

# Python

- Python scheint sich als Sprache für **robuste** Scripts zu etablieren.
- So wie man die Shell als Klebstreifen für's Verbinden von System-Programmen und -Dateien, kategorisieren kann, so kann man Python als soliden, aber im Gebrauch aufwändigeren Zement ansehen.
- Die Integration braucht signifikant mehr Zeit, aber das Resultat ist widerstandsfähiger.
- viel mehr cross-platform als Shell

## hello world

```
print("Hello")
```

## REPL/Shell

```
$ python3
...
>>> print("Foo")
Foo

$ python3
...
Python 3.7.3 (default, Jul 25 2020, 13:03:44)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
Type help() for interactive help, or help(object) for help about object.
>>> help()
Welcome to Python 3.7's help utility!
...
help> print
The "print" statement
*****
...

```

- ipython mit Vervollständigung, Hilfe zu Objekten etc.
- diverse IDEs für Python

## Docu

- <http://docs.python.org>

## Syntax

- sehr einfach \* block:

```
if 2>1:
    print("Wer hätte das gedacht!")

def do_stuff( param1, param2 ):
    print(param1)
    return 42

do_stuff( "a", "b" )
```

- Achtung: Space/Indent hat syntaktischen Einfluss!
- wie andere moderne Skript Sprachen auch objekt-orientiert, aber nicht strikt
- Strukturierung in Module:

```
import os
```

## ansible

- ansible ist in Python geschrieben
- man kann eigene ansible Module in Python schreiben
- siehe *ansible\_demo/library/less dpkg\_dep*