

Scriptsprache

script language

Scriptsprachen sind **Programmiersprachen**, deren Ziel im Grundsatz es ist, zumeist nur kleinere Anwendungen oder Anweisungsfolgen zu realisieren. Dabei weisen die Codesequenzen - die häufig auch als **Scripte** oder **Scripts** bezeichnet werden - bestimmte spezifische **Merkmale** auf. Eines dieser Merkmale ist es, dass **Scripte** vielfach nicht von einem **Compiler** in maschinenlesbaren **Code** übersetzt werden, sondern zur **Laufzeit** von einem **Interpreter** ausgeführt werden. **Scriptsprachen** sind oft auch spezifischen **Aufgaben** zugeordnet, so dass deren **Syntax** leicht zu überschauen ist.

Die weiteren signifikanten Merkmale von Scriptsprachen sind:

- die eingebaute Speicherverwaltung, das **Garbage Collection**,
- automatisch deklarierte Variablen,
- dynamische Klassenzugehörigkeit oder proto-typenbasierte **Vererbung** und
- fehlende Differenzierung von Typen (Typlos).

Die interpretierbaren Scriptsprachen haben gegenüber **Programmen** in kompilierten Programmiersprachen bei rechenintensiven Aufgaben einen deutlichen Geschwindigkeitsnachteil. Diesen Nachteil versuchen spezielle Methodiken in der Handhabung von Scriptsprachen zu umgehen. Hierzu gehören die Interpretierung durch eine **Virtual Machine**, die mögliche Einbindung von in Low-Level Programmiersprachen, wie die **Programmiersprachen C** etc. verfassten zeitkritischen Codesequenzen, oder ein Runtime-Optimizing sowie die Übersetzung in **Bytecode**.

Scriptsprachen und deren Schnittstellen.

Ursprünglich realisierten die Scriptsprachen nur interaktive **Schnittstellen** zur Eingabe von Kommandos an das jeweilige **Betriebssystem**. Durch die weitere Ergänzung der Scriptsprachen durch Variable und Ausdrücke sowie Kontrollstrukturen können eigen-ständige Codesequenzen - **Scripte** oder kleinere Programme - entstehen, die die Abläufe eines Systems automatisieren. Diese bezeichnet man unter **UNIX** als **Shell-Scripte** und unter dem Disk Operating System (DOS) und **Windows** als **Batch-Dateien**.

Ein weiteres **Segment** von Scriptsprachen ist deren Verwendung im Hinblick auf die Ergänzung von Funktionalitäten eines Anwendungsprogramms. Da-durch entstehen durch die Befehlsfolgen in einfachen flexibel zu verwendenden Textdateien unter Um-ständen völlig geänderte Funktionen. Beispielhaft für diese Art von Scriptsprachen seien genannt : **Visual Basic for Applications** (VBA) zur Verwendung in Microsoft-Programmen und Emacs-Lisp als die Scriptsprache des **Editors** Emacs.

Eine zusätzliche Gruppe definiert diejenigen Script-sprachen, welche als selbstständige - interpretative - Programmiersprachen angesehen werden können. Diese ermöglichen die Verwendung unterschiedlicher **Programmierparadigmen** und unterstützen die für eigenständige Software-Realisierungen notwendigen Konzepte. Beispiele für diese Art von Scriptsprachen sind:

Ruby, eine objektorientierte Scriptsprache

PHP, **Hypertext** Preprocessor, für die Entwicklung von Web-Applikationen

JavaScript, für dynamische Web-Applikationen

Python, eine objektorientierte Scriptsprache

Perl, für die Bearbeitung von **Texten**

REXX, eine Scriptsprache von IBM

LISP, zur Verarbeitung von Listen

Auch in dieser Gruppe gibt es jedoch noch Spezialisierungen. Dafür steht beispielsweise der unter UNIX verwendete Textprozessor **awk** oder die Scriptsprache **GLE** zur Generierung von **Grafiken**.

Verwendung finden Scriptsprachen häufig auch im **WWW**, wo sie vor allem für dynamische **Aspekte** oder die Realisierung ganzer Web-Anwendungen sowohl server- wie clientseitig eingesetzt werden. Die Bearbeitung von Scriptsprachen erfolgt immer in Form von Textdateien, so dass eine hohe Flexibilität gegeben ist. Auch durch den hohen Verbreitungsgrad bilden heute die Scriptsprachen häufig Anwendungsgebiete ab, welche ursprünglich den konventionellen Programmiersprachen vorbehalten waren.

Querverweise nach Scriptsprache von:

[tool command language](#) [practical extraction ...](#) [JavaFX](#) [Interpreter](#) [ActionScript](#) [NetWare](#)
[Groovy](#) [Ruby](#) [Bytecode](#) [Shell](#)