Vorbemerkungen

Dieses Buch hat zwei Themenstränge:

- *Scriptsprachen* und
- *dynamische Webauftritte*, in denen diese Scriptsprachen verwendet werden.

Eine Abrundung erfährt diese Themenwahl durch *XML*, das das Potenzial hat, auch das Web und die Programmierung im Web entscheidend zu beeinflussen.
Scriptsprachen werden in der Hauptsache als Mittel zum Zweck der Anwendungsprogrammierung im Web dargestellt. Einige dieser Sprachen wie Perl und VBScript stehen aber ebenso für sich selbst als im Grunde nicht zweckgebundene Allzweck- und Alltagssprachen.
Gegenstand der Programmierung sind neben den üblichen Exemplifizierungen der Sprachelemente vor allem Programme als Versatzstücke für Anwendungen, die auf serverseitiger Datenhaltung etwa in Datenbanken basieren. Diese Versatzstücke finden auch in zwei vollständigen Webauftritten Verwendung.
Die Programmierung und mit ihr die Programmbeispiele umfassen *alle* Ebenen solcher Anwendungen, d.h. nicht alleine die Präsentationsebene (GUI), sondern ganz besonders die im Server lokalisierten Strukturebenen. Unter diese fällt vor allem die Persistenzebene, die in der Regel Datei- und Datenbankverwaltungssysteme umfasst.
Die Anwendungen enthalten also einerseits Dialogelemente, die mittels clientseitigem Scripting komfortabel und ergonomisch gestaltet werden können. Andererseits umfassen sie die Möglichkeit der Datenhaltung und Datenmanipulation, die auf der Basis der Kombination von Servertechniken wie CGI und ISAPI mittels serverseitigem Scripting effizient realisiert werden kann. Und die Anwendungen haben immer eine dazwischenliegende mittlere Ebene, in der mittels Scripting Daten aufbereitet werden, um sie zu speichern oder anzuzeigen, oder sie in beliebiger Art und Weise weiterzuverarbeiten. All diese Aspekte fließen in zwei vollständigen Anwendungsbeispielen mit E-Commerce-Charakter, einer virtuellen Spielzeugauktion und einer virtuellen Weinhandlung, zusammen.
Für ihre Verwendung im Web sind Scriptsprachen auf einen bestimmten Zweck hin konfektioniert, z.B. für Interaktionen mit einem Browser oder einem Webserver, für die

Manipulation von Datenbeständen, für administrative Aufgaben usw. In der Regel geschieht dies über Bibliotheken mit Programmen für die Manipulation der Browserelemente, für eingehende und abgehende Datentransfers im Server, zur Manipulation von Datenbeständen etc. Bei den behandelten Scriptsprachen sind dabei z.B. Objektmodelle für Browser und Server impliziert, die sie, serverseitig kombiniert mit Programmanschlusstechniken wie ISAPI und CGI, zu einem leistungsfähigen Werkzeug für die Webprogrammierung machen. Der Sprachkern der Scriptsprachen ist an Sprachklassiker angelehnt, z.B. an C, C++ oder (Visual) Basic, d.h. sie entfernen sich syntaktisch meist nicht allzu weit von ihren Vielzweckvorbildern.

Zielsetzungen

Das Buch will vor allem eine Übersicht über die im Web heute üblichsten Scriptsprachen geben. Es will in diese Scriptsprachen und ihr jeweiliges Umfeld gründlich einführen. Es will damit die Möglichkeit geben, die Sprachen gegeneinander abzuwägen und aus ihnen eine optimale Wahl zu treffen, zumindest soweit Wahlmöglichkeiten bestehen. Und es will schließlich den Einsatz der Sprachen in konkreten Anwendungsfällen zeigen.
In vergleichbarer Weise wird auch XML präsentiert: Zunächst eine Einführung in den Gegenstand mit vielen Beispielen zu XML selbst und sodann Einsatzmöglichkeiten von Scriptsprachen (Perl) zur Programmierung einfacher XML-Prozessoren.
Dies ist ein Buch für Programmierer, d.h. Kenntnisse in einer prozeduralen Programmiersprache und Erfahrungen mit ihr werden vorausgesetzt. In welcher Sprache, ist nicht sehr erheblich, auch wenn C- und Java-Programmierer ein vertrauteres Umfeld vorfinden als Pascal-Programmierer. Weiter sollten die Interna von Web und Webdokumenten sowie Datenbanken nicht völlig fremd sein. Zu vielem werden aber Einstiegshilfen angeboten, als Exkurse im Buch selbst und als Einleitungen bzw. Ergänzungen auf der Website zum Buch, `http://www.wsite.de`.
Das Buch richtet sich aber auch an jene, die sich z.B. mit Webauftritten eher konzeptionell oder koordinierend befassen, d.h. zwar nicht selbst programmieren, aber die Methodik so weit beherrschen wollen, wie das für ihre Tätigkeit von Vorteil ist.
Unter den genannten Prämissen wendet sich das Buch nicht nur an den Profi, sondern auch an alle, die selbst anspruchsvolle Websites gestalten wollen. Für beide ist in Form von Spracheinführungen, Fallbeispielen, Anwendungsszenarien sowie elementaren Einführungen in assoziierte Themen wie SQL (sehr kurz), HTML (kurz) und XML (ausführlich) gesorgt.
Ziel allen Bemühens ist die Programmierung dynamischer bzw. interaktiver Webauftritte, die einerseits Datenhaltung in Datenbanken oder Dateien erforderlich machen und andererseits über komfortable und leistungsfähige grafische Bedienerschnittstellen verfügen. Deren Instrumente sind die Scriptsprachen Perl, VBScript, JavaScript und PHP. Dabei sind insbesondere solche Problemfelder relevant, wo zeitlich getrennte Kommunikationsvorgänge („Sitzungen“) in sich zusammenhängen und sich bedingen, wie dies etwa bei der Abwicklung von Tagungen und Kursen über das Web oder bei E-Shops der Fall ist, um nur einige wenige Beispiele zu nennen. Die Resultate von Kommunikationsvorgän-

gen müssen dabei serverseitig z.B. in Datenbanken festgehalten werden. Von solchen Daten hängen dann die weiteren Kommunikationsabläufe ab. Konkret: Auf die Anmeldung zu einem Kurs oder einer Tagung folgen Bestätigung und Rechnung, auf das Ausbleiben einer Zahlung erfolgen Mahnungen, auf eine Kundenanfrage hin wird ein Abwicklungsstatus mitgeteilt etc.
Dieses Buch kann alleine wegen seiner begrenzten Seitenzahl kein Ersatz für die umfangreichen und oft hervorragenden Dokumentationen, Handbücher und Tutorials im Web sein. Solche Unterlagen gibt es dort zu so gut wie jedem der Einzelthemen, und zwar mindestens in einem der gängigen Formate HTML, PDF oder PostScript sowie in installierbarer Form. Einschlägige Webadressen sind jeweils, die einzelnen Kapitel abschließend, unter „Quellen“ angegeben. Der Leser hat vermutlich den größten Nutzen, wenn er beides verwendet, sowohl das Buch als auch die passenden Dokumente aus dem Web.

Anmerkung: Das Buch wendet sich natürlich auch an den Sammler von Programmiersprachen.

Übersicht

Außer dem ersten (Einleitung) und dem sechsten (E-Commerce) bestehen alle Kapitel etwa je zur Hälfte aus einem an der Sprache und aus einem an Anwendungen orientierten Teil. Allerdings ist diese Unterteilung weniger strikt, als es vielleicht scheint. Denn alle Scriptsprachenkapitel haben einen frühen Abschnitt zum „Reinschmecken“, in dem nicht nur „Hello World!“ präsentiert wird, sondern vor allem auch Beispiele mit Anwendungsspezifischem, nicht nur mit Sprachtypischem.
Die Sprach- und das XML-Kapitel haben den Charakter von Monographien, sie könnten also mit kleinen Einschränkungen auch isoliert gelesen werden. Allerdings sind in zwei Fällen Kenntnisse in Perl erforderlich, und zwar im PHP-Kapitel – hier wird bei der Sprachdefinition darauf Bezug genommen – und im XML-Kapitel, in dem Perl Implementierungssprache von XML-Prozessoren ist.
Das Buch gliedert sich in die folgenden Kapitel:

Kapitel 1: Einleitung will auf die folgenden Kapitel vorbereiten. Dazu werden einige grundlegende Begriffe wie Statik und Dynamik im Web, Scriptsprachen, Modellierungen usw. eingeführt. Dem folgen erste Schritte, die meist gemeinschaftlich, d.h parallel in allen Scriptsprachen durchgeführt werden. Dies betrifft sowohl die sprachlichen Grundlagen als auch einige wichtige Anwendungsfälle. Dabei werden u.a. der Sitzungsbegriff und die Möglichkeiten der Sitzungsverwaltung im Web näher untersucht.
Außerdem ist ein Exkurs über HTML enthalten (Abschnitt 1.7).

Kapitel 2: Clientseitiges JavaScript ist eines der beiden Kapitel, dessen Thema die clientseitige Scriptprogrammierung ist. Sie beruht auf in HTML eingebetteten Programmen, die clientseitig unter Kontrolle eines Webbrowsers ablaufen. Nach einer Übersicht über Javascript werden das zugrunde liegende Objektmodell, die Einbettung der Sprache in den HTML-Kontext und die prozeduralen Elemente der Sprache

(`while`, `for`, `if` etc.) erarbeitet. Typische Anwendungen sind in kompakten Beispielen zusammengefasst.

Kapitel 3: VBScript und ASP hat zwar auch die clientseitige Programmierung zum Inhalt, Schwerpunkt ist jedoch die Serverseite, d.h. VBScript im Zusammenhang mit ASP. *Clientseitig* werden bestehende Analogien und Übereinstimmungen mit dem JavaScript-Objektmodell ausgiebig genutzt, d.h. wesentlich Neues gegenüber Kapitel 2 kommt allenfalls im Sprachlichen hinzu (VBScript ist ein Basic-, kein C-Abkömmling). Das *serverseitige* Objektmodell wird mittels einfacher anwendungsnaher Beispiele veranschaulicht, und die Programmierung mit Datei- und Datenbankkomponenten wird eingeführt. Dem liegen so genannte Active Server Pages (ASP) zugrunde, die ein Verfahren bezeichnen, mit dem Programmcode in HTML eingebettet und von einer passenden „ActiveX Scripting Engine", nämlich VBScript oder auch JScript, ausgeführt werden kann.
Außerdem sind zwei Exkurse enthalten; einer beschäftigt sich mit dem http-Protokoll (Abschnitt 3.5.1) und der andere mit relationalen Datenbanken und SQL (Abschnitt 3.7.1).

Kapitel 4: Serverseitiges Scripting mit Perl ist ähnlich strukturiert wie das vorangehende Kapitel. Grundlage sind CGI (Common Gateway Interface) und ISAPI (Internet Server Application Programming Interface) als Programmierschnittstellen und Perl als Sprache für die Programmierung an diesen Schnittstellen. Die Anwendungsbeispiele sind analog zu denen im vorangehenden Kapitel, und auch das Thema Persistenz ist in Form von Datenbankanbindungen über ODBC adäquat vertreten. Demonstriert werden u.a. auch der Mailversand mittels Formularen, Server Push und File Upload.

Kapitel 5: Serverseitiges Scripting mit PHP gibt Übersichten über Charakteristika und Einsatzmöglichkeiten von PHP in der Webprogrammierung. Entsprechend der zahlreichen Ähnlichkeiten werden bestehende Parallelen zu den Geschwisterprogrammiersprachen Perl und JavaScript ausgiebig genutzt. Beispielschwerpunkt ist auch hier der Umgang mit serverseitigen Datenquellen. Anwendungsbeispiele sind wie in Kapitel 3 ein Gästebuch und eine grafische SQL-Programmierschnittstelle, hier nicht nur über ODBC zu Access, sondern auch zu MySQL.

Kapitel 6: E-Commerce-Anwendungen besteht aus zwei umfangreicheren beispielhaften Anwendungen, einer virtuellen Weinhandlung in Perl und einer virtuellen Spielzeugauktion in VBScript/ASP. Beide beginnen mit der Vorstellung der Konzeption, der Präsentation der Kundenschnittstelle, dem Entwurf des Datenmodells und der Angabe der Arbeitsmittel. Dem folgt jeweils in einem zweiten Abschnitt die Kodierung, deren wichtigste Teile wiedergegeben und erläutert sind (vollständig sind sie auf der Website zum Buch bereitgestellt).

Kapitel 7: XML und Scriptprogrammierung befasst sich mit der Bedeutung von Scriptsprachen in XML, das eine zunehmend wichtige Rolle in Webauftritten spielen wird. Es wird gezeigt, wie eigene Datenstrukturen als XML-Dokumente erstellt und wie sie mittels Scriptprogrammen und XML-Parsern weiterverarbeitet bzw. wiedergegeben werden können. Das Kapitel enthält die notwendigen Vorbereitungen auf das Thema. Dazu gehört insbesondere eine Übersicht, in der XML als por-

tables Datenformat vorgestellt wird und in der Konzepte wie Wohlgeformtheit, DTDs, XML-Prozessoren u.a. eingeführt werden.

Website zum Buch

Verfügbarkeit und Kosten des Internet sind mittlerweile derart, dass die Distribution bzw. Beschaffung von Dokumenten, Software und Beispielen über das Web bzw. Internet kaum mehr Nachteile gegenüber einer CD-Beilage hat. Auch aus diesem Grunde sind alle Beispiele und Ergänzungen auf der Website `http://www.wsite.de` zur Einsicht bereitgestellt und können von dort einzeln oder geschlossen in gepackter Form auf den eigenen Computer transferiert werden.
Die Buchwebsite ist analog den sieben Kapiteln gegliedert. Unter diesen Gliederungspunkten sind die Beispiele aus den jeweiligen Kapiteln aufgenommen, außerdem fallweise Ergänzungen zu den Kapiteln wie Funktions- und Objektübersichten. Unter den zusätzlichen alphabetischen Gliederungspunkten sind u.a. eine SQL-Einführung und ergänzende Installationsdetails zu finden.
Im Einzelnen gliedert sich die Website wie folgt:

1. **Einleitung**
   Beispiele
2. **Clientseitiges JavaScript**
   Beispiele, Zusammenstellung von JavaScript-Objekten
3. **VBScript und ASP**
   Beispiele
4. **Serverseitiges Scripting mit Perl**
   Beispiele, Standardvariablen und Standardfunktionen
5. **Serverseitiges Scripting mit PHP**
   Beispiele
6. **E-Commerce-Anwendungen**
   Vollständige Programmcodes: virtuelle Weinhandlung, virtuelle Spielzeugauktion
7. **XML und Scriptprogrammierung**
   Beispiele, EBNF, XML-Referenz in EBNF

A. SQL: Kurzeinführung
B. HTML: Kurzübersicht
C: Installationen: Webserver, Sprachen und Werkzeuge (Ergänzungen)
D: Korrigenda

Die Beispiele

Alle Programmbeispiele sind möglichst unkompliziert geschrieben, d.h. erklärungsbedürftige Kodierungstechniken sind weitgehend vermieden. Die Beispiele sind überwiegend vollständig und unmittelbar ablauffähig. Lediglich Programmfragmente zur Exemplifizierung der Sprachsyntax, die in der Regel nur wenige Zeilen Umfang haben, müssen

meist noch in den richtigen Sprachkontext gebracht werden, um ablaufen zu können.
Um die Beispiele übersichtlich zu halten, sind sie möglichst knapp formuliert und auf das Grundsätzliche konzentriert. Ihre Überschaubarkeit wird auch dadurch gefördert, dass Leer- und Kommentarzeilen sowie Fehlerbehandlungen auf ein Minimum reduziert sind. Überhaupt wird mit Kodezeilen, die nicht direkt zur Lösung des jeweiligen Problems beitragen, sparsam umgegangen. Die erforderlichen Erläuterungen der Programme sind grundsätzlich im laufenden Text vorgenommen, bei kürzeren meist im Anschluss an das Programm und bei längeren vor bzw. nach zusammenhängenden Programmteilen.
Um Fehler im Programmtext möglichst gering zu halten, sind die abschließenden Tests auf der Basis des Buchmanuskripts durchgeführt worden: Die Programme wurden aus dem Manuskript per „Kopieren-und-Einfügen“ in einen Editor kopiert, gespeichert und ausgeführt. Eventuell noch vorhandene Fehler wurden direkt im Manuskript korrigiert; danach wurden die Programme wie eben beschrieben erneut getestet.
Die Beispiele haben in der Regel die folgende Form:

```
<!-- ./grundlagen/helloworld.asp -->
<%@Language=VBScript%><%
  ...
Programmcode
  ...
%>
```

Im Kommentar ist das relative Verzeichnis, in dem das Quellprogramm im Web zu finden ist, angegeben. An die Stelle des Punktes muss im wesentlichen nur noch die passende Adresse der Buchwebsite gesetzt werden.
Ihre Ablauffähigkeit stellten *alle* Programme in der folgenden Arbeitsumgebung unter Beweis:

- Windows 95, Windows 2000;
- Apache 2.3.12 und Internet Information Server 4.0 (IIS) bzw. Personal Webserver 4;
- JavaScript 1.3 unter Internet Explorer 5 und Netscape Navigator 4.7, VBScript 5;
- ASP/VBScript, PHP 3 mit Apache bzw. PHP 4 mit IIS, Perl 5.005 (ActiveState);
- ODBC, ADO, MySQL 3.23.22-Beta.

Der Hauptgrund für die Festlegung auf die Windows-Plattform ist, dass alle Scriptsprachen und Webbrowser, aber auch Entwicklungs- und Prüfwerkzeuge für z.B. XML, darauf ohne weiteres ablauffähig sind, selbst der MySQL-Datenbankserver. Die erforderlichen Installationsarbeiten sind fast immer geradlinig und führen schnell zum Erfolg. Und mit ODBC ist eine flexible Datenbankintegrationsmöglichkeit verfügbar. Alle Beispiele des Buches können somit auf einem einzigen Rechner unter einem einzigen Betriebssystem nachvollzogen werden, wozu die Voraussetzungen relativ einfach geschaffen werden können. Zudem sind alle Arbeitsmittel kostenlos verfügbar (viele davon sind Open Source). Zum Erlernen all dieser Techniken bestehen somit fast ideale Verhältnisse. (Eine vergleichbare Situation ist für Linux kaum herstellbar, denn der gesamte ASP-Komplex ist problematisch und nicht ohne erheblichen Mehraufwand, auch die Kosten betreffend, integrierbar.) Andererseits ist die Transformation des Erlernten in

eine Linux-Umgebung kaum mit nennenswertem Aufwand verbunden, ASP natürlich ausgenommen.

Viele der Beispiele wurden zusätzlich stichprobenartig getestet mit

- Windows NT 4 Workstation und Server (die meisten Beispiele), Linux (SUSE 6.2);
- OmniHTTPD (vor allem Perl-Programme), Internet Information Service 5 bzw. Personal Webserver 5;
- Netscape Navigator 6 pr 2;
- PHP 4, Perl 5.6 (ActiveState);
- MySQL 3.22.33 (Linux).

Allerdings kann erfahrungsgemäß niemals ganz ausgeschlossen werden, dass neue Softwareversionen sowie Modifikationen in der Kombination von Webserver, Betriebssystem, Sprache, Browser und Datenbank zu erheblichen Unverträglichkeiten bei zuvor einwandfrei funktionierenden Anwendungsprogrammen führen können.

Konventionen

Durch unterschiedliche Schriftarten und Schriftschnitte werden Textteile in diesem Buch wie folgt gegeneinander abgehoben:

`Courier` wird als Schrifttyp für Programmcode (JavaScript, Perl, PHP, SQL etc.), Programm- und Dateinamen, URLs und Internetadressen usw. verwendet.

`Courier kursiv` bezeichnet Platzhalter in Programmen, das heißt, an ihrer Stelle müssen bei Umsetzung noch gültige Werte eingesetzt werden.

`Courier fett / Courier fett kursiv` lenkt die Aufmerksamkeit auf Teile in Programmbeispielen, die meist im laufenden Text näher erläutert werden.

Times kursiv dient der Hervorhebung, weist also beispielsweise auf einen neuen oder wichtigen Begriff hin oder signalisiert einfach ein "Obacht!"

KAPITÄLCHEN werden gelegentlich verwendet, um Produkt- und Firmennamen hervorzuheben.

`name()` in Courier und mit leerer Klammer bezeichnet den Namen einer Methode im laufenden Text, z.B. "Die `name()`-Methode ist ...". Die leere Klammer `()` dient lediglich der Kennzeichnung von *name* als Methode und sagt nichts über den Klammerinhalt, d.h. die Parameter der Methode, aus. Ausnahme: VBScript, wo die Parameterübergabe beim Aufruf von Prozeduren gelegentlich ohne Parameterklammern auskommen *muss*.

Fachsprache:  oder  ?

Programmiersprachen sind in der Regel englischsprachig, das Internet ist englischsprachig, und auch die Primärliteratur ist fast immer englischsprachig. Fachtermini werden daher nur dann eingedeutscht, wenn sie in der Übersetzung geläufig sind. Das ist etwa bei der Datenbankterminologie mit wenigen Ausnahmen der Fall. Sonst werden überwiegend

die englischsprachigen Begriffe verwendet.
Englischsprachige Beispiele sind etwa: Script, Compiler und compilieren, Computer, Webbrowser, Entity, Host, Client, Server, Interface, Webserver, Webbrowser und Website, Parser und parsen etc.
Deutschsprachige Beispiele sind: Anweisung, Ausdruck, Begrenzer, Wohlgeformtheit, Modifizierer, Anforderung und Antwort (also nicht: Statement, Expression, Delimiter, Well-formedness, Modifier, Request und Response).