



Grundlagen der Programmierung

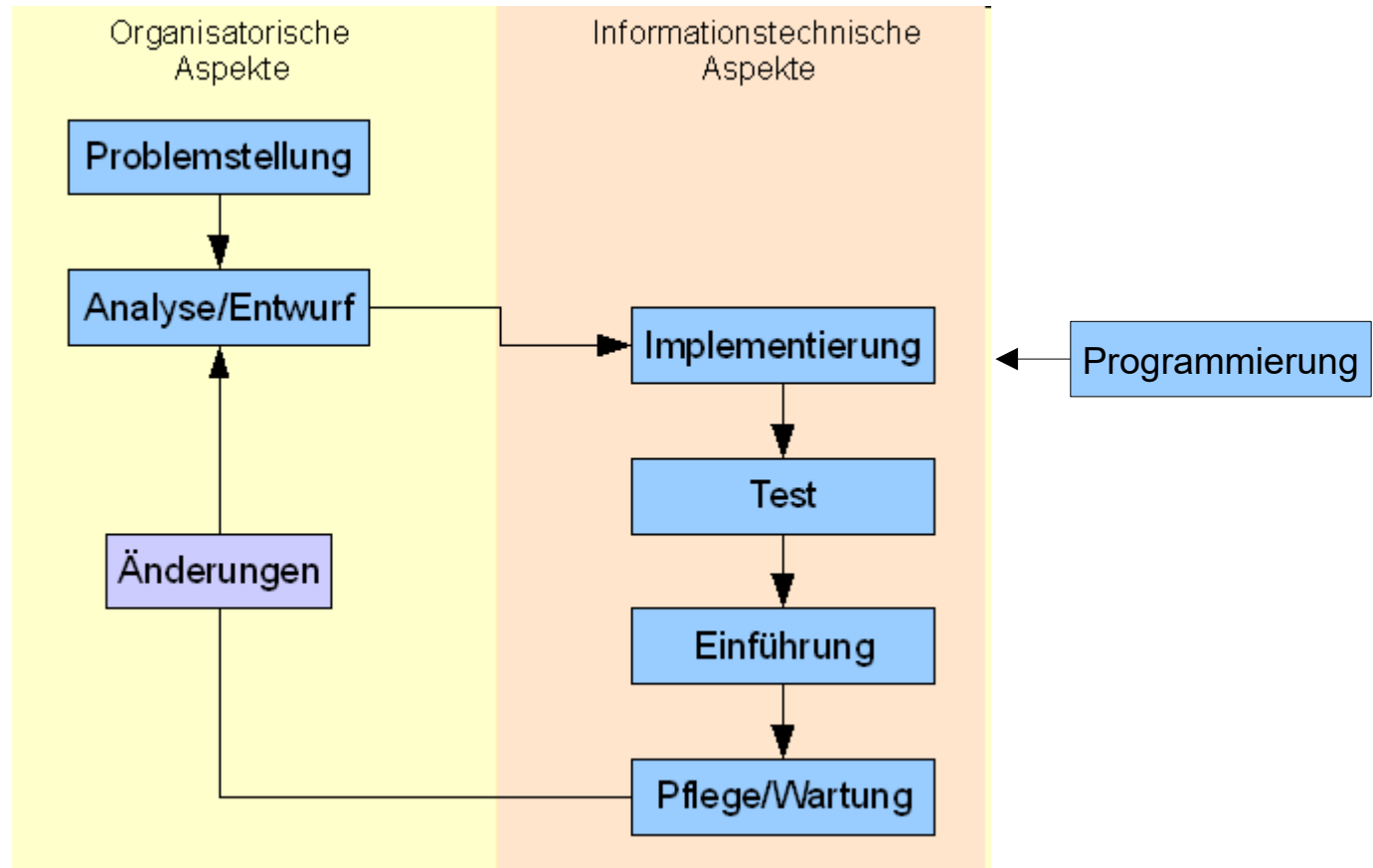
Stephan Karrer

Aphorismus

Ein Rechner ist ein Vollidiot mit Spezialbegabung. Er hat ein großes, präzises Gedächtnis und kann schneller rechnen als ein Mensch.

— Prof. Dr. Gerhard Goos (1962)

Programmierung ist lediglich ein Teil des Software-Lebenszyklus



Quelle: <https://wikipedia.de>

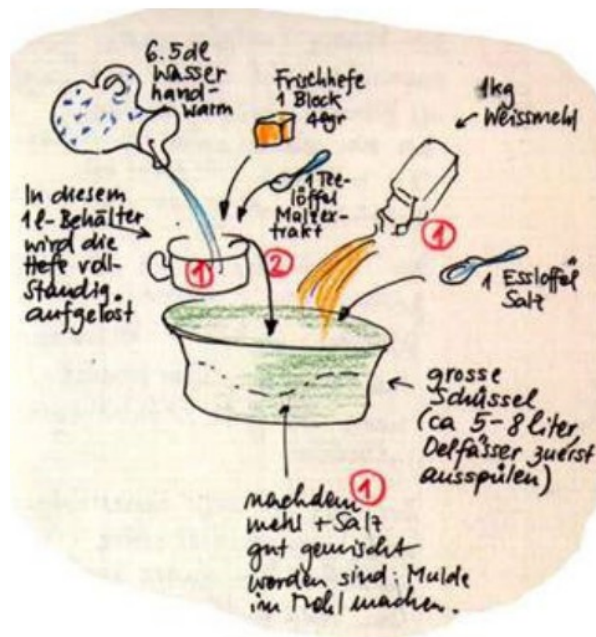
Zentraler Begriff: Algorithmus

- Abgeleitet von Muhammad ibn Musa al-Chwarizmi (lat. al-Gorithmus), persischer Mathematiker 9. Jh. n. Chr.
- Algorithmus ist eine Handlungsvorschrift, um ausgehend von bestimmten Ausgangszuständen ein bestimmtes Ziel (Ergebnis) zu erreichen
- Typische Elemente, die in Algorithmen enthalten sind:
 - Geordnete Folge von Arbeitsschritten
 - Fallunterscheidungen und Wiederholungen (Schleifen)
 - Eingaben: Parameter zur Beschreibung der Ausgangssituation, z.B. 2 Zahlen, deren Summe berechnet werden soll
 - Ausgaben: Ergebnis, z.B. Summe der Zahlen

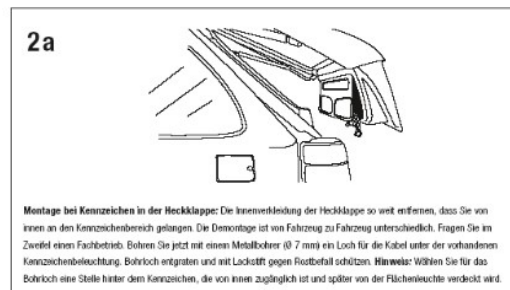
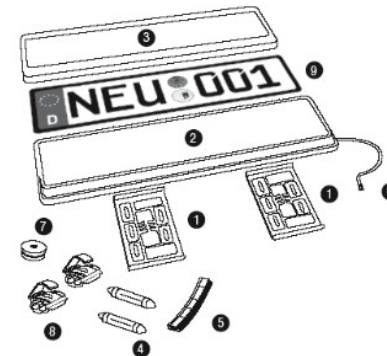


Beispiele aus dem Alltag

Kochrezept:



Montageanleitung:



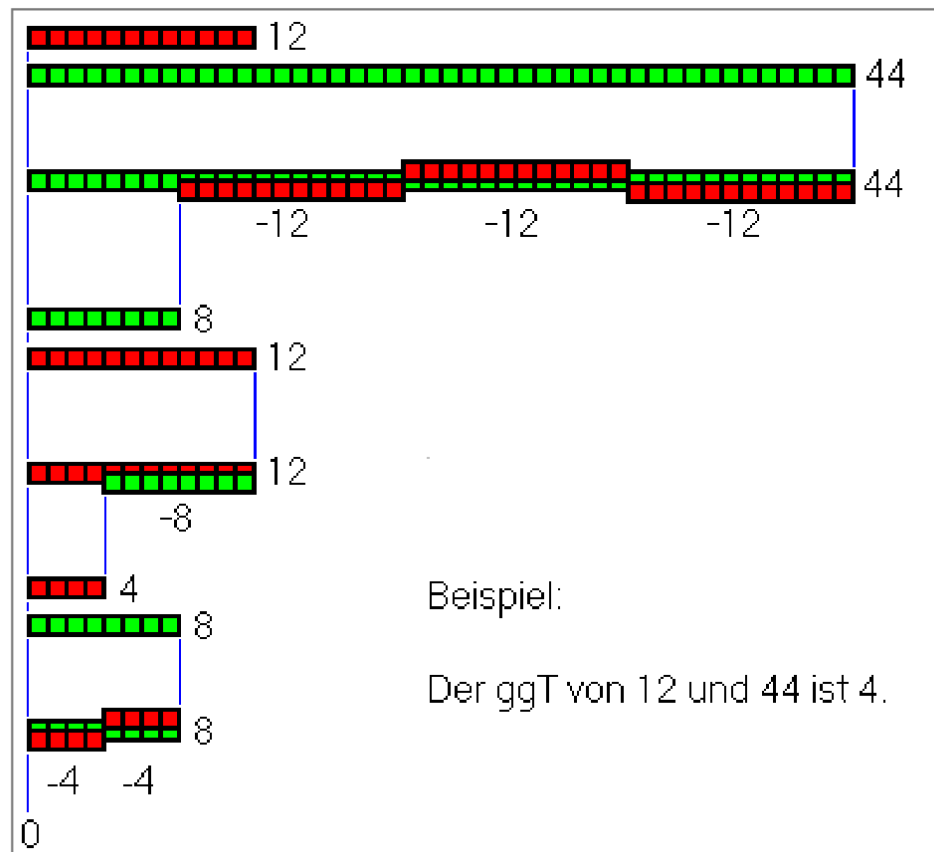
GGT - Algorithmus

- Algorithmus zur Berechnung des größten gemeinsamen Teilers (GGT) von zwei natürlichen Zahlen a und b (Euklid von Alexandria ca. 300 v.Chr.).

Eingabe: $a, b \in \mathbb{N}$

```
1  solange  $b \neq 0$   
2    wenn  $a > b$   
3      dann  $a = a - b$   
4    sonst  $b = b - a$ 
```

Ausgabe: a

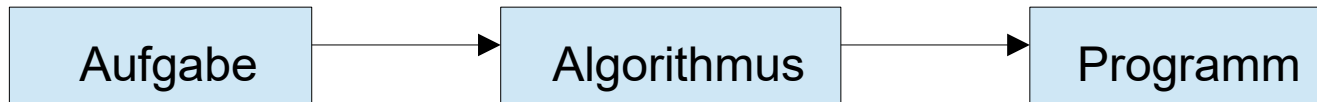


Algorithmus

- In der Software-Entwicklung üblicherweise als Vorschrift zur Durchführung einer Berechnung (Folge von einzelnen Schritten) mit folgenden Eigenschaften:
 - Präzision Die Bedeutung jedes Schritts ist eindeutig festgelegt.
 - Effektivität Jeder Schritt ist ausführbar.
 - Finitheit (statisch) Die Vorschrift ist ein endlicher Text.
 - Finitheit (dynamisch) Zur Ausführung wird nur endlich viel Speicher benötigt.
 - Terminierung Die Berechnung endet nach endlich vielen Schritten

- Weitere Eigenschaften, die oft als wünschenswert für einen Algorithmus angesehen werden:
 - Determinismus Die Folgeschritte sind immer eindeutig festgelegt.
 - Determiniertheit Bei gleicher Eingabe erzeugt die Vorschrift die gleiche Ausgabe – berechnet also eine Funktion.
 - Generalität Die Vorschrift kann eine Klasse von Problemen lösen.

Programm



- Formulierung eines Algorithmus (oder) mehrerer in einer Programmiersprache
- Es gibt verschiedene Programmiersprachen, aber sie alle sind formale Sprachen, d.h., sie sind exakt, durch strikte Regeln, definiert. Das unterscheidet sie von natürlichen Sprachen wie Deutsch oder Englisch.

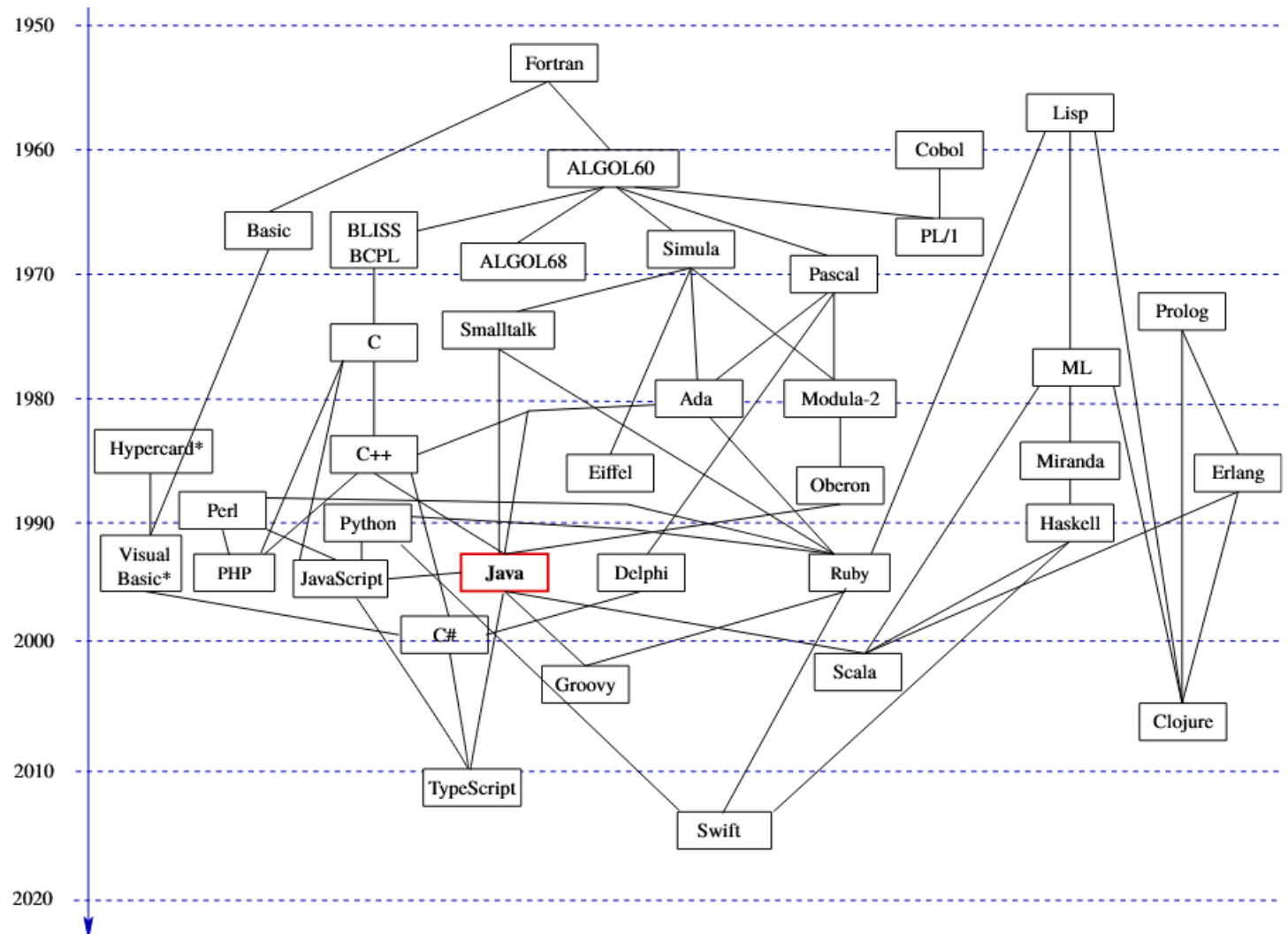
```
numbers = range(1 , 10)

for i in numbers:
    print("processing number:", i)
    if i % 2 == 0:
        print("even", i)
    else:
        print("odd", i)

print(string1, string2, "finished")
```

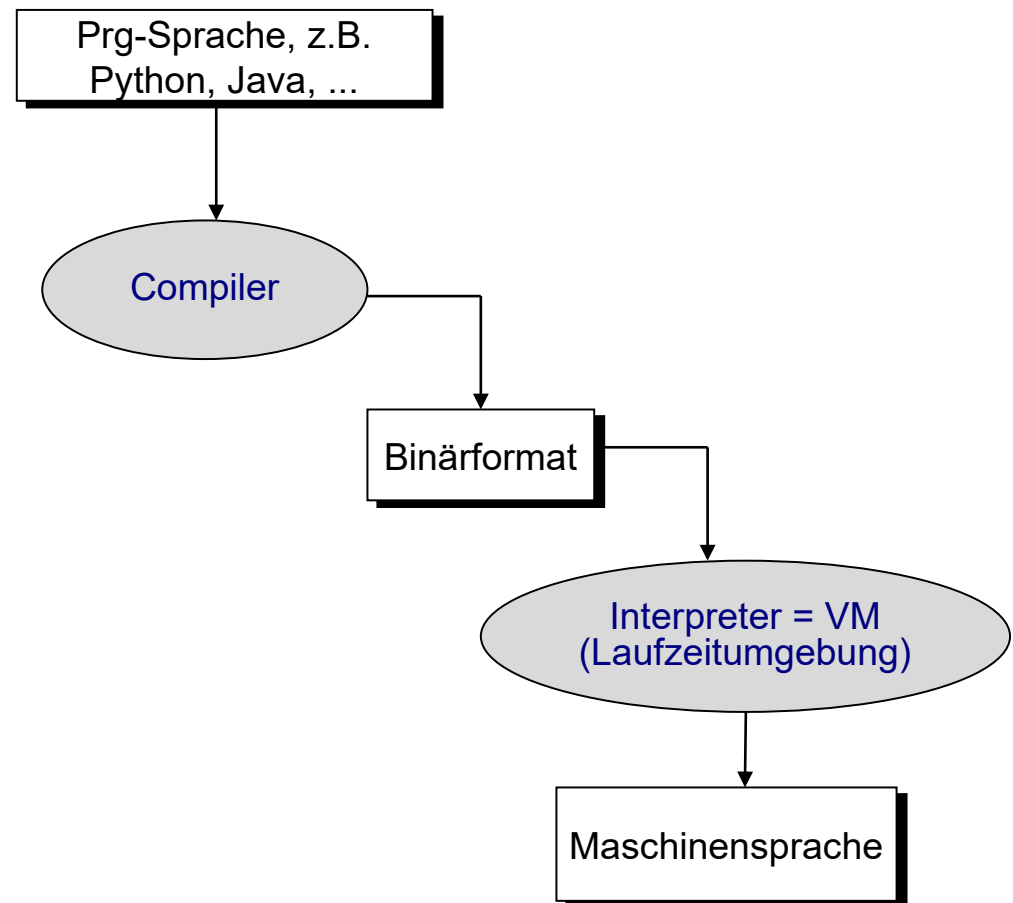
Programm-Snippet in Python

Programmiersprachen



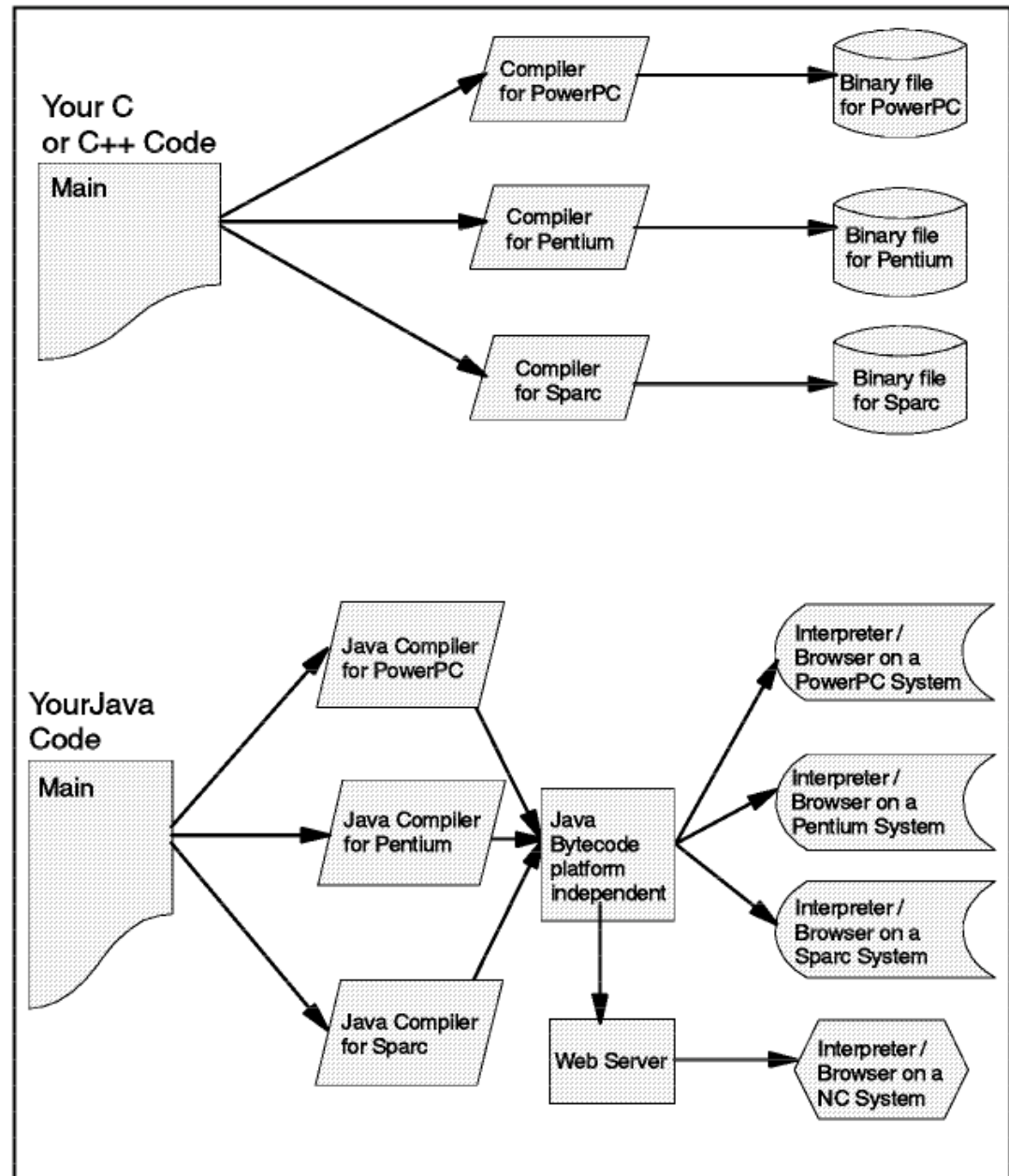
Trennung von Compiler und Interpreter

- Aktuell häufiger Ansatz bzgl. Umsetzung auf Binärformat des Rechners
- Plattform unabhängig, sofern der Interpreter auf mehreren Plattformen (Betriebssystemen) verfügbar ist.
- Das Zielsystem mit dem Interpreter ist oft nicht identisch mit dem Entwicklungssystem.
 - ▶ Entscheidend ist somit die Sprach-Version des Zielsystems !



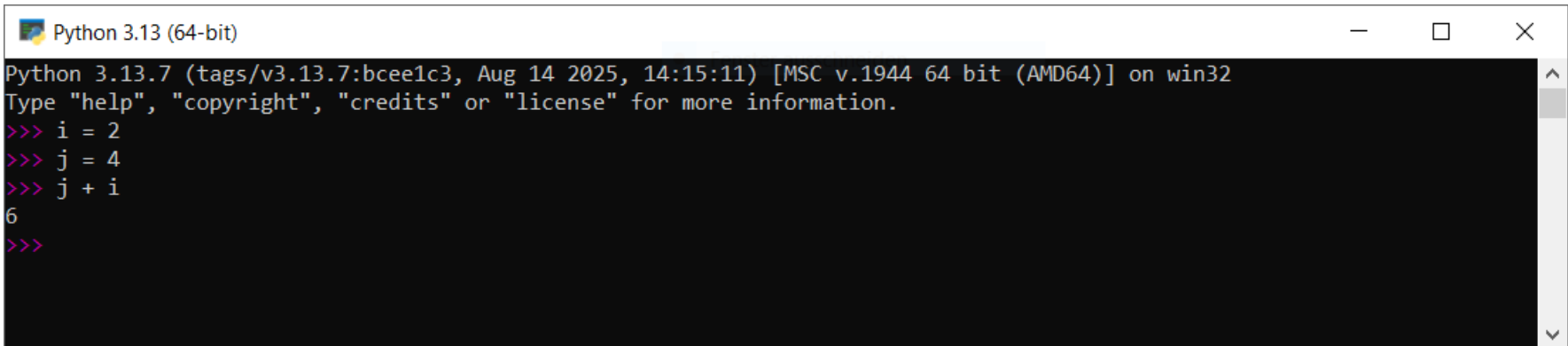
Vergleich:

C- versus
Java-Compiler



REPL-Shell

- REPL (Read - Evaluate - Print - Loop) : Wir tippen unseren Code direkt in das Tool (Kommando-Interpreter), der wird interpretiert und das Ergebnis ausgegeben.
- Beispiel: Python-Shell

A screenshot of a Python 3.13 (64-bit) REPL window. The window title is "Python 3.13 (64-bit)". The terminal text shows the Python version and build information: "Python 3.13.7 (tags/v3.13.7:bcee1c3, Aug 14 2025, 14:15:11) [MSC v.1944 64 bit (AMD64)] on win32". It then prompts the user to type "help", "copyright", "credits" or "license" for more information. The user enters three lines of code: ">>> i = 2", ">>> j = 4", and ">>> j + i". The output of the last line is "6". The prompt ">>>" is shown again on the next line.

```
Python 3.13 (64-bit)
Python 3.13.7 (tags/v3.13.7:bcee1c3, Aug 14 2025, 14:15:11) [MSC v.1944 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> i = 2
>>> j = 4
>>> j + i
6
>>>
```

Editoren und IDEs

- Selbst einfache Editoren bieten mittels Plugins elementare Unterstützung für Syntax-Hervorhebungen etc. an.
 - Notepad++
 - UltraEdit
 - ...

- IDEs (Integrated Development Environment) integrieren zusätzlich:
 - Code Assist
 - Hilfefunktion und Dokumentation
 - Debugger
 - Testwerkzeuge
 - Code-Formatting
 - Anbindung an Versionsverwaltung

Beispiel: Python-IDE PyCharm (JetBrains)

