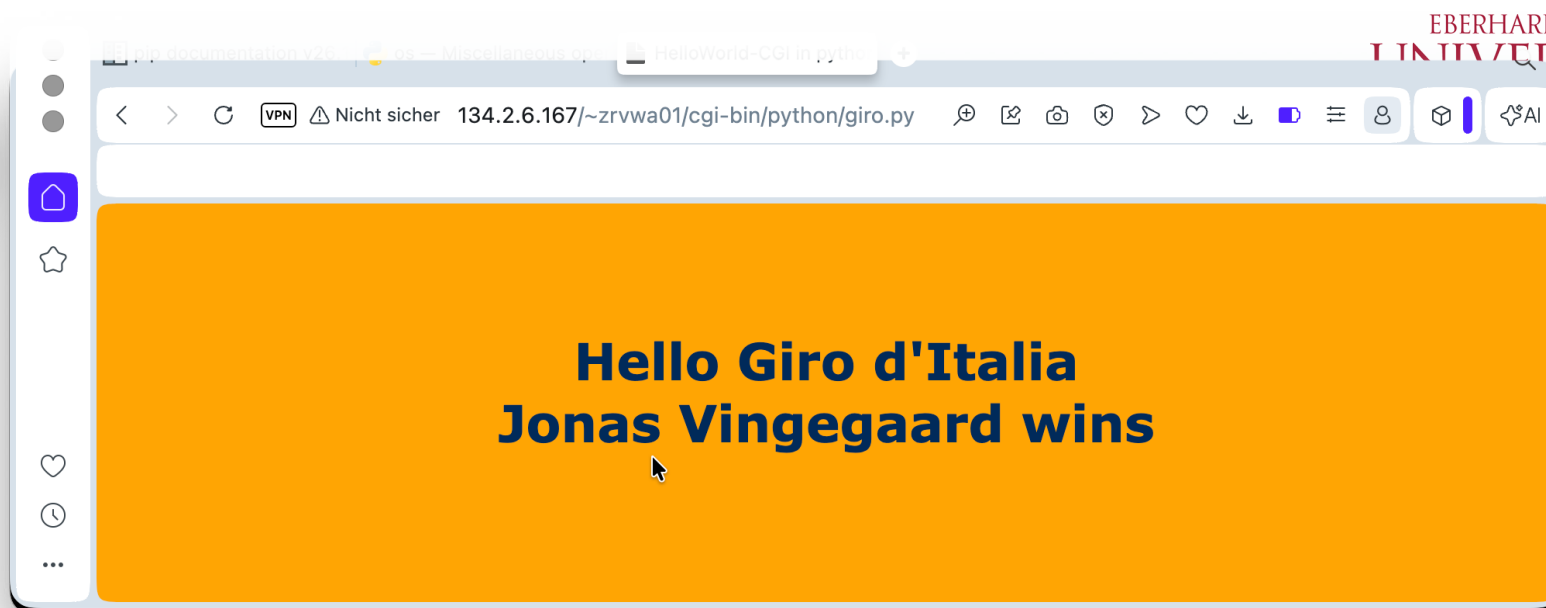
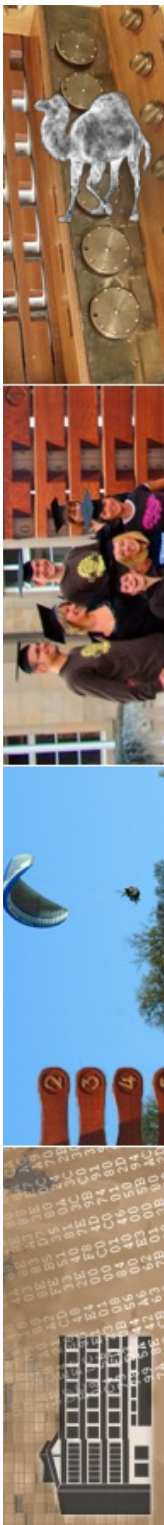
Here-Documents

- einbetten größerer „Raw-Texte“ in Python mittels

dreier Anführungszeichen ““ oder ““““

- für CGIs praktisch

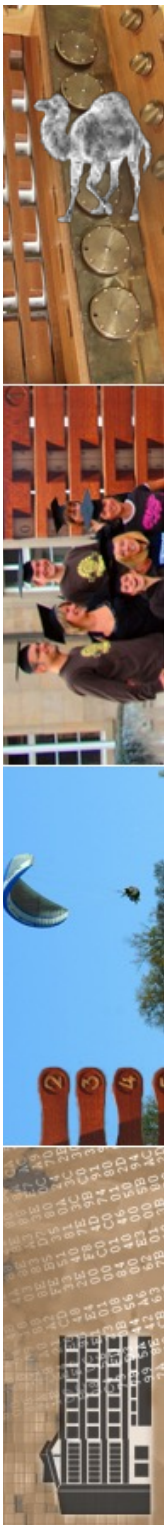




```
thomas — ssh — -bash — 78x20

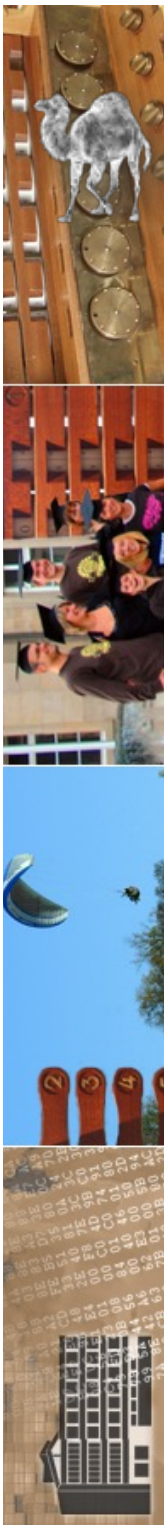
zrvwa01@infodienste =>
zrvwa01@infodienste => ./giro.py
Content-type: text/html

<!DOCTYPE html>
<HTML>
<HEAD>
  <TITLE>HelloWorld-CGI in python</TITLE>
  <link rel="stylesheet" type="text/css" href="./css/webkompendium.css">
</HEAD>
<BODY>
  <BR /><CENTER><H3>
    Hello Giro d'Italia<BR />Jonas Vingegaard wins
  </H3></CENTER><BR />
</BODY>
</HTML>
zrvwa01@infodienste =>
zrvwa01@infodienste =>
zrvwa01@infodienste =>
zrvwa01@infodienste =>
```



erste Dynamik

- als erstes Beispiel einer CGI-Dynamit integrieren wir eine Angabe der Serverzeit
- mittels Python-Modul `time` einfach möglich





```
zrvwa01@infodienste =>
zrvwa01@infodienste => more time.py
#!/usr/bin/python3
#
# Grundlagen Internet-Technologien
#
# CGI mit Python: Zeitausgabe

import time;

print ("Content-type: text/html

<!DOCTYPE html>
<HTML>
  <HEAD>
    <TITLE>Zeitausgabe als Python-cgi</TITLE>
    <link rel="stylesheet" type="text/css" href="./css/webkompendium.css">
  </HEAD>
  <BODY">

zeit = time.time()

print ("      <center><p><h2>Grundlagen Internet-Technologien</h2></p><p><hr></p><p><h3>Zeit auf dem Server: ")
print (zeit)

print ("      </h3></p></center>
      <p><hr></p><p><center><font size=\"-1\">
      <script>document.write(document.lastModified);</script></font></center></p>
    </BODY>
  </HTML>")

zrvwa01@infodienste =>
```





Python Modul cgi

- stellt u.a. die benötigte Funktionalität für *Formularverarbeitung* her
- darüber hinaus zahlreiche weitere nützliche Funktionen für die cgi-Programmierung

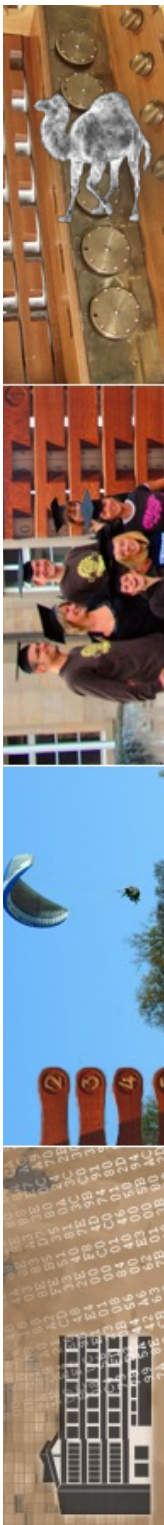




Table of Contents

cgi — Common Gateway Interface support

- Introduction
- Using the cgi module
- Higher Level Interface
- Functions
- Caring about security
- Installing your CGI script on a Unix system
- Testing your CGI script
- Debugging CGI scripts
- Common problems and solutions

Previous topic

webbrowser — Convenient Web-browser controller

Next topic

cgitb — Traceback manager for CGI scripts

This Page

[Report a Bug](#)
[Show Source](#)

cgi — Common Gateway Interface support

Source code: [Lib/cgi.py](#)

Support module for Common Gateway Interface (CGI) scripts.

This module defines a number of utilities for use by CGI scripts written in Python.

Introduction

A CGI script is invoked by an HTTP server, usually to process user input submitted through an HTML `<FORM>` or `<ISINDEX>` element.

Most often, CGI scripts live in the server's special `cgi-bin` directory. The HTTP server places all sorts of information about the request (such as the client's hostname, the requested URL, the query string, and lots of other goodies) in the script's shell environment, executes the script, and sends the script's output back to the client.

The script's input is connected to the client too, and sometimes the form data is read this way; at other times the form data is passed via the "query string" part of the URL. This module is intended to take care of the different cases and provide a simpler interface to the Python script. It also provides a number of utilities that help in debugging scripts, and the latest addition is support for file uploads from a form (if your browser supports it).

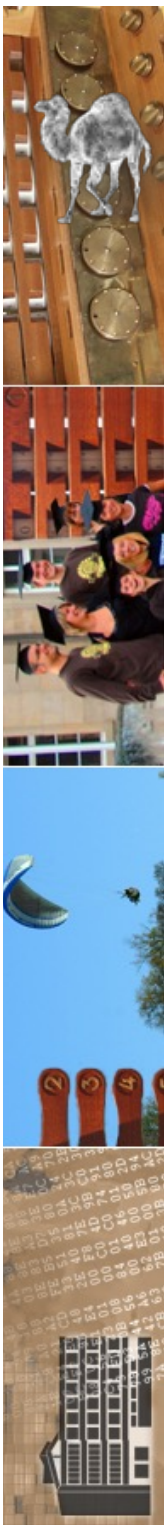
The output of a CGI script should consist of two sections, separated by a blank line. The first section contains a number of headers, telling the client what kind of data is following. Python code to generate a minimal header section looks like this:

```
print("Content-Type: text/html")    # HTML is following
print()                           # blank line, end of headers
```



Formularverarbeitung

- cgi-Modul enthält Klasse **FieldStorage**, welche die Verarbeitung komplexer Eingabeformulare ermöglicht
 - Objekt der Klasse FieldStorage verhält sich ähnlich wie `ass. Array`
 - ein einzelner Eintrag - also ein ausgefülltes Formularelement - ist ein Objekt der Klasse **MiniFieldStorage**; im Normalfall ist dies ein key-value-Paar
 - Beispiel:
 - `FieldStorage()`
 - `print FieldStorage("eingabe").value`





```
zrvwa01@infodienste => more cgi_1.py
#!/usr/bin/python3
#
# Grundlagen Internet-Technologien
#
# Formularverarbeitung mit dem Python-Modul CGI

### Import des CGI-Moduls
import cgi

print('Content-Type: text/html

<!DOCTYPE html>
<html><head><title>Python-CGI mit Formularverarbeitung</title><link rel="stylesheet"
type="text/css" href="./css/webkompendium.css"></head>
<body><p><center><h2>GIT - Python</h2></p><hr>
<h3>Auswertung aller Formularfelder<br><br>''')

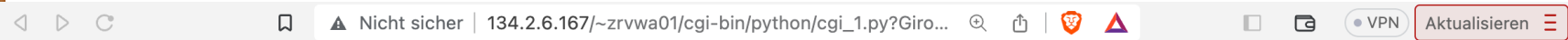
##### Auswertung des Formulars #####

form = cgi.FieldStorage()

### Ausgabe aller values des Formulars
for key in form.keys():
    print('<br>Schlüssel: ',key,' hat value ',form[key].value)

#####

print ('</h3><hr></center></body></html>')
zrvwa01@infodienste =>
```



GIT - Python

Auswertung aller Formularfelder

Schlüssel: Giro hat value Roglic
Schlüssel: Tour hat value Vingegaard



Funktionen Modul cgi



- `cgi.parse(fp=None, environ=os.environ, keep_blank_values=False, strict_parsing=False)`
Parse a query in the environment or from a file (the file defaults to `sys.stdin`).
The *keep_blank_values* and *strict_parsing* parameters are passed to [`urllib.parse.parse\_qs\(\)`](#) unchanged.
- `cgi.parse_qs(qs, keep_blank_values=False, strict_parsing=False)`
This function is deprecated in this module. Use [`urllib.parse.parse\_qs\(\)`](#) instead.
It is maintained here only for backward compatibility.
- `cgi.parse_qs1(qs, keep_blank_values=False, strict_parsing=False)`
This function is deprecated in this module. Use [`urllib.parse.parse\_qs\(\)`](#) instead.
It is maintained here only for backward compatibility.
- `cgi.parse_multipart(fp, pdict)`
Parse input of type *multipart/form-data* (for file uploads). Arguments are *fp* for the input file and *pdict* for a dictionary containing other parameters in the *Content-Type* header.
Returns a dictionary just like [`urllib.parse.parse\_qs\(\)`](#) keys are the field names, each value is a list of values for that field. This is easy to use but not much good if you are expecting megabytes to be uploaded — in that case, use the `FieldStorage` class instead which is much more flexible.
Note that this does not parse nested multipart parts — use `FieldStorage` for that.
- `cgi.parse_header(string)`
Parse a MIME header (such as *Content-Type*) into a main value and a dictionary of parameters.
- `cgi.test()`
Robust test CGI script, usable as main program. Writes minimal HTTP headers and form; all information provided to the script in HTML form.



Funktionen Modul cgi

- `cgi.print_environ()`
Format the shell environment in HTML.
- `cgi.print_form(form)`
Format a form in HTML.
- `cgi.print_directory()`
Format the current directory in HTML.
- `cgi.print_environ_usage()`
Print a list of useful (used by CGI) environment variables in HTML.
- `cgi.escape(s, quote=False)`
Convert the characters '&', '<' and '>' in string *s* to HTML-safe sequences.
Use this if you need to display text that might contain such characters in HTML.
If the optional flag *quote* is true, the quotation mark character (") is also translated; this helps for inclusion in an HTML attribute value delimited by double quotes, as in `<a href="...">`. Note that single quotes are never translated.
Deprecated since version 3.2: This function is unsafe because *quote* is false by default and therefore deprecated. Use [html.escape\(\)](#) instead.

Fehlersuche

- das wichtige Modul `cgitb` liefert TraceBack-Funktionalität für cgi-Scripte (erst ab Python 2.2)

- Syntax:

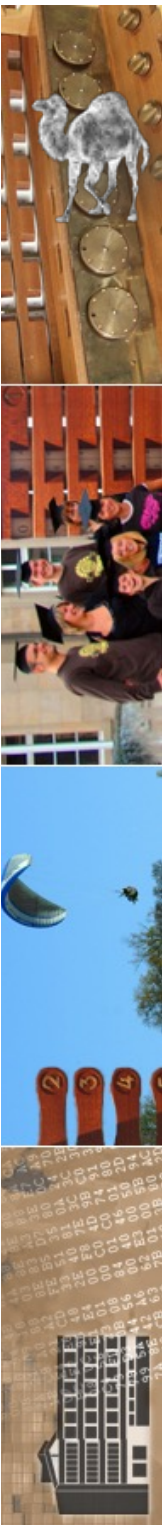
– `import cgitb; cgitb.enable()`

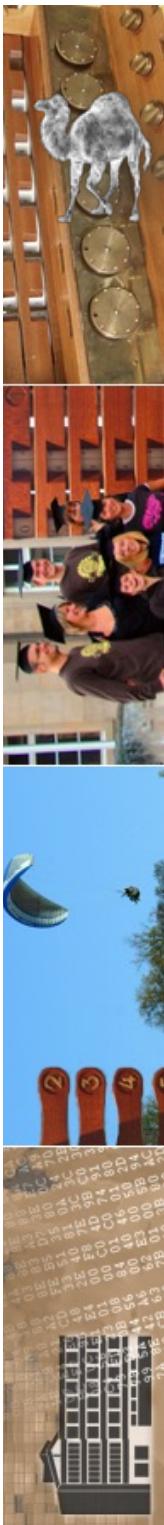
`enable([display[, logdir[, context[,
format]]]])`



Syntax Methode `enable`

- `enable([display[, logdir[, context[, format]]]])`
 - The optional argument *display* defaults to 1 and can be set to 0 to suppress sending the traceback to the browser. If the argument *logdir* is present, the traceback reports are written to files. The value of *logdir* should be a directory where these files will be placed. The optional argument *context* is the number of lines of context to display around the current line of source code in the traceback; this defaults to 5. If the optional argument *format* is "html", the output is formatted as HTML. Any other value forces plain text output. The default value is "html".





```
zrvwa01@infodienste => more cgi_2.py
#!/usr/bin/python3
#
# Grundlagen Internet-Technologien
#
# Python-CGI mit TB

### Import der Module
import cgi
import cgitb

### Aktivierung TB
cgitb.enable(display=1, logdir="/tmp")

print("""Content-Type: text/html

<!DOCTYPE html>
<html>
<head>
<title>Python-CGI mit Formularverarbeitung und TB</title>
<link rel="stylesheet" type="text/css" href="./css/webkompendium.css">
</head>
<body><p>
<center><h2>Grundlagen Internet-Technologien</h2></p><hr><p><h3>""")

##### Auswertung des Formulars #####

form = cgi.FieldStorage()

### Ausgabe aller values des Formulars
for key in form.keys():
    print('<br>Schlüssel: ',key,' hat value ',form[key].value)

#####

### Absichtlicher Fehler!
drucke("Hallo")

print ('</h3></p></center></body></html>')
zrvwa01@infodienste => █
```





Grundlagen Internet-Technologien

--> -->

NameError

Python 3.9.2: /usr/bin/python3
Sun Jun 11 11:08:39 2023

A problem occurred in a Python script. Here is the sequence of function calls leading up to the error, in the order they occurred.

/home/zrvwa01/public_html/cgi-bin/python/cgi_2.py in <module>

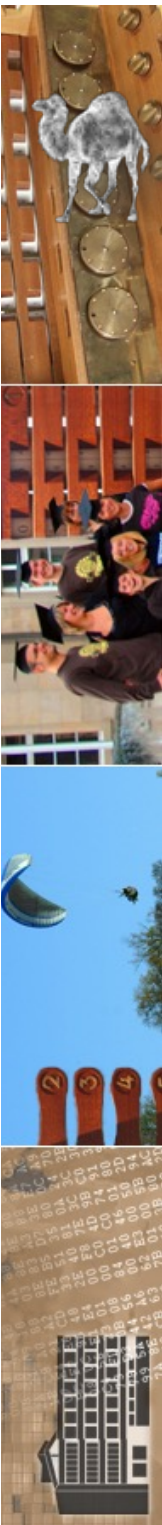
```
35
36 ### Absichtlicher Fehler!
=> 37 drucke("Hallo")
38
39 print ('</h3></p></center></body></html>')
```

drucke *undefined*

NameError: name 'drucke' is not defined
args = ("name 'drucke' is not defined",)
with_traceback = <built-in method with_traceback of NameError object>
/tmp/tmp1a7_4kji.html contains the description of this error.

URLs und Python

- ...kennen sich gut:
- Modul `urllib` stellt viele Funktionen bereit (wie PERL-Modul LWP)
 - `urlopen(url)` öffnet url wie eine Datei (zum Lesen)
 - `read()`
 - `readlines()`
 - `readline()`
 - `close()`
- erweitere Klasse: `urllib2`





jedoch

cgi — Common Gateway Interface support

Deprecated since version 3.11, removed in version 3.13.

This module is no longer part of the Python standard library. It was [removed in Python 3.13](#) after being deprecated in Python 3.11. The removal was decided in [PEP 594](#).



legacy-cgi - PyPI

pypi.org/project/legacy-cgi/

Type '/' to search projects

Hilfe Docs Sponsoren Einloggen Registrieren

legacy-cgi 2.6.4

pip install legacy-cgi

Veröffentlicht am: 27. Okt. 2025

Fork of the standard library cgi and cgi.tl modules removed in Python 3.13

Navigation

- Projekt-Beschreibung
- Veröffentlichungs-Historie
- Dateien zum Herunterladen

Projekt-Beschreibung

This is a fork of the standard library modules `cgi` and `cgi.tl`. They have been removed from the Python standard library in Python 3.13 by [PEP-594](#).

Installation

Depend upon `legacy-cgi` in your project. It is recommended to use the marker `python_version >= 3.13`, as while the package can install for older Python versions, it won't have effect with a standard Python installation as the built-in modules will take precedence.

For example, in a `requirements.txt` file:

```
legacy-cgi; python_version >= '3.13'
```

Verified details ✓

These details have been [verified by PyPI](#)

Projekt-Links

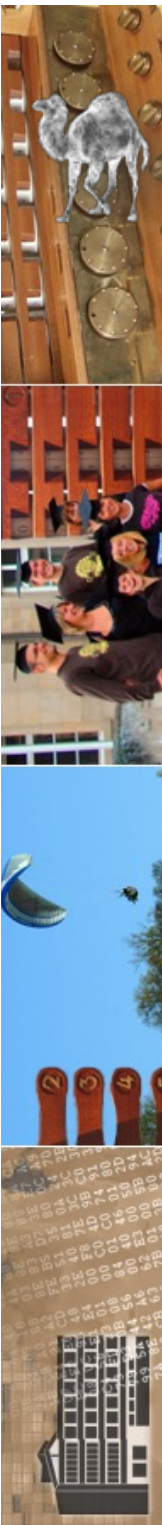
- repository

GitHub Statistics

das Modul html

- neu ab Version 3.2: Modul html

optionale „Erleichterungen“ beim Umgang mit HTML





html — HyperText Markup Language support

legacy-cgi · PyPI

docs.python.org/3/library/html.html#module-html

Previous topic

Structured Markup Processing Tools

Next topic

html.parser — Simple HTML and XHTML parser

This page

Report a bug

Improve this page

Show source

html — HyperText Markup Language support

Source code: [Lib/html/_init_.py](#)

This module defines utilities to manipulate HTML.

html.escape(*s*, *quote=True*)
Convert the characters `&`, `<` and `>` in string *s* to HTML-safe sequences. Use this if you need to display text that might contain such characters in HTML. If the optional flag *quote* is true (the default), the characters `"` and `'` are also translated; this helps for inclusion in an HTML attribute value delimited by quotes, as in `<a href="...">`. If *quote* is set to false, the characters `"` and `'` are not translated.

Added in version 3.2.

html.unescape(*s*)
Convert all named and numeric character references (e.g. `>`, `>`, `>`) in the string *s* to the corresponding Unicode characters. This function uses the rules defined by the HTML 5 standard for both valid and invalid character references, and the [list of HTML 5 named character references](#).

Added in version 3.4.

Submodules in the `html` package are:

- [html.parser](#) – HTML/XHTML parser with lenient parsing mode
- [html.entities](#) – HTML entity definitions

Python » English » 3.14.6 » 3.14.6 Documentation » The Python Standard Library » Structured Markup Processing Tools »

previous | next | modules | index

html — HyperText Markup Language support

Theme Auto | Quick search | Go

© Copyright 2001 Python Software Foundation.

This page is licensed under the Python Software Foundation License Version 2.

Examples, recipes, and other code in the documentation are additionally licensed under the Zero Clause BSD License.



das Modul `http.client`

- mit diesem Modul kann Python als *HTTP-Client* eingesetzt werden
 - zahlreiche Anwendungen wie Link-Checker, Crawler, Performance-Test, ...
 - in PERL: LWP

```
thomas@PetitMouton => python3
Python 3.8.3 (v3.8.3:6f8c8320e9, May 13 2020, 16:29:34)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> import http.client
>>>
>>> http.client.HTTPConnection('www.dfn.de')
<http.client.HTTPConnection object at 0x7ffc9011bdf0>
>>>
```



```
thomas@petitmouton-2 =>
thomas@petitmouton-2 => more HTTPclient.py
# Grundlagen Internet-Technologien
#
# HTTP-Client mit Python

import http.client

conn = http.client.HTTPSConnection("www.uni-tuebingen.de")
conn.request("GET", "/")
response1 = conn.getresponse()

print(response1.status, response1.reason, "\n\n")

data = response1.read()
print (data)
thomas@petitmouton-2 =>
```





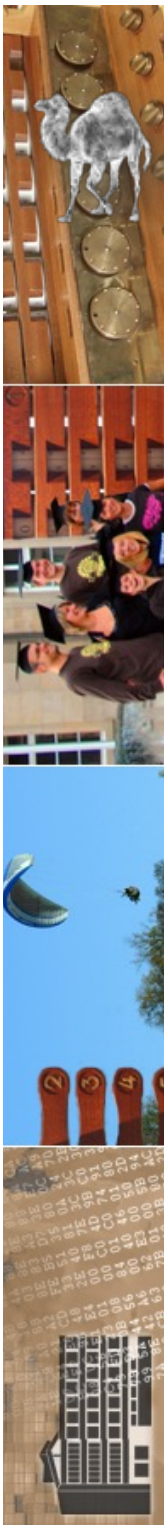
```
thomas@PetitMouton =>
thomas@PetitMouton =>
thomas@PetitMouton => python3 ./HTTPclient.py
301 Moved Permanently
```

```
b'<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">\n<html><head>\n
<title>301 Moved Permanently</title>\n</head><body>\n<h1>Moved Perma
nently</h1>\n<p>The document has moved <a href="https://uni-tuebinge
n.de/">here</a>.</p>\n</body></html>\n'
thomas@PetitMouton =>
thomas@PetitMouton =>
```



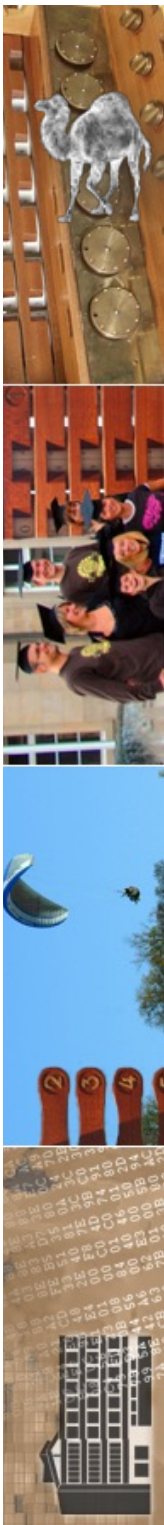
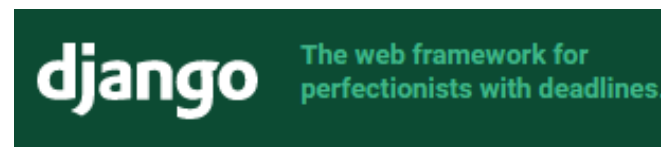
die Alternativen zu CGI

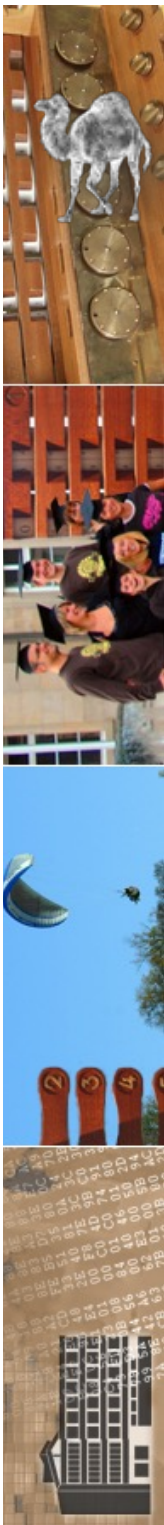
- Nachteil CGI: Performance, da bei jedem Request ein Prozess gestartet wird
- neue Ansätze
 - WSGI (Web Server Gateway Interface)
 - ASGI (Nachfolger von WSGI mit *asynchroner* Kommunikation (vgl. AJAX))
 - typisch an Frameworks gebunden
 - siehe folgendes Semester



Python Frameworks

- Python ist auch Basis mehrerer Frameworks für die Web-Entwicklung
 - → Grundlagen der Web-Entwicklung im Wintersemester
 - Django
<https://www.djangoproject.com/>
 - Flask
 - FastAPI





Webframework: Django 4.2 führt Spalten- und Tabellenkommentare ein

Das neue LTS-Release des Django-Webframeworks ist auf den PostgreSQL-Adapter Psycopg 3 vorbereitet und stellt Kommentarooptionen für Spalten und Tabellen bereit.

Lesezeit: 2 Min.  In Pocket speichern



(Bild: PureSolution/Shutterstock.com)

03.04.2023 16:44 Uhr | Developer

Von Matthias Parbel

Das in Python entwickelte Open-Source-Webframework Django hat Version 4.2 erreicht. Damit liegt ein neues LTS-Release (Long-term Support) vor, das Security-Updates und Bugfixes zum Schutz vor Datenverlust bis mindestens April 2026 garantiert. Wichtige Neuerungen in Version 4.2 umfassen die Anpassung an den PostgreSQL-Adapter Psycopg 3 und Optionen zum Anlegen von Kommentaren in Spalten und Tabellen.

Neues LTS-Release mit erweitertem Support

 Während der allgemeine Support für Django 4.1 ebenso wie der erweiterte für Version 4.0 ausläuft, übernimmt Django 4.2 die Rolle des aktuellen LTS-Release. Bis zum Jahresende liefert das Django-Entwicklungsteam Abhilfe bei





html — HyperText Markup Language · legacy-cgi · PyPI

The web framework for Python

www.djangoproject.com

django

OVERVIEW DOWNLOAD DOCUMENTATION NEWS CODE ISSUES COMMUNITY FOUNDATION ♥ DONATE

Search (⌘ K)

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

Los geht's mit Django

Django Developer Survey

The Django Software Foundation is once again partnering with JetBrains to run the 2026 Django Developer Survey 🇩🇪 Help us better understand how Django is being used around the world and guide future technical and community decisions. [Read more on the Django blog.](#)

Take the survey ✨

Meet Django

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web

Download latest release: 6.0.6

[DJANGO-DOKUMENTATION >](#)



Subscribe to the **FastAPI** and friends newsletter 📧

BlockBee sponsor Effortless crypto payments with instant payouts

FastAPI 🔍 Search 🌐 fastapi/fastapi 0.136.3 99.1k 9.4k

FastAPI Features Learn Reference FastAPI People Resources About Release Notes

FastAPI

FastAPI

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

Test passing coverage 77% pypi package v0.136.3 python 3.10 | 3.11 | 3.12 | 3.13 | 3.14

Documentation: <https://fastapi.tiangolo.com>

Source Code: <https://github.com/fastapi/fastapi>

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

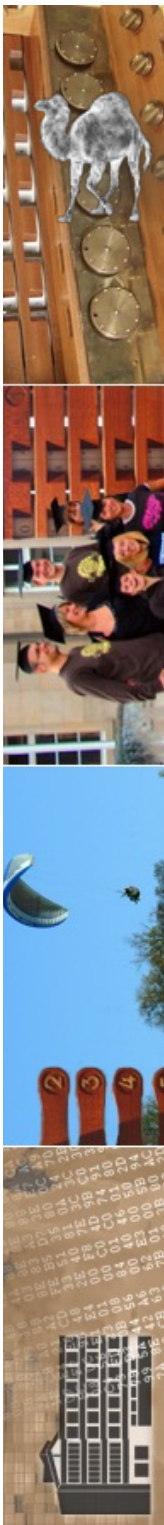
The key features are:

- **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. *

On this page

- Sponsors
 - Keystone Sponsor
 - Gold Sponsors
 - Silver Sponsors
- Opinions
- FastAPI Conf
- FastAPI mini documentary
- Typer**, the FastAPI of CLIs
- Requirements
- Installation
- Example
 - Create it
 - Run it
 - Check it
 - Interactive API docs
 - Alternative API docs
- Example upgrade
 - Interactive API docs upgrade
 - Alternative API docs upgrade
 - Recap
 - Deploy your app (optional)
 - About FastAPI Cloud
 - Deploy to other cloud providers
- Performance
- Dependencies
 - standard Dependencies

Ask AI



© 2026 Universität
Tübingen



Flask

gitproject — -bash — 89x14

```
(.venv) thomas@PetitMouton =>
(.venv) thomas@PetitMouton =>
(.venv) thomas@PetitMouton => more GitProject.py
from flask import Flask

app = Flask(__name__)

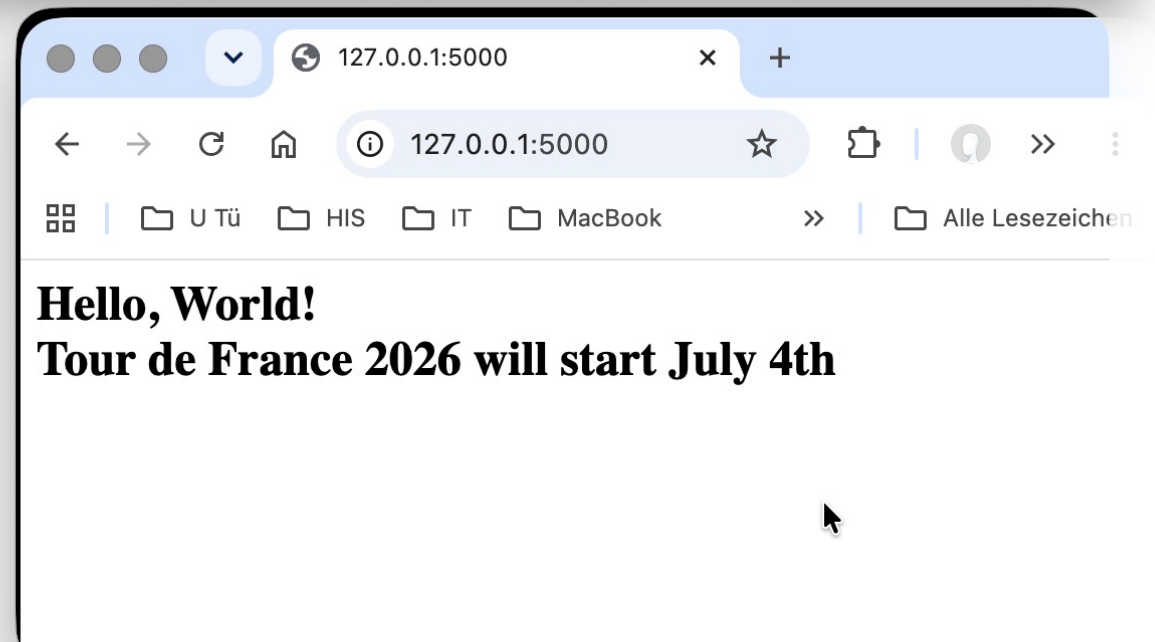
@app.route("/")
def hello_world():
    return "<p><h2>Hello, World!<br />Tour de France 2026 will start July 4th</h2></p>"
(.venv) thomas@PetitMouton =>
(.venv) thomas@PetitMouton =>
(.venv) thomas@PetitMouton =>
(.venv) thomas@PetitMouton =>
```




Flask

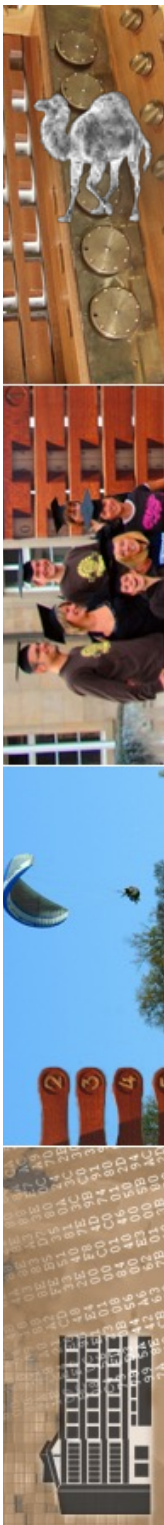
```
gitproject — flask --app GitProject.py run — 128x11

[thomas@PetitMouton ~]$ flask --app GitProject.py run
* Serving Flask app 'GitProject.py'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [10/Jun/2026 16:09:26] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [10/Jun/2026 16:09:26] "GET /favicon.ico HTTP/1.1" 404 -
```



Python und Datenbanken

- Python lässt sich ideal mit Datenbanken verbinden
 - Modul DBI bietet Abstraktionsschicht zum Datenbankmanagementsystem
- *wir behandeln DB-Anbindungen am Beispiel PHP*



...und nun...

- haben wir die Möglichkeiten von Python als cgi-Sprache kennen gelernt
 - dabei auch die Erweiterung von Python durch Module
 - und speziell das Python-Modul cgi
- als nächstes
 - **serverseitige Web-Programmierung mit PHP**



