



# EINFÜHRUNG IN PROGRAMMIERUNG UND DATENBANKEN

JOERN PLOENNIGS

## GRUNDLAGEN

Motivation

Computer und  
Architekturen

Programmierung  
und Datentypen

Verzweigungen und  
Schleifen

## MODELLIERUNG

Fehler und  
Debugging

Objektorientierung u.  
Softwareentwurf

Funktionen und  
Rekursion

## PROGRAMMIERSPRACHEN

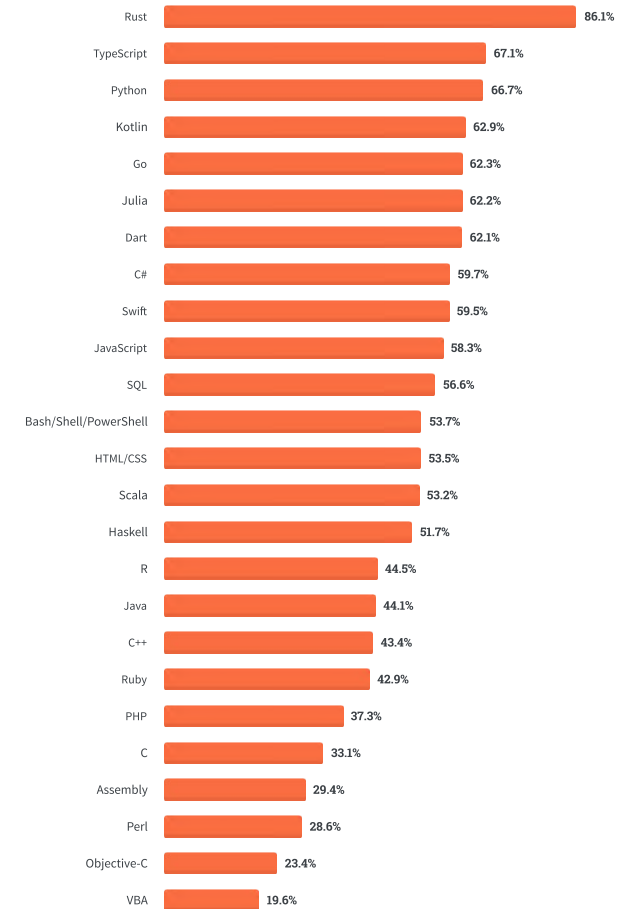


DALL-E 2: An Egyptian using a computer

# PROGRAMMIERSPRACHEN

Definition: Eine Programmiersprache ist eine formale Sprache für die Erstellung von Programmen zur Datenverarbeitung an einem Computer. Sie ist durch ihren **Zeichensatz**, **Syntax** und ihre **Semantik** definiert.

Die Programmiersprache erlaubt es dem Menschen (Programmierer), dem Computer in einer „verständlichen“ Form ein Programm zu erstellen.



Beliebteste Programmiersprache auf Stack Overflow

# GENERATIONEN VON PROGRAMMIERSPRACHEN

1. **Generation - Maschinenorientierte Sprachen** sind die Programmiersprachen, die Prozessoren direkt *ausführen* können und meist nur für den Prozessortyp verständlich sind. Der Programmcode wird Maschinencode genannt und ist *binär* (0/1) codiert.
2. **Generation – Assemblersprachen** nutzen Abkürzungen (ADD, MOV, ...) für Maschinenbefehle eines bestimmten Prozessortyps. Der Quelltext eines Assemblerprogramms wird mit Hilfe einer Übersetzungssoftware (Assembler) in den Maschinencode übersetzt.
3. **Generation - Höhere Programmiersprachen:** orientieren sich am Menschen, dem Programmierer. Ein sogenannter Compiler übersetzt den gegebenen abstrakteren Befehlssatz in den Maschinencode der gegebenen Zielarchitektur oder in eine Zwischensprache.

```
1101 0000 0000 0111 1011
1011 1111 1110 1000 1010
1101 0010 0000 0111 1111
```

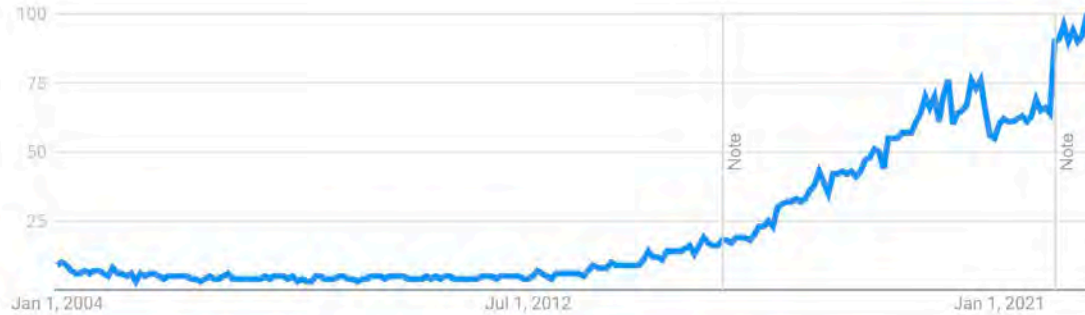
```
CLO
MOV AL, 2    ;kopiert 2 in Register AL
ADD AL, 3    ;addiert 3 zu Register AL
END
```

```
while i < 20:
    x = x + i * i
    if x > 100:
        i = i + 3
```





Interest over time



Python wird in Data Science sehr oft verwendet. Sie hat sehr viele Bibliotheken zur Analyse von Daten.

Heutzutage gibt es in Python Bibliotheken für fast jedes Anwendungsgebiet.

Google Trend `data science` worldwide

# PYTHON

Entworfen: 1991

Auf Lesbarkeit ausgelegt

The [Zen of Python](#) (PEP 20):

- Beautiful is better than ugly
- Explicit is better than implicit
- Simple is better than complex
- Complex is better than complicated
- Readability counts



Midjourney: A python programming a computer



# SYNTAX EINER PROGRAMMIERSPRACHE

Formaler Regelsatz, der definiert, wie der Computer die Programmiersprache interpretieren soll. Der Syntax umfasst meist Regeln zur Definition von Befehlen (Wörter), Variablen (Werte), Code-Blöcken (Sätze).

## Beispiel: Python Syntax Elemente

Syntax Beispiel	Bedeutung
1234	Zahlen Wert
True / False	Booleschen Werte
"Text"	Zeichenkette
# Ignoriere	Kommentar
+ - / *	Operatoren
<pre>if True:     print("Hello")     break</pre>	<p>Einrückung eines Code-Blocks in Python. Achtung! Alle Zeilen eines Code-Blocks müssen gleich eingerückt sein.</p> <p>Eine "if" Bedingung wird immer mit ":" abgeschlossen</p>

Definition der Bedeutung der Sprachelemente und der zulässigen Kombinationen von Elementen.

## Beispiel: Python Semantische Elemente

Semantik Beispiel	Bedeutung
<code>print("Hello")</code>	Gibt die Zeichenkette "Hello" aus
<code>exit()</code>	Beendet ein Programm
<code>pass</code>	Tut gar nix
<code>break</code>	Bricht einen Code-Block ab
<code>while</code>	Wiederhole den folgenden Code-Block

HÖRSAMFRAGE

FRAGEN?



DALL-E 2: A psychedelic DJ with a question mark for a head