

# Einleitung

---

»Dem Ingenieur ist nichts zu schwör!« – Diese Herangehensweise hat über viele Jahre mehr oder weniger das Bild von Ingenieurinnen und Ingenieuren geprägt. Seit der Entstehungszeit dieses Credo in der Ära der Reißbretter und Tabellenbücher haben sich die Ingenieurwissenschaften aber erheblich weiterentwickelt. Digitale Werkzeuge bestimmen mehr denn je den Ingenieuralltag. Sie ermöglichen es, in kurzer Zeit Probleme zu lösen, die noch vor einer Generation unlösbar schienen. Mit diesen gestiegenen (und permanent weiter steigenden) Möglichkeiten wachsen aber auch die Erwartungshaltung und die Anforderung an Ingenieur:innen.

Der souveräne Umgang mit digitalen Werkzeugen ist kein Selbstläufer, sondern muss mit Mühe und Aufwand erlernt werden. Um so ärgerlicher ist es, wenn die hart erarbeiteten Fähigkeiten aus der Mode kommen oder sich aufgrund von Lizenzbeschränkungen (zum Beispiel beim Wechsel aus der Hochschule in ein Startup) nicht anwenden lassen. Die Programmiersprache Python und das darauf basierende Ökosystem aus Zusatzpaketen bieten vor diesem Hintergrund eine exzellente Wahl. Einerseits steigt die Popularität von Python seit den 1990ern moderat, aber beständig – das genaue Gegenteil von sonst häufigen Hype-Zyklen. Zum anderen sind alle relevanten Komponenten freie Software und plattformunabhängig – mehr Freiheit geht nicht.

Leider ist die Ingenieur-Ausbildung teilweise erstaunlich ~~rückständig~~ konservativ, weswegen das Lösen von Ingenieur-Problemen mithilfe digitaler Werkzeuge bisher kaum als Lehrveranstaltung angeboten wird – oder wenn, dann oft als kommerziell motivierte externe »Produktschulung« mit den oben angedeuteten Lizenz- und Abhängigkeitsproblemen. Das vorliegende Buch soll dem Abhilfe schaffen.

## Über dieses Buch

---

Dieses Buch soll Sie befähigen, vielfältige Aufgabenstellungen mit Python zu lösen. Es vermittelt dabei sowohl das grundsätzliche Herangehen als auch ganz konkrete Tipps und Tricks. Um dieses Wissen möglichst interessant, nachvollziehbar und anwendungsfreundlich darzustellen, werden Sie in diesem Buch auf eine Vielzahl von Beispielen treffen. Manche nur für einen dreizeiligen Codeschnipsel, andere für ein komplexeres Szenario, das sich einer praxisrelevanten Fragestellung widmet. Dabei finden Sie einen thematischen Streifzug durch viele Ingenieurdisziplinen: Von der Biegelinie eines Balkens über optimale Mischverhältnisse in der Verfahrenstechnik und die  $j\omega$ -Wechselstromrechnung bis zur Fourier-Analyse und Simulation eines Regelkreises im Zustandsraum ist (fast) alles dabei, was das Herz begehrt.

Zum Ingenieuralltag gehört auch der Umgang mit Zielkonflikten: Ein Fahrradrahmen kann zum Beispiel entweder extrem leicht oder extrem stabil sein. Am sinnvollsten ist aber ein auf typische Anwendungsfälle zugeschnittener Kompromiss. So ähnlich ist es auch bei diesem Buch: Die behandelten Code-Beispiele könnten entweder sehr einfach gestrickt sein, wären

aber dafür weit weg von jedem Anwendungsbezug. Das andere Extrem wäre, an jedes Beispiel einen harten Realismusanspruch zu stellen und dabei die Verständlichkeit außer Acht zu lassen.

Ziel des Buches ist es, für eine möglichst große Zielgruppe möglichst oft den optimalen Bereich zwischen langweiliger Trivialität und demotivierender Komplexität zu treffen.

# Törichte Annahmen über den Leser (m/w/d)

---

Diese Buch richtet sich an Menschen, die sich für Technik interessieren, wie sie funktioniert und wie man sie verbessern kann. Wir vermuten, dass sich dieses Interesse in einer gewissen Vorbildung niedergeschlagen hat und wollen unsere Zielgruppe weder langweilen noch überfordern. Idealerweise haben Sie

- ✓ vier Semester oder mehr eines Ingenieurstudiengangs auf dem Buckel,
- ✓ keine Angst vor gelegentlichen Gleichungen,
- ✓ schon ein bisschen Programmiererfahrung (Variablen, Typen, Verzweigungen, Schleifen, ...),
- ✓ einen Computer, dem üblicherweise Sie Anweisungen geben (und nicht umgekehrt),
- ✓ »Mut zur Lücke« – also die Bereitschaft, auch mal etwas grob zu überfliegen, was für Sie aktuell gerade nicht so relevant ist, und
- ✓ Sinn für Humor sowie die Bereitschaft, Fehler zu machen, Fehler zu suchen und daraus zu lernen.

## Ein Wort zur Sprache

---

Die Entstehung des Buches fällt in eine Zeit, in der mitunter sehr aufgeregt emotional über das Für und Wider verschiedener Formulierungen gestritten wird. Einige potenzielle Leser:innen werden das Buch vielleicht hochmütig ignoriert haben, weil der grammatikalisch maskuline Titel »für Ingenieure« ihrem Anspruch an geschlechtsneutrale Sprache zuwiderläuft. Andere werden beim ersten Auftreten von Syntaxinnovationen wie »Ingenieur:innen« einen mittleren Empörungsanfall bekommen und eine rituelle Verbrennung des Buches in Erwägung ziehen. Wir vertrauen jedoch darauf, dass die große Mehrzahl der Leserinnen und Leser ihren gesunden Menschenverstand einsetzt und weder der einen noch der anderen Überreaktion zum Opfer fällt. Selbstverständlich sind mit dem Titel und anderen maskulinen oder femininen Formulierungen grundsätzlich alle angesprochen, unabhängig davon, welchem Geschlecht sie sich zuordnen. Es sollte aber nicht überraschen, dass uns als Autoren eines Buches voller formaler Logik und wichtiger (programmier-)sprachlicher Detailnuancen das Konzept des »generischen Maskulinums«, wonach »der Ingenieur« ein neutraler Oberbegriff von »der Ingenieur« und »die Ingenieurin« sein soll, nicht so 100 %

überzeugt. Andererseits haben wir auch (noch) keine ultimative Lösung und behelfen uns daher mit einem Mix aus klassischem und innovativem Sprachgebrauch. Ein ähnlicher Pragmatismus gilt in diesem Buch übrigens auch für den Umgang mit Anglizismen.

## Was Sie nicht lesen müssen

---

Nicht alles in diesem Buch ist für alle Leser:innen gleich wichtig. Grundsätzlich gilt, dass der Inhalt weiter vorn von eher allgemeinem Interesse ist und nach hinten immer spezieller wird. Andererseits kann es natürlich sein, dass Sie schon ein paar Erfahrungen mit Python haben und die ersten Kapitel deshalb nicht benötigen. Grundsätzlich gilt: Lesen Sie nur das, was Sie interessiert! Wenn Sie in einem der hinteren Kapitel ein Sprachkonstrukt oder einen Funktionsaufruf nicht verstehen, lohnt sich aber ein Blick in die vorderen (und ins Stichwortverzeichnis). Die Chance ist groß, dort Aufklärung zu finden. Wenn Sie inhaltlich mal bei einem Beispiel aussteigen: Machen Sie sich nichts draus. Die Ingenieurwissenschaften sind ein weites Feld und niemand kann alles.

Eine erfolgreiche Herangehensweise aus der Informatik ist es, in Schnittstellen zu denken. Sie müssen nicht wissen, wie ein Magnetron funktioniert (oder was das ist), um einen Mikrowellenherd zu benutzen. Es reicht, die Schnittstellen zu verstehen (Timer, Türkнопf, »Pling«). Genauso ist es mit Code. Sie müssen nicht immer im Detail verstehen, wie eine Funktion oder ein Code-Abschnitt genau arbeitet. Oft reicht es zu verstehen, was die Eingangsdaten sind und was das Ergebnis ist oder bewirkt. Konzentrieren Sie Ihre Aufmerksamkeit auf die Bereiche, die aktuell relevant für Sie sind.

## Wie dieses Buch aufgebaut ist

---

Das Buch ist in folgende Teile untergliedert:

### Teil I: Los geht's

Hier setzen wir den Zug aufs Gleis und rollen behutsam los. Den Anfang macht das (möglicherweise) allererste Ausführen eines Python-Programms – schön bequem online im Browser. Es folgen Installation und Inbetriebnahme sowie ein kompakter Überblick über die wichtigsten Sprachelemente von Python.

### Teil II: Rechnen und Plotten

Der zweite Teil beinhaltet gewissermaßen das Grundstudium des Buches und das heißt, es kommt einiges an Mathematik vor. Aber natürlich müssen Sie nicht seitenweise mit Zahlenkolonnen oder Buchstabensalaten rumrechnen. Diese Arbeit soll Ihnen Python ja abnehmen und hier lernen Sie, wie das funktioniert. Außerdem geht es um Visualisierung, also wie sie aus Zahlen und Formeln schöne Kurven und Bilder machen.

## Teil III: Fortgeschrittene Ingenieursmethoden

Jetzt geht es ans Eingemachte: Optimierung, Kurven fitten, Simulation, Regelung, Fourier-Analyse und diverse statistische Untersuchungen. Dazu noch ein eigenes Kapitel über Mechanik, vom verbogenen Balken bis zum schwingenden Doppelpendel. Diese Methoden kommen in vielen Ingenieursanwendungen vor und bilden mit hoher Wahrscheinlichkeit den produktiven Kern Ihrer zukünftigen Python-Fähigkeiten.

## Teil IV: Schneller, schöner, robuster

Zwischen einem »Es funktioniert ... irgendwie«-Prototyp und einem anwendungsbereiten Endprodukt liegt oft noch ein gutes Stück Arbeit. In diesem Teil lernen Sie, wie Sie ihren Code schneller machen, aber auch wie Sie mithilfe von Modularisierung, automatischen Tests und Objektorientierung unübersichtlichen Spaghetti-Code verhindern und auch mit größerer Komplexität klarkommen.

## Teil V: Der Top Ten Teil

Dieser Teil beinhaltet übersichtliche Auflistungen von nützlichen Paketen, typischen Fehlern und coolen Tricks. Außerdem sind hier die wichtigsten Links zu guten Referenzen und Dokumentationen zusammengefasst.

# Symbole, die in diesem Buch verwendet werden

---

Um den Text in überschaubare Häppchen zu gliedern und die Art des Inhalts gleich optisch deutlich zu machen, gibt es die folgenden Symbol-Abschnitte:



Mit der »Glühbirne« werden konkrete Tipps mit potenziell hoher Praxisrelevanz hervorgehoben.



Der Techniker kennzeichnet Abschnitte mit Hintergrundinfos und Detailwissen. Hier kann es schon ein bisschen spezieller werden. Gut für all jene, die es möglichst genau wissen wollen.



Das Vorsicht-Symbol soll Ihre Aufmerksamkeit auf mögliche Probleme oder Tücken lenken. Der alte Spruch »Gefahr erkannt ⇒ Gefahr gebannt« gilt schließlich auch beim Programmieren.



Das Fernglas signalisiert die Anwesenheit eines Beispiels.



Das Wegweisersymbol hebt interessante externe Quellen hervor, vom Mathematikbuch bis zur *SciPy*-Referenz.

## Wie es weiter geht

---

Um es gleich mal klar zu machen: Programmieren lernt man durch Programmieren. Dieses Buch von vorn bis hinten im Gartenstuhl sitzend oder auf dem Sofa liegend durchzulesen, ist also nicht die beste Idee. Je mehr Sie dagegen am Computer selber ausprobieren, umso mehr Lernerfolg und Nutzen werden Sie durch dieses Buch haben. Dafür haben wir eine Vielzahl an ausführlich kommentierten Codeschnipseln in den Text eingebaut. Zum Ausprobieren müssen Sie diesen Code natürlich nicht abtippen. Er steht unter <https://www.wiley-vch.de/ISBN9783527717675> sowie unter <https://python-fuer-ingenieure.de> zum Herunterladen oder direkten Online-Ausprobieren zur Verfügung. Neben der gesparten Tipparbeit gibt es zwei weitere Vorteile des Online-Begleitmaterials: Die aus dem Code resultierenden Abbildungen sind dort direkt nach dem jeweiligen Schnipsel in voller Farbenpracht eingebettet und alle Links aus dem Buch sind direkt anklickbar aufgelistet.

Das bedeutet aber nicht, dass wir Ihnen grundsätzlich vom Selbertippen abraten, im Gegenteil. Der Lerneffekt durch reines Kopieren und Ausführen unserer Beispiele und das bloße Abnicken der Ergebnisse ist sehr überschaubar. Stattdessen sollten Sie Ihrem Spieltrieb freien Lauf lassen und den Code nach eigenen Vorstellungen abändern, zerlegen und neu zusammenbauen. Haben Sie dabei keine Angst vor Fehlermeldungen! Die gehören zum Programmieren dazu und sind dazu da, um Ihnen bei der Problemlösung zu helfen.

Wenn Sie ein Buch »ordentlich durcharbeiten«, geht das an diesem natürlich nicht spurlos vorbei. Im Gegenteil: Unterstreichungen, Randnotizen und Eselsohren sind explizit erwünscht und helfen erwiesenermaßen beim Verstehen und Behalten.