

# Wie schreibt man ein Programm?

## Table of Contents

1. [Wie schreibt man ein Programm?](#)
  1. [Programmieren mit EVA](#)
  2. [Tipps zum Programmieren](#)

**Programmieren lernen** hat Ähnlichkeiten mit Mathematik lernen (→ Man muss viel **üben!**). Daneben gibt es auch einen kreativen Aspekt (wie bei einem Architekten; man spricht manchmal sogar von Software-Architektur!). Ein einfaches Prinzip ist . . .

## Programmieren mit EVA

- EVA steht für **Eingabe**, **Verarbeitung**, **Ausgabe**. Wir können beim Programmieren ein Problem in drei Schritten angehen:
  - **Eingabe:** Überlege, welche Eingabe das Programm oder die Prozedur benötigt! Im Beispiel der *abc-Formel* sind das die drei Variablen *a*, *b*, und *c*. An dieser Stelle machen wir uns auch Gedanken über weitere notwendige **Variablen**. Und natürlich über den sog. **Datentyp** dieser Variablen, also ob Zahl, Zeichenkette oder Boolean. Für die Eingabe gibt es in Pascal den Befehl *ReadLn*. Alternativ kann man auch mit sinnvollen Startwerten arbeiten und *Variablen initialisieren*. Hierher gehören auch "sprechende" Variablennamen.
  - **Verarbeitung:** Hier gilt es, zu denken (!). Die Kunst besteht darin, das Problem sinnvoll zu strukturieren. Frage dich beim Programmieren immer, kommt ein Befehl nach dem anderen, oder benötige ich eine Verzweigung, oder eventuell doch eine Schleife, weil etwas wiederholt wird? Hier spielen **Struktogramme** eine Rolle, da wir die **Logik** des Programmes erst mal im Kopf durchspielen. Die **Verarbeitung** besteht also im Arbeiten mit den Daten (= **Variablen**), man nennt so etwas auch einen **Algorithmus**.
  - **Ausgabe:** Für die Ausgabe benutzen wir den Befehl *WriteLn* bzw. *Write*. Gelegentlich haben wir auch die sog. *formatierte Ausgabe* verwendet.

---

## Tipps zum Programmieren

1. Schreibe wenig Code (= Zeilen) und teste dann (= compilieren).
2. Lese Fehlermeldungen des Compilers!
3. Verwende funktionierenden Code! Und erweitere ihn.
4. Verwende `WriteLn`, um dir den Wert von Variablen anzuseigen.

Kurz:

**Programmiere kleinschrittig & defensiv.**