

# Grundlagen zu Computing, Programmen und Programmiersprachen

## Soft- und Hardware

- Software umfasst alle immateriellen Teile des Computers
- Hardware wird für alle materiellen Teile des Computers verwendet
  - zur Hardware gehören folgende Komponenten
    - Grundbestandteile der Computerarchitektur
      - Hauptplatine (Motherboard)
      - Prozessor - CPU (Central Processing Unit)
      - Arbeitsspeicher – RAM (Random-Access Memory)
    - Massenspeicher / Speichermedien
    - Erweiterungskarten (Grafikkarte, Soundkarte, Netzwerkkarte)
    - Ergänzende Komponenten wie Netzteil, Gehäuse, Lüfter
    - Zusatzgeräte für
      - Ausgabe: Drucker, Bildschirm, Lautsprecher, ...
      - Eingabe: Tastatur, Maus, Joystick, Mikrofon, Scanner, ...

## Grundlagen zu Programmen

- Programm / Skript ist ein Lösungsweg zur Bearbeitung einer Aufgabe durch einen Computer
- ein Programm wird auch als Applikation oder Anwendung bezeichnet
  - die Kurzform App ist vor allem bei der Programmierung für mobile Geräte gebräuchlich
- ein Programm eine Beschreibung der Lösung einer Aufgabe in einer spezifischen Programmiersprache
- die Bearbeitungsvorschrift wird Algorithmus genannt
- ein Algorithmus führt
  - bei jeder möglichen Eingabe
  - die Verarbeitung nach einem eindeutigen Ablauf mit endlich vielen Schritten
  - mit einem reproduzierbaren Ergebnis durch
- ein Programm besteht aus Algorithmen, deren Arbeitsschritte in einer Programmiersprache (z. B. in Python, Java, C#, oder JavaScript) formuliert sind – der Quellcode (source code)
- Programme verarbeiten Daten
  - Ein Programm kann beispielsweise die Steuern berechnen
  - dem Programm werden die Daten mitgeteilt, die für die Berechnung benötigt werden
  - das Programm rechnet nach seinem Algorithmus mit den Daten und
  - liefert ein Ergebnis.
- diesen Datenfluss durch ein Programm nennt man EVA-Prinzip (Eingabe - Verarbeitung - Ausgabe).
  - Eingabe in das Programm werden Daten eingegeben, z. B. über eine Tastatur, eine Datei oder eine Datenbank.
  - Verarbeitung das Programm verarbeitet diese Daten nach einem vorgegebenen Algorithmus.
  - Ausgabe das Ergebnis des Programms wird ausgegeben, z. B. auf einen Bildschirm, einen Drucker oder eine Datei, Datenbank

# Einführung in Python

## Was ist Python?

- Python wurde in den 1990-er Jahren entwickelt
- Python ist eine Programmiersprache, die verschiedene Paradigmen unterstützt
  - strukturierte Programmierung
  - prozedurale Programmierung
  - objektorientierte Programmierung
  - funktionale Programmierung
- Strukturierte Programmierung
  - Zeichnet sich durch Kontrollstrukturen aus
    - Sequenz
      - hintereinander auszuführende Programmanweisungen
    - Selektion
      - bedingte Anweisung (IF ELSE)
    - Iteration
      - wiederholte Ausführung von Anweisungen bzw. eines Anweisungsblocks (FOR, WHILE)
- Prozedurale Programmierung
  - teilt Programme in kleinere Einheiten (Prozeduren oder Funktionen), die für bestimmte Aufgaben verantwortlich sind
  - sind die Prozeduren geschrieben und getestet, können sie immer wieder benutzt werden
  - spart Entwicklungszeit und verbessert die Lesbarkeit

- Objektorientierte Programmierung
  - bildet Objekte der realen Welt in einem Programm ab
  - im Mittelpunkt der objektorientierten Programmierung steht die Klasse
  - aus einer Klasse werden konkrete Objekte abgebildet
  - Objekte haben Eigenschaften (Attribute) und Methoden
  - Beispiel
    - Klasse Kunde
      - Name
      - Telefon
    - Hinzufügen\_Kunde
    - Ändern\_Kunde
    - Löschen\_Kunde
- funktionale Programmierung
  - alles basiert auf Funktionen
  - Berechnungen werden mithilfe von Funktionen definiert, die wiederum Funktionen aufrufen oder der Rückgabewerte nutzen
  - auch Elemente der strukturierten Programmierung wie die Iteration werden durch Funktionen ersetzt – durch so genannte rekursive Funktionen

## Skriptsprache vs. Compilersprache

- Skriptsprache (Interpretersprache)
  - eine Programmiersprache, die von einem Interpreter Zeile für Zeile übersetzt und auf der jeweiligen Plattform ausgeführt wird
- Compilersprache
  - der Quellcode einer Compilersprache wird zuerst von einem Compiler in die Maschinsprache (Assembler) der jeweiligen Plattform übersetzt
  - dieses Maschinensprachen-Programm wird in der Plattform ausgeführt
  -
- Python
  - geht einen Mittelweg zwischen Skript- und Compilersprache
  - in der interaktiven Konsole von Python wirkt die Sprache wie eine Skriptsprache
  - in den meisten Fällen wird Python in eine Art plattformunabhängigem Zwischencode übersetzt, der in Plattform interpretiert und ausgeführt (Python Virtual Machine) wird
  - der Vorteil ist ein sehr kompakter und schnell ausführbarer Zwischencode

## Python

- Python-Interpreter online nutzen
- z.B. auf der Webseite der Python Software Foundation (PSF):
  - <https://www.python.org/shell/>

```
Python 3.9.5 (default, May 27 2021, 19:45:35)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

- Python 3.9.5 (default, May 27 2021, 19:45:35)
  - aktuelle Python Version
- die Zeichen >>> sind der so genannte Eingabeprompt (Eingabeaufforderung)
- mit der Eingabe  
`print("Hallo Welt")`  
und dem abschließenden „Return“ (↵ Taste)  
wird der Befehl zur Ausgabe der Zeichenfolge *Hallo Welt* ausgeführt

```
Python 3.9.5 (default, May 27 2021, 19:45:35)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hallo Welt")
Hallo Welt
>>> █
```

- der Python-Interpreter kann auch als Taschenrechner benutzt werden:

```
Python 3.9.5 (default, May 27 2021, 19:45:35)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hallo Welt")
Hallo Welt
>>> 10+20
30
>>> █
```