

Programmieren

Ingenieurinformatik Teil 1, Wintersemester 2026/27

David Straub

Ziel dieser Veranstaltung

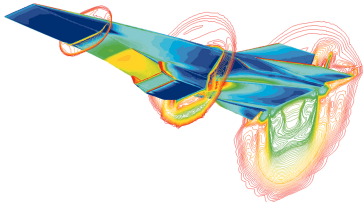
Sie in die Lage zu versetzen, Software zu produzieren,

- die Sie in Ihrem Studium und Beruf produktiver macht
- mit der Sie sich bei Ihren zukünftigen Teammitgliedern (und Ihrem zukünftigen Ich) nicht unbeliebt machen

Warum programmieren lernen?

Moderne Ingenieurarbeit läuft zunehmend durch Code.

- **Simulation:** Systeme am Computer testen und optimieren, bevor sie existieren
- **Messdaten:** vom Windkanal bis zum Prüfstand – ausgewertet wird mit Software
- **Zunehmend auch der Entwurf selbst:** parametrische CAD-Modelle, automatisierte Variantenstudien
- Und konkret für Sie: Numerik im 2. Semester, Praktika, Projektarbeit, Bachelorarbeit



Demo: Was Sie am Ende des Semesters können

Messwerte auswerten und darstellen – in 12 Zeilen:

```
import matplotlib.pyplot as plt

stunden = [8, 9, 10, 11, 12, 13, 14, 15]
temperaturen = [18.2, 19.1, 21.4, 24.0, 26.3, 27.1, 25.8, 22.5]

maximum = max(temperaturen)
print(f"Höchstwert: {maximum} °C")

plt.plot(stunden, temperaturen, "b-o")
plt.xlabel("Uhrzeit")
plt.ylabel("Temperatur in °C")
```

```
plt.grid(True)
plt.show()
```

Das war übrigens: Python

```
print("Hallo Welt!")
```

Zum Vergleich – dasselbe in Java:

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hallo Welt!");
    }
}
```

Warum Python?

Die einfachste der weitverbreiteten Programmiersprachen – und der Standard für genau das, was Ingenieur:innen brauchen:

- Simulation und wissenschaftliches Rechnen
- Datenanalyse und Visualisierung
- KI und Machine Learning

Quelloffen und kostenlos.

Und die KI-Frage: Löst ChatGPT das nicht alles?

Probieren wir es aus – mit einer echten Klausuraufgabe:

<https://kuepper.userweb.mwn.de/informatik/python-ws2024.pdf>

Finden Sie den Fehler

Eine Batteriezelle wird in 0,1-Schritten geladen, bis der Ladezustand genau 1,0 erreicht ist:

```
ladezustand = 0.0
while ladezustand != 1.0:
    ladezustand = ladezustand + 0.1
print("Batterie voll!")
```

Zehnmal 0,1 ist 1,0. Sieht gut aus. Läuft ohne Fehlermeldung. **Würden Sie es abnehmen?**

Warum Sie trotzdem fliegen lernen

Autopiloten fliegen präziser als Menschen.

Trotzdem beginnt jede Pilotenausbildung mit manuellem Fliegen – denn wer die Automation überwachen, ihre Fehler erkennen und die Verantwortung tragen soll, muss es selbst können.

Beim Programmieren gilt dasselbe:

- **Spezifizieren:** der KI präzise sagen, was gebraucht wird
- **Prüfen:** erkennen, ob das Ergebnis stimmt
- **Verantworten:** am Ende unterschreibt der Mensch

Formelles

- Modul Ingenieurinformatik (L1170)
 - **1. Semester: Programmierung (L1171)** <–
 - 2. Semester: Numerik für Ingenieure (L1172)
- Prüfung: 60 Minuten schriftlich, Teil 1 zählt 60%
 - eigene Unterlagen erlaubt, keine elektronischen Geräte

Termine

- 2 SWS Seminaristischer Unterricht
- 1 SWS (90 Min. alle 2 Wochen) Praktikum

Semesterfahrplan

Wochen	Was passiert
Heute	Einführung + Ihr erstes Programm
Woche 2–12	Der gesamte Stoff, Woche für Woche
Weihnachten	Übungsmaterial für die Ferien
1. Woche im Januar	Probeklausur unter Realbedingungen
2. Woche im Januar	Klausurwerkstatt: Besprechung + Training
danach	Prüfung

Gruppeneinteilung Praktikum

14-tägig im Wechsel, Beginn **12.10.**

Studiengruppe	Gruppe	Tag	Uhrzeit	Raum	Start
LRB1A	1	TBD	TBD	TBD	TBD
LRB1A	2	TBD	TBD	TBD	TBD
LRB1A	3	TBD	TBD	TBD	TBD
LRB1B	1	TBD	TBD	TBD	TBD
LRB1B	2	TBD	TBD	TBD	TBD
LRB1B	3	TBD	TBD	TBD	TBD

Matrix-Chat

<https://matrix.hm.edu>, Element-App (Desktop/Mobil)

Ein gemeinsamer Raum für alle: TBD

Bitte mit Foto im Profil 📷



Jetzt Sie: Ihr erstes Programm

Kein Installieren nötig – im Browser, auch am Handy:

- <https://davidstraub.de/teaching-materials/lite/lab/>
- dort: File → New → Notebook
- Ihre Dateien bleiben nur in **diesem Browser auf diesem Gerät** gespeichert – Wichtiges am Ende herunterladen!

Mini-Aufgabe (3 min):

1. Führen Sie aus:

```
print("Hallo LRB!")
```

2. Ändern Sie den Text und führen Sie erneut aus
3. Führen Sie aus: `print(3 * 7)` – und dann `print(3 * 7.0)`

Was ist da gerade passiert?

- Sie haben dem Computer eine **Anweisung** gegeben – er hat sie ausgeführt
- `print(...)` ist eine **Funktion**: Sie geben etwas hinein, sie tut etwas
- `3 * 7` und `3 * 7.0` – warum sieht das Ergebnis unterschiedlich aus?
- Merken Sie sich schon heute: **gleiche Rechnung, andere Darstellung der Zahl** – derselbe Unterschied steckt hinter der hängenden Batterie-Schleife von vorhin (*die volle Erklärung bekommen Sie in Woche 11!*)

So arbeiten wir in diesem Kurs

Jede Woche:

- Kurze Erklärungen im Wechsel mit **Mini-Aufgaben** – alle machen mit, am Gerät oder auf Papier
- Größere Aufgaben: erst **nachdenken** (Papier), dann entwickeln wir die Lösung **gemeinsam live**
- Kein Laptop dabei? Kein Problem: zu zweit arbeiten, Vorhersagen und Fehlersuche gehen auf Papier

Alle Folien zum Mitlesen und als PDF: <https://davidstraub.de/teaching-materials/>

Spielregeln

- Im **Praktikum**: ohne KI-Unterstützung programmieren – Fehler selbst zu finden ist der Kern der Sache
- **Außerhalb**: KI-Tools ausdrücklich erwünscht – zum Erklären, Üben, Vertiefen
 - z. B. KI-Quizze zu den Kursinhalten auf OneTutor: <https://hm.onetutor.ai/>
- Suchen Sie sich Programmierprojekte: Open Source, Smart Home, KI-Automatisierung, ...

Online-Ressourcen

- KI-Quizze zu den Kursinhalten (OneTutor): <https://hm.onetutor.ai/>
- Kursunterlagen „Ingenieurinformatik“ von Dr. Christina Mayr: <https://ingenieurinformatik-buch-fcbc5c.pages.gitlab.lrz.de>
- Offizielles Python-Tutorial: <https://docs.python.org/3/tutorial/index.html>
- OpenStax-Lehrbuch „Introduction to Python Programming“: <https://openstax.org/details/books/introduction-python-programming>
- Tutorial „Research Software Engineering with Python“: <https://alan-turing-institute.github.io/rse-course/html/index.html>

Bis nächste Woche!

Nächste Woche: **Erste Programme** – Variablen, Zahlen, Texte

Fragen?

- Jederzeit: im Matrix-Chat
- Persönlich: Sprechstunde TBD, Büro B 374