



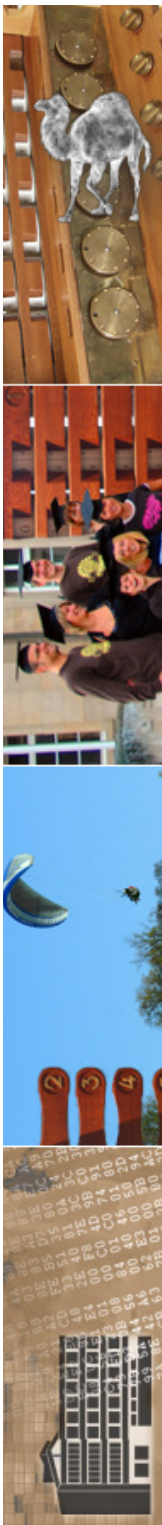
Grundlagen Internet-Technologien

INF3171

Serverseitige Web-Programmierung
mit CGI, Teil II: Python im Web

Version 1.0

06.06.2024





Aktuelles

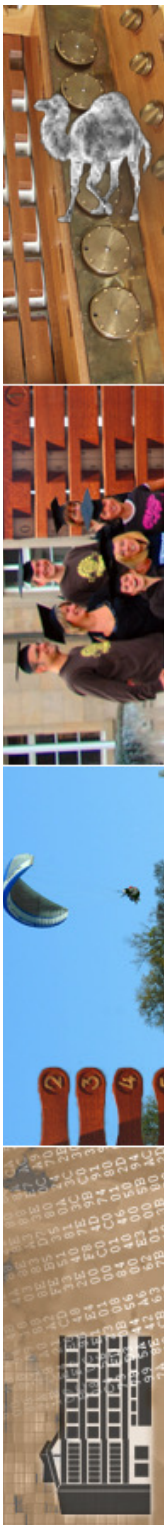
- Sommerfest auf dem Sand

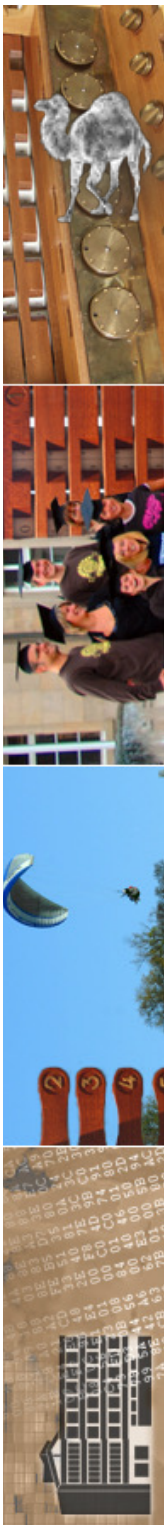
findet am 28.06.2024 ab 17h statt!!!!!!

diesen Monat ist es wieder so weit, das Sommerfest naht! Am 28.06. von 17 bis 24 Uhr veranstalten wir, die Fachschaften Informatik und Kognitionswissenschaft, wieder ein Sommerfest auf dem Sand.

Auf der Terrasse und im Garten wird es wieder allerlei zu futtern geben, von Pizza, über Grill bis Waffeln ist alles dabei! Für Getränke ist natürlich auch gesorgt: ob euer Herz gerade Bier, Mate oder Cocktails begehrt – mit oder ohne Alkohol –, bei uns seid ihr gut versorgt.

Auch um Musik wird sich den ganzen Abend gekümmert, unter anderem auch um Techno und Karaoke. Natürlich wird es auch wieder ein Volleyballturnier geben. Seht zu, wie eure Lieblingsprofs in legendären Matches gegen Fachschaftler*innen antreten, oder tobt euch danach selbst aus!





 Alert!

Chrome: Weitere Zero-Day-Lücke mit Update geschlossen

Zum dritten Mal innerhalb einer Woche aktualisiert Google den Chrome-Webbrowser. Erneut kursiert ein Exploit für eine Zero-Day-Lücke darin.



Sicherheitslücken in Google Chrome gefährden Nutzerinnen und Nutzer. (Bild: Bild erstellt mit KI in Bing Designer durch heise online / dmk)

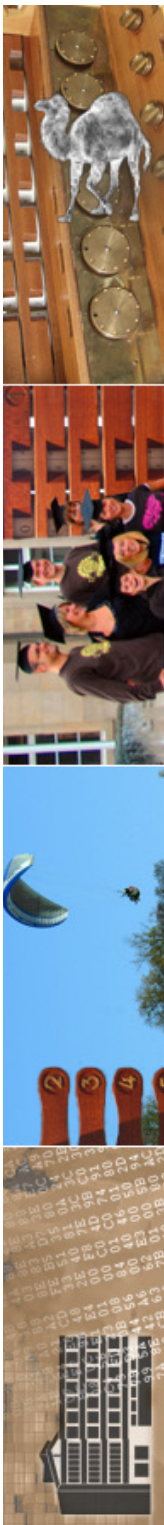
16.05.2024, 07:54 Uhr | Lesezeit: 3 Min. | Security

Von [Dirk Knop](#)

Google veröffentlicht erneut ein Notfall-Sicherheitsupdate für den Webbrowser Chrome. Für eine neue Zero-Day-Lücke im Browser kursiert abermals ein Exploit in freier Wildbahn. Dabei vollzieht der Anbieter auch den Versionssprung in den 125-Entwicklungsweig.

In der Versionsankündigung schreiben Googles Entwickler, dass die neue Version insgesamt neun Sicherheitslecks abdichtet. Knappe Informationen liefern sie lediglich für vier davon, fünf wurden demnach intern gefunden. Zwei wurden als hohes Risiko, eine als mittleres und eine als niedriger Bedrohungsgrad eingestuft.





Programmieren lernen mit Python: heiß geliebt, weil kurz und verständlich

Vom Skript auf der Konsole über Statistik bis zum neuronalen Netz: Python ist beliebt. Das liegt vor allem an der Eleganz und Verständlichkeit des Codes.

Lesezeit: 9 Min. In Pocket speichern

70



(Bild: Moritz Reichartz)

26.05.2023 13:30 Uhr | c't Magazin

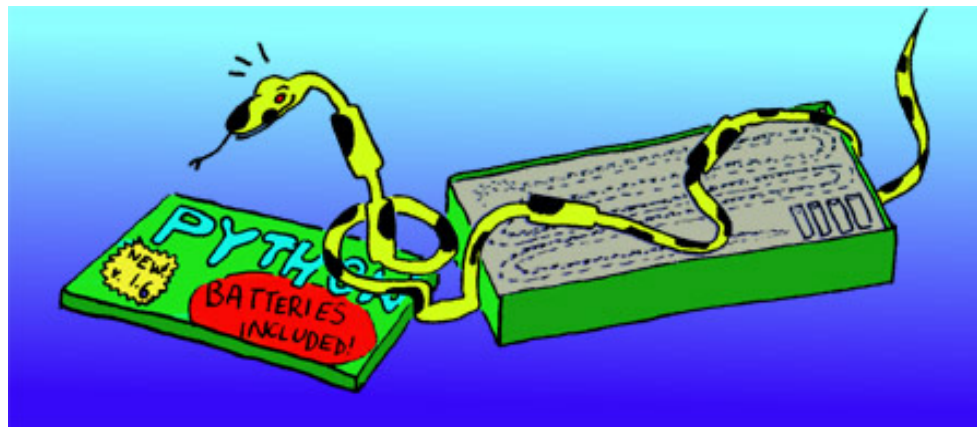
Von Pina Merkert

Programmieren ist eine mentale Herausforderung: Jedes Programm besteht aus Bausteinen, die man im Kopf wie ein Uhrwerk zusammenfügen muss, damit am Ende alle Teile perfekt ineinandergreifen. Das Ergebnis dieses Denkprozesses so aufzuschreiben, dass der Computer es ausführen kann, ist nur der letzte Schritt dieser Arbeit. Eine Programmiersprache sollte dabei vor allem nicht im Weg sein. Python ist dafür perfekt, weil die Sprache wahlweise prozedural, funktional oder objektorientiert arbeitet; so wie man es gerade braucht. Schnell zu lauffähigem Code zu kommen, ist das wichtigste Ziel und dem opfert Python sowohl die Prüfung von Datentypen als auch Klammern. Beispielsweise definiert die folgende Zeile in einem Schritt drei Variablen:



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

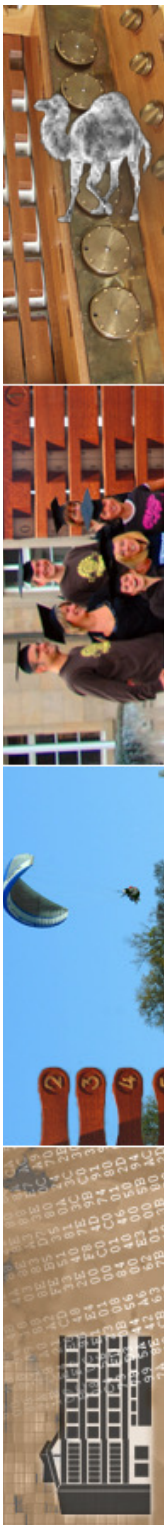




noch mehr: Paketmanager

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

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



pip documentation v23.1.2

🔍 Search

[Getting Started](#)

[Installation](#)

[User Guide](#)

[Topic Guides](#) ▾

[Reference](#) ▾

[Commands](#) ▾

PROJECT

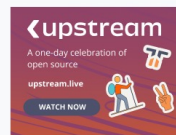
[Development](#) ▾

[UX Research & Design](#)

[Changelog](#)

[Code of Conduct](#) ↗

[GitHub](#) ↗



Upstream: Celebrating open source. WATCH NOW

Ad by EthicalAds · ⓘ



v: **stable**

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

Next >
[Getting Started](#)

Copyright © The pip developers
Made with [Sphinx](#) and [@pradyunsg's Furo](#)





pip documentation
v23.1.2

🔍 Search

- [Getting Started](#)
- [Installation](#)
- [User Guide](#)
- [Topic Guides](#) ▾
- [Reference](#) ▾
- [Commands](#) ▾
- PROJECT
- [Development](#) ▾
- [UX Research & Design](#)
- [Changelog](#)
- [Code of Conduct](#) ↗
- [GitHub](#) ↗



Monitor K8s Applications Effortlessly Get 100% K8s production visibility **Start Free**

Ad by EthicalAds... v: stable

User Guide



Running pip

pip is a command line program. When you install pip, a `pip` command is added to your system, which can be run from the command prompt as follows:

[Unix/macOS](#) [Windows](#)

```
python -m pip <pip arguments>
```

`python -m pip` executes pip using the Python interpreter you specified as `python`. So `/usr/bin/python3.7 -m pip` means you are executing pip for your interpreter located at `/usr/bin/python3.7`.

Installing Packages

pip supports installing from [PyPI](#), version control, local projects, and directly from distribution files.

The most common scenario is to install from [PyPI](#) using [Requirement Specifiers](#)

[Unix/macOS](#) [Windows](#)

```
python -m pip install SomePackage # latest version
python -m pip install SomePackage==1.0.4 # specific version
python -m pip install 'SomePackage>=1.0.4' # minimum version
```

For more information and examples, see the [pip install](#) reference.

Basic Authentication Credentials

This is now covered in [Authentication](#).

netrc Support

This is now covered in [Authentication](#).





Beispiel: jupyter Notebook

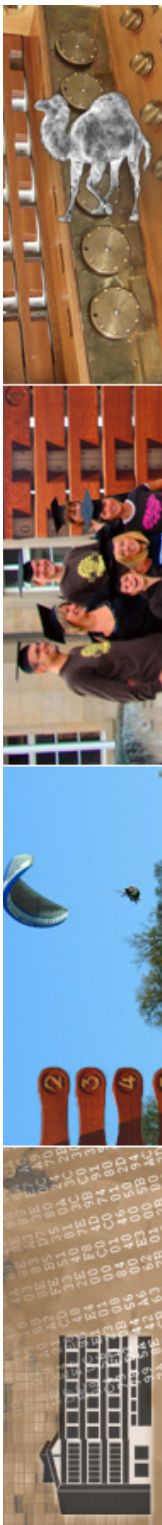
Jupyter Notebook

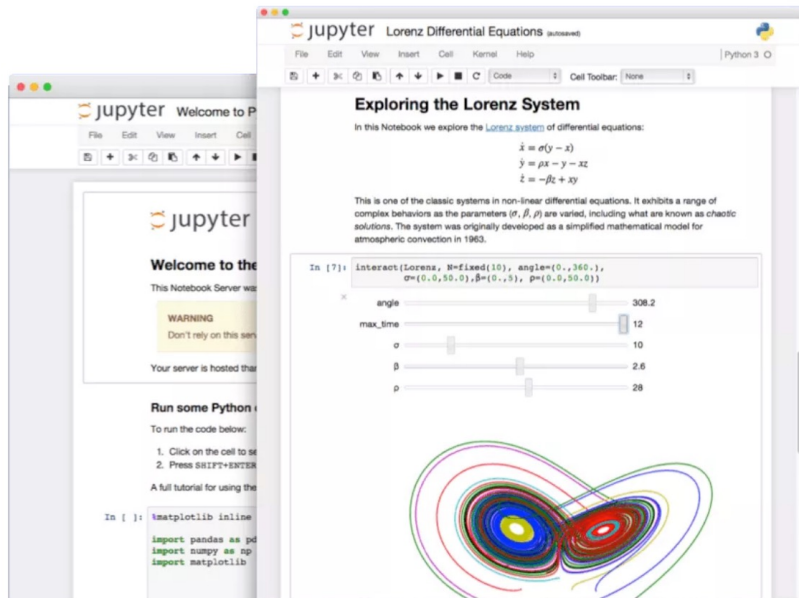
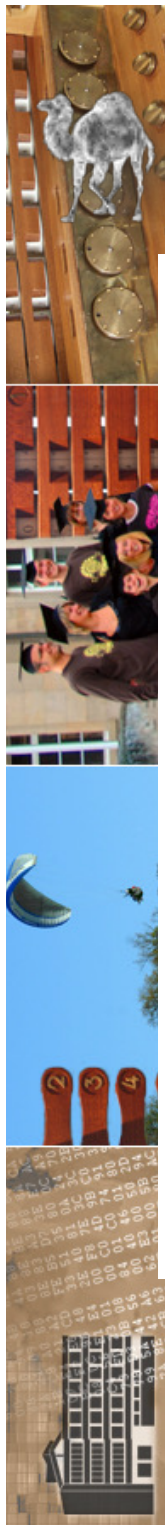
Install the classic Jupyter Notebook with:

```
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](#).



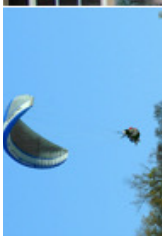
Interactive output

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

```

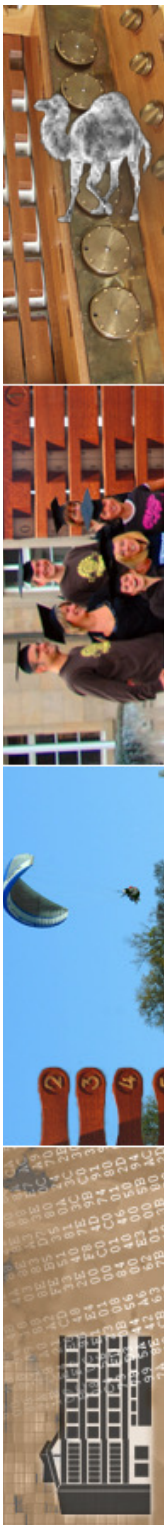
Installing collected packages: webencodings, wcwidth, pure-eval, PtyProcess, PickleShare, mistune, ipython-genutils, fastjsonschema, executing, backcall, appnope, websocket-client, webcolors, uri-template, traitlets, tornado, tinycss2, soupsieve, sniffio, six, Send2Trash, rfc3986-validator, pyzmq, pyyaml, python-json-logger, pyrsistent, pygments, pyparsing, 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-23.1 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 pyparsing-2.21 pygments-2.15.1 pyrsistent-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





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



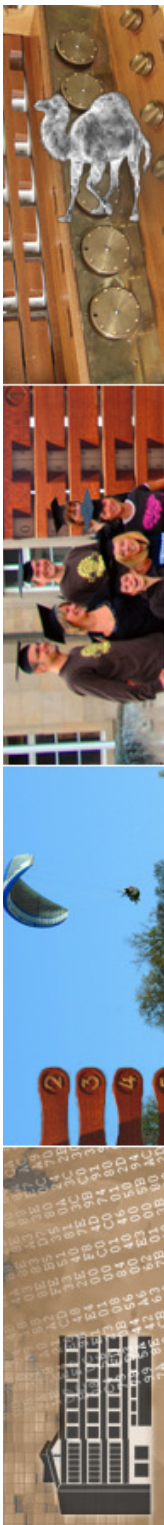


Dokumentation zu den Modulen

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

gibt es Informationen zu allen Modulen des Python-Systems

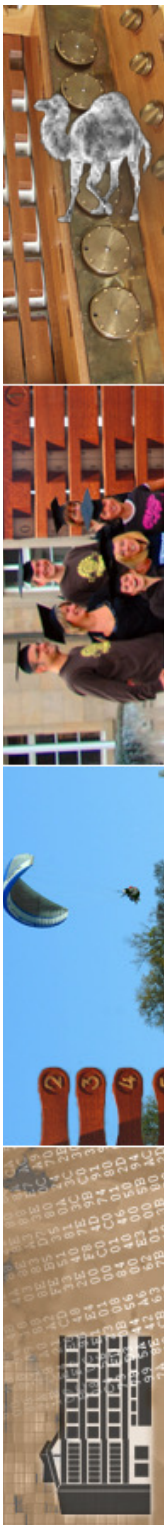
- Hinweis: umfangreich!



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__ | <i>Future statement definitions</i> |
| __main__ | <i>The environment where top-level code is run. Covers command-line interfaces, import-time behavior, and <code>__name__ == '__main__'</code>.</i> |
| _thread | <i>Low-level threading API.</i> |
| a | |
| abc | <i>Abstract base classes according to :pep: `3119`.</i> |
| aifc | Deprecated: <i>Read and write audio files in AIFF or AIFC format.</i> |
| argparse | <i>Command-line option and argument parsing library.</i> |
| array | <i>Space efficient arrays of uniformly typed numeric values.</i> |
| ast | <i>Abstract Syntax Tree classes and manipulation.</i> |
| asynchat | Deprecated: <i>Support for asynchronous command/response protocols.</i> |
| asyncio | <i>Asynchronous I/O.</i> |
| asyncore | Deprecated: <i>A base class for developing asynchronous socket handling services.</i> |
| atexit | <i>Register and execute cleanup functions.</i> |
| audioop | Deprecated: <i>Manipulate raw audio data.</i> |
| b | |
| base64 | <i>RFC 4648: Base16, Base32, Base64 Data Encodings; Base85 and Ascii85</i> |
| bdb | <i>Debugger framework.</i> |
| binascii | <i>Tools for converting between binary and various ASCII-encoded binary representations.</i> |
| binhex | <i>Encode and decode files in binhex4 format.</i> |
| bisect | <i>Array bisection algorithms for binary searching.</i> |
| builtins | <i>The module that provides the built-in namespace.</i> |
| bz2 | <i>Interfaces for bzip2 compression and decompression.</i> |





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

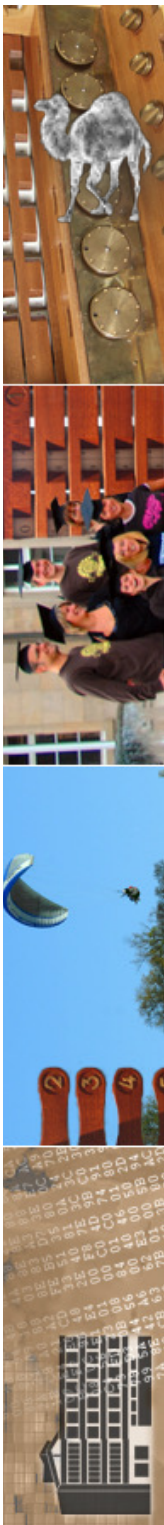




Table of Contents

- string** — Common string operations
 - String constants
 - Custom String Formatting
 - Format String Syntax
 - Format Specification Mini-Language
 - Format examples
 - Template strings
 - Helper functions

Previous topic

Text Processing Services

Next topic

re — Regular expression operations

This Page

Report a Bug
Show Source

string — Common string operations

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

See also: [Text Sequence Type — str](#)

[String Methods](#)

String constants

The constants defined in this module are:

string.ascii_letters

The concatenation of the [ascii_lowercase](#) and [ascii_uppercase](#) constants described below. This value is not locale-dependent.

string.ascii_lowercase

The lowercase letters `'abcdefghijklmnopqrstuvwxyz'`. This value is not locale-dependent and will not change.

string.ascii_uppercase

The uppercase letters `'ABCDEFGHIJKLMNOPQRSTUVWXYZ'`. This value is not locale-dependent and will not change.

string.digits

The string `'0123456789'`.

string.hexdigits

The string `'0123456789abcdefABCDEF'`.

string.octdigits

The string `'01234567'`.





Modul `math`

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

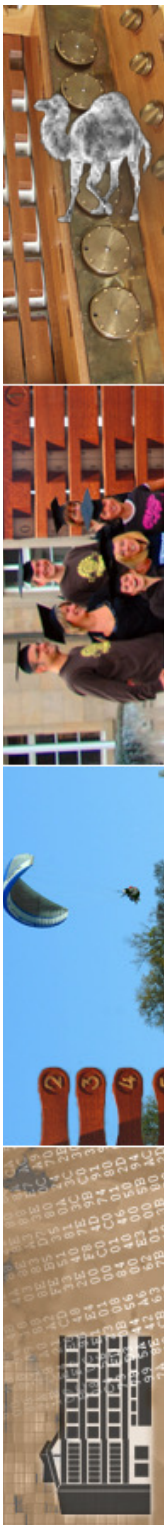




Table of Contents

math — Mathematical functions

- Number-theoretic and representation functions
- Power and logarithmic functions
- Trigonometric functions
- Angular conversion
- Hyperbolic functions
- Special functions
- Constants

Previous topic

numbers — Numeric abstract base classes

Next topic

cmath — Mathematical functions for complex numbers

This Page

Report a Bug
Show Source

math — Mathematical functions

This module provides access to the mathematical functions defined by the C standard.

These functions cannot be used with complex numbers; use the functions of the same name from the `cmath` module if you require support for complex numbers. The distinction between functions which support complex numbers and those which don't is made since most users do not want to learn quite as much mathematics as required to understand complex numbers. Receiving an exception instead of a complex result allows earlier detection of the unexpected complex number used as a parameter, so that the programmer can determine how and why it was generated in the first place.

The following functions are provided by this module. Except when explicitly noted otherwise, all return values are floats.

Number-theoretic and representation functions

`math.ceil(x)`

Return the ceiling of x , the smallest integer greater than or equal to x . If x is not a float, delegates to `x.__ceil__()`, which should return an `Integral` value.

`math.comb(n, k)`

Return the number of ways to choose k items from n items without repetition and without order.

Evaluates to $n! / (k! * (n - k)!)$ when $k \leq n$ and evaluates to zero when $k > n$.

Also called the binomial coefficient because it is equivalent to the coefficient of k -th term in polynomial expansion of the expression $(1 + x) ** n$.

Raises `TypeError` if either of the arguments are not integers. Raises `ValueError` if either of the arguments are negative.

New in version 3.8.

`math.copysign(x, y)`

Return a float with the magnitude (absolute value) of x but the sign of y . On platforms that support signed zeros, `copysign(1.0, -0.0)` returns `-1.0`.

`math.fabs(x)`

Table of Contents

cmath — Mathematical functions for complex numbers

- Conversions to and from polar coordinates
- Power and logarithmic functions
- Trigonometric functions
- Hyperbolic functions
- Classification functions
- Constants

Previous topic

math — Mathematical functions

Next topic

decimal — Decimal fixed point and floating point arithmetic

This Page

Report a Bug
Show Source

cmath — Mathematical functions for complex numbers

This module provides access to mathematical functions for complex numbers. The functions in this module accept integers, floating-point numbers or complex numbers as arguments. They will also accept any Python object that has either a `__complex__()` or a `__float__()` method: these methods are used to convert the object to a complex or floating-point number, respectively, and the function is then applied to the result of the conversion.

Note: On platforms with hardware and system-level support for signed zeros, functions involving branch cuts are continuous on *both* sides of the branch cut: the sign of the zero distinguishes one side of the branch cut from the other. On platforms that do not support signed zeros the continuity is as specified below.

Conversions to and from polar coordinates

A Python complex number `z` is stored internally using *rectangular* or *Cartesian* coordinates. It is completely determined by its *real part* `z.real` and its *imaginary part* `z.imag`. In other words:

```
z == z.real + z.imag*1j
```

Polar coordinates give an alternative way to represent a complex number. In polar coordinates, a complex number `z` is defined by the modulus `r` and the phase angle `phi`. The modulus `r` is the distance from `z` to the origin, while the phase `phi` is the counterclockwise angle, measured in radians, from the positive x-axis to the line segment that joins the origin to `z`.

The following functions can be used to convert from the native rectangular coordinates to polar coordinates and back.

`cmath.phase(x)`

Return the phase of `x` (also known as the *argument* of `x`), as a float. `phase(x)` is equivalent to `math.atan2(x.imag, x.real)`. The result lies in the range $[-\pi, \pi]$, and the branch cut for this operation lies along the negative real axis, continuous from above. On systems with support for signed zeros (which includes most systems in current use), this means that the sign of the result is the same as the sign of `x.imag`, even when `x.imag` is zero:



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)

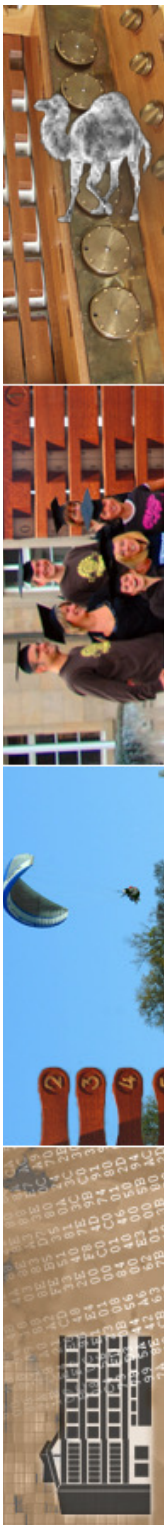




Table of Contents

time — Time access and conversions

- Functions
- Clock ID Constants
- Timezone Constants

Previous topic

io — Core tools for working with streams

Next topic

argparse — Parser for command-line options, arguments and sub-commands

This Page

Report a Bug
Show Source

time — Time access and conversions

This module provides various time-related functions. For related functionality, see also the [datetime](#) and [calendar](#) modules.

Although this module is always available, not all functions are available on all platforms. Most of the functions defined in this module call platform C library functions with the same name. It may sometimes be helpful to consult the platform documentation, because the semantics of these functions varies among platforms.

An explanation of some terminology and conventions is in order.

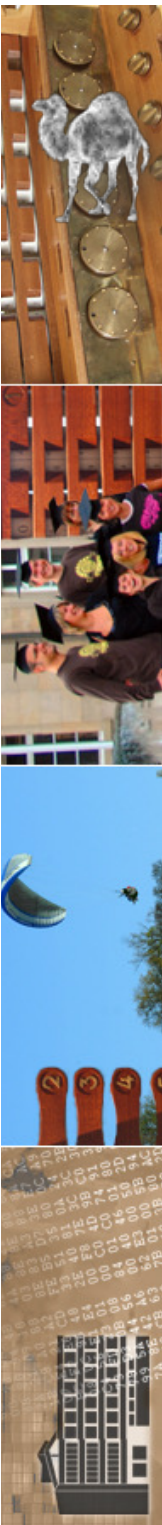
- The *epoch* is the point where the time starts, and is platform dependent. For Unix, the epoch is January 1, 1970, 00:00:00 (UTC). To find out what the epoch is on a given platform, look at `time.gmtime(0)`.
- The term *seconds since the epoch* refers to the total number of elapsed seconds since the epoch, typically excluding [leap seconds](#). Leap seconds are excluded from this total on all POSIX-compliant platforms.
- The functions in this module may not handle dates and times before the epoch or far in the future. The cut-off point in the future is determined by the C library; for 32-bit systems, it is typically in 2038.
- **Year 2000 (Y2K) issues:** Python depends on the platform's C library, which generally doesn't have year 2000 issues, since all dates and times are represented internally as seconds since the epoch. Function `strptime()` can parse 2-digit years when given `%y` format code. When 2-digit years are parsed, they are converted according to the POSIX and ISO C standards: values 69-99 are mapped to 1969-1999, and values 0-68 are mapped to 2000-2068.
- UTC is Coordinated Universal Time (formerly known as Greenwich Mean Time, or GMT). The acronym UTC is not a mistake but a compromise between English and French.





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.

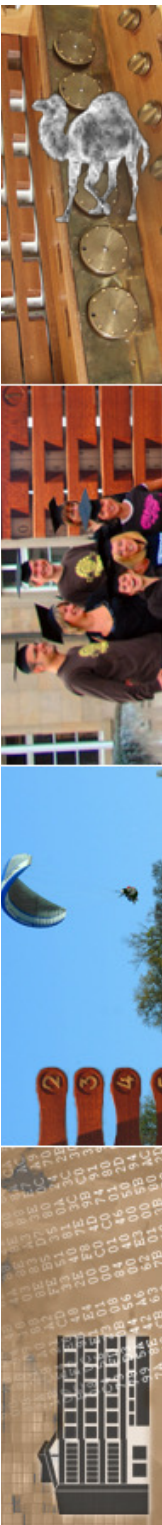




Table of Contents

os — Miscellaneous operating system interfaces

- File Names, Command Line Arguments, and Environment Variables
- Process Parameters
- File Object Creation
- File Descriptor Operations
 - Querying the size of a terminal
 - Inheritance of File Descriptors
- Files and Directories
 - Linux extended attributes
- Process Management
- Interface to the scheduler
- Miscellaneous System Information
- Random numbers

Previous topic

Generic Operating System Services

Next topic

io — Core tools for working with streams

This Page

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

os — Miscellaneous operating system interfaces

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

This module provides a portable way of using operating system dependent functionality. If you just want to read or write a file see `open()`, if you want to manipulate paths, see the `os.path` module, and if you want to read all the lines in all the files on the command line see the `fileinput` module. For creating temporary files and directories see the `tempfile` module, and for high-level file and directory handling see the `shutil` module.

Notes on the availability of these functions:

- The design of all built-in operating system dependent modules of Python is such that as long as the same functionality is available, it uses the same interface; for example, the function `os.stat(path)` returns stat information about `path` in the same format (which happens to have originated with the POSIX interface).
- Extensions peculiar to a particular operating system are also available through the `os` module, but using them is of course a threat to portability.
- All functions accepting path or file names accept both bytes and string objects, and result in an object of the same type, if a path or file name is returned.

Note: All functions in this module raise `OSError` in the case of invalid or inaccessible file names and paths, or other arguments that have the correct type, but are not accepted by the operating system.

exception `os.error`

An alias for the built-in `OSError` exception.

`os.name`

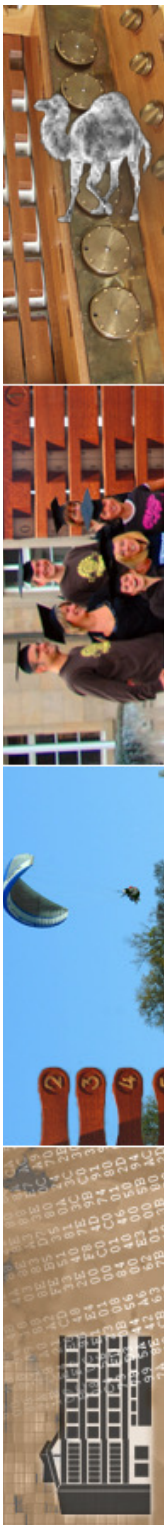
The name of the operating system dependent module imported. The following names have currently been registered:





Situation

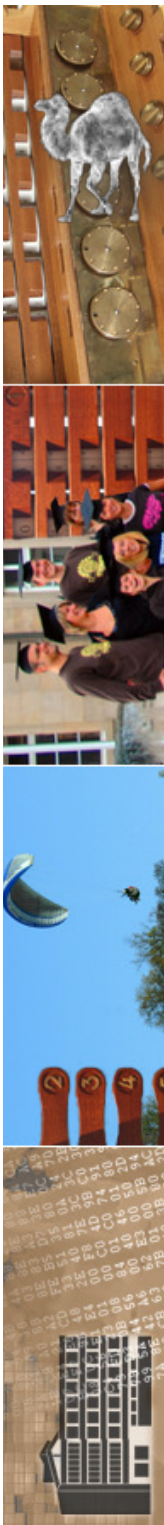
- bisher: Python ist einfache und nützliche (Systemadministration) Programmiersprache
- nun: wir nutzen Python als **CGI-Sprache**
- Fragen:
 - keine direkt ausführbaren Codes mit Python
 - was ist alles zu tun für CGI mit Python
 - **Formularverarbeitung** mit Python
 - **Generation von HTML** mit Python





Python und das Internet

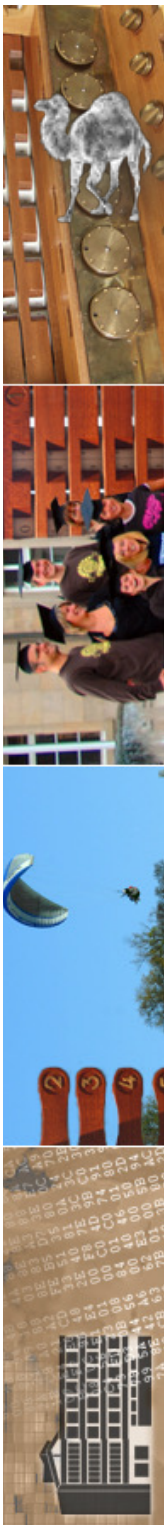
- Python ist - ähnlich wie Java - eine *für das Internet optimierte* Programmiersprache
- schon zum Standard-Umfang gehört eine sehr große Bibliothek nützlicher Klassen - ähnlich wie das Paket `java.net`
- dabei u.a. die Klasse `cgi` (vergleichbar ist etwa das PERL-Modul `CGI.pm`)





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





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](#)

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](#)



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:

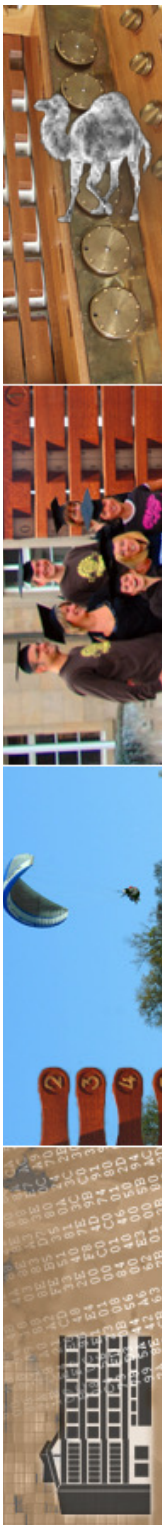
erste (!!!) Zeile im Scriptprogramm ist ein besonderer Kommentar:

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

- zusätzlich auf Unix x-bit setzen:

```
chmod +x <pythonscript>
```

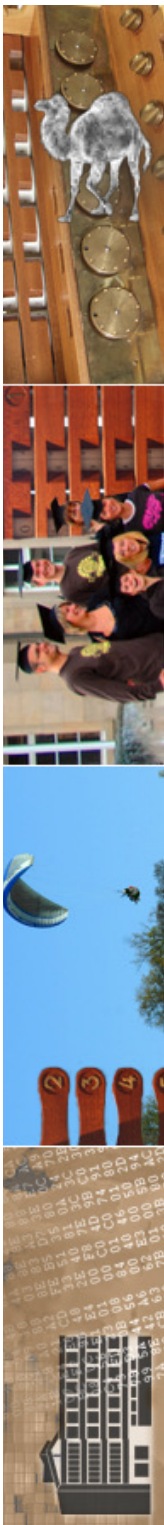
- und schon ist's ausführbar!
- gilt für *alle* *Scriptsprachen*





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 =>

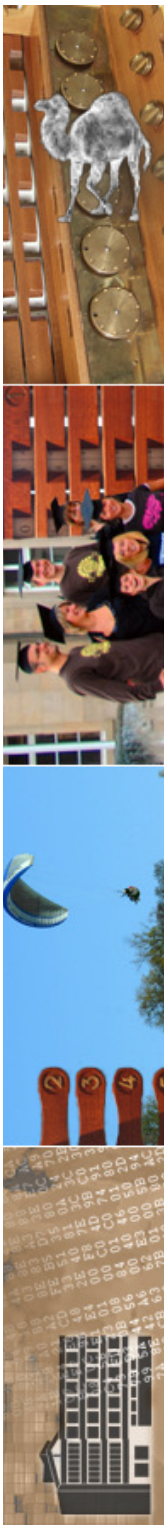
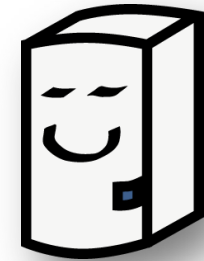
```





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 =>
```



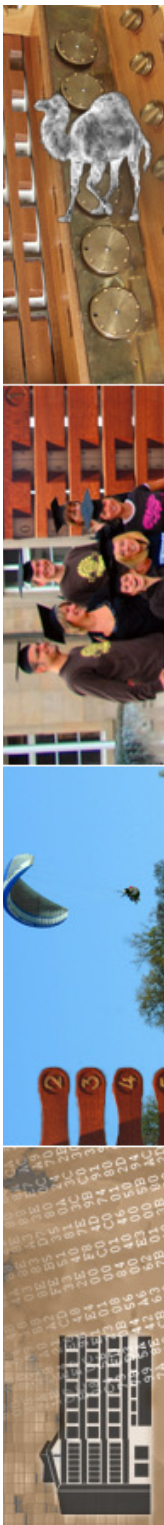


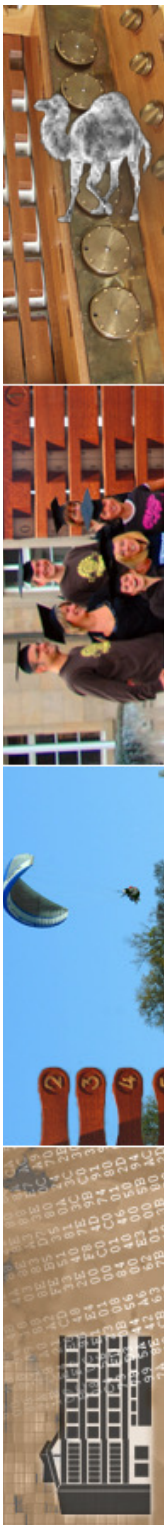
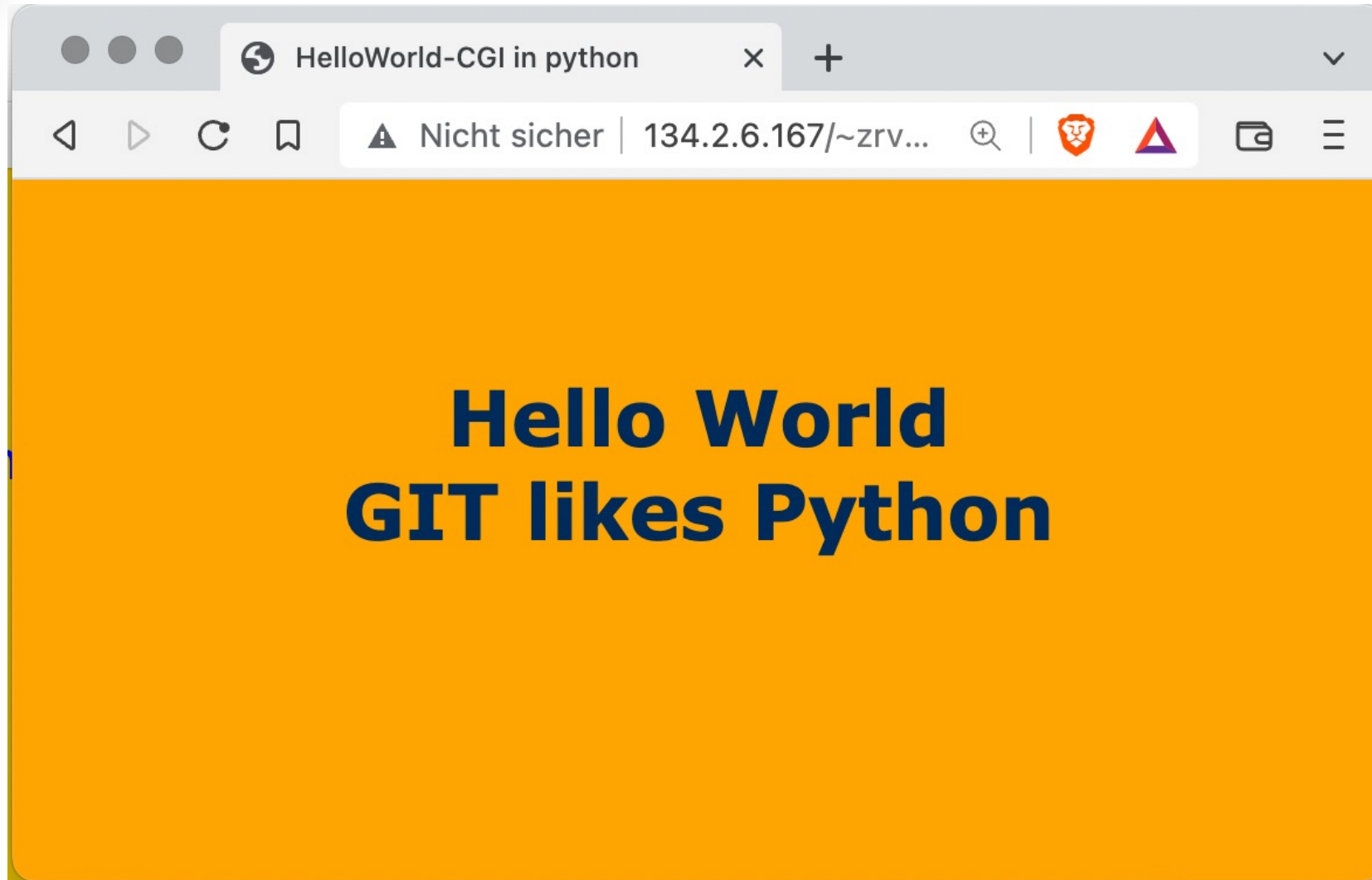
Ausführen im Web

- über die Default-cgi-URL

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

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





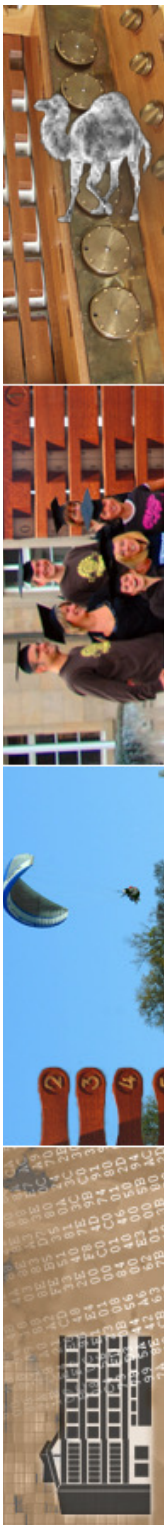


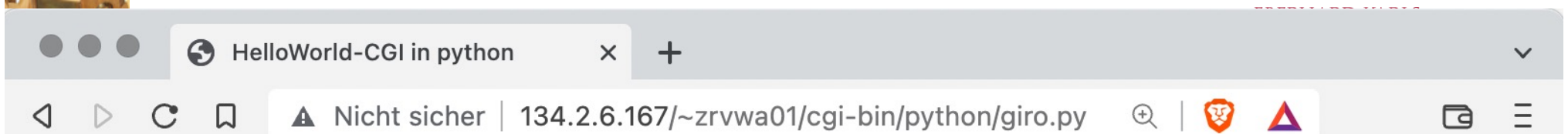
Here-Documents

- einbetten größerer „Raw-Texte“ in Python mittels

dreier Anführungszeichen ““ oder ““““

- für CGIs praktisch

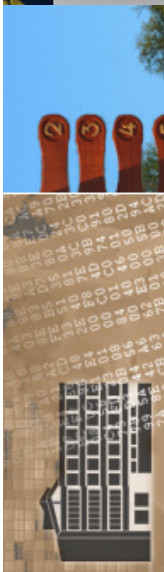


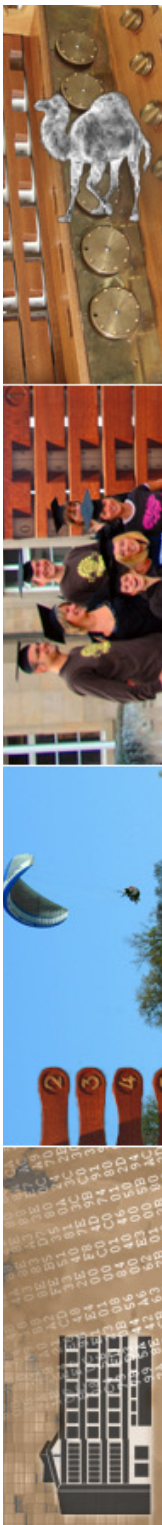


Hello Giro d'Italia Jay Hindley wins

```
zrvwa01@infodienste => more giro.py
#!/usr/bin/python3
#
# Grundlagen Internet-Technologien
#
# grundlegendes CGI mit Python
#
# here-Dokument in Python - Begrenzt durch '''
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 Giro d'Italia<BR />Jay Hindley wins
  </H3></CENTER><BR />
</BODY>
</HTML>''')
zrvwa01@infodienste =>
```

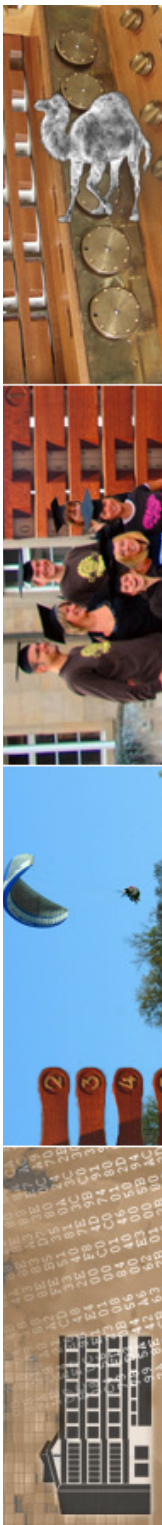






erste Dynamik

- 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/webkompodium.css">
  </HEAD>
  <BODY">

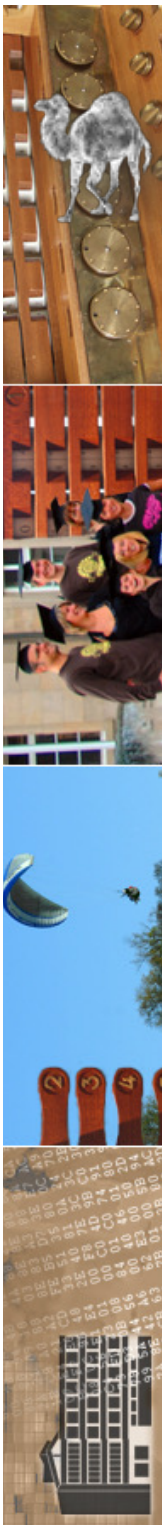
zeit = time.time()

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 =>
```





Nicht sicher | 134.2.6.167/~zrvwa01/cgi-bin/python/time.py



VPN

Aktualisieren

Grundlagen Internet-Technologien

Zeit auf dem Server: 1686474325.8802855

06/11/2023 11:05:25





Python Modul cgi

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

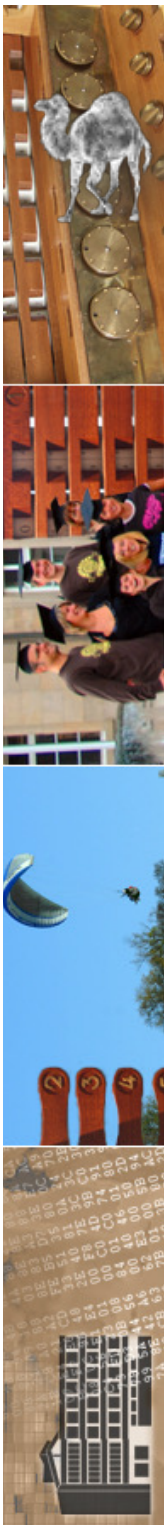




Table of Contents

- 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

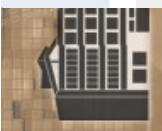
A CGI script is invoked by an HTTP server, usually to process user input submitted through an HTML `<FORM>` or `<ISINDEX>` element.

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:

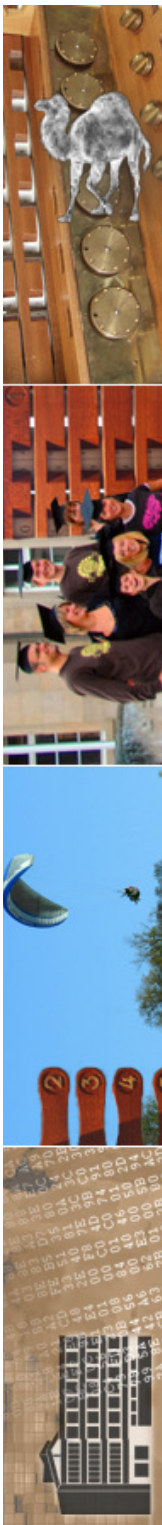
```
print("Content-Type: text/html")    # HTML is following
print()                          # blank line, end of headers
```





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 ``. 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.



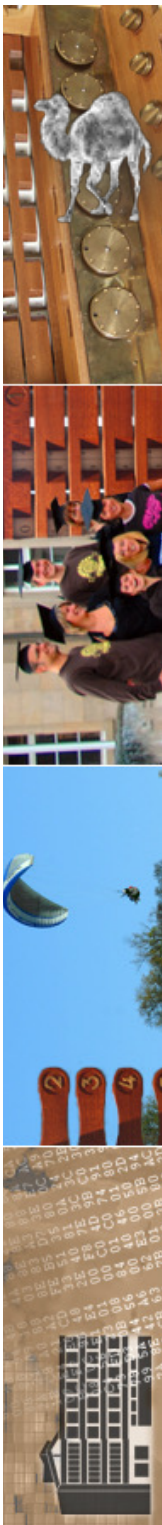
Fehlersuche

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

- Syntax:

```
– import cgitb; cgitb.enable()
```

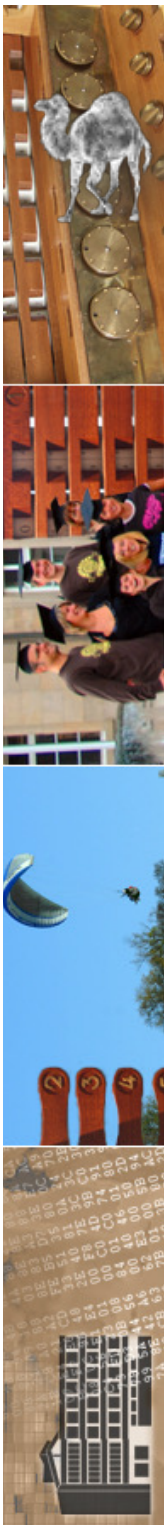
```
enable([display[, logdir[, context[,  
format]]]])
```

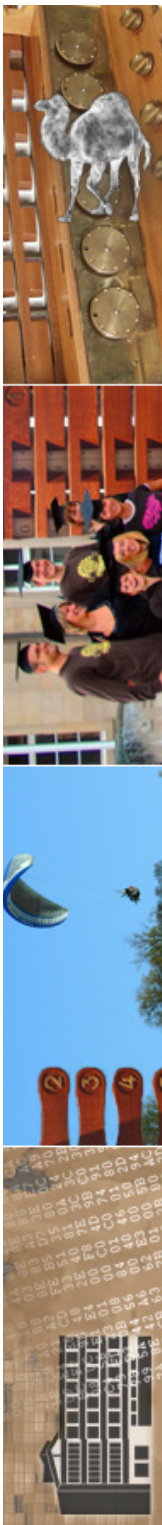




Syntax Methode `enable`

- `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 =>
```





134.2.6.167/~zrvwa01/cgi-bin/python/cgi_2.py

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>

```

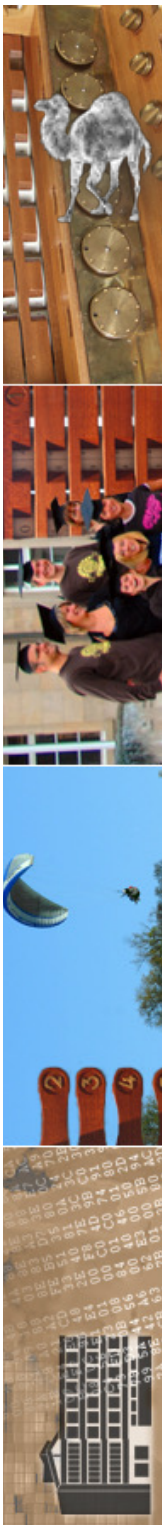
35
36 ### Absichtlicher Fehler!
=> 37 drucke("Hallo")
38
39 print ('</h3></p></center></body></html>')
drucke undefined
    
```

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()`
- erweiterbare Klasse: `urllib2`

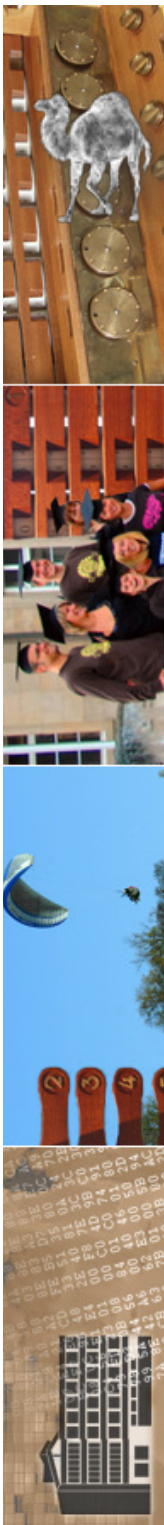




das Modul html

- neu ab Version 3.2: Modul html

optionale „Erleichterungen“ beim Umgang mit HTML



Previous topic

Structured Markup
Processing Tools

Next topic

`html.parser` — Simple
HTML and XHTML parser

This Page

Report a Bug
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 characters `"` and `'` are also translated; this helps for inclusion in an HTML attribute value delimited by quotes, as in ``.

New 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](#).

New in version 3.4.

Submodules in the `html` package are:

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



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'
```

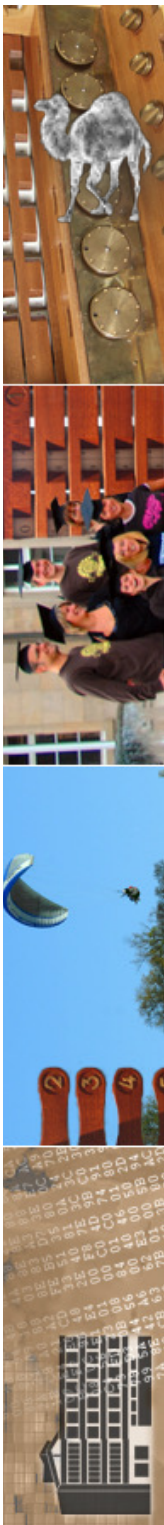
```
thomas@PetitMouton =>
thomas@PetitMouton =>
```





Python und Datenbanken

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





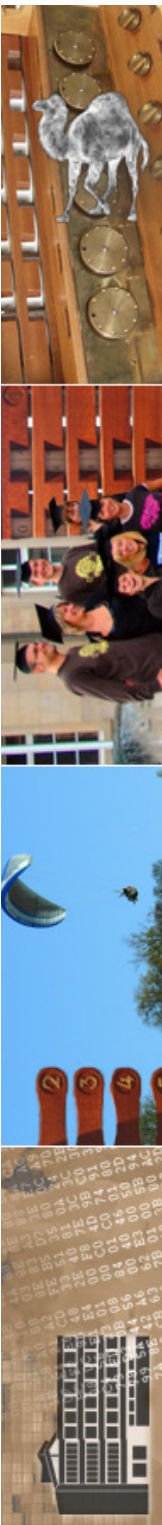
Python Frameworks

- Python ist auch Basis mehrerer Frameworks für die Web-Entwicklung
 - → Grundlagen der Web-Entwicklung im Wintersemester

- Django
<https://www.djangoproject.com/>

django The web framework for perfectionists with deadlines.

- TurboGears





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)

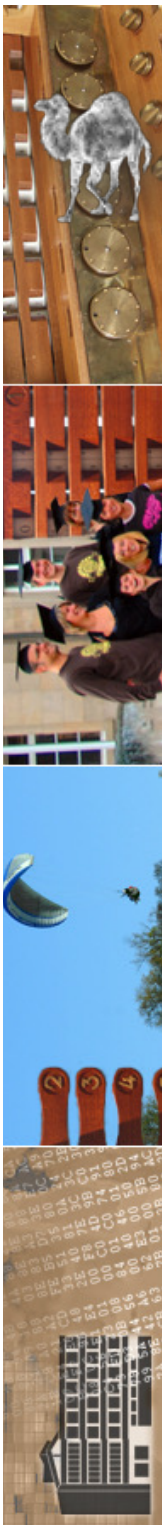
03.04.2023 16:44 Uhr | Developer

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





Until June 15, 2024, get PyCharm at 30% off. All money goes to the DSF!

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

Get started with Django

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 development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.



Ridiculously fast.

Django was designed to help developers take applications from concept to completion as quickly as possible.

Bildschirmfoto

Download latest release: 5.0.6

[DJANGO DOCUMENTATION](#)

Support Django!



Las Vegas Personal Injury Lawyer donated to the Django Software Foundation to support Django development. Donate today!

Latest news

...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**

