



Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences



Fachkonferenz Mathematik 2023

Python: Einsatzmöglichkeiten in Prüfungsumgebungen

Ronny Standtke

► Architektur, Holz und Bau ► Fachbereich Holz ► Bachelor Holztechnik

Vorstellung

- Ronny Standtke
- 2000 Studium Informatik und Maschinenbau an der TU Dresden und Universität Helsinki
- 2012 Doktorarbeit im Bereich IT-Sicherheit an der Universität Fribourg
- 2000 - 2007 Berater und Entwickler bei secunet Security Networks AG (in DE und CH)
- 2007 - 2021 Dozent an der Fachhochschule Nordwestschweiz:
 - Objektorientierte Programmierung
 - Systemadministration
 - Betriebssysteme
 - Programmierzentrum
 - Team- und Fachcoach für «physische» Informatik-Projekte (Raspberry-Pi, Arduino, 3D-Drucker, ...)
- seit 2018 Bereichsleiter an der Forschungsstelle Digitale Nachhaltigkeit der Universität Bern
 - Entwicklung von Lern- und Prüfungsumgebungen
- seit 2021 Professor für digitale Fertigung an der BFH



ronny.standtke@bfh.ch
079 786 81 82

Agenda

- Python im Unterricht
 - Perspektiven aus der Hochschule (Architektur, Holz und Bau)
- Python in Prüfungssituationen
 - Erfahrungen mit Lernstick / CAMPLA

Programmieren? Im ersten Semester!? Warum!?!?

Programmieren ist der Kern von vielen Tätigkeiten, mittlerweile auch von Nichtinformatiker:innen.

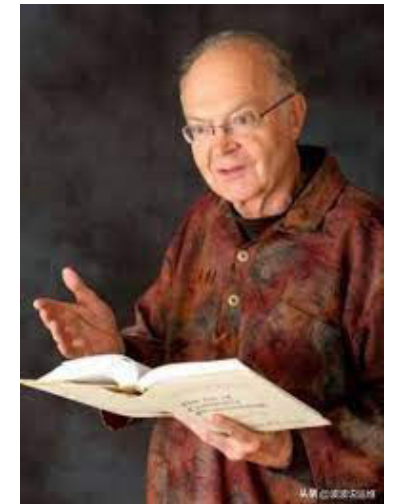
Programmieren == irgendwelchen Code eintippen?

Programmieren heisst Probleme lösen. **Richtig** lösen.

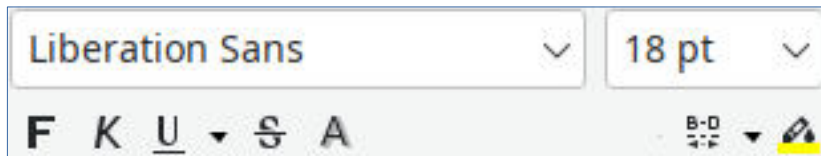
Programmieren == Probleme lösen

«Es wird oft gesagt, dass eine Person etwas erst richtig verstanden hat, nachdem sie es jemand anderem beigebracht hat. In Wirklichkeit hat eine Person etwas erst richtig verstanden, nachdem sie es einem Computer beigebracht hat, das heisst, es als Algorithmus ausgedrückt hat.»

— Donald Knuth



```
1 import threading
2 import time
3 from pivy import coin
4 from random import randrange
5
6 # show info about Python version
7 import sys
8 print('Python version: ', sys.version)
```



Programmierschnittstelle:

- unbegrenzter Gestaltungsspielraum
- kreativ und spannend
- ermöglicht Automatisierung
- wichtig für Forschung und Entwicklung
- unser Ziel!

Erweiterungen von Programmen:

- müssen nachinstalliert werden
- neue Funktionen
- Programmierschnittstelle (siehe oben) muss vorhanden sein

Basisfunktionen von Programmen

- einfach
- manuell und repetitiv
- eingeschränkter Funktionsumfang

Python-Entwicklungsumgebung

Zur Erstellung von Python-Programmen genügen eigentlich einfache Texteditoren, wie wir sie schon kennengelernt haben, z.B. `vi`, `nano` oder `gedit`. Ein deutlich höheres Komfort-Level und erweiterte Möglichkeiten bieten jedoch so genannte «Integrierte Entwicklungsumgebungen», die dann schon bei der Eingabe des Python-Quelltextes Fehler erkennen oder automatische Vervollständigungen anbieten.

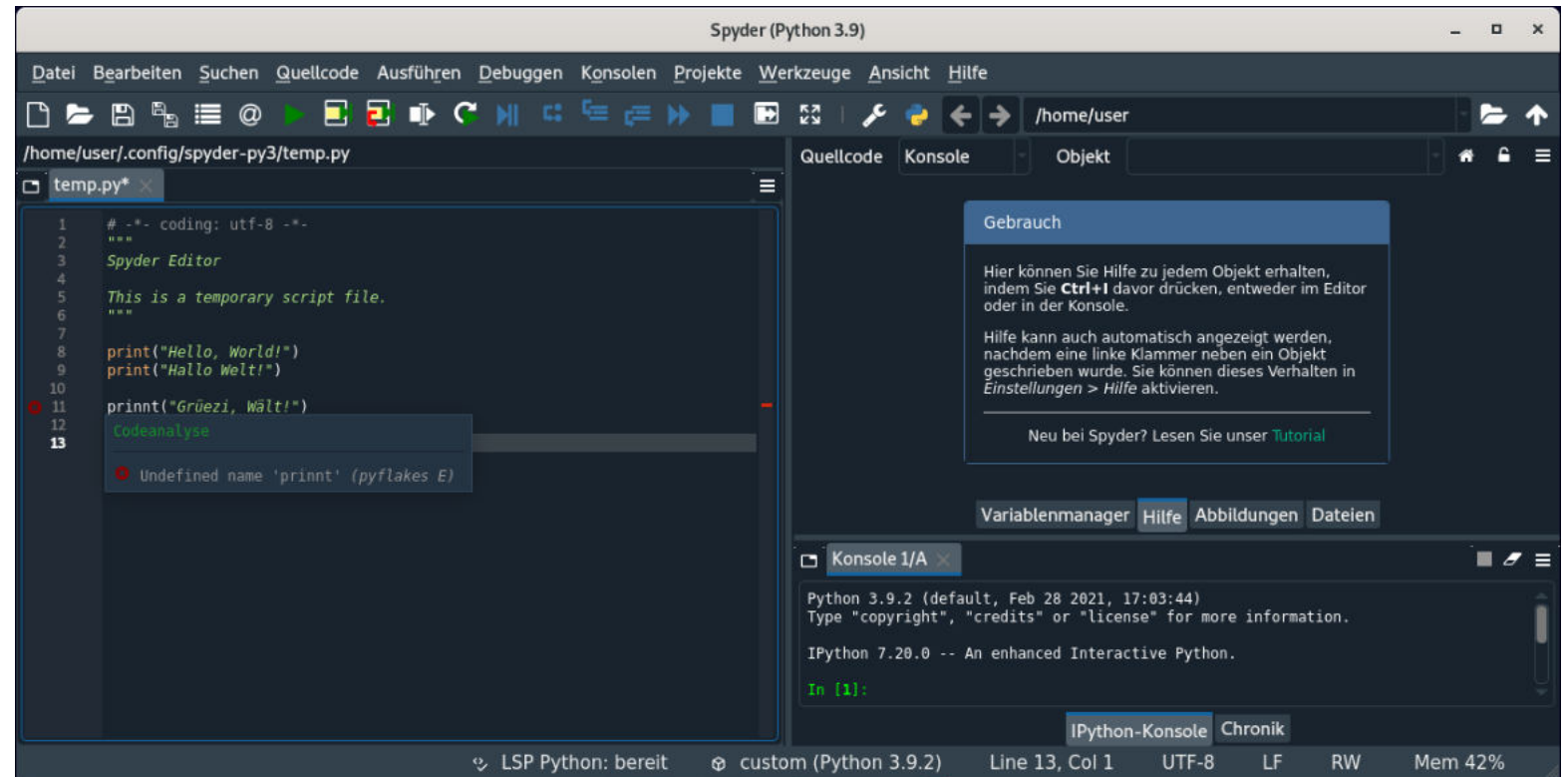
Eine relativ einfache Python-Entwicklungsumgebung ist die «Spyder IDE», siehe <https://www.spyder-ide.org>.

In der Lernumgebung können Sie einfach das Softwarepaket `spyder3` nach-installieren.

Starten Sie dann das Programm via Desktopumgebung (Dock → Anwendungen anzeigen), ignorieren eventuelle Hinweise zu Updates und stellen die Sprache via «Tools → Preferences → General → Advanced Settings» auf Deutsch um.

Unter Windows und macOS laden Sie das Installationsprogramm von folgender Seite herunter:

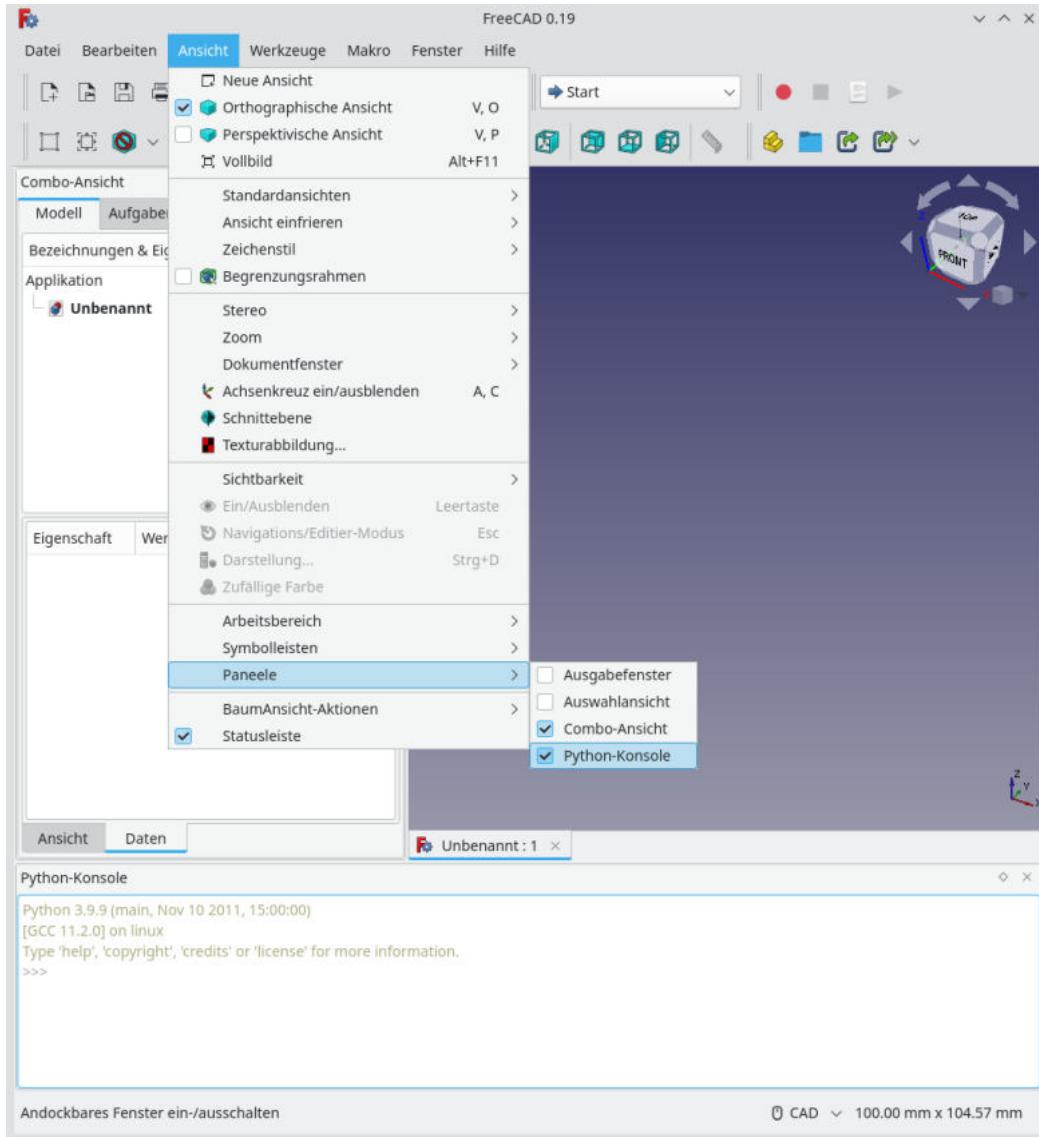
<https://github.com/spyder-ide/spyder/releases>



FreeCAD: Einführung Python-Konsole

In FreeCAD kann via «Ansicht ► Paneele ► Python-Konsole» eine integrierte Python-Konsole aktiviert werden.

Nicht erschrecken, Sie sehen dort «live», welche Python-Befehle von FreeCAD intern abgearbeitet werden, sobald Sie irgendeine Aktion ausführen (wie bei einem Raddampfer für Touristen).



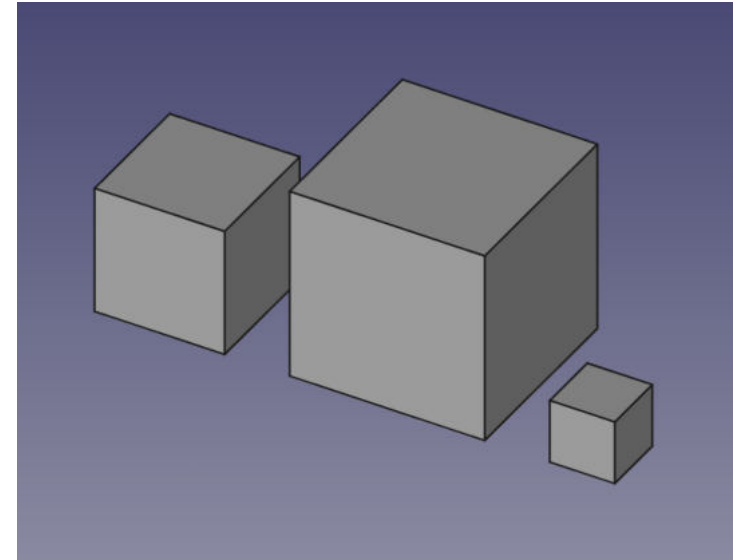
Aufgabe:

Probieren Sie, in der FreeCAD-Python-Konsole mit dem print-Befehl «Hello, World!» auszugeben.

Funktionen (Python)

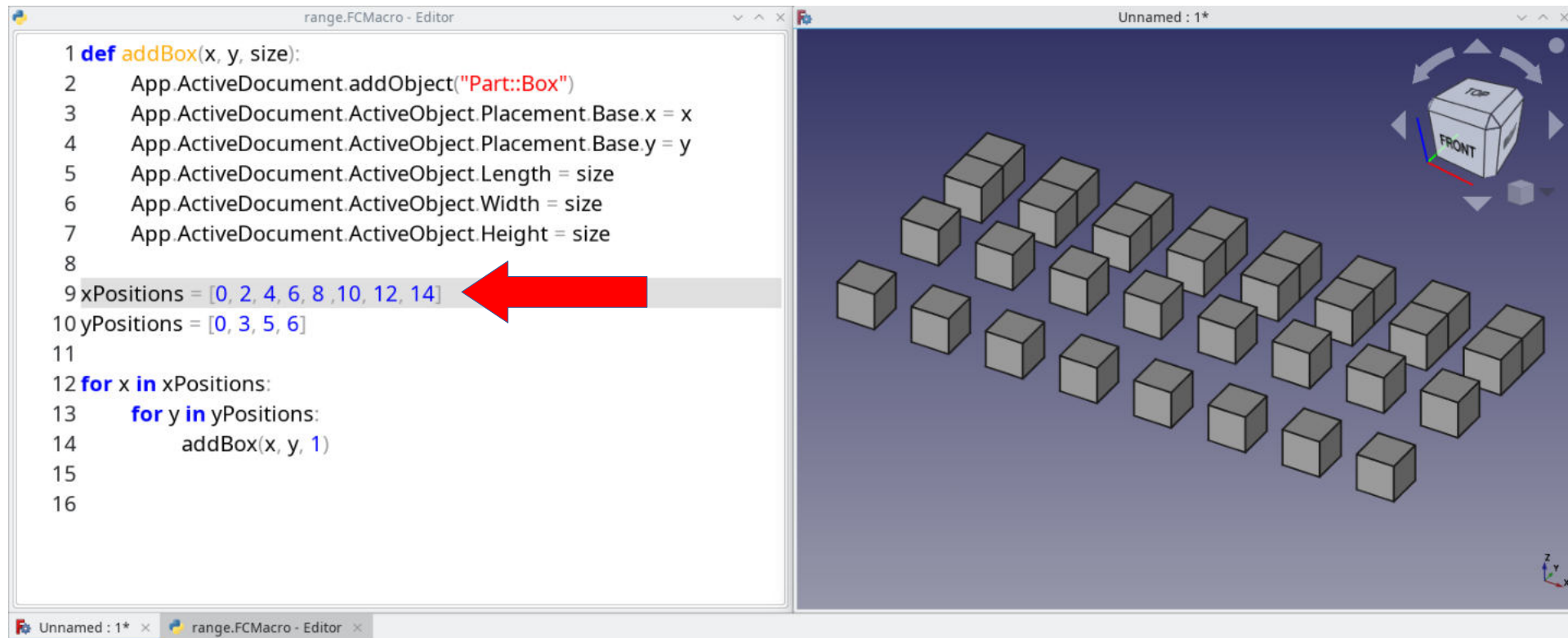
In CAD-Programmen können wir mit Funktionen viele Dinge automatisieren. Hier ein Beispiel für eine Funktion, mit der verschieden grosse Würfel an verschiedenen Positionen hinzugefügt werden können:

```
def addBox(position, size):  
    App.ActiveDocument.addObject("Part::Box")  
    App.ActiveDocument.ActiveObject.Placement.Base.x = position  
    App.ActiveDocument.ActiveObject.Length = size  
    App.ActiveDocument.ActiveObject.Width = size  
    App.ActiveDocument.ActiveObject.Height = size  
addBox(0, 2)  
addBox(3, 3)  
addBox(7, 1)
```



Aufgabe

Nehmen Sie an, Sie müssten nun viel mehr Würfel an regelmässigen x-Positionen erzeugen. Testen Sie den Ansatz, die Liste manuell zu erweitern:



The screenshot displays two windows from a CAD application. The left window, titled 'range.FCMacro - Editor', contains the following macro code:

```
1 def addBox(x, y, size):
2   App.ActiveDocument.addObject("Part::Box")
3   App.ActiveDocument.ActiveObject.Placement.Base.x = x
4   App.ActiveDocument.ActiveObject.Placement.Base.y = y
5   App.ActiveDocument.ActiveObject.Length = size
6   App.ActiveDocument.ActiveObject.Width = size
7   App.ActiveDocument.ActiveObject.Height = size
8
9 xPositions = [0, 2, 4, 6, 8, 10, 12, 14]
10 yPositions = [0, 3, 5, 6]
11
12 for x in xPositions:
13   for y in yPositions:
14     addBox(x, y, 1)
15
16
```

A red arrow points to the `xPositions` list on line 9. The right window, titled 'Unnamed : 1*', shows a 3D perspective view of a grid of 48 grey rectangular boxes arranged in a 6x4x2 pattern. A 3D coordinate system is visible in the top right corner of the 3D view, with axes labeled 'x', 'y', and 'z'. A small white box with 'TOP' and 'FRONT' labels is shown in the top right corner of the 3D view, with arrows indicating its orientation.

Welches Problem könnte bei diesem Ansatz in Zukunft entstehen?

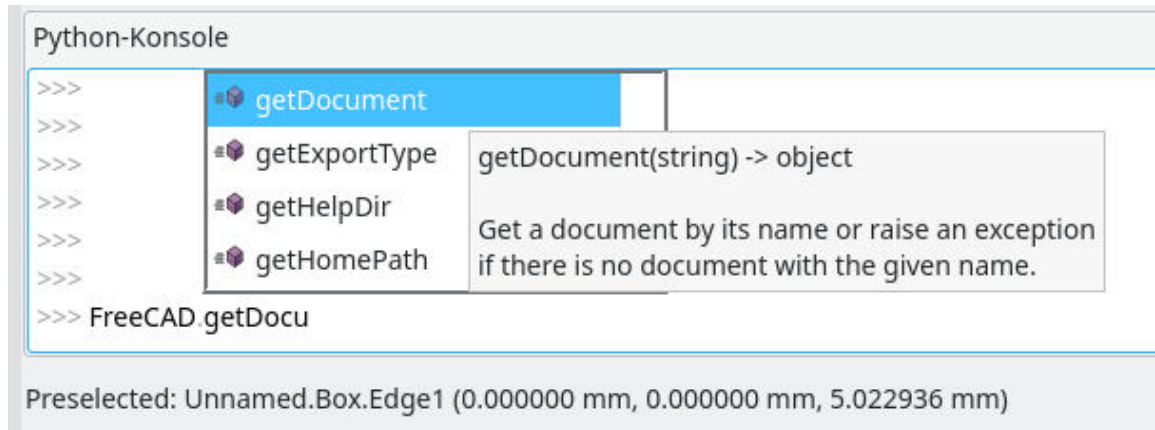
Python-Konsole: Code-Vervollständigung und Dokumentation

So ähnlich wie in der Entwicklungsumgebung Spyder haben Sie auch in der Python-Konsole von FreeCAD (noch sehr einfach gehaltene) Funktionen zur Code-Vervollständigung und Dokumentation.

Beim Programmieren haben die FreeCAD-Entwickler darauf Wert gelegt, dass das Programm selbst sauber in Module und Klassen strukturiert wurde.

Sobald Sie nach dem Namen einer Klasse einen Punkt schreiben (zum Beispiel ist «FreeCAD» selbst eine der Hauptklassen im Programm), öffnet sich eine Liste mit allen bekannten Eigenschaften und Methoden dieser Klasse. Wenn Sie anfangen, den Namen einer Eigenschaft oder Methode zu schreiben (z.B. «getDocument»), wird in der Liste zu dem passenden Eintrag gesprungen.

Wenn Sie den Mauszeiger über einem Eintrag in der Liste stehen lassen, bekommen Sie noch dessen Kurzdokumentation angezeigt:



The screenshot shows a Python console window titled "Python-Konsole". The input field contains the text ">>> FreeCAD.getDocu". A dropdown menu is open, listing several methods: "getDocument", "getExportType", "getHelpDir", and "getHomePath". The "getDocument" method is highlighted in blue. A tooltip is visible next to the "getDocument" entry, displaying the signature "getDocument(string) -> object" and the description "Get a document by its name or raise an exception if there is no document with the given name." At the bottom of the console, the text "Preselected: Unnamed.Box.Edge1 (0.000000 mm, 0.000000 mm, 5.022936 mm)" is visible.

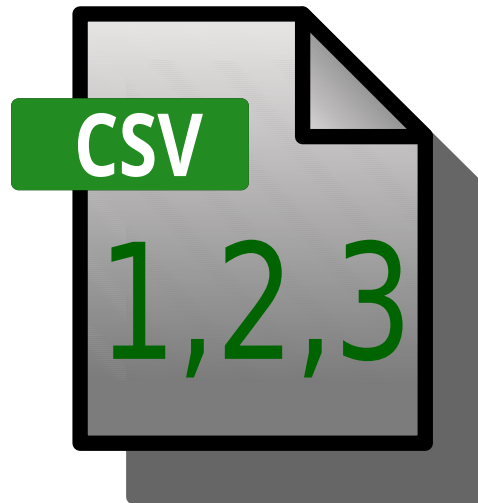
Praxisbeispiel: CAD mit CSV

Stellen Sie sich vor, eine Künstlerin oder ein Künstler erstellt den Prototypen einer «Cheese Lamp» und möchte, dass Sie sich beim Erstellen des CAD-Modells **absolut exakt** an die Vorgaben halten. Daher erhalten Sie die Koordinaten und Durchmesser der Löcher als Datensätze in einer CSV-Datei.

Bereits mit grundlegenden Programmierkenntnissen ist es nun möglich, daraus automatisch das entsprechende CAD-Modell zu erzeugen.



Praxisbeispiel: Programmierung Datenformate



JSON



nicht mit Jason
verwechseln...

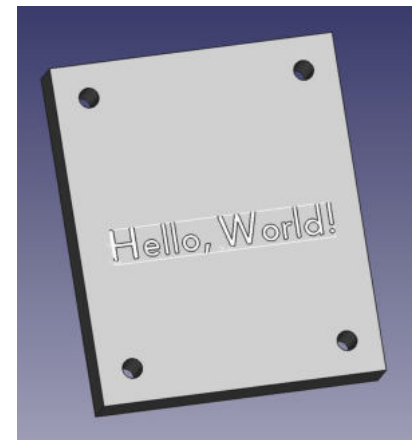
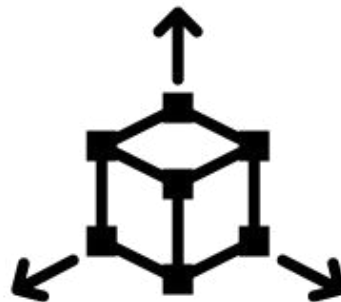


XML

Miniprojekt: Vom ERP zum fertigen Produkt

Mit den bisher im Informatikunterricht erworbenen Kompetenzen sind Sie nun in der Lage, die gesamte Kette von einem ERP-System zum fertigen Produkt an einem kleinen Projekt nachzuvollziehen.

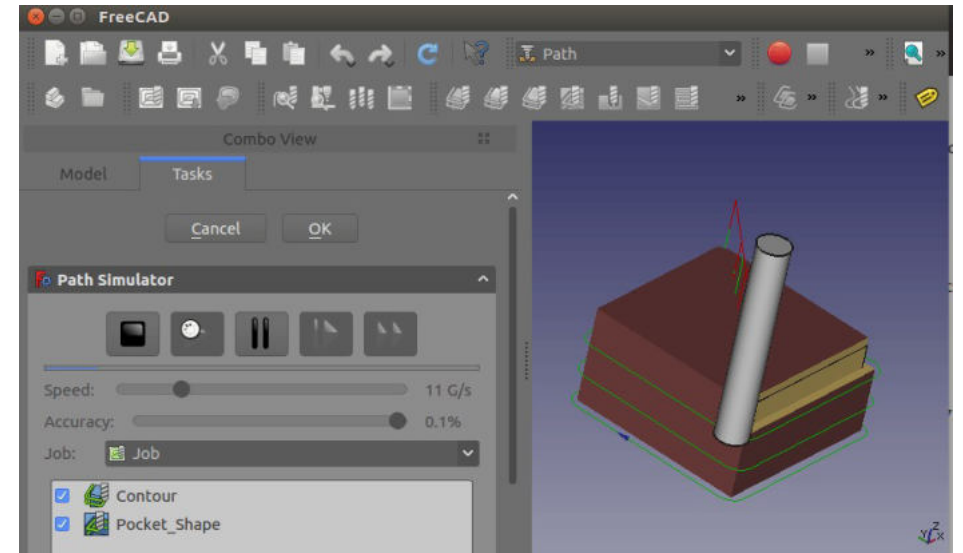
Die Idee ist, dass wir mit SQLite eine einfache Datenbank aufsetzen, die ein ERP-System repräsentieren soll, diese an ein parametrisches CAD-Projekt anbinden, mit dem wir wiederum ein physisches Produkt, in unserem Beispiel ein einfaches Türschild, herstellen (Anfang nächste Woche).



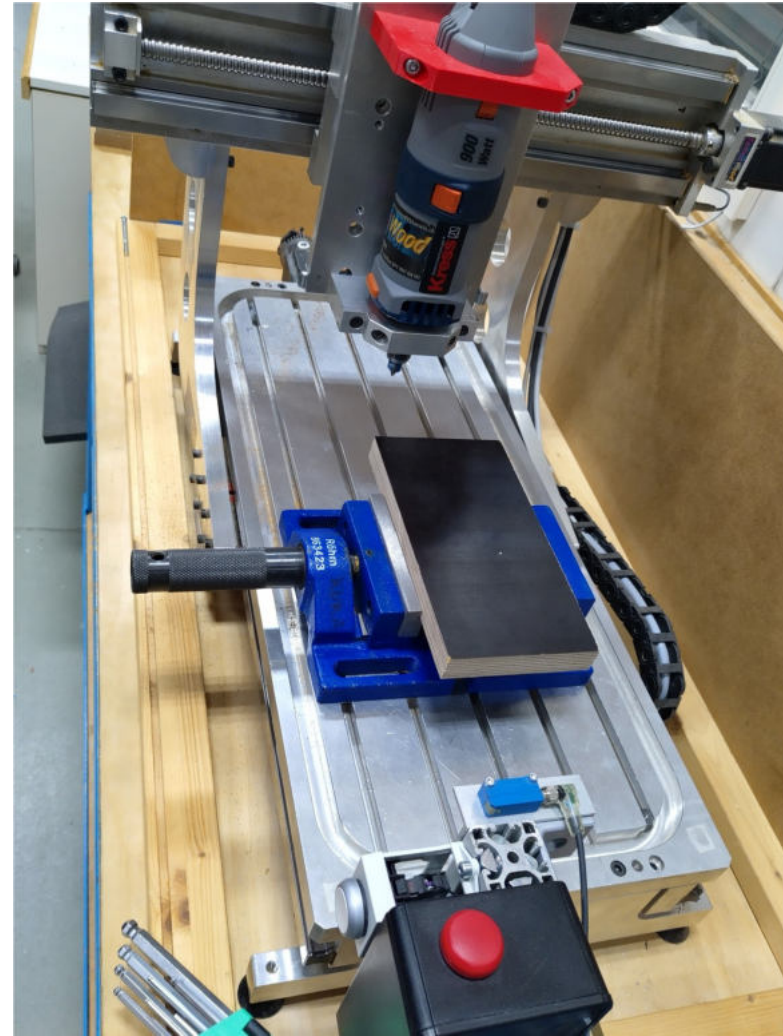
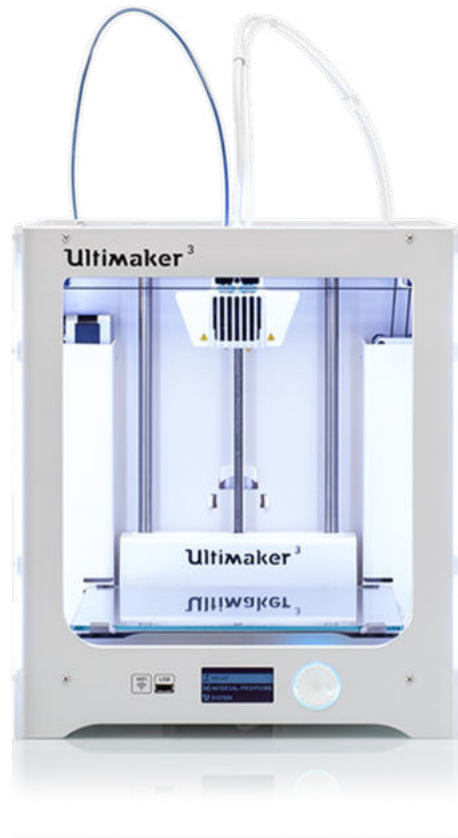
Pfaderstellung in FreeCAD

Mit dem Arbeitsbereich *Path* lassen sich in FreeCAD anhand der Dimensionen der Rohlinge, der Werkzeuge und der Werkstücke entsprechende Werkzeugpfade zur Herstellung der Werkstücke erzeugen.

Der Arbeitsbereich ist hier im Wiki beschrieben:
https://wiki.freecad.org/Path_Workbench/de



Und jetzt? Ab ins Labor!



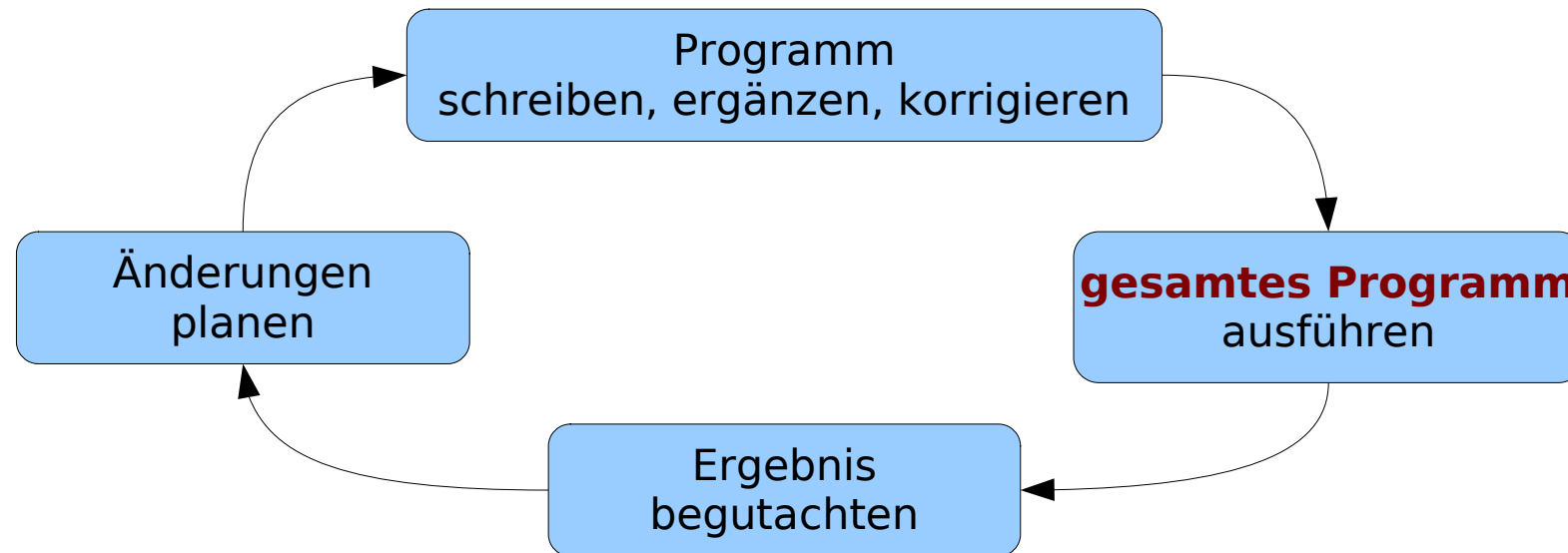
Warum Python?

- generische Programmiersprache
 - in allen aktuellen Sprachindizes unter den beliebtesten und verbreitetsten Sprachen
 - Erlerntes kann auch an anderen Stellen, z.B. in CAD- oder EPR-Systemen angewendet werden
- dennoch hervorragende numerische Bibliotheken für Data Science
- mit allen gängigen Plattformen kompatibel
- unter einer freien Lizenz veröffentlicht



Nur Python?

Grundsätzlich würde es schon reichen, mit den zur Verfügung gestellten Softwarebibliotheken, Python-Programme zur Lösung der jeweils gestellten Aufgaben zu schreiben. Allerdings wäre der Workflow an dieser Stelle ziemlich «grobgranular»:

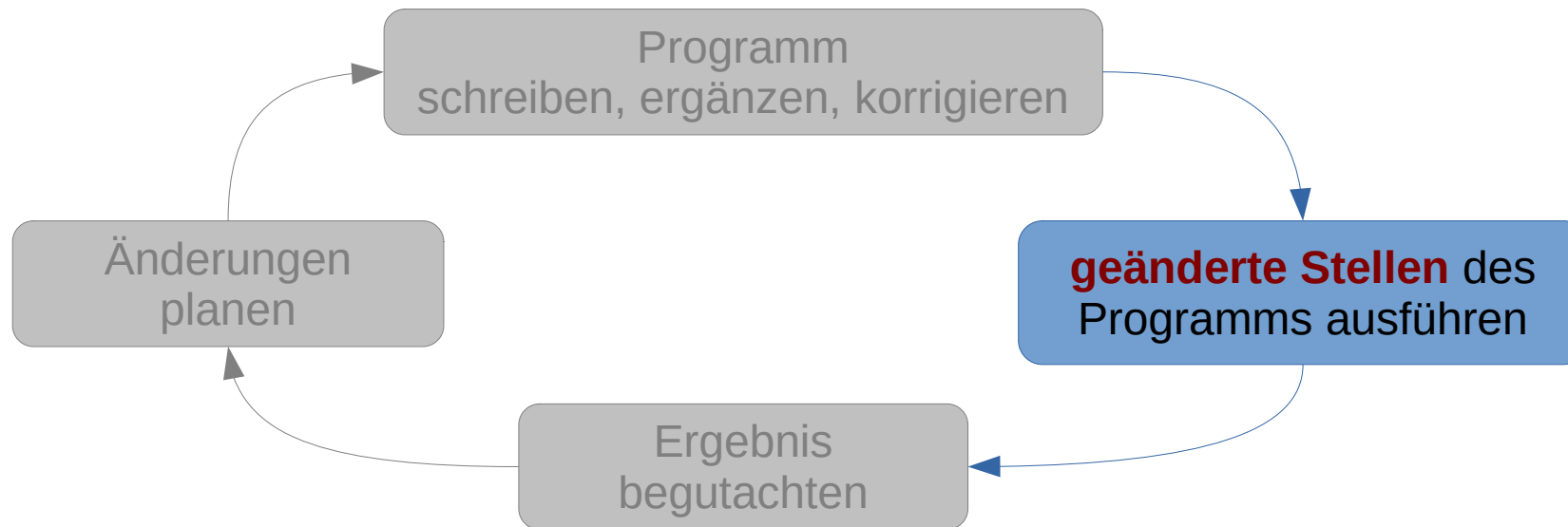


► Zur Verdeutlichung: Eine kleine Demo...

Exploratives Arbeiten

Eine für viele Menschen angenehmere Herangehensweise ist das **explorative** Arbeiten: in kleinen Schritten einfach und schnell Änderungen vornehmen können, ausprobieren, entdecken, untersuchen, experimentieren, ...

Ein erfreulicher Nebeneffekt dieser Arbeitsweise ist, dass dabei auch die Effizienz zunimmt.



Genau dieser explorative Arbeitsstil wird in Jupyter-Notebooks unterstützt.

Jupyter-Notebooks



Notebook



Python 3



Ansible



Bash



Julia 1.2.0



Octave



R



SciJava

(Jupyter = **J**ulia, **P**ython und **R**)

Jupyter-Notebooks

```
[1]: 1 import pandas as pd
2
3 fileNames = ["Com_00_3.1_m", "Com_00_34_m1", "Com_40_3.1_m", "Com_80_3.1_m", "V2A_00_31_m1", "V2A_40_31_m1", "V2A_80_31_m1"]
4 sheetNumbers = [3, 4, 5]
5
6 files=[]
7 for fileName in fileNames:
8     files.append(pd.read_excel(fileName + ".xlsx", sheet_name=sheetNumbers, skiprows=3, header=None))
```

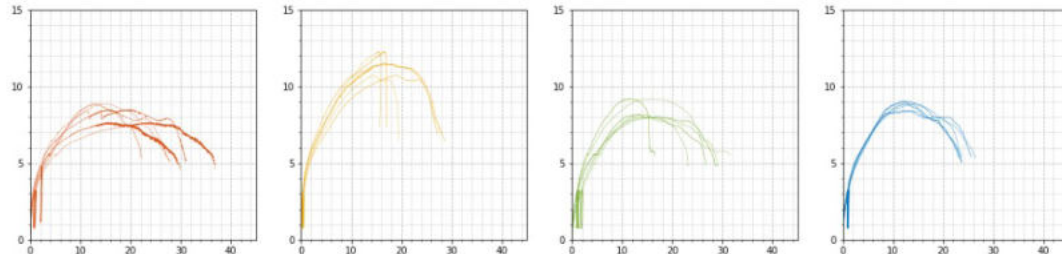
Daten bearbeiten

Wir nehmen folgende Überarbeitung der eingelesenen Daten vor:

- alle Werte in der Spalte "Standardkraft" werden durch 1000 dividiert
- alle Werten in den letzten vier Spalten werden negiert
- zwei neue Spalten mit den mittleren Abweichungen für die "top"- und "base"-Werte werden hinzugefügt

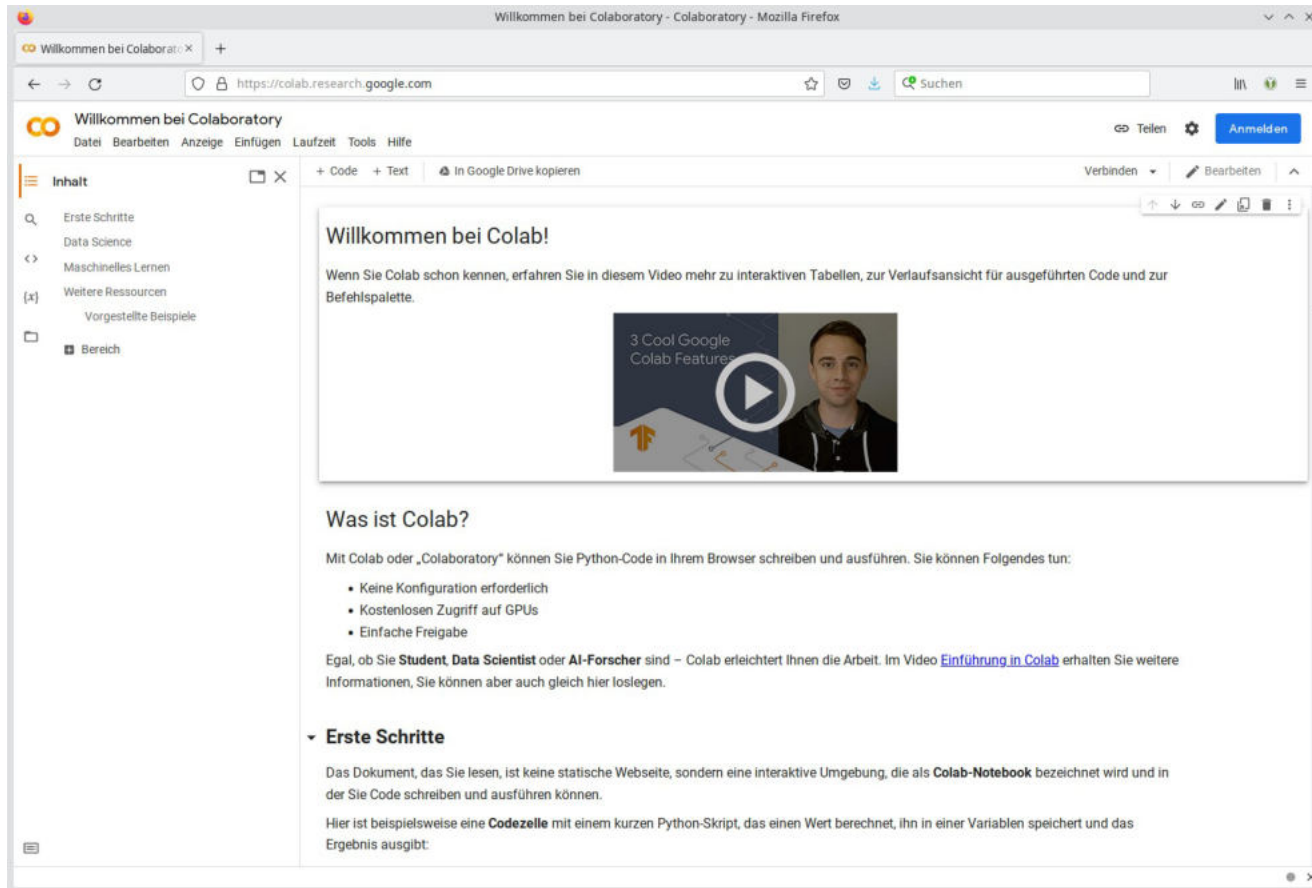
Daten plotten

Hier stellen wir die Daten mit `matplotlib` dar:



- Umgebung zur Erstellung von interaktiven Dokumenten
- leicht zugänglich: funktioniert direkt in allen gängigen Webbrowsern
- unterschiedliche «Zellen»:
 - Code
 - Beschreibungen
 - Ausgaben
 - Visualisierungen (Tabellen, Diagramme, ...)
 - interaktive Elemente
- standardisiert!

Standardisierung = Marktöffnung



Beispiel: **Google Colab**

Intro-Video:

<https://www.youtube.com/watch?v=inN8seMm7UI>

Aufgabe:

Öffnen Sie folgenden Link:

<https://colab.research.google.com>

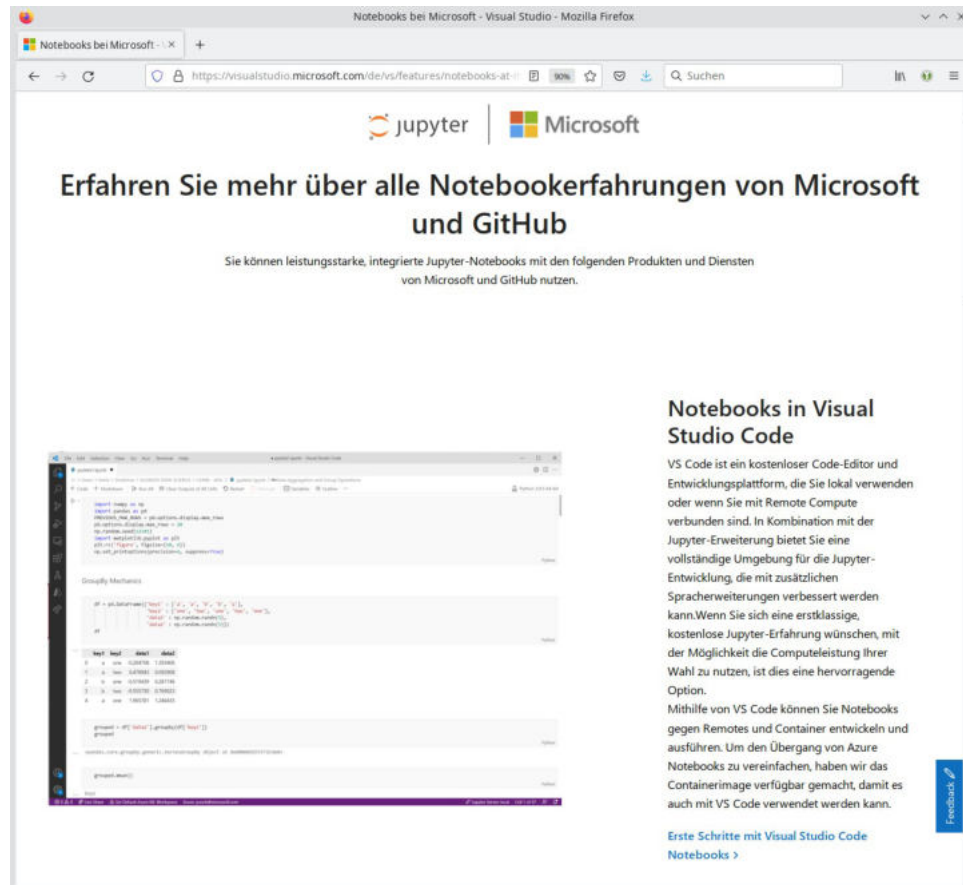
... und lesen Sie die ersten drei Abschnitte, einschliesslich «Maschinelles Lernen».

(Ausprobieren geht leider nur mit Google-Konto...)

Standardisierung = Marktöffnung

Beispiel: **Microsoft**

<https://visualstudio.microsoft.com/de/vs/features/notebooks-at-microsoft/>



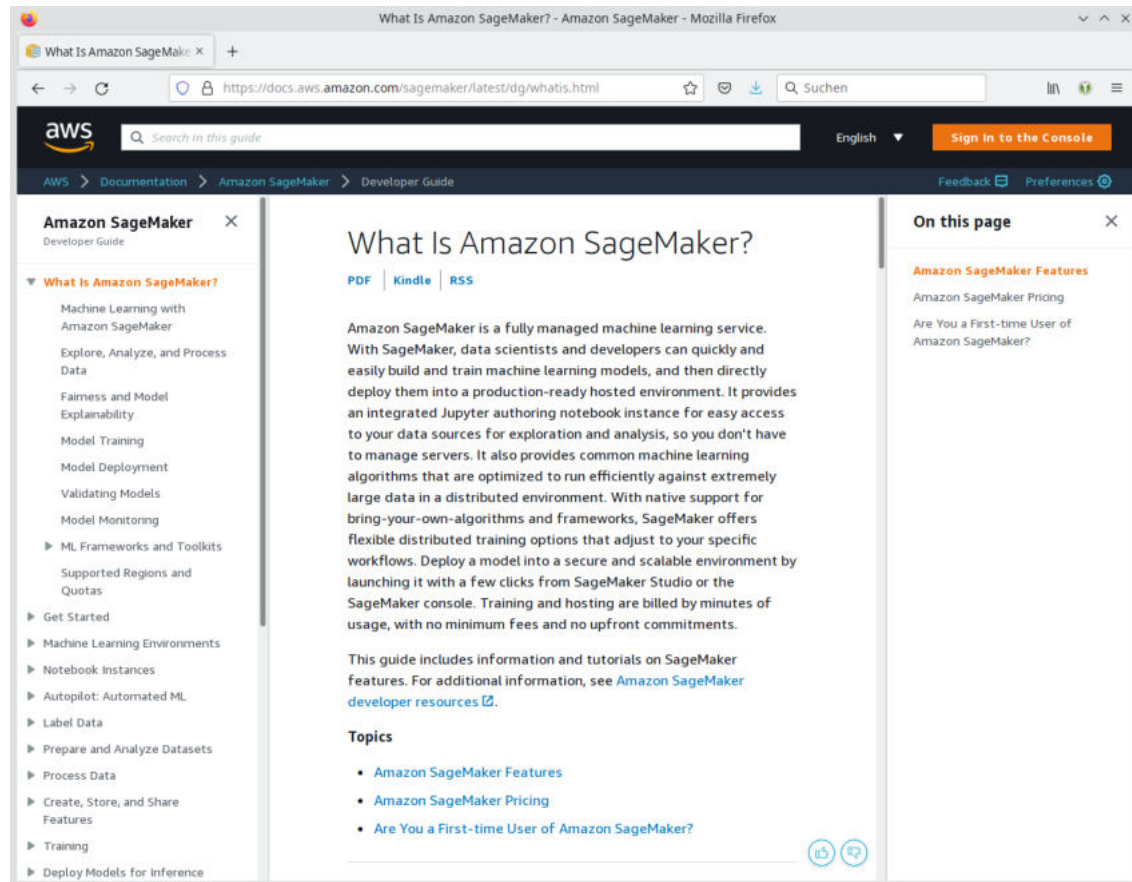
Aufgabe:

Bitte Link oben öffnen und kurz über die verschiedenen Integrationsmöglichkeiten von Jupyter-Notebooks informieren.

Standardisierung = Marktöffnung

Beispiel: **Amazon SageMaker**

<https://docs.aws.amazon.com/sagemaker/latest/dg/whatis.html>



Aufgabe:

Bitte Link oben öffnen und kurz über die Integration von Jupyter-Notebooks in Amazon SageMaker informieren.

Standardisierung = Marktöffnung

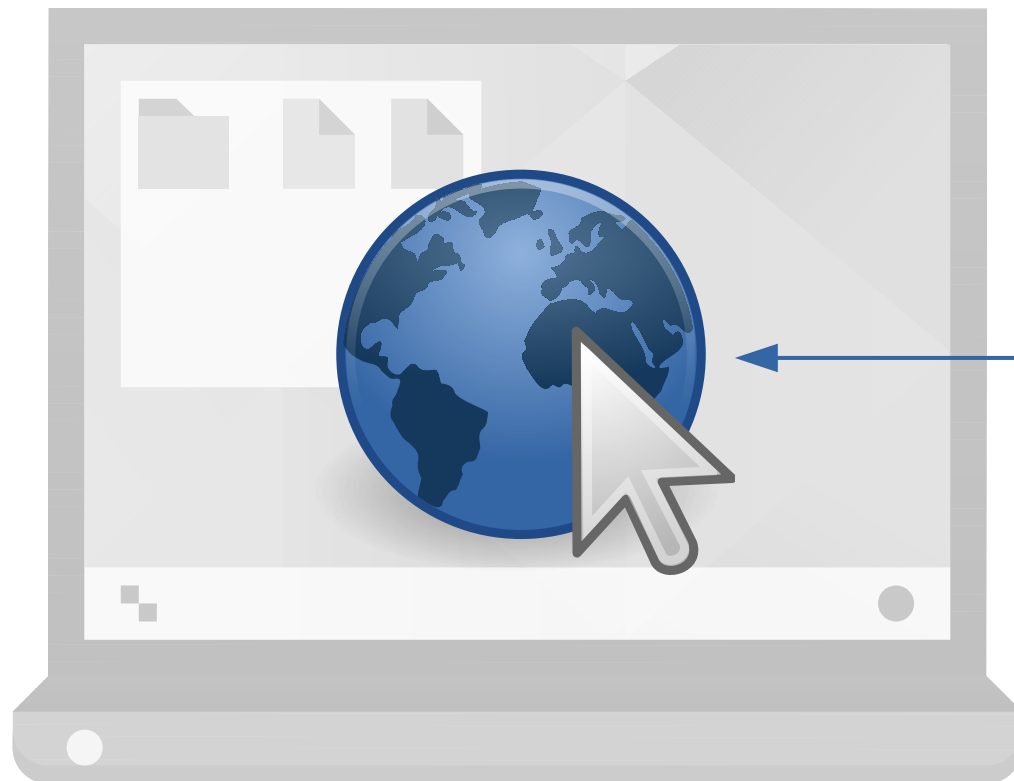
Einfach mal auf folgenden Seiten nach Jupyter suchen:

- <https://www.ibm.com/cloud/watson-studio>
- <https://www.kaggle.com>
- <https://datalore.jetbrains.com>
- <https://saturncloud.io>
- <https://cocalc.com>
- <https://mybinder.org> ← probieren wir in wenigen Minuten aus!
- ...

Klassische Online-Variante

Bearbeitung des Notebooks und
Anzeige der Resultate im Webbrowser

Ausführung des Python-Codes auf
Servern der Online-Anbieter



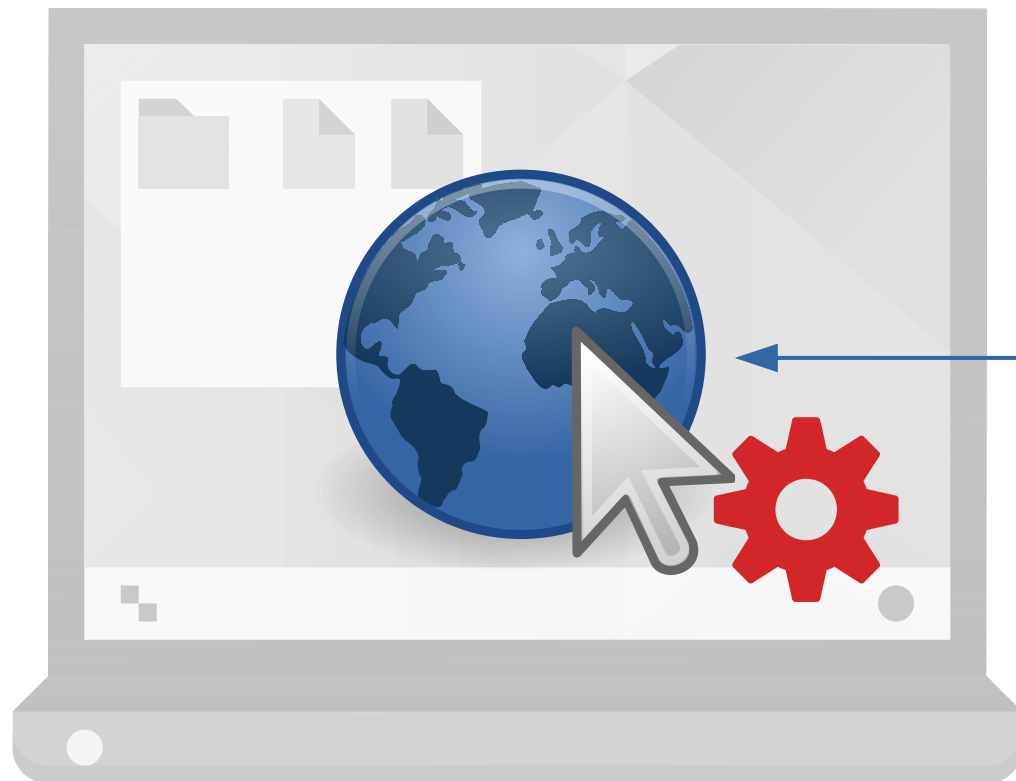
<https://colab.research.google.com>
<https://www.ibm.com/cloud/watson-studio>
<https://www.kaggle.com>
<https://datalore.jetbrains.com>
<https://saturncloud.io>
<https://cocalc.com>
<https://mybinder.org>

...

Neue Online-Variante: JupyterLite

Bearbeitung **und Berechnung** im Webbrowser

Bereitstellung einer einfachen statischen
Webseite (nur Download)



<https://jupyter.org/try-jupyter/lab/>
<https://jupyterlite.rtd.io/en/latest/try/lab>
<https://ipywidgets.readthedocs.io/en/stable/try/lab/>

...

Offene Standards in der Wissenschaft



The Nobel Prize in Physics 2017

The Royal Swedish Academy of Sciences has decided to award the Nobel Prize in Physics 2017

with one half to

and the other half jointly to

Rainer Weiss

LIGO/VIRGO Collaboration

Barry C. Barish

LIGO/VIRGO Collaboration

and

Kip S. Thorne

LIGO/VIRGO Collaboration

“for decisive contributions to the LIGO detector and the observation of gravitational waves”

Gravitational waves finally captured

On 14 September 2015, the universe’s gravitational waves were observed for the very first time. The waves, which were predicted by Albert Einstein a hundred years ago, came from a collision between two black holes. It took 1.3 billion years for the

theory of relativity. They are always created when a mass accelerates, like when an ice-skater pirouettes or a pair of black holes rotate around each other. Einstein was convinced it would never be possible to measure them. The LIGO project’s achievement was using a pair of gigantic

<https://www.gw-openscience.org>

Event Catalog

The Gravitational-wave Transient Catalog (GWTC) is a cumulative set of events detected by LIGO, Virgo, and KAGRA.

Open Data Workshop

Participants will receive a crash-course in gravitational-wave data analysis that includes lectures, software tutorials, and a data challenge.

Tutorials

Learn with tutorials that will lead you step-by-step through some common data analysis tasks.

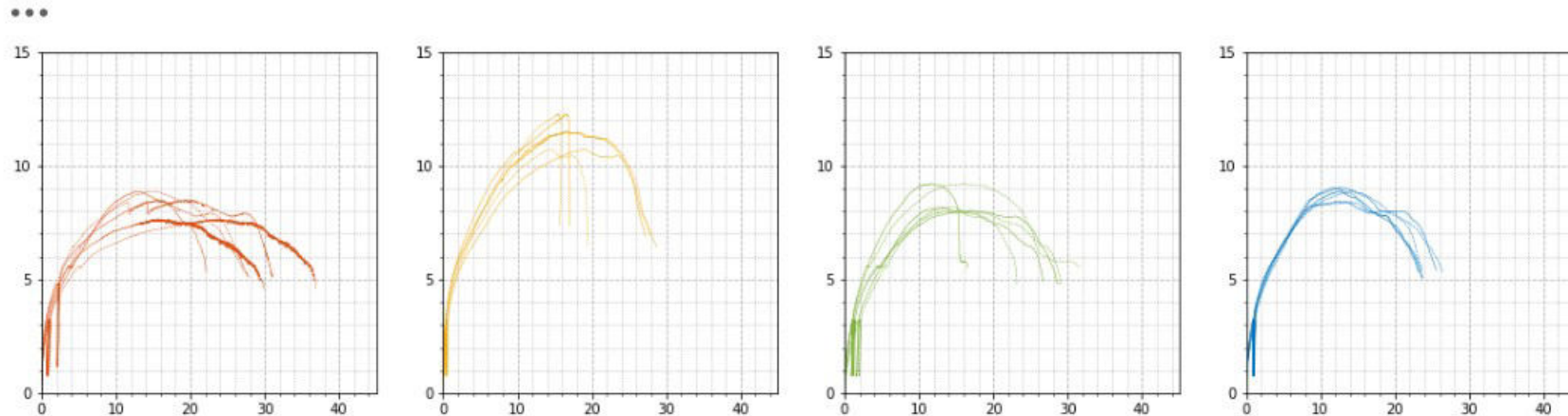
Beispiele

Das können wir genauso gut wie die Nobelpreisträger!

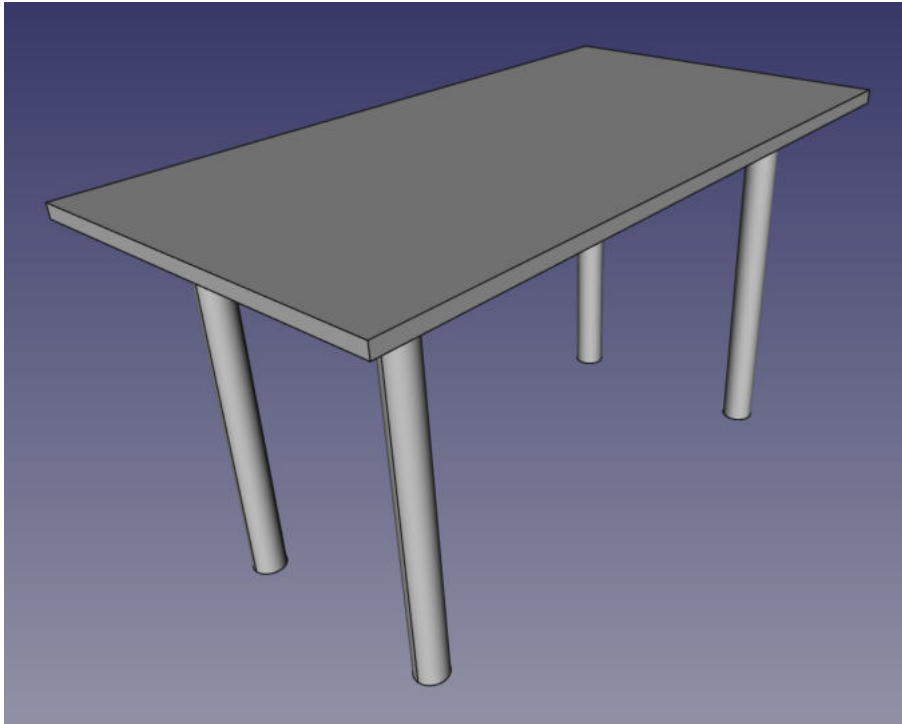
(Die Anzahl der freien Zugänge zu Mybinder ist beschränkt, der Schnellere ist der Geschwindere!)

Beispiel von Lukas Furrer (Assistent AHB) zur Fließverschiebung:

<https://mybinder.org/v2/gh/ronnystandtke/jupyter-example/main?labpath=Fließverschiebung.ipynb>



Beispiel aus der Lehre: Einführung in CAD

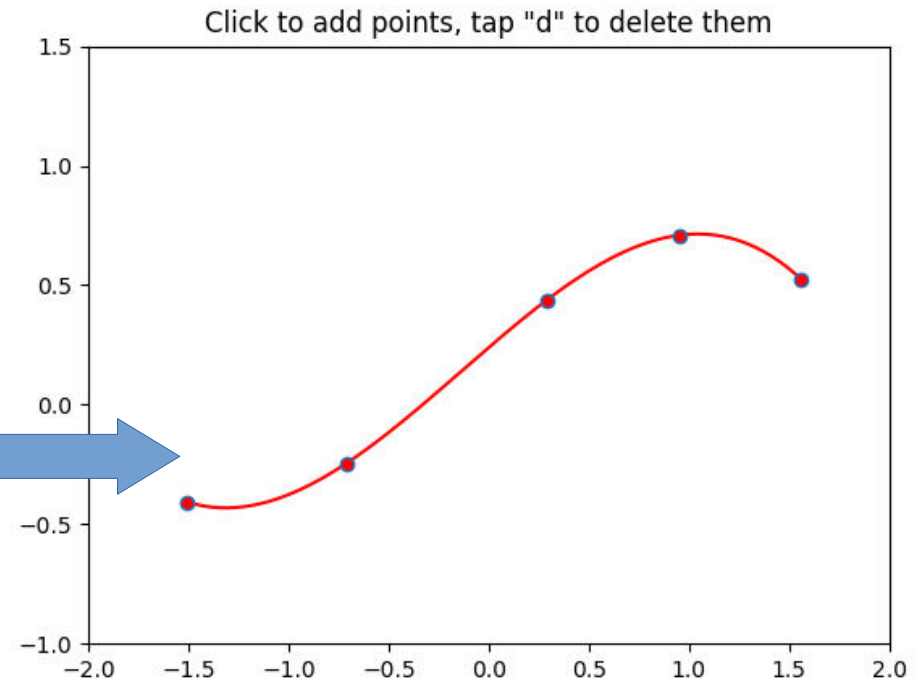


Grundlegende geometrische Formen (Quader, Zylinder, ...) sind in CAD-Programmen sehr einfach abbildbar.



In vielen Bereichen (z.B. Schiffs- und Automobilbau) werden jedoch aus aerodynamischen oder ästhetischen Gründen geschwungene Formen verwendet. Wie können die mathematisch abgebildet werden?

Beispiel aus der Lehre: Polynominterpolation?



Ist die Polynominterpolation eine geeignete mathematische Abbildung für geschwungene Formen?

Interaktives Übungsblatt:

https://ronnystandtke.github.io/runges_phenomenon/lab?path=Runges+phenomenon.ipynb

JupyterLab: Online-Angebote vs. lokale Installation

Python / Jupyter kann sowohl über die bereits erwähnten Online-Angebote verwendet werden als auch schnell und einfach lokal installiert werden. Hier ein Versuch, die Vor- und Nachteile der beiden Varianten gegenüberzustellen:

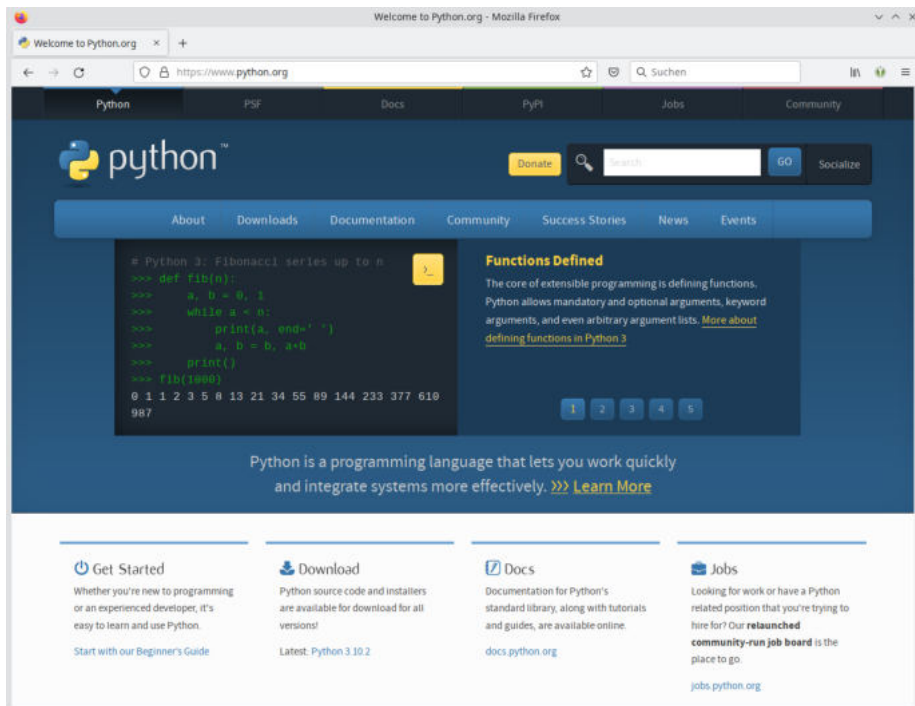
	Online-Angebote	lokale Installation
Aufwand	gering	höher
Kosten	potenziell höher	potenziell geringer
Abhängigkeitsverhältnis	höher	gering
Leistung	gratis: meist geringer kommerziell: potenziell höher	abhängig von lokaler Hardware
Verfügbarkeit	nicht immer gewährleistet	sehr hoch
Aktualität	häufig geringer	potenziell sehr hoch

JupyterLab (lokale Installation)

Um Python / Jupyter lokal zu installieren, gilt die Regel: «Viele Wege führen nach Rom.»
Wir werden jetzt zwei typischen Wege betrachten und gegenüberstellen:

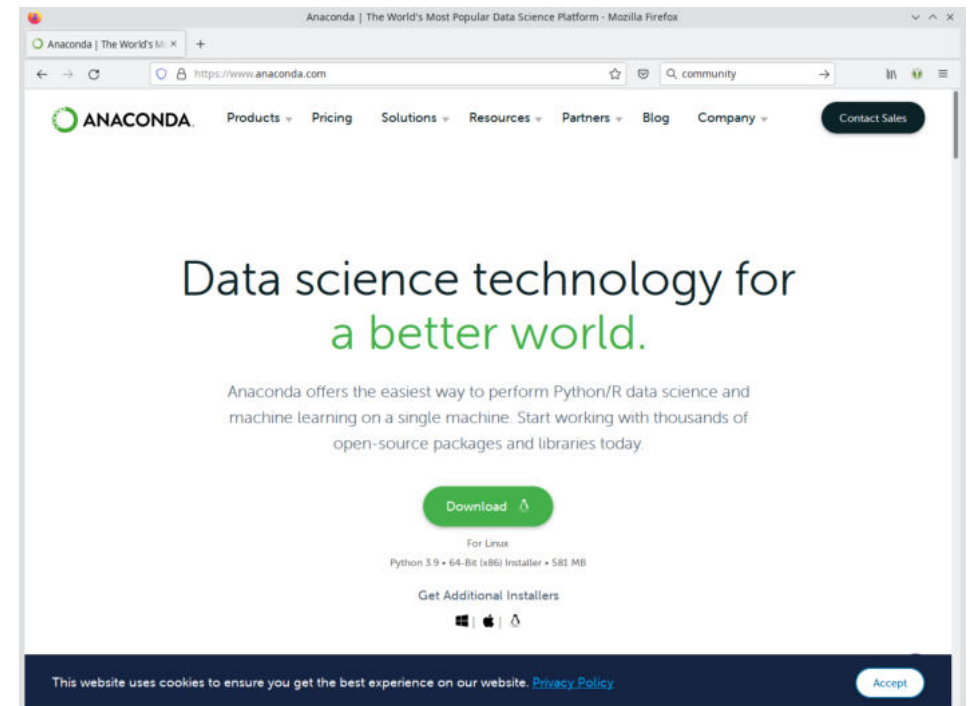
Python-Standardinstallation

<https://www.python.org>



Anaconda-Distribution

<https://www.anaconda.com>



JupyterLab (lokale Installation)

	python.org	anaconda.com
Organisation	Community	kommerzielles Angebot
Lizenzbedingungen	frei	kostenpflichtige Erweiterungen
Aktualität	immer auf dem neuesten Stand	neue Versionen werden erst mit etwas Verzögerung veröffentlicht
initialer Umfang	sehr schlank	250 Softwarepakete vorinstalliert
Paketverwaltung	via <i>pip</i>	via <i>conda</i>
Zugang	via Terminal / Shell-Befehle	via GUI («Anaconda Navigator»)

Beispiel aus der Forschung: Risikobewertung SBB-Bauten

Steckbrief



Kontakt

T +41 34 426 41 70

E E-Mail anzeigen

🌐 www.bfh.ch/de/dirk-proske

Prof. Dr. Dirk Proske
Studiengangsleiter Master Engineering

Adresse

Berner Fachhochschule
Architektur, Holz und Bau
Fachbereich Bauingenieurwesen
Pestalozzistrasse 20
3401 Burgdorf

Steckbrief



Kontakt

T +41 34 426 41 75

E E-Mail anzeigen

🌐 www.bfh.ch/de/jeanbaptiste-payeur

Dr. Jean-Baptiste Payeur
Dozent

in

Präsenzzeit

Montag
Dienstag
Mittwoch
Donnerstag
Freitag
Samstag

Adresse

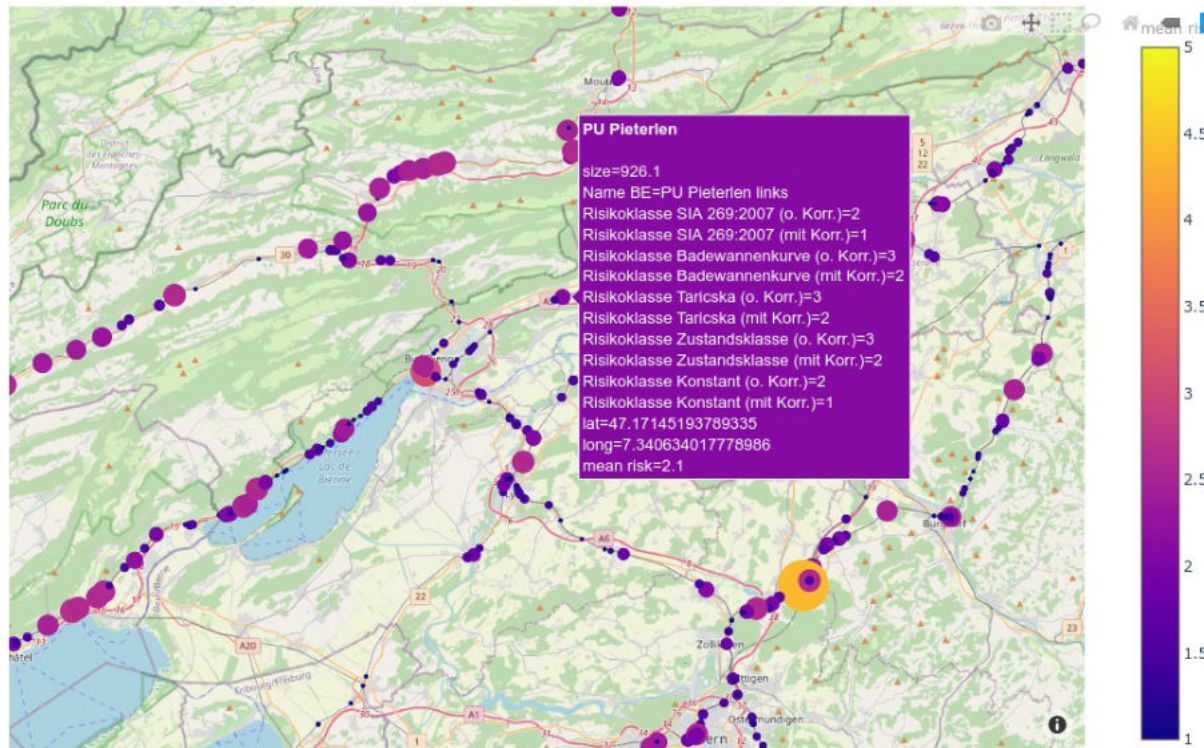
Berner Fachhochschule
Architektur, Holz und Bau
Bereich FDW
Pestalozzistrasse 20
3400 Burgdorf

Beispiel aus der Forschung: Risikobewertung SBB-Bauten

```
import plotly.express as px

fig = px.scatter_mapbox(df_plotly, lat="lat", lon="long",
                        hover_name='Name GO', hover_data = ['Name BE'] + names,
                        color='mean risk', size='size', size_max=30,
                        range_color=[1, 5], opacity=1.,
                        color_continuous_scale=px.colors.diverging.RdYlGn.reverse(),
                        zoom=7)

# The built-in plotly.js styles objects are: open-street-map, white-bg, carto-positron, carto-darkmatter, stamen-ter
# rain, stamen-toner, stamen-watercolor
fig.update_layout(mapbox_style="open-street-map")
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```

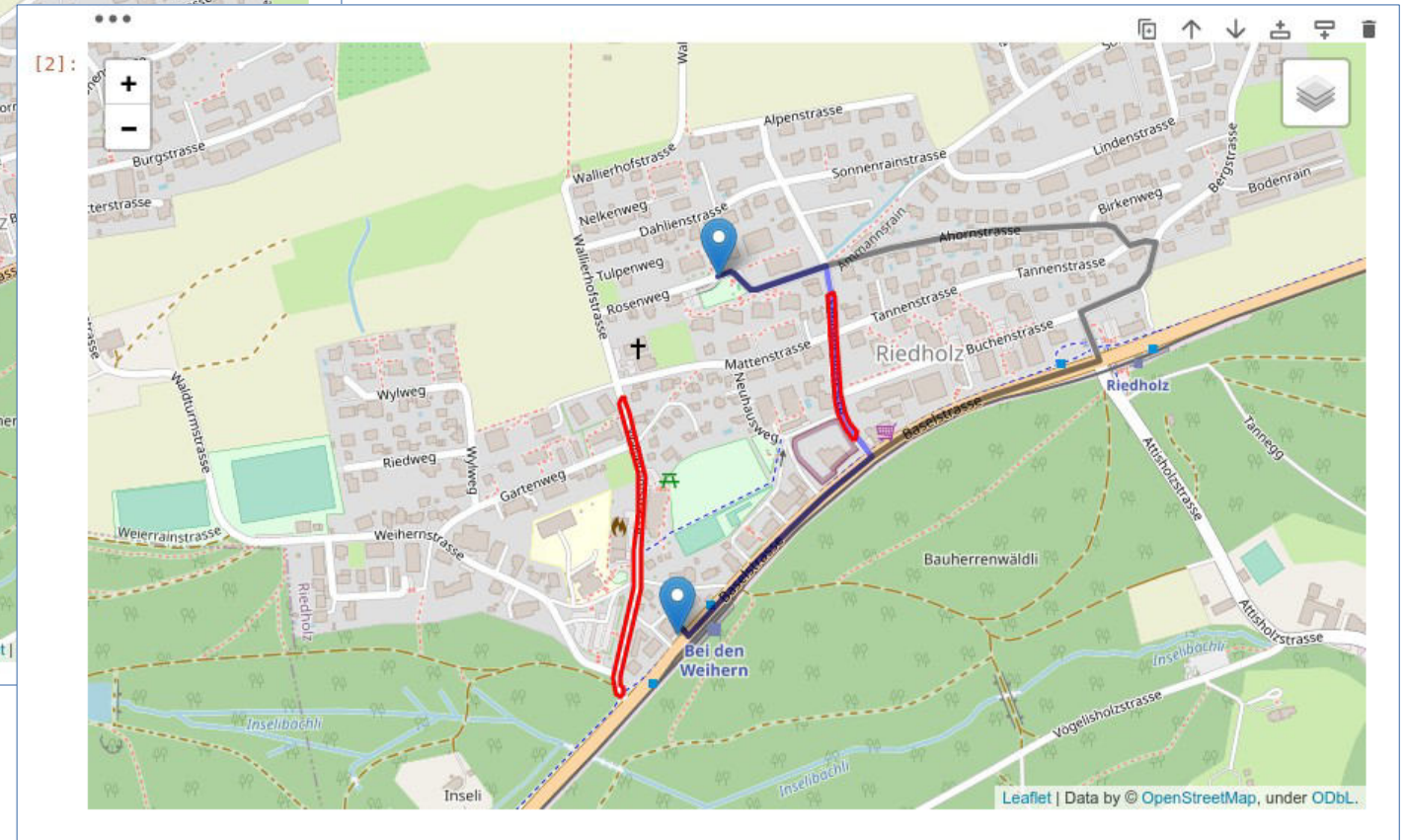
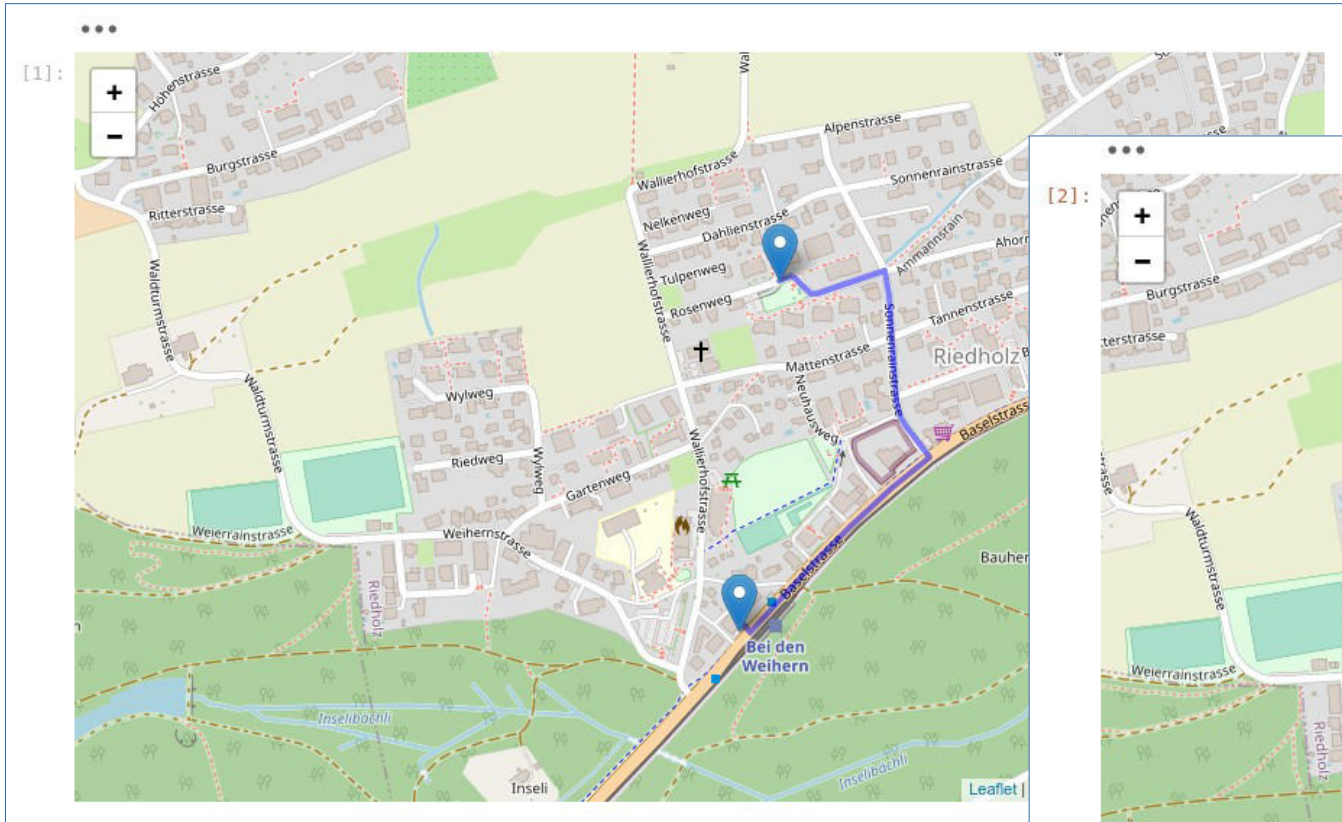


Zutaten:

- nicht-öffentliche Daten der SBB
- öffentliche Daten: Erdbebenzonen
- Python
- Jupyter
- GeoPandas:
<https://geopandas.org>



Beispiel aus der Forschung: Risikobewertung ASTRA



**openroute
service**

<https://openrouteservice.org>

Untersuchung eines bestimmten Punktes

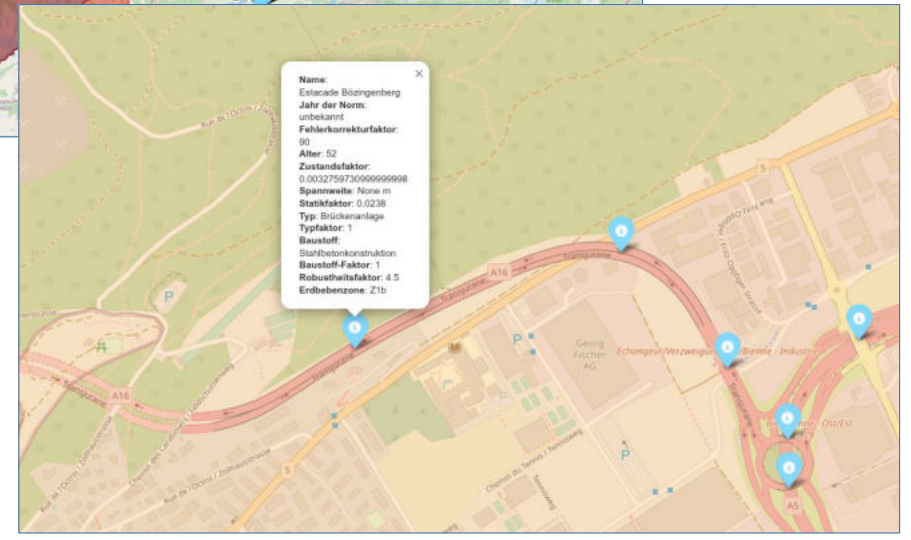
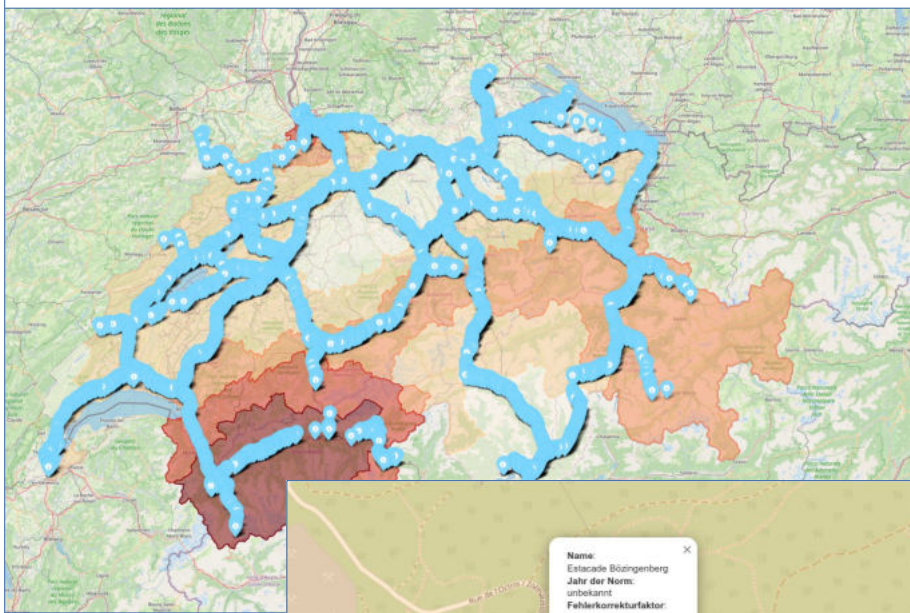
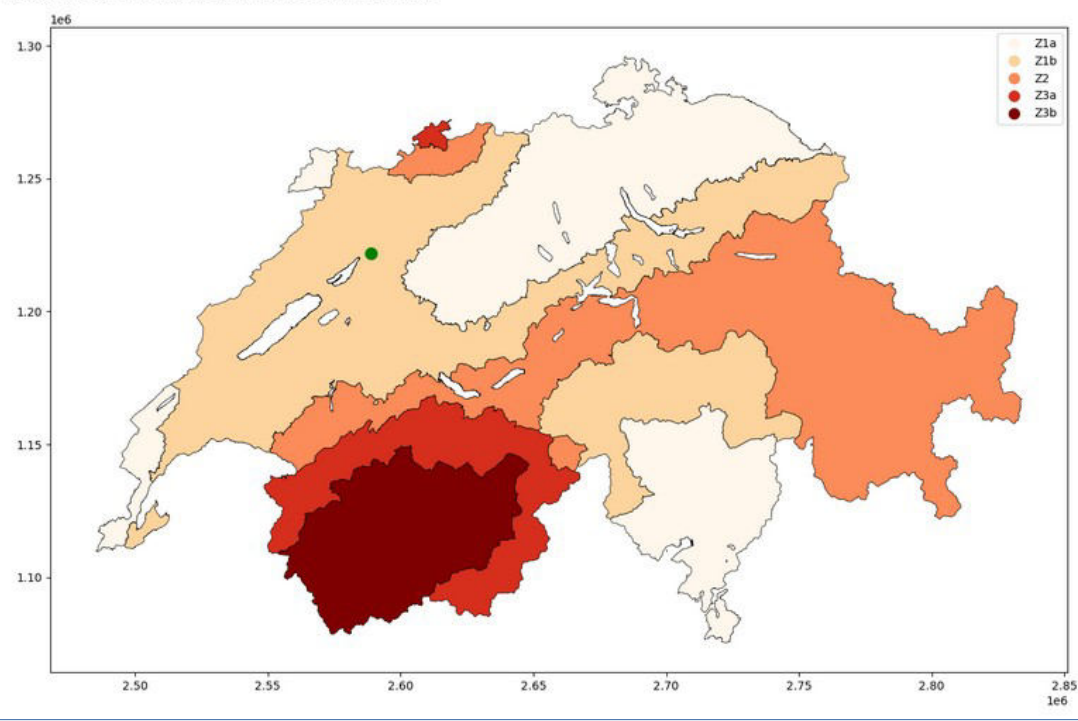
Hier ist eine kleine interaktive Anwendung, in der für einen beliebigen Punkt die Erdbebenzone ermittelt werden kann:

•••

x-Position y-Position

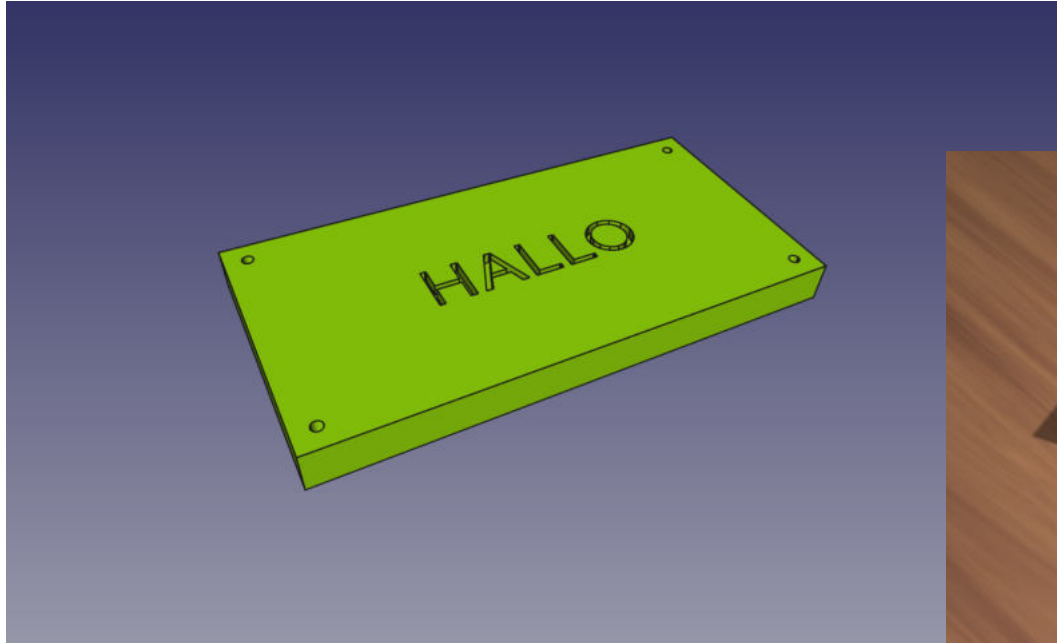
Navigation icons: Home, Up, Down, Refresh, Print, Close

Ergebnis: Der Punkt liegt in Erdbebenzone Z1b.



<https://mybinder.org/v2/gh/ronnystandtke/erdbebenzonen/HEAD?labpath=Erdbebenzonen.ipynb>

Ein kleiner optischer Vergleich...



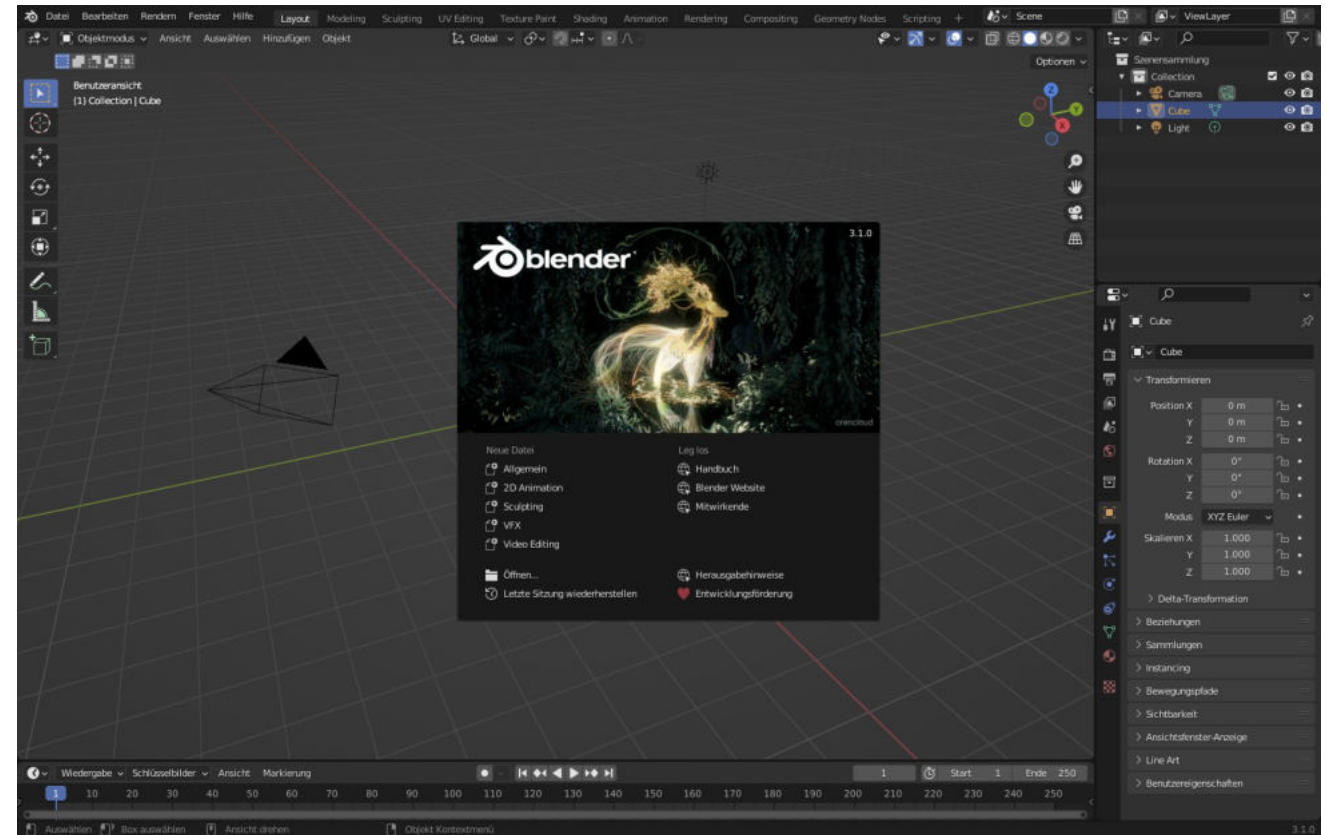
Einführung Blender



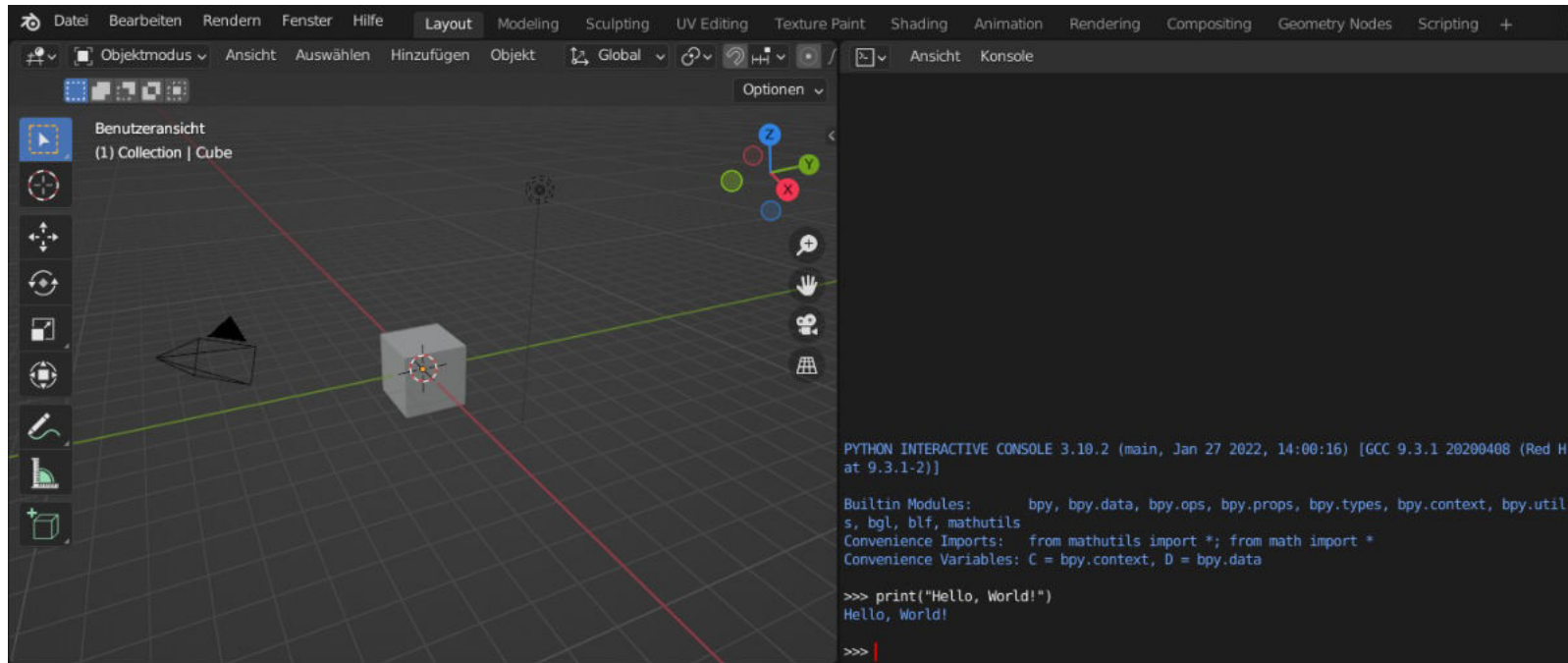
Zur Erzeugung hochqualitativer grafischer Darstellungen von CAD-Zeichnungen werden darauf spezialisierte Programme, sogenannte 3D-Grafiksuiten, verwendet.

Blender ist eine freie und leistungsstarke Software zur Erstellung von dreidimensionalen Grafiken, Animationen und Spielen, siehe:

<https://www.blender.org>



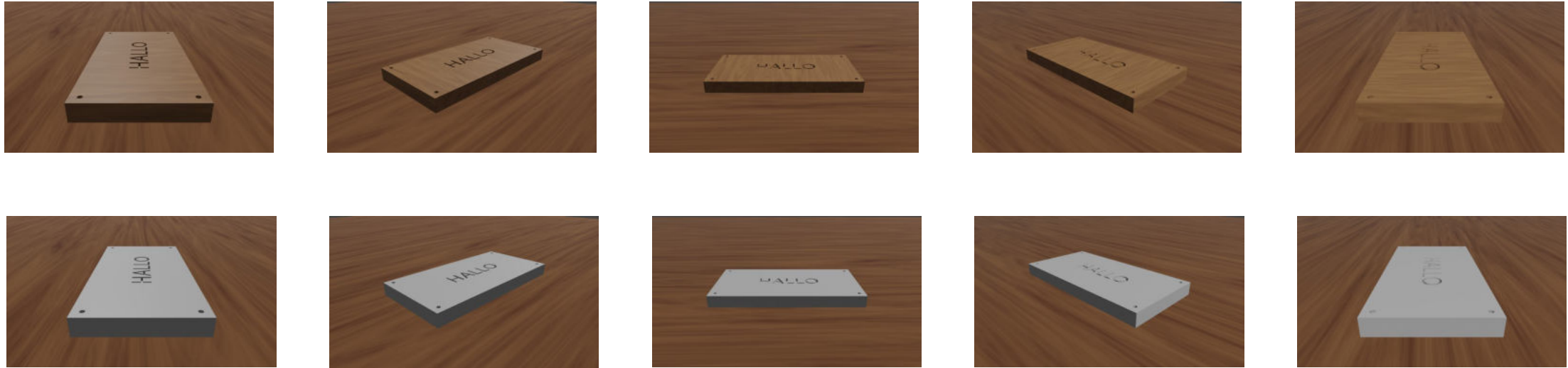
Python-Konsole



Aufgabe:

Teilen Sie das 3D-Ansichtsfenster und legen Sie als Editor für den neuen Bereich die Python-Konsole fest. Probieren Sie einige Ihnen bekannte Python-Befehle (Textausgabe, Variablen, Operatoren, Verzweigungen, Schleifen, ...).

Projektidee: Automatisches Rendering



Wir wollen ein Python-Script schreiben, dass von unserem FreeCAD-Projekt automatisch Renderings aus verschiedensten Winkeln mit unterschiedlichen Materialien anfertigt, zum Beispiel um schnell und einfach Bilder für einen Webshop erzeugen zu können.

Auf Moodle finden Sie ein vorbereitetes Blender-Projekt, dem schon mehrere Materialien und eine Ebene als «Tischplatte» hinzugefügt wurde. Für die weiteren Schritte sollten Sie dieses Projekt herunterladen und in Blender öffnen.

Animationen

Beispiel: Simulation / Animation einer Pfadplanung unseres Kuka-Roboters mit inverser Kinematik



BlenderBIM

<https://blenderbim.org>

BlenderBIM ist eine Erweiterung für Blender, mit der neben vielen anderen Funktionen auch IFC-Modelle einfach gelesen und bearbeitet werden können.



Free, open source, native IFC authoring platform

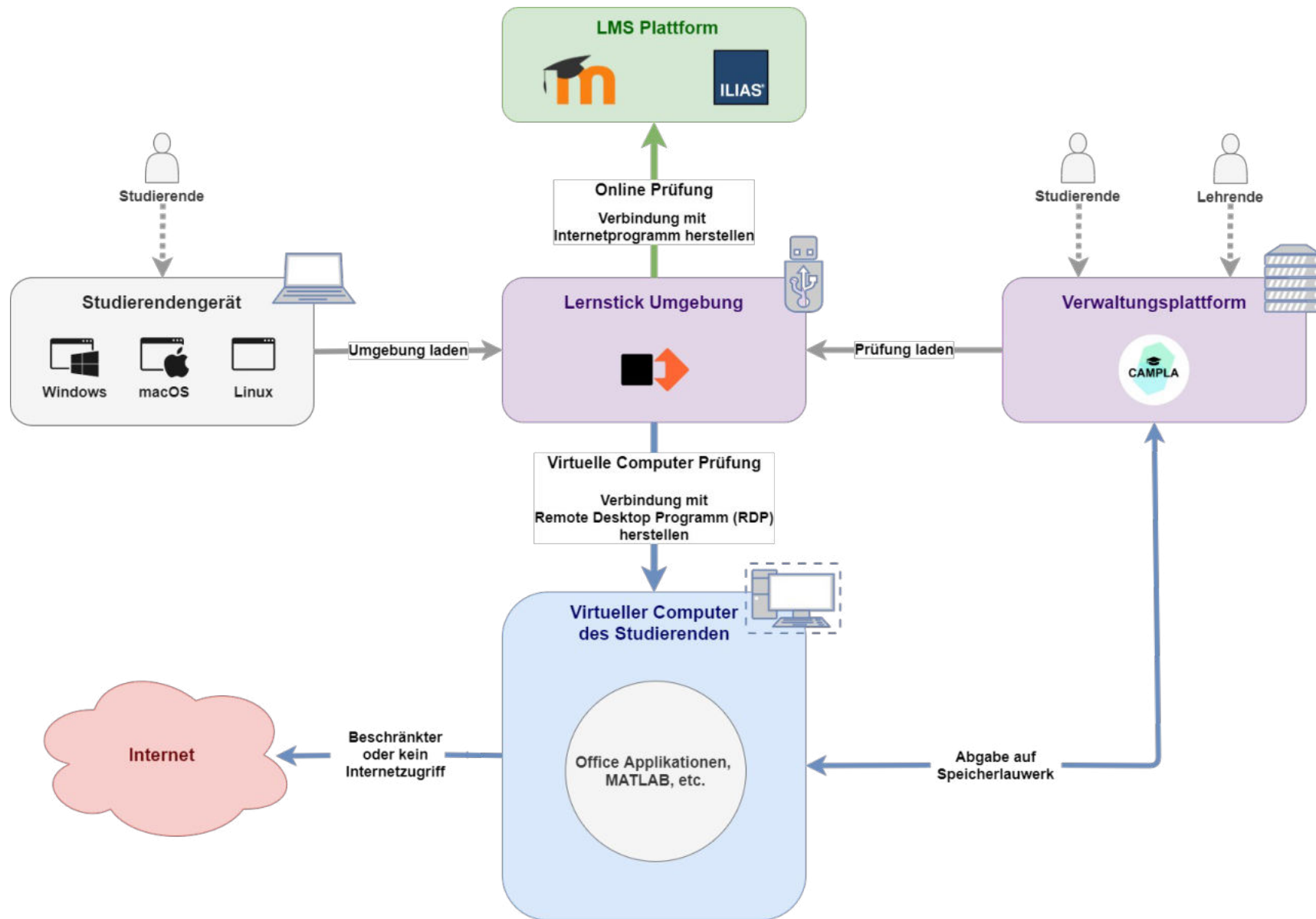
Get v0.0.230506

- Audit and analyse
- Author IFC models
- Drawing generation
- Structural analysis
- MEP systems
- Costing and scheduling
- Facility management
- Live building sensors

Python in Prüfungssituationen

CAMPLA / Lernstick

<https://campla.github.io>



CAMPLA Systemlandschaft



Hauptseite - FreeCAD Do x Neuer Tab

Mit Google suchen oder Adresse eingeben

FreeCAD Wiki FreeCAD-Reporting Blender Documentation BlenderBIM

Blender 3.6.2

Benutzeransicht (1) Collection | Cube

studio blender.org

- Neue Datei
- Leglos
- Allgemein
- Handbuch
- 2D Animation
- Blender Website
- Sculpting
- Mitwirkende
- VFX
- Video Editing
- Öffnen...
- Herausgabenweise
- Letzte Sitzung wiederherstellen
- Entwicklerförderung

Options: Position, Rotation, Modus, Skalierung, Delta-Transformation, IFC Derived Placements, Beziehungen, Sammlungen, Instancing, Bewegungspfade

FreeCAD 0.21.0

Beispiele

- ArchDetail.FCStd 225Kb
- EngineBlock.FCStd Werner Mayer 78Kb
- FemCalculixCantile ver2D.FCStd 116Kb
- FemCalculixCantile ver3D.FCStd 148Kb
- FemCalculixCantile ver3D_newSolver.F CStd 158Kb
- Schenkel.stp 576Kb
- draft_test_objects.F CStd FreeCAD Developers 117Kb

Daten angepasst werden

Menü Bearbeiten -> Einstellungen -> Start -> Zusätzlichen Ordner anzeigen

Spyder (Python 3.11)

Gebrauch

Hier können Sie Hilfe zu jedem Objekt erhalten, indem Sie **Ctrl+I** davor drücken, entweder im Editor oder in der Konsole.

Hilfe kann auch automatisch angezeigt werden, nachdem eine linke Klammer neben ein Objekt geschrieben wurde. Sie können dieses Verhalten in *Einstellungen* > *Hilfe* aktivieren.

Konsole 1/A x

```
Python 3.11.2 (main, Mar 13 2023, 12:18:29) [GCC 12.2.0]
Type "copyright", "credits" or "license()" for more information.

IPython 8.5.0 -- An enhanced Interactive Python.

In [1]:
```

Python-Konsole Chronik

