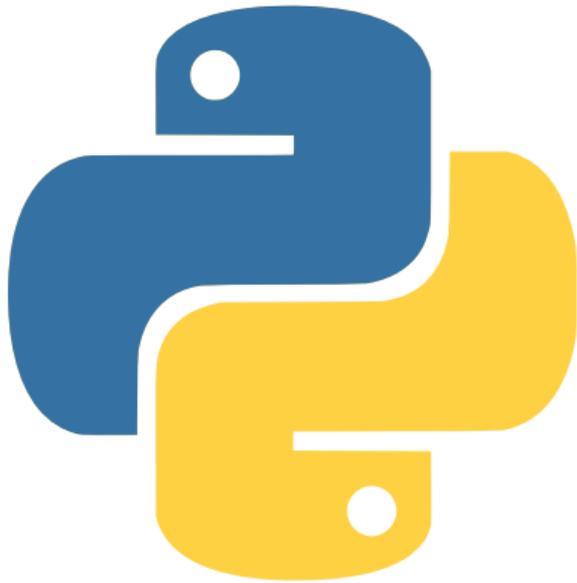


Universität Stuttgart

Projekt digit@L – BOOST. SKILLS. SUPPORT.



Dominik
Göddeke

Programmierkurs Python

Motivation: Warum
Programmieren lernen?
Technikpädagogik und NwT

Warum Programmieren lernen?

Warum Programmieren lernen?

- Programmierkenntnisse sehr hilfreich in **Studium und Berufsleben**

Warum Programmieren lernen?

- Programmierkenntnisse sehr hilfreich in **Studium und Berufsleben**
- Statistische Untersuchungen „beyond Excel“

Warum Programmieren lernen?

- Programmierkenntnisse sehr hilfreich in **Studium und Berufsleben**
- Statistische Untersuchungen „beyond Excel“
- **Spannende Projekte** im Rahmen von B.Sc. und M.Sc. Arbeiten

Warum Programmieren lernen?

- Programmierkenntnisse sehr hilfreich in **Studium und Berufsleben**
- Statistische Untersuchungen „beyond Excel“
- **Spannende Projekte** im Rahmen von B.Sc. und M.Sc. Arbeiten
- Im Berufsalltag: lehrreiche Projekte für Schüler*innen

Warum Programmieren lernen?

- Programmierkenntnisse sehr hilfreich in **Studium und Berufsleben**
- Statistische Untersuchungen „beyond Excel“
- **Spannende Projekte** im Rahmen von B.Sc. und M.Sc. Arbeiten
- Im Berufsalltag: lehrreiche Projekte für Schüler*innen
- Konzeption **neuartiger Unterrichtseinheiten**

Warum Programmieren lernen?

- Programmierkenntnisse sehr hilfreich in **Studium und Berufsleben**
- Statistische Untersuchungen „beyond Excel“
- **Spannende Projekte** im Rahmen von B.Sc. und M.Sc. Arbeiten
- Im Berufsalltag: lehrreiche Projekte für Schüler*innen
- Konzeption **neuartiger Unterrichtseinheiten**
- **Digital Literacy** an der Uni und im Berufsleben, insb. im Bezug auf AI/ML

Beispielprojekt

Statistische Auswertung und graphische Aufbereitung

Statistische Auswertung und graphische Aufbereitung

- Analyse der Volatilität des Bitcoin-Kurses

Statistische Auswertung und graphische Aufbereitung

- Analyse der Volatilität des Bitcoin-Kurses
- Viele Datenquellen im Netz, in Maschinen-lesbarem CSV-Format

Statistische Auswertung und graphische Aufbereitung

- Analyse der Volatilität des Bitcoin-Kurses
- Viele Datenquellen im Netz, in Maschinen-lesbarem CSV-Format
 - Bspw. <https://www.cryptocompare.com>

Statistische Auswertung und graphische Aufbereitung

- Analyse der Volatilität des Bitcoin-Kurses
- Viele Datenquellen im Netz, in Maschinen-lesbarem CSV-Format
 - Bspw. <https://www.cryptocompare.com>
 - Auch witzig: <https://www.angio.net/pi/digits.html>

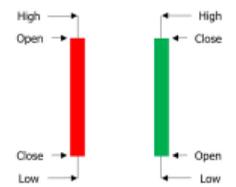
Statistische Auswertung und graphische Aufbereitung

- Analyse der Volatilität des Bitcoin-Kurses
- Viele Datenquellen im Netz, in Maschinen-lesbarem CSV-Format
 - Bspw. <https://www.cryptocompare.com>
 - Auch witzig: <https://www.angio.net/pi/digits.html>
- Hilfreich im Python-Universum: [numpy](#), [scipy](#), [pandas](#), [csv](#), ...

Statistische Auswertung und graphische Aufbereitung

- Analyse der Volatilität des Bitcoin-Kurses
- Viele Datenquellen im Netz, in Maschinen-lesbarem CSV-Format
 - Bspw. <https://www.cryptocompare.com>
 - Auch witzig: <https://www.angio.net/pi/digits.html>
- Hilfreich im Python-Universum: [numpy](#), [scipy](#), [pandas](#), [csv](#), ...

time	high	low	open	close	volume	order	direction	bid	ask	volume
2023-02-21 16:00:00	20486.3	20223.8	20425.0	2421.5	4.03403e+07	20320.9	direct	50.2051	26175	-4.66476
2023-02-21 17:00:00	20498.3	20088.8	20110.7	4827.33	3.31159e+08	20429.8	direct	81.0743	26142.1	38.5316
2023-02-21 18:00:00	20105	27028.5	27051.5	2464.54	6.01477e+07	20110.7	direct	54.2216	26114.4	30.6695
2023-02-21 19:00:00	20181.4	27029.5	20104.1	3090.21	3.09069e+08	27051.5	direct	50.2013	26085.9	32.7541
2023-02-21 20:00:00	20294.4	27090.5	20264.5	3882.03	8.10610e+07	20104.1	direct	44.6407	26056.9	35.1131
2023-02-21 21:00:00	20308	20007.8	20124.6	2562.29	7.27707e+07	20204.9	direct	50.5505	26029.6	3.48442
2023-02-21 22:00:00	20149.3	27066.3	20048.9	3396.76	9.81459e+07	20124.6	direct	54.3401	26000.6	8.18039
2023-02-21 23:00:00	20239.4	27097.3	20018.1	1043.54	5.46620e+07	20048.9	direct	54.5221	25972	34.8642
2023-02-21 00:00:00	20146.5	27011.3	27041.1	3460.48	6.87182e+07	20018.1	direct	55.7005	25944.5	37.8828
2023-02-21 01:00:00	27768.4	27019.8	27064.9	3385.03	8.83064e+07	27041.1	direct	46.6768	25918.7	31.3505
2023-02-21 02:00:00	27770.3	27411.9	27720.8	2452.33	8.75453e+07	27066.9	direct	42.7046	25891.0	38.8720
2023-02-21 03:00:00	27961.4	27716.3	27805	3344.70	9.73848e+07	27720.8	direct	45.142	25862.3	27.4446
2023-02-21 04:00:00	27939.4	27834	27999.4	873.77	2.43097e+07	27805	direct	48.529	25836.2	-27.7776
2023-02-21 05:00:00	28040.8	27809.7	27861.1	1257.96	3.52092e+07	27999.4	direct	50.6341	25797.4	-31.8308
2023-02-21 06:00:00	27912.9	27815.7	27857.2	657.74	1.83210e+07	27861.1	direct	49.0490	25763.5	-30.3427
2023-02-21 07:00:00	27977.3	27809.9	27903.7	637.09	2.26206e+07	27807.2	direct	49.0029	25731	-29.1474
2023-02-21 08:00:00	28003.4	27768.3	27829.7	886.08	2.78309e+07	27857.2	direct	51.3016	25698.3	-29.9761
2023-02-21 09:00:00	28035.7	27703	27807.3	1281.02	3.57477e+07	27935.7	direct	50.8970	25656.6	-23.5034
2023-02-21 10:00:00	28089.4	27683.3	28040.9	2147.62	5.98209e+07	27807.3	direct	58.6101	25623.4	-28.6212



Marcus Brändle, Universität Stuttgart, eigenes Programm. Abbildungen CC-0

Beispielprojekt

Machine Learning

Machine Learning

- Anwendung: KI-basierte Bilderkennung ohne eigenes Training

Machine Learning

- Anwendung: KI-basierte Bilderkennung ohne eigenes Training
 - Fertige Netze für verschiedene Objektkategorien, bspw. RetinaNet

Machine Learning

- Anwendung: KI-basierte Bilderkennung ohne eigenes Training
 - Fertige Netze für verschiedene Objektkategorien, bspw. RetinaNet
- Nur 8 Zeilen Python-Code erforderlich (!)

Machine Learning

- Anwendung: KI-basierte Bilderkennung ohne eigenes Training
 - Fertige Netze für verschiedene Objektkategorien, bspw. RetinaNet
- Nur 8 Zeilen Python-Code erforderlich (!)
 - github.com/OlafenwaMoses/ImageAI/tree/master/imageai/Detection

Machine Learning

- Anwendung: KI-basierte Bilderkennung ohne eigenes Training
 - Fertige Netze für verschiedene Objektkategorien, bspw. RetinaNet
- Nur 8 Zeilen Python-Code erforderlich (!)
 - github.com/OlafenwaMoses/ImageAI/tree/master/imageai/Detection

```
1 from imageai.Detection import ObjectDetection
2 import os
3
4 execution_path = os.getcwd()
5
6 detector = ObjectDetection()
7 detector.setModelTypeAsRetinaNet()
8 detector.setModelPath( os.path.join(execution_path ,
9 "retinanet_resnet50_fpn_coco-eeacb38b.pth") )
9 detector.loadModel()
10
11 detections = detector.detectObjectsFromImage(input_image=os.path.join(execution_path
, "parkinglot.jpg"), output_image_path=os.path.join(execution_path ,
"parkinglotnew.jpg"), minimum_percentage_probability=30)
```

Machine Learning

- Trainierte Objekte bspw. stop_sign, truck, car, bus

Machine Learning

- Trainierte Objekte bspw. stop_sign, truck, car, bus
- Recht gute Erfolgsquote



Original image: The U.S. National Archives, <https://nara.getarchive.net/media/an-overview-of-a-parking-lot-vehicles-and-trees-in-the-aftermath-of-mount-pinatubos-258753>, CC0

Machine Learning

- Trainierte Objekte knife, fork, spoon

Machine Learning

- Trainierte Objekte knife, fork, spoon
- Totalversagen



Original image: Karen Arnold,
<https://www.publicdomainpictures.net/en/view-image.php?image=335183&picture=cutlery-set-clipart>, CC0

Beispielprojekt

Programmieren ohne Quelltext

Programmieren ohne Quelltext

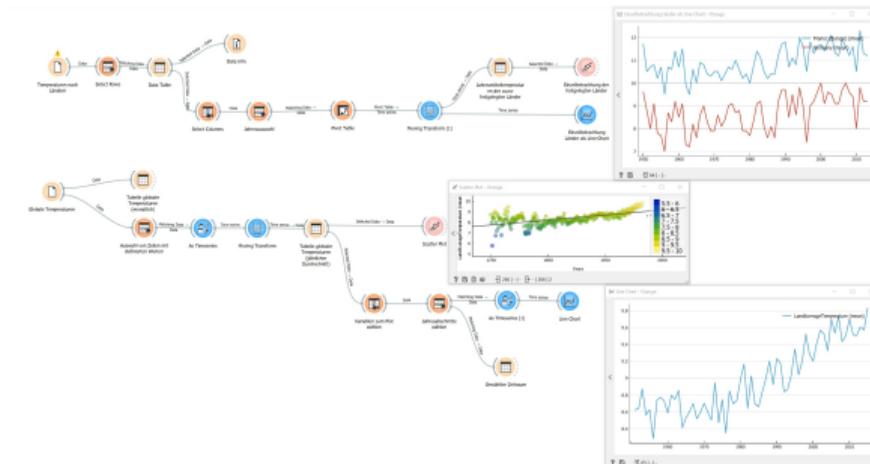
- Low-barrier Ansatz, „Zusammenklicken“ einer Lösung

Programmieren ohne Quelltext

- Low-barrier Ansatz, „Zusammenklicken“ einer Lösung
- Lego Mindstorms, hier **Orange** für statistische Datenaufbereitung, ...

Programmieren ohne Quelltext

- Low-barrier Ansatz, „Zusammenklicken“ einer Lösung
- Lego Mindstorms, hier **Orange** für statistische Datenaufbereitung, ...



Beispiel: Marcus Brändle, Universität Stuttgart.

Orange non-code content: Orange, Data Mining Fruitful & Fun, <https://orange.biolab.si/>, CC-BY-SA

Beispielprojekt

Prozedurale Computerkunst

Prozedurale Computerkunst

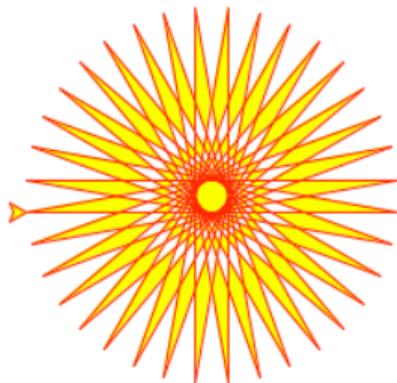
- Turtle Grafik: <https://de.wikipedia.org/wiki/Turtle-Grafik>
 - Kindheitserinnerung des Dozenten: Spirograph

Prozedurale Computerkunst

- Turtle Grafik: <https://de.wikipedia.org/wiki/Turtle-Grafik>
 - Kindheitserinnerung des Dozenten: Spirograph
- Python-Paket: `turtle`

Prozedurale Computerkunst

- Turtle Grafik: <https://de.wikipedia.org/wiki/Turtle-Grafik>
 - Kindheitserinnerung des Dozenten: Spirograph
- Python-Paket: `turtle`



```
from turtle import *
color('red', 'yellow')
begin_fill()
while True:
    forward(200)
    left(170)
    if abs(pos()) < 1:
        break
end_fill()
done()
```

Copyright 2001-2023, Python Software Foundation, Zero Clause BSD License for Python software and documentation,
<https://docs.python.org/3/license.html#bsd0>

Beispielprojekt

Anfassbare Mathematik: 3D Druck der platonischen Körper

Anfassbare Mathematik: 3D Druck platonischer Körper

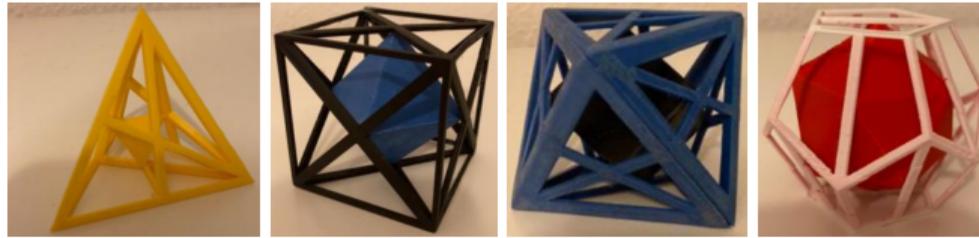
- Haptisch verständliche Repräsentationen komplexer Zusammenhänge

Anfassbare Mathematik: 3D Druck platonischer Körper

- Haptisch verständliche Repräsentationen komplexer Zusammenhänge
- Motivation: Mathematik begreifbar machen als 3D-Unterrichtsmodelle

Anfassbare Mathematik: 3D Druck platonischer Körper

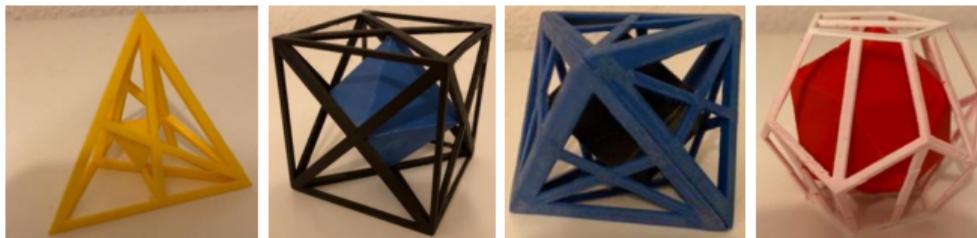
- Haptisch verständliche Repräsentationen komplexer Zusammenhänge
- Motivation: Mathematik begreifbar machen als 3D-Unterrichtsmodelle
 - Programmierung und 3D-Druck der sechs platonischen Körper sowie ihrer dualen Körpern



Maria Zinn, Goethe Universität Frankfurt am Main, Studienarbeit. Abbildungen CC-0

Anfassbare Mathematik: 3D Druck platonischer Körper

- Haptisch verständliche Repräsentationen komplexer Zusammenhänge
- Motivation: Mathematik begreifbar machen als 3D-Unterrichtsmodelle
 - Programmierung und 3D-Druck der sechs platonischen Körper sowie ihrer dualen Körpern



Maria Zinn, Goethe Universität Frankfurt am Main, Studienarbeit. Abbildungen CC-0

- Konzeption von Unterrichtsideen im Kontext der Bildungsplanvorgaben der Sekundarstufe I/II

Impressum, Danksagung und Quellen



Stiftung
Innovation in der
Hochschullehre



Gefördert durch die Stiftung Innovation in der Hochschullehre im Rahmen des Projekts digit@L, <https://stiftung-hochschullehre.de>

Gefördert mit Mitteln der Deutschen Forschungsgemeinschaft (EXC 2075 - 390740016) im Rahmen der Exzellenzstrategie

Autor: Dominik Göddeke, IANS, Universität Stuttgart



Weitere Quellen:

- Logos Universität Stuttgart, IANS, SimTech: Universität Stuttgart, alle Rechte vorbehalten
- Logo Python: <https://freesvg.org/387>, CC-0
- Logo Stiftung: Stiftung Innovation in der Hochschullehre, alle Rechte vorbehalten
- Logo ZOERR: Universität Tübingen, alle Rechte vorbehalten



Veröffentlicht auf dem Zentralen OER Repositorium Baden-Württemberg, <https://www.zoerr.de>