



Webdesign

FüUstgSBw
First Edition

HF Weidinger
L Siegerth

Inhaltsverzeichnis

1 Grundlagen	4
1.1 Links	5
1.2 Coding Guidelines	5
1.2.1 CSS	5
1.2.2 HTML	5
1.3 Entstehung des Internet	7
2 HTML	8
2.1 Grundlagen HTML	9
2.2 Inhalte in HTML	12
2.2.1 Tables	13
2.2.2 Headings	19
2.2.3 Paragraphs	20
2.2.4 Images	22
2.2.5 Links	23
2.2.6 Formulare	24
2.2.7 Textformatierung	28
2.2.8 Listen	28
2.3 Struktur in HTML	29
2.3.9 Blockelemente vs. Inline Elemente	29
2.3.10 Semantische Strukturierung	31
3 CSS	33
3.1 CSS - Basics	36
3.1.11 Schriften	37
3.1.12 Borders (Simple)	37

3.1.13 Farben	39
3.1.14 Außen und Innenabstände	40
3.2 Layouting with CSS	42
3.3 Advanced CSS Techniques	42
3.3.15 text-shadow	42
3.3.16 box-shadow	44
3.3.17 vendor-prefixes	44
4 WEBSERVER	45
4.1 HTTP und das Web	45
4.1.18 Webserver und Protokolle	45
4.2 Apache	48
4.2.19 Virtual Hosting	48
4.2.20 Einführung und Installation	57
4.2.21 Konfiguration	61
4.2.22 Sicherheit	65
4.2.23 Indexes	72
4.2.24 SSI	79
4.2.25 CGI	82
4.2.26 PERL mit mod_perl.so	86
4.2.27 Apache und PHP	89
4.2.28 Die Log-Dateien des Apache	91
4.2.29 Secure Socket Layer (SSL)	94
4.3 Tomcat	98
4.3.30 Der Apache Tomcat - Ein Applikationsserver	98
4.3.31 Architektur des Tomcat	100
4.3.32 Installation	100
4.4 Internet Information Services	101
4.4.33 Herkunft und Installation der IIS	101
4.4.34 Datensicherung	104
4.4.35 Remoteverwaltung	105
4.4.36 Authentifizierung und Autorisierung	106
4.4.37 Bereitstellen von Webseiten	110

1 Grundlagen

Weitere Informationen zur Entstehung des Internet

Informationen zu TCP/IP und Weiteres zur Entstehung des Internets, sind in der Tour 2 des Einstiegskapitels von SELFHTML zusammengefasst.

Erarbeiten Sie sich die in Tour 2 bei SELFHTML aufbereiteten Informationen zum Thema Grundlagen des Internets und betrachten Sie sie als Hintergrundwissen bzw. Basis der folgenden Kapitel. [17, SELFHTML - Tour 2]

Benötigte Software

- Ein einfacher Texteditor (Notepad) - empfohlen wird Notepad++, da umfangreicher und trotzdem Open-Source
- Ein Browser, um seine Werke betrachten zu können - IE, Firefox, Chrome, Opera, etc...

Warum überhaupt HTML, CSS etc... Ich kann doch auch in C++ programmieren

Der größte Vorteil in der Programmierung mit HTML, CSS, und co. ist, dass Sie absolut Plattform unabhängig sind.

Eine Webseite, die in validem HTML mit validem CSS geschrieben ist, wird auf jedem Gerät gleich dargestellt. Egal ob Sie einen PC mit Windows, einen Mac mit OSX, ein Android mit Chrome oder ein iPad mit Safari nutzen, Sie haben immer exakt das gleiche Ergebnis. Sofern Sie sich an die Standards halten. Der eigentlich noch bessere Vorteil, ist, dass sie keinerlei Software auf dem Client installieren müssen. Selbst komplexeste Anwendungen, die zusätzlich zu HTML noch Datenbanken und Programmiersprachen wie Ruby, Pearl, Python oder PHP laufen alle serverseitig und übermitteln Ihnen nur reines HTML.

Anwendungen in HTML sind absolut Plattform - unabhängig

1.1 Links

Weiterführende Links

HTML Validator des W3C

1.2 Coding Guidelines

1.2.1 CSS

Kommentare in CSS

Parallel zu HTML gibt es auch in CSS keine Unterscheidung zwischen einzeiligen und mehrzeiligen Kommentaren.

Kommentare werden in CSS mit `/*` eingeleitet und mit `*/` wieder beendet

Listing 1.1

```

1 h1 {
2   color:green;
3   /* Kommentar in CSS*/
4
5   /******
6   Optisch hervorgehobener Kommentarblock
7   *****/
8 }
```

CSS

1.2.2 HTML

Einfache Kommentare

Wie in einer normalen Programmiersprache bietet auch HTML die Möglichkeit den Quellcode zu kommentieren.

Die eingebundenen Kommentare werden dann vom Browser dann nicht im gerenderten Frontend angezeigt

Listing 1.2

```

1 <!-- Dies ist ein Kommentar -->
2
3
```

HTML

Mehrzeilige Kommentare

HTML unterscheidet nicht zwischen ein- und mehrzeiligen Kommentaren.

Ein Kommentar Absatz wird daher genauso erstellt wie ein einzeiliger Kommentar.

Listing 1.3

```
1  <!--  
2      Dies ist ein mehrzeiliger Kommentar  
3  
4      Zweite Zeile des Kommentars  
5  -->
```

HTML

Einrückungen

In HTML sollten Tags, die innerhalb eines anderen Tags liegen, immer um 4 Zeichen, bzw. eines Tabs eingerückt werden. Es empfiehlt sich in der Praxis, wenn Sie mit mehreren Personen arbeiten eine Einrückung um 4 Zeichen. Diese werden immer genau gleich dargestellt. Die Breite eines Tabs kann von System zu System variieren und somit könnte Code auch optisch gesprengt werden.

Einrückungen am Beispiel einer unordered list:

Listing 1.4

```
1  <ul>  
2      <li>Listenelement 1</li>  
3      <li>Listenelement 2</li>  
4      <li>Listenelement 3</li>  
5  </ul>
```

HTML

Einrückungen bei einer ganzen HTML Seite

Innerhalb eines HTML - Dokuments hat es sich als praktisch erwiesen, erst ab den Elementen head bzw. body einzurücken. Würde man schon body und html auch einrücken, wäre man beim ersten Inhalt schon bei einer Tiefe von 8 Zeichen.

Listing 1.5

```

1  <!doctype html>
2  <html>
3
4  <head>
5
6      <title>Titel der Seite</title>
7
8  </head>
9
10 <body>
11
12     <div class="div1">
13         <h1>Überschrift 1</h1>
14
15         <ul>
16             <li>Element 1</li>
17             <li>Element 2</li>
18         </ul>
19
20     </div> <!-- end of DIV1 -->
21
22 </body>
23
24 </html>

```

HTML

coding is art

Im Gegensatz zu z.B. der Programmiersprache **Python** hat man im Quellcode von HTML fast keinerlei Vorgaben, was Zeilenumbrüche oder Einrückungen angeht. Es ist theoretisch möglich, und auch teilweise praktisch vollzogen, den kompletten Inhalt einer HTML - Datei in eine Zeile zu schreiben. In der Praxis wird dies auf einigen Seiten durchgeführt. Selbst ein Leerzeichen nimmt Speicherplatz ein - auch wenn nur gering. Es gibt jedoch mehrere Gründe, warum ein sauberer Code mehr Vorteile hat:

- Eine eventuelle Fehlersuche geht später schneller
- Nachträgliche Änderungen und Hinzufügen von Informationen ist problemlos
- Sie selber sehen Ihre Fehler schneller
- Der Hörsaalleiter sieht gemachte Tippfehler schneller und kann Ihnen schneller helfen

Ein Negativbeispiel

So soll es nicht aussehen:

Listing 1.6

```

1  <ul><li>erster Aufzählungspunkt</li><li>zweiter
2  Aufzählungspunkt</li><li>dritter Aufzählungspunkt, der
3  mehr Platz benötigt</li><li>nach Ende keine
4  Einrückung mehr</li></ul><div><p>Absatz in p in div</p></div>

```

HTML

1.3 Entstehung des Internet

2 HTML

Was ist HTML

HTML ist die Auszeichnungssprache für die Inhalte einer Webseite. Mittels HTML wird die Strukturierung von Inhalten wie Texten, Bildern, Hyperlinks oder anderen Ressourcen vorgenommen.

HTML-Dokumente .html .htm bilden die Grundlage des World Wide Web WWW. Es ist möglich in HTML nicht nur die sichtbaren Inhalte zu strukturieren, sondern auch zusätzliche Informationen, wie zum Beispiel den Autor, das Datum oder eine Beschreibung der Seite zu hinterlegen.



Die Definitionen, wie die Sprache aufgebaut ist und auch die Weiterentwicklung unterliegt dem World Wide Web Consortium W3C.

Derzeit ist die aktuelle Version HTML5

Das Acronym HTML steht für HyperText Markup Language

Websites, Homepages, Webpages, etc....

Die oft gehörten Begriffe

- Website,
- Homepage,
- Webpage,
- Internetseite,
- HTML-Seite und
- Webauftritt



stellen im groben Sinne Synonyme dar.

Im normalen Sprachgebrauch kann und wird zwischen diesen Begriffen nicht unterschieden. Im engeren Sinne lassen sich jedoch kleinere Unterschiede feststellen. Beispiel: Ein **Webauftritt** kann aus mehreren einzelnen HTML-Seiten bestehen. Beispiel 2: Korrekt betrachtet ist eine **Homepage** nur eine einzelne Seite. Erst wenn es mehrere Pages sind, kann von einer Site gesprochen werden.

Da umgangssprachlich diese Abgrenzungen jedoch nicht berücksichtigt werden, ist dies relativ irrelevant

2.1 Grundlagen HTML

Versionen von HTML

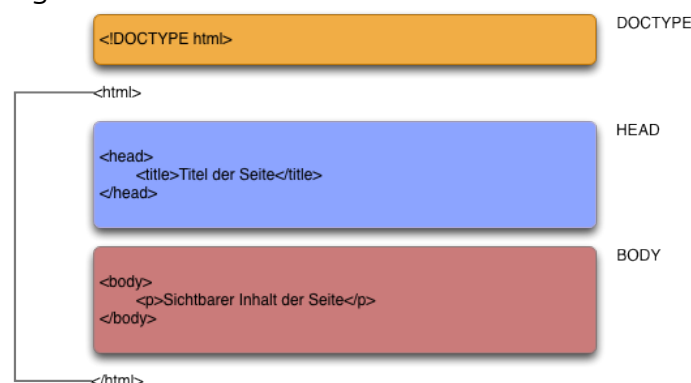
Die Standards für HTML werden durch das World Wide Web Consortium W3C entwickelt und veröffentlicht.



Version	Erscheinungsdatum	Merkmale
HTML	03.11.1992	Initial Release
HTML2	11.1995	Formulare werden eingeführt
HTML3.2	01.1997	Tabellen, Textfluss um Bilder, Einbindung von Applets
HTML4	12.1997	Stylesheets, Skripte und Frames
XHTML 1.1	2001	Eine Neuformulierung von HTML 4.01 mit Hilfe von XML
HTML 5	04.2009	

HTML - Struktur

Eine HTML Datei besteht grundsätzlich aus 3 Bereichen:



1. der Dokumenttypdeklaration (**Doctype**) ganz am Anfang der Datei, die die verwendete Dokumenttypdefinition DTD angibt, z. B. HTML 4.01 Strict,
2. dem HTML-Kopf (**HEAD**), der hauptsächlich technische oder dokumentarische Informationen enthält, die üblicherweise nicht im Anzeigebereich des Browsers dargestellt werden
3. dem HTML-Körper (**BODY**), der jene Informationen enthält, die gewöhnlicherweise im Anzeigebereich des Browsers zu sehen sind.

Damit gelangen wir zu folgender Struktur:

Listing 2.1

```

1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/st... dtd">
2  <html>
3
4      <head>
5      </head>
6
7      <body>
8      </body>
9
10 </html>

```

HTML - HEAD

Der head - Bereich einer HTML-Datei dient der Übergabe von zusätzlichen Daten, die nicht direkt zum anzuzeigenden Inhalt gehören. Diese Informationen können zum Beispiel externe Ressourcen wie CSS - oder Javascript-Dateien sein. Oder es können Informationen über den Author, das Erstellungsdatum, oder eine Beschreibung sein.

Mögliche Elemente im head

Element	Bedeutung	Anwendung
title	TITEL	Der in den Tabs eines Browsers angezeigte Titel der Seite
meta	ZUSATZINFORMATIONEN	z.B. über den Author der Seite oder das Veröffentlichungsdatum
base	BASIS-URI oder -Frame	
link	Verknüpfung zu einer Ressource	z.B. externe Stylesheets
script	CODE einer anderen Programmiersprache	hauptsächlich für die Einbindung von Javascript genutzt
style	Layout - Eigenschaften	
object	externe Dateien	

das folgende Listing zeigt einen beispielhaften head eines üblichen HTML - Dokuments

Listing 2.2

```

1  <head>
2
3
4      <meta charset="utf-8">
5      <meta name="description" content="Development Framework">
6      <meta name='author' content='Kevin Siegerth'>
7
8
9      <title>ITSysAdminFwWebSK</title>
10
11
12     <link rel="stylesheet" media="screen" href="/css/style.css" >
13
14     <script type="text/javascript" src="etc/markItUp.js"></script>
15
16     <style type="text/css" media="print,screen"><!--
17         h1 {
18             color:green;
19         }
20     --></style>
21
22 </head>
```

HTML - BODY

Der body eines HTML - Dokuments beinhaltet die Daten, die im Browser angezeigt werden sollen.

Er beinhaltet somit alle Elemente wie:

- Links
- Tabellen

- Bilder
- Texte
- etc...

Das folgende Listing zeigt einen beispielhaften Body für eine einfache HTML Seite.

Listing 2.3

```

1  <body>
2
3  <h1>Willkommen auf meiner Webseite</h1>
4
5  <h2>Hier habe ich eine Liste für euch erstellt</h2>
6
7  <ul>
8      <li>Links</li>
9      <li>Tabellen</li>
10     <li>Bilder</li>
11     <li>Texte</li>
12     <li>etc...</li>
13 </ul>
14
15 <h2>Und hier noch eine kleine Tabelle</h2>
16
17 <table>
18     <tr>
19         <td>Eintrag</td>
20         <td>1</td>
21     </tr>
22     <tr>
23         <td>Eintrag</td>
24         <td>2</td>
25     </tr>
26 </table>
27
28 </body>

```

HTML

HTML - Doctype

Der DOCTYPE in HTML dient nur dem Zweck, den Rechnern, die diese Seite öffnen, eine Vorinformation zu geben. Der Computer kann sich dann direkt darauf einstellen, welche Tags folgen können und wie er diese interpretieren muss. Ein normaler Webbrowser wie Google Chrome, Firefox, Safari oder auch Opera (**NEIN**, der Internet Explorer IE ist kein **normaler** Browser) ignoriert diese Angabe gänzlich. Diese Angaben haben erst eine Relevanz, wenn automatisierte Parse-Roboter (Crawler) die Seite öffnen. Ein Crawler ist ein Programm, das eine Webseite abrufen und die ausgelesenen Inhalte verarbeitet. Im Großteil der Fälle dient der Abruf der Seiten der Indizierung der Inhalte für schnellere Suchanfragen.

Um diesen Vorgang noch zu verbessern, gibt es neue Techniken und Möglichkeiten um die semantischen Angaben einer Seite durch viele Zusatzinformationen zu optimieren. Dies wird erreicht durch zusätzlich neue Tags, die für den menschlichen Betrachter nicht sichtbar sind.

Werden zusätzliche Sprachen also in HTML eingebunden, müssen diese im DOCTYPE deklariert werden, so dass die Maschine weiß, wie Sie damit umzugehen hat.

Beispielhaft verschiedene DOCTYPEs wären:

DOCTYPE

!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"	HTML 4.01
"http://www.w3.org/TR/html4/loose.dtd"	
!doctype html	HTML5
?xml version="1.0" encoding="UTF-8"?>/span>	XHTML + rdfa

Relevant ist in diesem Lehrgang lediglich der DOCTYPE !doctype html. Dieser DOCTYPE gibt an, dass das folgende Dokument in HTML5 geschrieben wurde

Grundbegriffe für HTML

Tag

Was ist nun ein HTML-TAG? - Als HTML-TAG wird ein einzelner HTML-"Befehl" bezeichnet. Dabei kommt der Begriff TAG aus dem Englischen und hat dort die Bedeutung von "Etikett, Anhänger, Aufkleber, Marke, Kennzeichnung, Auszeichner". Am besten lässt sich es mit dem Begriff "Kennzeichnung" veranschaulichen. Man kennzeichnet Bereiche, dass diese in bestimmter Weise angezeigt werden sollen. Dabei kann man dann definieren: hier beginnt der Bereich und dort hört er wieder auf.

Ein Tag ist ein Markup für einen bestimmten Bereich. Es gibt 3 Arten von Tags:

- öffnende Tags - div
- schließende Tags - /div
- selbstschließende Tags - img/

Element

Ein Bereich der von 2 zusammengehörigen Tags umschlossen wird oder eine selbstschließender Tag ist, nennt man Element.

Ein Element besteht somit fast immer aus den 3 Bereichen:

- öffnender Tag
- Inhalt des Elements
- schließender Tag

Attribut

Ein Attribut kann einem Element zugewiesen werden. Es können einem Element mehrere verschiedene Eigenschaften (Attribute) zugewiesen werden. Ein Element muss aber keine Attribute besitzen.

Die Zuweisung von Attributen an Elemente erfolgt innerhalb des öffnenden Tags.

p class="class_name"

2.2 Inhalte in HTML

TAGS in HTML

Ein Element in HTML wird parallel zu XML meist durch einen öffnenden und schließenden Tag abgegrenzt. In HTML nennen sich diese Begrenzer **tag**. öffnender HTML Tag span schließender HTML Tag /span

daraus ergibt sich die folgende Syntax für z.B. eine Überschrift

h1Überschrift/h1

Listing 2.4

```

1 <h1>Inhalt</h1>
2
3 <span>Inhalt</span>
4
5 <div>Inhalt</div>

```

HTML

Selbstschließende Tags

HTML kennt zu den normalen Tags, die einen Inhalt einschließen, auch selbstschließende Tags. Es handelt sich bei diesen Tags entweder um Tags für die Einbindung von externen Ressourcen oder Funktionselemente wie Buttons.

Ein selbstschließender Tag - wie der Name schon sagt - schließt sich selbst d.h. er benötigt keinen eigenen schließenden Tag. Ein Beispiel hierfür ist der Tag für die Ausgabe eines Bilder oder ein Button zum Abschicken eines Formulars. Der Tag wird innerhalb des öffnenden Tags direkt wieder mit einem Slash "/" geschlossen

img /

Seit **HTML5** ist es nicht mehr nötig die selbstschließenden Tags auch explizit mit dem Slash zu schließen, da durch die richtige Umsetzung der Spezifikation der nachfolgende Inhalt automatisch wieder ausgeschlossen wird. Auf Grund der Einheitlichkeit oder auch Umsetzungsprobleme von den leider immer noch zu stark vertretenen alten Browsern sollte der Tag trotzdem explizit geschlossen werden - **HTML5** stellt trotzdem auch alles richtig dar.

Listing 2.5

```

1 <input type="text" />
2 

```

HTML

2.2.1 Tables

Table

[Englisch]: a table

\ "Tisch\ "

Das wohl erste an das man denkt, wenn man das Wort Table hört, ist wohl ein Tisch. Doch was hat dieser Tisch mit **HTML** zu tun?

Natürlich nichts! Jedoch hat das Wort Table im Englischen 2 Bedeutungen:

- Tisch
- Tabelle

Nachdem es uns wohl im Schwerpunkt nicht darum gehen wird, möglichst viele Bilder von Tischen auf unseren Seiten unterzubringen -- außer wir sind ein Schreiner mit Online-Shop -- wenden wir uns dem Thema **Tabellen in HTML** zu. Tabellen in HTML Das folgende Beispiel zeigt den minimalen Aufbau einer Tabelle.

Tabellen bestehen aus Reihen und Zeilen (rows tr und columns td). Der tr-Tag leitet in diesem Beispiel eine Zeile ein. Innerhalb dieser Zeile erstellen wir mit dem td-Tag eine Zelle. Ein tr kann dabei eine beliebige Anzahl an Zellen haben. Diese werden dann innerhalb der einen Zeile von links nach rechts nebeneinander ausgegeben.

Recherchieren Sie, wofür die Abkürzungen tr, th und td stehen könnten. Es erleichtert es Ihnen auf jeden Fall, sich den Tag zu merken, wenn Sie den vollen Namen kennen.

Listing 2.6

HTML

```

1 <table>
2     <tr>
3         <td>Zelleninhalt</td>
4     </tr>
5 </table>

```

Sinnvolle Tabellen

Das folgende Beispiel stellt nun schon eine etwas sinnvollere Tabelle dar.

Zelleninhalt	Zelleninhalt	Zelleninhalt
Zelleninhalt	Zelleninhalt	Zelleninhalt
Zelleninhalt	Zelleninhalt	Zelleninhalt

Listing 2.7

HTML

```

1 <table>
2     <tr>
3         <td>Zeileninhalt</td>
4         <td>Zeileninhalt</td>
5         <td>Zeileninhalt</td>
6     </tr>
7     <tr>
8         <td>Zeileninhalt</td>
9         <td>Zeileninhalt</td>
10        <td>Zeileninhalt</td>
11    </tr>
12    <tr>
13        <td>Zeileninhalt</td>
14        <td>Zeileninhalt</td>
15        <td>Zeileninhalt</td>
16    </tr>
17 </table>

```

Semantik in Tabellen

im folgenden Beispiel wurde unsere Tabelle um die beiden Element

- thead und
- tbody

ergänzt.

Diese beiden Elemente dienen dazu innerhalb der Tabelle 2 verschiedene Bereiche zu definieren:

- Den Tabellenkopf mit der Beschreibung und
- die Tabelleninhalte

Kopfzelle 1	Kopfzelle 2	Kopfzelle 3
Zelleninhalt	Zelleninhalt	Zelleninhalt
Zelleninhalt	Zelleninhalt	Zelleninhalt
Zelleninhalt	Zelleninhalt	Zelleninhalt

Warum sollte man in Tabellen verschiedene Bereiche definieren können?

- Überlegen Sie sich ein Beispiel aus der Praxis, in dem in einer Tabelle 2 Bereiche vorkommen.
- Ich habe mit CSS die Möglichkeit einzelne Zeilen auch bunt einzufärben, um damit Übersicht zu schaffen. Warum sollte ich also trotzdem noch mittels HTML die Kopfzeile definieren?

Listing 2.8

```

1 <table class="example">
2     <thead>
3         <tr>
4             <th>Kopfzelle 1</th>
5             <th>Kopfzelle 2</th>
6             <th>Kopfzelle 3</th>
7         </tr>
8     </thead>
9     <tbody>
10        <tr>
11            <td>Zelleninhalt</td>
12            <td>Zelleninhalt</td>
13            <td>Zelleninhalt</td>
14        </tr>
15        <tr>
16            <td>Zelleninhalt</td>
17            <td>Zelleninhalt</td>
18            <td>Zelleninhalt</td>
19        </tr>
20        <tr>
21            <td>Zelleninhalt</td>
22            <td>Zelleninhalt</td>
23            <td>Zelleninhalt</td>
24        </tr>
25    </tbody>
26 </table>

```

HTML

TableEnd?

1. Finden Sie heraus ob es auch einen Bereich für Fußzeilen gibt
Falls ja, testen Sie an selbst gewählten Beispielen, wie sich dieser verhält.
2. Denken Sie sich ein eigenes Beispiel aus, warum es solch einen Bereich innerhalb einer Tabelle geben sollte.

Experimente mit Tabellen

Mit Hilfe der Attribute colspan und rowspan ist es möglich, dass sich einzelne Zellen über mehrere Zeilen bzw. auch Spalten erstrecken können. Das folgende Beispiel macht dies deutlich:

Kopfzelle 1	Kopfzelle 2	Kopfzelle 3	
Zelleninhalt		Zelleninhalt	

Kopfzelle 1	Kopfzelle 2	Kopfzelle 3			
Zelleninhalt	Zelleninhalt	Zelleninhalt	Zelleninhalt		
Zelleninhalt	Zelleninhalt	Zelleninhalt	Zelleninhalt		
Zelleninhalt	Zelleninhalt	Zelleninhalt	Zelleninhalt		
Zelleninhalt	Zelleninhalt	Zelleninhalt	Zelleninhalt		
Zelleninhalt	Zelleninhalt				
Zelleninhalt	Zelleninhalt		Zelleninhalt		
Zelleninhalt	Zelleninhalt	Zelleninhalt	Zelleninhalt	Zelleninhalt	Zelleninhalt
Zelleninhalt	A table in a table	Zelleninhalt	Zelleninhalt	Zelleninhalt	Zelleninhalt

- Erarbeiten Sie sich an Hand des Beispiels, wie die Nutzung von colspan und rowspan funktioniert
- Erstellen Sie selbst eine einfache Tabelle mit 3 * 3 Felder. Was passiert wenn sie dem genau in der Mitte liegenden Feld das Attribut colspan=2 zuweisen

Listing 2.9

```

1 <table class="example">
2     <thead>
3         <tr>
4             <th>Kopfzelle 1</th>
5             <th>Kopfzelle 2</th>
6             <th>Kopfzelle 3</th>
7         </tr>
8     </thead>
9     <tbody>
10        <tr>
11            <td colspan="3" class="green">Zelleninhalt</td>
12            <td>Zelleninhalt</td>
13        </tr>
14        <tr>
15            <td>Zelleninhalt</td>
16            <td>Zelleninhalt</td>
17            <td>Zelleninhalt</td>
18            <td>Zelleninhalt</td>
19        </tr>
20        <tr>
21            <td>Zelleninhalt</td>
22            <td>Zelleninhalt</td>
23            <td>Zelleninhalt</td>
24            <td>Zelleninhalt</td>
25        </tr>
26        <tr>
27            <td rowspan="2" class="green">Zelleninhalt</td>
28            <td>Zelleninhalt</td>
29            <td>Zelleninhalt</td>
30            <td>Zelleninhalt</td>
31        </tr>
32        <tr>
33            <td>Zelleninhalt</td>
34            <td>Zelleninhalt</td>
35            <td>Zelleninhalt</td>
36        </tr>
37        <tr>
38            <td>Zelleninhalt</td>
39            <td colspan="2" rowspan="2" class="green">Zelleninhalt</td>
40        </tr>
41        <tr>
42            <td>Zelleninhalt</td>
43            <td>Zelleninhalt</td>
44        </tr>
45        <tr>
46            <td>Zelleninhalt</td>
47            <td>Zelleninhalt</td>
48            <td>Zelleninhalt</td>
49            <td>Zelleninhalt</td>
50            <td>Zelleninhalt</td>
51            <td>Zelleninhalt</td>
52        </tr>
53        <tr>

```

Selbststudium

Aufgabe

Bauen Sie nachfolgende Tabelle inhaltlich korrekt nach

laufende Nummer	Lehrgang		Teilnehmer	
	Nr.	Bezeichnung	PK	Name
1	269317	Java	123456-D-78901	Saturn
2			109876-I-54321	Venus
3			209684-N-43829	Mars
4			567483-G-78965	Pluto
5	268318	C++	keiner	

Ein paar kurze Überlegungen zu Tabellen

Wie wir in diesem Kapitel gemerkt haben, kann der Quellcode einer Tabelle leicht sehr komplex werden und man kann den Überblick verlieren und vergessen vllt. ein tr zu schließen oder ein komplettes td weg zu lassen. Deswegen ein paar Überlegungen zu Tabellen in HTML.

1. Nutzen Sie Tabellen nur dann, wenn Sie wirklich eine Tabelle benötigen. Überlegen Sie sich, ob nicht eine Liste reichen würden
2. Nutzen Sie Tabellen niemals, um das Layout einer kompletten Seite damit umsetzen zu wollen. Es kann einen sehr schnell dazu verleiten. Aber Tabellen sind dazu da, viel **Inhalt** optisch und semantisch geschickt darzustellen. Nicht für das Layout --> dafür gibt es CSS

2.2.2 Headings

Headings - Überschriften

Überschriften dienen in **HTML** der Strukturierung der Inhalte. HTML kennt 6 Stufen von Überschriften. Je niedriger die Nummer, desto wichtiger (besser) ist die Überschrift. [Schulnotensystem]

Eine Überschrift H1 stellt somit immer die wichtigste Überschrift auf einer Seite dar.

Eine h1 - Überschrift sollte nur einmalig auf einer Seite vertreten sein. Als Überschrift für genau den Artikel, der auf der Seite zu finden ist.

Eine Suchmaschine wie zum Beispiel **Google** bewertet die Wichtigkeit des Textes einer Überschrift nach der Klassifizierung mit dem jeweiligen H Tag.

Beispiel der standardmäßigen Darstellung von Überschriften in einem Browser

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

Listing 2.10

```

1 <h1>Heading 1</h1>
2 <h2>Heading 2</h2>
3 <h3>Heading 3</h3>
4 <h4>Heading 4</h4>
5 <h5>Heading 5</h5>
6 <h6>Heading 6</h6>

```

HTML

Hierarchische Strukturierung

Erstellenn nur mittels Überschriften eine HTML-Seite, die die folgende Struktur logisch abbildet:

- HTML
 - Grundlagen
 - Inhalte
 - Tabellen
 - Listen
 - Formulare
 - Radio-Buttons
 - Checkboxes
 - Input
 - Textarea
 - Überschriften
- Struktur
 - Aside
 - Header
 - Footer
 - Navigation

2.2.3 Paragraphs

HTML ignoriert Umbrüche im Code

Zeilenumbrüche sollten nur innerhalb von Elementen genutzt werden, und nicht für das Layout "missbraucht" werden. Beispiel:

Dies ist ein einzelner Paragraph. Dies ist der zweite Satz des Paragraphs. Dies ist der dritte Satz des Paragraphs

Listing 2.11

```

1 <p>Dies ist ein einzelner Paragraph.
2 Dies ist der zweite Satz des Paragraphs. Dies ist der dritte Satz des Paragraphs</p>

```

HTML

Beispiele für Zeilenumbrüche:

Beispiel für Zeilenumbrüche

Dies ist ein einzelner Paragraph.
Dies ist der zweite Satz des Paragraphs.
Dies ist der dritte Satz des Paragraphs

Listing 2.12

```
1 | <p>Dies ist ein einzelner Paragraph.<br />Dies ist der zweite Satz des Paragraphs.</p>
```

HTML

Zusammenfassung

- p Ein Paragraph in HTML entspricht dem Tastenkommando "ENTER" in Word.
- br/Ein Zeilenumbruch in HTML entspricht dem Tastenkommando "SHIFT-ENTER" in Word.

Paragraphen und Zeilenumbrüche

Paragraphen dienen wie überall dazu, einen längeren Text in logische Abschnitte zu unterteilen. Genauso wie bei Word, sollte hierzu **NICHT** der einfache Zeilenumbruch [WORD: SHIFT-ENTER] genutzt werden. Sondern es sollte dafür ein neuer Paragraph erstellt werden [WORD: ENTER]. In HTML wird dies durch den Tag p realisiert. Der einfache Zeilenumbruch ist der selbstschließende Tag br/

Beispiel

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus lacinia nunc eu quam vestibulum pretium. Aliquam ullamcorper turpis ut nisi imperdiet dignissim. Quisque lacinia suscipit justo, ut mattis nisl dignissim ac. Morbi eget nibh eu neque auctor laoreet sed id leo. Suspendisse potenti. Maecenas sed sapien purus. Ut molestie pharetra justo sed hendrerit. Proin id odio vitae diam ullamcorper rhoncus. Nulla facilisi.

Nam non nisl nisl. Vivamus ac pulvinar erat. In hac habitasse platea dictumst. Praesent tincidunt, neque eu pellentesque luctus, risus tortor porttitor sapien, et molestie nibh est sit amet neque. Integer sollicitudin lacus hendrerit risus dictum nec egestas ligula condimentum. In hac habitasse platea dictumst. Praesent ornare magna eu nisl ullamcorper et tincidunt dolor dapibus. Phasellus viverra elit ac nisi semper sed feugiat purus interdum. Nulla facilisi. Phasellus ornare tincidunt facilisis. Sed fringilla, dui adipiscing blandit rhoncus, mi dui mollis erat, sit amet facilisis risus ipsum id ipsum. Mauris et blandit felis. Mauris ligula nunc, cursus et ultricies quis, pharetra in arcu.

Donec suscipit mi sed erat tempus tempus. Duis a magna sapien. Aenean elit nibh, dignissim dignissim vestibulum eu, accumsan quis nibh.

Sed ut nisi eget neque volutpat iaculis vel eu ante.

Sed placerat, augue vitae fringilla placerat, metus neque viverra odio, in laoreet nunc nibh euismod orci. Aliquam eu purus dolor.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nam eu ullamcorper dui.

Listing 2.13

```

1 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus lacinia nunc e am ves
2
3 <p>Nam non nisl nisl. Vivamus ac pulvinar erat. In hac habitasse platea dictumst. Praes
4
5 <p>Donec suscipit mi sed erat tempus tempus. Duis a magna sapien. Aenean elit nibh, digniss
6 Sed ut nisi eget neque volutpat iaculis vel eu ante.<br />
7 Sed placerat, augue vitae fringilla placerat, metus neque viverra odio, in laoreet nunc nibh
8 Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas

```

2.2.4 Images

Bilder in HTML

Bilder werden in HTML mittels des IMG tag eingebunden. Der Image-Tag ist auch ein selbstschließender Tag. Die wichtigsten Attribute, der Img enthalten darf sind:

- src

Hier wird der Pfad des Bildes angegeben, dass eingebunden werden soll. Die Pfadangabe ist relativ von der HTML-Datei. Alternativ kann der Pfad auch absolut angegeben werden.

- alt

Das alt - Attribut gibt eine **alternative** Darstellung an. Falls ein User innerhalb seines CLients die Darstellung von Bildern abgeschaltet hat, wird ihm dieser Text präsentiert. Des weiteren ist diese Beschreibung relevant in Bezug auf Barrierefreiheit. Personen, die einen ScreenReader nutzen, wird an der Stelle des Bildes diese Beschreibung vorgelesen.

- title

Das title - Attribut gibt dem Bild einen Titel. Dieser wird zum Beispiel beim MouseOver angezeigt



Listing 2.14

```

1 

```

2.2.5 Links

Das wichtigste im Web

Links sind der wichtigste Bestandteil des WWW.

Ein Link ist eine Verknüpfung zu einer Seite, Datei oder auch einem bestimmten Abschnitt innerhalb eines Dokuments. Durch Klicken eines Links erhält der Browser die Anweisung, die Ressource aufzurufen.

Sie ermöglichen die Navigation nicht nur innerhalb einer Seite - sondern auch auf andere Seite.

Ohne direkte Links zu anderen Seiten, wären Sie immer dazu gezwungen, die URI einer Seite einzeln im Browser einzugeben und dann ENTER zu drücken.



HTML Link Syntax

Ein Link macht den kompletten Inhalt, der innerhalb des Tags a steht, klickbar.

Es kann somit nicht nur Text, sondern auch Bilder oder ganze Container mit einem link versehen werden. Das Wichtigste Attribut eines links ist href. Die Abkürzung steht für Hyperlink Reference und gibt das Ziel an, auf welches der Link verweisen soll.

Listing 2.15

```
1 <a href="#">Link</a>
```

HTML

Link zu einer E-Mail Adresse

Mit Hilfe der normalen Links können auch Verknüpfungen zu e-Mail Adressen erstellt werden. Funktionsweise: Ein Klick auf solch einen Link öffnet das im Betriebssystem als Standard eingestellte e-Mail-Programm und übergibt die hinterlegte Email-Adresse. Schick mir eine Mail

Listing 2.16

```
1 <a href="mailto:test@example.org">Schick mir eine Mail</a>
```

HTML

Links in neuen Fenstern

Mittels der Attribute eines Links kann eingestellt werden, dass ein Link z.B. in einem neuen Fenster geöffnet werden soll. Dies geschieht durch Angabe des Attributes: target Mögliche Werte für das target Attribut:

Wert	Auswirkung
_blank	neues Fenster
_self (standard)	in eigentlichen Fenster
_top	im eigentlichen Fenster (nur bei Nutzung von Frames relevant)
_parent	im Eltern-Frame (nur bei Nutzung von Frames relevant)
frameName	im benannten Frame

Beispiel Google in neuem Fenster

Listing 2.17

```
1 <a href="http://www.google.de" target="_blank">Google in neuem Fenster</a>
```

HTML

verschiedene Beispiele für Links

Bilder als Links



Container als Links

Ein komplettes DIV als Link

Buttons als Links

Button

Text als Links Dieser Text ist ein Link

Ein Klick auf den Hund öffnet das Bild in seinem Ursprung

Listing 2.18

```
1 <!-- IMAGE-Link -->
2 <a href="./img/1342449951le-pic.jpg"></a>
3
4 <!-- DIV-Link -->
5 <a href=""><div class="blackbox">Ein komplettes DIV als Link</div></a>
6
7 <!-- button-Link -->
8 <a href=""><button class="button">Button</button></a>
9
10 <!-- Text-Link -->
11 <a href="">Dieser Text ist ein Link</a>
12
```

HTML

2.2.6 Formulare

Formulare in HTML

Zu Beginn der Zeiten des Internet, war der Schwerpunkt lediglich bei der Bereitstellung von Daten. Der Austausch von Daten und Informationen wurde noch auf den herkömmlichen Transportwegen wie z.B. physikalischen Datenträgern, eMail oder FTP vollzogen. Mit der Einführung von HTML2 im Jahr 1995 wurde die Möglichkeit geschaffen mit HTML Formulare auszugeben. Anfangs wurden diese meist lediglich dazu benutzt etwas Interaktivität auf den Seiten zu schaffen. Es wurden Kontaktformulare angeboten, so dass nicht jedes mal eine angegebene Email Adresse kopiert und in sein eigenes lokales Email-Programm kopiert werden

musste. Der User konnte direkt auf der Seite Kontakt mit dem Author aufnehmen.

Diese Idee entwickelte sich dahin weiter, dass es für die User auch möglich sein sollte, Kommentare zu hinterlassen um zusammen auf einfache Weise zu diskutieren - die ersten einfachen Foren war geboren.

Aus dieser Idee heraus - die Inhalte von Webseiten zusammen zu erstellen - kamen dann die ersten Ideen für Content Management Systeme, professionelle Lösungen für Foren, Formulare zum Suchen auf den Seiten, etc.

Mittlerweile ist auf fast keiner Seite mehr nur noch statischer Inhalt angeboten, sondern der User kann überall interagieren - mit **Formularen**.

Egal ob es die Auswahl eines Hotels - das Hinterlassen eines Kommentars, das posten eines Status auf Facebook oder die Suche nach einem Artikel auf Ebay ist - all das sind Formulare.

Ein Formular in HTML wird mittels des Tags form realisiert. Jedes Formular muss die Attribute action und method besitzen. Die beiden Attribute beschreiben, **wohin** und **wie** die Inhalte übergeben werden sollen

Mögliche Werte für die Attribute: action

Wert Auswirkung

" " (deprecated) übergibt die Werte an sich selber (lädt dadurch die Seite neu)

formular.php übergibt die Werte an die Datei formular.php

method

Wert Auswirkung

post überträgt die Daten im HTTP - Header

get überträgt die Daten in der URL - z.B. http://google.de?suche=FORMULARINHALT

Ein erstes Formular

<input type="text"/>	Submit
----------------------	--------

und der dazugehörige Code:

Listing 2.19

```

1 <form action="" method="">
2     <input type="text"/>
3     <input type="submit"/>
4 </form>

```

HTML

das input element

das input Element stellt das meistgenutzte Eingabefeld innerhalb eines Formulars dar. Das input Element ist ein variables Element und kann je nach Definition durch das Attribut type ein verschiedenes Aussehen haben und verschiedene Daten aufnehmen.

Beispiele:

Wert Auswirkung

type="text"

type="password"

type="search"

type="submit"

Als weiteres wichtiges Attribut für das input Element betrachten wir das Attribut maxlength. Durch setzen dieses Attributs ist es möglich, die maximale Eingabelänge eines Feldes zu begrenzen. maxlength="40" würde eine Feld zum Beispiel auf 40 Zeichen begrenzen

mehrzeilige Eingabebereiche

Wir haben bereits das Element input fuer Formulare kennen gelernt. Mit dessen Hilfe war es uns schon möglich ein Formular zu erstellen, um einfache Daten aufzunehmen. Stellen wir uns jedoch vor, wir möchten es dem Nutzer ermöglichen uns ein längeres Kommentar zu übersenden. Das Element input stellt uns nur eine einzige Zeile zur Verfügung. Dies macht es offensichtlich sehr schwer einen größeren Text einzutragen und diesen im Überblick zu behalten

Zur Lösung dieses Problems ist in HTML das Element textarea vorgesehen, welches einen beliebig großen Eingabebereich zur Verfügung stellt.

Beispiel: Inhalt der Textarea

Mittels der Attribute cols und rows definiert man, wieviele Spalten bzw. Reihen vorgegeben werden sollen. Moderne Browser erlauben zudem, das textarea Element mit der Maus auf die jeweiligen Bedürfnisse des Nutzers temporär zu vergrößern

Listing 2.20

```
1 <textarea cols="75" rows="5">Inhalt der Textarea
```

HTML

Aufgabe

Erstellen Sie ein einfaches Formular, dass zur Registrierung eines Nutzers auf einer Homepage dienen soll (Erstellen Sie nur den HTML Rahmen, das Formular muss noch keinerlei Funktionen besitzen):

Normalerweise findet man bei Registrierungsformularen diese verschiedenen Felder:

- Vorname
- Name
- Nutzernamen
- Passwort
- Passwort wiederholen
- Geburtsdatum
- Geschlecht
- Haustiere (Hund, Katze, Maus, Ente)
- Beschreibung oder Kommentar
- Newsletter (ja oder nein)

Überlegen Sie sich, welcher Typ von Eingabefeld für das jeweilige Feld von Vorteil wäre und nutzen Sie diesen. Gruppieren Sie ihr Formular in logische Abschnitte mit dem Tag fieldset. Zum Beispiel in:

Beschriftungen für die Felder

Jedes Eingabefeld kann das Attribut name besitzen. Das Attribut name gibt dem jeweiligen Eingabefeld einen **eindeutigen** Bezeichner. Dies ist speziell dann wichtig, wenn die Daten aus dem Formular später verarbeitet werden sollen. Das Script, an das das Formular gesendet wird, muss zuordnen können, welches Feld es z.B. wo in die Datenbank schreiben soll.

Des weiteren sollte jedem Formularelement eine eindeutige ID zugeordnet werden. Der Vorteil des id - Attributes ist, dass es uns ermöglicht eine eindeutige Beziehung zu diesem Feld zu erstellen - ein label

label Der label-Tag dient lediglich dazu einen Text einem Formularelement zuordnen zu können. Wenn Sie einem Formular-Element mittels label for='ID DES FORMULARELEMENTS' zugeordnet haben, springen Sie bei dem Klick auf ein Label automatisch in das zugehörige Formularfeld.

Labels sollten sie speziell bei Checkboxes verwenden. Es ist einfacher den kompletten Text auch anklicken zu können, als nur das kleine Checkbox Feld

Möchten Sie mehrere Checkboxes zusammenfassen, kommen Sie mit dem label-Tag nicht weiter, das dieser nur einer einzigen id zugeordnet werden kann. Für die Gruppierung mehrerer Formularelemente sieht html das fieldset mit legend als Beschreibung vor.

Ihre Haustiere Hund ☐ Katze ☐ Maus ☐

Listing 2.21

```

1  <fieldset>
2
3    <legend>Ihre Haustiere</legend>
4
5    <label for="hund">Hund</label>
6    <input name="haustier" type="checkbox" id="hund" value="hund" />
7
8    <label for="katze">Katze</label>
9    <input name="haustier" type="checkbox" id="katze" value="katze" />
10
11   <label for="maus">Maus</label>
12   <input name="haustier" type="checkbox" id="maus" value="maus" />
13
14 </fieldset>

```

HTML

Checkboxes - Auswahlfelder

Man kann auch diesen Text klicken für die Auswahl der Checkbox ☐ Beispiel für Checkboxes
 Hiermit stimmen Sie den AGB zu: ☐ Hiermit stimmen Sie der Zusendung von Newslettern zu ☐

Radioboxes - nur eine Auswahl ist möglich

Hatten Sie bei den Checkboxes noch die Auswahl, auch mehrere Elemente anzuwählen, bieten Radio-Boxes die Möglichkeit, genau eine Auswahl zu treffen.

Überlegen Sie sich 2 mögliche Beispiele, in denen Radio-Buttons sinnvoller sind, als Checkboxes

☐ Mastercard

- ☐ Visa
- ☐ American Express

Finden Sie heraus, wie man Radio-Buttons gruppiert, so dass immer nur einer von mehreren markiert werden kann. (wie im obigen Beispiel)

2.2.7 Textformatierung

Einfache Formatierungen

This text is bold

This text is italic

This is computer output

This is subscript and superscript

~~Dieser Text ist durchgestrichen~~

b vs. strong

In den meisten normalen Browsern werden die beiden TAGs b und strong identisch dargestellt. Sie dürfen jedoch nicht als komplementär angesehen werden.

Unterschiede: b Der Tag b dient lediglich der optische fetteren Darstellung eines Ausdrucks. Inhaltlich (semantisch) hat dieser Ausdruck nicht mehr Bedeutung als normaler Text. strong Der Tag strong hat zusätzlich zur optischen Gestaltung eine semantische Bedeutung: Ein Ausdruck der mit strong markiert wurde ist als wichtiger zu betrachten, als normaler Text. Warum sollte mich das interessieren wenn beide Tags exakt gleich dargestellt werden?

Beispiel:

Ein Screenreader (Gerät, dass den Inhalt einer Homepage vorlesen kann - wird benutzt von Menschen mit Sehbehinderungen) würde bei dem Tag strong diesen Ausdruck auch betont vorlesen. Dies ist bei dem Tag b nicht der Fall.

em vs. i

Parallel zu den tags b und strong verhalten sich auch die Tags EM und I. EM hat eine semantische Bedeutung und I besitzt lediglich die optischen Eigenschaften für den Browser

2.2.8 Listen

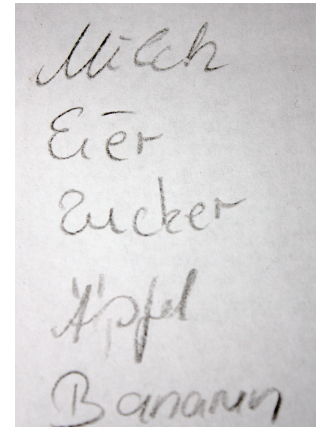
Geordnete und ungeordnete Listen

ungeordnete Liste	nummerierte Liste
▪ Listenelement	1. erstes Element
▪ Listenelement	2. zweites Element
▪ Listenelement	3. drittes Element
▪ Listenelement	4. viertes Element
▪ Listenelement	5. fünftes Element

Nummerieren, Definieren, oder einfach nur so....

HTML kennt 3 verschiedene Möglichkeiten Listen zu erstellen:

- unsortierte Listen
- sortierte Listen
- Definitionslisten



Unsortierte Listen - unordered lists [ul]

Die einfachste Form einer Liste ist die Aufzählung.

In HTML wird die unsortierte Liste mit dem Element `ul` eingeleitet und dementsprechend mit `/ul` wieder geschlossen.

Die einzelnen Elemente innerhalb einer Liste werden mit dem Tag `li` definiert.

Gemäß den Spezifikation von HTML ist es nicht nötig, jedes einzelne `li` auch wieder explizit zu schließen. Um eventuelle Darstellungsprobleme im vornherein zu verhindern und auch einen einheitlichen Coding-Stil zu gewährleisten, sollten alle `li` auch wieder mit `/li` geschlossen werden.

- Milch
- Apfel
- Bananen
- Joghurt

Listing 2.22

```

1 <ul>
2   <li>Milch</li>
3   <li>Äpfel</li>
4   <li>Bananen</li>
5   <li>Joghurt</li>
6 </ul>

```

HTML

2.3 Struktur in HTML

2.3.9 Blockelemente vs. Inline Elemente

Gruppierung von Elementen

HTML-Elemente können mit Hilfe der beiden Tags `span` und `div` zu Gruppen zusammengefasst werden. Der Unterschied zwischen diesen beiden Elementen liegt in Ihrer Darstellungsform.

Grundsätzlich haben diese beiden Elemente keinerlei Auswirkung auf das Design. Jedoch hat ein `DIV`-Container von Haus aus eine Breite von **100%**. Ein `span`-Tag hingegen hat keine feste Breite, sondern gliedert sich in den fließenden Tag mit ein.

Ein span sollte also benutzt werden, wenn man nur einzelne Buchstaben oder Wörter innerhalb eines Textes markieren will; ein div hingegen, wenn man einen ganzen Block gruppieren und hervorheben will.

HTML Block Elemente

BLOCK Elemente werden normalerweise -- (kann mit CSS überschrieben werden) -- in einer neuen Linie dargestellt.

Sie haben grundsätzlich eine Breite von 100%. Beispiele für Block Elemente:

- h1
- p
- ul
- table
- div

Inhalt des orange P

Listing 2.23

```

1 <style type="text/css">
2     <!--
3         .orange {
4             background:orange;
5         }
6     -->
7 </style>
8
9
10
11 <p class="orange">Inhalt des orange <span class="tag">P</span></p>

```

HTML Inline Elemente


HTML Inline-Elemente werden im Gegensatz zu den Block-Elementen normalerweise innerhalb des normalen Textfluss dargestellt. Beispiele für Inline Elemente:

- span
- img
- strong
- em
- a

Inhalt des orangeSPAN

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean vitae eros eros, vel ultricies dolor. Aenean ultricies adipiscing viverra. Morbi in adipiscing quam. Proin neque lacus, lacinia placerat porta at, adipiscing ut neque. Nunc metus arcu, consectetur id lobortis in, lacinia nec augue. Ut vitae ipsum sit amet velit ornare dapibus. Curabitur eget libero vitae sem hendrerit porttitor id sed arcu. Sed condimentum ligula at nunc suscipit nec auctor tortor interdum. Proin non leo leo, sed consequat leo. Aliquam eu libero mi. Etiam non orci ac turpis porta laoreet. Ut sollicitudin porttitor mi, vel consectetur lorem lobortis ut. Nunc ipsum ipsum, faucibus eget rhoncus ac, faucibus eu magna.

Listing 2.24



```

1 <style type="text/css">
2   <!--
3     .orange {
4       background:orange;
5     }
6   -->
7 </style>
8
9
10 <span class="orange">Inhalt des orange SPAN</span>
11
12
13 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean vitae eros eros, vel ut
14 Aenean ultricies adipiscing viverra. Morbi in adipiscing quam. Proin neque lacus, lacinia
15 adipiscing ut neque. Nunc metus arcu, consectetur id lobortis in, <span class="orange">lac
16 Ut vitae ipsum sit amet velit ornare dapibus. <span class="orange">Curabitur</span> eget l
17 porttitor id sed arcu. Sed condimentum ligula at nunc suscipit nec auctor tortor interdum.
18 consequat leo. Aliquam eu libero mi. Etiam non orci ac turpis porta laoreet. Ut sollicitud
19 lorem lobortis ut. Nunc ipsum ipsum, faucibus eget rhoncus ac, faucibus eu magna.</p>

```

2.3.10 Semantische Strukturierung

Bekämpfung des Missbrauchs

Mit der Einführung von HTML5 wurden endlich Elemente geschaffen, um eine Seite semantisch korrekt zu strukturieren. In den Zeiten vor HTML5 wurden meist normale Container dazu missbraucht, mittels Ihrer ID zu beschreiben, was sie beinhalten. Durch die Analyse von Milliarden von Webseiten konnte ein eindeutiges Muster ermittelt werden, welche Bereiche auf einer Webseite zwingend notwendig sind. In fast allen Fällen gab es Container mit den folgenden IDs:

- #nav bzw. #navigation
- #header
- #footer
- #content
- #content_left bzw. #content_right

Da bei dieser empirischen Untersuchung derart viele Übereinstimmungen in der Struktur von Webseiten gefunden wurde, veranlasste dies die **WHATWG** (Community des W3C zur Weiterentwicklung von HTML) gewisse Elemente zu schaffen, die exakt nur diesem Zweck dienen. Durch die Nutzung dieser neuen Elemente wird versucht eine größere Vereinheitlichung zu schaffen und zusätzlich den Computern mehr Hilfe zu bieten, die Struktur einer Seite an Hand eindeutiger Objekte zu verstehen. Die wichtigsten neugeschaffenen Elemente sind:

- nav
- header
- footer
- article
- aside

Eine genauere Erklärung dieser Elemente erübrigt sich auf Grund der Eindeutigkeit der Namen.

Diese Elemente verhalten sich in der Handhabung mit CSS exakt wie ein Container. --> auch ältere Browser stellen diese Elemente richtig dar. (nämlich ohne optische Darstellung, so wie einen Container)

Exakte Definition der ElementeWHATWG

3 CSS

CSS - Cascading Style Sheets

Die **Cascading Style Sheets**, kurz **CSS** genannt, sind eine deklarative Sprache für Stilvorlagen von strukturierten Elementen. Sie werden hauptsächlich in Verbindung mit HTML eingesetzt.

"CSS macht Webseiten schön"

Der Grundgedanke ist, Inhalt und Aussehen voneinander zu trennen. Dafür gibt es mehrere Gründe. Einer der Gründe ist, dass nicht jeder Client (egal ob Mensch vor dem Computer oder reine Maschine) eine optische Aufbereitung von Informationen benötigt. Einer Maschine ist es egal, ob Text in gelb oder blau dargestellt wird. Eine Maschine liest nur das reine HTML und erkennt an Hand der korrekten Strukturierung, was eine Überschrift ist und was nur reiner Text. Unter anderem aus diesem Grund heraus, kam die Entscheidung, Inhalt und Design zu trennen. Somit müssen weniger Daten übertragen werden. In heutigen Zeiten von VDSL ist dies zwar irrelevant, jedoch gibt es noch weitere Gründe.

HTML bietet zwar unter anderem die Möglichkeit CSS direkt innerhalb des HTML Codes einzusetzen, jedoch müsste dieser bei jedem Element erneut geschrieben werden. D.h. bei einer Seite mit nur 5 Überschriften, muss 5 mal definiert werden, wie die Überschrift auszusehen hat. Es kann hierbei leicht einmal zu kleinen Tippfehlern kommen, geschweige denn von der Übersichtlichkeit des Codes. Die zweite Möglichkeit CSS zu nutzen, ist, die Stylesheets innerhalb einer HTML Datei im Kopf-Bereich zu definieren. Diese gelten dann für alle Elemente innerhalb dieser Datei. Doch eine normale Webseite besteht unglücklicherweise nicht nur aus einer Datei, sondern meist aus vielen Hundert.

Daraus folgt der logische Schritt, CSS über externe Dateien einzubinden. Es wird hierbei mindestens eine Datei erstellt, von welcher aus sich zentral das Design aller kompletten Einzelseiten steuern lässt. Die Wartbarkeit des Codes nimmt hierbei immens zu. Zusätzlich kann man auch problemlos durch Austauschen einer einzigen Datei, dem gesamten Webauftritt ein einheitlich neues Design verpasst werden.

Möglichkeiten der Einbindung von CSS

Wie bereits erwähnt haben wir 3 verschiedene Möglichkeiten CSS einzubinden:

1. externe Dateien

2. Embedded CSS
3. Inline CSS

das attribut color legt die Farbe der Schrift fest

Anbei ein Beispiel einer HTML Datei, die alle 3 verschiedenen Varianten kombiniert.

Listing 3.1

```

1  <!DOCTYPE html>
2  <html>
3
4  <head>
5
6      <link rel="stylesheet" type="text/css" href="style.css" />
7
8      <style type="text/css">
9          h1 {
10             color:green;
11          }
12      </style>
13
14 </head>
15
16 <body>
17
18     <h1>Grüne Überschrift</h1>
19
20     <h1 style="color:yellow">Gelbe Überschrift</h1>
21
22 </body>
23
24 </html>

```

HTML

Hierarchie von CSS

Es wird stark angeraten, CSS nur mittels der externen Dateieinbindung zu nutzen. Lediglich für schnelle Testzwecke sollte inline CSS verwendet werden. Der Grund für diesen Rat, liegt in der Wartbarkeit und Entwicklung eines Web-Projekts. Falls innerhalb von CSS einem Element 2 mal das gleiche Attribut zugewiesen, jedoch mit verschiedenen values, nutzt der Browser immer das, was er als letztes gelesen hat. Da ein Browser eine HTML - Seite von oben nach unten liest, kommen wir zu folgender Hierarchie:

Inline-CSS > Embedded CSS > Externer CSS-Datei

Für den Fall, dass Sie nun die 3 verschiedenen Möglichkeiten der Einbindung genutzt haben: Sie schreiben in Ihrer externen CSS - Datei neue Werte und wundern sich warum nichts passiert? Wahrscheinlich haben Sie vergessen, dass Sie genau bei diesem Element auch Inline Code genutzt haben.

Betrachten Sie nachfolgenden Code und beantworten Sie folgende Fragen:

- In welcher Farbe wird die Überschrift H1 dargestellt?
- In welcher Farbe wird die Überschrift H2 dargestellt?
- In welcher Farbe wird die Überschrift H3 dargestellt?

Listing 3.2

HTML

```

1  <!-- DIESER BLOCK STELLT DIE EXTER EINGEBUNDENE DATEI DAR -->
2
3      h1 {
4          color:blue;
5      }
6
7      h2 {
8          color:orange;
9      }
10
11     h3 {
12         color:fuchsia;
13     }
14
15 <!--      ENDE DER EXTERNEN EINGEBUNDENEN DATEI DAR      -->
16
17
18 <!DOCTYPE html>
19 <html>
20
21 <head>
22
23     <link rel="stylesheet" type="text/css" href="style.css" />
24
25     <style type="text/css">
26         h1 {
27
28         }
29         h2 {
30             color:green;
31         }
32         h3 {
33             color:green;
34         }
35     </style>
36
37 </head>
38
39 <body>
40
41     <h1>Äberschrift 1</h1>
42
43     <h2 style="color:yellow">Gelbe Äberschrift</h2>
44
45     <h2 style="color:green">Gelbe Äberschrift</h2>
46
47 </body>
48
49 </html>

```

Inder

Inder können sehr toll programmieren. Und wenn man ihnen eine klatscht, werden sie grün :D

3.1 CSS - Basics

CSS Syntax

Listing 3.3

```

1  h1 {
2      color: #333
3  }
4  /* ALLE h1 bekommen die Farbe #333 */
5
6
7
8  h1.orange {
9      color:orange
10 }
11 /* alle h1, die auch die klasse orange habe */
12 /* <h1 class="orange"></h1> */
13
14
15 #content {
16     border: 5px;
17 }
18 /* genau das Element mit der id content */
19 /* <div id="content"></div> */
20
21
22 li a {
23     color:#990000;
24 }
25 /* alle links, die innerhalb eines li stehen*/
26
27 li a.blue {
28     color:blue
29 }
30 /* links, die innerhalb eines li stehen UND die Klasse blue haben */
31
32

```

CSS

Ein komplettes Beispiel

Hier könnte auch Ihr Inhalt stehen

Listing 3.4

```

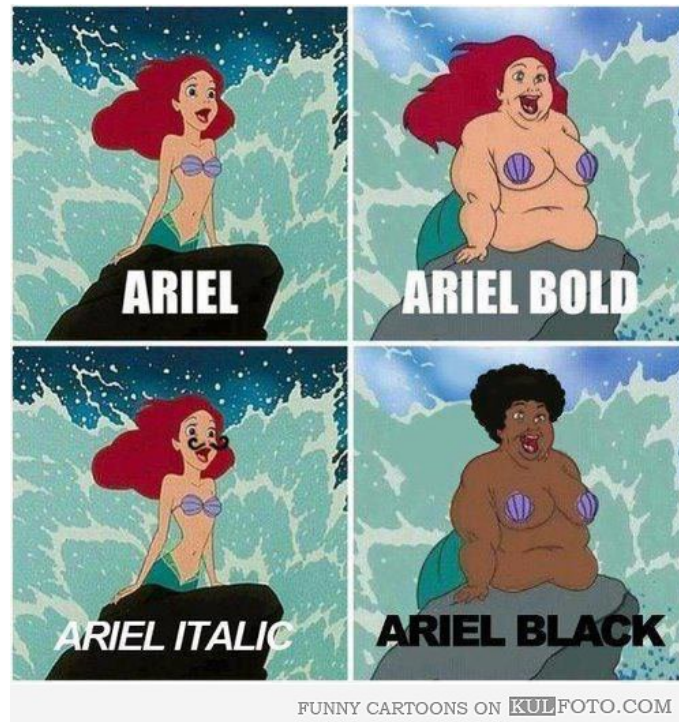
1 .beispielKlasse {
2     border:2px solid red;
3     background: orange;
4     padding:20px;
5 }

```

CSS

3.1.11 Schriften

Arial = Ariel ?



Schriften formatieren mit CSS

Für die Definition des Aussehens von Schriften bietet CSS viele verschiedene Möglichkeiten:

font-family definiert die zu benutzende Schriftart font-family: Arial, Helvetica, sans-serif;

font-weight definiert die Dicke der Schrift font-weight: bold;

font-style font-style: italic;

font-size definiert die Schriftgröße font-size: 15px;

Mit diesen 4 Attributen können sie 99% aller Anforderungen an Schriften abdecken. Recherchieren Sie dennoch, ob es noch weiteren gebräuchliche Attribute für Schriften gibt.

3.1.12 Borders (Simple)

Ränder um Elemente

Es kann jedem HTML Element ein Rand zugewiesen werden. In diesem Paragraph wird darauf eingegangen, wie man Ränder um Elemente setzen kann.

Für Ränder mittels CSS gibt es 2 verschiedene Schreibweise. Die Kurzschreibweise und die Langschreibweise.

```
border-top: 1px solid #990045;  
border-right: 1px solid #990045;  
border-bottom: 1px solid #990045;  
border-left: 1px solid #990045;
```



```
border: 1px solid #990045;
```

das border attribut im detail

```
border-bottom: 1px solid #990045;
```



Beispiel:

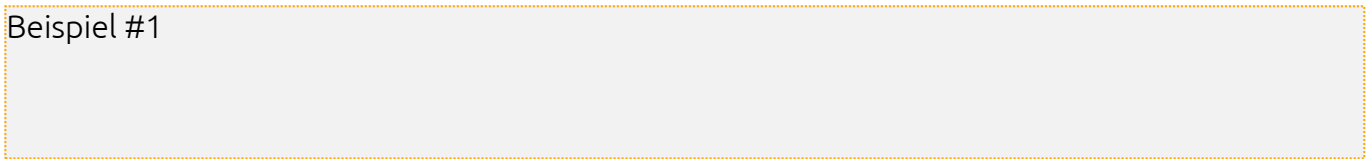
Beispiel #1



Beispiel #1



Beispiel #1



Beispiel #1



Listing 3.5

```

1  .beispiel1 {
2      border:2px solid green;
3  }
4
5  .beispiel2 {
6      border:4px dashed red;
7  }
8
9  .beispiel3 {
10     border:1px dotted orange;
11 }
12
13 .beispiel4 {
14     border-bottom:22px solid green;
15     border-top:4px solid fuchsia;
16     border-left:5px solid blue;
17     border-right:4px solid orange;
18 }

```

CSS

3.1.13 Farben

Farben in CSS

Sie haben in CSS mehrere verschiedenen Möglichkeiten Farben anzugeben. Die einfachste Angabe erfolgt über die englische Bezeichnung der jeweiligen Farbe. Das Problem welches hierbei jedoch sehr schnell auftritt: wieviele Farbnamen kennen Sie im Englischen und kennen Sie auch die jeweiligen Abstufungen?

Beispiel: blau = blue hellblau = lightblue doch wie sieht es auch mit aliceblue oder darkshadowblue



Finden Sie heraus wie die Bezeichnungen für 16 wichtigsten Farben lauten

Man sieht direkt, dass man sich hinter einigen Farbenbezeichnungen nur wenig vorstellen kann. Sie eignen sich somit nur für schnelle Tests, aber nicht für einen produktiven Einsatz.

CSS nutzt im Allgemeinen Hexadezimale Angaben für die Zahlen. Diese basieren auf dem RGB Farbmodell

Alternativ können die Farben auch dezimal angegeben werden. Dies wird praktisch jedoch nur sehr selten angewandt, da es einfach zu tippen ist.

Recherchieren Sie die Bedeutung der Kurzschreibweise von CSS - Farbcodes und erschließen Sie sich folgende Beispiele

- Was kommt heraus bei: #333
- Was kommt heraus bei: #F3A
- Was kommt heraus bei: #F3AB
- Was kommt heraus bei: #F

- Was kommt heraus bei: #FH3434
- Was kommt heraus bei: #blue
- Was kommt heraus bei: #mellongreen

Listing 3.6

```

1 color: teal;
2
3 color: #333333;
4 color: #9900FF;
5
6 color: rgb(255,0,0);
7 color: rgb(90%, 25%, 4%);

```

CSS

Was kann alles farbig werden?

Sie können mit CSS fast alle Attribute mit Farbe hinterlegen: Die für Sie gängigsten Attribute die sie mit Farbe hinterlegen werden:

- Standard element wie p oder h mittels color:
- Ränder entweder mit border-color: oder der Kurzschreibweise border:
- Hintergründe von Elementen mittels background-color: oder der Kurzschreibweise background:

Beispiele:

Listing 3.7

```

1 div {
2     background: #990033;
3     border: #cccccc;
4 }
5
6 h1 {
7     color: #456789;
8     border-bottom: #789787;
9 }
10
11 span {
12     border-color: orange;
13     border: 1px 1px 3px 5px;
14 }

```

CSS

3.1.14 Außßen und Innenabstände

Innenabstand von Elementen

Beispiele sagt oftmals mehr als tausend Worte:

padding: 0px

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy

eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

padding:50px

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

padding:15px

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Wie die Beispiele verdeutlichen ermöglichen die Werte von padding, den Innenabstand von Elementen zu definieren. Es kann hierbei wieder die Kurz und die Langschreibweise gewählt werden, um zum Beispiel links und rechts verschiedene Innenabstände zu definieren

Listing 3.8

```

1  div {
2      padding: 15px;
3  }
4  /* alle 4 Seiten haben ein padding von 15px */
5
6  div {
7      padding-top: 10px;
8      padding-bottom: 20px;
9  }
10 /* Oben und Unten sind verschiedene Werte fuer padding definiert */

```

CSS

3.2 Layouting with CSS

3.3 Advanced CSS Techniques

3.3.15 text-shadow

text-shadow

text-shadow: 1px 1px 5px #333;



Die text-shadow - Eigenschaft wurde bereits in CSS 2.0 eingeführt. Jedoch wurde diese CSS-Eigenschaft von keinem Browser implementiert. In CSS 2.1 wurde die Textschatten-Eigenschaft entfernt und erst wieder in CSS3 in Kraft gesetzt. Daraufhin fingen fast alle Browser-Hersteller mit der Umsetzung der Text-Schatten an.

Listing 3.9

```
1 h1 {
2   text-shadow: 1px 1px 5px #333;
3 }
```

CSS

Einfacher Textschatten

Einfacher Textschatten

Breiter Textschatten

Einfacher Textschatten

Das ganze noch etwas schöner

Einfacher Textschatten

Eine einfacher Kontur mittels Textschatten

Kontur

Glow - Effekte mittels CSS

Glow-Effekt

Letterpress

Letterpress

Emboss

Emboss

Retro Design

Retro Design

Neon - Effekte



Neon

feuerartiger Effekt



feuerartiger

3.3.16 box-shadow

3.3.17 vendor-prefixes

4 WEBSERVER

4.1 HTTP und das Web

4.1.18 Webserver und Protokolle

Was ist ein Webserver?

Betrachtet man nur die physikalischen Komponenten des Internets, dann ist dieses Netzwerk eine gigantische Ansammlung von aktiven Komponenten (Switches, Router, usw.) und Computern, die verschiedenste Dienste in diesem Netzwerk zur Verfügung stellen (Webserver). Was aber ist ein Webserver genau?

- Was ist die Hauptaufgabe eines Webserver?
- Nennen Sie drei Protokolle, die häufig zur Kommunikation mit einem Webserver benutzt werden!
- Auf welcher Schicht im ISO/OSI-Modell ist ein Webserver einzuordnen? Begründen Sie Ihre Antwort!
- Wo, außer im Internet, kann man Webserver finden? Nennen Sie mindestens drei Möglichkeiten!

Um Nachrichten, Webseiten oder Binärdateien mit anderen Webservern oder Browsern austauschen zu können, benötigen Webserver Protokolle, wie z. B. das "Hyper Text Transfer Protocol", kurz HTTP.

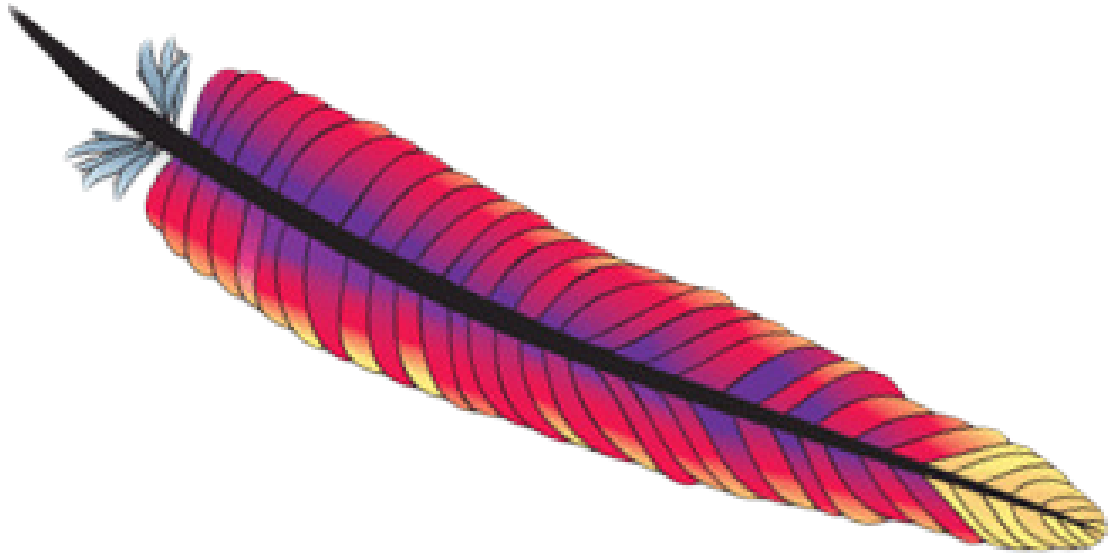
- Was ist ein Protokoll (Artikel "Kommunikationsprotokoll")?
- Welche Hauptaufgabe hat das Protokoll HTTP (Artikel "HTTP")?
- Zu welcher Protokollfamilie gehört HTTP?
- Auf welchen Schichten des ISO/OSI-Modells ist HTTP beheimatet?
- Wie werden die Kommunikationseinheiten in HTTP bezeichnet?
- Wie viele unterschiedliche Arten solcher Kommunikationseinheiten gibt es in HTTP? Nennen Sie auch deren Namen!
- Was ist ein "HTTP-Header", und was ist ein "Message-Body"? Welche Informationen enthalten HTTP-Header bzw. Message-Body?
- Was bedeuten die Angaben "HTTP-Get" und "HTTP-Post"? Worin unterscheiden sich diese beiden Dinge?
- Worin liegt der Unterschied zwischen den Versionen "HTTP 1.0" und "HTTP 1.1"?

Ein **URI** ("Uniform Resource Indicator/Identifizier") bezeichnet die vollständige Adresse, um beliebige Daten auf einem Server im Internet zu finden. Ein Beispiel für einen URI könnte sein:

http://www.fueustgsbw.de:80/intern/service/lehre_und_ausbildung/index.html

- Wie heißen die Teile, in die sich der obige URI gliedert (Hinweis: Es sind fünf Teile)?
- Worin liegt der Unterschied zwischen einem URI und einem URL?

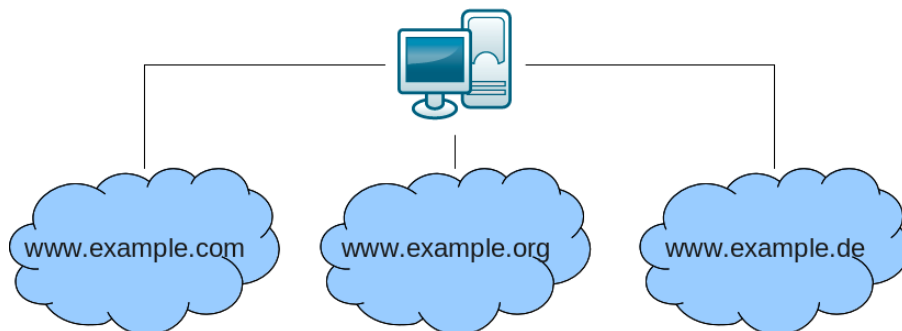
4.2 Apache



4.2.19 Virtual Hosting

IP-Basiertes Virtual Hosting

Unter dem Begriff "Virtual Hosting" versteht man die Möglichkeit auf einem einzigen physikalischen Webserver mehrere Webangebote gleichzeitig bereitzustellen.



Virtual Hosting kann IP-Basiert oder Namensbezogen erfolgen. Um IP-Basiertes Virtual Hosting betreiben zu können, braucht ein Webserver mehrere IP-Adressen, bzw. eine IP-Adresse mit mehreren freien Ports. Die Reihenfolge der Schritte zur Konfiguration von V-Hosts ist:

1. Konfiguration der IP-Einstellungen am Webserver durch Sys-/Netzwerkadmin.
2. Freischalten von IP-Adresse und Port an einer evtl. vorhandenen Firewall durch den Sys-/Netzwerkadmin.
3. Konfiguration des Apache Webservers mittels des VirtualHost-Kontainers.

Der Container VirtualHost wird verwendet, um in der Hauptkonfigurationsdatei des Apache Webservers einen Virtuellen Host darzustellen. Am Ende dieser Sektion ist ein Beispiel dazu angegeben.

1. Wo können Sie in der Onlinedokumentation ansehen, welche Direktiven für einen VirtualHost-Kontainer zulässig sind?
2. Warum ist hier die Direktive ServerName sehr wichtig? Recherchieren Sie unter: <http://httpd.apache.org/docs/2.4/dns-caveats.html>
3. Benutzen Sie das unten angegebene Konfigurationsbeispiel, um einen Webauftritt für SelfHTML zu erstellen! SelfHTML soll unter der URL <http://localhost:8080> für alle Nutzer erreichbar sein! Welche Konfigurationsschritte sind dazu notwendig?

Für die Angabe eines Ports gibt es drei Möglichkeiten:

- Keine Angabe, z. B. 127.0.0.1: In diesem Fall wird Port 80 angenommen.
- Angabe eines Ports, z. B. 127.0.0.1:8080: Es wird der angegebene Port verwendet.
- Angabe eines *, z. B. 127.0.0.1:*: Ein so konfigurierter Virtual Host beantwortet Anfragen auf allen Ports.

Erstellen Sie einen weiteren virtuellen Host, der auf Port 80 den MediaWiki bereitstellt!

```
1 <VirtualHost "127.0.0.1:8080">
2     ServerAdmin    root@127.0.0.1
3     DocumentRoot   "E:/apache/httpd/2.4.2/htdocs/port_8080"
4     ServerName      "www.example.org"
5     DirectoryIndex  index.php
6 </VirtualHost>
7
```

Sobald eine größere Anzahl von Virtuellen Hosts auf einem Apache Webserver konfiguriert wird, kann es leicht passieren, dass dessen Konfigurationsdatei sehr unübersichtlich wird. Um dies zu vermeiden, bietet er die Möglichkeit einzelne Konfigurationsabschnitte, z. B. die Konfiguration eines Virtual Hosts, in eine andere Datei auszulagern und diese in die Konfiguration des Servers einzubinden.

Benutzen Sie die Apache Onlinedokumentation, um zu Recherchieren wie die beiden Direktiven Include und IncludeOptional funktionieren, bzw. worin sich beide unterscheiden!

- Speichern Sie die Konfigurationen der beiden Virtual Hosts in jeweils einer eigenen Datei (port_80.conf und port_8080.conf)
- Konfigurieren Sie Ihren Apache so, dass beide VHosts dargestellt werden. Sollte die Konfiguration von VHost 1 nicht gelesen werden können, muss der Start des Apache Servers unterbrochen werden. Bei VHost 2 ist diese Maßnahme nicht notwendig, bzw. nicht erwünscht!
- Konfigurieren Sie jetzt VHost3 und speichern Sie seine Konfiguration in der Datei generic.conf. Binden Sie diese Datei an erster Stelle ein, und zwar genauso wie die Konfiguration von VHost2.
- Testen Sie den Zugriff auf die URL <http://localhost>! Welcher VHost antwortet?
- Verschieben Sie VHost 3 auf den ersten Platz, so dass die Reihenfolge der VHosts 3, 1, 2 lautet und testen Sie erneut den Zugriff auf <http://localhost>! Welcher VHost antwortet jetzt?
- Legen Sie für die beiden virtuellen Hosts 1 und 3 explizit den Port 80 fest (VirtualHost "127.0.0.1:80")! Welche Antwort erhalten Sie, wenn Sie jetzt auf <http://localhost> zugreifen?
- Konfigurieren Sie einen weiteren virtuellen Host der Unter der IP-Adresse 10.3.11.xx (xx = Platznummer) die Datei begruessung.html anzeigt, und PERL-Skripte verarbeiten kann. Speichern Sie auch dessen Konfiguration in einer eigenen Datei, und binden Sie diese an dritter Stelle so ein, dass der Serverstart unterbrochen wird, falls die Datei nicht verfügbar ist.
- Konfigurieren Sie den virtual Host VirtualHost 10.3.11.xx explizit für Port 80 (VirtualHost 10.3.11.xx:80)!
- Tragen Sie die folgende Direktive in Ihre Konfiguration ein: NameVirtualHost 10.3.11.xx:*
- Testen Sie den Zugriff auf die URL <http://10.3.11.xx:80>! Welche Antwort erhalten Sie?

```

1 #VHost 1 - port_80.conf
2 <VirtualHost "127.0.0.1">
3     ServerAdmin    root@127.0.0.1
4     DocumentRoot   "E:/Apache/httpd/2.4.2/htdocs/port_80"
5     ServerName      "www.example.org"
6 </VirtualHost>
7
8 #VHost 2 - port_8080.conf
9 <VirtualHost "127.0.0.1:8080">
10     ServerAdmin    root@127.0.0.1
11     DocumentRoot   "E:/Apache/httpd/2.4.2/htdocs/port_8080"
12     ServerName      "www.example.de"
13     DirectoryIndex index.php
14 </VirtualHost>
15
16 #VHost 3 - generic.conf
17 <VirtualHost "127.0.0.1:*">
18     ServerAdmin    root@127.0.0.1
19     DocumentRoot   "E:/Apache/httpd/2.4.2/htdocs/generic"
20     ServerName      "www.example.com"
21 </VirtualHost>
22

```

Für die Aktivierung des Name-Based V-Hosting, muss die NameVirtualHost-Direktive genutzt werden. Um z. B. einen namens-basierten virtuellen Host auf der IP-Adresse 10.2.11.120 und

dem DNS-Namen `selfhtml.it-training.fus` zu erstellen, muss eine Konfiguration eingerichtet werden, wie sie unten abgebildet ist. Dieses Beispiel ist jedoch nur ein Auszug aus der Hauptkonfigurationsdatei. An dem `VirtualHost`-Kontainer hat sich nichts geändert. Lediglich die `NameVirtualHost`-Direktive ist neu hinzugekommen.

Um das Name-Based V-Hosting zu aktivieren, muss die `NameVirtualHost`-Direktive mit der IP-Adresse und dem Port konfiguriert werden, auf dem der V-Host aktiviert werden soll. Des weiteren muss jeder Name-Based V-Host **Z W I N G E N D** eine `ServerName`-Direktive erhalten, um von den anderen V-Hosts unterscheidbar zu bleiben!

Listing 4.3

```

1 Listen 10.2.11.120:80
2
3 NameVirtualHost "10.2.11.120:80"
4
5 <VirtualHost "10.2.11.120:80">
6     DocumentRoot "E:/Apache/httpd/2.4.2/htdocs/selfhtml.it-training.fus"
7     ServerAdmin   root@127.0.0.1
8     ServerName    "selfhtml.it-training.fus"
9 </VirtualHost>
10

```

Der Host-Server antwortet nicht mehr!

Sobald das Name-Based Virtual Hosting aktiviert wurde antwortet der Webserver selbst nicht mehr. Antworten werden nur noch von den virtuellen Servern verschickt!

Um sicherzustellen, dass immer eine Antwort von Ihrem Webserver kommt, sollte ein Default V-Host eingerichtet werden. Für den Default V-Host gelten folgende Regeln:

- Er sollte immer als erstes in der Liste der V-Hosts stehen, so dass er standardmäßig antwortet.
- Die Direktiven `DocumentRoot` und `ServerName` sollten die gleichen Argumente aufweisen, wie die gleichnamigen Direktiven im Hauptserver.

Das der Default V-Host immer als erstes in der Liste der V-Hosts stehen sollte, hängt damit zusammen, wie der Apache Webserver Client-Anfragenabarbeitet. Wenn eine Anfrage eintrifft:

1. Prüft der Apache zuerst, ob die IP-Adresse einer der IP-Adressen entspricht, die in einer `NameVirtualHost`-Direktive angegeben wurden.
2. Wenn dies der Fall ist, wird jeder einzelne `VirtualHost`-Kontainer durchsucht, bis einer gefunden wird, dessen `ServerName`-Direktive dem angeforderten Server entspricht.
3. Ist ein solcher Server gefunden worden, wird dessen Konfiguration verwendet. Falls nicht, wird der erste V-Host in der Liste genutzt, dessen IP-Adresse passend ist!

Wenn zwei virtual Hosts mit gleicher IP-Adresse/Port-Konfiguration auf treten, diese aber unterschiedliche Einträge in der `ServerName`-Direktive haben, wird automatisch das Name-Based VHosting aktiviert. Ein hinzufügen der `NameVirtualHost`-Direktive ist nicht mehr notwendig!

Listing 4.4

```

1 Listen 10.2.11.120
2
3 NameVirtualHost "10.2.11.120:*"
4
5 <VirtualHost "10.2.11.120:*">
6     DirectoryIndex error.html
7     DocumentRoot    "E:/Apache/httpd/2.4.2/htdocs"
8     ServerAdmin      root@10.2.11.120
9     ServerName        "FEA11-119VD02.IT-TRAINING.FUS"
10 </VirtualHost>
11

```

Ein Server - mehrere Namen

Soll ein Name Based Virtual Host unter mehreren DNS-Namen verfügbar sein, kann die ServerAlias-Direktive genutzt werden, um alle gewünschten Namen anzugeben. Alle Aliasnamen müssen auch auf einem DNS-Server eingetragen sein, damit der V-Host erreichbar ist!

Listing 4.5

```

1 NameVirtualHost "10.2.11.120:80"
2
3 <VirtualHost "10.2.11.120:80">
4     DocumentRoot    "E:/Apache/httpd/2.4.2/htdocs/selfhtml.it-training.fus"
5     ServerAdmin      root@10.2.11.120
6
7     # Dieser Server hat zwei zusätzliche Aliasnamen
8     ServerAlias      selfhtml.it-training.de selfhtml.it-training.org
9     ServerName        "selfhtml.it-training.fus"
10 </VirtualHost>
11

```

Umleitung - Redirect

Es gibt verschiedene Gründe, warum der Apache eine Nutzeranfrage auf eine andere URL umleiten muss. In unserem Falle geht es darum, dass der Standard V-Host alle Anfragen auf einen anderen virtuellen Host umleiten soll, so dass immer eine korrekte Antwort vom Webserver geliefert wird.

Dies geschieht am einfachsten mit der Direktive Redirect. Diese hat die folgende Syntax.

Redirect [Status] URL-Pattern URL

Sie nimmt drei Argumente entgegen:

- Status: Hierbei handelt es sich um eine HTTP-Statusmeldung, die dem Client übermittelt wird. Der Status kann sein:
 - permanent: Dies ist die HTTP-Statusmeldung 301, die dem Client mitteilt, dass eine Resource dauerhaft umgezogen ist.
 - temp: Die HTTP-Statusmeldung 302 teilt dem Client mit, dass eine Resource nur zeitweilig umgezogen ist.

Das Apache-Modul `mod_rewrite` ermöglicht ein regelbasiertes Umlenken von Client-Anforderungen auf eine andere URL. Es handelt sich hierbei um ein Modul mit umfassenden

Möglichkeiten und hoher Flexibilität, dass sogar Abfragen auf externe Datenbanksystem unterstützt.

- Konfiguration des Hauptservers:
 - Die HTML-Dokumente des Hauptservers sollen für jeden Host, aus dem Verzeichnis `E:\Apache\httpd\2.4.2\htdocs` abrufbar sein!
 - Der Name des Hauptservers ist "`wwwXX.it-training.fus`", wobei XX Ihre zweistellige Platznummer darstellt (z. B. `www01` oder `www11`)!
 - Der Server beantwortet Anfragen auf Port 80!
 - Das Error-Log heißt `logs\error.log` und muss stündlich ausgetauscht werden.
 - Fehler werden erst ab der Stufe "error" geloggt.
 - Es muss zusätzlich ein Access-Log im CLF-Format konfiguriert werden! Auch hier ist ein stündlicher Wechsel der Logs einzustellen (`logs\access.log`)!
 - Der Server muss mit verschiedenen MIME-Types umgehen können!
- Konfiguration des benannten virtuellen Servers "`wwwXX.ssi.it-training.fus`"
 - Die HTML-Dokumente dieses Servers sollen für jeden Host, aus dem Verzeichnis `E:\Apache\httpd\2.4.2\htdocs\ssi` abrufbar sein!
 - Dieser Server beantwortet Anfragen auf Port 80!
 - Das Error-Log heißt `logs\ssi\error.log` und muss stündlich ausgetauscht werden.
 - Fehler werden erst ab der Stufe "error" geloggt.
 - Es muss zusätzlich ein Access-Log im CLF-Format konfiguriert werden! Auch hier ist ein stündlicher Wechsel der Logs einzustellen (`logs\ssi\access.log`)!
 - Der Server muss mit verschiedenen MIME-Types umgehen können!
 - Die Standardwebsite dieses Servers ist `index.shtml`. Diese soll auch dann geladen werden, wenn der Nutzer keine explizite Angabe einer HTML-Seite gemacht hat!
 - Sollte die Standardwebseite nicht geladen werden können, muss der Nutzer auf die Seite `error.html` umgeleitet werden. Benutzen Sie hierfür eine Standardfehlerseite, so wie Sie sie im Unterricht erstellt haben!
 - Aktivieren Sie SSI für diesen Server! Hinweis: Die HTML-Seite `index.shtml` ist eine HTML-Seite mit SSI-Kommandos. Sie können an dieser Stelle entweder eine neue HTML-Seite entwerfen, oder eine aus den vorangegangenen Übungen nutzen!
- Konfiguration des benannten virtuellen Servers "`wwwXX.selfhtml.it-training.fus`":
 - Die HTML-Dokumente dieses Servers sollen für jeden Host, aus dem Verzeichnis `E:\Apache\httpd\2.4.2\htdocs\selfhtml` abrufbar sein!
 - Dieser Server beantwortet Anfragen auf Port 80!
 - Das Error-Log heißt `logs\selfhtml\error.log` und muss stündlich ausgetauscht werden.
 - Fehler werden erst ab der Stufe "error" geloggt.
 - Es muss zusätzlich ein Access-Log im CLF-Format konfiguriert werden! Auch hier ist ein stündlicher Wechsel der Logs einzustellen (`logs\selfhtml\access.log`)!
 - Der Server muss mit verschiedenen MIME-Types umgehen können!
 - Die Standardwebsite dieses Servers ist `index.htm`. Diese soll auch dann geladen werden, wenn der Nutzer keine explizite Angabe einer HTML-Seite gemacht hat!
 - Sollte die Standardwebseite nicht geladen werden können, muss der Nutzer auf die Seite `error.html` umgeleitet werden. Benutzen Sie hierfür eine Standardfehlerseite, so wie Sie sie im Unterricht erstellt haben!
 - Dieser Server soll ebenfalls unter den Namen "`wwwXX.selfhtml.it-training.org`" und "`wwwXX.selfhtml.it-training.de`" verfügbar sein!
- Konfiguration des benannten virtuellen Servers "`wwwXX.mediawiki.it-training.fus`":

- Führen Sie die Konfiguration dieses Servers mit Hilfe von .htaccess-Dateien durch!
- Die HTML-Dokumente dieses Servers sollen für jeden Host, aus dem Verzeichnis E:\Apache\httpd\2.4.2\htdocs\mediawiki abrufbar sein!
- Dieser Server beantwortet Anfragen auf Port 80!
- Das Error-Log heißt logs\mediawiki\error.log und muss stündlich ausgetauscht werden.
- Fehler werden erst ab der Stufe "error" geloggt.
- Es muss zusätzlich ein Access-Log im CLF-Format konfiguriert werden! Auch hier ist ein stündlicher Wechsel der Logs einzustellen (logs\mediawiki\access.log)!
- Der Server muss mit verschiedenen MIME-Types umgehen können!
- Die Standardwebsite dieses Servers ist index.php. Diese soll auch dann geladen werden, wenn der Nutzer keine explizite Angabe einer HTML-Seite gemacht hat!
- Sollte die Standardwebseite nicht geladen werden können, muss der Nutzer auf die Seite error.html umgeleitet werden. Benutzen Sie hierfür eine Standardfehlerseite, so wie Sie sie im Unterricht erstellt haben!
- Der MediaWiki muss korrekt angezeigt werden!

Listing 4.6

```

1 <VirtualHost "10.2.11.120:8080">
2   Alias "/perl"      "E:/Apache/httpd/2.4.2/perl"
3   DirectoryIndex hallo.html
4   DocumentRoot "E:/Apache/httpd/2.4.2\htdocs"
5   ServerName "FEA11-119VD02.IT-TRAINING.FUS"
6
7   <Directory "htdocs">
8     AddHandler application/x-httpd-php .php
9     Options Includes
10    Require all granted
11  </Directory>
12
13  <Directory "E:/Apache/httpd/2.4.2/perl">
14    Options ExecCGI
15    PerlHandler ModPerl::Registry
16    PerlSendHeader On
17    AddHandler perl-script .pl
18    Require all granted
19  </Directory>
20 </VirtualHost>
21

```

4.2.20 Einfuehrung und Installation

Einführung

Der Apache-Server ist eines der erfolgreichsten freien Software-Projekte in der Geschichte. Man geht derzeit davon aus, dass 54 % aller Websites von einem Apache auf verschiedensten Plattformen ausgeliefert werden. Dabei ist der Apache im Betrieb sehr zuverlässig, performant und durch seine modulare Architektur leicht erweiterbar.

Durch eine große Anzahl von Zusatzmodulen wird der Apache zur geeigneten Plattform für dynamische Inhalte, für sichere Transaktionen oder Workflow-Management. Die Flexibilität im

Aufbau und die Vielzahl der unterschiedlichen Erweiterungen bringen eine hohe Komplexität bezüglich der Konfigurationsdateien mit sich, weshalb hier, in dieser Unterrichtsunterlage, nur ein Teil seiner Features und Module beleuchtet werden kann.

Es gibt verschiedene Versionen des Apache. Am häufigsten werden die Versionen 1.3, 2.0 sowie 2.2 genutzt, wobei zwischen den Versionen 1.3 und 2.0 ein großer Qualitätssprung stattfand. Bei einer Neuinstallation sollte darauf geachtet werden, immer die aktuellste Version, mit ihren aktuellsten Patches, zu nutzen, damit Sicherheitslücken so schnell wie möglich geschlossen werden. Die derzeit aktuellste Version des Apache ist 2.4.2.

Recherchieren Sie im Internet, wo Sie...

- die aktuelle Online-Dokumentation zum Apache HTTP-Server finden.
 - das aktuelle Installationspakete für Windows (inklusive OpenSSL) finden.
- Speichern Sie sich diese Links als Bookmarks auf Ihrem PC!

Installation

Die Echtheit des Installationspaketes überprüfen

Laden Sie die aktuellste Version (2.4.x) des Apache HTTP-Servers für Windows, und die dazugehörige SHA1-Datei herunter! ([Download here](#))

Mit Hilfe der SHA1-Checksumme einer Datei kann deren Echtheit verifiziert werden. Für das Installationspaket wird die Checksumme "64D78A9C90E005E8F4F55F4E1C3720E856BBC005" bereitgestellt. Um eine Überprüfung dieses Pakets durchführen zu können, wird zusätzlich noch eine Software benötigt, die SHA1-Checksummen generieren kann. Eine solche Software ist z. B. "digestIT2004".

Laden Sie die Software "digestIT2004" aus dem Internet herunter ([Download here](#)), und installieren Sie sie (Complete Installation). Führen Sie jetzt eine Verifikation des Apache-Installationspaketes, gemäß der folgenden Anleitung durch!

1. Markieren und Kopieren Sie sich die SHA1-Checksumme (64D78A9C90E005E8F4F55F4E1C3720E856BBC005).
2. Klicken Sie mit der rechten Maustaste auf das Installationspaket. Es erscheint ein Kontextmenü.
3. Klicken Sie auf den Menüeintrag "digestIT 2004" mit der linken Maustaste.
4. Wählen Sie den Menüpunkt "Verify SHA1 Hash"!

Wenn alles ordnungsgemäß verlaufen ist, sollte die Meldung "Digest matches. Verification succeeded." erscheinen, und die Verifikation ist erfolgreich abgeschlossen.

Die Software installieren

Installieren Sie, nach den folgenden Vorgaben, den Apache HTTP-Server auf Ihrem Laptop!

- Network Domain: apache.org
 - Server Name: localhost
 - Administrator's Email Address: Admin@localhost.apache.org
 - Install for all users, on port 80, as a service -- Recommended.
 - Setup Type: Typical
 - Installationspfad: DATENPARTITIONapachehttpd2.4.2
- Ist die Installation erfolgreich verlaufen, erscheint am unteren rechten Bildschirmrand ein kleines Symbol, eine Feder mit einem grünem Pfeil.

Ist die Installation erfolgreich verlaufen, erscheint am unteren rechten Bildschirmrand ein kleines Symbol, eine Feder mit einem grünem Pfeil.



Die Steuerkonsole (Apache Service Monitor)

Mit einem Doppelklick können Sie den Apache Service Monitor öffnen.

Welche Möglichkeiten/Funktionen bietet der Apache Service Monitor?

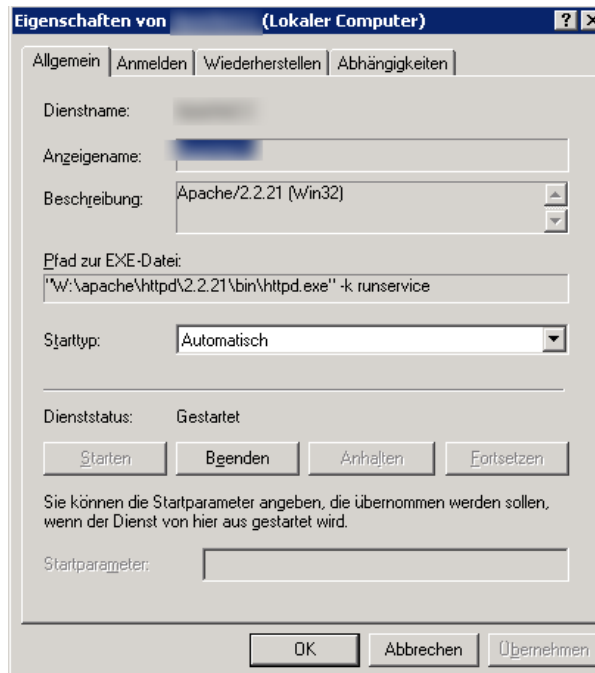
[illegible]

Der Apache in der Windows Dienstkonsole

Wenn der Apache HTTP-Server als Dienst installiert wird, ist er selbstverständlich auch in der Windows Dienstekonsolle zu finden.

Unter welchem Namen ist der HTTP-Server in der Windows Dienstekonsolle zu finden?

Nach einem Doppelklick auf den HTTP-Server Dienst öffnet sich dessen Eigenschaftenfenster.



Achten Sie an dieser Stelle darauf, dass der Apache automatisch, beim Systemstart, mitgestartet wird.

Das Installationsverzeichnis

Nach der Installation des HTTP-Servers finden sich in dessen Installationsverzeichnis diverse Unterverzeichnisse. Diese sind:

Welche Bedeutung haben die unten aufgeführten Unterverzeichnisse?

bin

cgi-bin

conf

error

logs

```
modules
```

```
htdocs
```

4.2.21 Konfiguration

Der erste Blick in die Konfigurationsdatei

Der Apache HTTP-Server wird nicht wie viele andere Windowsprogramme mittels der Windows Registry, sondern mit Hilfe einer eigenen Textdatei konfiguriert. Üblicher weise heißt diese Datei `httpd.conf`. Sie kann jedoch auch einen beliebigen anderen Namen tragen und durch zusätzliche Dateien erweitert werden. Zu finden ist diese Datei im Unterverzeichnis "conf" des Installationsverzeichnis.

Fertigen Sie jetzt eine Sicherheitskopie der Datei `httpd.conf` an!

Direktiven

Zur Konfiguration des Apache werden sogenannte "Direktiven" in die Datei `httpd.conf` eingetragen. Eine solche Direktive ist wie folgt aufgebaut:

```
Direktive  Argument
ServerRootE:/apache/httpd/2.4.2
```

Für die Erstellung von Direktiven gelten die folgenden Regeln:

- Nur eine Direktive pro Zeile
- Die Direktiven sind nicht Casesensitiv, d. h. `ServerRoot` = `serverroot`.
- Die Argumente der Direktiven sind Casesensitiv.
- Einrückungen können verwendet werden, um die Lesbarkeit der Konfigurationsdatei zu verbessern.
- Der Backslash (`\`) kann verwendet werden, um sehr lange Zeilen in mehrere Zeilen zu verteilen (Linebreak char).
- Bei der Angabe von Datei- und Verzeichnispfaden wird immer der Slash (`/`) als Trennzeichen verwendet, nicht der Backslash (`\`).

Kommentare

Recherchieren Sie in der Onlinedokumentation des Apache HTTP-Server, wie Kommentare in die Konfigurationsdatei eingetragen werden, bzw. welche Regeln dafür gelten!

- Fügen Sie die DocumentRoot-Direktive in Ihre Konfigurationsdatei ein, und verweisen Sie mit ihr auf das htdocs-Verzeichnis (z. B. E:/apache/httpd/2.4.2/htdocs)!
- Testen Sie erneut den Zugriff auf <http://localhost> mit Hilfe Ihres Browsers! Welche Antwort erhalten Sie jetzt?
- Versuchen Sie jetzt, mit Hilfe des Internet Explorers, auf die Adresse <http://localhost/index.html> zuzugreifen! Welche Antwort erhalten Sie von Ihrem Browser?
- Versuchen Sie erneut auf die Adresse <http://localhost/index.html> zuzugreifen, dieses mal aber mit Hilfe des FireFox! Wie unterscheiden sich die Antworten des FireFox und des Internet Explorers von einander?

Weitere wichtige Direktiven

Mit den beiden Direktiven `Listen` und `DocumentRoot` hat der Apache HTTP-Server bereits alle Informationen, die er für einen rudimentären Betrieb benötigt. Für den produktiven Betrieb eines Apache HTTP-Servers genügen diese Angaben jedoch noch nicht. Es existieren weitere Direktiven, die zur Mindestkonfiguration dieses Webserver gehören. Die folgende Auflistung zeigt einige davon.

Direktive	Bedeutung
DefaultType	
ErrorLog	
LogLevel	

DefaultType

LogLevel

- Recherchieren Sie in der Onlinedokumentation (Dynamic Shared Objects), wie das Laden von Modulen funktioniert (Syntax)!
- Recherchieren Sie in der Dokumentation was die Direktive `DirectoryIndex index.html` bedeutet!
- Welches Modul muss geladen werden, damit die `DirectoryIndex`-Direktive funktioniert?
- Laden Sie das betreffende Modul, tragen Sie diese Direktive in Ihre Datei `httpd.conf` ein und prüfen Sie was passiert, wenn Sie die URL `http://localhost`, nach einem Webserverneustart, aufrufen!
- Was passiert, wenn Sie die folgende URL in Ihrem Browser aufrufen: `http://localhost/login.php`?
- Lesen Sie in der Dokumentation über die Direktive `FallbackResource` nach, und benutzen Sie diese, um im Falle eines HTTP-404-Fehlers auf eine Fehlerseite zu verweisen!
- Worin besteht der wesentliche Unterschied zwischen der `FallbackResource`-Direktive und der Direktive `ErrorDocument`?
- Konfigurieren Sie Ihren Apache jetzt so, dass er statt der `FallbackResource`-Direktive die `ErrorDocument`-Direktive nutzt, um im Falle eines HTTP-404-Fehlers die Fehlerseite zu zeigen!

Bedingtes laden von Modulen

In manchen Umgebungen kann es notwendig sein, dass ein Apache-Modul nur dann geladen werden darf, wenn eine bestimmte Bedingung erfüllt ist. Um dies zu erreichen, muss der `IfModule`-Kontainer eingesetzt werden.

Ein Kontainer besteht aus einem öffnenden Tag (z. B. `IfModule [Bedingung]`) und einem schließenden Tag (`/IfModule`). Beide Tags umschließen Direktiven, die nur dann ausgeführt werden, wenn der betreffende Kontainer bei einer Anfrage angesprochen wird.

Im konkreten Falle des `IfModule`-Kontainers heißt das, dass die in einem solchen Kontainer eingeschlossenen Direktiven nur dann ausgeführt werden, wenn `[Bedingung]` zutrifft. Das Beispiel auf dieser Seite zeigt den `IfModule`-Kontainer, zusammen mit der `DirectoryIndex`-Direktive. Diese wird nur dann ausgeführt, wenn das Modul `dir_module` geladen wurde.

Listing 4.7

```

1 <IfModule dir_module>
2   DirectoryIndex index.html
3 </IfModule>
```

4.2.22 Sicherheit

Directory-Kontainer

Directory-Kontainer stellen einen zentralen Sicherheitsmechanismus innerhalb der Konfiguration des Apache HTTP-Servers dar. Mit ihrer Hilfe kann der Zugriff auf Verzeichnisse, die über den Webserver erfolgen, reguliert werden. Wie bereits beim `IfModule`-Kontainer besprochen, gelten auch hier die Anweisungen innerhalb des Containers nur für das angegebene Verzeichnis.

```
1 <Directory />
2     AllowOverride none
3     Require all denied
4 </Directory>
5
```

In machen Situationen kann es notwendig sein, dass die Konfiguration der Zugriffsrechte eines Verzeichnisses nicht durch den Administrator des Webserver, sondern durch eine andere Person erfolgen muss. In so einem Fall gibt es zwei Optionen:

- Der Serveradministrator gibt der betreffenden Person Zugriff auf die Hauptkonfiguration des Webservers (httpd.conf).
- Es werden .htaccess-Dateien genutzt (man beachte den Punkt vor dem Namen).

Im Allgemeinen dürfte die zweite Option, die Nutzung der .htaccess-Dateien, die sinnvollere Alternative sein.

Unter MS-Windows ist es nicht möglich, dass ein Dateiname mit einem Punkt beginnt. Deshalb muss hier die AccessFile-Direktive in der Hauptkonfigurationsdatei genutzt werden, um den Namen der .htaccess-Dateien zu ändern, z. B. in htaccess (ohne Punkt)!

Benutzung von .htaccess-Dateien

.htaccess-Dateien ermöglichen es, den Zugriff auf die Verzeichnisstruktur des Webservers auf der Basis einzelner Verzeichnisse zu regeln. Dies geschieht, indem .htaccess-Dateien in die Verzeichnisse gelegt werden, für die Zugriffsbeschränkungen eingerichtet werden müssen.

Die Direktiven einer .htaccess-Datei gelten immer für das Verzeichnis, in dem sie sich befindet, und alle darunter liegenden Unterverzeichnisse.

.htaccess-Dateien nutzen die gleiche Syntax, wie die Hauptkonfigurationsdatei httpd.conf. So kann z. B. der Directory-Kontainer für das Verzeichnis htdocs/selfhtml durch eine entsprechende .htaccess-Datei ersetzt werden.

Innerhalb der .htaccess-Datei entfällt die Angabe des Directory-Kontainers, da der gesamte Inhalt der Datei nur für ein Verzeichnis gültig ist.

Testen Sie, ob die auf dieser Seite gezeigte .htaccess-Datei bereits Auswirkungen hat! Wenn nicht, was muss an der Hauptkonfigurationsdatei geändert werden, um die .htaccess-Datei wirksam werden zu lassen?

Bei der Nutzung von .htaccess-Dateien treten im Wesentlichen drei negative Effekte auf:

- Direktiven aus verschiedenen .htaccess-Dateien werden immer in der Reihenfolge angewandt, in der sie vom Webserver vorgefunden werden. D. h. Direktiven aus .htaccess-Dateien in den unteren Verzeichnisebenen können Direktiven aus .htaccess-Dateien den höheren Verzeichnisebenen überschreiben.

Es existieren die folgenden Verzeichnisse: `htdocs/selfhtml` und `htdocs/selfhtml/css`. Beide Verzeichnisse haben eine eigene `.htaccess`-Datei. In der `.htaccess`-Datei des Verzeichnisses `htdocs/selfhtml` ist die Direktive `Require all granted` angegeben. Das bedeutet, dass ein Nutzer Zugriff auf beide Verzeichnisse hat, da die Direktiven aus `.htaccess`-Dateien auch immer für alle Unterverzeichnisse gelten.

Im Verzeichniss `htdocs/selfhtml/css` ist jedoch in der `.htaccess`-Datei die Direktive `Require all granted` gesetzt. Daraus folgt, dass der Zugriff auf das Verzeichnis `htdocs/selfhtml/css` für alle Nutzer gesperrt ist, weil die `Require all granted`-Direktive als zweites gefunden wird, und somit Gültigkeit hat. Nutzer können also nur auf das Verzeichnis `htdocs/selfhtml` zugreifen.

1. Wie verhält es sich zwischen einer `.htaccess`-Datei und der Hauptkonfigurationsdatei? Kann eine `.htaccess`-Datei Direktiven aus der Hauptkonfigurationsdatei überschreiben, und umgekehrt?
2. Was passiert, wenn ein Benutzer, in seinem Browser, die folgende URL aufruft: `http://localhost/selfhtml/htaccess` oder `http://localhost/selfhtml/.htaccess`?
3. Wie kann verhindert werden, dass sich ein Nutzer den Inhalt einer `.htaccess`-Datei anzeigen lässt (Hinweis: Schlagen Sie hierfür die `FilesMatch`-Direktive nach!)

Listing 4.9

```

1 # httpd.conf
2 <Directory "htdocs/selfhtml">
3     Require all granted
4 </Directory>
5
6 #.htaccess-Datei
7     Require all granted

```

Zugriffskontrolle durch Autorisierung

Autorisierung mit Hilfe von Ausdrücken

Das bereits bekannte Kern-Autorisierungsmodul `authz_core_module` stellt den Grundbaustein für alle Autorisierungsmechanismen des Apache HTTP-Server dar. Neben den bereits bekannten Optionen `Require all granted` und `Require all denied` bietet es noch die Möglichkeit Umgebungsvariablen auszuwerten und abhängig vom Ergebnis einer solchen Auswertung den Zugriff zu gewähren.

Nutzen Sie die Apache Onlinedokumentation (<http://httpd.apache.org/docs/2.4/expr.html>), um die folgenden Aufgaben zu erledigen:

- Das Verzeichnis `htdocs` darf nur zwischen 07:00 Uhr und 16:00 Uhr für die Nutzer verfügbar sein.
- Legen Sie die beiden Verzeichnisse `htdocs/msie` und `htdocs/mozilla` an.
- Auf das Verzeichnis `htdocs/msie` dürfen nur solche Nutzer Zugriff haben, welche einen Internet Explorer als Browser benutzten (Variable `HTTP_USER_AGENT`).
- Auf das Verzeichnis `htdocs/mozilla` dürfen nur solche Nutzer Zugriff haben, welche den Browser Mozilla Firefox oder Google Chrome benutzen.
- Zugriffe auf das `htdocs`-Verzeichnis, die mit Hilfe veralteter Browserversionen erfolgen (MSIE 6.x oder Firefox 3.x) sollen direkt abgewiesen werden!

Zusätzlich zu dem Modul `authz_core_module` stellt der Apache noch weitere Module bereit, welche unterschiedlichste Autorisierungsmechanismen zur Verfügung stellen.

authnz_ldap_module	Modul zur Authentifizierung und Autorisierung mittels eines LDAP-Dienstes, z. B. Microsoft Active Directory
authz_dbd_module	Ermöglicht die Autorisierung mit Hilfe einer SQL-fähigen Datenbank z. B. MySQL
authz_dbm_module	Dieses Modul benutzt Nutzernamen und Passwörter, welche in DBM-Dateien gespeichert sind.
authz_groupfile_module	Liest Informationen über Gruppenzugehörigkeiten aus Klartextdateien und benutzt diese zur Autorisierung.
authz_host_module	Autorisierung mittels IP-Adresse oder Hostname
authz_owner_module	Ermöglicht die Zugriffssteuerung auf Dateien mit Hilfe der Besitzrechte.
authz_user_module	Ein Autorisierungsmodul, welches basierend auf dem Nutzernamen eine Autorisierung durchführt.

Autorisierung mittels IP-Adresse/Hostname

authz_host_module stellt die Möglichkeit bereit eine Autorisierung mittels IP-Adresse oder Hostname durchzuführen.

authz_host_module ist kein Ersatz für eine Firewall!

- Kommentieren Sie die Zugriffsregeln aus der letzten Übung aus!
- Konfigurieren Sie den Zugriff auf das htdocs-Verzeichnis so, das ihr eigener Client (Windows 7) keinen Zugriff mehr hat.
- Ändern Sie die Autorisierung für das htdocs-Verzeichnis so, das nur noch lokale Verbindungen zugreifen dürfen!
- Verboten Sie jetzt alle Verbindungen aus Ihrem eigenen IP-Subnetz (10.2.11.64/26)!
- Verändern Sie die Autorisierung des htdocs-Verzeichnisses erneut, so das alle Computer, die Mitglied in der Domäne IT-TRAINING.FUS sind, zugriff haben.

Zugriffsbedingungen kombinieren

Benutzen Sie die Apache Onlinedokumentation, um herauszufinden welche Bedeutung die drei Direktiven RequireAll, RequireAny und RequireNone haben!

Richten Sie die folgenden Zugriffsregeln ein:

- Alle Computer, die der Domäne IT-TRAINING.FUS angehören, dürfen zwischen 7:00 Uhr und 16:00 Uhr auf das Verzeichnis htdocs zugreifen. Computer, welche dieser Domäne nicht angehören, dürfen nur zwischen 8:00 Uhr und 12:00 Uhr zugreifen.
- Alle Computer, die sich im IP-Subnetz des Webserver befinden (10.2.11.64/26) dürfen mit dem Mozilla FireFox auf das Verzeichnis htdocs/selfhtml zugreifen. Alle Computer die sich nicht in diesem IP-Subnetz befinden dürfen zusätzlich zum FireFox auch den Browser Internet Explorer nutzen!

Was bewirken die unten angegebenen Konfigurationen?

Listing 4.10

```
1 # Konfiguration 1
2   Require all denied
3   Require local
4
5 # Konfiguration 2
6   <RequireAny>
7     <RequireAll>
8       Require %{TIME_HOUR} -gt 7
9     </RequireAll>
10  </RequireAny>
11  Require local
```

Abwärtskompatibilität

Im Apache HTTP-Server wurden bis zur Version 2.2 die Direktiven Allow From und Deny From, bzw. Order, statt der Require-Direktive eingesetzt. Um abwärtskompatibel zu bleiben, hat der Apache in der Version 2.4 ein Modul namens mod_access_compat. Dies enthält die oben genannten Direktiven, so dass diese auch weiterhin noch eingesetzt werden können.

Das Modul "mod_autoindex.so" bietet viele verschiedene Möglichkeiten, um einen einfachen, schmucklosen Verzeichnisindex ansprechender zu gestalten. Eine dieser Möglichkeiten ist beispielsweise das hinzufügen von Icons. Hierzu muss jedoch zu erst die Option FancyIndexing aktiviert werden.

Seite 73 von 115

```
1 <Directory "htdocs/download">
2     Require all granted
3
4     # FancyIndexing aktivieren
5     IndexOptions FancyIndexing
6
7     Options          Indexes
8 </Directory>
9
```

Seite 74 von 115

Nehmen Sie die Direktiven aus `ispiel{TypesConfig}` in Ihre Hauptkonfigurationsdatei auf, und benutzen Sie die Direktive `AddIconByType`, um das Icon "compressed.gif" für ZIP-Dateien festzulegen! Welchen MIME-Type benötigen Sie hierfür? Testen Sie Ihr Ergebnis!

```
1 LoadModule mime_module "modules/mod_mime.so"
2 TypesConfig "conf/mime.types"
```

Verzeichnisalias ermöglichen es, Ressourcen außerhalb des DocumentRoot-Verzeichnisses einzubinden. Zuständig ist hierfür das "alias module" (mod alias.so).

Seite 76 von 115

Worin unterscheiden sich die beiden folgenden Directory-Kontainer (Hinweis: Benutzen Sie Ihr Error-Log, um den Unterschied festzustellen)?

- Als Icon für das übergeordnete Verzeichnis soll dir.gif verwendet werden!
- Für *.msi-Dateien soll das Icon compressed.gif genutzt werden.
- Das Standardicon für alle unbekannten Dateitypen muss unknown.gif sein.
- Im Index sollen Verzeichnisse immer an erster Stelle aufgelistet werden, und dann erst die Dateien.
- Die Icons müssen ein Teil der Links sein.
- Dateinamen müssen in ihrer vollen Länge erhalten bleiben und dürfen nicht abgeschnitten werden.
- Die Spalte "Description" soll ausgeblendet werden.
- Die Standardsortierung für den Verzeichnisindex ist eine aufsteigende Reihenfolge.

```
1 <Directory "/icons">
2     Require all granted
3 </Directory>
4
5 <Directory "E:/apache/httpd/2.4.2/icons">
6     Require all granted
7 </Directory>
8
```

Was sind Server Side Includes?

HF Weidinger - L Siegerth

ITSysAdminFwWebSK

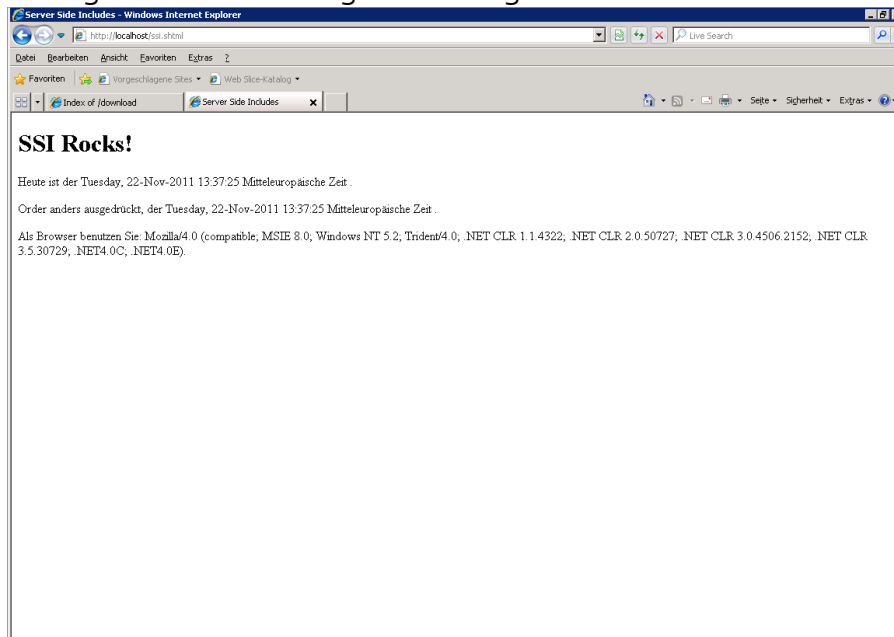
Seite 79 von 115

- Fügen Sie die beiden Direktiven `AddType` und `AddOutputFilter` Ihrer Hauptkonfigurationsdatei hinzu! Was bedeuten sie?
- Laden Sie das Modul `include_module`!

Listing 4.14

Seite 81 von 115

Ein Zugriff auf die obige HTML-Datei zeigt dieses Ergebnis:



- Testen Sie den Zugriff auf ssi.shtml !
- Zusätzlich zur Option Includes gibt es noch die Option IncludesNOEXEC ! Worin unterscheiden sich die beiden?

Listing 4.15

```

1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
2      "http://www.w3.org/TR/html4/strict.dtd">
3  <html>
4      <head>
5          <title>Server Side Includes</title>
6      </head>
7      <body>
8          <h1>SSI Rocks!</h1>
9          <p>Heute ist der <!--#echo var="DATE_LOCAL" -->.</p>
10         <!--#config timefmt="%d.%m.%Y" -->
11         <p>Oder anders ausgedrückt, der <!--#echo var="DATE_LOCAL" -->.</p>
12         <p>Als Browser benutzen Sie: <!--#echo var="HTTP_USER_AGENT" -->.</p>
13     </body>
14 </html>
15

```

HTML

4.2.25 CGI

PERL over CGI

Das Common Gateway Interface stellt eine Schnittstelle zwischen dem Apache HTTP-Server, und externen Programmen, die den Bildschirminhalt generieren, dar. Diese externen Programme werden auch oft als CGI-Programme oder CGI-Skripte bezeichnet. Mit CGI kann auf eine sehr einfache Art und Weise dynamischer Content erzeugt werden.

Ein Zitat aus <http://de.wikipedia.org/wiki/PERL>

Perl ist eine freie, plattformunabhängige und interpretierte Programmiersprache (Skriptsprache), die mehrere Programmierparadigmen unterstützt.

Entworfen worden ist diese Sprache 1987, von Larry Wall, einem Linguisten. Sie sollte als Werkzeug zur Analyse und Verarbeitung von Textdateien dienen. Da sie seit vielen Jahren eine weite Verbreitung in der Welt des Internets gefunden hat, soll sie auch hier, für die Demonstration von CGI-Programmen dienen.

Im Internet existieren zwei große Distributionen von Perl:

- Strawberry Perl
- Active Perl

Bei Active Perl handelt es sich um die Perl-Variante der Firma Activestate. Es existiert in den folgenden Versionen:

- Community Edition
- Business Edition
- Enterprise Edition

Lediglich die Community Edition ist als kostenlose Version erhältlich, darf jedoch nur zu Testzwecken und für den nicht kommerziellen Einsatz genutzt werden. Strawberry Perl hingegen ist eine Open Source Implementierung der Sprache Perl, die ohne Einschränkungen durch eine Lizenz genutzt werden darf. Im folgenden wird in dieser Unterrichtsunterlage Strawberry Perl zum Einsatz kommen.

Laden Sie Strawberry Perl in der aktuellsten Version aus dem Internet herunter und installieren Sie es! Als Installationsverzeichnis soll E:perlstrawberry5.14.2.1 dienen!

CGI mit ScriptAlias konfigurieren

Um den Apache HTTP-Server dazu überreden zukönnen das Common Gateway Interface zu aktivieren, ist es notwendig das cgi_module (mod_cgi.so) zu laden. Da es sich bei CGI-Skripten um ausführbare Programme handelt, sollten diese nicht mit anderen Dateien, wie z. B. HTML-Dateien vermischt, sondern in einem separaten Verzeichnis aufbewahrt werden. Um dieses Verzeichnis für Nutzer zugänglich zu machen, und gleichzeitig die Ausführung der CGI-Programme zu erlauben, wird die ScriptAlias-Direktive verwendet.

1. Schlagen Sie die Syntax der ScriptAlias-Direktive in der Onlinedokumentation nach!
2. Versehen Sie das Verzeichnis [ServerRoot]/cgi-bin mit dem ScriptAlias cgi-bin Hinweis: [ServerRoot] stellt einen Platzhalter für Ihr Serverroot-Verzeichnis, z. B. E:/apache/httpd/2.4.2, dar!
3. Welcher Schritt muss noch unternommen werden, damit CGI-Skripte aus dem Verzeichnis cgi-bin ausgeführt werden können?

Listing 4.16

```
1 LoadModule cgi_module "modules/mod_cgi.so"
2
```

CGI mit einem Alias konfigurieren

Eine andere Möglichkeit CGI zu aktivieren besteht darin, dass die Option ExecCGI, in Zusammenhang mit einem Verzeichnisalias und einem Actionhandler genutzt wird.

Legen Sie das Verzeichnis E:/apache/httpd/2.4.2/cgi-bin_2 an!

Analog zur ersten Variante, muss statt einem ScriptAlias, ein Alias angelegt werden. Im zweiten Schritt, ist es notwendig einen Verzeichniskontainer für das cgi-bin_2-Verzeichnis anzulegen.

Zeile zwei, AddHandler cgi-script .cgi .pl bewirkt, dass alle Dateien, mit den Endungen .cgi oder .pl als CGI-Skripte betrachtet und ausgeführt werden. Zeile Nummer 7, Options ExecCGI hat zur Folge, dass die Ausführung von CGI-Skripten erlaubt wird. Durch Entfernung dieser Option kann die Ausführung von CGI-Skripten unterbunden werden, ohne dass die restliche Konfiguration dazu verändert werden müsste.

Listing 4.17

```

1  Alias "/cgi-bin_2" "E:/apache/httpd/2.4.2/cgi-bin_2"
2
3  <Directory "E:/apache/httpd/2.2.21/cgi-bin_2">
4      #Hinzufuegen eines Actionhandlers fuer CGI
5      AddHandler      cgi-script .cgi .pl
6
7      AllowOverride None
8      Require all granted
9
10     #Die Ausfuehrung von CGI-Skripten zulassen
11     Options          ExecCGI
12 </Directory>
13
```

Jetzt wird es spannend - Das erste Perl-Programm

Wenn es so etwas wie eine Tradition im Bereich der Programmierung gibt, dann die, dass das erste Programm eines jeden Programmiers die Worte "Hallo Welt!" auf dem Bildschirm ausgibt. Da diese Unterrichtsunterlage keinesfalls mit dieser Tradition brechen möchte, wird sie auch hier fortgesetzt:

- Erstellen Sie das unten abgebildete PERL-Programm und speichern Sie es als Datei, im Verzeichnis E:/apache/httpd/2.4.2/cgi-bin, mit dem Namen hallo_welt.pl ab.
- Testen Sie das Programm, in dem Sie folgende URL aufrufen:

http://localhost/cgi-bin/hallo_welt.pl

Es sollte der Schriftzug "Hallo Welt!" erscheinen!

Das Programm hallo_welt.pl ist wie folgt aufgebaut:

Die erste Zeile ist eine spezielle Kommentarzeile, zu erkennen an den Zeichen #!. Sie hat die Aufgabe, dem Browser mitzuteilen, wo sich der Kommandointerpreter, in diesem Falle die Datei perl.exe, befindet.

Die zweite Zeile benutzt das print-Kommando, um den Contenttype und den für die Ausgabe zu verwendenden Zeichensatz an den Browser weiterzugeben. Der Browser unterdrückt die Ausgabe dieses Textes auf dem Bildschirm, verarbeitet aber trotzdem die Informationen.

Die dritte Zeile ist die alles entscheidende. Sie sorgt für die Bildschirmausgabe von "Hallo Welt!".

Listing 4.18

```
1 #!"E:/perl/strawberry/5.14.2.1/perl/bin/perl.exe"  
2 print "Content-type: text/plain; charset=iso-8859-1";  
3 print "Hallo Welt!";  
4
```

Die Spannung steigt - Das zweite Perl-Programm

Das zweite Versuch mit CGI ein PERL-Programm aufzurufen soll etwas komplexer gestaltet werden. Im ersten Schritt wird eine HTML-Datei erstellt, die das PERL-Programm mittels eines SUBMIT-Buttons aufruft.

Erstellen Sie die unten abgebildete HTML-Datei begruessung.html und legen Sie sie im Verzeichnis htdocs ab!

Das Beispiel enthält drei Tags, die für die Dateneingabe und -Verarbeitung zuständig sind:

`<input type="text" name="name">` erzeugt ein Textfeld, in das der Nutzer einen beliebigen Freitext eintragen kann. In unserem Beispiel soll hier der Vor-/Nachname eines Benutzers eingetragen werden.

`<input type="submit" value="Begrüßen">` erstellt einen Button mit der Aufschrift "Begrüßen". Hierbei handelt es sich um einen sogenannten "SUBMIT-Button", was bedeutet, dass beim Drücken des Buttons alle Daten aus dem HTML-Formular an ein Ziel übertragen werden. In diesem Falle ist das PERL-Skript `hallo_formular.pl` das Ziel.

`<form action="cgi-bin/hallo_formular.pl" method="POST">` legt fest, dass durch das Drücken eines SUBMIT-Buttons die HTTP-Methode POST genutzt wird, um alle Formulardaten an das PERL-Skript `hallo_formular.pl` zu übertragen.

Erstellen Sie die Datei `hallo_formular.pl` und speichern Sie sie im Verzeichnis `cgi-bin_2` ab. Testen Sie ob HTML-Formular und PERL-Skript funktionieren!

Listing 4.19



```

1  #begrueßung.html
2  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
3      "http://www.w3.org/TR/html4/strict.dtd">
4  <html>
5      <head>
6          <title>Ein HTML-Formular mit PERL</title>
7      </head>
8      <body>
9          <h1>Sei gegrÄsst!</h1>
10         <form action="cgi-bin/hallo_formular.pl" method="POST">
11             <div>
12                 <p>Sag mir deinen Namen!</p>
13                 <input type="text" name="name">
14             </div>
15             <div>
16                 <input type="submit" value="BegrÄssen">
17             </div>
18         </body>
19     </html>
20
21  #hallo_formular.pl
22  #!"E:/perl/strawberry/5.14.2.1/perl/bin/perl.exe"
23
24  print "Content-type: text/plain; charset=iso-8859-1
25
26  ";
27
28  read (STDIN, $query, $ENV{CONTENT_LENGTH});
29
30  print ("Hallo ");
31  print (split(/./, $query, 0));
32  print ("", wie geht es Dir?");
33

```

4.2.26 PERL mit mod_perl.so

Konfigurieren von mod_perl.so

Eine zweite Variante PERL in den Apache HTTP-Server einzubinden, stellt das Modul mod_perl.so dar. Dieses Modul sorgt genauso wie mod_cgi.so für die Ausführung von PERL-Programmen, aber mit dem Vorteil, dass es nur PERL-Programme, und keine anderen Skripte/Programme zulässt. Dadurch kann verhindert werden, das "aus Versehen" andere Programme, wie z. B. PHP- oder C++ Programme ausgeführt werden.

Als erstes muss die Installation von mod_perl.so erfolgen. Diese geschieht mit dem PERL eigenen Installationsprogramm "pip" (Perl Installation Program).

1. Öffnen Sie eine Windows-Command-Shell (cmd.exe)

2. Setzen Sie, mit Hilfe des set-Kommandos, die Umgebungsvariable HTTP_PROXY auf den folgenden Wert:

```
http://username:passwort@192.168.4.250:3128
```

3. Führen Sie das folgende PIP-Kommando aus, um zusätzliche Dateien für das Modul mod_perl.so zu installieren

```
pip http://strawberryperl.com/package/kmx/mod_perl/5.12_x64/  
mod_perl-2.0.4-MSWin32-x86-multi-thread-5.12.par
```

4. Führen Sie das folgenden PIP-Kommando aus, um zusätzliche Dateien für das das Modul libapreq zu installieren:

```
pip http://strawberryperl.com/package/kmx/mod_perl/5.12_x86/  
libapreq2-2.12-MSWin32-x86-multi-thread-5.12.par
```

5. Kopieren Sie die folgenden Dateien in das Verzeichnis [ServerRoot]/modules:

- mod_perl.so
- mod_apreq2.so
- libapreq2.dll
- Konfigurieren Sie den Apache so, dass die folgenden beiden Module geladen werden:
 - perl_module (mod_perl.so)
 - apreq_module (mod_apreq2.so)
- Legen Sie das Verzeichnis [ServerRoot]/perl an.
- Kopieren Sie das PERL-Skript "hallo_formular.pl" in das Verzeichnis perl, und entfernen Sie die erste Zeile (Kommentarzeile) aus dieser Datei.
- Passen Sie die Datei begruessung.html so an, dass das PERL-Skript aus dem Verzeichnis perl genutzt wird.
- Konfigurieren Sie in Ihrer Hauptkonfigurationsdatei das Verzeichnis perl so, wie in unten angegeben!
- Legen Sie einen Alias für das Verzeichnis perl an!
- Starten Sie den Apache Webserver neu!
- Testen Sie, ob begruessung.html funktioniert!

Hier werden drei Direktiven verwendet, die für die Ausführung von PERL-Programmen verantwortlich sind. Zwei davon, sind neu.

Die Direktive PerlHandler ModPerl::Registry stellt die Verbindung zwischen mod_perl.so und dem PERL-Skript her, in dem ein sogenannter "Perlhandler" konstruiert wird. Dieser ermöglicht es mod_perl.so dann, das Skript aufzufinden und auszuführen.

Die zweite Direktive ist PerlSendHeader On. Sie sorgt dafür, dass mod_perl.so jedes PERL-Programm nach HTTP-Header-Zeilen durchsucht und diese ausführt. Im Falle des vorangegangenen HalloWelt-Beispiels ist dies die Zeile:

```
print "Content-type: text/plain;charset=iso-8859-1 ";
```

Testen Sie was passiert, wenn Sie die Direktive PerlSendHeader auf den Wert "Off" einstellen!

Listing 4.20

```
1 <Directory "E:/apache/httpd/2.4.2/perf">
```

```
1 <Directory E:/apache/httpd/2.4.2/perl >
2     Options          ExecCGI
3     PerlHandler      ModPerl::Registry
4     PerlSendHeader   On
5     AddHandler        perl-script .pl
6     Require           all granted
7 </Directory>
8
```


4.2.27 Apache und PHP

PHP installieren und konfigurieren



Eine weitere Programmiersprache die im Verlauf der Jahre große Verbreitung im Internet gefunden hat ist PHP. Die Abkürzung PHP stellt ein rekursives Akronym (Ein Akronym ist dann rekursiv, wenn das Akronym selbst in der Bedeutung vorkommt!), mit der Bedeutung "PHP Hypertext Preprocessor" dar. Erschaffen wurde PHP im Jahr 1995 von Rasmus Lerdorf, der es später mit den beiden Softwareentwicklern Andi Gutmans und Zeev Suraski gemeinsam weiterentwickelte.

Im Laufe seiner Entwicklung wurde PHP stark von den Sprachen Perl, C, C++ und Java beeinflusst, weshalb an einigen Stellen die Syntax eine hohe Ähnlichkeit mit diesen Sprachen hat. Da PHP auf fast 75 % aller Webseiten zur Erzeugung/Darstellung von dynamischen Inhalten genutzt wird, ist es zum de facto Standard im Internet geworden. Aktuell läuft die Entwicklung an der Version 5.4.4 (Stand 14.06.2012).

Unter der Adresse <http://windows.php.net/download/> können alle aktuellen PHP-Versionen für Microsoft Windows heruntergeladen werden.

- Laden Sie PHP 5.4.4 VC9 x86 Thread Safe als Zip-Datei herunter!
- Entpacken Sie die Zip-Datei in das Verzeichnis E:\PHP
- Legen Sie eine Kopie der Datei php.ini-production an und benennen Sie diese um in: php.ini!
- Laden Sie die Datei php5apache2_2.dll als Modul für Apache!
- Fügen Sie in Ihrer Hauptkonfigurationsdatei, dem Verzeichniskontainer für htdocs, die Direktive AddHandler application/x-httpd-php .php hinzu!
- Starten Sie den Apache neu!
- Erstellen Sie die unten angegebene PHP-Datei, und speichern Sie diese mit dem Namen index.php im Verzeichnis [ServerRoot]/htdocs!
- Rufen Sie die Datei index.php

im Browser auf!

Listing 4.21

```
1 <?php
2     phpinfo();
3 ?>
```

PHP

PHP + MySQL + Mediawiki

Diese Sektion soll ein einfaches und doch realistisches Beispiel dafür geben, welchen Einsatzzweck PHP in der Praxis hat. Mit Hilfe des Datenbankmanagementsystems MySQL und der PHP-Software MediaWiki wird auf dem Apache Webserver ein kleines Wiki-System erzeugt. Da an dieser Stelle der Einsatz von PHP im Vordergrund steht, werden die Installationen von MySQL und MediaWiki nur rudimentär erläutert.

MySQL herunterladen und installieren

MySQL ist ein relationales Datenbankmanagementsystem, das als Grundlage für sehr viele dynamische Webauftritte dient. Ursprünglich wurde es von dem schwedischen Unternehmen MySQL AB entwickelt, dass jedoch im Jahr 2008 von Sun Microsystems aufgekauft wurde. 2010 wurde die Firma Sun Microsystems schließlich von Oracle aufgekauft.

MySQL existiert sowohl als kommerzielle Software, als auch als Open Source Produkt, in der Community Edition. Diese steht unter dem folgenden Link zum Download zur Verfügung:

<http://www.mysql.de/downloads/mysql>

Laden Sie MySQL, als Windows MSI-Installer von der obigen Adresse herunter und installieren Sie es mit den folgenden Angaben!

- Installationstyp: Complete
- Haken setzen bei: Launch the MySQL Instance Configuration Wizard
- Detailed Configuration
- Server Machine
- Multifunctional Database
- InnoDB Tablespace Settings: E:MySQLDataFiles
- Online Transaction Processing (OLTP)
- Haken setzen bei: Enable TCP/IP Networking (Port: 3306)
- Haken setzen bei: Enable Strict Mode
- Zeichensatz: Standard Character Set
- Haken setzen bei: Install as Windows Service (Servicename: MySQL)
- Haken setzen bei: Launch the MySQL Server automatically
- Haken setzen bei: Include Bin Directory
- Haken setzen bei: Modify Security Settings (root-Passwort: password)

Nach erfolgter Installation sollte MySQL als Windows Dienst, unter dem Namen glqq MySQLgrqq laufen.

MediaWiki installieren

MediaWiki wird unter <http://www.mediawiki.org/wiki/Download/de> als Archivdatei, im Format *.tar.gz, zum herunterladen zur Verfügung gestellt.

- Laden Sie MediaWiki von der obigen Adresse herunter und entpacken Sie den Inhalt des Archives nach E:apachehttpd2.4.2htdocs, so dass dort das Verzeichnis mediawiki-1.x.x entsteht!
- Benennen Sie das Verzeichnis mediawiki-1.x.x in mediawiki um!

Bevor die Installation des MediaWiki gestartet werden kann, müssen noch einige Einstellungen für die Nutzung von PHP vorgenommen werden. PHP wird mit Hilfe der Datei `php.ini` konfiguriert.

Öffnen Sie die Datei `E:\php\php.ini`, und nehmen Sie die im folgenden beschriebenen Änderungen vor!

- Suchen Sie die Zeile `;extension_dir="ext"` und entfernen Sie das Kommentarzeichen (;)!
- Suchen Sie die Zeile `;extension=php_mysql.dll` und entfernen Sie das Kommentarzeichen!
- Speichern und schließen Sie die Datei.
- Kopieren Sie die Datei `E:\php\php.ini` nach `C:\Windows`

ODER

- Setzen Sie die Umgebungsvariable `PHPRC` mit dem Wert: `E:\php`
- Starten Sie die Installation von Mediawiki, in dem Sie die Adresse `http://localhost/mediawiki/index.php` aufrufen!
- Klicken Sie den Link "Please set up the wiki first" an.
- Installieren Sie MediaWiki mit den folgenden Konfigurationseinstellungen:
 - Sprache: de - Deutsch
 - Sprache des Wikis: de - Deutsch
 - Datenbankserver: localhost
 - Datenbankname: my_wiki
 - Datenbanktabellenpräfix:
 - Name des Datenbankbenutzers: root
 - Passwort des Datenbankbenutzers: password
 - Haken entfernen bei: Dasselbe Konto wie während des Installationsvorgangs verwenden
 - Name des Datenbankbenutzers: mediawiki
 - Passwort des Datenbankbenutzers: password
 - Haken setzen bei: Sofern nicht bereits vorhanden, muss nun das Konto erstellt werden
 - Speicher-Engine: InnoDB
 - Datenbankzeichensatz: UTF-8
 - Name des Wikis: my_mediawiki
 - Name des Projektnamensraums: Entspricht dem Namen des Wikis
 - Administratorkonto - Name: wikiadmin
 - Administratorkonto - Passwort: passwordwiki
 - Administratorkonto - E-Mail-Adresse:
 - Option: Nein, das Wiki soll nun installiert werden

Befolgen Sie die weiteren Anweisungen auf dem Bildschirm!

4.2.28 Die Log-Dateien des Apache

Das Error-Log

Das Error-Log

Ein ganz wesentlicher Aspekt bei Verwaltung eines Serverproduktes, wie z. B. eines Webserver ist es, Informationen über das Verhalten des Servers und über die Zugriffe auf den Server gewinnen zu können. Der Apache HTTP-Server bietet hierzu ein umfassendes Logging an.

Das Error-Log ist die zentrale Logdatei des Apache Webserver. Hier laufen alle Fehler- und Diagnosemeldungen auf, die während des Serverbetriebs entstehen. Unter Microsoft Windows heißt die betreffende Datei standardmässig `error.log` und liegt im Verzeichnis `[ServerRoot]/logs`.

Der Inhalt dieser Datei sollte in regelmässigen Zeitabständen immer wieder kontrolliert bzw. im Fehlerfall als erstes geprüft werden.

Die Meldungen im Error-Log haben ein festes Format, das durch den Admin nicht geändert werden kann. Ein Beispiel für eine Fehlermeldung könnte so aussehen:

```
[Mon Nov 28 08:57:37 2011] [error] [client 127.0.0.1]
Premature end of script headers: login.pl
```

Der erste Teil der Meldung ist das Datum, an dem die Meldung aufgelaufen ist. Der zweite Teil ([error]) stellt den Schweregrad der Fehlermeldung dar. Der dritte Teil gibt die Adresse des Clients an, der durch einen Serverzugriff den Fehler ausgelöst hat. Als letztes kommt schließlich die eigentliche Fehlermeldung.

Konfiguriert wird das Error-Log mit Hilfe der Direktive `ErrorLog`. `ErrorLog "logs/error.log"`
Auch wenn die `ErrorLog`-Direktive nicht angegeben wurde erstellt der Apache eine Log-Datei namens `error.log`, im Verzeichnis `[ServerRoot]/logs`.

Access Logs

Zusätzlich zum Error-Log kennt der Apache sogenannte Access-Logs. Diese enthalten verbindungspezifische Informationen zu den Fehlermeldungen im Error-Log. Das Format einer Access-Log Datei ist frei wählbar. Ein Access-Log kann mittels der Direktive `CustomLog` erstellt werden.

Es ist erlaubt mehrere `CustomLog`-Direktiven in der Hauptkonfigurationsdatei zu verwenden, um mehrere Access-Logs zu konfigurieren.

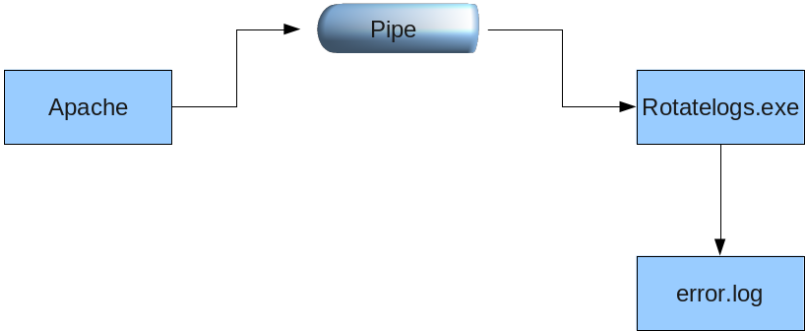
- Wie lautet die Syntax der Direktive `CustomLog`?
- Welche Aufgabe hat die Direktive `LogFormat`?
- Wo finden Sie in der Onlinedokumentation Informationen über die einzelnen Formatparameter der Direktive `LogFormat`?
- Was verbirgt sich hinter der Abkürzung CLF?
- Wie kann ein Format für ein bestimmtes Access-Log festgelegt werden (Syntax)?

Rotierende Log-Dateien (Log rotation)

Bei einem Webserver auf den täglich mehrere tausend Zugriffe erfolgen, kann eine Log-Datei sehr schnell anwachsen und dadurch unübersichtlich werden. Um so etwas zu vermeiden, kann der Apache seine Log-Dateien in konfigurierbaren Zeitabständen automatisch austauschen. Dieser Mechanismus wird als "Log Rotation" bezeichnet und ist sowohl für das Error-Log als auch für alle Access-Logs getrennt einstellbar.

Sowohl bei der `ErrorLog`- als auch bei der `CustomLog`-Direktive sind zwei neue Angaben hinzugekommen. Der erste Teil ist `|bin/rotatelog.exe`. Der senkrechte Strich, am Anfang der Zeile wird als "Pipe-Zeichen" bezeichnet. Er sorgt für das Pipelining der Logging-Informationen an das Programm `rotatelog.exe`, welche dann die Meldungen in die jeweilige Log-Datei schreibt, und alle `n`-Sekunden (hier ist `n = 21600`) eine neue Datei beginnt.

Unter dem Begriff "Pipelining" versteht man eine Art der Interprozesskommunikation. Das heißt zwei Prozesse tauschen Informationen untereinander aus. In diesem speziellen Fall gibt der Apache Webserver Informationen an das Programm `rotatelog.exe` weiter.



Konfigurieren Sie Ihr Error-Log so, dass es alle 30 Sekunden rotiert!

[illegible]

Listing 4.22

```
1 ErrorLog "|bin/rotatelogs.exe logs/error.log 21600"
2 LogFormat "%h %l %u %t \"%r\" %>s %b" common
3 CustomLog "|bin/rotatelogs.exe logs/access.log 21600" common
```

Bedingtes Logging - Das setenvif_module-Modul

Um schon während des Loggings eine Auswahl treffen zu können, welche Informationen protokolliert werden sollen, kennt der Apache die Möglichkeit, mit Hilfe des Moduls `mod_setenvif` so Bedingungen für das Logging zu formulieren.

Recherchieren Sie mit Hilfe der Onlinedokumentation nach der Bedeutung der Anweisungen aus dem unten angegebenen Beispiel!

```
1 LoadModule setenvif_module "module/mod_setenvif.so"
2 SetEnvIf Remote_Addr "127.0.0.1" dontlog
3 CustomLog "logs/access.log" env!=dontlog
4
```

Listing 4.24



Seite 94 von 115

1. Erstellen Sie die folgenden Arbeitsverzeichnisse:
 - E:\ssl
 - E:\ssl\keys
 - E:\ssl\certs
2. Laden Sie eine Kopie der Datei openssl.cfg von der folgenden URL herunter: <http://www.dylanbeattie.net/docs/openssl.conf>
 Achtung: Hierbei handelt es sich um eine öffentlich zugängliche und stark vereinfachte Version der Konfigurationsdatei für OpenSSL, die keinesfalls im Produktivbetrieb eingesetzt werden sollte!
3. Erstellen Sie die Datei E:\ssl\database.txt. Bei dieser Datei handelt es sich um eine leere Datei.
4. Erstellen Sie die Datei E:\ssl\serial.txt. Diese Datei enthält lediglich die Zeichenkette 01.
5. Erstellen Sie ein selbst-signiertes CA Zertifikat für Ihren Webserver, inklusive Private Key.

```
openssl req -new -x509 -days 365 -sha1 -newkey rsa:1024 -nodes -keyout E:\ssl\keys\server.key -out E:\ssl\certs\server.cert -config E:\OpenSSL\1.0.1c\win64\bin\openssl.cfg
```
6. Recherchieren Sie in der OpenSSL Onlinehilfe, was die einzelnen Parameter des Kommandos `openssl req` bedeuten! Sie finden die OpenSSL Onlinehilfe unter: <http://www.openssl.org/docs/apps/req.html>

Den Apache mit SSL konfigurieren - Der Hauptserver

Der erste und auch wichtigste Schritt bei der Konfiguration von SSL für den Apache HTTP-Server ist, dem Webserver mitzuteilen, dass er auf dem HTTPS-Port 443 arbeiten soll. Dies geschieht mit Hilfe der Listen 443-Direktive. Der zweite Schritt ist optional. Um x509- und pkcs7-Zertifikatsdateien korrekt zu unterstützen, sollte der Webserver deren MIME-Typen kennen. Verwenden Sie hierzu die AddType-Direktive.

Während die ersten beiden Schritte für den Hauptserver zu konfigurieren sind, sind alle weiteren Schritte für jeden einzelnen V-Host durchzuführen.

Listing 4.25

```
1 AddType application/x-x509-ca-cert .crt
2 AddType application/x-pkcs7-crl .crl
3
```

Den Apache mit SSL konfigurieren - Die virtuellen Hosts

Jeder V-Host der mit SSL arbeiten soll benötigt mindestens drei Dinge:

- Die SSL-Engine muss eingeschaltet werden
- Ein SSL-Zertifikat muss vorhanden sein, und evtl. auch ein Privat Key-File.
- Die Angabe welche Verschlüsselungsverfahren (engl. CipherSuits) genutzt werden dürfen.

Die SSL-Konfiguration eines virtuellen Servers könnte also so aussehen, wie unten angegeben. Dieses Beispiel enthält jedoch nur die für die Nutzung von SSL absolut notwendigen Direktiven.

Die komplexeste Direktive dürfte zweifelsohne SSLCipherSuite sein. Sie besteht aus einer Liste verschiedener Verschlüsselungsverfahren und präfixen, die durch Doppelpunkte von einander

getrennt werden. Der Client und der Webserver einigen sich bei ihrem Handshake auf eines der erlaubten Verfahren, und nutzen dies anschließend für die Verschlüsselung der Verbindung.

Die Liste kann aus vier verschiedenen Komponenten bestehen:

- Schlüsselaustauschalgorithmien
- Authentifizierungsalgorithmen
- Verschlüsselungsalgorithmen
- MAC-Digest-Algorithmen

Zusätzlich gibt es Aliasnamen, die stellvertretend für eine ganze Gruppe von Algorithmen stehen, wie z. B. ALL, HIGH, MEDIUM, LOW oder SSLv2. Die in der Liste verwendeten Präfixe bedeuten:

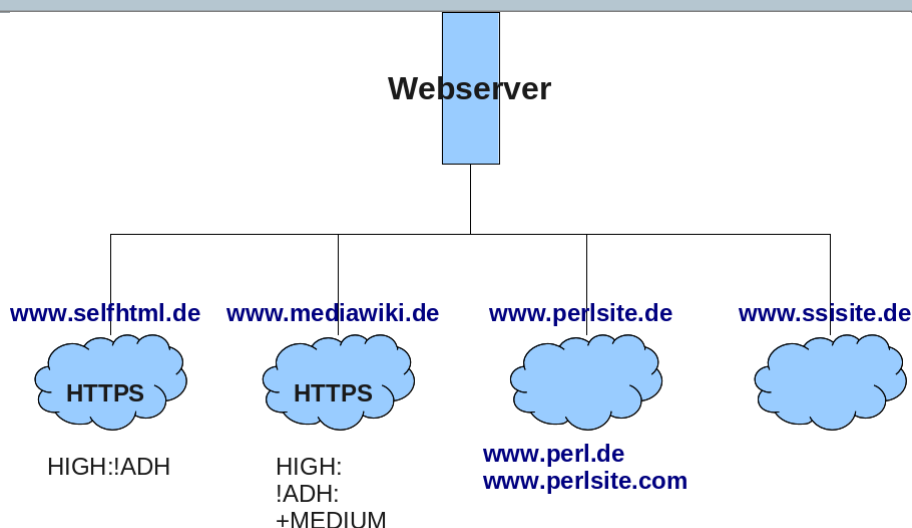
- Ohne Präfix : Hinzufügen des Algorithmus/der Gruppe von Algorithmen zur Liste
- +: Der Algorithmus soll an der genannten Stelle in die Liste eingefügt werden.
- -: Der Algorithmus soll aus der Liste entfernt werden, um später an einer anderen Stelle wieder eingefügt werden zu können.
- !: Unwiderruflicher Ausschluss des Algorithmus/der Gruppe von Algorithmen aus der Liste.

Ein Beispiel für die SSLCipherSuite-Direktive ist unten angegeben. Es bedeutet:

- ALL : Alle Algorithmen sind zulässig
- !ADH : Schließe alle Algorithmen aus, die zur Gruppe der "Anonymen Diffie-Hellman-Algorithmen" gehören.
- Entferne die Gruppe der MEDIUM-Verschlüsselungsalgorithmen an dieser Stelle aus der Liste.
- Füge die Gruppe der SSLv3-Algorithmen an dieser Stelle in die Liste ein.
- Füge die Gruppe der MEDIUM-Algorithmen an der neuen Stelle wieder in die Liste ein.

Die Erstellung eines CipherSuite-Strings sollte grundsätzlich durch einen Security-Admin, und nicht durch den Webadmin selbst erfolgen!

Fertigen Sie jetzt eine Sicherheitskopie ihrer Hauptkonfigurationsdatei an! Erstellen Sie einen Webserver, der die folgenden Webauftritte bereitstellt! Alle V-Hosts haben eigene Error- und Access-Logs (CLF), die im 2 Stunden Rhythmus rotieren. Jeder V-Host stellt eine Seite dar, ohne dass diese vom Nutzer angegeben werden muss (z. B. www.selfhtml.de zeigt direkt die Seite `index.htm`). Wenn ein Nutzer versucht ohne HTTPS-Verbindung auf einen der beiden mit HTTPS gesicherten Hosts zuzugreifen (z. B. <http://selfhtml.de:443>), soll er auf die Adresse <https://selfhtml.de> umgeleitet werden.

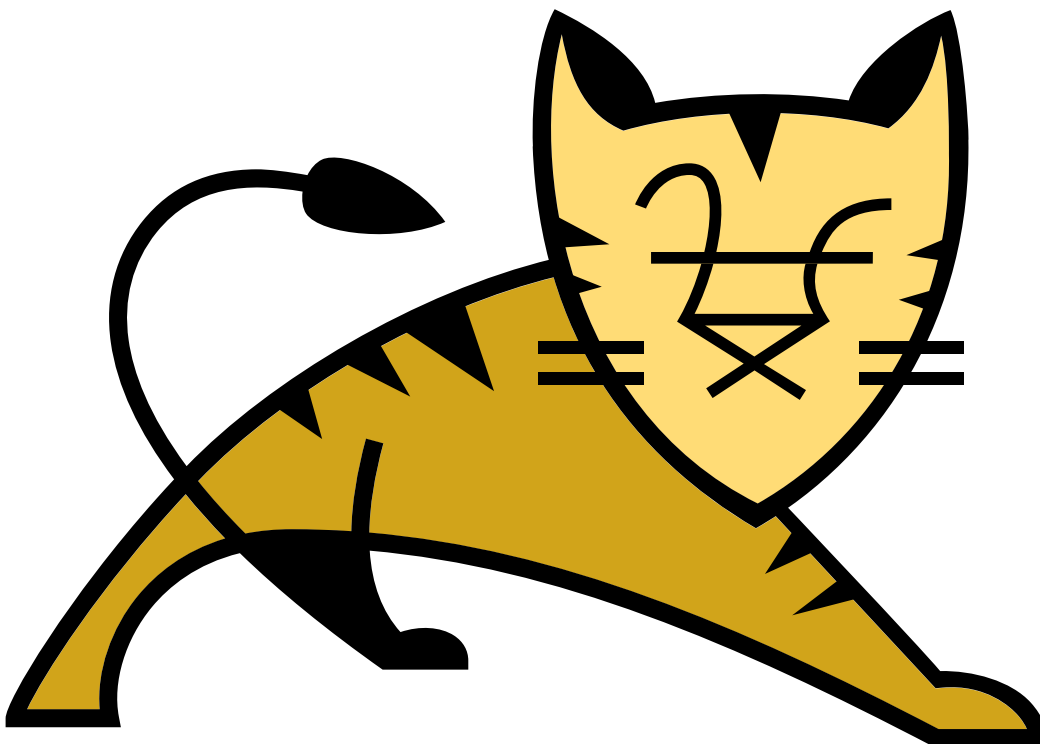


Listing 4.26

```
1 # Die SSL-Engine einschalten
2 SSLEngine On
3
4 # Das Serverzertifikat + Private Key festlegen
5 SSLCertificateFile "E:/ssl/certs/server.crt"
6 SSLCertificateKeyFile "E:/ssl/keys/server.key"
7 SSLCipherSuite ALL:!ADH:-MEDIUM:+SSLv3:+MEDIUM
8
```

4.3 Tomcat

Apache Tomcat



4.3.30 Der Apache Tomcat - Ein Applikationsserver

Was ist ein Applikationsserver

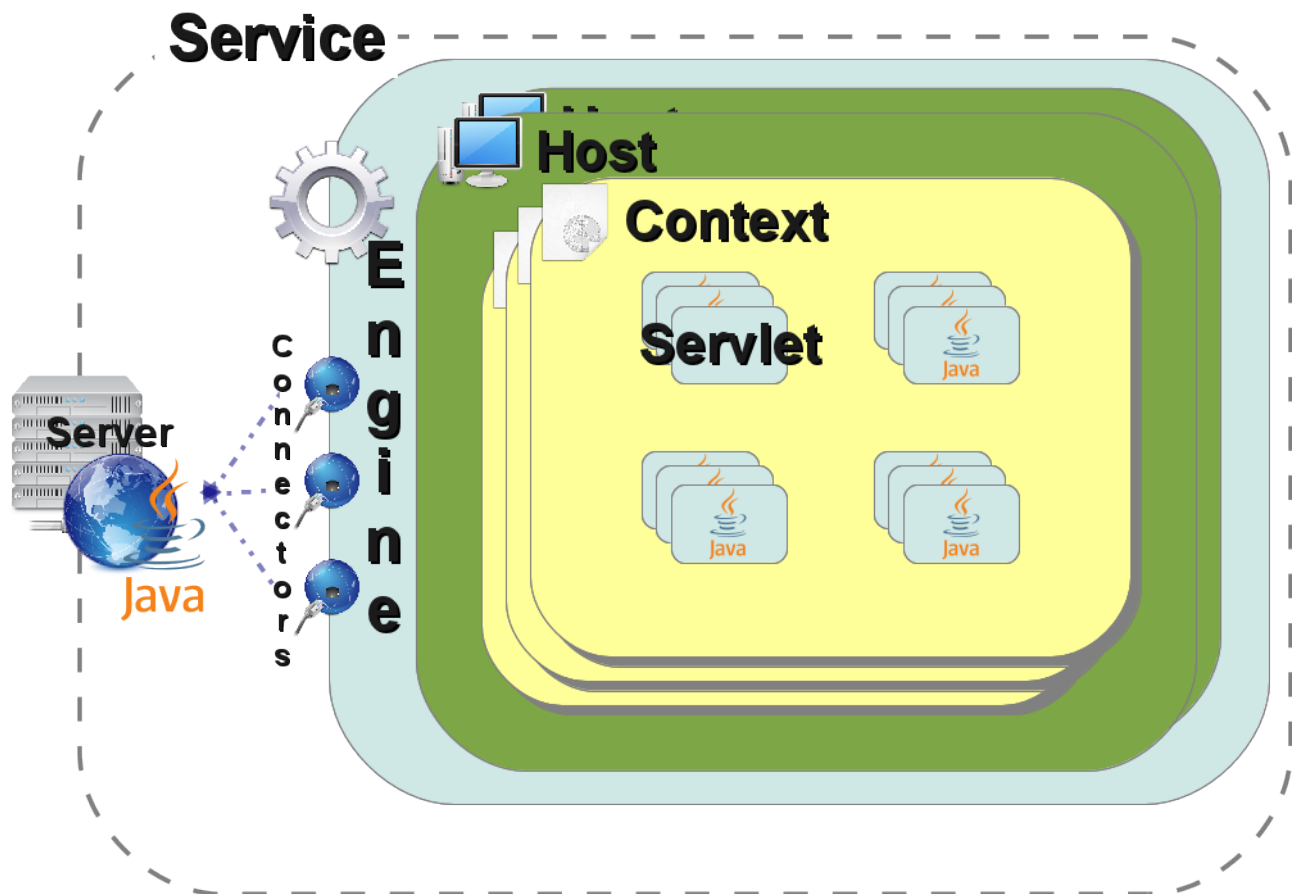
Ein Anwendungsserver (engl. application server) ist im Allgemeinen ein Server in einem Computernetzwerk, auf dem Anwendungsprogramme ausgeführt werden. Im engeren Sinne bezeichnet der Begriff eine Software, die spezielle Dienst zur Verfügung stellt, wie beispielsweise Transaktionen, Authentifizierung oder den Zugriff auf Verzeichnisdienste und Datenbanken über definierte Schnittstellen. (Quelle: Wikipedia <http://de.wikipedia.org/wiki/Anwendungsserver>)

Für die meisten Webinhalte, wie z. B. HTML, PHP oder PERL ist ein Webserver absolut ausreichend. Sollen jedoch komplexe Anwendungsprogramme über ein Netzwerk verfügbar

gemacht werden ändert sich dies, da hier zusätzliche Technologien wie z. B. Java oder ähnliche zum Einsatz kommen. Diese Fähigkeitslücke der Webserver schließen die Applikationsserver bzw. Anwendungsserver. Einer der Application Server ist der Apache Tomcat, der hier als stellvertretendes Beispiel für diese Gruppe der Server dienen soll.

4.3.31 Architektur des Tomcat

Architektur



Servlets



Servlets sind Java-Programme, die nicht auf einem Client, sondern Serverseitig abgearbeitet werden. Sie nehmen Anfragen von Clients entgegen und bearbeiten diese. Die Antwort eines Servlets kann, so wie dies auch bei PHP der Fall ist, eine dynamisch generierte Seite sein. Eine mit den Java Servlets vergleichbare Technologie stellen die Microsoft Active Server Pages, kurz ASP-Seiten dar.

Context



4.3.32 Installation

4.4 Internet Information Services

4.4.33 Herkunft und Installation der IIS

Herkunft

Die Internet Information Services (kurz IIS) sind eine Entwicklung der Firma Microsoft, für deren Windows Produktpalette. Sie stellen einen Webserver Dienst mit vielen modularen Zusatzfunktionen dar, ähnlich dem marktführenden Konkurrenten Apache.

Die Entwicklung der IIS begann mit Microsoft Windows NT 3.51.

Microsoft Windows NT 3.51	IIS 1.0	Als freies Add-On
Microsoft Windows NT 4.0	IIS 2.0	
Microsoft Windows NT 4.0	IIS 3.0	Service Pack 3
Microsoft Windows NT 4.0	IIS 4.0	Option Pack
Windows 2000	IIS 5.0	
Windows XP	IIS 5.1	Professional + Media Center Edition
Windows Server 2003	IIS 6.0	
Windows Server 2008 / Vista	IIS 7.0	
Windows Server 2008 R2	IIS 7.5	Auch auf Windows 7 verfügbar

Die IIS sind nicht standardmässig installiert. Sie müssen als Zusatzoption nachinstalliert werden!

Nutzen Sie Wikipedia, um herauszufinden, wer die IIS entwickelt hat bzw. wo sie ursprünglich entwickelt wurden!

Die IIS sind in der Lage, nicht nur HTML-Seiten auszuliefern, sondern sie stellen auch noch viele andere Dienste zur Verfügung:

- HTTP
- FTP
- SMTP
- NNTP
- WebDAV
- IIS Admin Service

Bei der Installation der IIS werden nur der HTTP-Dienst und die IIS Admin Services installiert. Alle anderen Dienste müssen nachinstalliert bzw. aktiviert werden.

Alle Dienste, die die IIS zur Verfügung stellen können, sind in sogenannten ISAPI-DLL Modulen verpackt. Diese können nach Bedarf hinzugefügt oder entfernt werden.

Installation

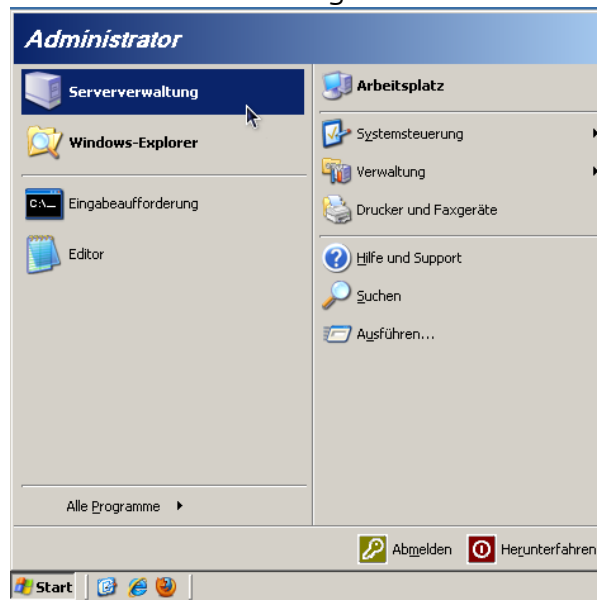
Es gibt drei Möglichkeiten die IIS zu installieren:

1. Die Installation über den Server-Konfigurationsassistenten
2. Die Installation mit Hilfe der Systemsteuerung
3. Die unbeaufsichtigte Installation

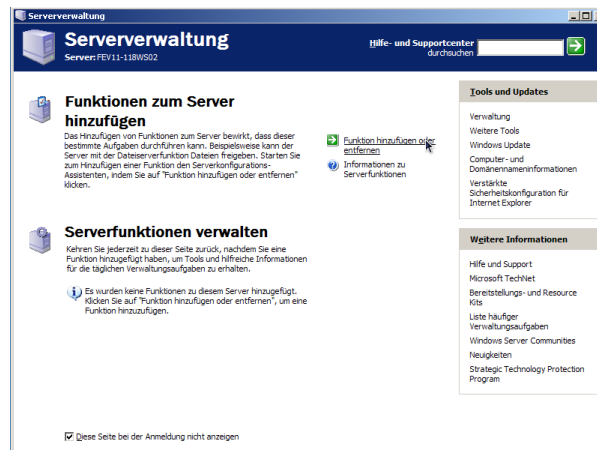
Microsoft empfiehlt aus Sicherheitsgründen die Installation der IIS nur auf einem NTFS-formatierten Laufwerk vorzunehmen, da dieses Dateisystem mehr Sicherheit bietet, als FAT bzw. FAT32.

Installation über den Server-Konfigurationsassistenten

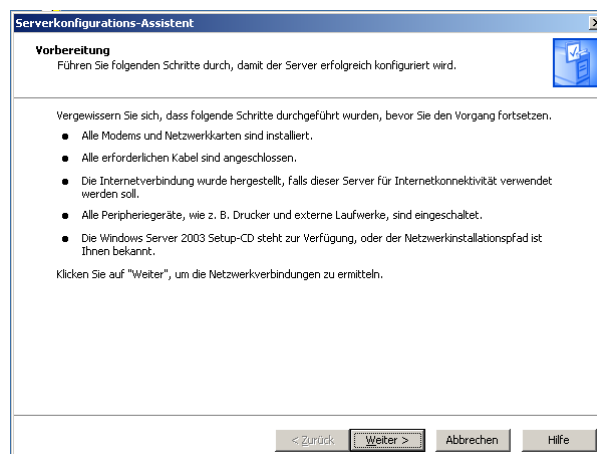
1. Klicken Sie im Startmenü auf Serververwaltung.



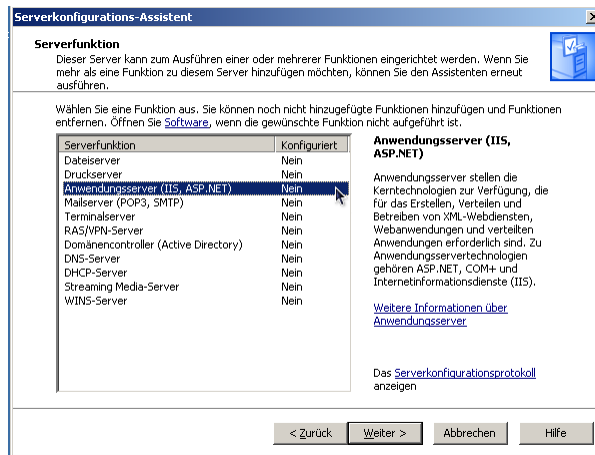
2. Klicken Sie unter Serverfunktionen verwalten auf Funktion hinzufügen oder entfernen.



3. Klicken Sie auf Weiter.



4. Klicken Sie unter Serverfunktion auf Anwendungsserver (IIS, ASP.NET) und dann auf Weiter.



5. Wählen Sie zusätzliche Anwendungsserveroptionen aus, falls Sie diese benötigen und klicken dann auf Weiter.
6. Klicken Sie auf Weiter.
7. Klicken Sie auf Fertigstellen



Die Internet Information Services werden installiert. Nach der Installation kann das IIS Admin Tool im Windows Startmenü aufgerufen werden.



Bei dieser Installationsmethode wird ASP.NET automatisch aktiviert!

Installation mit Hilfe der Systemsteuerung

1. Klicken Sie im Startmenü auf Systemsteuerung.
2. Klicken Sie auf Software.
3. Klicken Sie auf Windows-Komponenten hinzufügen/entfernen.

4. Klicken Sie im Listenfeld Komponenten auf Anwendungsserver.
5. Klicken Sie auf Details.
6. Klicken Sie auf Internetinformationsdienste-Manager.
7. Klicken Sie auf Details, um eine Liste der optionalen IIS-Komponenten anzuzeigen. Wählen Sie die benötigten Komponenten aus, um diese zu installieren.
8. Klicken Sie auf OK, bis Sie zum Assistenten für Windows-Komponenten zurückkehren.
9. Klicken Sie auf Weiter, um den Assistenten für Windows-Komponenten abzuschließen.

Wichtige Verzeichnisse der IIS

Während der Installation der IIS werden die folgenden Verzeichnisse angelegt:

- \InetPub
- %SYSTEMROOT%\Help\IISHelp
- %SYSTEMROOT%\System32\InetSrv
- %SYSTEMROOT%\System32\InetSrv\MetaBack

Diese Verzeichnisse enthalten Benutzerdaten und können nicht verschoben werden.

Bei der Deinstallation von IIS wird das Verzeichnis IISHelp entfernt. Die Verzeichnisse InetPub und InetSrv bleiben auf dem Computer.

Optionale Komponenten

Die IIS umfassen optionale Komponenten die jederzeit nachinstalliert werden können. Die kann auf unterschiedlichen Wegen geschehen:

- Über die Option Windows Komponenten hinzufügen/entfernen
- Im IIS Admin Tool

Hinzufügen optionaler Komponenten

Benutzen Sie das Microsoft Technet, um herauszufinden, wie optionale Komponenten nachinstalliert bzw. deinstalliert werden können! Der folgende Link führt sie zum Technet: <http://technet.microsoft.com/de-de/library/default.aspx>

4.4.34 Datensicherung

Sichern und Wiederherstellen der Metabase

Die Metabase enthält eine hierarchische Struktur, die die Konfiguration der IIS speichert. Seit Version 6 Sie besteht aus zwei XML-Dateien: MetaBase.XML und MBSchema.XML. Beide sind im Verzeichnis %SYSTEMROOT%\System32\InetSrv abgelegt. Nur Mitglieder der Gruppe der Administratoren können diese Dateien verändern.

Werden die IIS gestartet, wird der Inhalt der Datei MetaBase.XML in einen Bereich des Arbeitsspeichers kopiert, der als "In-Memory Metabase" bezeichnet wird. Die Metabase besteht somit insgesamt aus den drei Teilen:

- MetaBase.XML
- MBSchema.XML
- In-Memory Metabase

Werden Veränderungen an der Konfiguration der IIS vorgenommen, werden diese zuerst in der In-Memory MetaBase zwischengespeichert und anschließend persistent in die XML-Dateien übertragen. Diese Übertragung findet in bestimmten Zeitintervallen statt.

IIS-Admins sollten in regelmäßigen Zeitabständen eine Sicherung der Metabase durchführen, so dass ein Verlust der aktuellen Konfiguration ausgeschlossen ist. Die erste automatische Sicherung der MetaBase erfolgt direkt im Anschluss an die Installation der IIS. Diese Sicherung ist notwendig, damit der Urzustand der IIS-Installation bei Bedarf wiederhergestellt werden kann.

Die Metabase kann mit Hilfe des IIS Admin Tools oder eines Skripts gesichert werden. Beim Sichern werden Kopien der beiden XML-Dateien MetaBase.XML und MBSchema.XML erstellt. Jede Sicherung kann bzw. sollte mit einem Passwort versehen werden.

Die Sicherungskopien der MetaBase werden im Verzeichnis %SYSTEMROOT%\System32\InetSrv\MetaBack gespeichert.

Sichern der Metabase

1. Starten Sie das IIS Admin Tool im Windows Startmenü.
2. Klicken Sie mit der rechten Maustaste auf den lokalen Computer.
3. Wählen Sie im Kontextmenü Alle Aufgaben aus.
4. Klicken Sie auf Konfiguration Sichern Wiederherstellen.
 \"save
5. Klicken Sie auf Sicherungskopie erstellen.
 \"save
6. Geben Sie der Sicherungskopie einen Namen und ein Passwort.
 \"save
7. Klicken Sie auf OK.

Wiederherstellen der Metabase

1. Starten Sie das IIS Admin Tool im Windows Startmenü.
2. Klicken Sie mit der rechten Maustaste auf den lokalen Computer.
3. Wählen Sie im Kontextmenü Alle Aufgaben aus.
4. Klicken Sie auf Konfiguration Sichern/Wiederherstellen.
 \"save
5. Klicken Sie im Listenfeld Sicherungskopien auf die automatische Sicherungskopie, die Sie wiederherstellen möchten.
 \"restore
6. Klicken Sie auf Wiederherstellen.
 \"restore
7. Geben Sie das Passwort ein und klicken Sie auf OK.
 \"restore
8. Klicken Sie auf OK und der Vorgang wird beendet.
 \"restore
9. Klicken Sie auf Schließen

4.4.35 Remoteverwaltung

Einrichten der Remoteverwaltung

Benutzen Sie das Microsoft Technet, um herauszufinden, wie die Remoteverwaltung der IIS per IIS Admin Tool und HTTP konfiguriert werden kann! Der folgende Link führt sie zum Technet: <http://technet.microsoft.com/de-de/library/default.aspx>

4.4.36 Authentifizierung und Autorisierung

Anonyme Authentifizierung

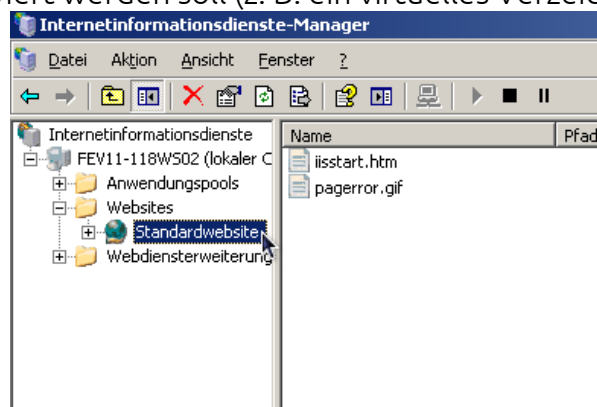
Die Internet Information Services können so konfiguriert werden, dass jeder Nutzer zuerst einen Nutzernamen und ein Passwort eingeben muss, bevor er Zugriff auf eine bestimmte Website erhält. Dies wird als Authentifizierung bezeichnet. Es stehen mehrere Authentifizierungsverfahren zur Verfügung:

- Anonyme Authentifizierung
- Standardauthentifizierung
- Digestauthentifizierung
- Erweiterte Digestauthentifizierung
- Integrierte Windows-Authentifizierung
- UNC-Authentifizierung
- .NET Passport-Authentifizierung
- Authentifizierung von FTP-Sites

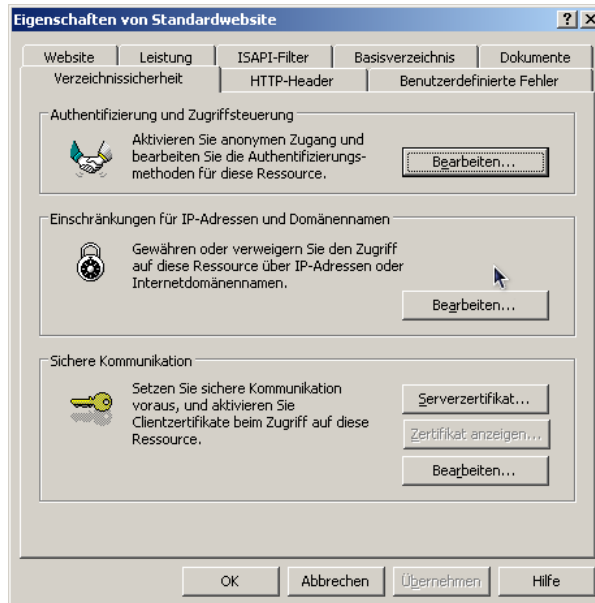
Mit Hilfe der anonymen Authentifizierung können Nutzer, ohne die Angabe von Nutzernamen und Passwort, auf öffentliche Bereiche einer Website Zugriff erhalten. Für dieses Verfahren wird auf dem Webserver das Nutzerkonto "IUSR_Computername" genutzt, das für das Setup zur Gruppe "Gäste" hinzugefügt wird. Mit Ausnahme des Kontos "Gast", haben Mitglieder der Gruppe "Gäste" die gleichen Rechte, wie die der Gruppe "Benutzer".

Um die Anonyme Authentifizierung zu aktivieren, gehen Sie wie folgt vor:

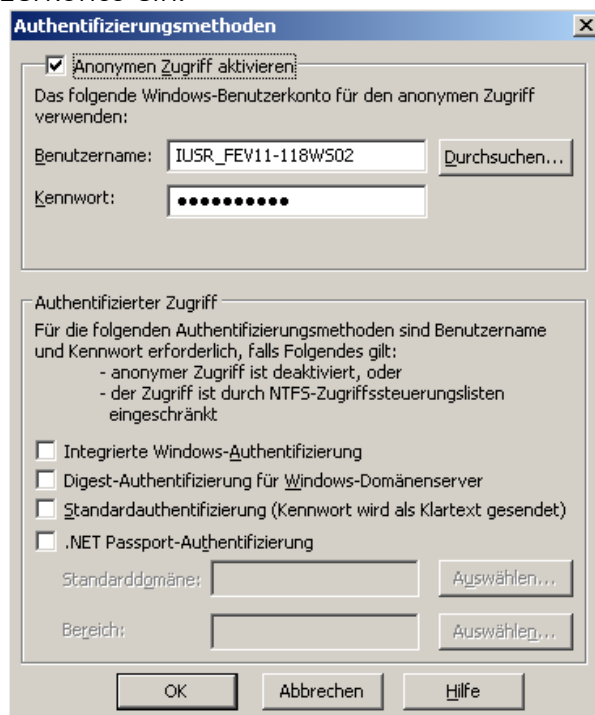
1. Wählen Sie im IIS-Manager das gewünschte Objekt, aus für das die anonyme Authentifizierung aktiviert werden soll (z. B. ein virtuelles Verzeichnis oder eine Website).



2. Öffnen Sie mit der rechten Maustaste das Kontextmenü des betreffenden Objekts und wählen Sie den Punkt "Eigenschaften".
3. Klicken Sie auf die Registerkarte "Verzeichnissicherheit" bzw. "Dateisicherheit".



4. Klicken Sie unter "Authentifizierung und Zugriffssteuerung" auf die Schaltfläche bearbeiten.
5. Aktivieren Sie das Häkchen für die anonyme Authentifizierung und tragen Sie ein entsprechendes Benutzerkonto ein.



6. Klicken Sie auf "OK".
7. Klicken Sie auf "OK".

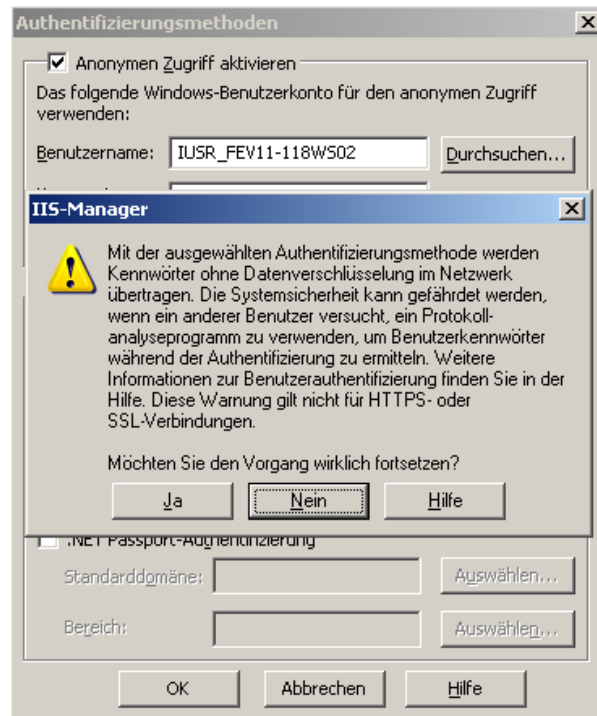
Standardauthentifizierung

Die Standardauthentifizierung stellt eine Methode dar, bei der Nutzernamen und Passwörter unverschlüsselt über das Netzwerk übertragen werden. Es ist jedoch möglich, die Standardauthentifizierung in Zusammenhang mit den Verschlüsselungstechniken des Webserverns zu nutzen, um so die Sicherheit zu erhöhen.

Wird für eine Website die Standardauthentifizierung aktiviert, so müssen auch entsprechende Windowsbenutzerkonten erstellt und NTFS-Rechte gesetzt werden.

1. Wählen Sie im IIS-Manager das gewünschte Objekt, aus für das die Standardauthentifizierung aktiviert werden soll (z. B. ein virtuelles Verzeichnis oder eine Website).
2. Öffnen Sie mit der rechten Maustaste das Kontextmenü des betreffenden Objekts und wählen Sie den Punkt "Eigenschaften".

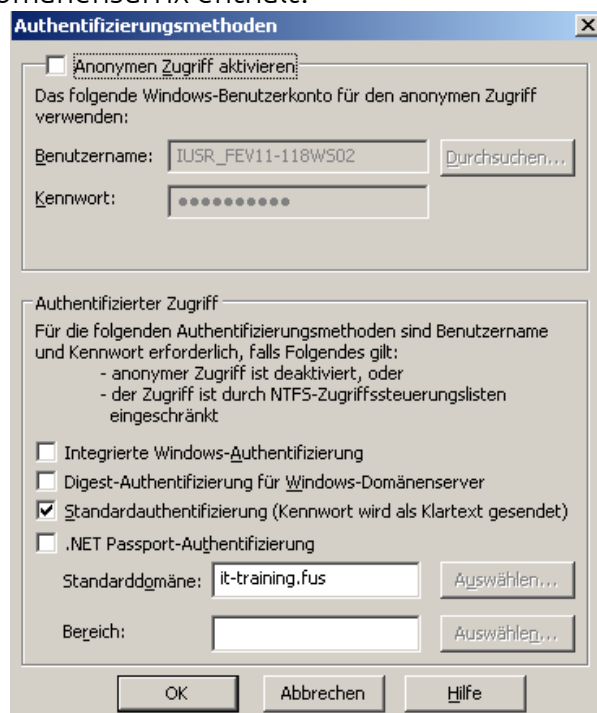
3. Klicken Sie auf die Registerkarte "Verzeichnissicherheit" bzw. "Dateisicherheit".
4. Klicken Sie unter "Authentifizierung und Zugriffssteuerung" auf die Schaltfläche bearbeiten.
5. Deaktivieren Sie das Häkchen für die anonyme Authentifizierung
6. Aktivieren Sie das Häkchen für die Standardauthentifizierung.
7. Da diese Methode, wie bereits erwähnt, die Nutzernamen und Passwörter im Klartext verschickt, erscheint an dieser Stelle eine Warnmeldung, die Sie auf diesen Umstand hinweist. Klicken Sie auf JA.



8. Geben Sie im Feld Standarddomäne ein Standarddämonensuffix an. Dieses wird immer dann verwendet, wenn der Nutzer bei der Authentifizierung kein Domänensuffix angibt.

Beispiel 1: Als Standarddomäne wird "it-training.fus" eingetragen. Der Nutzer gibt als Nutzernamen den Wert "cMueller" ein. Sein Nutzernamen wird durch die Standarddomäne in "cMueller@it-training.fus" ergänzt.

Beispiel 2: Als Standarddomäne wird "it-training.fus" eingetragen. Der Nutzer gibt als Nutzernamen den Wert "cMueller@FueUstgSBw.bw" ein. Es erfolgt keine Ergänzung, da der Nutzernamen bereits ein Domänensuffix enthält.



9. Klicken Sie auf OK.
10. Klicken Sie auf OK.

Welche Aufgabe hat das Textfeld "Bereich" im Dialogfeld "Authentifizierungsmethoden"?
Was passiert mit dort eingetragenen Werten?

Außer der Tatsache, dass Nutzernamen und Passwörter unverschlüsselt übertragen werden, stellt auch der Tokencache ein Problem dar. Die Aufgabe des Tokencaches ist es, Anmeldeinformationen zwischenspeichern. D. h. hat sich ein Nutzer einmal am Webserver angemeldet, wird dessen Anmeldeinformation, das sog. Benutzertoken, im Tokencache zwischengespeichert. Dies sorgt dafür, dass ein einmal angemeldeter Nutzer sich nicht bei jeder weiteren Webseite, die eine Authentifizierung fordert, erneut anmelden muss.

Problematisch ist, dass ein Angreifer diesen Tokencache nutzen könnte, um sich unberechtigten Zugriff zum Webserver zu verschaffen. Dies kann jedoch durch die richtige Konfiguration des Tokencaches verhindert, bzw. das Risiko kann verringert, werden.

Die Einstellungen für den Tokencache werden in der Windowsregistrierungsdatei vorgenommen.

1. Öffnen Sie den Windows Registrierungseditor Regedit32.exe
2. Wählen Sie den Registrierungsschlüssel `\\HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\InetInfo\Parameters`.
3. Klicken Sie auf das Menü "Bearbeiten" und wählen Sie den Menüpunkt "Neu".
4. Wählen Sie den Typ "DWORD-Wert"
Tokencache
5. Geben als Name "UserTokenTTL" ein.
6. Doppelklicken Sie auf "UserTokenTTL" und geben Sie als Wert eine Zahl ein. Diese legt die Sekunden fest, wie lange ein Benutzertoken gültig ist, bzw. wann es aus dem Cache entfernt wird.
Tokencache
7. Schließen Sie den Registrierungseditor.
8. Starten Sie die IIS neu.

Integrierte Windows-Authentifizierung

Im Gegensatz zur Standard-/Basisauthentifizierung stellt die Integrierte Windows-Authentifizierung eine sogenannte "sichere Authentifizierungsmethode" dar. Sicher heißt in diesem Falle, dass kein Nutzernamen und kein Passwort über das Netzwerk übertragen werden, sondern nur ein Hashwert. Vormalig wurde diese Art der Authentifizierung auch als NTLM-Authentifizierung bezeichnet.

1. Wählen Sie im IIS-Manager das gewünschte Objekt, auf das die Integrierte Windows-Authentifizierung aktiviert werden soll (z. B. ein virtuelles Verzeichnis oder eine Website).
2. Öffnen Sie mit der rechten Maustaste das Kontextmenü des betreffenden Objekts und wählen Sie den Punkt "Eigenschaften".
3. Klicken Sie auf die Registerkarte "Verzeichnissicherheit" bzw. "Dateisicherheit".
4. Klicken Sie unter "Authentifizierung und Zugriffssteuerung" auf die Schaltfläche bearbeiten.
5. Deaktivieren Sie das Häkchen für die anonyme Authentifizierung
6. Aktivieren Sie das Häkchen für die integrierte Windows-Authentifizierung.
7. Klicken Sie auf "OK".
8. Klicken Sie auf "OK".

Autorisierung

NTFS-Berechtigungen einsetzen

Welche Empfehlungen gelten beim Sichern von Dateien mit NTFS-Berechtigungen? Ermitteln Sie dies mit Hilfe des Microsoft TechNet!

1. Wählen Sie im IIS-Manager die Website/Datei aus die Sie sichern möchten.
2. Klicken Sie mit der rechten Maustaste und wählen Sie im Kontextmenü den Punkt "Berechtigungen".
3. Vergeben Sie, die entsprechenden Berechtigungen.
4. Klicken Sie auf "OK".

Benutzung von Websiteberechtigungen

Websiteberechtigungen sind nur als Ergänzung zu den NTFS-Berechtigungen zu sehen, nicht aber als alleiniges Mittel, um eine Website zu sichern. Im Gegensatz zu NTFS-Berechtigungen gelten Websiteberechtigungen unabhängig von Nutzerkonten, also immer für alle Nutzer.

Ermitteln Sie, mit Hilfe des Microsoft TechNet, welche Regeln gelten, wenn Sie NTFS-Berechtigungen zusammen mit Websiteberechtigungen einsetzen und wie Sie Websiteberechtigungen einrichten!

Ermöglichen von anonymen Zugriffen

Anonymer Zugriff ist die am häufigsten verwendete Zugriffsform. Hiermit kann ein Nutzer ohne weitere Authentifizierung auf eine Website zugreifen. Wie diese Art der Authentifizierung konfiguriert wird, wurde bereits im Vorfeld beschrieben. Genutzt wird sie, in dem der gewünschten Website die NTFS-Berechtigungen:

- Lesen und Ausführen
- Ordnerinhalt auflisten
- Lesen

zugewiesen werden.

4.4.37 Bereitstellen von Webseiten

Die Standardwebsite

Das Bereitstellen von Websites durch die Internet Information Services geschieht in zwei Schritten:

- Erstellen der Website
- Erstellen einer Konfiguration für die IIS, zum Bereitstellen der Website.

An dieser Stelle wird nur der zweite Aufzählungspunkt behandelt, das Erstellen der Konfiguration für die Bereitstellung der Website durch die IIS.

Nach der Installation stellen die IIS automatisch eine Website dar, die Standardwebsite. Diese besteht aus zwei Dateien, die sich im Verzeichnis InetPub\wwwroot befinden. Sie heißen:

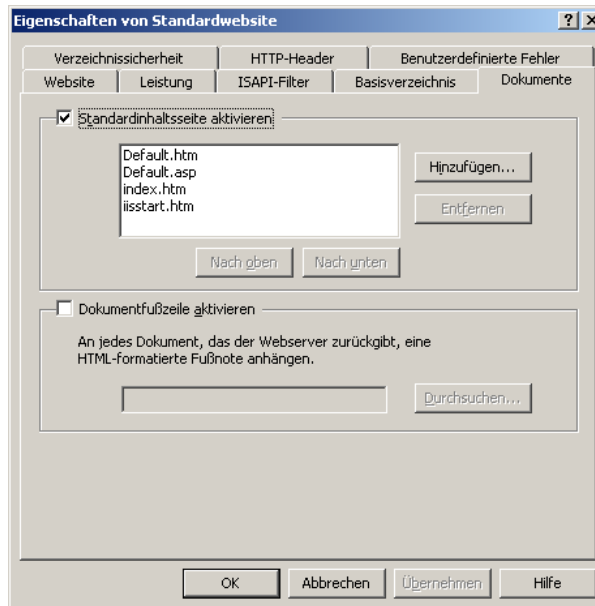
- iisstart.htm

- pagerror.gif

Diese Seite zeigt die Meldung an, das auf diesem Webserver noch keine Standardwebseite eingerichtet wurde, bzw. das diese sich noch in Bearbeitung befindet.

Um eine eigene Standardwebsite für Ihren Webserver zu konfigurieren, unternehmen Sie folgende Schritte:

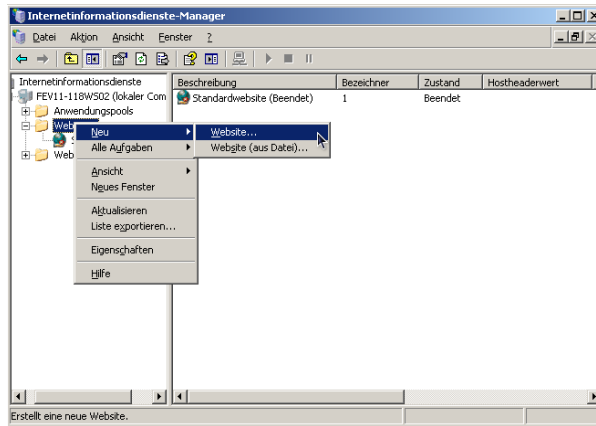
1. Öffnen Sie die IIS Admin Tools
2. Erweitern Sie die Hierarchie in der linken Fensterhälfte: Internetinformationsdienste → (lokaler Computer) → Websites → Standardwebseite
3. Klicken Sie mit der rechten Maustaste auf Standardwebseite und öffnen Sie die Eigenschaften der Standardwebseite.
4. Wählen Sie die Registerkarte "Dokumente".



5. Setzen Sie den Haken bei "Standardinhaltseite anzeigen", falls dies nicht bereits geschehen ist.
6. Benutzen Sie die Schaltfläche "Entfernen", um alle nicht benötigten Dateinamen aus der Liste zu entfernen.
7. Fügen Sie, mit Hilfe der Schaltfläche "Hinzufügen" den Dateinamen Ihrer Standardwebsite hinzu.
8. Klicken Sie auf "OK".
9. Testen Sie mit dem Browser Ihrer Wahl, ob die Änderung funktioniert.

Erstellen neuer Websites

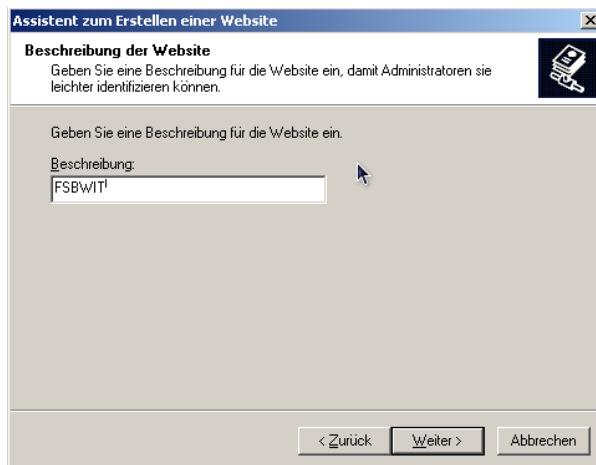
1. Öffnen Sie die IIS Admin Tools
2. Um Ressourcenkonflikte zu vermeiden, beenden Sie die Standardwebseite
3. Klicken Sie mit der rechten Maustaste auf den Ordner "Websites" um das Kontextmenü zu öffnen, und klicken Sie auf die Eigenschaft "Neu" → "Website..."



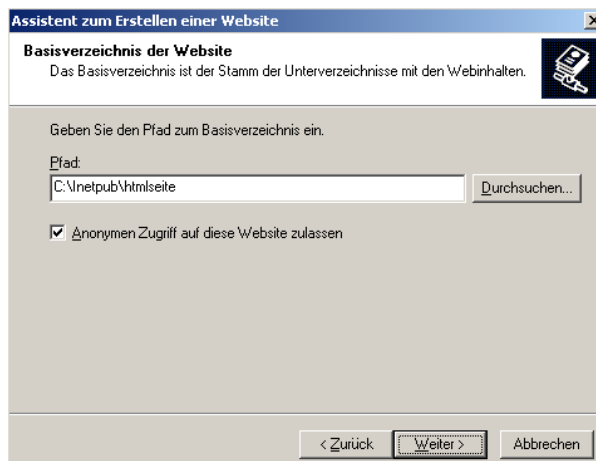
4. Klicken Sie auf "Weiter"



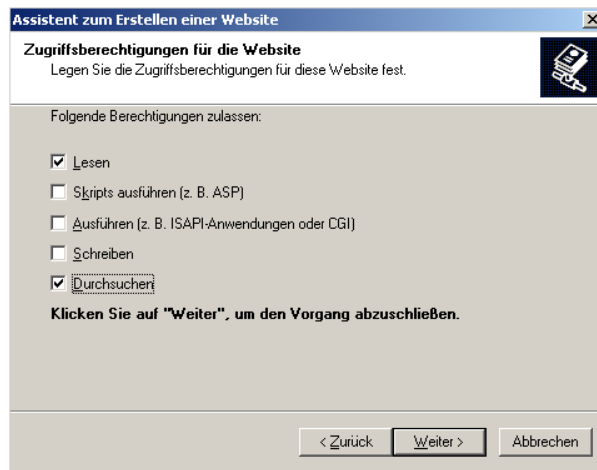
5. Geben Sie einen Namen für Ihre Website an. Dieser wird nur innerhalb der Admin Tools angezeigt und klicken Sie auf "Weiter".



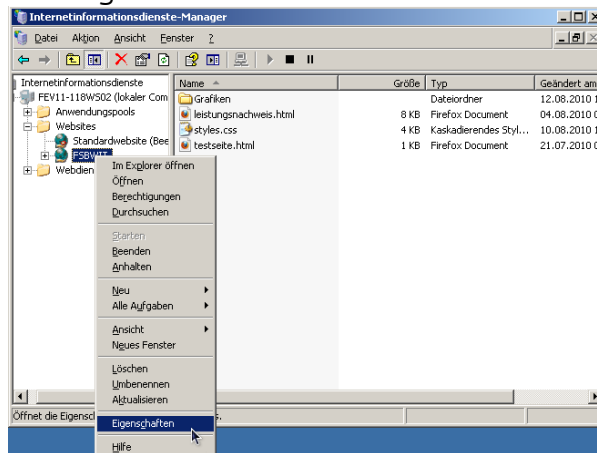
6. Geben Sie das Verzeichnis an, in dem Ihre Website erstellt werden soll und klicken Sie auf "Weiter".



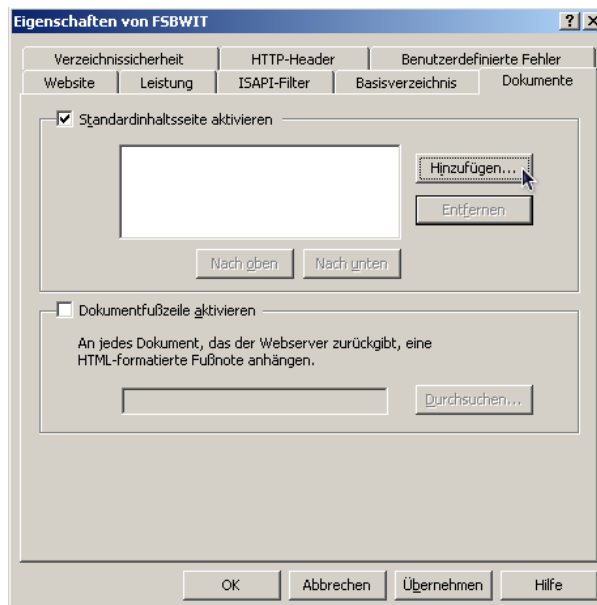
7. Legen Sie die Websiteberechtigungen für Ihre Website fest und klicken Sie auf "Weiter".



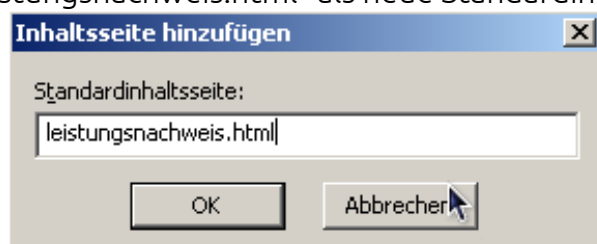
8. Klicken Sie auf "Fertigstellen".
9. Klicken Sie mit der rechten Maustaste auf den Ordner "Websites" um das Kontextmenü zu öffnen, und klicken Sie auf "Eigenschaften"



10. Öffnen Sie die Registerkarte "Dokumenten" und entfernen Sie alle Einträge aus der Liste der Standardinhaltsseiten.



11. Fügen Sie die Seite "leistungs-nachweis.html" als neue Standardinhaltsseite hinzu.



12. Klicken Sie auf "OK".

Virtuelle Verzeichnisse

Mit Hilfe eines Webserverns können nicht nur Websites angezeigt werden, sondern auch Dateien jeglicher Art veröffentlicht werden. Diese müssen sich dazu entweder im Stammverzeichnis der Website, oder in einem virtuellen Verzeichnis befinden.

Virtuelle Verzeichnisse können sein:

- Ein beliebiges Verzeichnis auf dem Webserver, außerhalb des Stammverzeichnisses Ihrer Website.
- Eine Windowsfreigabe auf einem Dateiserver in Ihrem Netzwerk.

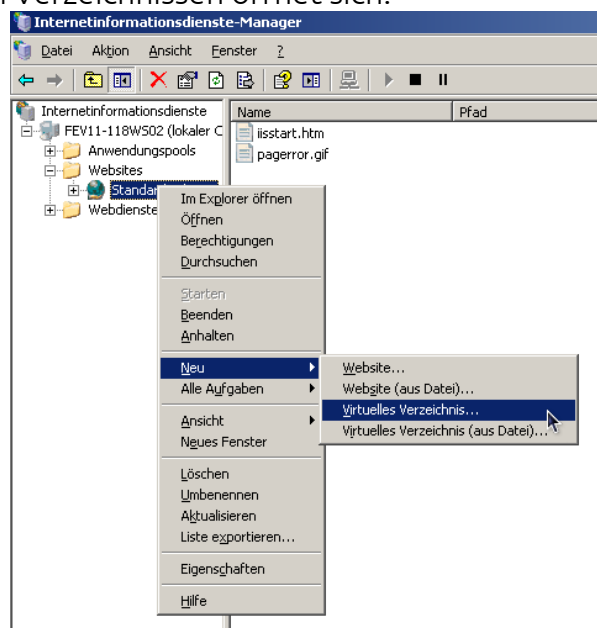
Ein virtuelles Verzeichnis ist ein Verzeichnis, das nicht im Basisverzeichnis enthalten ist, das jedoch Clientbrowsern so angezeigt wird, als ob dies der Fall wäre.

Virtuelle Verzeichnisse können auf zwei unterschiedliche Arten angelegt werden:

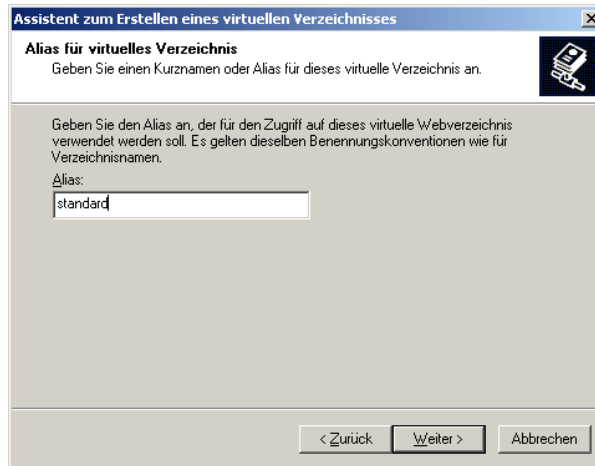
- Mit Hilfe der IIS Admin Tools
- Mit dem Windows Explorer

Um ein Virtuelles Verzeichnis mit Hilfe der IIS Admin Tools anzulegen...

1. Öffnen Sie die IIS Admin Tools
2. Erweitern Sie die Hierarchie in der linken Fensterhälfte: Internetinformationsdienste → (lokaler Computer) → Websites
3. Klicken Sie mit der rechten Maustaste auf die Website, der Sie ein virtuelles Verzeichnis hinzufügen wollen.
4. Wählen Sie im Kontextmenü den Punkt Neu → Virtuelles Verzeichnis. Der Assistent zum Erstellen von virtuellen Verzeichnissen öffnet sich.

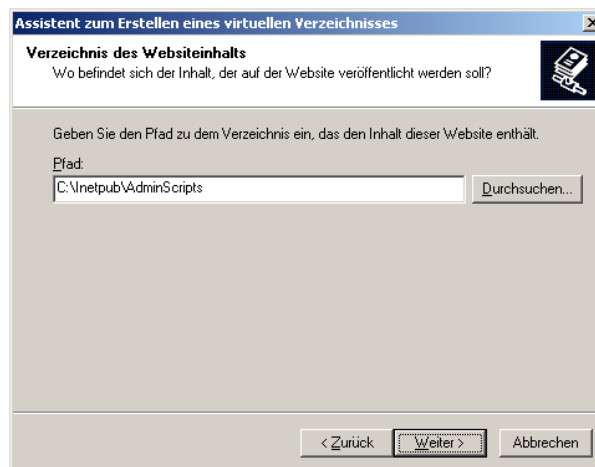


5. Klicken Sie auf "Weiter".
6. Geben Sie im Feld Alias einen kurzen, einprägsamen Namen für Ihr Verzeichnis an.



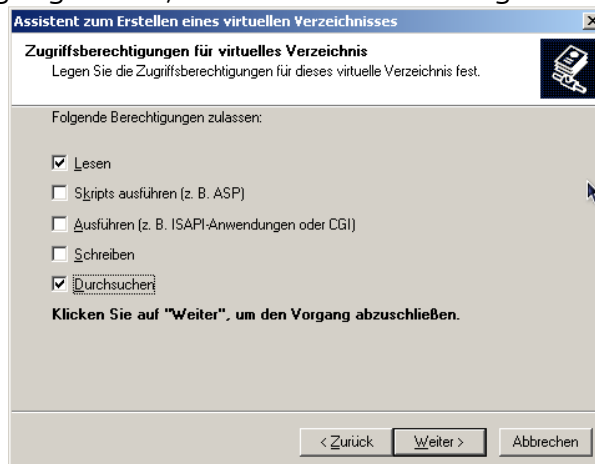
7. Klicken Sie auf "Weiter".

8. Geben Sie im Feld Pfad das physikalische Verzeichnis an, auf das Ihr virtuelles Verzeichnis verweist.



9. Klicken Sie auf "Weiter".

10. Legen Sie die Berechtigungen fest, die für Ihr Verzeichnis gelten sollen.



11. Klicken Sie auf "Weiter".

12. Klicken Sie auf "Fertigstellen", um den Vorgang zu beenden.

Finden Sie, mit Hilfe des Microsoft Technet, heraus, wie Sie im Windows Explorer Virtuelle Verzeichnisse anlegen können! Der folgende Link führt sie zum Technet: <http://technet.microsoft.com/de-de/library/default.aspx>