

1. Was ist eigentlich ein Algorithmus?

Stell dir vor, du möchtest einem Roboter beibringen, wie man ein Marmeladenbrot schmiert. Der Roboter weiß nichts – er braucht eine ganz genaue Anleitung. Wenn du sagst: "Tu die Marmelade aufs Brot", würde er vielleicht das ganze Glas auf die Scheibe werfen.

Ein **Algorithmus** ist genau das: Eine Schritt-für-Schritt-Anleitung zur Lösung eines Problems. Er besteht aus klaren Einzelschritten, die nacheinander abgearbeitet werden. Damit ein Computer (der ja eigentlich nur Strom und Plastik ist) das versteht, muss die Anleitung extrem genau sein.

Ein Algorithmus muss drei wichtige Eigenschaften haben:

1. **Eindeutigkeit:** Jeder Schritt muss so klar sein, dass man ihn nicht missverstehen kann.
2. **Ausführbarkeit:** Jeder Schritt muss machbar sein (z.B. kann ein Mensch nicht "fliege zum Mond" ausführen).
3. **Endlichkeit:** Die Anleitung muss irgendwann zu einem Ende führen.

Wir nutzen Algorithmen ständig: beim Kochen nach Rezept, beim Aufbauen von LEGO-Sets oder beim Lösen einer Matheaufgabe. Computer nutzen sie, um uns den Weg auf Google Maps zu zeigen oder uns Videos auf TikTok vorzuschlagen.

2. Das EVA-Prinzip: So arbeitet der Computer

Egal wie kompliziert ein Programm wirkt, es folgt fast immer dem **EVA-Prinzip**. Das ist das Grundgesetz der Informatik:

Eingabe: Du gibst dem Computer Informationen (z. B. du tippsst eine Nachricht, drückst einen Knopf oder eine Kamera erkennt dein Gesicht).

Verarbeitung: Der Algorithmus arbeitet. Er rechnet, vergleicht und entscheidet nach festen Regeln.

Ausgabe: Das Ergebnis wird angezeigt (z. B. die Nachricht wird abgeschickt, die Waschmaschine startet oder das Handy entsperrt sich).

3. Algorithmen lauern überall

Du begegnest ihnen jeden Tag, oft ohne es zu merken:

Beim Kochen: Ein Rezept ist ein Algorithmus. (Eingabe: Zutaten \rightarrow Verarbeitung: Kochen \rightarrow Ausgabe: Essen).

Im Navi: Es berechnet aus Millionen Straßen den schnellsten Weg.

Beim Gaming: Wenn du einen Knopf drückst, berechnet der Algorithmus, ob dein Charakter springt oder schießt.

In Social Media: Ein Algorithmus beobachtet, was du lange anschaust, und zeigt dir mehr davon.

4. Computational Thinking: Denken wie ein Informatiker

Man muss kein Programmierer sein, um wie einer zu denken. Wenn wir große Probleme lösen wollen, nutzen wir **Computational Thinking** (informatisches Denken). Das besteht aus 4 einfachen Schritten:

- a. **Aufteilen (Zersetzen):** Ein großes Problem ist gruselig. Zerlege es in viele kleine, machbare Häppchen. (Statt "Zimmer aufräumen" sagst du: "Zuerst die Socken, dann die Bücher, dann der Müll").
- b. **Muster erkennen (Analysieren):** Schau genau hin. Habe ich dieses Problem schon mal gelöst? Gibt es Ähnlichkeiten?
- c. **Vereinfachen (Abstraktion):** Lass unwichtige Details weg. Wenn du eine Karte von der Schule zeichnest, musst du nicht jeden Grashalm malen – nur die wichtigen Gebäude.
- d. **Schritte folgen (Algorithmus):** Erstelle die fertige Anleitung, die man einfach nur noch abarbeiten muss.

5. Sortieralgorithmen: Ordnung ins Chaos bringen

Computer müssen ständig Dinge sortieren: Kontakte im Handy nach dem Alphabet, Preise im Online-Shop von "günstig nach teuer" oder Highscores in Spielen. Damit das schnell geht, nutzen sie **Sortieralgorithmen**.

Warum brauchen wir das?

Stell dir vor, du hast 1.000 unsortierte Spielkarten und suchst die Herz-Dame. Ohne Sortierung musst du im schlimmsten Fall alle 1.000 Karten einzeln anschauen. Sind sie sortiert, findest du sie sofort.

1. Bubble-Sort (Der langsame Tauscher)

Stell dir vor, Zahlen sind wie **Luftblasen** im Wasser. Die schweren (großen) Zahlen sinken nach unten bzw. wandern im Register ganz nach rechts.

- **So geht's:** Du schaust dir immer **zwei Nachbarn** an. Ist der linke größer als der rechte? Dann tauschen sie die Plätze. Das machst du so lange, bis die größte Zahl ganz hinten "angeploppt" ist.
- **Vorteil:** Super einfach zu verstehen.
- **Nachteil:** Dauert ewig, wenn man viele Zahlen hat.

2. Quick-Sort (Der schnelle Teiler)

Das ist der Profi-Weg. Er nutzt den Trick "**Teile und herrsche**".

- So geht's: 1. Du pickst dir eine zufällige Zahl aus der Mitte (das Pivot).
 - 2. Jetzt machst du zwei Gruppen: Alle, die kleiner sind, kommen nach links. Alle, die größer sind, nach rechts.
 - 3. Das machst du mit den kleineren Gruppen immer wieder, bis alles perfekt sortiert ist.
- **Vorteil:** Extrem schnell, auch bei Millionen von Daten.

- **Nachteil:** Etwas komplizierter im Kopf zu behalten.
-

Kurzer Vergleich für die Klasse:

Algorithmus	Wie er arbeitet	Tempo
Bubble-Sort	Vergleicht immer nur die direkten Nachbarn.	Langsam (wie eine Schnecke)
Quick-Sort	Teilt die Menge immer wieder in zwei Hälften.	Schnell (wie ein Rennwagen)