



Skriptsprachen

Moderne, objekt-orientierte
Skriptsprachen mit Betonung auf Ruby



Ein erster Überblick



- Merkmale
 - Implizit deklarierte Variablen
 - Automatische Typumwandlungen oder typenlose Programmierung
 - Automatische Speicherverwaltung
 - Meist interpretiert
 - Für kleine Projekte

- Typische Einsatzbereiche
 - Automatisierung von Abläufen
 - Shell-Programmierung, Starten anderer Programme
 - In Anwendungspaketen
 - Textverarbeitung
 - Reguläre Ausdrücke, String-Handling
 - Kleine „Wegwerf“-Programme
 - Administrative Routineaufgaben



- Beispiel: Shell-Programmierung

- Unix etc.: sh, csh, ksh, bash, tcsh; tcl

Bourne-Shell (System V), C-Shell (Berkeley Unix), Korn Shell, „Bourne Again“-Shell, verbesserte C-Shell

- DOS/Windows: BAT-Dateien, cmd; Power Shell
- OS/2: REXX

- Beispiel: Textverarbeitung

- awk (Aho, Weinberger, Kernighan), sed („streaming editor“)

- Nach wie vor aktuell!

- Kommerzielles Seminarangebot, Nov. 2008 in Wiesbaden (Ordix AG):
5 Tage zu Shell, Awk und Sed für 1690 € netto
- Details:

<http://www.ordix.de/trainingsshop/siteengine/action/load/kategorie/Programmierung/nr/50/index.html>

Online-Demo



Skriptsprachen, traditionell



- Beispiel: Anwendungspakete
 - VBScript, VBA (MS Office, aber auch AutoCAD, ARIS, ...)
 - Mathematica (M., Wolfram Research)
 - JavaScript (Acrobat, A. Reader; Adobe)
- Insgesamt:
 - Bis ca. Mitte der 1990er Jahre waren Skriptsprachen beliebte Hilfsmittel unter Spezialisten und Systemadministratoren
 - Sie ergänzten klassische Programmiersprachen, ohne mit ihnen zu konkurrieren
- Dann kam das WWW ...



- Das WWW
 - HTML (seit ca. 1991) allein genügte den rasch wachsenden Nutzern sehr bald nicht mehr. Gewünscht waren insbesondere:
 - Formulare mit eingebauter Validierung
 - Formulare u.a. HTML-Bestandteile, die sich im Dialog mit dem Anwender verändern
 - Dynamisch erzeugte HTML-Seiten, insb. mit Datenbank-Anbindung
- Folgen auf Clientseite
 - Entwicklung eines API-Standards für Seiteninhalte (*Document Object Model*, DOM)
 - Entwicklung von Skriptsprachen, Einbettung in HTML zur Programmierung des DOM („DOM scripting“) → DHTML
 - MS: VBScript, JScript (JavaScript + MS-Extras)
 - Netscape: LiveScript, später: JavaScript/ECMAScript
 - (Sun: Java applets)



- Folgen auf Serverseite
 - Erweiterung des einfachen HTTP-Servers um Möglichkeiten zur dynamischen Erzeugung von Inhalten
 - Frühe Template-Technik: *Server Side Includes* (SSI), Apache
 - Entwicklung einer Schnittstelle zur Einbindung von Zusatzprogrammen, die HTML-Seiten dynamische generieren: Das *Common Gateway Interface* CGI
- Die große Stunde von Perl
 - HTTP ist ein i.w. textbasiertes Protokoll
 - Die Skriptsprache Perl bot zu dieser Zeit schon gute Unterstützung für Arrays, Hashes und insb. Reguläre Ausdrücke, die sich als sehr hilfreich erwiesen bei der Verarbeitung von HTTP-Daten und der Erzeugung von HTML
 - Die Folge: Perl verbreitete sich rasch als „DIE“ Sprache zur CGI-Programmierung, Skriptsprachen insgesamt wurden mehr beachtet!



- Probleme mit CGI
 - Hohe Systemlast: Jeder Zugriff startet einen neuen BS-Prozess, bei Perl ist vor Ausführung der ersten Codezeile der Interpreter zu laden und der gesamte Code zu interpretieren
 - Oftmals unnötiger Aufwand
Statt eine ganze HTML-Seite dynamisch zu generieren, ist es oftmals sinnvoller, statische HTML-Seiten mit (wenigen) Stellen zu versehen, die sich unter Programmkontrolle expandieren lassen (Schablonentechnik, engl. *templates*)
- PHP: Die Template-Engine
 - Die an C erinnernde Skriptsprache PHP wurde zu genau dem Zweck geschaffen, HTML-Templates zu erzeugen und zu expandieren
 - Der PHP-Interpreter wird Teil des Prozesses des HTTP-Servers und steht damit sofort und ohne Prozess-Overhead zur Verfügung
 - Die Folge: Rasche Verbreitung von PHP als „eingebettete“ Sprache auf Serverseite



Dynamische Seitengenerierung mit PHP



- PHP
 - Ursprünglich als einfaches Werkzeug zur Verarbeitung von HTML-Schablonen incl. Datenbank-Anbindung entstanden, entwickelt sich PHP schnell weiter, hin zu einer universellen Skriptsprache.
 - Inzwischen sind zahlreiche Erweiterungen für viele Anwendungsfälle erhältlich. Template-Beispiel, mit "PI"-Notation (damit XML-konform!):

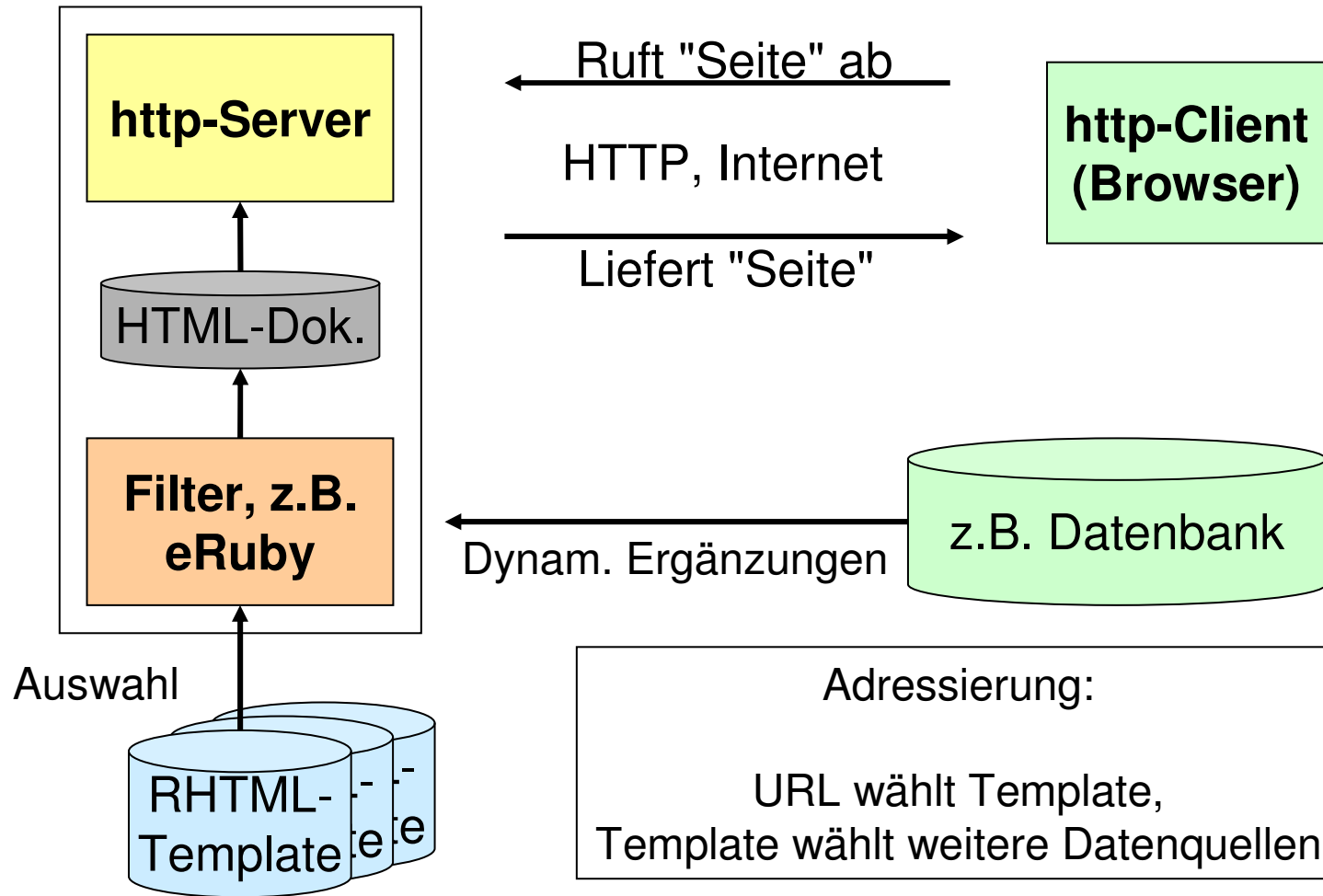
```
<?php
  if ($expression) {
    ?>
    <em>Dies ist richtig.</em>
    <?php
  } else {
    ?>
    <em>Dies ist falsch!</em>
    <?php
  }
?>
```



Dynamische Seitengenerierung: *Templates*



Prinzip der Seitengestaltung mit Schablonen (hier: Ruby)





- Merkmale
 - *Very high level languages*
 - Konsequent objekt-orientiert
 - Leistungsfähige und komplexe eingebaute Klassen
 - Dynamisch, mit *reflection*
 - Erhöhung der Entwickler-Produktivität
 - Viel Funktionalität mit wenig Code, gute Wartbarkeit*
 - Kurze Entwicklungszyklen*
 - Unterstützung für „agile“ Methoden der Software-Entwicklung
- Typische Einsatzbereiche
 - Prozessintegration
 - Automatisierung
 - *Rapid Prototyping*
 - Webbasierte Anwendungen
 - Trend zu *general purpose*-Sprachen



- Beispiel Ruby
 - Vollwertige, streng objekt-orientierte Programmiersprache
 - Unterstützt dennoch auch imperative, funktionale und deklarative Programmierung (*multi-paradigm*-Sprache)
 - Modularisierung, integrierte Unterstützung für Testing und Dokumentation und einfache *C-Extensions* befähigen die Sprache auch zum Einsatz für größere Projekte
 - Besonders sauberer Sprachentwurf, konsequent dynamisch, daher für didaktische Zwecke bestens geeignet (und unser Schwerpunkt!)
- Beispiel Python
 - Ebenfalls eine vollwertige, objekt-orientierte Skriptsprache
 - Die derzeit am weitesten verbreitete Skriptsprache der neuen Generation
- Beispiel Groovy
 - Eine noch junge Sprache, die viele Vorteile von Ruby übernommen hat, aber auf Java aufbaut und alle Java-Bibliotheken nutzen kann



Skriptsprachen, aktuell



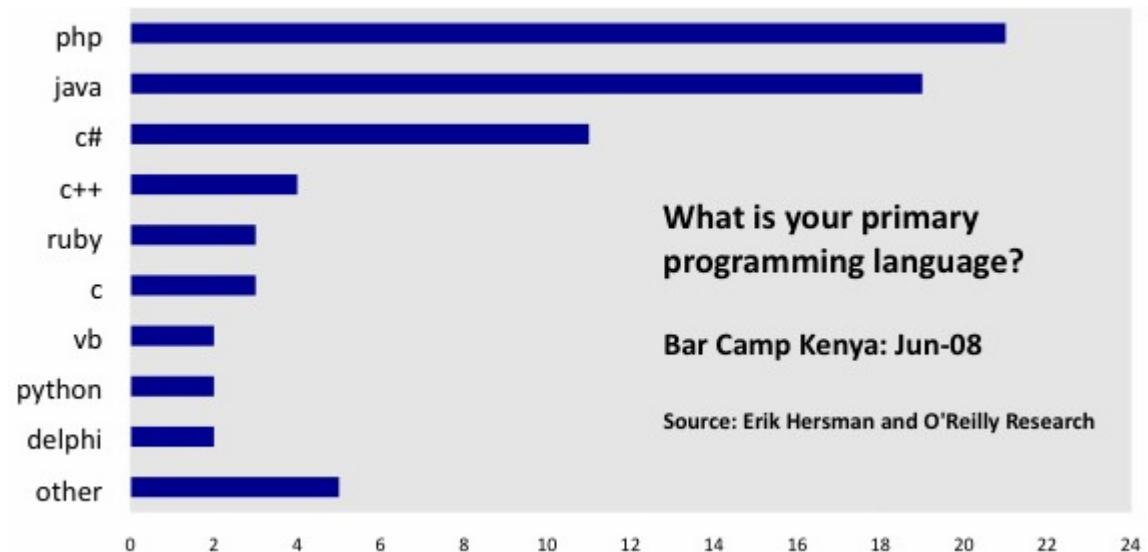
- Verbreitung und Bedeutung von Programmiersprachen
 - I.a. sehr branchen- und unternehmensabhängig
 - Keine klaren Kriterien, manches ist Interpretationssache

- Der Tiobe-Index

- Ein einfacher Popularitätsindex auf der Basis von Suchmaschinentreffern
- Quelle: <http://www.tiobe.com>

Online-Demo

- Die Verkaufszahlen von O'Reilly
 - <http://radar.oreilly.com>



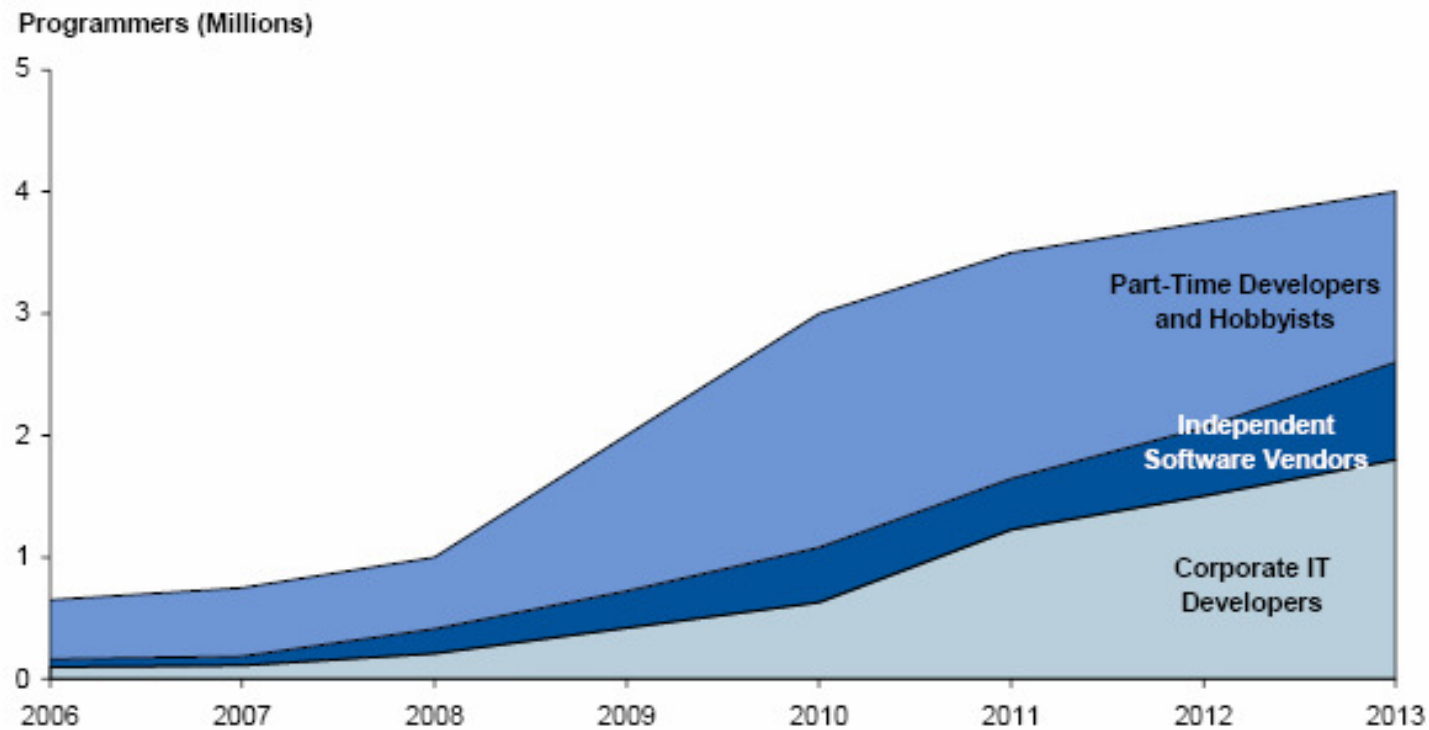


Skriptsprachen, aktuell



- Gartner-Prognose für Ruby, 3.12.2007:
 - „Findings: The Ruby Language Will Reach 4 Million Programmers by 2013“

Figure 1. Worldwide Ruby Programmer Count



Source: Gartner (December 2007)



- Shell-Programmierung u.a. klassische Skriptsprachen
 - Zu wenig produktiv, von den neueren Skriptsprachen zunehmend verdrängt, nur noch in Nischen gerechtfertigt (dort aber noch wichtig!)
 - Im Kurs werden sie keine weitere Rolle spielen
- Perl und PHP
 - Zwar immer noch weit verbreitet, aber mit absteigendem Trend
 - Unsauberer Wildwuchs statt Sprachen mit klarem Entwurf
 - Noch immer Sicherheitslücken in PHP
 - Im Kurs werden wir sie nicht näher behandeln
- JavaScript
 - Eine eingebettete Sprache mit erheblicher und im Zuge von „Web 2.0“ noch zunehmender Verbreitung
 - Im Kurs wird sie eventuell in Vergleichen herangezogen



Skriptsprachen in dieser LV



- Ruby
 - Unsere zentrale Skriptsprache für diese LV: Universell, sauber strukturiert und für die Ausbildung bestens geeignet. An ihr werden wir alle wichtigen Eigenschaften moderner Skriptsprachen exemplarisch kennen lernen.
 - **Diese Sprache sollten Sie am Ende des Kurses aktiv beherrschen!**
- Python
 - Der einzige ernsthafte Mitbewerber von Ruby, zur Zeit im „Westen“ sogar deutlich weiter verbreitet.
 - Nach Möglichkeit werden wir grundlegende Ideen und Konzepte von Skriptsprachen durch Vergleich ihrer Umsetzungen in Ruby und Python behandeln
- Groovy
 - Noch sehr neu und wenig verbreitet, sehr ressourcenhungrig, aber vielleicht die zukünftige Skriptsprache der Wahl für die große Java-Community
 - Ziel: Beobachten, erste Experimente mit Groovy-Programmen, Groovy in den o.g. Vergleich mit einbeziehen

Motto: „Don't learn languages, learn ideas.“



Skriptsprachen in dieser LV



- Eine Übersicht der Entwicklung der wichtigsten Computersprachen
 - <http://www.levenez.com/lang/> Online-Demo
- Eine Sammlung „aller“ Computersprachen
 - <http://people.ku.edu/~nkinners/LangList/Extras/langlist.htm>
 - Hinweis: „*Language Categories*“ ist lesenswert!
- Übrigens...
 - "*A Real Programmer can program FORTRAN in any language.*"
 - "*How to write unmaintainable code*" - Was Sie schon immer ahnten ;-)
<http://mindprod.com/jgloss/unmain.html>