

Python

Höhere interpretierte Programmiersprache

Definition

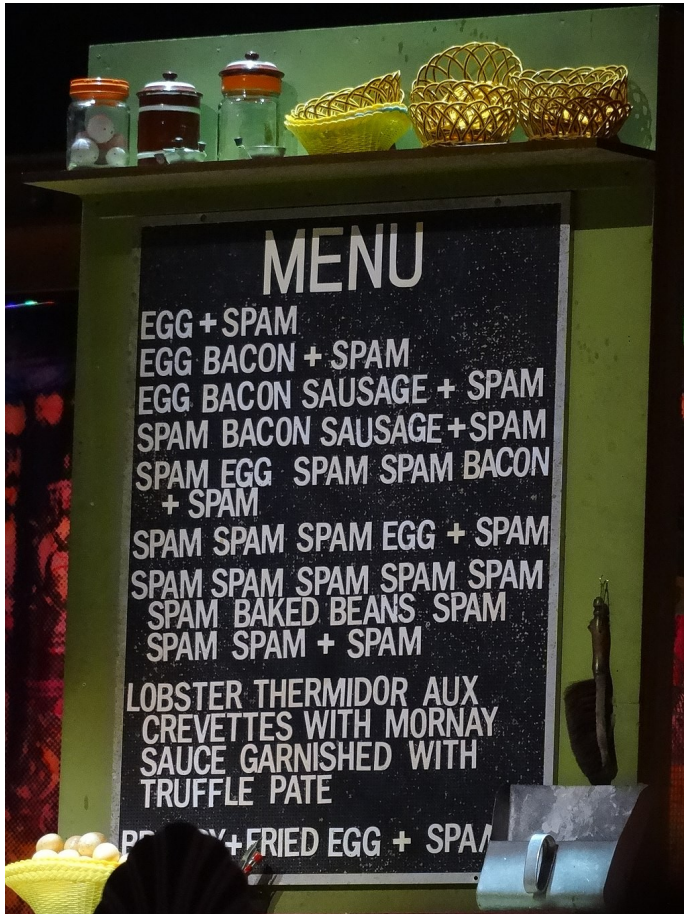
Python ist eine objektorientierte, üblicherweise interpretierte Programmiersprache, die wegen der klaren Syntax und der guten Lesbarkeit enorm an Popularität gewonnen hat.

Python ist eine dynamisch typisierte und (meistens) interpretierte Sprache.

Kurze Geschichte von Python

- Die Sprache wurde Anfang der 1990er Jahre von Guido van Rossum am Zentrum für Mathematik (Centrum voor Wiskunde en Informatica) in Amsterdam entwickelt. Ursprünglich war sie als Nachfolger für die **Lehrsprache ABC** entwickelt worden. Guido van Rossum hatte auch an der Entwicklung der Sprache ABC mitgewirkt, so dass seine Erfahrungen mit ABC auch in Python einfließen.
- Der Name der Programmiersprache Python hat nichts mit der gleichnamigen Schlange zu tun. Die **britische Komikertruppe Monty Python** mit ihrem legendären Flying Circus stand Pate für den Namen.

Monty Pythons Spam-Sketch



<https://www.youtube.com/watch?v=anwy2MPT5RE>

Warum Python?

- Python genießt einen Ruf als **einfach zu erlernende Programmiersprache** mit klarer Struktur. Ihr Programmcode ist intuitiv nutzbar und gleichzeitig leicht lesbar.
- Trotz der Einfachheit bietet Python eine gute Skalierbarkeit und ist für komplexe Softwareprojekte einsetzbar. Aufgrund der ausdrucksstarken, minimalistischen Syntax sind Anwendungen mit wenigen Codezeilen und geringer Anfälligkeit für Programmierfehler realisierbar.
- Um für Einfachheit und Übersichtlichkeit zu sorgen, kommt Python mit sehr wenigen Schlüsselwörtern aus und verwendet **Einrückungen als Strukturierungselemente**.

Einrückungen und keine geschweifte Klammern

C/Java/JavaScript-Syntax

```
if (EXPRESSION) {  
    int x = 3;  
}
```

Python:

```
if EXPRESSION:  
    x = 3
```

Was ist eine PEP?

PEP steht für **Python Enhancement Proposal**. Es ist ein zentrales Werkzeug im Entwicklungsprozess von Python und dient als technische Spezifikation oder als eine Art Python-Entwicklungsrichtlinie.

PEPs sind **Vorschläge zur Verbesserung von Python**, die von der Community erstellt und diskutiert werden. Sie können neue Features, Prozessverbesserungen oder Umgebungsvorgaben enthalten.

PEPs spielen eine wesentliche Rolle in der **Evolution von Python**, da sie einen organisierten, transparenten und gemeinschaftlichen Ansatz zur Verbesserung der Sprache darstellen.

Eine Übersicht aller PEPs findet sich hier: <https://peps.python.org/>

PEP 8

PEP 8 enthält die Coding-Konventionen für Python-Code, der die Standardbibliothek in der Haupt-Python-Distribution umfasst.

<https://peps.python.org/pep-0008>

Module

Die **Python Standard-Bibliothek** ist in jeder Python-[Installation](#) verfügbar. Sie bringt eine große Anzahl sehr nützlicher Module mit.

Jeder angehende Python-Entwickler muss sich daher mit der Python-Standardbibliothek vertraut machen.

Ein Modul wird mit dem Schlüsselwort **import** in den Code importiert.

```
import random
```

Übersicht aller installierten Module

eine Übersicht aller installierten Module kann man sich mit dem Befehl

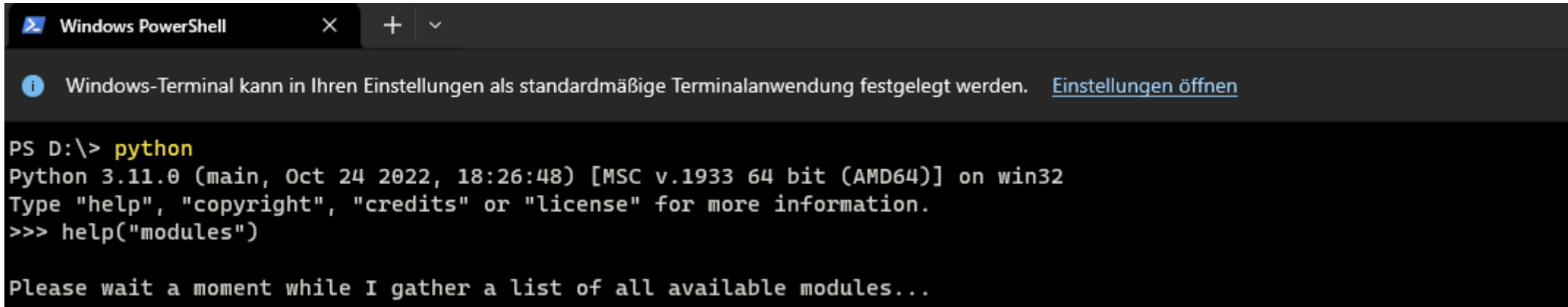
```
help("modules")
```

in der interaktiven Python-Shell ausgeben lassen.

Die interaktive Python-Shell öffnet man, in dem man im Terminal (Powershell, Bash, Eingabeaufforderung,...) folgendes eintippt

```
python
```

Python interaktive Shell und Hilfemodus



```
Windows PowerShell
Windows-Terminal kann in Ihren Einstellungen als standardmäßige Terminalanwendung festgelegt werden. Einstellungen öffnen

PS D:\> python
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> help("modules")

Please wait a moment while I gather a list of all available modules...
```

Der Hilfemodus

Mit `help` lassen sich Informationen und Hilfen über alle Python-Module und Klassen einholen.

```
help("random.randint")
```

liefert die Beschreibung für die Funktion `randint` aus dem `random`-Modul.

Schnittstellen, APIs und Bibliotheken

Zudem gibt es Schnittstellen oder Interfaces für viele Systemaufrufe und Bibliotheken. Die Programmiersprache lässt sich unter Verwendung von **C** oder **C++** erweitern.

Bekannte Data-Science Bibliotheken wie **Tensorflow** und **Sklearn** haben Schnittstellen für Python.

Bekannte Pakete wie **numpy**, **pandas** oder **matplotlib** machen aus Python eine ideale Sprache für die Data Science und das Machine Learning.

Wofür wird Python genutzt?

Web-Development, Serverseitige Anwendungen

Data Science/Machine Learning

Wissenschaft

Automatisierung und Steuerung

Systemprogrammierung

Netzwerkaufgaben