

# Programmierung für Data Science und AI

Einführung in Algorithmen, Programmierung und Python

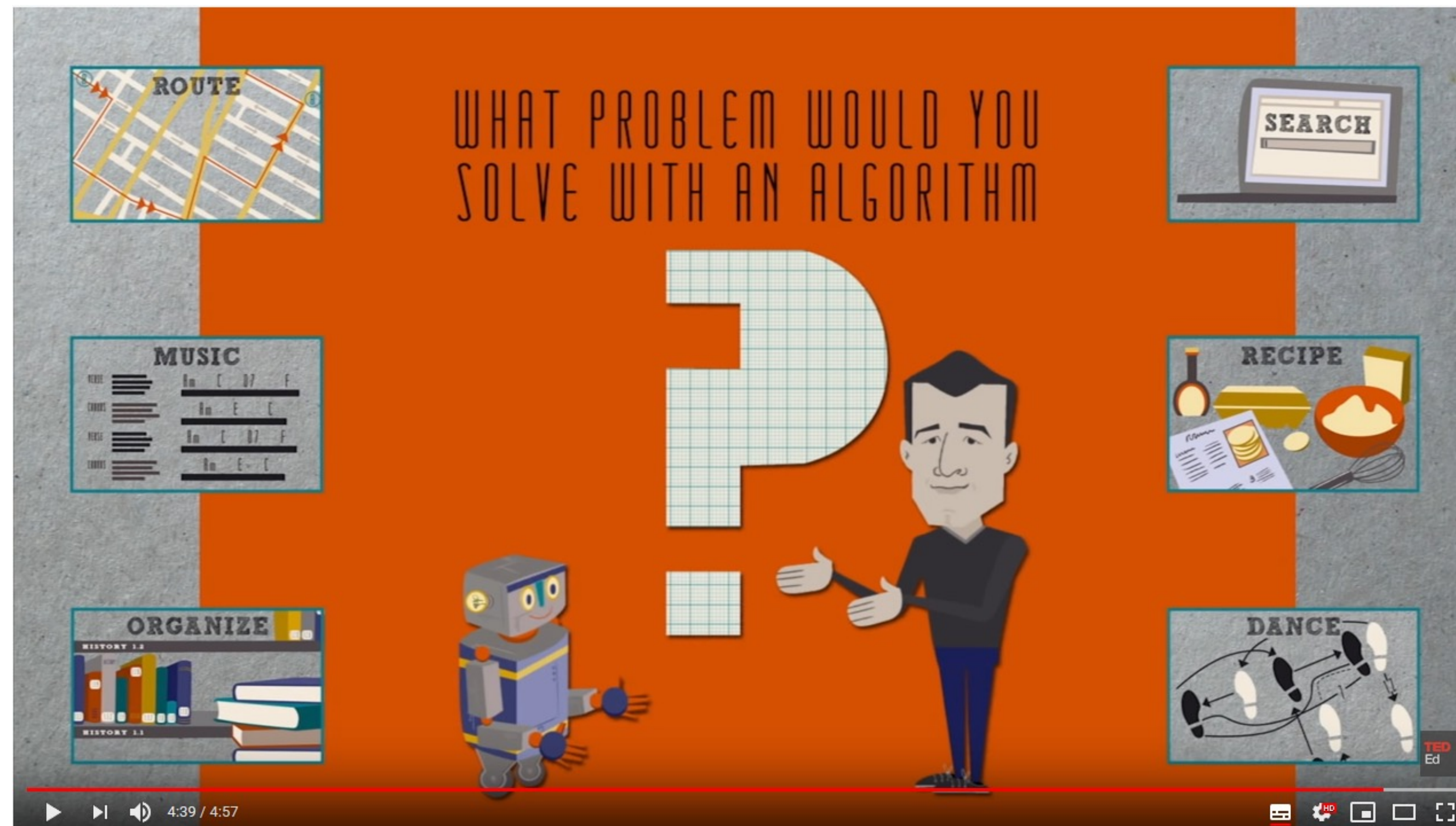
16. September 2022

# Agenda

- Algorithmen
- Programmierung
- Python und Bibliotheken
- Installation
- Ausführen von Python Snippets und Scripts
- Dokumentation und wo Sie Antworten auf Ihre Fragen erhalten

# Was ist ein Algorithmus?

<https://www.youtube.com/watch?v=6hfOvs8pY1k>



# Algorithmen

In der Mathematik und Informatik:

Ein Algorithmus ist ein Satz von Anweisungen zur Lösung einer Klasse von Problemen, oder zur Durchführung einer Berechnung.

# Eigenschaften eines Algorithmus

- Beginnt mit einem **Anfangszustand** und einer möglichen Anfangseingabe;
- geht dann durch eine **endliche Anzahl** wohldefinierter **aufeinanderfolgender Zustände**;
- produziert dabei eventuell eine **Ausgabe**;
- endet in einem endgültigen **Endzustand**.

# Weitere wichtige Eigenschaften von Algorithmen

## **Effektivität**

Ein Algorithmus ist effektiv, wenn der das **Problem löst**.

## **Effizienz**

Der Algorithmus benötigt so **wenig Rechnerressourcen** wie möglich, z. B.:

- kürzeste Rechenzeit;
- geringster Speicherverbrauch.

# Wie können Sie Algorithmen modellieren?

## **Flussdiagramme**

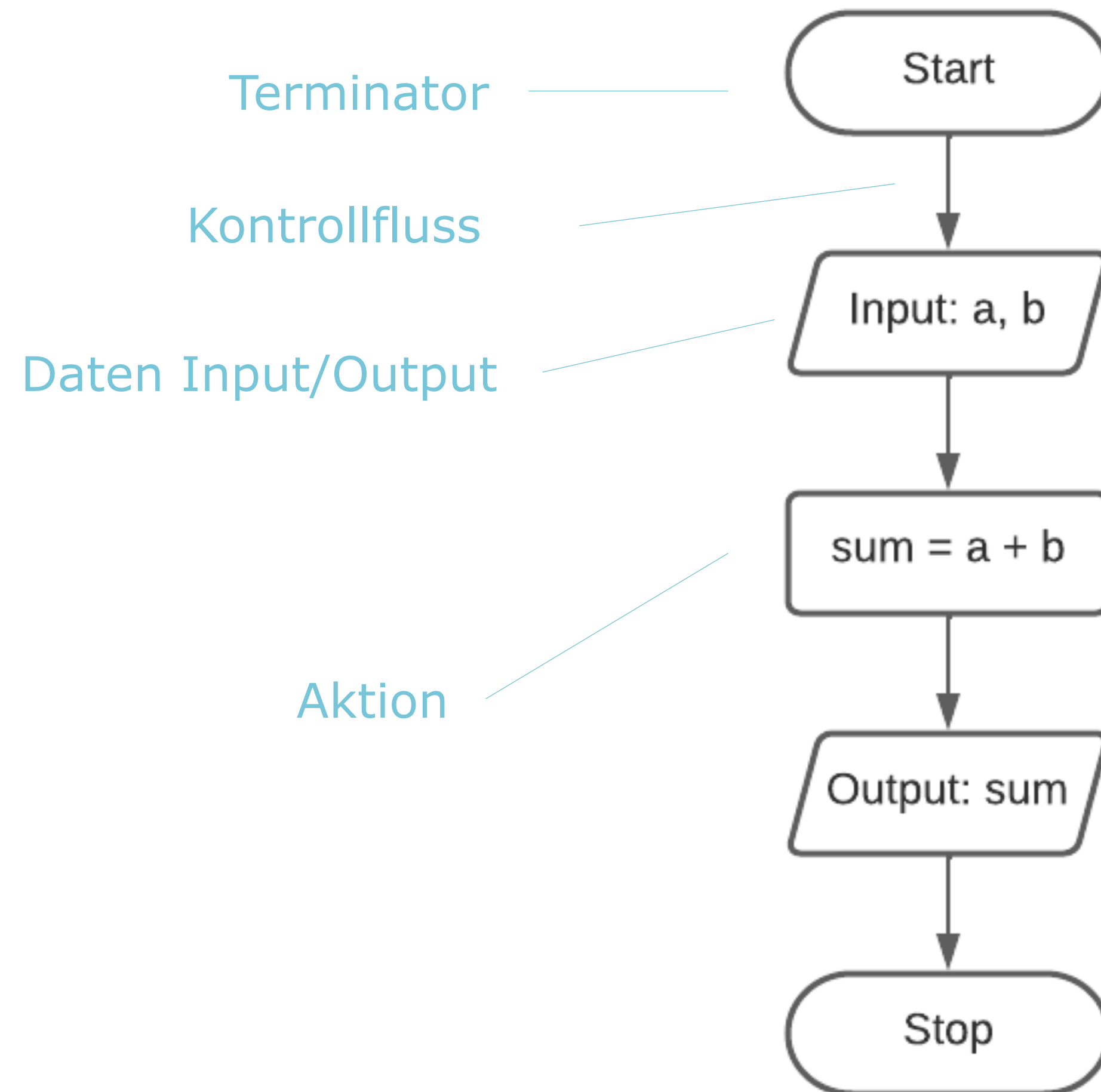
**Visuelle** Abfolge von Schritten und Entscheidungen, die zur Durchführung eines Prozesses erforderlich sind.

## **Pseudocode**

Informelle, dennoch detaillierte Beschreibung eines Algorithmus mittels **natürlicher Sprache**, allenfalls kombiniert mit mathematischer Notation

# Flussdiagramm

## Beispiel Addition



a = 5  
b = 8

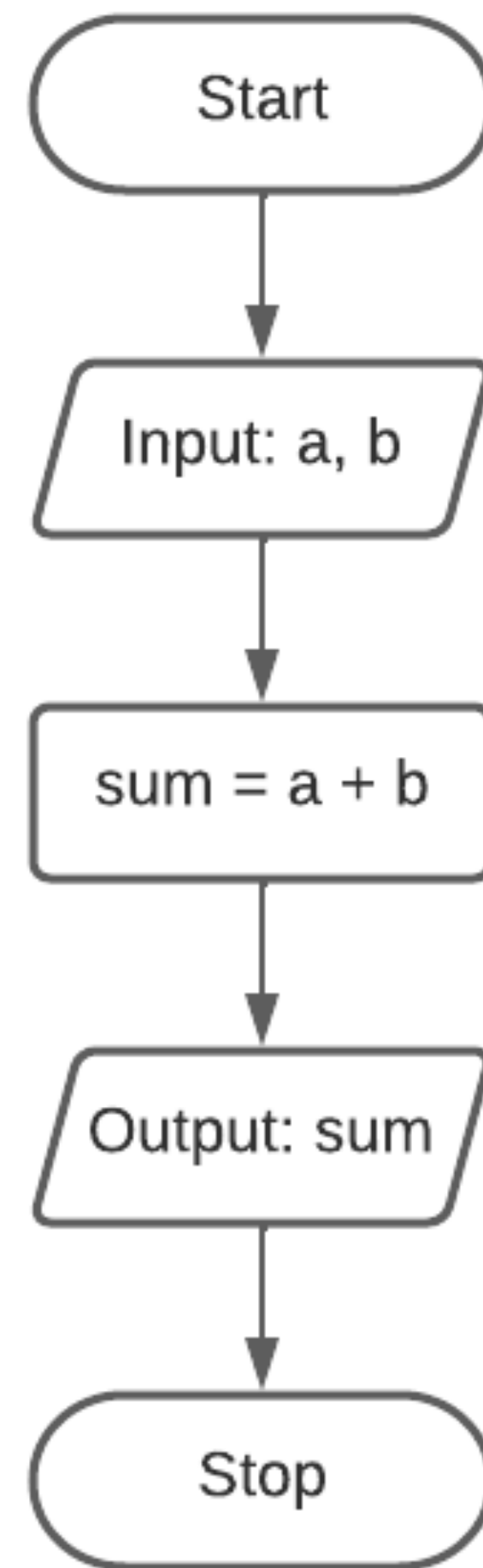
sum = 5 + 8

13



# Flussdiagramm vs. Pseudocode

## Beispiel Addition



### Pseudocode:

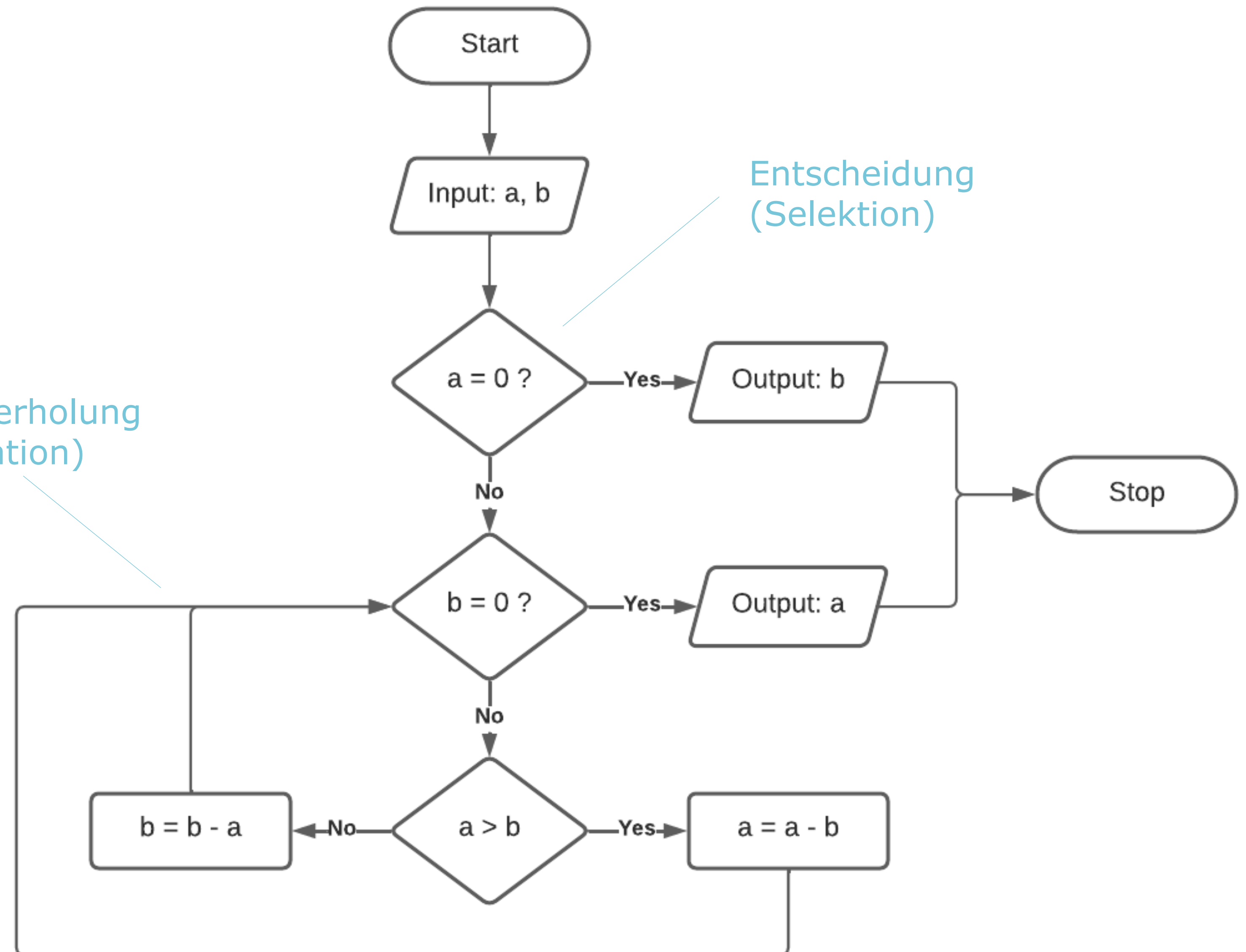
```
Einlesen Werte a und b  
Berechne sum = a + b  
Gebe das Resultat sum aus
```

# Flussdiagramm

Beispiel Grösster gemeinsamer Teiler (GGT)  
für zwei positive, ganze Zahlen

a	b	GGT
<hr/>		
<b>6</b>	<b>4</b>	
2	4	
2	2	
2	0	--> <b>2</b>

Wiederholung  
(Iteration)



# Flussdiagramm vs. Pseudocode

## Beispiel GGT – grösster gemeinsamer Teiler

Einlesen Werte a und b, beide  $\geq 0$

falls  $a = 0$ , dann:

    gebe b aus

    Stop

sonst:

    solange  $b \neq 0$ :

        falls  $a > b$ , dann:

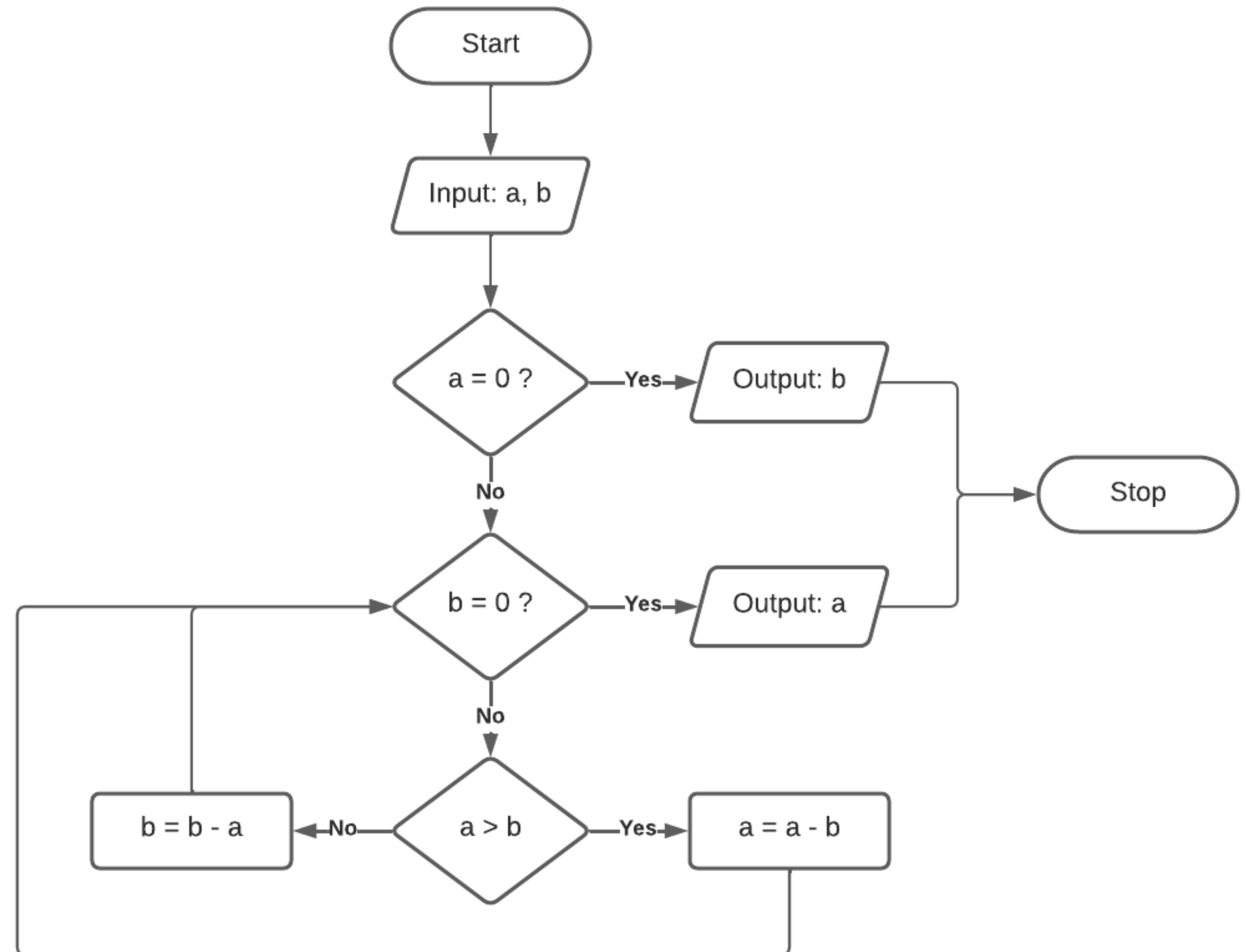
            berechne  $a = a - b$

        sonst:

            berechne  $b = b - a$

    gebe a aus

    Stop



## Ein Programm ...

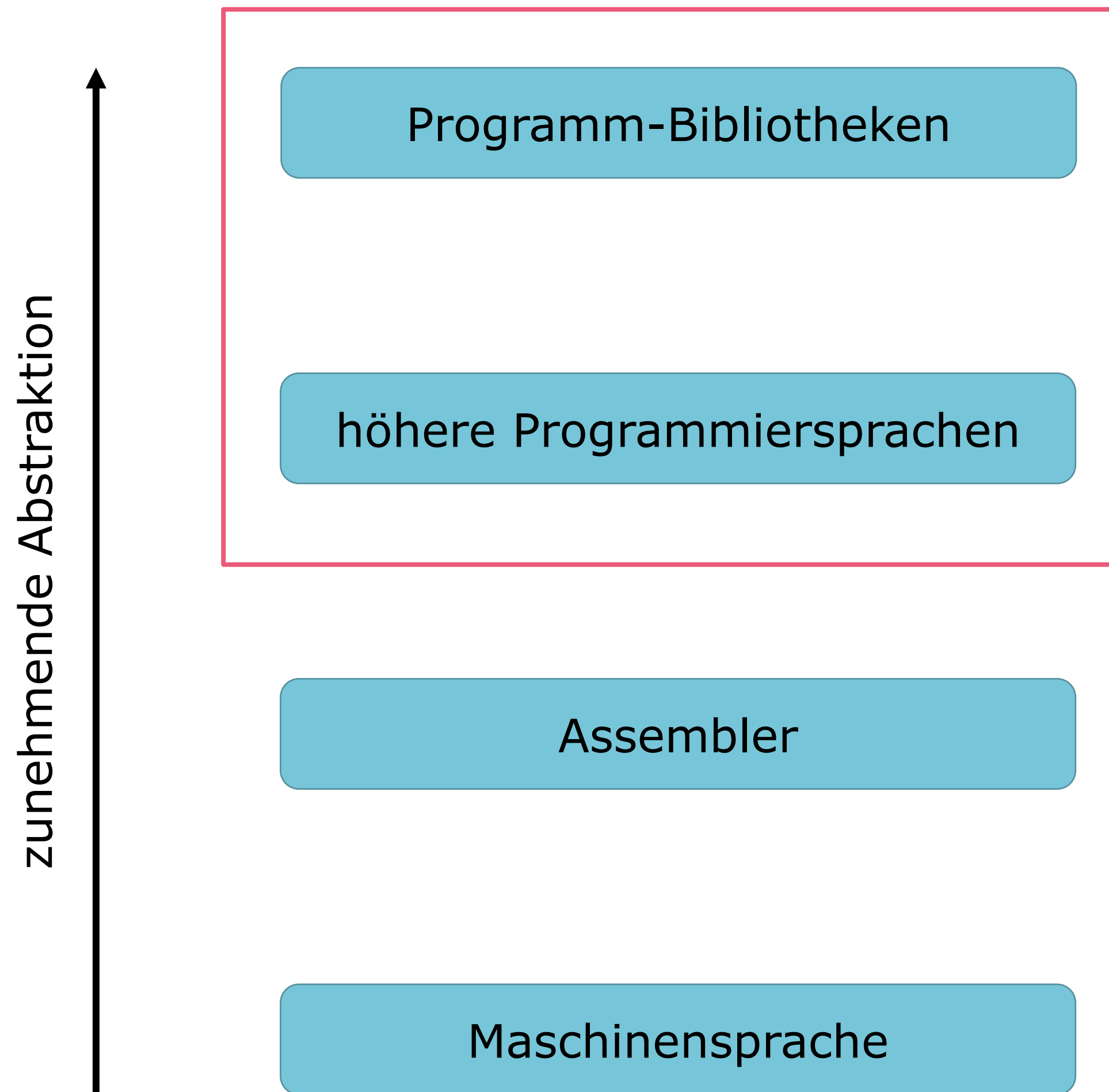
- ist eine **Folge von Befehlen** für einen Computer;
- ist eine **Reihe von Algorithmen**, die zur richtigen Zeit in der richtigen Reihenfolge ablaufen;
- hat **zu jedem Zeitpunkt** einen bestimmten **Zustand**;
- kann sein **Zustand verwenden**, um sein **Verhalten zu ändern**.

# Programmierung

Programmierung ist der **Prozess** des **Entwurfs** und der **Implementierung** eines ausführbaren Computerprogramms.

Wie gehen Sie vor?











# Spektrum der Programmiersprachen



- Vorgefertigte Programme für (häufig verwendete) Teillösungen
- Von Menschen lesbare/schreibbare Syntax
- sehr mächtig
- müssen in ausführbaren Maschinencode kompiliert werden
- Low-level-Programmiersprache
- elementare Anweisungen, die in ausführbaren Maschinencode umgewandelt werden müssen
- Steueranweisungen für Hardware
- kann direkt von einer CPU ausgeführt werden

# Top 10 Programmiersprachen, gem. TIOBE Index, September 2022

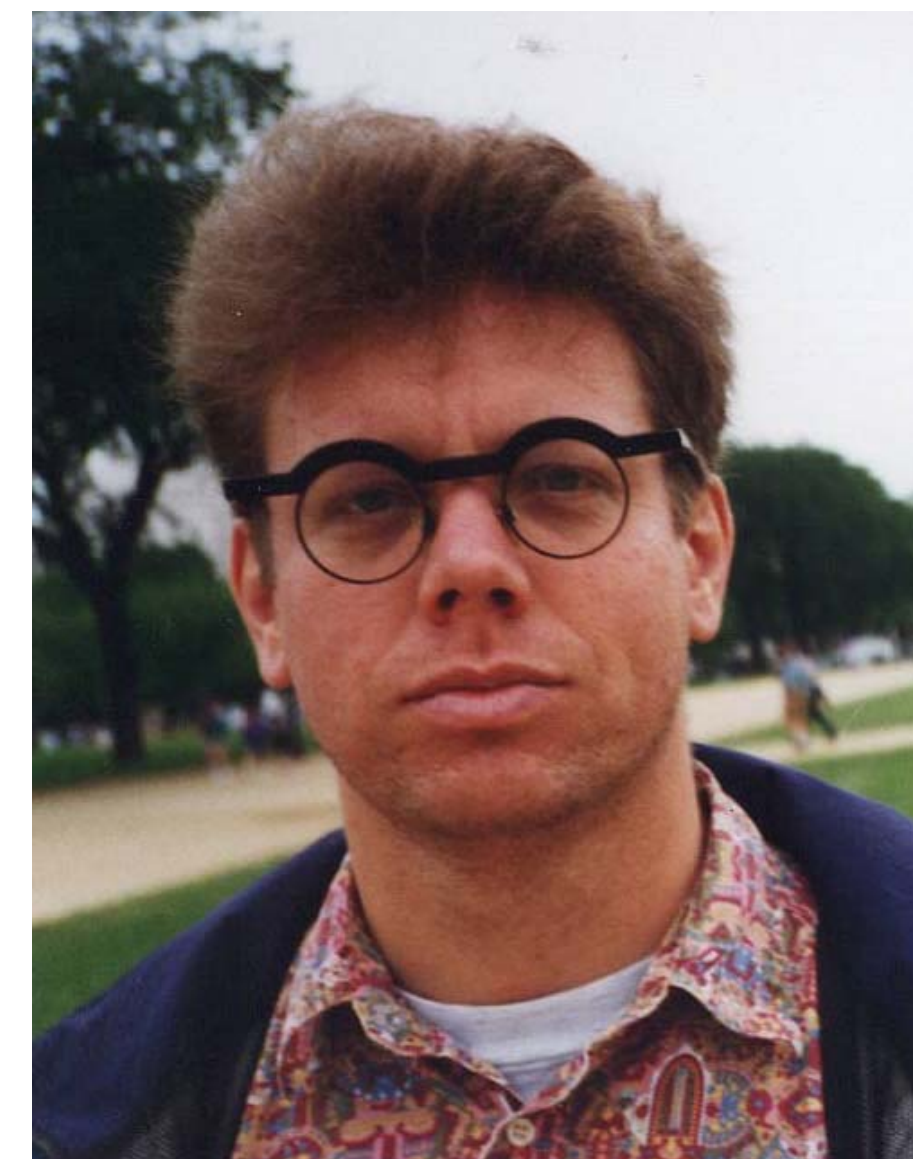
<https://www.tiobe.com/tiobe-index/>

Sep 2022	Sep 2021	Change	Programming Language		Ratings	Change
1	2	▲		Python	15.74%	+4.07%
2	1	▼		C	13.96%	+2.13%
3	3			Java	11.72%	+0.60%
4	4			C++	9.76%	+2.63%
5	5			C#	4.88%	-0.89%
6	6			Visual Basic	4.39%	-0.22%
7	7			JavaScript	2.82%	+0.27%
8	8			Assembly language	2.49%	+0.07%
9	10	▲		SQL	2.01%	+0.21%
10	9	▼		PHP	1.68%	-0.17%

# Python



- Objekt-orientierte Skriptsprache
- Umfangreiche Standardbibliothek und Open Source Bibliotheken
- 1991 öffentlich freigegeben (open source)
- Entwickelt von Guido van Rossum am National Institute for Mathematics and Computer Science, Amsterdam
- Populär für wiss. Berechnungen, Data Science und AI/ML



Guido van Rossum, 1995



# Die Python Standard Bibliothek bietet umfangreiche Funktionen

- Text und binäre Datenverarbeitung
- Mathematik
- File- und Verzeichniszugriff
- Datenspeicherung
- Datenkompression
- Kryptographie
- Systemsoftware Services
- Netzwerkprotokolle
- Nebenläufiges Programmieren
- Inter-Prozess-Kommunikation
- JSON, XML, und andere Datenformate
- Multimedia
- Internationalisierung
- GUI
- etc.

# Populäre Python Bibliotheken für AI und Data Science

## Wissenschaftliches Rechnen und Statistik

- NumPy (Numerical Python)
- SciPy (Scientific Python)
- StatsModel

## Visualisierung

- Matplotlib
- Seaborn

## Datenmanipulation und –analyse

- Pandas (Panel data)

# Populäre Python Bibliotheken für AI und Data Science

## **Machine Learning, Deep Learning, Reinforcement Learning**

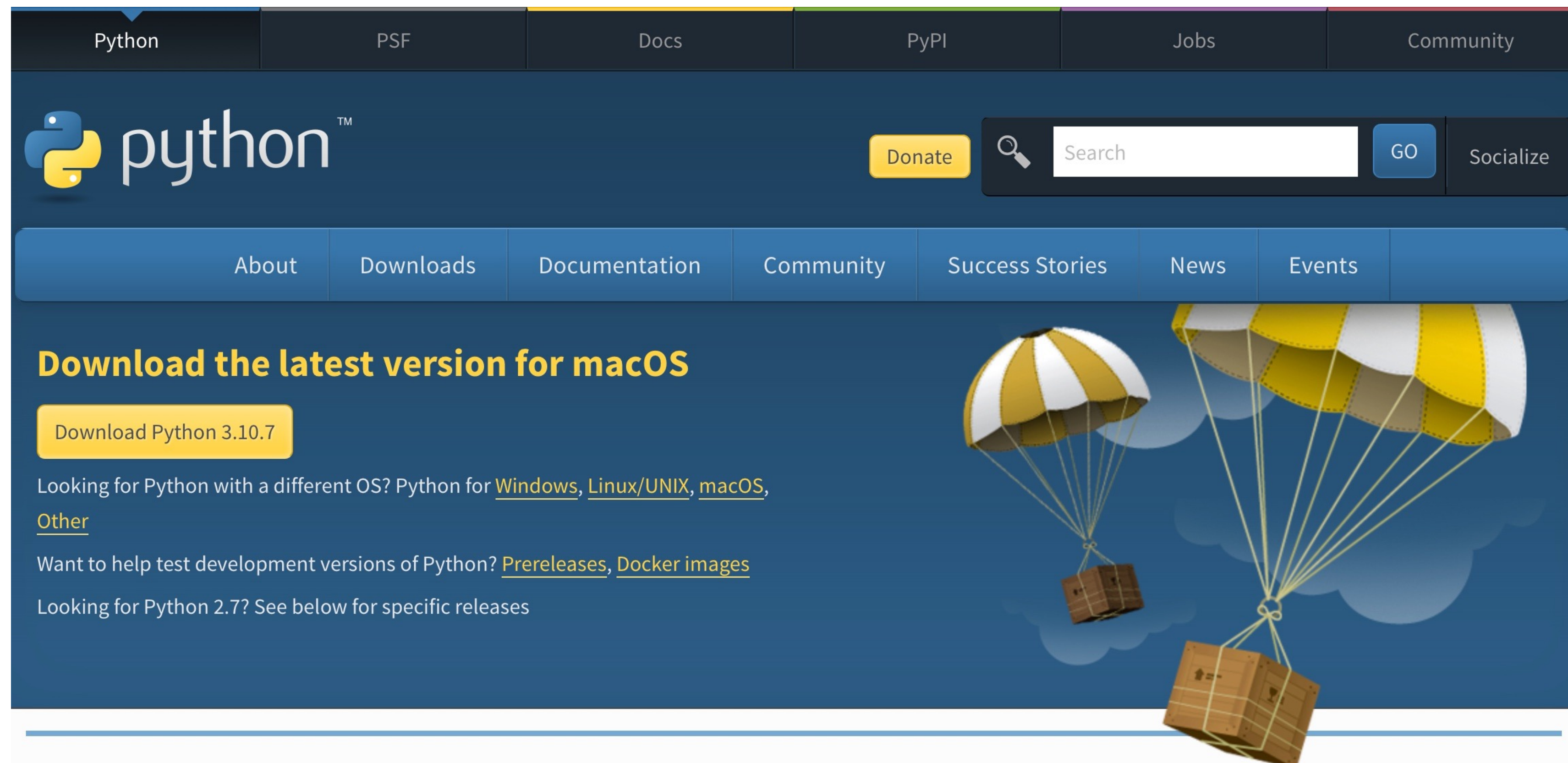
- scikit-learn
- Keras
- TensorFlow
- OpenAI Gym

## **Verarbeitung von natürlicher Sprache (NLP)**

- NLTK (Natural Language Toolkit)
- TextBlob
- Gensim

# Installation von Python

<https://www.python.org/downloads/>



# Python Package Index (PyPI)

Finden, installieren und veröffentlichen von Python-Pakete

z. B.:

- NumPy: `pip install numpy` (MacOS: `pip3 install ...`)
- Pandas: `pip install pandas`
- Matplotlib: `pip install matplotlib`
- Seaborn: `pip install seaborn`
- JupyterLab: `pip install jupyterlab`

# Ausführen von Python Programmen

Es gibt folgende Möglichkeiten:

1. Python/IPython Interaktiver Modus
2. Python/IPython Skript Modus
3. JupyterLab/Jupyter Notebook
4. Integrated Development Environment (IDE)

# IPython – Interaktiver Modus

## Ausführen von Python Snippets

- Starten von IPython: Eingabe `ipython` in der Command Line (Terminal)
- Eingabe von Snippets, Ergebnisse werden sofort ausgegeben  
→ REPL: Read-, Execute-, Print-Loop
- Verlassen von IPython: Eingabe `exit` oder `ctrl-d`

### Vorteil:

- Experimentieren mit der Sprache
- schnelles Lernen, dank unmittelbarem Feedback

### Nachteil:

- nicht persistent

# IPython Skript Modus

## Ausführen von Python Programmen (Skripts)

- Ausführen von Code, der aus einer **.py-Datei** geladen wurde

### Vorgehen:

1. Öffnen der Command Line (Terminal)
2. Wechsel in das Verzeichnis, wo das Python Script liegt, z. B.:

```
cd C:\Users\YourAccount\prog\demo
```

### 3. Ausführen des Skript

```
ipython RollDieDynamic.py 6000 1
```

Skript das ausgeführt  
werden soll

Command Line  
Argumente



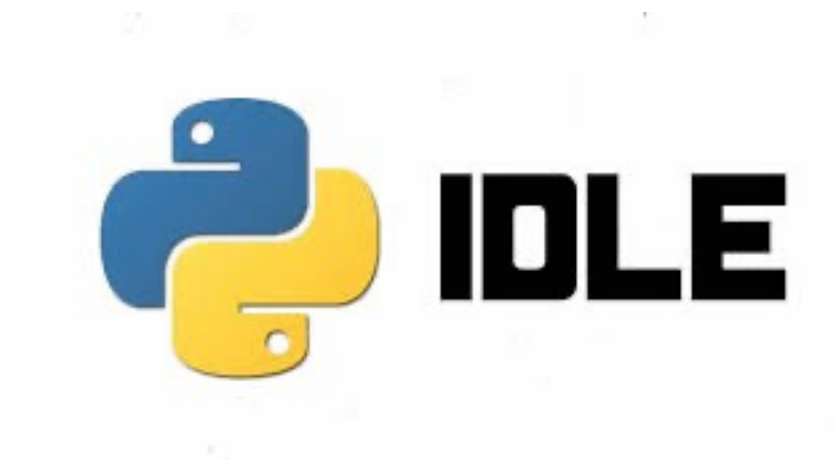
# JupyterLab / Jupyter Notebooks

- Browser basiertes Tool
- Code kann direkt in einem Dokument, das Text, Bilder, Animationen, Videos, etc. enthält, eingebettet und ausgeführt werden
- Ist besonders bei Data Science Studien vorteilhaft, da Ergebnisse gut dokumentiert und reproduzierbar weiter gegeben werden können

## Starten von JupyterLab:

- Eingabe `jupyter lab` in der Command Line (Terminal)
- Browser öffnet sich: <http://localhost:8888/lab>

# Integrated Development Environments (IDEs)



and many others ...

# Python Dokumentation

Python Standard Library

<https://docs.python.org/3/library/index.html>

Python Dokumentation

<https://docs.python.org/3/>

# Wo erhalten Sie Antworten auf Ihre Fragen?

- Tutorat
- <https://python-forum.io>
- <https://stackoverflow.com>
- Project Jupyter Google Group: <https://groups.google.com/forum/#!forum/jupyter>
- Jupyter real-time chat room: <https://gitter.im/jupyter/jupyter>

# Hausaufgabe

## Uebung01

Intallieren Sie Python und Jupyter Lab,  
sowie die zusätzlichen Bibilotheken NumPy, Pandas, Matplotlib und Seaborn