

# **Beschreibung und Vergleich serverseitiger, dynamischer Internetprogrammierung**

Jakob Heidebrecht  
MatrikelNr. 370578

HfT-Stuttgart

# Inhaltsverzeichnis

<b>1. Einleitung</b> .....	2
<b>2. Server-Client Architektur</b> .....	3
2.1 HTML.....	3
2.2 Browser.....	3
2.3 Webserver.....	3
<b>3. CGI/Perl</b> .....	4
3.1 Technologie.....	4
3.2 Syntax.....	5
<b>4. PHP</b> .....	6
4.1 Technologie.....	6
4.2 Syntax.....	7
<b>5. Java Servlets</b> .....	8
5.1 Servlet Technologie.....	8
5.2 Servlet-Container.....	8
5.3 Servlet API.....	9
5.4 Servlet-Aufbau.....	9
<b>6. JSP</b> .....	11
6.1 Servlet-Technologie.....	11
6.2 JSP-Syntax.....	11
<b>7. ASP</b> .....	13
7.1 Technologie.....	13
7.2 Syntax.....	13
<b>8. ASP.NET</b> .....	14
8.1 ASP = ASP.NET ? .....	14
8.2 .NET-Framework.....	14
8.3 Syntax.....	15
<b>9. Vergleich</b> .....	16
9.1 Bedienbarkeit.....	16
9.2 Portierbarkeit.....	16
9.3 Performance.....	17
9.4 Resümee.....	17
<b>Literaturverzeichnis</b> .....	19

## Einleitung

Stellen Sie sich vor, Sie melden sich bei Ihrem Online-Emailaccount an, tragen Benutzernamen und Passwort ein - nichts geschieht. Sie sehen ein statisches Bild - es passiert nicht das Geringste. So oder ähnlich würde das Internet ohne dynamische Internet-Programmiersprachen aussehen.

Ende der 1960er Jahre, als das Internet entstand, war das oberste Ziel Informationen auszutauschen. Lange war das Internet nicht für die Massenverwendung vorgesehen, erst 1989 mit der Entstehung des World Wide Web und HTML wurde das Internet für breite Massen zugänglich. Ende der 1990er wuchs das Internet schnell und es bestand großer Bedarf, HTML-Seiten dynamisch zusammenzusetzen.

Erst dynamische Technologien wie CGI, Servlets und Scriptssprachen ermöglichten einen dynamischen Zusammenbau einer HTML-Seite, die nur Ihre persönlichen Emails und Kontaktdaten enthält. Erst solche Technologien erlaubten es, die gleiche Seite Millionen mal anders aussehen zu lassen.

Als der Bedarf an dynamischen Technologien stieg, entstanden sehr viele Sprachen, einige werden nicht mehr eingesetzt, andere sind so aktuell wie früher. Diese Arbeit kann nicht den vollen Umfang der dynamischen Internet-Programmiersprachen aufzeigen, es sollen lediglich vier dieser Technologien beschrieben und verglichen werden, die alle auf dem Webserver ablaufen.

Diese Arbeit soll zuerst mit einer Einführung in die Server/Client Architektur verdeutlichen, welche Aufgaben einzelne Komponenten wie Browser, Webserver und Servlet Container haben. Danach werden die vier Technologien CGI mit Perl, PHP, Java Servlets/JSP und ASP vorgestellt. Abschließend werden die Vor- und Nachteile dieser Technologien aufgezeigt und ein Vergleich erstrebt, der aussagen soll, ob es die Beste unter diesen Sprachen ist, oder man einen Kompromiss zwischen ihnen finden kann.

## 2. Server-Client Architektur

### 2.1 HTML

Um Inhalte strukturiert und flexibel darzustellen, wird heute die Auszeichnungssprache HTML verwendet. Diese wurde 1989 am CERN in Genf festgelegt und gilt als Standard, der vom World Wide Web Consortium (W3C) weiterentwickelt wird.

HTML liegt zur Zeit in der Version 4.01 vor.

### 2.2 Browser

Um HTML-Seiten anzuschauen, benötigt man einen WWW-Browser, dieser kann die Auszeichnungen (engl. Markups) auswerten und den Inhalt der HTML-Seite dementsprechend darstellen.

Grundsätzlich unterscheidet man zwischen zwei Arten von Browsern: textbasierte und grafische Browser.

Die textbasierten Browser bieten lediglich die Möglichkeit Text darzustellen, meistens gibt es Möglichkeiten Grafiken der HTML-Seiten abzuspeichern oder mit externen Programmen darzustellen. Die Ansicht dieser ist jedoch nicht möglich. Ihre Anwendung empfiehlt sich für Blinde, da Text einfacher in Sprache umgewandelt werden kann, und für gezielte Webrecherche für erfahrene Benutzer, da keine Bilder oder Werbung den Nutzer irritiert. Beispiele für textbasierte Browser sind Links, Lynx und w3m.

Grafische Browser setzen ein grafisches Betriebssystem voraus und bieten alle Möglichkeiten der Darstellung. Mit ihnen ist es möglich Text, Bilder, Filme, Java-Applets usw. anzeigen zu lassen, für „Nicht-HTML“ Elemente verwenden diese Browser Plugins, die für die Darstellung z.B. von Java-Applets verwendet werden. Diese Art Browser werden heute teilweise mit weiteren Funktionen ausgestattet wie E-Mail, die den Komfort der Benutzer vergrößern sollen. Beispiele für komplette Pakete sind Mozilla, Opera und Netscape, diese Browser bieten weitere Funktionen. In letzter Zeit hat eine Bewegung eingesetzt, die sich vermehrt nur auf das Internet konzentriert. Beispiele für diese Browser sind: Galeon und Firefox.

### 2.3 Webserver

Als Server werden im allgemeinen Sprachgebrauch zwei verschiedene Sachen bezeichnet.

Zum Einen wird ein Rechner, auf dem Dienste laufen, die vom Internet aus abgerufen werden können, als Server bezeichnet. Solche Rechner stehen in Rechenzentren, haben einen sehr guten Anschluss ans Internet und sind speziell auf Internetanfragen ausgelegt, Sie haben sehr effiziente Hardware und Software und haben meistens 64 Bit Technologie, ein Vorteil dieser ist zum Beispiel „unbegrenzter“ Arbeitsspeicher, mit 32 Bit kann man 4 GB adressieren mit 64 Bit 4 Milliarden mal so viel. Diese „Server“ werden zum hosten (bereitstellen) von Webseiten für Clients (Benutzer) benutzt.

Auf diesen Rechnern läuft Software, die darauf ausgelegt ist Webseiten zu hosten, diese Software wird ebenfalls Server genannt, genauer Web-Server oder HTTP-Server. Die Funktion eines HTTP-Servers kann wie folgt erfolgen: Ein Client am Browser schickt eine Anfrage (HTTP-Request) an den HTTP-Server, in dem er z.B. eine URL in die Adresszeile des Browsers eingibt: wie <http://www.google.de>. Der Server verarbeitet die Anfrage intern, holt die benötigten Seiten aus dem Inhalt, der dem HTTP-Server zur Verfügung steht und liefert die Gewünschte Antwort (HTTP-Response) an den Browser. Bekannte Vertreter dieser Web-Server sind IIS (Internet Information Services) der nur unter Windows läuft und Apache HTTP-Server, wohl der bekannteste Web-Server.



## 3. CGI / Perl

### 3.1 Technologie

CGI (Common Gateway Interface) ist eine Schnittstelle des Web-Servers, die erlaubt Anfragen eines Web-Browsers von Programmen ausführen zu lassen, und die Ausgabe wieder an den Web-Browser zurückzuliefern. Diese Schnittstelle macht eine Web-Seite erst interaktiv und gibt ihr die Möglichkeit, auf Eingaben des Nutzers zu reagieren und Daten auf dem Web-Server nachträglich zu verändern.

Das CGI steht zur Verfügung, sobald ein Web-Server mit CGI-Unterstützung installiert ist. Um die Schnittstelle nutzen zu können, benötigt man ein Verzeichnis, das CGI-Programme enthält und auf das der Web-Server zugreifen darf. Durch die CGI-Schnittstelle können nicht nur Programme ausgeführt werden, sondern auch eine Reihe von Daten an diese Programme weitergeleitet, die der Web-Server speichert und die ein CGI-Script auslesen kann um Daten verarbeiten zu können. Diese Daten werden in so genannten CGI-Umgebungsvariablen gespeichert.

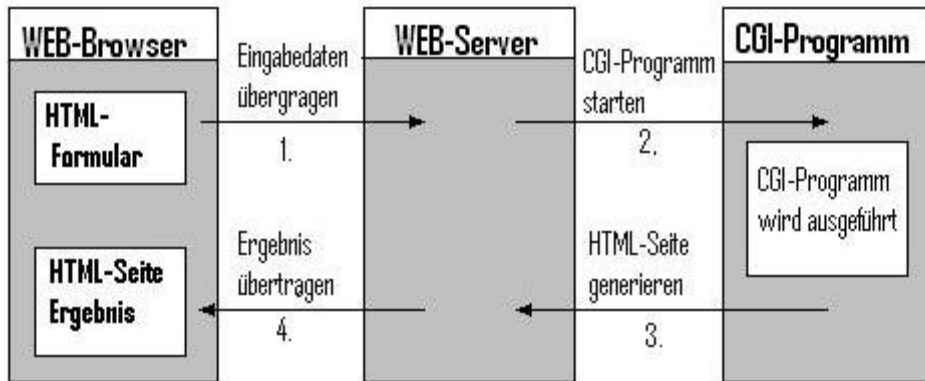
Derzeitig liegt die CGI-Schnittstellendefinition in der Version 1.1 vor. Es gibt jedoch andere Schnittstellen für den Web-Server von kommerziellen Herstellern, wie z.B. NSAPI von Netscape und ISAPI von Microsoft, beide Produkte sind auf Produkte der jeweiligen Hersteller optimiert und bieten ein wesentliches Geschwindigkeitsvorteil im Vergleich mit CGI. Der Vorteil von CGI sprechen trotzdem für sich, CGI ist kostenfrei, kommerziell unabhängig und produktübergreifender Standard, der in dieser Fülle nicht durch kommerzielle Produkte ersetzt werden kann.

Die meistgenutzte Programmiersprache für CGI-Scripts ist schon immer Perl gewesen und ist es immer noch. Perl (practical extraction and report language), 1987 von Larry Wall entwickelt, befindet sich zur Zeit in der Version 5.8.7. Damals war der Anlass für Larry Wall, eine neue Sprache zu entwickeln, der Bedarf einer Sprache, die effizient Berichte aus Log-Dateien zusammenfassen kann. Wie der Name schon sagt, war Perl damals eine bessere Shell, die gut mit Dateien und Text umgehen konnte und Berichte ausgeben konnte. Mit den späteren Versionen wurde Perl immer mächtiger. Mit der Version 5 hat sich Perl zu einer modernen Sprache weiterentwickelt. Es wurde die Möglichkeit eingeführt, die Sprache durch weitere Module zu ergänzen, welches einer der größten Vorteile von Perl ist. Ein weiterer wichtiger Punkt war die Objektorientierung, die ebenfalls mit der Version 5 eingeführt wurde.

Perl ist nahezu auf allen Betriebssystemen lauffähig und gehört bei Linux sogar zu den Systembestandteilen.

Perl bietet wesentliche Vorteile gegenüber anderen Programmiersprachen, warum es auch als CGI Scriptsprache Nr. 1 eingesetzt wird. Grundsätzlich können alle Sprachen, die vom Standard-Input lesen und auf das Standard-Output schreiben können, für CGI verwendet werden. Den Nachteil anderer Sprachen stellt die Kompilierung, so wie Linkung auf das jeweilige System, bevor diese ausgeführt werden können. Perl jedoch wird als Scriptsprache nicht kompiliert, sondern auf dem Perl-Interpreter ausgeführt, was die Scripte universell einsetzbar macht. Des Weiteren ist Perl extrem leistungsfähig und der Perl-Interpreter hochgradig ausgereift. Ein weiterer Vorteil von Perl liegt aufgrund seiner Geschichte in den mächtigen Funktionen für Zeichenkettenoperationen oder für das Lesen und Schreiben von Dateien.

Die Möglichkeiten von CGI sind vielfältig. Da CGI-Scripte auf dem Server ausgeführt werden, können diese auf Datenbanken zugreifen, auf dem Dateisystem lesen und schreiben und andere Programme ausführen. Der Rückgabewert dieser Programme ist dann der dynamische HTML-Code, so funktionieren z.B. Gästebücher.



**Bild 3**

### 3.2 Syntax

Ein CGI-Script kann auf verschiedene Arten aus einer HTML-Datei aufgerufen werden.

1. Über ein Formular: Dabei wird als ACTION eine CGI-Datei angegeben:  
**<form action="/cgi-bin/gästebuch.pl method="post" >**  
 So werden dem CGI-Script Daten übergeben, die es verarbeiten kann und einen Rückgabewert schreiben kann.
2. Über Verweise: Dabei wird die CGI-Datei in das href eines Links reingeschrieben. Wenn dieser Link betätigt wird, wird das CGI-Programm ausgeführt.  
**<a href="/cgi-bin/statistik.pl"> Hier zur Statistik </a>**
3. Über Grafikreferenzen: Funktioniert so wie Link, nur, dass vom CGI-Programm als Ergebnis ein GIF oder JPEG geliefert werden muss. Das eignet sich besonders gut für Bildbasierte Zähler.  
****
4. Über Server Side Include: Damit lässt sich sehr gut dynamischer Text einbinden, z.B. einen textbasierten Zähler.  
**<!-- #exec cgi="/cgi-bin/text\_counter.pl" -->**
5. Über automatisches Laden im <meta>-Tag: z.B. eine Willkommensmeldung  
**<meta http-equiv="refresh" content="0" URL="/cgi-bin/welcome.pl" >**

## 4. PHP

### 4.1 Technologie

PHP (steht heute für: **PHP: Hypertext Preprocessor**, früher für: **Personal Home Page Tools**) wurde ursprünglich nur für die Programmierung von dynamischen Web-Seiten entwickelt. 1995 als Rasmus Lerdorf PHP entwickelt hat, war PHP nichts anderes als eine Ansammlung von Perl-Skripten. Im Laufe der Jahre wurde es immer weiter entwickelt und mit der Umsetzung in C wurde PHP leistungsfähiger als Perl.

PHP ist eine interpretierte Scriptsprache, d.h. der Code wird nicht für Betriebssysteme kompiliert, sondern erst bei der Ausführung des Codes interpretiert.

Es gibt zahlreiche Möglichkeiten PHP einzusetzen, dazu müssen entsprechende Module für PHP installiert werden. Installiert man den Pop-Interpreter auf dem System, so kann man PHP-Skripte ablaufen lassen.

Doch das wichtigste Einsatzgebiet von PHP ist in einem Webserver. Für die meisten sich auf dem Markt befindlichen Web-Server kann man eine Pop-Unterstützung installieren.

So bekommt der Web-Server die Fähigkeit, PHP selbst und ohne CGI-Schnittstelle auszuführen.

Der Referenz-Web-Server zur Ausführung von PHP-Skripten, ist der Apache HTTP-Server. Dieser steht kostenfrei zur Verfügung und wird meist auf Linux mit der Datenbank MySQL eingesetzt. Diese Konstellation ist unter dem Namen LAMP (**L**inux **A**pache **M**ySQL **P**HP) bekannt und hat den CGI-Standard von der Spitzenposition der Internetprogrammierung abgelöst.

Da PHP im Web-Server ausgeführt wird, werden anders als CGI-Skripte PHP-Skripte nicht ausgelagert, sondern befinden sich im HTML-Code. In speziellen `<?php ... ?>`-Tags steht dann der PHP-Code zwischen normalem HTML-Code, was den Einstieg in die Internetprogrammierung einfach macht. Seiten, in denen PHP-Code vorkommt, bekommen die Endung `.php`, im Web-Server ist diese Endung für den PHP-Interpreter vorkonfiguriert, so dass er PHP-Dateien selbstständig an den PHP-Interpreter schickt.

Folgende Grafik soll die Funktionsweise von PHP-Anfragen an den Web-Server verdeutlichen:

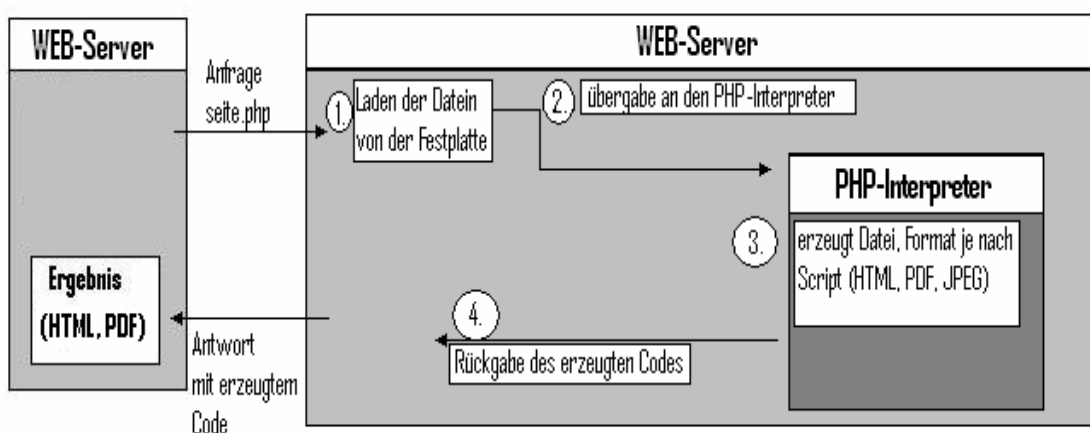


Bild 5

## 4.2 Syntax

Die PHP-Datei eines „Hallo Welt“-Programms in PHP sieht so aus:

```
<HTML><HEAD>
    <TITLE>PHP-Hallo-Welt</TITLE>
</HEAD>
<BODY>    <?php echo "Hallo Welt"; ?>
</BODY>
</HTML>
```

Das Ergebnis, welches zurückgeliefert werden würde, sieht ähnlich aus, nur steht statt

```
<?php echo „Hallo Welt“; ?>    einfach Hallo Welt.
```

Der an den Browser zurückgelieferte Code enthält keinerlei PHP-Elemente mehr, so dass eine Kompatibilität zu allen Browsern, die HTML verstehen, gewährleistet werden kann.

Als Scriptsprache wird PHP bei jedem Aufruf neu interpretiert, was die Ausführung von PHP-Scripten verlangsamt. Bei umfangreichen Projekten kann dies zu ernsthaften Performanceeinbrüchen führen. Abhilfe schafft hier ein vorinterpretierter Code.

Anders als bei Java sind Vorinterpretierer (Caches) bei PHP kein Standard, so dass man existierende Caches wie eAccelerator von Hand einbinden muss.

Ähnlich wie Perl hat sich auch PHP erst zu einer ernsthaften Sprache entwickeln müssen.

Im Jahr 2000, mit der Version 4.0, hat PHP einen neuen Kern bekommen, die so genannte Zend Engine. Es wurde ein einfaches objektorientiertes Modell hinzugefügt und eine Sessionverwaltung, letzteres ist wesentlich für Internetprogrammierung.

2004 mit der Version 5.0 wurde die objektorientierte Programmierung ernsthaft möglich, des Weiteren existierte ab diesem Zeitpunkt eine Ausnahmebehandlung.

PHP enthält eine große Zahl von Erweiterungen, mit dem PEAR (**P**HP **E**xtension and **A**pplication **R**epository) steht ein Klassenarchiv zur Verfügung, mit dem sich ganze Teilaufgaben, wie z.B. die Datenbankbindung oder Authentifizierung erledigen lassen.

Der im PEAR enthaltene PHP-Code folgt klaren, allgemeinen Richtlinien und wird ständig durch ein Gremium von PHP-Entwicklern überwacht.

PHP ist typenlos, d.h. Typen werden nicht vom Programmierer, sondern erst zur Laufzeit bestimmt. Das macht es leichter PHP-Scripte zu schreiben, aber auch leichter Fehler reinzuprogrammieren. Mehr dazu in Kapitel 5 beim abschließenden Vergleich.

## 5. Java Servlets

### 5.1 Die Servlet Technologie

Die Servlet Technologie bildet die Grundlage zur Entwicklung von serverseitigen Web-Anwendungen unter Benutzung der Programmiersprache Java. Wenn man mit Java serverseitige Anwendungen entwickeln will, ist es unumgänglich sich mit ihr auseinanderzusetzen, da auch die Java Server Pages (JSP) auf dieser Technologie aufbauen.

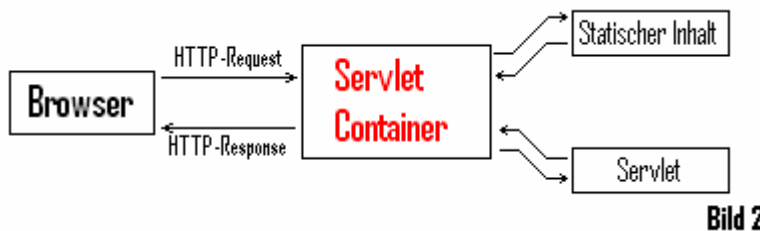
Die Servlet Technologie wurde 1996 von Sun Microsystems vorgestellt, im wesentlichen bietet diese Technologie die Erstellung von Java Klassen, die dynamisch von speziellen Web Servern geladen und ausgeführt werden können.

### 5.2 Servlet Container

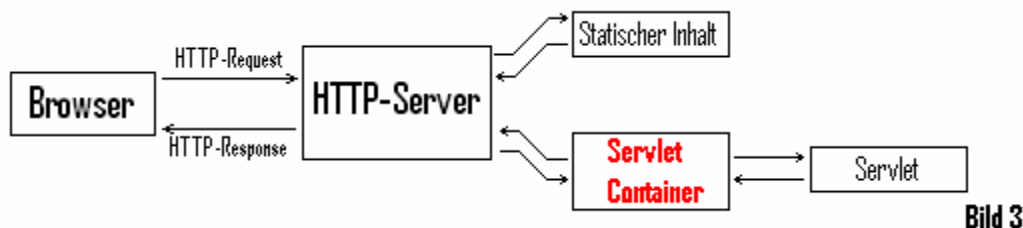
Ein Servlet Container, früher auch als Servlet Engine bezeichnet, ist ein Web Server, der Java Servlets und Java Server Pages ausführen kann. Zurzeit ist eine große Menge von Servlet Containern erhältlich, der berühmteste und von Java als offiziell anerkannter Servlet Container ist Tomcat. Tomcat wurde ursprünglich von Sun Microsystems entwickelt.

1999 wurde er an die Apache Software Foundation übergeben und in das Apache Jakarta Projekt einbezogen. Zurzeit liegt Tomcat in der Version 5.5.9 vor.

Wie oben erwähnt, kann ein Servlet Container, da eigenständiger Web Server, auch als solcher eingesetzt werden. Dabei liefert er die dynamisch zusammengesetzten Inhalte aus den Servlets, so wie die statisch vorliegenden Inhalte, HTML-Dateien.



Der Client sendet Anfragen an den Web Server, dieser liefert statische Inhalte: HTML-Dateien. Ebenso führt er Servlets aus und generiert somit dynamisch Inhalte. Da der Servlet Container speziell für Servlets ausgelegt ist, kann er diese auch schnell, aber statischen Inhalt langsamer als robustere HTTP Server liefern und wird deshalb in üblich verwendete Web Server eingebunden, so wie in den Apache Web Server oder den Microsoft Internet Information Services. Dabei übernimmt der Servlet Container lediglich die Aufgabe, Servlets auszuführen. Um den statischen Inhalt kümmert sich der HTTP Server, siehe Bild 3.



### 5.3 Servlets API

Ein Servlet wird das erste Mal geladen, wenn es das erste Mal benötigt wird, danach bleibt es im Speicher, so lange die Anwendung läuft oder der Servlet Container in Speichermangel gerät. Für die Servletentwicklung bietet Java die Spezifikation Application Programming Interface (API). Diese besteht aus zwei Paketen: javax.servlet und javax.servlet.http. Die so genannte Java Servlet API befindet sich zur Zeit in der Version 2.4 und ist die Grundlage für die Entwicklung von Servlets. Das Paket javax.servlet enthält die grundlegenden Klassen und Schnittstellen, das Paket javax.servlet.http dagegen enthält fortgeschrittenere Klassen und Schnittstellen. Es ist möglich mit beiden zu entwickeln. Wenn man Klassen auf „grüner Wiese“ bauen möchte, empfiehlt es sich das javax.servlet Paket zu nehmen, möchte man vermehrt fertige Funktionen der Servlet API einsetzen, kann man auf das javax.servlet.http Paket zurückgreifen.

### 5.4 Servlet Aufbau

Die Schnittstelle javax.servlet.Servlet ist die Oberklasse aller Servlets. Jedes Servlet muss diese Schnittstelle direkt oder indirekt implementieren. Der Lebenszyklus eines Servlets wird von drei Methoden bestimmt: init(), service() und destroy().

Die **init()**-Methode wird vom Servlet-Container direkt nach der Instanzierung des Objektes aufgerufen und muss vollständig durchlaufen, bevor Anfragen an das Servlet gesendet werden können. Diese Methode wird benutzt, um einmalige Aktionen durchzuführen, die notwendig für ein Servlet sind, jedoch nicht bei jedem Aufruf dessen durchgeführt werden sollen, so z. B. das laden von Datenbankteibern.

Die **service()**-Methode wird bei jedem Aufruf des Servlets aufgerufen. Sie enthält daher Programmcode, der jedes mal ausgeführt werden soll und HTTP-Anfragen verarbeiten kann. Da ein Servlet Multi-Thread fähig ist, ist es wichtig darauf zu achten, dass der Zugriff auf alle Ressourcen synchronisiert abläuft.

Die **destroy()**-Methode wird vom Servlet-Container aufgerufen bevor er das Servlet aus dem Service nimmt. Dies geschieht meistens wenn der Servlet-Container heruntergefahren wird oder zusätzlichen Speicherplatz braucht. Die destroy()-Methode säubert alle Ressourcen, gibt Speicher, Zeiger auf Dateien (Filehandle) usw. frei.

Servlets lassen keinen statischen HTML-Code zu. Jegliche Ausgabe erfolgt über den PrintWriter. Will man lediglich „Hallo Welt!“ auf dem Browser ausgeben lassen, muss man folgendes in eine Klasse hineinschreiben:

```
import javax.servlet.*;
import java.io.PrintWriter;
import java.io.IOException;

public class MyServlet implements Servlet {

    /** Konstruktor */
    public MyServlet() {
    }
    public void init(ServletConfig config) throws ServletException{
    }
    public void destroy() {
    }
    public String getServletInfo() {
        return null;
    }
    public ServletConfig getServletConfig(){
        return null;
    }
}
```

```

public void service(ServletRequest request, ServletResponse response)
throws ServletException, IOException{
    PrintWriter out = response.getWriter();
    out.println("<HTML>");
    out.println("<HEAD>");
    out.println("<TITLE> MyServlet </TITLE>");
    out.println("</HEAD>");
    out.println("<BODY>");
    out.println("Hallo Welt!");
    out.println("</BODY>");
    out.println("</HTML>");
}
}

```

Die ersten vier Methoden müssen übernommen werden, da diese von der Schnittstelle `javax.servlet.Servlet` vorausgesetzt werden. Die Methode `service` führt die Schreiboperationen aus. Diese Klasse hat lediglich das Ziel „Hallo Welt!“ auf dem Browser auszugeben. Für sehr einfache Anwendungen scheint die Servlet- Technologie zu umfangreich und zu mächtig. Gerade auch deswegen hat Sun 1999 die Java Server Pages vorgestellt. Diese Technologie soll es Entwicklern einfacher machen, statischen und dynamischen Code zu mixen und somit einfacher in die dynamische Internetprogrammierung einzusteigen.

## 6. JSP

### 6.1 Technologie

Die Java Server Pages (im weiteren JSP genannt) sollten als eine Erweiterung der Servlet- Technologie verstanden werden, da die JSP auf Basis der Servlets gebaut wurden und die Servlet- Technologie zum Ablaufen brauchen.

Mit den JSP sollten die Nachteile der Servlet- Technologie beseitigt werden. Dem Entwickler wird ermöglicht, statischen Inhalt nebst Java-Code zu schreiben. So dass eine Trennung von statischen Inhalten wie z.B. das Design und den dynamischen Inhalten erfolgen kann. Erfolgt ein Aufbau der JSP-Seite klar strukturiert, ist es durchaus möglich das Design jemandem zu überlassen, der keine Erfahrung mit Java hat.

Eine JSP- Seite sieht auf den ersten Blick aus wie jede andere HTML-Seite, mit dem Unterschied, dass sie die Endung .jsp hat und in speziellen JSP- Tags (<% ... %>) Java-Code steht, der erst vom Servlet Containers ausgeführt werden muss.

Beim ersten Aufruf einer .jsp- Seite wird der gesamte Inhalt der Seite übersetzt und vom JSP- Compiler des Servlet Containers kompiliert. Es entsteht eine Servlet- Klasse, die vom Servlet Container ausgeführt werden kann. Der statische Inhalt, sowie der Java-Code werden in eine Servlet- Klasse übersetzt und kompiliert, sodass weiteres verarbeiten der JSPs dem der Servlets entspricht. So wie die Servlets haben auch die JSP eine API, diese besteht ebenfalls aus zwei Paketen: javax.servlet.jsp und javax.servlet.jsp.tagex. Desweiteren werden die oben bereits erwähnten Servlet- Pakete benötigt. Das Paket javax.servlet.jsp beinhaltet zwei Schnittstellen und vier Klassen. Die Schnittstelle JspPage muss von allen JSP-Servlet-Klassen implementiert werden. Das erinnert an die Schnittstelle javax.servlet.Servlet, die von allen Servlets implementiert werden muss, was aber keine Überraschung ist, da javax.servlet.jsp.JspPage die Schnittstelle javax.servlet.Servlet implementiert.

### 6.2 JSP- Syntax

Die JSP-Syntax ist mit anderen Scriptsprachen vergleichbar. Sie besteht aus statischem HTML-Code, der mit dynamischem Java-Code in <% .. %>- Tags angereichert ist.

Demzufolge sieht die einfachste Ausgabe von „Hallo Welt!“ mit JSP folgendermaßen aus:

```
<HTML>
  <HEAD>
    <TITLE> MyJSP </TITLE>
  </HEAD>
  <BODY>
    <% out.print("Hallo Welt!"); %>
  </BODY>
</HTML>
```

Die Syntax von JSP bietet viele Möglichkeiten, Java in den HTML-Code einzubinden.

Eine dieser Möglichkeiten sind die sogenannten Direktiven. Direktiven dienen dazu, die Transformation der JSP- Seiten zum Servlet zu erleichtern.

Diese Direktiven beinhalten Anweisungen für den JSP-Compiler. Die Aktuelle Version beinhaltet 5 Direktiven, im Einzelnen „include“, „page“, „taglib“, „attribute“ und „tag“.

Sie alle haben folgende Syntax:

```
<%@ param="value" %>
```

Im Folgenden wird die Funktionalität der Direktiven anhand von zwei Beispielen erläutert.

**<%@ include file="fileToInclude" %>** weist den Compiler an den Inhalt der Datei "fileToInclude" an dieser Stelle in die Originaldatei zu kopieren und eignet sich gut um Seitenelemente wie z.B. das Menü auszulagern und so anderen Dateien zur Verfügung zu stellen.

Die Direktive „page“ hat im Gegensatz zu include mehrere mögliche Parameter z.B. import: import wird benutzt um Bibliotheken wie in Javaklassen einzubinden.

**<%@ page import="java.util.List,java.util.Iterator" %>**

Ein weiteres Element der JSP- Syntax sind Aktionen. JSP-Aktionen sind XML-Tags, die Webserverfunktionalität einbinden. Sie haben folgende Syntax:

```
<jsp:include page="mycommon.jsp">  
  <jsp:param name="extraparam" value="myvalue"/>  
</jsp:include>
```

Die Aktion jsp:include bietet die komfortable Möglichkeit, Dateien einzubinden und diesen vorher Variablen zu übergeben.

Eine Beschreibung sämtlicher Direktiven und Aktionen würde zu weit führen und wird für Anfänger nicht zwingend benötigt.

## 7. ASP

### 7.1 Technologie

**ASP** (Active Server Pages) ist keine Programmiersprache sondern eine von Microsoft entwickelte Technologie für interaktive Internetseiten. ASP wird auch als Framework bezeichnet, also eine Ansammlung von Programmen und Dateien für die Web-Entwicklung. Um ASP auszuführen, wird ein Web-Server benötigt, der ASP unterstützt.

Das Referenzmodell hierbei ist der IIS (Internet Information Services) von Microsoft.

ASP läuft, da von Microsoft entwickelt, auch bevorzugt auf Microsoft Betriebssystemen ab.

Dabei läuft ASP als ein Modul des Servers, das ASP-Seiten auswertet und an den Web-Server weiterreicht.

Es gibt jedoch Möglichkeiten ASP auch auf Linux zu portieren, wenn ein Apache Web-Server verwendet wird.

ASP-Seiten haben die Endung `.asp` und beinhalten Scriptelemente zwischen HTML-Code. Die Reihenfolge wann ASP und wann HTML-Code kommen soll ist nicht festgelegt. ASP-Code wird dabei in `<% ... %>`-Tags eingefasst. Als Scriptsprache werden VBScript oder JavaScript verwendet, standardmäßig wird VB Script verwendet. Weitere einsetzbare Sprachen für ASP sind Perl, Python oder REXX. Diese Scriptsprachen sind kein Teil von ASP, sondern sollten eher als Plugins verstanden werden.

Da es sich jedoch um Scriptsprachen handelt, müssen diese bei jedem Aufruf der Seite zeilenweise interpretiert werden.

Die eigentlichen Sprachelemente von ASP bestehen wesentlich aus fünf Objekten:

<b>Objekt</b>	<b>Funktion</b>
1. Request	Informationen abfragen
2. Response	Informationen ausgeben
3. Server	Tool-Sammlung
4. Application	Globale Variablen einer Web-Applikation
5. Session	Globale Variablen eines Web-Besuchers

Mit Hilfe dieser Objekte ist es möglich, mit einer Scriptsprache auf sämtliche Daten des Web-Servers, Formulare und Sessionverwaltung zuzugreifen.

### 7.2 Syntax

Neben normalen `<% .. %>`-Tags existieren in ASP die `@`-Direktiven. Sie sind Zustands- und Sprachunabhängig und fangen immer in der ersten Zeile an. Mit diesen `@`-Direktiven lässt sich z.B. die verwendete Scriptsprache festlegen oder Sessionverwaltung für bestimmte Seiten festlegen.

```
<% @ Language="VBScript"  
    EnableSessionState = false %>
```

Bevor dann normaler Scriptcode kommt, muss die `@`-Direktive mit einem `%>`-Tag geschlossen werden.

Eine weiteres Werkzeug zur Strukturierung der Dateien bietet das SSI (Server Side Include), mit dem sich auch ASP-Dateien an entsprechender Stelle hinzufügen lassen.

```
<!-- #include file="~\include\incDatei.asp" -->
```

## 8. ASP.NET

### 8.1 ASP = ASP.NET ?

Als Nachfolger von ASP hat Microsoft mit **ASP.NET** eine echte Weiterentwicklung gebracht. Dabei hat ASP.NET mit seinem Namensvorgänger nicht mehr viel zu tun, der gravierendste Unterschied zu ASP ist, dass ASP-Seiten noch zeilenweise vom Interpreter abgearbeitet werden, wohingegen ASP.NET-Anwendungen am Server kompiliert und somit nicht mehr interpretiert werden müssen. Deswegen werden bei ASP.NET „richtige Programmiersprachen“ wie z.B. C#, VB.NET oder C++ verwendet und keine Scriptsprachen mehr.

Die Kompatibilität jedoch hat unter der Weiterentwicklung gelitten. ASP.NET kann nur auf Windows-Systemen ab Version 2000 installiert werden, Windows NT und 98 werden nicht mehr unterstützt. Dabei ist es uneingeschränkt und nur auf dem Internet Information Services und dem XSP-Webserver lauffähig. Auf dem Apache HTTP Server ist lediglich eine eingeschränkte Funktionsfähigkeit gewährleistet.

### 8.2 .NET Framework

ASP.NET ist ein Subsystem des **.NET**-Frameworks. Ende der 1990er Jahre übernahm Microsoft die Java-Technologie von SUN und erweiterte sie nach eigenen Bedürfnissen. Dabei geriet die Plattformunabhängigkeit in Mitleidenschaft und SUN unterband dies durch eine Gerichtsverfügung. Um Konkurrenz gegen Java zu bieten, fühlte sich Microsoft genötigt eine eigene Technologie zu entwickeln, die Java entgegengestellt werden konnte. Dabei entstand das .NET-Framework, das man im weitesten Sinne als Programmierumgebung bezeichnen kann.

Was ist genau das .NET-Framework?

Microsoft hat wie gewohnt versucht, alle seine Neuentwicklungen unter einen Namen zu fassen und somit Verwirrung geschaffen. Plötzlich hatten Betriebssysteme, Webdienste usw. die Endung .NET, wobei man bei manchen Produkten von .NET wieder abgerückt ist.

Das .NET-Framework ist ein Rahmenwerk von Klassenbibliotheken, Framework Class Library (FCL) genannt, und Diensten. Des Weiteren enthält es eine Laufzeitumgebung, die Common Language Runtime (CLR). Sprachen, die auf dem .NET-Framework aufbauen, werden nicht mehr systemspezifisch kompiliert, sondern in die Common Intermediate Language (CIL) übersetzt, welche dem Java-Bytecode entspricht. Dieser Code wird von der CLR ausgeführt. Mit Standarten wie Common Type System (CTS) und Common Language Specification (CLS) wird dafür gesorgt, dass Variablentypen und Sprachmerkmale sprachenübergreifend in allen .NET-Sprachen verwendet werden können.

Zu den .NET-Sprachen gehören:

C#	Speziell für .NET entwickelt
C++, Visual Basic	Auf .NET portiert
J#	Portierung der Sprache Java auf die .NET-Technologie

Weitere Sprachen von Drittanbietern werden für das .NET-Framework zur Verfügung gestellt, z.B. Delphi von Borland.

Die Plattformunabhängigkeit bei .NET ist nur theoretisch gegeben, da Microsoft .NET nur für seine eigene Betriebssysteme anbietet. Jedoch im Rahmen des Open-Source-Projektes Mono steht das .NET-Framework, wenn auch nicht vollständig auf Linux zur Verfügung.

Das .NET-Framework als plattformunabhängig zu bezeichnen, ist wegen fehlender Normierung der Klassenbibliothek, sowie lizenzrechtlicher Probleme nicht legitim.

Ein Teilbereich des .NET-Frameworks ist ASP.NET, dieser bietet die Möglichkeit der Erstellung von serverseitigen Webapplikationen. Dabei werden .NET-Sprachen eingesetzt. Das .NET-Framework

stellt HTML-Elemente wie Schaltflächen, Textfelder und Tabellen bereit. ASP.NET-Applikationen haben die Endung .aspx. Diese aspx-Dateien enthalten, wie oben bereits erwähnt, im Gegensatz zu asp-Dateien keinen Scriptcode und werden nicht interpretiert. Sie werden beim ersten Aufruf kompiliert und stehen durch Caching-Verfahren im Speicher bereit, was die Ausführungsgeschwindigkeit erhöht.

### 8.3 Syntax

Im Gegensatz zu ASP, in dem ASP- und HTML-Blöcke sich in der asp-Seite abwechseln konnten, existiert bei ASP.NET eine feste Struktur wie eine aspx-Seite aussehen muss.

1. Am Anfang steht die @Page-Direktive `<%@ Page ... %>`, in der z.B. festgelegt wird, welche Sprache benutzt wird.
2. Es folgt ein Block mit serverseitigem Skriptcode:  
`<script runat="server"> ... </script>`
3. Am Schluss steht der HTML-Code: `<HTML>...</HTML>`.

Durch diese Trennung möchte man mehr Struktur in die Internetprogrammierung einbringen. Dabei können weiterhin Code-Schnipsel in den HTML-Code eingesetzt werden. Eine Möglichkeit hierfür bieten die ASP.NET-Steuerelemente. Sie unterliegen den Regeln von XML und haben die Form:

`<asp:Name Attribut="Wert" ... > ... </asp:Name>`.

Mit diesen Steuerelementen kann man oft alte ASP-`<% .. %>`-Schnipsel ersetzen.

Zusätzlich ist es immer noch möglich Code-Schnipsel in HTML einzufügen. Z.B. sind Wertausgaben wie `<%= wert %>` immer noch möglich.

## 9. Vergleich

Im folgenden Kapitel sollen die zuvor behandelten Technologien miteinander verglichen werden. Dabei sollen insbesondere die Gesichtspunkte Portierbarkeit, Performance und Bedienbarkeit betrachtet werden.

### 9.1 Bedienbarkeit

Web-Seiten sollen nicht nur schnell und sicher funktionieren. Ein gutes Design der Seite ist heute ebenso wichtig geworden. Jedoch sind Menschen, die etwas von Design verstehen, also Grafiker und Designer, selten ausgebildete Programmierer. Deswegen ist es einerseits wichtig den Programmiersprachen-Code vom HTML-Code zu trennen und andererseits den HTML-Code dem Designer leicht zugänglich zu machen.

Dies ist vor allem bei den Java Servlets nicht gegeben, zum Einen wird dort HTML-Code im Java-Code mit „`out.println("<HTML>");`“ auf die Ausgabe geschrieben, zum Anderen werden Servlet-Klasse kompiliert, so dass nach jeder Änderung, egal wie klein sie ist, die Anwendung noch mal kompiliert werden muss.

Perldateien werden zwar nicht kompiliert, enthalten jedoch auch keinen reinen HTML-Code und liegen extern vor.

Besonders gute Bedienbarkeit für Laien bieten jsp, asp, aspx und php Dateien. Diese enthalten Programm-Schnipsel mit HTML-Code vermischt und müssen nicht extra vom Programmierer kompiliert werden.

Was für Laien als besonders geeignet erscheint, kann für erfahrene Programmierer ein Nachteil sein. Keine feste Struktur bei PHP und ASP z.B. führt schnell zu unübersichtlichem „Spaghetti-Code“ der wesentlich schwieriger zu handhaben ist.

Ein wesentlicher Aspekt der Bedienbarkeit für Profianwender sind die Möglichkeiten der Sprachen. ASP.NET bietet eine echte Verbesserung zu seinem Vorgänger, da jetzt mehr Funktionalität bereits in der Klassenbibliothek zur Verfügung steht. In Java stehen ebenfalls zahlreiche Klassen aus der Servlet-API zur Verfügung, was die Entwicklung vereinfacht. PHP dagegen wurde speziell für die Web-Entwicklung geschrieben. Es bietet mit seinen speziell dafür enthaltenen Funktionen wie z.B. die Datenbankbindung, Email-Funktion und ausgereiften Zeichenkettenfunktionen großen Komfort für die Entwickler. Nachteilig ist die Typenlosigkeit von Scriptsprachen. Diese wirkt sich negativ auf große Projekte aus, da dort die Koordination und Kommunikation im Vordergrund stehen.

### 9.2 Portierbarkeit

Idealerweise sollten Web-Anwendungen nicht nur möglichst alle Browser unterstützen, sondern möglichst portierbar sein. Die Portierbarkeit von Web-Anwendungen sollte einen reibungslosen Umzug auf ein anderes Betriebs- oder Server-System erlauben.

Volle Portierbarkeit können z.B. Servlets und JSP-Anwendungen vorweisen. Sie setzen lediglich eine Java Laufzeitumgebung voraus, die jedoch für alle großen Server-Betriebssysteme vorhanden ist. Der Servlet-Container für die Ausführung der Java-Anwendungen kann ebenfalls in die meisten Webserver integriert werden, aber auch selbstständig als Webserver laufen.

PHP als Scriptsprache benötigt ebenfalls lediglich eine Unterstützung des Web-Servers und kann problemlos auf allen Betriebssystemen betrieben werden, auf den der Web-Server läuft. CGI ist als ältester Standard für dynamische Web-Programmierung auf den meisten Web-Servern implementiert. Perl ist auf Linux-Systemen bereits ein Systembestandteil, kann jedoch auch auf Windows-Systemen installiert werden.

ASP und ASP.NET dagegen sind als Microsoft-Entwicklungen, will man sie mit vollen Funktionsumfang betreiben, auch nur auf Microsoft-Systemen lauffähig. Es gibt Open-Source-Projekte, die das .NET-Framework auch auf Linux portieren, jedoch nur auf Kosten des Funktionsumfangs.

### 9.3 Performance

Auf Web-Anwendungen können zeitweise mehrere tausend Anfragen gleichzeitig erfolgen, deswegen ist die Performance der Anwendung für solche Seiten entscheidend. Als CGI die ersten interaktiven Web-Seiten lieferte, war das Internet sehr viel kleiner als heute, deswegen sieht diese Schnittstelle aus heutiger Sicht eher veraltet aus. Bei jedem Zugriff auf eine CGI-Seite wird ein neuer Prozess gestartet, sowie das Perlskript neu interpretiert, das erzeugt einen hohen Betriebssystem-Overhead und wirkt sich negativ auf die Reaktionsgeschwindigkeit der Anwendung aus.

PHP, eine neuere Entwicklung, läuft als ein Modul des Web-Servers. Es wird nicht jedes Mal ein neuer Prozess gestartet aber die PHP-Seite jedes mal neu interpretiert, was sich bei vielen gleichzeitigen Zugriffen ebenfalls negativ auf die Performance der Anwendung auswirken kann. Es gibt jedoch Werkzeuge, die PHP-Seiten vorkompilieren und somit die Performance erhöhen. Die Performance von ASP kann man mit der von PHP vergleichen, da diese ebenfalls vom Web-Server interpretiert wird.

Einen echten Vorteil haben dagegen Technologien, die vorkompilierte Dateien bereitstellen. Java Servlet Anwendungen werden nach Erstellung in den Java-Bytecode vorkompiliert, der bei Ausführung von der Java VM ausgeführt wird. Diese Technologie hat einen echten Performance-Vorteil gegenüber interpretierten Sprachen. Wie oben erwähnt geht dieser Vorteil jedoch auf Kosten der Bedienbarkeit der Anwendung.

Einen Zwischenweg dazu geht die JSP-Technologie: Diese bietet eine höhere Performance als interpretierte Sprachen, da JSP-Seiten nicht bei jedem Aufruf neu interpretiert werden, sondern beim ersten Aufruf in eine Servlet-Klasse umgewandelt und kompiliert werden, aber auch eine bessere Bedienbarkeit als Servlets, da JSP-Seiten nicht vom Programmierer kompiliert werden müssen und HTML-Code einfach hineingeschrieben werden darf.

ASP.NET geht dabei den gleichen Weg, ASP.NET-Seiten werden ebenfalls beim ersten Aufruf in die CIL (Common Intermediate Language) vom Just-In-Time-Compiler der CLR vorkompiliert und bei allen späteren Aufrufen von der CLR ausgeführt.

### 9.4 Resümee

Zu sagen welche Technologie allen anderen überlegen ist, fällt äußerst schwer, denn es kommt sehr auf den Einsatzzweck der Technologie an. Im Internet existieren sehr verschiedene Anwendungen, es fängt bei einfachen Gastbüchern an, geht über komplexe Internet-Shop-Anwendungen und hat ein offenes Ende, da es bereits mit CGI möglich war, „echte“ Programmiersprachen zu benutzen. Dabei ist HTML lediglich eine Oberfläche, unter der jede mögliche Anwendung stehen kann.

Die einzigen Aussagen, die man machen kann, sind Richtungsweisungen für die einzelnen Technologien.

Für CGI/Perl lässt sich sagen, dass es früher als der Standard für dynamische Internetprogrammierung mittlerweile im unteren und mittleren Anwendungsbereich von PHP abgelöst wurde. PHP, als allein für Internetprogrammierung entwickelte Sprache, bietet wesentlich mehr Komfort für den Programmierer. Die Lernkurve für Einsteiger ist sehr viel flacher als bei Perl, Java und Co. Man kann ohne lange Einarbeitungsphasen mit MySQL und Apache sehr schnell Anwendungen erstellen. Viele Funktionen sind sehr mächtig, z.B. gibt es sehr gute Zeichenkettenoperationen und Datenbankanbindungen. Da PHP auf Grund von Benutzbarkeit auf einige Programmierkonstrukte verzichtet wie z.B. auf die Deklaration von Variablen, eignet es sich nicht optimal für größere Projekte.

ASP ist in diesem Punkt PHP ähnlich. Es eignet sich gut für kleinere Projekte, da auch hier die Lernkurve flacher als z.B. bei ASP.NET ist.

Will man jedoch mehr Aufwand in ein Web-Projekt investieren und achtet auf hohe Qualität und robuste Sprachen, ist man mit Java Servlets und JSP beim richtigen Produkt. Die Lernkurve ist wesentlich höher als bei Scriptsprachen, und der Aufwand lohnt sich möglicherweise bei kleineren Projekten nicht.

Die Vorteile einer „echten Programmiersprache“ sind Sicherheit, Robustheit des Programms und weniger Fehler im Code. Diese sind für ein großes Projekt sicher wichtiger, als schnelle Erfolge mit weniger sauberem Code.

ASP.NET ist der JSP-Technologie ähnlich und wird sicher das Mittel der Wahl für Anwendungen auf Microsoft-Systemen darstellen. Microsoft spricht von einer optimalen Skalierbarkeit für alle

Anwendungsfälle, doch wird die Lernkurve sicher für kleiner Applikationen zu hoch sein und die Wahl in diesem Fall auf ASP fallen. Für größere Projekte bietet ASP.NET viele Vorteile, unter anderem die Sessionverfolgung auf Server-Farmen, die es mit ASP noch nicht gab.

## Literaturverzeichnis

Münz S, Nefzger W. HTML & Web-Publishing Handbuch: CGI 1.0/1.1 – Perl: Funktionen und Module – XML und DTDs – XML-Daten darstellen. München: Franzis' Verlag; 2002. 592 Seiten

Schmidt K. PHP 4. Böblingen: C&L Computer und Literaturverlag; 2000. 824 Seiten

Weltner T. Microsoft Active Server Pages lernen und beherrschen. Unterschleißheim: Microsoft Press Deutschland; 1999. 440 Seiten

Kurniawan B. Java for the Web with Servlets, JSP, and EJB: A Developers' s Guide to Scalable J2EE Solutions. New Riders Publishing; 2002 953 Seiten

Wikipedia, Common Gateway Interface,  
[http://de.wikipedia.org/wiki/Common\\_Gateway\\_Interface](http://de.wikipedia.org/wiki/Common_Gateway_Interface)  
zuletzt heruntergeladen am: 11.07.2005

Wikipedia, Perl,  
<http://de.wikipedia.org/wiki/Perl>  
zuletzt heruntergeladen am: 11.07.2005

Wikipedia, Java Server Pages,  
<http://de.wikipedia.org/wiki/JSP>  
zuletzt heruntergeladen am: 12.07.2005

Wikipedia, PHP  
<http://de.wikipedia.org/wiki/Php>  
zuletzt heruntergeladen am: 13.07.2005

Wikipedia, .NET,  
<http://de.wikipedia.org/wiki/.NET>  
zuletzt heruntergeladen am: 14.07.2005

Lohrer M. Einstieg in ASP.NET. Galileo Computing; 2002 550 Seiten  
zuletzt heruntergeladen am: 14.07.2005