

LF2 - Webseiten erstellen

1 HTML

1.1 Einführung in HTML

Stundenempfehlung

FIAE	FISI	ITSK	IFK	ITSE
4	4	4	4	4

[1.1.1 Übersicht über dieses Kapitel](#)

[1.1.2 Übersicht über Webtechnologien](#)

[1.1.3 HTML/XHTML - Grundkonzepte](#)

[1.1.4 HTML-Grundgerüst](#)

[1.1.5 Zeichenvorrat](#)

[1.1.6 Dateinamen im Web](#)

[1.1.7 Kapitelübersicht](#)

[1.1.8 Übungen zur Selbstkontrolle](#)

1.1.1 Übersicht über dieses Kapitel

In diesem Kapitel werden Grundlagen zu Webtechnologien und zur Webseitenerstellung mit HTML behandelt.

Sie erhalten einen Überblick über die aktuellen Webtechnologien, zu denen auch HTML gehört. Danach werden die ersten Themen zu HTML behandelt. Sie lernen das TagKonzept kennen, das Grundgerüst einer HTML-Datei, die verwendbaren Zeichen und Regeln für die Bildung von Dateinamen im Internet.

Zu allen behandelten Themen werden Hyperlinks auf [SelfHTML](#) angegeben. Dadurch lernen Sie gleichzeitig dieses umfassende Nachschlagewerk kennen.

Begriffe

Seite

Eine **Seite** ist eine einzelne HTML-Seite.

site

Eine **site** (auch **Website**) ist eine Webpräsenz, die aus mehreren Seiten besteht.

HTML-Editoren

Für die Erstellung von HTML-Editoren gibt es Quellcode-Editoren und WYSIWYG<http://de.wikipedia.org/wiki/WYSIWYG> Editoren. Da in den ersten beiden Modulen ("HTML", "CSS") der Quellcode im Vordergrund steht, empfehle ich einen Quellcodeeditor.

Ich empfehle den Quellcodeeditor "[Scriptly](#)".

1.1.2 Übersicht über Webtechnologien

HTML

HTML ist die Abkürzung von **H**yper**T**ext **M**arkup **L**anguage.

Im **World Wide Web** werden neben HTML weitere Technologien angewendet. In diesem Abschnitt werden Sie einen Überblick über ausgewählte Web-Technologien erhalten

(1.1) Weitere Web-Technologien

Aufgabe

Beantworten sie die folgenden Fragen mit Hilfe des angegebenen Links: =>
[SelfHTML -> Web-Technologien](#)

Hinweis

Beachten Sie bitte bei SelfHTML, dass die letzte Aktualisierung 2007 durchgeführt wurde.

HTML

Welche Ausgaben hat HTML?

Welche Version ist die aktuell benutzte HTML-Version?

CSS (Stylesheets)

Was bedeutet die Abkürzung CSS?

Wozu wird CSS verwendet?

JavaScript und DOM

Was ist JavaScript?

Was bedeutet DOM?

Perl und CGI

Was bedeutet CGI und wozu wird CGI verwendet?

Erläutern Sie kurz, was Perl ist.

PHP

Erläutern Sie kurz, was PHP ist.

Java

Wozu wird Java in Zusammenhang mit HTML verwendet?

ActiveX / ASP.NET

Was bedeutet ASP?

Was ist ActiveX?

Flash

Was ist Flash?

=> Lösung

HTML

Welche Ausgaben hat HTML?

Auszeichnungssprache, mit der die Browseransicht einer Webseite definiert wird. HTML kann u. a. Absätze, Listen, Aufzählungen und Tabellen definieren. Im beschränkten Umfang ermöglicht HTML auch die Formatierung.

Hypertext: Mit Hilfe von Hyperlinks kann HTML Verweise, mit denen innerhalb eines Dokuments oder auch zu anderen Dokumenten gesprungen werden kann.

CSS (Stylesheets)

Was bedeutet die Abkürzung CSS?

Cascading Stylesheets

Wozu wird CSS verwendet?

CSS wird zur Formatierung von Webseiten verwendet (Version 1) und zur Positionierung der Elemente auf der Webseite (Version 2.0). Version 2.1 wird von allen Browsern unterstützt. Version 3.0 ist seit 2000 in Entwicklung.

JavaScript und DOM

Was ist JavaScript?

JavaScript ist eine Skript-Sprache, die auf dem Browser ausgeführt wird. Damit werden Interaktionen im Browser ermöglicht, ohne dass die Webseite neu vom Webserver geladen werden muss.

Was bedeutet DOM?

DOM: Document Object Model

Die Elemente einer Webseite sind hierarchisch strukturiert. DOM ist eine Schnittstelle für den Zugriff auf HTML-Dokumente.

Perl und CGI

Was bedeutet CGI und wozu wird CGI verwendet?

CGI: **C**ommon **G**ateway **I**nterface (Allgemeine Vermittlungsrechner-Schnittstelle)

Schnittstelle, über die es möglich ist, von HTML aus Programme oder Skripte aufzurufen.

Erläutern Sie kurz, was Perl ist.

Script-Sprache, in der CGI-Programme geschrieben werden können.

PHP

Erläutern Sie kurz, was PHP ist.

Serverseitige Skriptsprache zur Erzeugung dynamischer Webseiten. Die PHP-Skripte werden auf dem Server beim Aufruf einer PHP-Seite ausgeführt.

Java

Wozu wird Java in Zusammenhang mit HTML verwendet?

Java ist eine plattformunabhängige Sprache. Sie wird in Zusammenhang mit HTML verwendet, um Applets und Servlets für Webseiten zu erstellen.

Applets: Java-Programme, die eine grafische Oberfläche haben können und auf Webseiten geladen werden können. Sie laufen clientseitig.

Servlets: Zur Erstellung von Servlets wird Java als serverseitige Sprache verwendet. Servlets sind die Java-Variante für eine serverseitige Skript-Sprache.

ActiveX / ASP

Was bedeutet ASP?

ASP: **A**ctive **S**erver **P**ages

Unter dem Namen ASP vereint Microsoft die Technologien für serverseitige Skriptsprachen. Als Programmiersprachen können VBScript und C# (C sharp) verwendet werden.

Was ist ActiveX?

ActiveX ist im Zusammenhang mit Webanwendungen eine Technologie von Microsoft, um vom Browser aus direkt auf den Rechner zugreifen zu können. Die Zugriffe erfolgen über ActiveX-Steuerelemente.

Flash

Was ist Flash?

Flash ist eine Technologie von Adobe. Damit besteht die Möglichkeit, dynamische grafische Elemente auf der Webseite einzufügen. Weite Verbreitung hat die

Bereitstellung von Videos erlangt, die über den Flash-Player angezeigt werden. Mit der Flash-Technologie lassen sich auch interaktive Anwendungen wie Menüsteuerungen erzeugen.

1.1.3 HTML/XHTML - Grundkonzepte

Tags - Die HTML-Elemente

HTML-Dateien sind Text-Dateien. Sie können mit jedem Editor gelesen und bearbeitet werden.

Eine HTML-Datei unterscheidet Texte, die im Browser angezeigt werden. Das sind die 'Nutzdaten' und die HTML-Steuerelemente, mit denen gesteuert wird, wie die Nutzdaten ausgegeben werden.

Die **HTML-Steuerelemente** werden als **Tags** bezeichnet. Jeder Tag wird in spitze Klammern eingeschlossen. Bei den Tags wird zwischen paarweisen und einzelnen Tags unterschieden.

Beispiele für paarweise Tags:

<code><html> </html></code>	Anfang und Ende einer HTML-Datei
<code><head> </head></code>	Anfang und Ende des Kopfbereiches einer HTMLDatei
<code><title> </title></code>	Anfang und Ende des Titels einer HTML-Datei
<code><body> </body></code>	Anfang und Ende des Körpers der HTML-Datei.
<code><p> </p></code>	Absatz (für Text)
<code> </code>	Fett-Schrift
<code><i> </i></code>	Kursiv-Schrift

Die paarweisen Anfangs- und Endtags werden wie Klammern verwendet.

Beispiel für einzelne Tags:

<code>
</code>	Zeilenumbruch
<code><hr></code>	Horizontale Linie

Verschachteln von Tags

Ein Grundprinzip bei der Anordnung ist die Verschachtelung der Tags.

=> SelfHTML -> Verschachtelung von Tags

Beispiel

```
<p>normal <b>fett</b> <i>kursiv</i> <b><i>fett kursiv</i></b></p>
```

Ausgabe:

normal **fett kursiv** ***fett kursiv***

In dem Absatz wird zuerst unformatierter Text ausgegeben, dann fett formatierter Text (**...**), danach kursiver Text (*<i>...</i>*) und abschließend fett-kursive formatierter Text (**<i>...</i>**).

Beachten Sie, dass verschachtelte Tags so angeordnet werden, dass der Tag, der als letzter geschrieben wird, als erster geschlossen wird.

1.1.4 HTML-Grundgerüst

Der grundlegende Aufbau einer HTML-Datei ist vorgegeben. Er wird als HTMLGrundgerüst bezeichnet.

Der HTML-Editor **scriptly** kann das HTML-Grundgerüst direkt beim Erstellen einer Datei erzeugen.

(1.2) Erstellen einer neuen HTML-Datei Weboc-ton-Scriptly

Befehl: Datei => Neu => HTML-Datei

=> Das folgende Dialogfenster wird geöffnet:



Bild 1.1: Dialogfeld zur Festlegung grundlegender Eigenschaften des HTMLGrundgerüsts

Das folgende Skript wurde mit den im Bild dargestellten Einstellungen erzeugt.

HTML-Datei - die mit "Webocton-Scriptly" erzeugt wurde

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>Titel</title>
    <meta http-equiv="content-type" content="text/html;
charset=ISO-8859-1">
    <meta name="description" content="">
    <meta name="author" content="">
    <meta name="keywords" content="">
    <meta name="generator" content="Webocton-Scriptly
(www.scriptly.de)">
  </head>
  <body>

</body>
</html>
```

Die Hierarchie in einer HTML-Datei

1. Ebene: Anfang und Ende einer HTML-Datei (`<html> ... </html>`). Alle Befehle, mit denen die Seite beschrieben wird, müssen innerhalb dieser Klammern stehen.

Die einzige Ausnahme bildet die Dokumenttyp-Deklaration. Sie kam später zum HTMLCode dazu und steht vor dem ersten HTML-Tag.

2. Ebene: Die zweite Ebene besteht aus dem Kopf (`<head> ... </head>`) und dem Körper (`<body> ... </body>`). Der Kopfbereich ist der Vorspann der HTML-Datei und enthält z. B. den Titel der Seite. Im Körper stehen der Inhalt der Webseite.

3. und weitere Ebenen: Die Befehle innerhalb des Kopfes und des Körpers. In der dritten Ebene kann es paarweise Tags geben, in denen weitere Tags stehen können.

Verschachtelung der Tags

Hier wird deutlich, dass die Tags verschachtelt werden. Diese Verschachtelung führt zu einer Baumstruktur, deren Wurzel das `<html>..</html>`-Tag-Paar ist. Die Baumstruktur wird als DOM (Document Object Model) bezeichnet.

Erläuterungen zum HTML-Grundgerüst

Dokumenttyp-Deklaration

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

Die Dokumenttyp-Deklaration ist in HTML optional. Sie gibt an, zu welchem Dokumenttyp die HTML-Seite gehört, und enthält den Verweis auf die Seite, auf der die gültigen Tags und Attribute festgelegt sind. Weitere Informationen zum Dokumenttyp finden Sie in <http://www.selfhtml.org/>.

Ich empfehle Ihnen in diesem Modul, den hier angegebenen Dokumenttyp auszuwählen.

<html>-Tag

Das `<html>`-Tag-Paar ist das Wurzeltag-Paar einer HTML-Seite.

Alle weiteren Tags der HTML-Seite befinden sich innerhalb dieses Tag-Paares.

<head>-Tag

Das `<head>`-Tag-Paar kennzeichnet den Bereich des Seitenkopfs.

In diesem Paar können das `<title>`-Tag-Paar zur Definition des Seitentitels, verschiedene Meta-Tags sowie Tags mit Links zu Stylesheet-Dateien enthalten sein.

Definition des verwendeten Zeichensatzes

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

Mit diesem Meta-Tag wird der verwendete Zeichensatz festgelegt. In diesem Beispiel wird der 8-Bit-Zeichensatz "ISO 8859-1" angegeben. Er ist der Standard-Zeichensatz für Westeuropa und enthält unter anderem die deutschen Sonderzeichen.

Damit wird sichergestellt, dass Webserver und Webbrowser die Zeichen mit demselben Zeichensatz kodieren bzw. dekodieren.

Durch die Definition des Zeichensatzes mit diesem Meta-Tags entfällt die Notwendigkeit, die deutschen Sonderzeichen maskieren zu müssen.

<body>-Tag

Im `<body>`-Tag-Paar werden die Tags mit den Daten eingetragen, die die Seite im Browser anzeigen soll. In diesem Grundgerüst ist der Body-Bereich noch leer.

Weiterführende Informationen

=> [SelfHTML -> Grundgerüst](#)

1.1.5 Zeichenvorrat

(1.3) Regeln für die Texteingabe in eine HTML-Datei

Beim Editieren von HTML- Dateien sind gewisse Regeln zu beachten. Diese dienen dazu, dass die Datei und deren Inhalt im Browser wie gewünscht dargestellt und vom Bearbeiter gut zu lesen ist. wird. Bitte machen Sie sich mit den Grundregeln vertraut:

=> [SelfHTML / Texteingabe](#)

Aufgabe

Klären Sie den Begriff `Whitespace` mit Hilfe des Internets.

Welche Bedeutung haben Whitespaces für die Gestaltung des HTMLQuellcodes.

=> Lösung

Klären Sie den Begriff `Whitespace` mit Hilfe des Internets.

Whitespace: Bereiche, die im Text leer bleiben.

Sie werden durch Leerzeichen, Zeilenumbrüche und Tabulator-Zeichen erzeugt.

Welche Bedeutung haben Whitespaces für die Gestaltung des HTMLQuellcodes.

Wenn im HTML-Quellcode Whitespace eingegeben werden, dann wird für eine Whitespace-Zeichenfolge nur ein Leerzeichen im Browser dargestellt. Dadurch besteht die Möglichkeit, den HTML-Quellcode so zu formatieren, dass er für Menschen gut lesbar ist und ohne dabei Rücksicht auf die Browserdarstellung nehmen zu müssen.

(1.4) Welche Zeichen dürfen in HTML-Daten benutzt werden?

Zeichenvorrat, Umlaute und Sonderzeichen sind genauso zu beachten wie der eingesetzte Code. Nur wer sich hierüber informiert, versteht, warum Text nicht immer so angezeigt wird, wie man es gerne hätte.

Machen Sie sich bitte mit dem Zeichenvorrat und mit der Maskierung bestimmter Zeichen vertraut. Nur so wissen Sie auch, welche Zeichen Sie auf welche Weise einsetzen können.

=> [SelfHTML -> Zeichenvorrat](#)

=> [SelfHTML -> Zeichenreferenz](#)

Aufgabe

Der HTML-Quelcode besteht aus HTML-Elementen (Tags und Attribute innerhalb der Tags) und dem Text, der im Browser angezeigt wird.

Die Betrachtungen zum Zeichenvorrat beziehen sich auf den Text, der im Browser dargestellt werden soll.

Wie können die deutschen Sonderzeichen für HTML maskiert werden?

Müssen die deutschen Sonderzeichen maskiert werden?

Welche Zeichen müssen maskiert werden?

=> Lösung

Wie können die deutschen Sonderzeichen für HTML maskiert werden?

```
ä      &auml;  Ä
&Auml;  Ö
&ouml;  Ö
&Ouml;  ü
&uuml;  Ü
&Uuml;  ß
&szlig;
```

Müssen die deutschen Sonderzeichen maskiert werden?

Sie brauchen nicht maskiert werden, wenn im Header ein Meta-Tag mit dem Zeichensatz angegeben wird. Im oben dargestellten HTML-Grundgerüst ist ein solcher Meta-Tag enthalten:

```
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
```

Welche Zeichen müssen maskiert werden?

Zeichen, die für die HTML-Elemente eine besondere Bedeutung haben, müssen maskiert werden. Das sind die spitzen Klammern, das Anführungs- und das &-Zeichen. Das &Zeichen muss maskiert werden, weil mit ihm alle HTML-Codes beginnen.

```
<      &lt;
>      &gt;
&      &amp;
"      &quot;
```

Dateinamen

Dateinamen sollten eindeutig sein, sie sollten mit dem Namen schon auf den Inhalt verweisen. Auch sollten Sie mit dem Titel übereinstimmen. Dies ist nicht nur gut, wenn man durch Suchmaschinen gefunden werden möchte, sondern dient auch der einfachen Strukturierung.

Die Startdatei

Wenn Sie im Browser eine Adresse eingeben, geben Sie i. allg. keinen Dateinamen ein. Eingegeben wird nur die URL wie <http://www.daa-mws-virtuell.de/>. Übertragen an den Browser wird aber eine Datei.

Die Webserver sind so konfiguriert, dass sie bestimmte Dateinamen automatisch aufrufen, wenn nur die URL übertragen wird. Übliche Namen sind:

```
index.htm index.html  
index.php
```

Diese Dateinamen werden als Default-Dateinamen bezeichnet.

Hinweis

Der automatische Aufruf einer Startdatei in dem Fall, dass keine Datei in der Adresse angegeben wurde, funktioniert nur beim Aufruf über einen Webserver, lokal z.B. über `localhost`.

Wenn eine Webseite direkt von der Festplatte aufgerufen wird, wird immer ein Dateiname benötigt.

(1.5) Aufgaben

Informieren Sie sich über die Konventionen zu Dateinamen für Web-Projekte.

=> SelfHTML -> Dateinamen

Welche Zeichen sollten für Dateinamen nicht verwendet werden?

=> Lösung

Welche Zeichen sollten für Dateinamen nicht verwendet werden?

Verwenden Sie möglichst nur die Zeichen von `a .. z`, `0 .. 9` und den Unterstrich `_`. Schreiben Sie alle Zeichen des Dateinamens klein.

Das Tag-Konzept

Beispiele für paarweise Tags:

<code><html></code>	<code></html></code>	Anfang und Ende einer HTML-Datei
<code><head></code>	<code></head></code>	Anfang und Ende des Kopfbereiches einer HTML-Datei
<code><title></code>	<code></title></code>	Anfang und Ende des Titels einer HTML-Datei
<code><body></code>	<code></body></code>	Anfang und Ende des Körpers der HTML-Datei.
<code><p></code>	<code></p></code>	Absatz (für Text)
<code></code>	<code></code>	Fett-Schrift
<code><i></code>	<code></i></code>	Kursiv-Schrift

Beispiel für einzelne Tags:

<code>
</code>	Zeilenumbruch
<code><hr></code>	Horizontale Linie

Verschachteln von Tags

```
<p>normal <b>fett</b> <i>kursiv</i> <b><i>fett kursiv</i></b></p>
```

HTML-Grundgerüst

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>Titel</title>

    <meta http-equiv="content-type" content="text/html; charset=UTF-
8">
    <meta name="description" content="">
    <meta name="author" content="">
    <meta name="keywords" content="">
    <meta name="generator" content="Webocton-Scriptly
(www.scriptly.de)">

    <link href="style.css" type="text/css" rel="stylesheet">
  </head>  <body>

  </body>
</html>
```

Maskierte Zeichen

Deutschen Sonderzeichen

ä ä Ä
 Ä ö
 ö Ö
 Ö ü
 ü Ü
 Ü ß
 ß

Zeichen, die maskiert werden müssen

< <
 > >
 & &
 " "

Dateinamen im Web

Default-Name

Datei, die aufgerufen wird, wenn eine Adresse ohne Dateiname aufgerufen wird.

Beispiele

index.htm, index.html, index.php

Empfehlung für die Bildung von Dateinamen

Verwenden Sie möglichst nur die Zeichen von a .. z, 0 .. 9 und den Unterstrich _.
 Schreiben Sie alle Zeichen des Dateinamens klein.

1.1.8 Übungen zur Selbstkontrolle

Hinweis:

Die folgenden Übungen können Sie testen, wenn Sie eine HTML-Datei erstellen, dort Ihre Lösungen eingeben und die Datei im Browser aufrufen.

(1.6) Übungen

1. HTML-Grundgerüst

Schreiben Sie das HTML-Grundgerüst ohne Dokumenttyp-Definition und ohne MetaTags auf. Verwenden Sie dabei keine Hilfe.

2. Formatierter Absatz

Gewünschte Browserausgabe:

Das ist ein *formatierter Absatz*

Geben Sie den HTML-Code für diesen Text an.

3. Zeichen maskieren

Gewünschte Browser-Ausgabe:

Es gilt: 10 > 5 ist wahr.
Deutsche Umlaute können maskiert werden:
"ä" - ä
"Ä" - Ä
"ß" - ß

Geben Sie den HTML-Code an. Maskieren Sie die deutschen Sonderzeichen und die Zeichen, die in HTML für die HTML-Elemente verwendet werden

=> Lösung

HTML-Code:

```
<p>Es gilt: 10 &gt; 5 ist wahr.</p> <p>Deutsche Umlaute können  
maskiert werden:<br>  
&quot;&auml;&quot; - &amp;auml;<br>  
&quot;&Auml;&quot; - &amp;Auml;<br>  
&quot;&szlig;&quot; - &amp;szlig;</p>
```

1.2 Texte und deren Formatierung

Stundenempfehlung

FIAE	FISI	ITSK	IFK	ITSE
4	4	4	4	4

[1.2.1 Überblick über dieses Kapitel](#)

[1.2.2 Absätze und Zeilenumbrüche](#)

[1.2.3 Überschriften](#)

[1.2.4 Attribute für Absatz- und Überschriften-Tags](#)

[1.2.5 Tags zur Zeichenformatierung](#)

[1.2.6 Kapitelübersicht](#)

1.2.1 Überblick über dieses Kapitel

Texte, die auf einer Webseite dargestellt werden, werden ähnlich wie in einem Textverarbeitungsprogramm (Microsoft Word, Open Office Writer) in Absätze unterteilt. Um hierarchische Gliederungen innerhalb einer Webseite erzeugen zu können, gibt es spezielle Überschriften-Tags für Absätze.

In diesem Kapitel werden die Tags für Absätze, Überschriften und Zeilenumbrüche sowie erläutert. Weiterhin werden Möglichkeiten zur Formatierung von Texten mit HTML-eigenen Elementen gezeigt.

1.2.2 Absätze und Zeilenumbrüche

Hinweis

Absätze werden grundsätzlich in der HTML-Datei im `<body>...</body>`-Bereich platziert. In den Beispielen gebe ich nur die Tags an. Zum Nachvollziehen der Beispiele sollten sie jeweils ein HTML-Grundgerüst erstellen, in dem Sie die Beispiel-Codes einfügen.

(2.1) Tag für einen Absatz `<p>` `</p>`

p: Paragraph, Absatz

Ein Absatz wird durch `<p>` eingeleitet und durch `</p>` beendet.

Weitere Informationen: [SelfHTML -> Textabsätze](#)

Beispiel

HTML-Datei "a2_texte/absaetze.html"

```
<p>Das ist ein Absatz.</p>

  <p>Das      ist
    der      zweite
Absatz.</p>
```

Beachten Sie insbesondere den zweiten Absatz. Dort sind zusätzliche Leerzeichen vor den Absatz und zwischen den Wörtern und zwei Zeilenumbrüche eingefügt.

Browserausgabe

Das ist ein Absatz.

Das ist der zweite Absatz.

Bild 2.1: Browserausgabe der beiden Absätze

Die Browserausgabe weicht beim 2. Absatz deutlich vom HTML-Quelltext ab. Die zusätzlichen Leerzeichen und Zeilenumbrüche werden im Browser nicht dargestellt.

Aufgabe

Testen Sie das Skript und experimentieren Sie mit der Formatierung des Quellcodes.

Hinweis

In der Browserausgabe befindet sich zwischen den Absätzen ein relativ großer Abstand. Lassen Sie sich jetzt nicht davon stören. Wie die Abstände zwischen zwei Absätzen eingestellt werden, wird im folgenden Modul unter Verwendet von CSS gezeigt.

(2.2) Beispiel: Absatz mit langen Text

Betrachten Sie die Ausgabe des folgenden Textes:

Probieren Sie auch einmal einen Text aus, der länger als eine Zeile im Browser ist. Fügen Sie im Quellcode Zeilenumbrüche ein, damit er nicht zu breit wird. In der Browserdarstellung wird der Text automatisch und unabhängig von den im Quellcode eingefügten Zeilenumbrüchen umgebrochen.

Dieser Text wurde durch den folgenden Quellcode erzeugt:

HTML-Datei "a2_texte/absaetze.html"

```
<p>Probieren Sie auch einmal einen Text aus, der länger  
als eine Zeile im Browser ist. Fügen Sie im Quellcode  
Zeilenumbrüche ein, damit er nicht zu breit wird. In der  
Browserdarstellung wird der Text automatisch und  
unabhängig von den im Quellcode eingefügten Zeilenumbrüchen  
umgebrochen. </p>
```

Vergleichen Sie die Zeilenumbrüche beider Darstellungen. Sie sind voneinander völlig unabhängig. Wenn Sie das ausprobieren, können Sie auch die Breite des Browserfensters ändern. Mit ihr ändern sich auch die Zeilenumbrüche.

(2.3) Abstand zwischen Absätzen sowie zwischen Wörtern erzeugen ` `

Ein leerer Absatz wird durch den folgenden Code erzeugt: `<p></p>`

Beispiel: Abstand zwischen zwei Absätzen erzeugen

HTML-Datei "a2_texte/absaetze.html"

```
<p>1. Absatz</p>  
<p></p>  
<p> </p>  
<p>4. Absatz nach zwei leeren Absätzen</p>
```

Zwischen den beiden Absätzen mit Inhalt befinden sich zwei leere Absätze.

Browseransicht

1. Absatz
4. Absatz nach zwei leeren Absätzen

Bild 2.2: Browseransicht des HTML-Codes mit zwei leeren Absätzen

Die leeren Absätze zeigt der Browser nicht an. Wenn Sie mit Hilfe von Absätzen einen Abstand zwischen Texten erzeugen wollen, dann dürfen die Absätze nicht leer bleiben. Das gleiche gilt übrigens auch für Tabellenzellen.

Das maskierte Leerzeichen ` `

Ein einfaches Leerzeichen im Absatz reicht auch nicht aus, damit er im Browser angezeigt wird. Zu diesem Zweck kann das maskierte Leerzeichen verwendet werden.

heißt übrigens "*no breakable space*", ein Leerzeichen, das keinen Umbruch erzeugt.

Mit ihm können auch Abstände durch Leerzeichen erzeugt werden und Zeilenumbrüche zwischen den beiden damit verbundenen Wörtern verhindert werden.

Aufgabe

Probieren Sie es aus. Fügen Sie das maskierte Leerzeichen in die leeren Absätze des oben dargestellten HTML-Codes ein und rufen sie die Seite erneut im Browser auf.

(2.4) Tag für Zeilenumbrüche `
`

Sie kennen bestimmt die Möglichkeit, wie in einem Textverarbeitungsprogramm innerhalb eines Absatzes ein Zeilenumbruch eingefügt wird (mit Umschalt + Enter). HTML verfügt auch über einen Tag, der Zeilenumbrüche in Absätzen erzeugen kann, der `
`-Tag.

Scriptly stellt ihn in der XHTML-Version `
` mit dem abschließenden Schrägstich zur Kennzeichnung, dass dieser Tag ein einzelner Tag ist. Beide Darstellungsformen haben dieselbe Wirkung.

HTML-Datei "a2_texte/absaetze.html"

```
<p>Absatz<br> mit  
Zeilenumbruch.</p>
```

Experimentieren Sie mit dem `
`-Tag.

(2.5) Präformatierter Absatz `<pre>` `</pre>`

Bei dem Einführungsbeispiel für Absätze haben Sie gesehen, dass der Browser die Darstellung der Texte verändert. Wenn die Notwendigkeit besteht, dass ein Text genauso ausgegeben werden soll, wie er im Quelltext steht, dann kann dafür der `<pre>`-Tag verwendet werden. Alle Zeichen innerhalb des `<pre>` `</pre>`-Tag-Paars werden dann wie eingegeben im Browser ausgegeben. Üblicherweise wird für die

Ausgabe eine dicken gleiche Schrift wie `Courier` oder `Courier New` verwendet.

Weitere Informationen: [SelfHTML -> Präformatiert](#)

HTML-Datei "a2_texte/absaetze.html"

```
<pre>Das    ist
        der    zweite    Absatz.</pre>
```

Browserausgabe

```
Das    ist
        der    zweite    Absatz.
```

Bild 2.3: Präformatierte Ausgabe eines Absatzes

Ich verwende die Möglichkeit der präformatierten Darstellung von Texten bei den abgebildeten HTML-Codes, bei Formeln und auch dort, wo ich genaue Ausrichtungen durch Leerzeichen benötige.

1.2.3 Überschriften

(2.6) Tags für Überschriften `<h1>` `</h1>`

Überschriften werden verwendet, um Texte hierarchisch zu strukturieren.

In diesem Modul ist die Modulüberschrift "1 HTML" eine Überschrift 1. Ordnung. Die Kapitelüberschrift "1.2 Texte und deren Formatierung" ist eine Überschrift 2. Ordnung und die Abschnittsüberschriften wie "1.2.1 Überblick über dieses Kapitel" ist eine Überschrift 3. Ordnung. Weiterhin verwende ich noch Überschriften bis zum 6. Grad. Die Formatierung der Überschriften habe ich mit CSS realisiert.

Überschriften in HTML

Für die Überschriften-Tags verwendet HTML den Buchstaben **h** (haeding).

HTML ermöglicht Überschriften bis zur 6. Ordnung. Die Ordnung der Überschrift wird durch die Zahl hinter **h** angegeben.

Weitere Informationen: [SelfHTML -> Überschrift](#)

Beispiel

HTML-Skript "a2_texte/02_ueberschriften.html"

```
<h1>Überschrift 1. Ordnung</h1>
<h2>Überschrift 2. Ordnung</h2>
<h3>Überschrift 3. Ordnung</h3>
<h4>Überschrift 4. Ordnung</h4>
<h5>Überschrift 5. Ordnung</h5>
<h6>Überschrift 6. Ordnung</h6>
```

Browserausgabe

Überschrift 1. Grades

Überschrift 2. Grades

Überschrift 3. Grades

Überschrift 4. Grades

Überschrift 5. Grades

Überschrift 6. Grades

Bild 2.4: Browserausgabe der Überschriften

Aufgabe

Testen Sie die Überschriften-Tags in einem HTML-Skript.

Hinweise zur Anwendung der Überschriften-Tags

Die Anwendung von Überschriften in HTML-Texten erleichtert die Formatierung mit CSS. Daher sollten Sie Überschriften einsetzen.

Wählen Sie die Ordnung der Überschrift immer nach der Hierarchie-Position der Überschrift aus und nicht nach der Größe der Schrift im Browser ohne CSS-Formatierung.

1.2.4 Attribute für Absatz- und Überschriften-Tags

(2.7) Attribute für HTML-Tags

Attribute der HTML-Tags

Attribute ermöglichen, den HTML-Tags bestimmte Formatierungen zuzuordnen. Sie werden direkt in der Form name="wert" als Name-Wert-Paare in den Tag eingefügt. Es gibt aber auch einige Attribute, die nur aus dem Namen bestehen und keinen Wert haben.

Weitere Informationen: [SelfHTML -> Attribute](#)

Anzeigen der erlaubten Attribute mit Scriptly

Scriptly öffnet ein Menü mit den Attributen, wenn innerhalb des Start-Tags hinter der Tag-Bezeichnung ein Leerzeichen eingegeben wird.

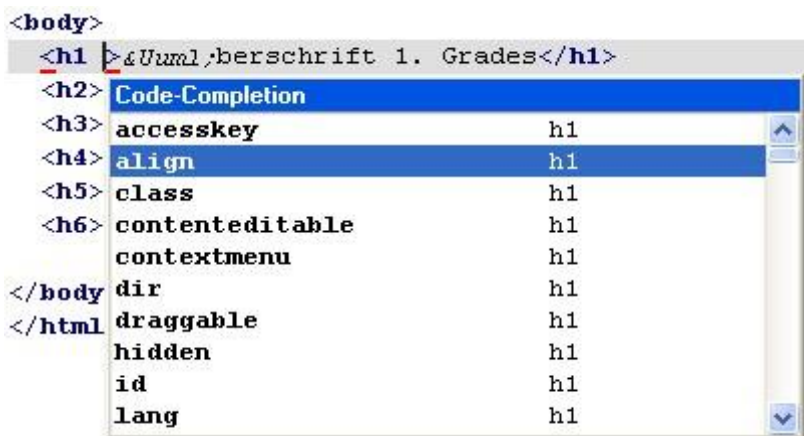


Bild 1.5: Geöffnetes Attribut-Menü "Code-Completion" für den <h1>-Tag

Ein Teil der Attribute werden für die ereignisgesteuerte Bedienung unter Verwendung mit JavaScript (alle Attributnamen, die mit on beginnen) verwendet, ein weiterer Teil für die Zuordnung von CSS-Formatierungen (class, id) usw.. Übrig bleiben nur wenige Attribute, die für die Formatierung wirklich eine Rolle spielen.

Beispiel: Align-Attribut

Ich möchte hier auf das align-Attribut eingehen.

align: Ausrichtung

Mit dem Align-Attribut kann die Ausrichtung des Absatzes (links, zentriert, ..) festgelegt werden.

align=	Bedeutung
--------	-----------

"left"	zinksbündig
"center"	zentriert
"right"	rechtbündig
"justify"	Blocksatz

Zuweisen des Attributwertes mit Scriptly

Nach der Auswahl des Attributs wird es in den Tag eingefügt und die Schreibmarke befindet sich an der Position, in der der Attribut-Wert eingegeben wird. Falls die möglichen Werte vorgegeben sind, wird ein weiteres Menü geöffnet.

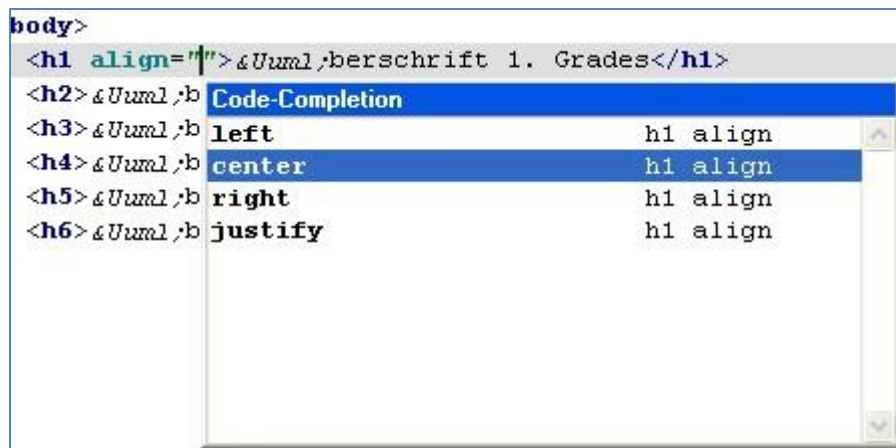


Bild 2.6: Geöffnetes Menü zur Auswahl des Wertes zum Attribut

Aufgabe

Weisen Sie den ersten vier Überschriften die vier möglichen align-Attribute zu und testen Sie die Ausgabe im Browser.

=> Lösung

```
<h1>Überschrift 1. Ordnung</h1>
<h2>Überschrift 2. Ordnung</h2>
<h3>Überschrift 3. Ordnung</h3>
<h4>Überschrift 4. Ordnung</h4>
<h5>Überschrift 5. Ordnung</h5>
<h6>Überschrift 6. Ordnung</h6>
```

Browseransicht

Überschrift 1. Grades

Überschrift 2. Grades

Überschrift 3. Grades

Überschrift 4. Grades

1.2.5 Tags zur Zeichenformatierung

Neben der Möglichkeit, durch Attribute der Tags Formatierungen durchzuführen, verfügt HTML auch über Tags, die Formatierungen ermöglichen. Diese Tags können zur Zeichenformatierung eingesetzt werden.

(2.8) Zeichenformatierung in HTML

Ausgewählte Tags zur Zeichenformatierung

Tag-Paar	Formatierung
<code> </code>	fett
<code><i> </i></code>	kursiv
<code><u> </u></code>	unterstrichen
<code><sub> </sub></code>	tief gestellt
<code><sup> </sup></code>	hoch gestellt

Weitere Informationen: SelfHTML -> Physische Formatierung

Im vorherigen Kapitel haben Sie bereits Beispiele für die Zeichenformatierung kennen gelernt.

Übung

Die folgende Browserausgabe soll erzeugt werden.

Der Wert von **x^2** mit *x* = 7 ist **49**.

Bild 2.8: Beispiel für die Zeichenformatierung

Geben Sie den HTML-Code an, um diese Ausgabe zu erzeugen.

=> Lösung

```
<p>Der Wert von <b>x<sup>2</sup></b>  
mit <i>x = 7</i> ist <b>49</b>.</p>
```

(2.9) Schriftformatierung ** **

Zur zeichenweisen Formatierung der Schrift stellt HTML den ****-Tag bereit. Mit dem ****-Tag können die Schriftart, die Schriftfarbe und die Schriftgröße festgelegt werden.

Hinweis

Der ****-Tag ist als "missbilligt" (engl. deprecated) eingestuft und sollen durch CSSFormatierungen ersetzt werden. D.h. sie sollen nicht mehr benutzt werden. Sie werden hier nur aufgeführt, damit Sie auch ohne CSS die Möglichkeit haben, die HTML-Seiten farblich ein wenig zu gestalten.

Attribute des -Tags

Attribut	Erläuterung
face	Schriftart festlegen. Als Werte sind der oder durch Kommas getrennt die Namen der Schriftarten anzugeben.
color	Schriftfarbe festlegen. Als Wert wird eine Farbe in Hexadezimaldarstellung anzugeben im RGB-Format nach dem Muster "FF0000". Es können auch HTML-Farbname wie "red" angegeben werden.
size	Schriftgröße festlegen. Die Größe kann auf einer Skala von 1 bis 7 als ganze Zahl angegeben werden. Möglich sind auch relative Änderungen der Schriftgröße in der Form +2 oder -1.

Beispiel

HTML-Datei "a2_texte/04_schriftformatierung.html"

```
<p><font color="#800080" size="3" face="Arial Black">
  Das ist ein formatierter Text.</font></p>
```

Weitere Informationen: [SelfHTML -> Schriftformatierung](#)

Aufgabe

Beantworten Sie mit Hilfe des angegebenen SelfHTML-Links die folgenden Fragen:

- 1 Wo darf der -Tag im HTML-Code eingesetzt und wo nicht eingesetzt werden?
- 2 Wie muss die Standard-konforme Formatierung mehrerer aufeinanderfolgender <p>-Tags mit derselben Formatierung erfolgen?
- 3 Welche Probleme treten bei der Festlegung einer Schriftart auf?

=> Lösung

- 1 Wo darf der -Tag im HTML-Code eingesetzt und wo nicht eingesetzt werden?

In SelfHTML wird davon gesprochen, dass -Tag nur Inline-Elemente (z. B. die Tags zur Zeichenformatierung) enthalten darf. Block-Elemente, zu denen Absätze, Tabellen, Aufzählungen und Listen gehören, darf er nicht enthalten.

- 2 Wie muss die Standard-konforme Formatierung mehrerer aufeinanderfolgender <p>-Tags mit derselben Formatierung erfolgen?

Der ``-Tag muss in jedem `<p>`-Tag eingetragen werden.

3 Welche Probleme treten bei der Festlegung einer Schriftart auf?

Der Zeichensatz mit dem angegebenen Namen muss auf dem Rechner installiert werden, auf dem die Seite aufgerufen wird.

Das ist insbesondere bei unterschiedlichen Betriebssystemen problematisch. Verdana und Arial gibt es nur auf Windows-Systemen, auf Linux- und MAC-Systemen heißt der vergleichbare Zeichensatz Helvetica. Zur Sicherheit sollten, wenn Zeichensätze angegeben werden, alle drei Zeichensätze angegeben werden.

1.2.6 Kapitelübersicht

Tags für Absätze

<code><p></code>	<code></p></code>	Absatz
<code>
</code>		Zeilenumbruch
<code><pre></code>	<code></pre></code>	Präformatierter Absatz (wie im HTML-Quellcode)
<code><h1></code>	<code></h1></code>	Überschrift 1. Ordnung
<code><h2></code>	<code></h2></code>	Überschrift 2. Ordnung
<code><h3></code>	<code></h3></code>	Überschrift 3. Ordnung
<code><h4></code>	<code></h4></code>	Überschrift 4. Ordnung
<code><h5></code>	<code></h5></code>	Überschrift 5. Ordnung
<code><h6></code>	<code></h6></code>	Überschrift 6. Ordnung

Attribute zur Formatierung der Absatz-Tags

`align="center"` Ausrichtung des Absatzes

Tags zur Zeichenformatierung

<code></code>	<code></code>	fett
<code><i></code>	<code></i></code>	kursiv
<code><u></code>	<code></u></code>	unterstrichen
<code><sub></code>	<code></sub></code>	tief gestellt
<code><sup></code>	<code></sup></code>	hoch gestellt

1.3 HTML-Seiten verlinken

Stundenempfehlung

FIAE	FISI	ITSK	IFK	ITSE
4	4	4	4	4

[1.3.1 Überblick über dieses Kapitel](#)

[1.3.2 Planung der Verlinkung mehrerer Seiten mit dem Sitemap](#)

[1.3.3 Hyperlinks einfügen](#)

[1.3.4 Weiteres zu Hyperlinks](#)

[1.3.5 Kapitelschwerpunkte](#)

1.3.1 Überblick über dieses Kapitel

Im Namen HTML steckt "Hypertext". Mit Hypertext wird die Möglichkeit verstanden, von einer Seite durch Anklicken eines Hyperlinks auf eine andere Seite wechseln zu können. Die Sprünge können zu einer beliebigen Seite im Internet erfolgen.

In diesem Kapitel wird zunächst gezeigt, wie die Navigationsstruktur einer Website durch einen Seitenbaum sichtbar gemacht werden kann. Dann wird erläutert, wie Hyperlinks in HTML-Seiten eingefügt werden und wie innerhalb einer Seite Sprungziele definiert und wie sie angesprungen werden.

1.3.2 Planung der Verlinkung mehrerer Seiten mit der Sitemap

Situation

Für eine Website sind die folgenden Seiten geplant:

```

Home                               (Startseite)
Shop
  Rechner
  Drucker
Service
  Rechner
  Drucker
Impressum
Sitemap
  
```

Die eingerückte Struktur soll die Menüebenen kennzeichnen.

Vorgaben für die Verlinkung

Für die aufgeführten Seiten soll die Verlinkung erstellt werden. Die Verlinkung soll so erfolgen, dass von jeder Seite aus die Seiten der 1. Ebene aufgerufen werden

können. Die Seiten der 2. Ebene sollen nur von den übergeordneten Seiten aufgerufen werden können.

(3.1) Übung

1 Sitemap erstellen

Stellen Sie die Menüstruktur der Website in einer Baumstruktur (Sitemap) dar. Geben Sie für die einzelnen Seiten den geplanten Dateinamen an.

Vervollständigen Sie die vorliegende Baumstruktur.

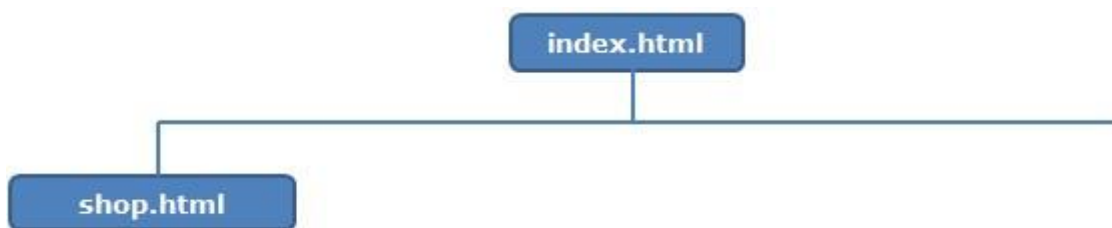


Bild 3.1: Vorbereitete Baumstruktur für geplante Website

=> Lösung

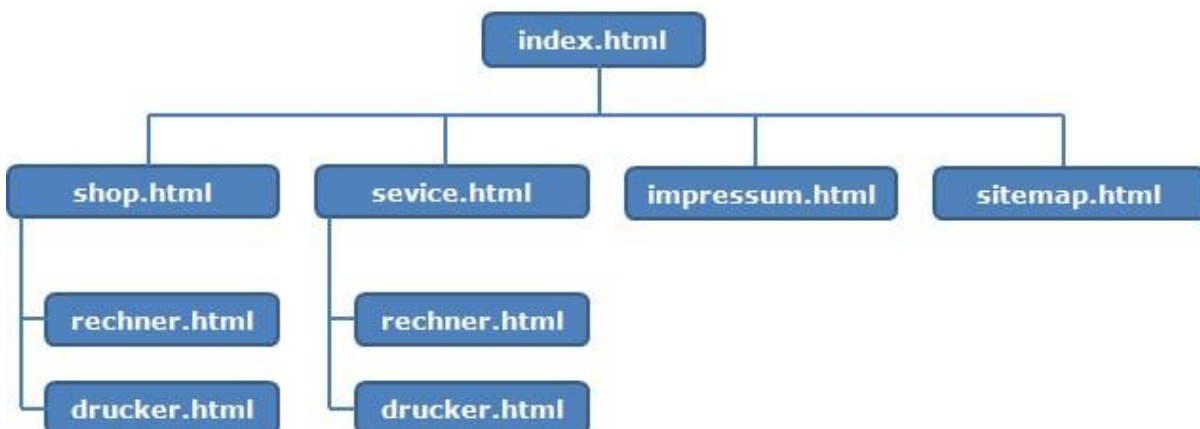


Bild 3.2: Vollständige Baumstruktur für die geplante Website

2 HTML-Seiten vorbereiten

Diese Seitenstruktur soll praktisch realisiert werden. Die Seiten sollen nur eine Überschrift sowie die Hyperlinks enthalten.

Auf die Startseite sollen alle Seiten verlinkt werden. Die Hyperlinks der 1. Ebene sollen wie in der Baumstruktur dargestellt horizontal angeordnet werden und die Hyperlinks der 2. Ebene sollen in den betreffenden Dateien vertikal an der linken Seite angeordnet werden

Bearbeitungshinweis

Erstellen Sie eine Datei als Vorlage. Die restlichen Dateien werden später, wenn die Hyperlinks hinzugefügt worden sind, durch Kopieren dieser Vorlagedatei erstellt. Dann werden die Hyperlinks gleich mit kopiert.



Bild 3.3: Musterlösung für die Vorlagedatei "shop.html"

1.3.3 Hyperlinks einfügen

Hyperlinks

Hyperlinks oder kurz "Links" werden im deutschen als Verweise bezeichnet.

Aufgabe

Informieren Sie sich bitte zunächst auf [SelfHTML -> Verweise](#) über Allgemeines und über Beispiele zu Hyperlinks.

Hyperlinks mit dem <a>-Tag definieren

Hyperlinks werden durch <a>-Tags mit dem Attribut href erzeugt. a steht für anchor (dt. Anker) und href für hyper reference (dt. Hypertext-Referenz). href erwartet eine absolute oder eine relative Adresse zu der betreffenden Seite.

Beispiele:

Absolute Adresse

```
<p><a href="http://www.daa-mws-virtuell.de">MWS der DAA</a></p>
```

Der Text innerhalb der beiden Tags zeigt der Browser als Hyperlink an und kann angeklickt werden. Hyperlinks werden innerhalb von Texten definiert. Die Texte sollten

zur Einhaltung der Standardkonformität in Textabsätzen oder alternativ in Tabellenzellen

Absolute Adressen führen aus dem Projekt raus zu anderen Websites im Netz.

Relative Adresse

```
<p><a href="/shop/artikel1.html">Artikel 1</a></p>
```

```
<p><a href="artikel1.html">Artikel 1</a></p>
```

```
<p><a href="../shop/artikel1.html">Artikel 1</a></p>
```

Relative Adressen werden bei Hyperlinks verwendet, die Dateien innerhalb Ordnerstruktur der eigenen Website aufrufen.

1. Der erste Beispiellink beginnt mit einem Schrägstrich "/", das bedeutet, dass die Pfadangabe bei der Wurzel des Datenverzeichnisses des Webserver beginnt. Aufgerufen wird die Datei "artikel1.html" im Verzeichnis "/shop".
2. Der 2. Hyperlink ruft die Datei "artile1.html" auf, die sich im selben Verzeichnis befindet wie die Datei, in der sich der Hyperlink befindet.
3. Der 3. Hyperlink geht ausgehend von der Datei, in der sich der Hyperlink befindet, ein Verzeichnis zurück und von dort in das Verzeichnis "shop", in dem die Datei "artikel1.html" aufgerufen wird.

(3.2) Übung

Erstellen Sie die Dateien, die in Aufgabe 3.1 geplant und vorbereitet wurde, mit den Hyperlinks. Nutzen Sie die Möglichkeit aus, so viele Dateien wie möglich zu kopieren.

Die folgende Übersicht zeigt die zu erstellenden Seiten mit den Hyperlinks. Die Seite "Sitemap" soll zusätzlich zum vertikalen Hauptmenü auf alle HTML-Seiten Hyperlinks in Form einer Baumstruktur erhalten.

Home

[Home](#) [Shop](#) [Service](#) [Impressum](#) [Sitemap](#)

Inhalt der Seite

Shop

[Home](#) [Shop](#) [Service](#) [Impressum](#) [Sitemap](#)

[Rechner](#)
[Drucker](#)

Shop- Rechner

[Home](#) [Shop](#) [Service](#) [Impressum](#) [Sitemap](#)

[Rechner](#)
[Drucker](#)

Shop - Drucker

[Home](#) [Shop](#) [Service](#) [Impressum](#) [Sitemap](#)

[Rechner](#)
[Drucker](#)

Service

[Home](#) [Shop](#) [Service](#) [Impressum](#) [Sitemap](#)

[Rechner](#)
[Drucker](#)

Service - Rechner

[Home](#) [Shop](#) [Service](#) [Impressum](#) [Sitemap](#)

[Rechner](#)
[Drucker](#)

Service - Drucker

[Home](#) [Shop](#) [Service](#) [Impressum](#) [Sitemap](#)

[Rechner](#)
[Drucker](#)

Impressum

[Home](#) [Shop](#) [Service](#) [Impressum](#) [Sitemap](#)

Inhalt der Seite

Sitemap

[Home](#) [Shop](#) [Service](#) [Impressum](#) [Sitemap](#)

[Home](#)
[Shop](#)
- [Rechner](#)
- [Drucker](#)
[Service](#)
- [Rechner](#)
- [Drucker](#)
[Impressum](#)
[Sitemap](#)

Bild 3.4: Übersicht über die HTML -Seiten

Das folgende Bild zeigt die Ordnerstruktur. Die HTML-Dateien der untergeordneten Seiten sollen in Unterordnern gespeichert werden.



Bild 3.5: Ordnerstruktur für die Dateien des Projektes "sitemap1"

Aufgabe

Erstellen Sie jetzt alle 9 HTML-Dateien mit den abgebildeten Titeln und Hyperlinks.

1.3.4 Weiteres zu Hyperlinks

Die wesentlichen Grundlagen zu Hyperlinks haben Sie im vorherigen Abschnitt kennengelernt.

Offen sind jetzt noch die folgenden Fragen:

- **Hyperlinks zu einer Position innerhalb einer Seite**
Wie kann ein Hyperlink auf eine Position innerhalb einer (langen) Seite positioniert werden. Oder wie kann ein Inhaltsverzeichnis für eine Seite erstellt werden, wie es dieses Kapitel enthält.
- **Dateien zum Download bereitstellen**
Wie kann eine Datei, z.B. eine ZIP-Datei, mit Hilfe eines Hyperlinks zum Download bereitgestellt werden.
- **Zielfenster für Hyperlinks festlegen**
Wie kann festgelegt werden, dass ein Hyperlink ein neues Fenster öffnet.

Aufgabe

Bitte lesen Sie sich zunächst die folgenden SelfHTML-Seiten durch:

- [SelfHTML -> Projekt-externe Verweise](#)
- [SelfHTML -> Projekt-interne Verweise](#)
- [SelfHTML -> Zielfenster bestimmen](#)
- [SelfHTML -> Download-Verweise](#)

•

Beantworten Sie die folgenden Fragen:

1- Wie wird ein Anker (eine Textmarke) definiert? Geben sie ein Beispiel an.

2- Wie wird ein Hyperlink definiert, der in der eigenen HTML-Datei zum in der ersten Frage definierten Anker springen soll?

3- Wie wird ein Hyperlink definiert, der die Datei "abc.zip" zum Download bereitstellen soll.

4- Wie wird ein Hyperlink definiert, der die Seite "xyz.html" aufrufen soll und in einem neuen Fenster öffnen soll.

=> Lösung

1- Wie wird ein Anker (eine Textmarke) definiert? Geben sie ein Beispiel an.

Anker werden durch die Tags der Form

```
<p><a name="anker1">Hier ist 'Anker1'</a></p>
```

definiert. Der Name des Ankers (der Textmarke) ist "Anker 1".

Der Anker wird auf der HTML-Seite wie folgt dargestellt:

Hier ist 'Anker1'

2- Wie wird ein Hyperlink definiert, der in der eigenen HTML-Datei zum in der ersten Frage definierten Anker springen soll?

```
<p><a href="#anker1">Zu 'Anker 1'</a></p>
```

In der Adresse wird der Name des oben definierten Ankers als Hyperlink ohne den Dateinamen mit einem vorangestellten #-Zeichen eingetragen.

Auf der HTML-Seite wird dieser Hyperlink wie folgt dargestellt:

Zu 'Anker 1'

3- Wie wird ein Hyperlink definiert, der die Datei "abc.zip" zum Download bereitstellen soll.

Der Hyperlink wird unter Angabe der Adresse der Datei wie ein normaler Hyperlink erstellt. Der Browser entscheidet an Hand der Dateityps, ob die Datei im Browser geöffnet oder zum Download bereitgestellt wird.

```
<p><a href="abc.zip">Download der Datei 'abc.zip'</a></p>
```

Der Hyperlink wird auf der HTML-Seite wie folgt dargestellt:

Download der Datei 'abc.zip'

4 Wie wird ein Hyperlink definiert, der die Seite "xyz.html" aufrufen soll und in einem neuen Fenster öffnen soll.

Um für einen Hyperlink festzulegen, dass das Ziel standardmäßig in einem neuen Fenster geöffnet werden soll, wird das `target`-Attribut verwendet. Es enthält verschiedene Werte, von denen die meisten für Frames verwendet werden. Für ein neues Fenster wird der Wert `"_blank"` verwendet.

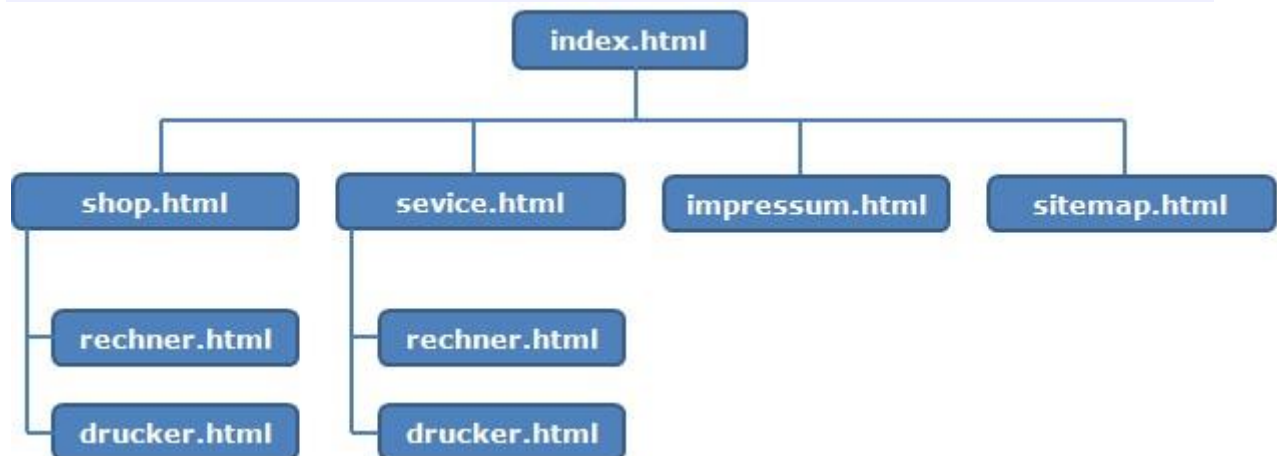
```
<p><a target="_blank" href="#anker1">Zu 'Anker 1' im neuen
Fenster</a></p>
```

Darstellung im Browser:

[Zu 'Anker 1' im neuen Fenster](#)

1.3.5 Kapitelschwerpunkte

Beispiel für eine Baumstruktur zur Darstellung der Seiten-Hierarchie



Hyperlinks

Absolute Adresse

```
<p><a href="http://www.daa-mws-virtuell.de">MWS der DAA</a></p>
```

Relative Adresse

```
<p><a href="/shop/artikel1.html">Artikel 1</a></p>
```

```
<p><a href="artikel1.html">Artikel 1</a></p>
```

```
<p><a href="../shop/artikel1.html">Artikel 1</a></p>
```

Anker (Textmarke)

```
<p><a name="anker1">Hier ist 'Anker1'</a></p>
```

Hyperlink zu einem Anker

```
<p><a href="#anker1">Zu 'Anker 1'</a></p>
```

Hyperlink zum Download einer ZIP-Datei

```
<p><a href="abc.zip">Download der Datei 'abc.zip'</a></p>
```

Öffnen des Hyperlink-Ziels in einem neuen Fenster

```
<p><a target="_blank" href="#anker1">
```

```
Zu 'Anker 1' im neuen Fenster</a></p>
```

1.4 Listen und Aufzählungen

Stundenempfehlung

FIAE	FISI	ITSK	IFK	ITSE
4	4	4	4	4

[1.4.1 Überblick über dieses Kapitel](#)

[1.4.2 Einfache Listen und Aufzählungen](#)

[1.4.3 Verschachtelte Listen und Aufzählungen](#)

[1.4.4 Formatierung des Listen/Aufzählungen](#)

[1.4.5 Kapitelschwerpunkte](#)

[1.4.6 Übungen](#)

1.4.1 Überblick über dieses Kapitel

Listen und Aufzählungen ermöglichen die übersichtliche Darstellung von Informationen zu einem Sachverhalt oder Gegenstand.

HTML stellt eigene Tags bereit, um Listen und Aufzählungen definieren zu können. Sie können verschachtelt werden und ihre Darstellung kann mit Hilfe von HTML-Attributen beeinflusst werden.

In Zusammenhang mit CSS werden Listen zur Definition von horizontalen und vertikalen Menüs verwendet, die auch verschachtelt sein können.

In diesem Kapitel werden die Grundlagen zu Listen und Aufzählungen behandelt.

Weitere Informationen zu Listen und Aufzählungen finden Sie in: [SelfHTML -> Listen](#)

1.4.2 Einfache Listen und Aufzählungen

(4.1) HTML-Tags Für Listen und Aufzählungen

Liste

- Element 1
- Element 2
- Element 3

```
<ul>
  <li>Element 1</li>
  <li>Element 2</li>
  <li>Element 3</li>
</ul>
```

Eine Liste wird durch das Tag-Paar ` .. ` begrenzt. **ul** steht für **unordered list** (ungeordnete Liste). Die einzelnen Elemente der Liste werden durch Tag-Paare ` .. ` (**list item**,) definiert.

Aufzählung

1. Element 1
2. Element 2
3. Element 3

```
<ol>
  <li>Element 1</li>
  <li>Element 2</li>
  <li>Element 3</li>
</ol>
```

Die HTML-Syntax der Aufzählung unterscheidet sich nur in den begrenzenden Tags von der Liste. Sie wird durch ` .. ` -Tags begrenzt. **ol** steht für **ordered list** (nummerierte Liste).

Aufgabe

Erstellen Sie eine HTML-Datei, erstellen Sie darin eine Liste und eine Aufzählung und rufen Sie die Datei im Browser auf.

1.4.3 Verschachtelte Listen und Aufzählungen

Listen können in zwei Richtungen erweitert werden. Innerhalb eines Listenelements können Absätze und vollständige Listen eingefügt werden.

Liste mit Absätzen in einem Element

<ul style="list-style-type: none"> □ Element 1 <ul style="list-style-type: none"> Absatz 1.1 Absatz 1.2 □ Element 2 	<pre> Element 1 <p>Absatz 1.1</p> <p>Absatz 1.2</p> Element 2 </pre> <p>← Äußere Liste</p> <p>← Absätze in einem Listenelement</p>
--	--

Wenn in einem Listenelement längere Texte geschrieben werden sollen, die ohne Listenzeichen, aber auf der gleichen Höhe wie die Listenelemente angezeigt werden sollen, dann können Sie durch normale Absätze innerhalb der ` .. `-Tags erzeugt werden.

Verschachtelte Listen/Aufzählungen

<pre> 1. Element 1 • Element 1.1 1. Element 1.1.1 2. Element 1.1.2 • Element 1.2 2. Element 2 </pre>	<pre> Element 1 Element 1.1 Element 1.1.1 Element 1.1.2 Element 1.2 Element 2 </pre> <p>Äußere Liste</p> <p>1. Verschachtelung</p> <p>2. Verschachtelung</p>

Listen und Aufzählungen können verschachtelt werden. Die Verschachtelung erfolgt so, dass eine neue Liste/Aufzählung in einem ` .. `-Tag eingetragen wird.

Verschachtelungen und Absätze gemeinsam anwenden

<ul style="list-style-type: none"> • Element 1 • Element 2 <p>Absatz 2.1</p> <ol style="list-style-type: none"> 1. Element 2.1 2. Element 2.2 <p>Absatz 2.2</p> <ul style="list-style-type: none"> • Element 3 	<pre> Element 1 Element 2 <p>Absatz 2.1</p> Element 2.1 Element 2.2 <p>Absatz 2.2</p> Element 3 </pre> <p>← Äußere Liste</p> <p>← Absatz im Listenelement</p> <p>← Verschachtelte Aufzählung im selben Listenelement</p>
---	--

Absätze und Listen/Aufzählungen können gemeinsam in einem Listenelement der übergeordneten Liste eingetragen werden.

(4.2) Übung

Erzeugen Sie den HTML-Code für die folgende Liste einschließlich der Formatierungen:

<ul style="list-style-type: none"> • Computer <ul style="list-style-type: none"> ◦ Mainboard <p><i>Ausgewählte Mainboard-Komponenten</i></p> <ul style="list-style-type: none"> ▪ Prozessor ▪ RAM ▪ Grafikkarte ◦ Netzteil ◦ Gehäuse • Display • Eingabegeräte <ul style="list-style-type: none"> ◦ Tastatur ◦ Maus ◦ Scanner • Drucker <p><i>Druckertypen:</i></p> <ul style="list-style-type: none"> ◦ Laserdrucker ◦ Tintenstrahldrucker
--

=> Lösung

```
<ul>
  <li><b>Computer</b>
    <ul type="circle">
      <li>Mainboard
        <p><i>Ausgewählte Mainboard-Komponenten</i></p>
        <ul type="square">
          <li>Prozessor</li>
          <li>RAM</li>
          <li>Grafikkarte</li>
        </ul>
      </li>
      <li>Netzteil</li>
      <li>Gehäuse</li>
    </ul>
  </li>
  <li><b>Display</b></li>
  <li><b>Eingabegeräte</b>
    <ul type="circle">
      <li>Tastatur</li>
      <li>Maus</li>
      <li>Scanner</li>
    </ul>
  </li>
  <li><b>Drucker</b>
    <p><i>Druckertypen:</i></p>
    <ul type="circle">
      <li>Laserdrucker</li>
      <li>Tintenstrahldrucker</li>
    </ul>
  </li>
</ul>
```


1.4.4 Formatierung des Listen/Aufzählungen

Zur Formatierung von Listen und Aufzählungen stellt HTML einige Attribute bereit. Beeinflusst werden können die Symbole bei Listen und die Art der Nummerierung bei Aufzählungen. Weiterhin kann der Anfangswert und der Wert eines Listenelements bei Aufzählungen gesetzt werden.

(4.3) Übung

Informieren Sie sich auf [SelfHTML -> Listen -> HTML-Eigenschaften](#) über die Formatierungsmöglichkeiten von Listen/Aufzählungen.

Erstellen Sie für die folgenden drei Listen den HTML-Code:

1. Liste mit Quadraten als Symbolen

- Element 1
- Element 2

2. Aufzählung mit kleinen Buchstaben

- a. Element 1
- b. Element 2
- c. Element 3

3. Aufzählung mit römischen Zahlen

- I. Element 1
- II. Element 2
- III. Element 3
- IV. Element 4

=> Lösung

1. Liste mit Quadraten als Symbolen

- Element 1
- Element 2

```
<ul type="square">
  <li>Element 1</li>
  <li>Element 2</li>
</ul>
```

2. Aufzählung mit kleinen Buchstaben

<ul style="list-style-type: none"> a. Element 1 b. Element 2 c. Element 3 	<pre><ol type="a"> Element 1 Element 2 Element 3 </pre>
--	---

3. Aufzählung mit römischen Zahlen

<ul style="list-style-type: none"> I. Element 1 II. Element 2 III. Element 3 IV. Element 4 	<pre><ol type="I"> Element 1 Element 2 Element 3 Element 4 </pre>
--	--

1.4.5 Kapitelschwerpunkte

HTML-Struktur einer Liste/Aufzählung

```
<ul>
  <li>Element 1</li>
  <li>Element 2</li>
  <li>Element 3</li>
</ul>
<ol>
  <li>Element 1</li>
  <li>Element 2</li>
  <li>Element 3</li>
</ol>
```

Verschachteln von Listen/Aufzählungen

```
<ul>
  <li>Element 1
    <p>Absatz 1.1</p>
    <p>Absatz 1.2</p>
  </li>
  <li>Element 2</li>
</ul>
```

← Äußere Liste

← Absätze in einem Listenelement

```
<ol>
  <li>Element 1
    <ul>
      <li>Element 1.1
        <ol>
          <li>Element 1.1.1</li>
          <li>Element 1.1.2</li>
        </ol>
      </li>
      <li>Element 1.2</li>
    </ul>
  </li>
  <li>Element 2</li>
</ol>
```

Äußere Liste

1. Verschachtelung

2. Verschachtelung

Formatieren von Listen/Aufzählungen

```
<ul type="square">
  <li>Element 1</li>
  <li>Element 2</li>
</ul>
<ol type="a">
  <li>Element 1</li>
  <li>Element 2</li>
  <li>Element 3</li>
</ol>
```

1.4.6 Übungen

(4.4) Erstellen einer Liste

Erstellen Sie eine Übersicht über Netzwerkhardware als Baumstruktur in Listenform. Verwenden Sie auch Absätze, z. B. für Erläuterungen und Hervorhebungen (fett, kursiv).

Die Liste soll mindestens 15 Einträge und drei Verschachtelungsebenen haben.

Hinweis:

Nutzen Sie diese Übung gleichzeitig zur Wiederholung wichtiger Begriffe zur NetzwerkHardware.

=> Lösung

- **Koppelemente**
 - **Repeater**

Layer 1

 - *Signalverstärkung*
 - **Hub**

Layer 1
 - **Switch**

Layer 2

 - Frameweiterleitung
 - Cut-Through
 - Store-and-Forward
 - Fragment-Free
 - Überlastverhalten
 - Back-Pressure
 - Flow-Control
 - Head-of-Line-Blocking
 - Switch-Port-Eigenschaften
 - MDI/MDIX/Auto-MDI
 - Autosensing
 - Autonegotiation
- **Signalübertragung**
 - **Kabel**
 - Koaxialkabel
 - Twisted-Pair-Kabel
 - Lichtwellenleiter-Kabel
 - Multimode Stufenindex
 - Multimode Gradientenindex
 - Monomode
 - **Stecker/Buchsen**
 - RJ45
 - **Patch-Panele**

```
<ul>
  <li><b>Koppelemente</b>
    <ul type="circle">
      <li><b>Repeater</b>
        <p><i>Layer 1</i></p>
        <ul type="square">
          <li><i>Signalverstärkung</i></li>
        </ul>
      </li>
      <li><b>Hub</b>
        <p><i>Layer 1</i></p>
      </li>
      <li><b>Switch</b>
        <p><i>Layer 2</i></p>
        <ul type="square">
          <li>Frameweiterleitung</li>
        </ul>
      </li>
    </ul>
  </li>
</ul>
```

```

        <li>Cut-Through</li>
        <li>Store-and-Forward</li>
<li>Fragment-Free</li>
        <li>Überlastverhalten</li>
        <li>Back-Pressure</li>
        <li>Flow-Control</li>
        <li>Head-of-Line-Blocking</li>
        <li>Switch-Port-Eigenschaften</li>
        <li>MDI/MDIX/Auto-MDI</li>
        <li>Autosensing</li>
        <li>Autonegotiation</li>
    </ul>
</li>
</ul>
</li>
<li><b>Signalübertragung</b>
    <ul type="circle">
        <li><b>Kabel</b>
            <ul type="square">
                <li>Koaxialkabel</li>
                <li>Twisted-Pair-Kabel</li>
                <li>Lichtwellenleiter-Kabel</li>
                <li>Multimode Stufenindex</li>
                <li>Multimode Gradientenindex</li>
                <li>Monomode</li>
            </ul>
        </li>
        <li><b>Stecker/Buchsen</b>
            <ul type="square">
                <li>RJ45</li>
            </ul>
        </li>
        <li><b>Patch-Panele</b>
        </li>
    </ul>
</li>
</ul>

```

1.5 Tabellen

Stundenempfehlung

FIAE	FISI	ITSK	IFK	ITSE
4	4	4	4	4

[1.5.1 Überblick über dieses Kapitel](#)

[1.5.2 Grundlegender Aufbau einer Tabelle](#)

[1.5.3 Ausgewählte Attribute für Tabellen](#)

[1.5.4 Zellen verbinden](#)

[1.5.5 Kapitelschwerpunkte](#)

[1.5.6 Übungen](#)

1.5.1 Überblick über dieses Kapitel

HTML bietet auch die Möglichkeit, Tabellen zu erstellen. Tabellen ermöglichen die ausgerichtete Ausgabe von Daten und beispielsweise auch von Formularen.

In der Zeit, als die Browser noch nicht hinreichend bzw. standardgerecht die Positionierung mit CSS unterstützten, wurden Tabellen auch zur Layout-Gestaltung von Webseiten eingerichtet. Der Einsatz von Tabellen für diesen Zweck ist heute mittlerweile verpönt. Das bezieht sich aber nicht auf die Einsatzgebiete, die oben genannt wurden.

In diesem Kapitel wird der Grundaufbau einer Tabelle behandelt, dann folgen HTML-Formatierungen für Tabellen und abschließend das Verbinden von Zellen. Diese Themen stellen eine gute Einführung in das Thema HTML-Tabellen dar, behandeln es aber nicht umfassend.

1.5.2 Grundlegender Aufbau einer Tabelle

In diesem Abschnitt wird der grundlegende Aufbau von Tabellen behandelt. Vertiefende Informationen finden Sie in: [SelfHTML -> Tabellen -> Aufbau](#)

(5.1) Tabellen-Grundstruktur

Tags für eine Tabelle

<code><table> ... </table></code>	Äußere Begrenzung der Tabelle
<code><tr> ... </tr></code>	Begrenzen eine Tabellenzeile (Table Row)
<code><td> ... </td></code>	Zelle einer Tabelle für die Daten (Table Data)

Reihenfolge des Erstellens einer Tabelle

1. `<table> .. </table>`-Tags als Begrenzer der Tabelle erstellen:

```
<table>

</table>
```

2. Erstellen der Tabellen-Zeilen durch die Tags `<tr> .. </tr>`

```
<table>
  <tr>
  </tr>
  <tr>
  </tr>
</table>
```

Diese Tabelle erhält zwei Zeilen. Später können weitere Zeilen hinzugefügt werden.

3. Erstellen der Tabellen-Zellen `<td> .. </td>` in den Zeilen

```
<table>
  <tr>
    <td>(1.1)</td>
    <td>(1.2)</td>
  </tr>
  <tr>
    <td>(2.1)</td>
    <td>(2.2)</td>
  </tr>
</table>
```

Beachten Sie, dass Sie als Programmierer dafür verantwortlich sind, dass in jeder Tabellenzeile die gleiche Anzahl Zellen sind.

Die Tabelle erzeugt die folgende Browseransicht:

(1.1) (1.2)

(2.1) (2.2)

Es ist noch nicht viel zu erkennen, da noch keine Rahmenlinien angezeigt werden und die Tabelle nur so breit ist wie der Inhalt Platz benötigt.

Zu erkennen ist aber, dass die in die Zellen eingetragenen Werte der ersten Zeile in der oberen Reihe nebeneinander stehen und die der unteren Zeile darunter.

Hinweis

Tabellenzellen werden in den Browsern i. allg. nur angezeigt, wenn sie nicht leer sind. Ein Leerzeichen reicht aber nicht aus. Falls eine leere Tabellenzelle erzeugt werden soll, muss sie den Wert ` ` erhalten.

Aufgabe

Erstellen Sie eine Tabelle nach diesem Muster, z.B. mit zwei Zeilen und drei Spalten. Geben Sie in jede Zelle einen Wert ein, an dem Sie die Zeile und die Spalte erkennen.

(5.2) Tabelle mit Angabe der Tabellenbreite und mit Rahmenlinie

Um die Breite einer Tabelle festzulegen und zur Anzeige der Rahmenlinien werden Attribute für den `<table>`-Tag benötigt.

<code>border="1"</code>	Stärke der Rahmenlinie: 1 Pixel
<code>width="200"</code>	Breite der Tabelle: 200 Pixel

Die in (5.1) erstellte Tabelle mit vorgegebener Breite und mit Rahmenlinien sieht wie folgt aus:

1.1	1.2
2.1	2.2

```
<table border="1" width="200">
  <tr>
    <td>1.1</td>
    <td>1.2</td>
  </tr>
  <tr>
    <td>2.1</td>
    <td>2.2</td>
  </tr>
</table>
```

(5.3) Übung

Diese Übung zielt darauf ab, dass Sie die Zuordnung der Tabellenzellen im HTML-Code zur Browseransicht beherrschen.

Aufgabe

Erstellen Sie eine Tabelle mit fünf Zeilen und drei Spalten. Legen Sie die Tabellenbreite und die Rahmenstärke fest.

Orientieren Sie sich an dem folgenden Muster.

1a	1b	1c
2a	2b	2c
3a	3b	3c
4a	4b	4c
5a	5b	5c

=> Lösung

```
<table border="1" width="300">
<tr>
<td>1a</td>
<td>1b</td>
<td>1c</td>
</tr>
<tr>
<td>2a</td>
<td>2b</td>
<td>2c</td>
</tr>
<tr>
<td>3a</td>
<td>3b</td>
<td>3c</td>
</tr>
<tr>
<td>4a</td>
<td>4b</td>
<td>4c</td>
</tr>
<tr>
<td>5a</td>
<td>5b</td>
<td>5c</td>
</tr>
</table>
```

1.5.3 Ausgewählte Attribute für Tabellen

In diesem Abschnitt werden nur ausgewählte Tabellen-Eigenschaften vorgestellt. Vertiefende Informationen erhalten Sie in: [SelfHTML -> Tabellen -> Gestaltung](#) HTML kennt Attribute für den `<table>`- und für den `<td>`-Tag.

(5.4) Formate für den `<table>`-Tag

Die folgende Tabelle soll erstellt werden.

Der dafür verwendete HTML-Code sieht wie folgt aus:

```
<table
  border="1"
  cellpadding="3"
  cellspacing="0"
  style="border-
collapse: collapse"
  width="200"
  bordercolor="#808080"
  bgcolor="#E0E0EF"
  height="60"
>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
</table>
```

Stärke der Rahmenlinie: 1 px

Abstand Text-Zellrahmen: 3 px

Abstand zw. Zellrahmen: 0 px

CSS-Style: nur eine Rahmenlinie

Minimale Breite der Tabelle: 200 px

Farbe der Rahmenlinie: #808080

Hintergrundfarbe: #E0E0EF

Minimale Höhe der Tabelle: 60px

Nutzen Sie bitte [SelfHTML -> Tabellen -> Gestaltung](#) für weitere Informationen zu den vorgestellten `<table>`-Attributen. Klären Sie insbesondere die möglichen Werte für die einzelnen Attribute.

Aufgabe

Erstellen Sie diese Tabelle in einer HTML-Datei und experimentieren Sie mit den Eigenschaften.

(5.5) Formate für den <td>-Tag

Hier wird eine Tabelle dargestellt, die nur eine Zelle hat. Solche Tabellen können verwendet werden, wenn ein Rahmen benötigt wird.

oben / zentriert

HTML-Coder der Tabelle mit einer Zelle

```
<table border="1"
cellpadding="3"
cellspacing="0"
style="border-collapse:
collapse"
bordercolor="#808080"
>
  <tr>  <td
nowrap
bgcolor="#FFFA9D"
width="250"
height="60"
align="center"
valign="top"
  >oben zentriert</td>
</tr>
</table>
```

Zeilenumbruch in der Zelle
 verhindern
 Hintergrundfarbe der Zelle
 Breite der Zelle (in Pixel)
 Höhe der Zelle (in Pixel)
 Horizontale Ausrichtung
 Vertikale Ausrichtung

Nutzen Sie auch hier wieder [SelfHTML -> Tabellen -> Gestaltung](#) für weitere Informationen zu den vorgestellten <td>-Attributen. Klären Sie insbesondere die möglichen Werte für die einzelnen Attribute.

Aufgabe

Erstellen Sie diese Tabelle in einer HTML-Datei und experimentieren Sie auch hier wieder mit den Eigenschaften.

1.5.4 Zellen verbinden

Wenn Sie bereits mit Tabellen in einem Textverarbeitungs- oder einem Tabellenkalkulationsprogramm, dann kennen sie auch die Möglichkeit, Zellen zu verbinden. Für HTML-Tabellen besteht diese Möglichkeit auch.

Vertiefende Informationen: [SelfHTML -> Tabellen -> Zellen verbinden](#)

(5.6) Einführungsbeispiel - Zellen einer Zeile verbinden

Gegeben ist die folgende Tabelle:

1	2
3	4

```
<table border="1" cellpadding="3"
cellspacing="0" style="border-
collapse: collapse" width="200"
bordercolor="#808080">
<tr>
  <td>1</td>
  <td>2</td>
</tr>
<tr>
  <td>3</td>
  <td>4</td>
</tr>
</table>
```

Die Zellen der oberen Zeile sollen verbunden werden. Sie wurden im Quellcode rot markiert.

Dazu wird in dem 1. `<td>`-Tag der zu verbindenden Zellen das Attribut `colspan="2"` eingetragen. In einer Zeile befindliche Zellen werden über Spalten (col = column, Spalte) 'gespannt'. Als wert erhält dieses Attribut die Anzahl der Spalten (Zellen in einer Zeile), die verbunden werden sollen.

Neben dem Eintragen des `colspan`-Attributs müssen außer der ersten Zelle alle weiteren Zellen, deren Platz durch die erste Zelle ausgefüllt werden soll, im HTML-Code gelöscht werden. Das ist hier die Zelle, die die Zahl '2' enthalten hat.

1		
3		4

```

<table border="1" cellpadding="3"
cellspacing="0" style="border-
collapse: collapse" width="200"
bordercolor="#808080">
<tr>
  <td colspan="2">
</td>
</tr>
<tr>
  <td>3</td>
  <td>4</td>
</tr>
</table>

```

Aufgabe

Testen Sie dieses Beispiel. Erstellen Sie sich dann eine größere Tabelle mit z.B. 5 Zellen in einer Zeile und mindestens 2 Spalten und verbinden Sie 2. bis 4. Zelle der 2. Zeile.

(5.7) Übung - Zellen einer Spalte verbinden

Gegeben ist die folgende Tabelle:

1	2
3	4

```

<table border="1" cellpadding="3"
cellspacing="0" style="border-
collapse: collapse" width="200"
bordercolor="#808080">
  <tr>
    <td>1</td>
    <td>2</td>
  </tr>
  <tr>
    <td>3</td>
    <td>4</td>
  </tr>
</table>

```

Die Zellen der rechten Spalte sollen verbunden werden. Das sind die Zellen mit den Werten 2 und 4.

Wenn Zellen einer Spalte verbunden werden, dann werden Zellen über mehrere Zeilen 'gespannt'. Das zugehörige HTML-Attribut für den `<td>`-Tag heißt `rowspan`. Dieses Attribut wird wieder in den Tag der erste (oberste) Zelle der Zellen, die verbunden werden sollen, in der Form `rowspan="2"` eingetragen. Alle weiteren Zellen, die zu diesem Verbund gehören, müssen gelöscht werden.

Aufgabe

Übertragen sie die vorgegebene Tabelle in eine HTML-Seite und erzeugen Sie den Verbund.

Ansicht der gewünschten Tabelle

=> Lösung

1	2
3	

```

<table border="1" cellpadding="3"
cellspacing="0"
style="border-collapse: collapse"
width="200"
bordercolor="#808080">
<tr>
<td>1</td>
<td rowspan="2">2</td>
</tr>
<tr>
<td>3</td>
</tr>
</table>

```

(5.8) Übung: Zellen einer Zeile und einer Spalte verbinden

In HTML besteht auch die Möglichkeit, Zeilen und Spalten zu einem Verbund zusammenzufassen. In dem Fall wird im ersten `<td>`-Tag der Tabelle sowohl das `colspan`- als auch das `rowspan`-Attribut angegeben werden.

Aufgabe

Erstellen Sie die folgende Tabelle mit 3 Spalten und 4 Zeilen. Verbinden Sie dann die Zellen so wie in der unten abgebildeten Tabelle dargestellt.

=> Lösung

```
<table border="1" cellpadding="3" cellspacing="0" style="border-collapse: collapse" width="300" bordercolor="#808080">
<tr>
  <td colspan="2" rowspan="3">&nbsp;</td>

  <td>&nbsp;</td>
</tr>
<tr>

  <td>&nbsp;</td>
</tr>
<tr>

  <td>&nbsp;</td>
</tr>
<tr>

  <td>&nbsp;</td>
</tr>
<tr>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
</tr>
</table>
```

1 Erstellen einer HTML-Tabelle

1.1	1.2
2.1	2.2

```
<table border="1" width="200">
  <tr>
    <td>1.1</td>
    <td>1.2</td>
  </tr>
  <tr>
    <td>2.1</td>
    <td>2.2</td>
  </tr>
</table>
```

2 Formatieren der Tabelle mit HTML-Attributen

<table>-Tag-Attribute

<pre> <table Stärke der Rahmenlinie: 1 px border="1" Abstand Text-Zellrahmen: 3 px cellpadding="3" Abstand zw. Zellrahmen: 0 px cellspacing="0" CSS-Style: nur eine Rahmenlinie style="border-collapse: Minimale Breite der Tabelle: 200 collapse" width="200" Farbe der Rahmenlinie: #808080 bordercolor="#808080" Hintergrundfarbe: #E0E0EF bgcolor="#E0E0EF" height="60" > <tr> ... </tr> </table> </pre>	<p>Minimale Höhe der Tabelle: 60px</p> 
---	---

<td>-Tag-Attribute

<pre> <table border="1" cellpadding="3" cellspacing="0" style="border-collapse: collapse" bordercolor="#808080" > <tr> <td nowrap bgcolor="#FFFA9D" width="250" height="60" </pre>	<p>oben / zentriert</p> <p>Zeilenumbruch in der Zelle verhindern Hintergrundfarbe der Zelle Breite der Zelle (in Pixel) Höhe der Zelle (in Pixel) Horizontale Ausrichtung Vertikale Ausrichtung</p>
<pre> align="center" valign="top" >oben zentriert</td> </tr> </table> </pre>	

3 Zellen einer Tabelle verbinden

Zellen mehrerer Spalten verbinden (**colspan**)

1	
3	4

```

<table border="1" cellpadding="3"
cellspacing="0" style="border-
collapse: collapse" width="200"
bordercolor="#808080">
<tr>
<td colspan="2">1</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
</table>

```

Zellen mehrerer Zeilen verbinden (**rowspan**)

1	2
3	

```

<table border="1" cellpadding="3"
cellspacing="0" style="border-
collapse: collapse" width="200"
bordercolor="#808080">
<tr>
<td>1</td>
<td rowspan="2">2</td>
</tr>
<tr>
<td>3</td>
</tr>
</table>

```

1.5.6 Übungen

(5.9) Übung

Erstellen Sie eine Tabelle in HTML. Orientieren Sie sich bitte an der folgenden Tabelle:

1.1		1.3	1.4
		2.3	2.4
3.1	3.2	3.3	
4.1			

Die Farben für den Hintergrund der Tabelle und der Zellen sowie für die Rahmenlinien können Sie selbst wählen.

=> Lösung

```
<table border="1" cellpadding="3" cellspacing="3"
width="400" bordercolor="#800000" bgcolor="#FFFFCC">
<tr>
  <td colspan="2" rowspan="2" bgcolor="#E0E0EF">1.1</td>
  <td>1.3</td>
  <td>1.4</td>
</tr>
<tr>

  <td>2.3</td>
  <td rowspan="3">2.4</td>
</tr>
<tr>
  <td>3.1</td>
  <td>3.2</td>
  <td>3.3</td>
</tr>
<tr>
  <td colspan="3">4.1</td>

</tr>
</table>
```

1.6 Bilder

Stundenempfehlung

FIAE	FISI	ITSK	IFK	ITSE
4	4	4	4	4

[1.6.1 Überblick über dieses Kapitel](#)

[1.6.2 Einführung in das Thema "Bilder auf Webseiten"](#)

[1.6.3 Bilder in HTML-Seiten einbinden](#)

[1.6.4 HTML-Attribute für Bilder](#)

[1.6.5 Bilder als Hyperlinks](#)

[1.6.6 Kapitelschwerpunkte](#)

[1.6.7 Übungen](#)

1.6.1 Überblick über dieses Kapitel

Dieses Kapitel behandelt Themen rund um das Thema "Bilder in HTML-Seiten".

Ein Problem ist die Bereitstellung rechtefreier Bilder. Dazu enthält dieses Kapitel einen kurzen Abschnitt über elementare Bildbearbeitungsfunktionen.

Dann wird gezeigt, wie Bilddateien in HTML-Seiten eingefügt und formatiert werden können. Danach wird gezeigt, wie Bilder für Hyperlinks genutzt werden können.

Abgeschlossen wird dieses Kapitel durch eine kleine Übung.

1.6.2 Einführung in das Thema "Bilder auf Webseiten"

Prinzip der Einbindung eines Bildes in eine HTML-Datei

Das Grundprinzip bei der Einbindung eines Bildes in eine HTML-Seite besteht darin, dass das Bild als Datei vorliegen muss. Dann kann die Datei über eine Bild-Referenz (ein ``-Tag) in die Datei eingefügt werden.

Bilddateien für HTML-Seiten

Da die Dateien der Bilder vom Webserver zum Webbrowser übertragen werden, sind die Webentwickler bestrebt, die Dateigrößen so klein wie möglich zu machen. Dazu werden zwei Maßnahmen eingesetzt:

□ **Komprimierte Dateiformate**

Für Bilder im Web werden vorwiegend die folgenden 3 Dateiformate eingesetzt:

- JPEG
- GIF
- PNG

Jedes dieser Formate hat Vor- und Nachteile. Für Bilder wird meistens das JPEGFormat verwendet. Für Grafiken mit wenigen Farben wird das GIF-Format verwendet. Das PNG-Format kann für beide Anwendungen verwendet werden, erzeugt aber oft die größte Datei.







□ **Optimierung der Pixelzahl**

Die Optimierung der Pixelzahl bedeutet, dass die Anzahl der Pixel genau so festgelegt wird, dass ein Pixel des Bildes auf genau einem Pixel des Bildschirms abgebildet wird. Dadurch ist eine Optimale Darstellung gesichert in Verbindung mit einer minimierten Pixel- und dadurch Dateigröße.

Die Pixelzahl kann z.B. auch mit dem Windows-Programm `Paint` vermindert werden.

Musterbilder

In der folgenden Tabelle zwei Bilder in den drei oben angegebenen Formaten abgebildet. Für die ersten Übungen mit Bildern können diese Bilder verwenden. Sie können aber auch eigene Bilder verwenden.

JPEG	bild1.jpg 	bild2.jpg 
GIF	bild1.gif 	bild2.gif 
PNG	bild1.png 	bild2.gif 

Das folgende Bild habe ich mit einer Digitalkamera aufgenommen. Es hat die in der Breite 3668 und in der Höhe 2736 Pixel. Die Dateigröße beträgt ca. 4 MByte.



Bild 5.1: blume.jpg

In dieser Größe ist es nicht für den Einsatz auf einer Webseite geeignet.

(6.1) Bilder für Webseiten bearbeiten

Die Bildbearbeitung ist ein umfangreiches und eigenständiges Thema, deren Behandlung den Rahmen eines HTML-Kurses sprengen würde. Da aber für die Nutzung von Copyright-freien (eigenen) Bildern elementare Grundkenntnisse benötigt werden, bitte ich Sie, sich mit den Bildbearbeitungsfunktionen

- Ausschneiden eines Teilbildes
- Anpassen der Größe
- Speichern in einem Format, dass für Webseiten geeignet ist zu beschäftigen.

Auswahl des Bildbearbeitungsprogramms

Um Bilder für Webseiten zu bearbeiten, gibt es eine Vielzahl von Bildbearbeitungsprogrammen.

Windows stellt in der Zubehör-Gruppe das sehr einfache Bildbearbeitungsprogramm **Paint** bereit. Dieses Programm stellt die genannten Funktionen bereit und reicht für die ersten Schritte aus.

Weiterhin gibt es kostenfrei das Open-Source-Programm **Gimp**.

Sprechen Sie mit Ihrem Lernbegleiter, welches Programm er Ihnen empfiehlt bzw. bereitstellen kann.



Aufgabe


Wählen Sie ein Bildbearbeitungsprogramm aus. Laden Sie das Bild "blume.jpg" runter und öffnen Sie es in dem Bildbearbeitungsprogramm.

Verändern Sie die Pixelgröße z.B. auf die Breite von 400 Pixel und die Höhe von 300 Pixel. Das Verhältnis von Höhe zu breite soll erhalten bleiben.

Speichern Sie das Bild als JPEG, als GIF und als PNG. Vergleichen Sie danach die Qualität der Bilder und die Größe der Bilddatei.

=> Lösung


Forma t		Bemerkungen
JPG		<p>Dateigröße: 12 kB</p> <p>Die Farben sind erhalten geblieben, die Auflösung hat sich verschlechtert.</p> <p>Bei JPEG-Bildern kann die Komprimierung eingestellt werden. Dieses Bild wurde zu stark komprimiert, so dass die Verschlechterung der Auflösung bereits deutlich zu erkennen ist.</p>
GIF		<p>Dateigröße: 71 kB</p> <p>Die Auflösung ist erhalten geblieben. Die Farben wurden auf 256 reduziert.</p> <p>Suchen Sie nach den Verschlechterungen. Sie befinden sich an den Blütenblättern und an den grünen Blättern.</p>

PNG		Dateigröße: 227 kB Die Auflösung und die Anzahl der Farben sind erhalten geblieben.
-----	--	--

1.6.3 Bilder in HTML-Seiten einbinden

(6.2) Der -Tag

Bilder werden in HTML-Dateien mit Hilfe des ``-Tags eingefügt.

	<pre></pre>
---	---

`img`: Image, Bild `src`:
Source, Quelle, `alt`:
Alternativer Text

Vertiefende Informationen: [SelfHTML -> Grafiken -> einbinden](#)

Aufgabe

Fügen Sie Bilder in eine HTML-Seite einschließlich der Alt-Attribute ein.

Rufen Sie die HTML-Seite im Browser auf und kontrollieren Sie die alt-Texte.

1.6.4 HTML-Attribute für Bilder

*Der ****-Tag benötigt immer mindestens das **src-Attribut**. Empfohlen wird weiterhin das **alt-Attribut**.* Daneben gibt es weitere Attribute, die angegeben werden können bzw. auch angegeben werden sollten.

(6.3) HTML-Attribute für -Tags

Beantworten Sie mit Hilfe der oben angegebenen SelfHTML-Seite [SelfHTML -> Grafiken -> einbinden](#) die folgenden Fragen.

Breite und Höhe

1. Warum sollte die Breite und die Höhe bei -Tags immer mit angegeben werden?
2. Wie kann die Höhe und die Breite einer Bild-Datei ermittelt werden?
- 3 Geben Sie einen Beispiel für einen Tag mit Breiten- und Höhen-Angabe an.
4. Können für die Breite und die Höhe andere Werte angegeben werden?

Rahmen um das Bild

- 5 Wie kann festgelegt werden, dass ein Rahmen um ein Bild angezeigt werden soll?

=> Lösung

Breite und Höhe

1. Warum sollte die Breite und die Höhe bei -Tags immer mit angegeben werden?

Der Browser liest zuerst die HTML-Datei ein und erst danach die Bilddateien.

Durch die Angabe der Breite und der Höhe der Bilder kennt er bereits beim Einlesen der HTML-Datei den Platz für die Größe des Bildes und kann den erforderlichen Platz reservieren. Dadurch werden Verzögerungen beim Seitenaufbau und Korrekturen des Seitenaufbaus beim Laden der Bilder vermieden.

2. Wie kann die Höhe und die Breite einer Bild-Datei ermittelt werden?

- Durch das Kontextmenü Menü des Datei- oder Webbrowsers.
- Durch ein Bildbearbeitungsprogramm
- Mit Hilfe des HTML-Editors.

Wenn Sie Scriptly verwenden und die Datei in den gewünschten Ordner der Website kopiert wurde, dann können Sie die Bilddatei direkt aus den Ordner in HTML-Seite ziehen. In dem Fall erhält das Bild automatisch die beiden Attribute für die Breite und die Höhe.

3 Geben Sie einen Beispiel für einen Tag mit Breiten- und Höhen-Angabe an.

```

```

4. Können für die Breite und die Höhe andere Werte angegeben werden?

Ja. Allerdings wird dabei nicht die Grafikdatei geändert und das Bild könnte durch nicht proportionale Änderungen verzerrt werden.

Rahmen um das Bild

5 Wie kann festgelegt werden, dass ein Rahmen um ein Bild angezeigt werden soll?

Mit Hilfe des `border`-Attributs. Die Rahmenstärke wird in Pixel angegeben.

1.6.5 Bilder als Hyperlinks

Im Kapitel "1.3 HTML-Seiten verlinken" wurde der `<a>`-Tag zur Erzeugung von Hyperlinks verwendet, deren Träger Texte waren. Es besteht auch die Möglichkeit, anstelle des Textes eine Grafik zwischen das `<a>`-Tag-Paar einzufügen. Dadurch wird der Träger des Hyperlinks ein Bild.

(6.4) Hyperlinks mit Bildern

Beantworten Sie mit Hilfe von [SelfHTML -> Grafiken -> Verweise](#) die folgenden Fragen.

1. Geben Sie den HTML-Code an, wie ein Bild als Träger eines Hyperlinks eingesetzt werden kann.

2. Warum sollten Bilder, die als Hyperlinks genutzt werden, keinen Rahmen erhalten?

Der Rahmen erhält die Hyperlink-Farbe, i. allg. blau. Dadurch wird das Design der Webseite negativ beeinflusst.

=> Lösung**1. Geben Sie den HTML-Code an, wie ein Bild als Träger eines Hyperlinks eingesetzt werden kann.**

```
<p><a href="bilddateien.zip">  
  </a></p>
```

2. Warum sollten Bilder, die als Hyperlinks genutzt werden, keinen Rahmen erhalten?

Der Rahmen erhält die Hyperlink-Farbe, i. allg. blau. Dadurch wird das Design der Webseite negativ beeinflusst. Dazu sollte `border="0"` in den ``-Tag eingetragen werden.



```
<p><a href="bilddateien.zip">  
  </a></p>
```


1.6.6 Kapitelschwerpunkte

Technische Anforderungen an Bild-Dateien für Webseiten

Komprimierte Dateiformate

Für Bilder im Web werden vorwiegend die folgenden 3 Dateiformate eingesetzt:

- JPEG
- GIF
- PNG


Jedes dieser Formate hat Vor- und Nachteile. Für Bilder wird meistens das JPEG-Format verwendet. Für Grafiken mit wenigen Farben wird das GIF-Format verwendet. Das PNGFormat kann für beide Anwendungen verwendet werden, erzeugt aber oft die größte Datei.

Optimierung der Pixelzahl

Die Optimierung der Pixelzahl bedeutet, dass die Anzahl der Pixel genau so festgelegt wird, dass ein Pixel des Bildes auf genau einem Pixel des Bildschirms abgebildet wird. Dadurch ist eine Optimale Darstellung gesichert in Verbindung mit einer minimierten Pixel- und dadurch Dateigröße.

Die Pixelzahl kann z.B. auch mit dem Windows-Programm [Paint](#) vermindert werden.

Bilder in HTML-Seiten einbinden

	<pre></pre>
---	---

Weitere Attribute für -Tags

```

```

Hyperlinks mit Bildern

```
<p><a href="bilddateien.zip">
  </a></p>
```

1.6.7 Übungen

(6.5) Über Thumbnails Bilder anzeigen

In dieser Übung sollen Sie eine kleine Webseite mit vier Seiten erstellen. Ich habe Ihnen Bilder vorbereitet, sie können aber auch eigene Bilder verwenden. Es sollen Thumbnails (kleine Vorschaubilder) angezeigt werden. Bei Anklicken eines der Thumbnails soll darunter ein vergrößertes Bild angezeigt werden.

Die vier Seiten sollen das folgende Aussehen haben:

Startseite

Home



Angezeigte Seite nach Anklicken des linken der drei kleinen Bilder

Startseite



Das angeklickte Bild soll dann unter den drei kleinen Bildern vergrößert angezeigt werden.

Angezeigte Seite nach Anklicken des mittleren der drei kleinen Bilder

[Startseite](#)



Angezeigte Seite nach Anklicken des rechten der drei kleinen Bilder

[Startseite](#)



Die Bilder können Sie sich hier runterladen:

- [Originalbilder](#)
- http://www.daa-mws-virtuell.de/content/7/html3/it/it06_2011/IT.VI.a/01/original_bilder.zip
- [Vorbereitete Bilder](#)

Aufgabe

Legen Sie die Größe der Thumbnails und der vergrößerten Bilder fest und bereiten Sie die Bilder vor.

Erstellen Sie einen Seitenbaum und danach die HTML-Seiten. Testen Sie die erstellten Seiten.

Musterlösung: 06_bilder/bildergalerie/bilder1.html

1.7 Übung

Stundenempfehlung

FIAE	FISI	ITSK	IFK	ITSE
5	5	5	5	5

1.7.1 Überblick über dieses Kapitel

1.7.2 Inhalt der Webseiten

1.7.1 Überblick über dieses Kapitel

In diesem Kapitel haben Sie die Möglichkeit, selbst kreativ zu werden und HTMLDateien zu erstellen.

Wichtig ist dabei, dass Sie möglichst viele der behandelten HTML-Themen in Ihre Webseiten aufnehmen.

1.7.2 Inhalt der Webseiten

Was Sie auf Ihren HTML-Seiten behandeln wollen, können Sie selbst auswählen.

Ich möchte Ihnen aber eine Vorstellung geben, was hier erwartet wird. Das folgende Bild habe ich dem Webshop

"http://www.mindfactory.de/product_info.php/info/p619005_1000GB-Samsunghttp://www.mindfactory.de/product_info.php/info/p619005_1000GB-Samsung-Spinpoint-F3-HD103SJ-32MB-3-5Zoll--8-9cm--SATA-3Gb-s.htmlSpinpoint-F3-HD103SJ-32MB-3-5Zoll--8-9cm--SATA-3Gb-s.html" entnommen.

1000GB Samsung Spinpoint F3 HD103SJ 32MB 3.5" (8.9cm) SATA 3Gb/s

Lagernd | > 5 St.



€ 40,^{76*}
inkl. 19% UST
zzgl. Versandkosten



€ 2,49*

Abbildungen ähnlich. Änderungen und Irrtümer vorbehalten.

Bewertungen: ★★★★★

(224 Meinungen)

PRODUKTBESCHREIBUNG

Artikelnummer: 39133

Seit dem 31.03.2011 im Sortiment

Die schnelle Festplatte von Samsung ist mit einer großen Speicherkapazität, einem riesigen Cache und der schnellen SATA-Schnittstelle ausgerüstet. Dadurch sorgt sie für eine hohe Performance und schnellen Zugriffszeiten - ideal geeignet für Menschen mit hohem Speicherplatzbedarf und Ansprüchen in Geschwindigkeit und Qualität.

TECHN. DATEN

ZUBEHÖR

BEWERTUNGEN

HINWEIS

Für die Richtigkeit und Vollständigkeit der hier aufgeführten Daten wird keine Haftung übernommen.



ALLGEMEIN:

Kapazität:	1000GB
Modelserie:	Spinpoint F3
Umdrehungsgeschwindigkeit:	7.200U/min
Mittlere Zugriffszeit:	8,9ms
Cache:	32MB
Formfaktor:	3.5" (8.9cm)
Schnittstelle:	SATA 3Gb/s
Stromanschluss:	SATA 15-pol

Bild 7.1: Vorlage für eine Webseite

Hinweise

- Realisieren Sie nur solche Inhalte, die nicht auf andere Seiten führen. D.h. z. B. der Hyperlink für den Warenkorb soll weggelassen werden.
- Die Rahmen mit den abgerundeten Ecken können Sie weglassen.
- Um im oberen Bereich links und rechts Inhalte zu platzieren, können Sie eine unsichtbare Tabelle (border="0") verwenden.
Das Produktbild wird auf der Original-Seite bei Anklicken in einem neuen Fenster vergrößert geöffnet.
Die Stern-Reihen für die Bewertung können Sie für Ausbildungszwecke von der Seite als Bildschirmbild abnehmen.
- Die Zeile "Technische Daten / Zubehör / Bewertungen" erfordert, dass der obere Teil dieser Seite dreimal angelegt werden muss und die drei Auswahlmöglichkeiten drei Hyperlinks erfordern.
Zur Formatierung der technischen Daten soll eine Tabelle verwendet werden.
- Die HTML-Seiten sollen noch nicht formatiert werden.

Aufgabe

Suchen Sie sich eine Webseite im Internet, z.B. eine Seite eines Webshops und bauen Sie einen Teil der Seite nach.

Hinweis

Zu dieser Aufgabe gibt es keine Musterlösung.