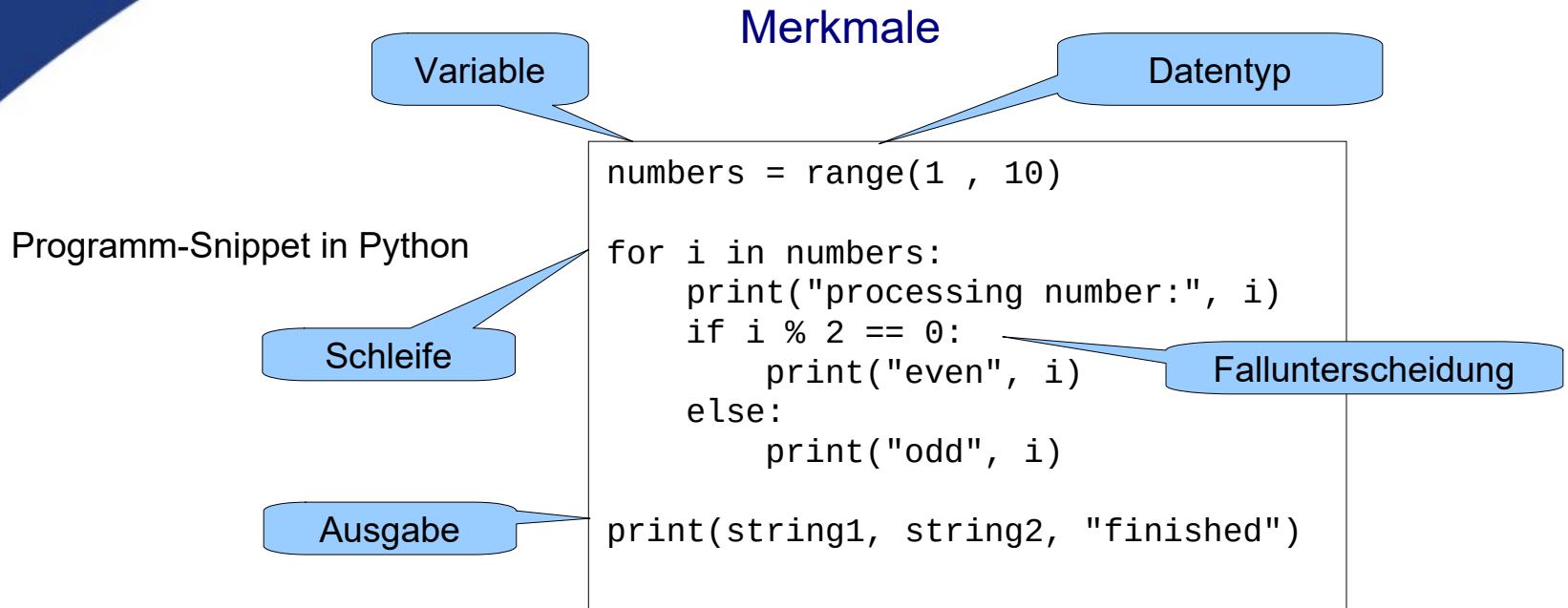


Python - Einführung

Stephan Karrer



- Dynamische Typisierung
 - Nachteil: mehr Typ-Fehler zur Laufzeit
 - Vorteil: weniger Schreibarbeit, Flexibilität
- Objekt-Orientierung (ist erstmal kaum sichtbar)
- Funktionale Sprachelemente
- Modular

Python: Historie und Versionen

- Februar 1991: Guido van Rossum veröffentlichte die erste öffentliche Version von Python: die Version 0.9.0
- 1994: Python 1.0, die erste Version, die als stabil gilt.
- 2000: Python 2.0
- 2008: Python 3.0
 - ▶ In wesentlichen Teilen nicht kompatibel mit Version 2

- Die Weiterentwicklung von Python liegt heute in den Händen der
Python Software Foundation (PSF):

The mission of the Python Software Foundation is to promote, protect, and advance the Python programming language, and to support and facilitate the growth of a diverse and international community of Python programmers.

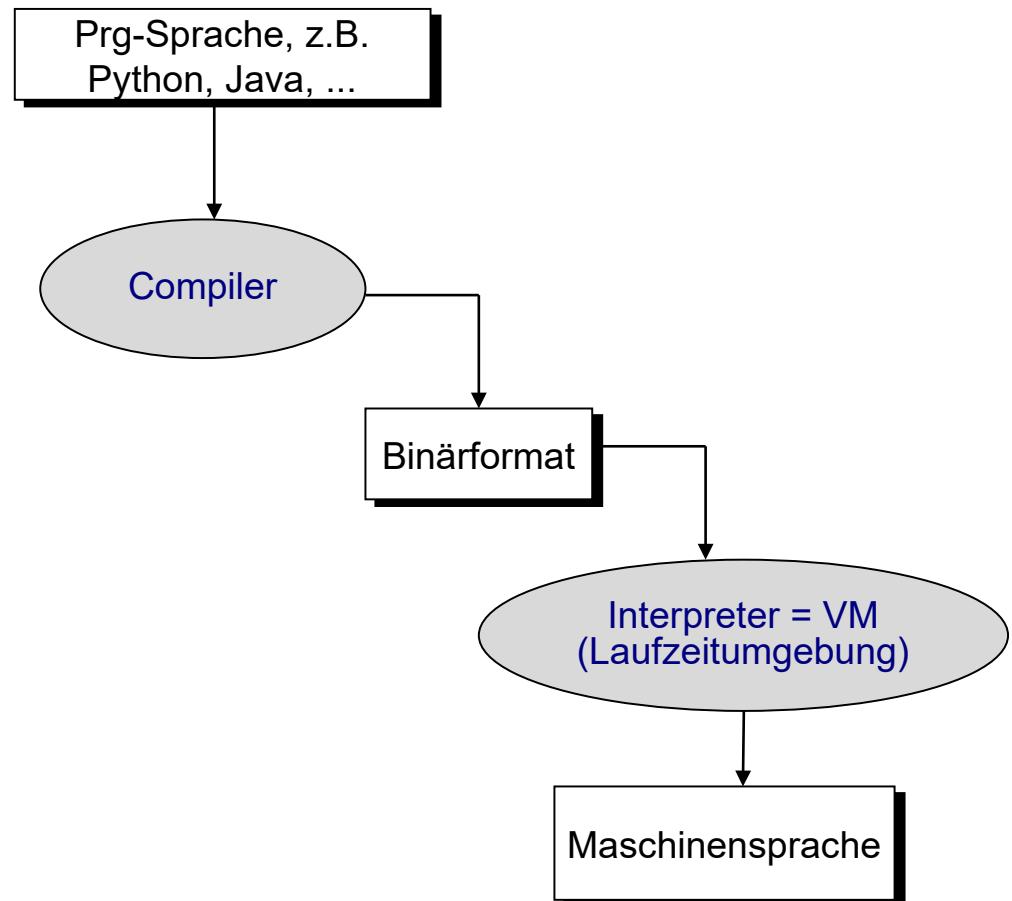
<https://www.python.org/psf/mission/>

Wesentliche Einsatzgebiete

- Web Development
 - Unterstützung durch Frameworks wie Django, Flask, FastAPI, ...
- Datenanalyse und Mathematik
 - Unterstützung durch Bibliotheken wie Pandas, NumPy, ...
- Künstliche Intelligenz (AI) und maschinelles Lernen
 - Unterstützung durch Bibliotheken wie TensorFlow, PyTorch, ...
- Automatisierung und Skript-Programmierung
 - selbstverständlich auch Online verfügbar

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 !

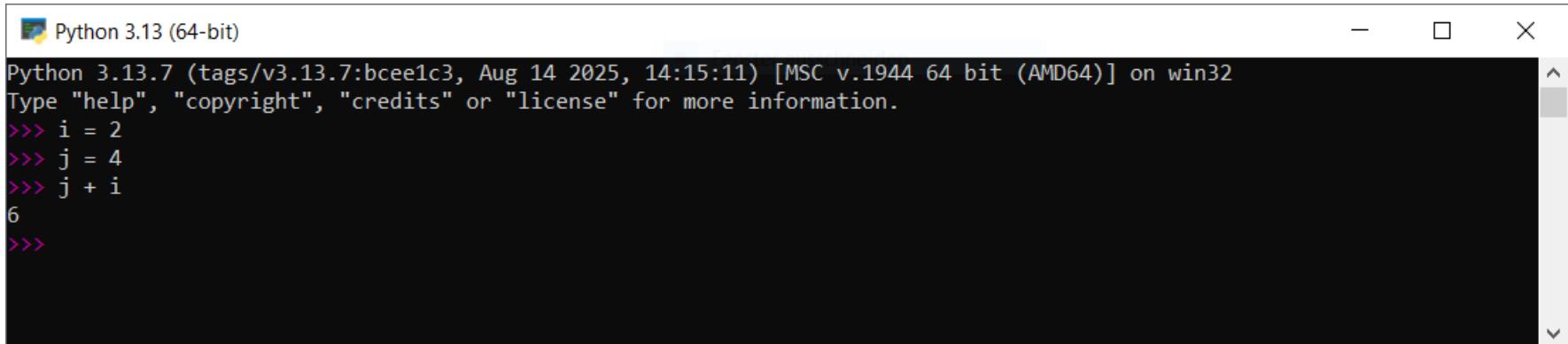


Verfügbarkeit

- Python ist frei erhältlich und Open Source
- Distributionen sind für alle relevanten Betriebssysteme verfügbar:
 - Linux
 - Windows
 - Mac
 - ...
- Distribution enthält auch die Dokumentation
 - selbstverständlich auch Online verfügbar
- Distribution enthält auch ein einfaches Werkzeug zum Ausführen des Codes, die Python-Shell (Read - Evaluate - Print – Loop = REPL)
- Viele Online-Tutorials bieten eine entsprechende Ausführungsumgebung an (z.B. <https://www.w3schools.com/python/>)

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



The screenshot shows a Windows-style terminal window titled "Python 3.13 (64-bit)". The window contains the following text:

```
Python 3.13.7 (tags/v3.13.7:bceee1c3, 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)

The screenshot shows the PyCharm IDE interface. The top navigation bar includes 'PythonProject', 'Version control', and a 'Trial' button. The left sidebar displays the 'Project' structure, showing a 'PythonProject' folder containing a '.venv' library root and a 'Sample.py' file. The main area shows the code editor with 'Sample.py' open, containing the following Python script:

```
# Sample
string1 = "Stephan"
string2 = 'Karren'

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")
```

The bottom panel shows the 'Run' tab with the 'Sample' configuration selected. The run history lists the output of the script, showing it processes odd and even numbers from 1 to 7.

Output
processing number: 1 odd 1
processing number: 2 even 2
processing number: 3 odd 3
processing number: 4 even 4
processing number: 5 odd 5
processing number: 6 even 6
processing number: 7

At the bottom, the status bar shows '4:17 (8 chars) CRLF UTF-8 4 spaces Python 3.13 (PythonProject)' and a file count of '00001-09'.