

Abstracts *HTML - die Sprache des WWW*

HTML ist die Sprache zum Erstellen von Web-Dokumenten.
HTML (Hyper Text Markup Language) ist eine Auszeichnungssprache im Textformat und damit plattformunabhängig.
HTML verwendet **Marken (Tags)**, die in spitzen Klammern dargestellt werden.

HTML ermöglicht:

- o Darstellung und Formatierung von Text
- o Darstellung von Grafik
- o Verweise auf andere Seiten (Links)
- o Formular Eingaben

empfehlenswerte Online-Dokumente :

Selfhtml : sehr guter, ausführlicher Online-Kurs (<http://selfhtml.teamone.de/>)

W3C-Recommendations : HTML-Spezifikationen (<http://www.w3.org/>)

HTML-Versionen :

HTML 1.0 erste Sprachversion

HTML 2.0 ab 1995

HTML 3.2 ab 1997 , Tabellen, diverse Befehle zur physischen Textauszeichnung.

HTML 4.0 ab 1998 , Frames, CSS und Scriptsprachen.

XML-Konformität : XHTML-Dokumente erfordern zusätzliche Regeln

XHTML 1.0 ab 2000

XHTML 1.1 ab 2001

HTML - Editoren :

HTML Dokumente lassen sich mit jedem Text Editor (TextPad) erstellen, effizienter jedoch mit speziellen HTML-Editoren.

Mit textbasierenden HTML-Editoren arbeitet man direkt mit HTML-Befehlen im HTML-Quelltext . Tags können über Symbolleisten und Menüpunkte eingefügt werden.
WYSIWYG (= What You See Is What You Get) -Editoren zeigen die HTML-Seite bei der Erstellung bereits so an, wie sie (in etwa!) ein WWW-Browser ausgibt.

einige bekannte HTML-Editoren :

Dreamweaver, Homesite, Golive, HotMetall, FrontPage,
Mozilla- u. Netscape Composer, FrontPage Express,
Textverarbeitungsprogramme wie StarOffice, OpenOffice

HTML - Dateinamen :

Dateinamen : Unix-kompatibel ausführen !
 Unterscheidung von Groß- und Kleinschreibung, keine Sonderzeichen !
Dateiendungen : HTML-Dateien **".html "** oder **".htm "**
 mit Server Side Includes **".shtml "** oder **".shtm "**
 mit PHP-Code **".php"**

Default-Dateiname für die Startseite : "index.html" oder "index.htm"

Allgemeine Regeln für HTML :

Marke (Tag) allgemein : < **Markenname** [**Attribute**] >

HTML-Grundgerüst :

```
<HTML>      <!-- HTML - Marke -->
<HEAD>      <!-- Kopf - Marke -->
            <!-- Kopfteil für Verwaltungsinformation -->
</HEAD>     <!-- Kopf - Endemarke -->
<BODY>      <!-- Körper - Marke -->
            <!-- Körperteil für Dokument Inhalt -->
</BODY>     <!-- Körper - Endemarke -->
</HTML>     <!-- HTML - Endemarke -->
```

HTML-Marken können in Klein- oder Großbuchstaben ausgeführt werden.

XHTML hingegen unterscheidet zwischen Groß- und Kleinschreibung!

In XHTML werden alle Marken-, Attributnamen und Werte klein geschrieben.

Viele Marken bestehen aus einleitender und schließender Marke ,

einige wenige sind "Standalone-Tags" ohne schließender Marke.

Schließende Marken werden mit einem Schrägstrich "/" gekennzeichnet.

Marken können auch ineinander verschachtelt werden.

Tags können mit bestimmten **Attributen** für zusätzliche Angaben erweitert werden.

Beispiel : seite1.html - Quelltext

```
<HTML>                                     <!-- Kommentar -->

<HEAD>
  <TITLE> Seite 1 </TITLE>                 <!-- Titel der Seite -->
</HEAD>

<BODY bgcolor=#FFDD99 text=#442233>

<H1 align="center">HTML-Dokument</H1><!-- 1.Überschrift -->
<H2> Einführung </H2>                   <!-- 2.Überschrift -->

<P>                                     <!-- Absatz -->
  ein HTML Dokument besteht aus Elementen für Abs&auml;tze,
  &Uuml;berschriften, Listen, Verweise auf andere Dokumente
  oder eingebundene Bilder usw.
</P>

<BR>                                     <!-- Zeilenumbruch -->

<A HREF="seite2.html"> 2.Seite </A>      <!-- Link -->

</BODY>
</HTML>
```

Attribute in HTML- Marken :

- alleinstehende Attribute , z.B. <hr noshade>
- Attribute mit Wertzuweisung, z.B.<h1 align=center>
- Attribute mit Namen, z.B.

Streng SGML-konform müssten Attributwerte immer in Anführungszeichen stehen, in HTML können die Anführungszeichen bei vordefinierten Werten entfallen.

Dokumentweite Einstellungen :

Farbeinstellungen für Hintergrund ,Text und Verweise über Body-Attribut :

<BODY ... >	
Attribute	BGCOLOR="Farbe" Hintergrundfarbe
	BACKGROUND="Datei" Hintergrundbild
	TEXT="Farbe" Schriftfarbe
	LINK="Farbe" für nicht ausgewählte Links
	ALINK="Farbe" für aktiven Link
	VLINK="Farbe" für bereits ausgewählte Links

Beispiel: <body bgcolor=#442233 text=#FFDD99 link=#FF9944 >

Farbwerte :

Farben können entweder über den Farbwert in Hexadezimalform #RRGGBB für Rot-,Grün-,Blau-Anteil mit den hexadezimalen Werten von jeweils 00 bis FF, oder über Farbnamen (140 Farben) festgelegt werden, wie white,silver,cyan,blue,gray,black,...,darkgray,lightyellow,...,seagreen,aquamarine,indigo, ...

Hintergrundbild :

Ein Hintergrundbild wird als Grafikdatei (GIF oder JPEG) eingebunden.
Das Bild wird dabei über das ganze Fenster wiederholt dargestellt (Wallpaper).

Beispiel: <body background="wall.gif"

Seitenränder : mit CSS Stylesheet Angaben

Beispiel:
<head>
 <style type="text/css">
 body { margin-left:40px; margin-right:40px; margin-top:60px;
 margin-bottom:60px }
 </style>
</head>

Titel :

im Kopfteil <HEAD>
 <TITLE> Titel der HTML-Seite </TITLE>
 </HEAD>

Textformatierung :

Überschriften : Headline

<H1>...</H1> Überschrift 1.Ordnung (groß)

...
<H6>...</H6> Überschrift 6.Ordnung (klein)

<H1 align="center"> ... </H1> zentriert

Absatz :

<P>...</P> Paragraph = Absatz
stellt den Text innerhalb der Marken als eigenen Absatz dar.
(der Browser unterdrückt dabei Zeilenumbrüche und mehrfache Leerzeichen des Quelltextes !)
Ausrichtung über Attribut align="center" , "right", "justify" (Blocksatz), "left"

Zeilenumbruch :

 Break = Zeilenumbruch erzwingen
 <NOBR> No break = kein Umbruch
 </NOBR>
 <WBR> Word Break = Umbruch innerhalb eines Wortes

vorformatierter Text :

<PRE>...</PRE> alles wird so angezeigt, wie im Quelltext ,
 mit Leerzeichen und Zeilenumbrüchen.

Bsp. :

```
<PRE>
      Mails      1. Quartal   2. Quartal   3.Quartal   4. Quartal
                  343         251         624         159
</PRE>
```

Adressen : besondere Darstellung für Adressen, Autorennennungen, etc.

<ADDRESS>... </ADDRESS>

Zitate : Darstellung von längeren Zitaten

<BLOCKQUOTE> ...</BLOCKQUOTE>

Linien :

<HR> Horizontale Linie
 Attribute : ALIGN=[LEFT,CENTER,RIGHT]
 NOSHADE , SIZE=[Pixel] , WIDTH=[Pixel] oder [%]

Ersatzzeichen (Entities) :

HTML- verwendet den ASCII-Zeichensatz, für weitere Zeichen werden Entities verwendet.
 Ab HTML 4 kann auf den Latin-1-Zeichensatz (ISO 8859-1) umgestellt werden.

einige Ersatzzeichen :

 nonbreaking space = nicht umbrechbares Leerzeichen
 ä ä
 Ä Ä
 ö ö
 ...
 Ü ü
 ß ß
 < <
 > >
 & &
 " "
 € □ (Euro-Zeichen)

Zeichensatz Meta Angabe : für Latin-1 (ISO-8859-1)

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

Funktionale (logische) Textelemente :

logische Elemente beschreiben die Bedeutung, nicht die konkrete Darstellung.

<code>...</code>	emphased = betont (meist kursiv)
<code>...</code>	Strong = stark (fett)
<code><CITE>...</CITE></code>	Cite = Zitat (meist kursiv)
<code><CODE>...</CODE></code>	Programmquelltext darstellen
<code><SAMP>...</SAMP></code>	Beispiel (in nichtproportionaler Schrift)
<code><KBD> ...</KBD></code>	Darstellung als Texteingabe
<code><VAR> ...</VAR></code>	Darstellung als Variable

Typografische (physische) Textelemente :

physische Elemente beschreiben die konkrete Darstellung im Browser.

<code>...</code>	Bold	=	Fettdruck
<code><I>...</I></code>	Italic	=	Kursivschrift
<code><U>...</U></code>	Underline	=	unterstrichen
<code><BIG>...</BIG></code>	Big	=	größere Schrift
<code><SMALL>...</SMALL></code>	Small	=	kleinere Schrift
<code><SUB>...</SUB></code>	tiefgestellt		
<code><SUP>...</SUP></code>	hochgestellt		
<code>...</code>	Schrift (Größe, Farbe und Schriftart)		
	Attribute :		
	SIZE=[1..7] (Default = 3)		
	COLOR=[#000000 .. #FFFFFF] als RGB-Wert oder [Farbname]		
	FACE = Arial, Courier, Times Roman, ...		

Bsp .:

```
<font size=7> großer Text </font>
<font size=+1> Text um etwas größer </font>
<font color=#FFFFFF>Weißer Text</font>
<font size=5 color="green" face=Arial">Text</font>
```

Block-Element :

`<DIV> ... </DIV>` Bereich (i.a. für CSS-Formatierungen)

Bsp.:

```
<div align="center">
<h1> zentrierte Überschrift </h1>
...
</div>
```

Listenelemente :

punktierte Liste (UL = unordered list) : Aufzählungsliste

Bei einer punktierten Liste werden alle Listeneinträge mit einem Aufzählungszeichen (Bullet) versehen.

```
<UL>
  <LI> Text </LI>
  <LI> Text </LI> ...
</UL>
```

Mit beginnt jeweils ein neuer Punkt in der Liste (li = list item = Listeneintrag).

Das Ende-Tag am Ende eines Listeneintrags ist nicht zwingend erforderlich, im Sinne SGML-konformer Kodierung jedoch sauber.

Attribute :

```
<ul type=circle>    für rundes Bullet
<ul type=square>    rechteckiges Bullet
<ul type=disc>       Dateisymbol als Bullet
```

numerierte Liste (OL = ordered list) : Nummerierungen

bei einer numerierten Liste werden alle Listeneinträge automatisch durchnummeriert.

```
<OL>
  <LI> Text </LI>
  <LI> Text </LI>
  ...
</OL>
```

das Verschachteln bewirkt keine Numerierungshierarchien wie 1., 1.1., 1.1.1.,

Attribute :

```
<ol type=A> für A., B., C. ...           <ol type=a> für a., b., c. ...
<ol type=I> für I., II., III., IV. (römisch)  <ol type=i> für i., ii., iii., iv.
<ol start= i > mit Startwert i
<li value= i> Nummerierung mit i fortsetzen
```

Bsp. :

```
<ol start=10>
  <li> beginnt mit 10.
  <li> weiter mit 11.
  <li value=100> weiter mit 100.
  <li> weiter mit 101.
</ol>
```

Definitionsliste (DL = definition list) :

Definitionslisten bestehen aus Einträgen mit Stichworten und dazu beschreibenden Texten. Mit <DT> .. </DT> wird ein Stichwort (dt = definition term) definiert, und mit <DD> .. </DD> der zugehörige Text (dd = definition data).

```
<DL>
  <DT>Stichwort</DT> <DD> Text </DD>
  <DT>Stichwort</DT> <DD> Text </DD>
  ...
</DL>
```

Das Verschachteln von Definitionslisten ist möglich und ergibt baumartige Strukturen.

Verweis Elemente (Links) : *Verbindung zu anderen Seiten*

Verweise sind ein wesentlicher Teil eines Hypertext -Projekts zur Strukturierung der Dokumente. Die Einstiegsseite (home page) enthält Verweise zu weiteren Seiten und diese wiederum Rücksprungverweise, Verweise zu über- oder untergeordneten Seiten und auch Verweise zu anderen WWW-Adressen.

Der Verweistext soll aussagekräftig sein und eventuell mit Symbolen ergänzt werden.

Schema: ` Verweistext `

(a = anchor = Anker, href = hyper reference = Hypertext-Referenz).

Das **Verweisziel** kann sein :

intern : eine andere HTML- oder Grafikdatei am gleichen (Server) Rechner
oder eine Stelle (Anker) innerhalb der gleichen HTML-Datei

extern : eine WWW - , FTP-, E-Mail-, Newsgroup-, Gopher-, Telnet Adresse

interne Verweise :

zu Dateien im gleichen Verzeichnis :

```
<a href="datei.html"> Verweistext </a>
```

zu Dateien in anderen Verzeichnissen (relativ) :

```
<a href="verzeichnis/datei.html"> ...
```

```
<a href="../datei.html"> ein Verzeichnis höher </a>
```

```
<a href="../../grafik/bild.gif"> im Nebenverzeichnis </a>
```

Verweise innerhalb der eigenen HTML-Datei

Anker für Verweise definieren : ` Text `

Verweis zu Anker setzen : ` Text `

Verweis zu Anker innerhalb einer anderen Datei

```
<a href="datei.html#Ziel"> Text </a>
```

externe Verweise :

externe Verweise werden über URL-Angaben durchgeführt .

URL (Uniform Resource Locator) : im Internet eindeutiger Dokumentname

allgemeine Form : **protokoll://adresse/dateinamen**

HTTP URL : <http://www.htlstp.ac.at>

FTP URL : <ftp://ftp.tuwien.ac.at>

...

Datei URL : <file:///C:/htmldok/seite1.htm>

Beispiel: ` HTBLuVA St.Pölten `

Achtung : Verweise sollten niemals als absolute Dateinamen (wie C:\html\Seite1.html oder <file:///C:/html/Seite1.html>) ausgeführt werden, da sie dann am Server **nicht verwendbar sind !**

Verweis zu Mail-Adressen:

` E-Mail an Franz Hacker `

Grafiken in HTML-Dateien :

Inline Image : *Grafik innerhalb eines Dokumentes.*

(img = image = Bild, src = source = Quelle) angegeben.

Attribute SRC = "Grafikdatei" [*.GIF , *.JPG, ...]
ALT="alternativer Text" , "Ersatztext"

Beispiel: ` Text`

Der **alternative Text** wird dann angezeigt , wenn die Grafik aus irgendwelchen Gründen beim Anwender nicht dargestellt werden kann und bereits eingeblendet bevor die Grafik geladen ist. Das ALT -Attribut ist nach dem HTML Standard immer anzugeben.

Grafikbeschriftung :

Attribut : ALIGN=top der folgende Text wird oben bündig zur Grafik dargestellt
=middle mittig
=bottom untenbündig.

Rahmen um Grafiken :

Attribut BORDER = [Pixel]

Beispiel: ``

Breite und Höhe einer Grafik : Grafik skalieren

Durch Angabe der Breite und Höhe entnimmt der Browser bereits der HTML-Datei die Grafikgröße und nicht erst aus der Header-Information der Grafikdatei. Dadurch kann eine entsprechend große Freifläche vorgesehen und die restliche WWW-Seite vorab aufgebaut werden. Grafiken können dadurch auch verkleinert, vergrößert u. verzerrt dargestellt werden.

Attribut e width= [Pixel] für die Breite
height= [Pixel] für die Höhe

Beispiel: ``

Neben Pixelangaben für width und height sind auch Prozentangaben möglich .

Abstand zwischen Grafik und Text definieren :

mit hspace= [Pixel] wird der Abstand links,rechts zum Text festgelegt

mit vspace= [Pixel] wird der Abstand darüber,darunter festgelegt

Beispiel: `Text`

Grafiksymbol in einen Verweis einbinden (grafisches Bedienelement) :

Anstelle eines Verweistextes kann auch ein Grafiksymbol dargestellt werden .

Beispiel: ``

Verweis-sensitive Grafiken :

Verweis-sensitive Grafiken besitzen klickempfindliche Bereiche für Verweise (Bsp.: Landkarte)
Die Bereiche werden über eine Mappe mit Area Elementen definiert.

```
<MAP>
  < AREA ... >
  < AREA ... > ...
</MAP>
```

Die Area Elemente legen die klickempfindlichen Bereiche fest.

<AREA> Attribute SHAPE="Formtyp" [circle,poly,rect]
 CHOORDS ="Koordinaten" [je nach SHAPE-Attribut]
 HREF="Verweisziel"
 ALT = "Ersatztext"

Formtypen :

shape=rect für ein Rechteck mit Angabe der Koordinaten für x1,y1,x2,y2

shape=circle für einen Kreis mit Angabe der Koordinaten für x,y,r

shape=polygon) für ein Polygon mit x1,y1,x2,y2 ... xn,yn

Beispiel:

```
<map name="Karte">
  <area shape=rect coords="15,5,249,49" href="ort1.gif" alt="Ort1">
  <area shape=rect coords="1,51,149,99" href="ort2.gif" alt="Ort2">
  <area shape=circle coords="250,150,20" href="ort3.gif" alt="Ort3">
</map>

```

Achtung : für die internen Namen der Elemente (Attribut name) gilt Groß-Klein-Unterscheidung !

Grafikformate im WWW :

Im WWW werden vor allem zwei Grafik-Dateiformate verwendet , **GIF und JPEG**.

GIF (Graphics Interchange Format) arbeitet als verlustfreies Komprimierungsverfahren ,
JPEG (Joint Photographic Expert Group) erreicht höhere Kompressionsraten, jedoch mit
Bildatenverlusten. JPEG eignet sich besonders für fotorealistische Bilder mit vielen Farben,
GIF hingegen für Grafiken mit Schriften und scharfen Konturen.

GIF ermöglicht auch mehrere Grafiken in einer einzigen Grafikdatei zu speichern
und damit animierte Grafiken zu realisieren, sowie transparente Hintergründe zu definieren.

GIF unterstützt auch die Interlaced-Funktion (schichtweises Aufbauen der Grafik).

Das GIF-Format kann nur bis zu maximal 256 Farben verarbeiten, JPEG bis zu 16,7 Mio.

Bei JPEG kann der Kompressionsfaktor angegeben werden, je höher umso kleiner die Dateigröße
und desto schlechter die Bildqualität (5 bis 10 reichen für eine gute Bildqualität)

PNG (Portable Network Graphic) ist eine Neuentwicklung fürs WWW

und wird von moderneren Browser bereits unterstützt.

komprimiert verlustfrei wie das GIF-Format

unterstützt 16,7 Mio. Farben wie das JPEG-Format

transparente Hintergrundfarben wie das GIF-Format

Interlaced-Funktion (schichtweise Aufbauen der Grafik)

Bildinformation zu Gammakorrektur

Bildinformationen wie Bildtitel, -autor, -beschreibung, Copyright, Erstellungsdatum, ...

Tabellen :

Aufbau einer Tabelle :

	Spalte 1	Spalte 2	Spalte 3	Spalte 4
Zeile 1				
Zeile 2				
Zeile 3				

Tabelle :

<TABLE>

Tabellenelemente

</TABLE>

Tabellenelemente :

<CAPTION> ... </CAPTION>

caption = Tabellen Überschrift

<TH> ... </TH>

table header = Spalten Überschrift

<TR> ... </TR>

table row = Tabellen Zeile

<TD> ... </TD>

table data = Tabellen Daten

BSP. :

```
<TABLE>
  <CAPTION> <H2> TABELLEN </H2> </CAPTION>
  <TH> Inhalt </TH>
  <TH> Erläuterung </TH>
  <TR>
    <TD> erste Spalte der ersten Zeile </TD>
    <TD> zweite Spalte der ersten Zeile </TD>
  </TR>
  <TR>
    <TD> erste Spalte der zweiten Zeile </TD>
  </TR>
</TABLE>
```

Attribute zu TABLE :

BORDER = "Pixel" Dicke des Außenrahmens einer Tabelle in Pixeln
CELLSPACING = "Pixel" Dicke der Gitternetzlinien zwischen den Zellen
CELLPADDING = "Pixel" Abstand der Zellendaten zum Zellenrand

Ausrichtungsattribute :

ALIGN = "Horizontale Ausr." Werte : left / right / center / justify / char
VALIGN = "Vertikale Ausr." Werte : top / middle / bottom / baseline
COLSPAN = "Mehrfach Bereich für Spalten"
ROWSPAN = "Mehrfach Bereich für Zeilen"
WIDTH = "Zellenbreite"

Beispiel :

```
<TABLE BORDER="3" CELLSPACING="1" CELLPADDING="6">
  <TH COLSPAN="5" ALIGN="left">
    <TR VALIGN="TOP" ALIGN="right">
      ...
```

unsichtbare Tabellen dienen zur Ausrichtung von Text und Grafik.

Beispiel - unsichtbare Tabelle :

```
<TABLE BORDER=0 >
...

```

Gesamtbreite und Gesamthöhe der Tabelle festlegen :

Beispiel:

```
<table border width=80% height=300>
```

Die Angaben sind nur wirksam, wenn der Tabelleninhalt kleiner ist,

Breite von Spalten festlegen :

Beispiel:

```
<th width=300> Titel 1.Spalte</th>
```

da die Angabe spaltenweit gilt, braucht sie nur einmal pro Spalte (erste Zeile) angegeben werden.

Höhe von Zeilen festlegen :

Beispiel:

```
<tr height=300> ... </tr>
```

Die Angabe gilt für die gesamte Zeile .

Tabellen - Hintergrundfarben definieren: ab HTML 4.0

Beispiel:

```
<table border bgcolor=#CCFFFF bordercolor=#000099>
  <tr bgcolor=#CCCCFF>
    <td bgcolor=#CCFFCC>
```

Spalten vordefinieren: ab HTML 4.0 kann die Anzahl und Breite der Spalten vordefiniert werden.

Beispiel :

```
<table>
  <colgroup>
    <col width="80">
    <col width="100">
  </colgroup>
  <tr>
    ...
```

Rahmen (Frames) :

Mit Hilfe von Frames kann eine Aufteilung auf mehrere Fensterbereiche erfolgen.
Jedes Fenster kann dabei ein eigenes Dokument darstellen .

Grundgerüst mit Frame-Set :

```
<html>
  <head>
  </head>
  <frameset ...>
    Frame-Definitionen
  </frameset>
  <noframes>
    Text für Browser, die keine Frames umsetzen
  </noframes>
</html>
```

Attribute zu FRAMESET : zur Aufteilung der Frames

COLS = Breite : Angabe in n (Pixel) , n% (prozentuell) , * (restlicher Bereich)

ROWS = Höhe : Angabe in n , n% , *

„cols“ und „rows“ können nur alternativ und nicht gemeinsam verwendet werden !

die cols - Aufteilung erfolgt von links nach rechts gemäß der Reihenfolge der Attributwerte

Beispiel : Aufteilung für zwei Frames nebeneinander

```
<frameset cols="200,*">
```

Frame 1 Breite = 200 Pixel	Frame 2 Breite = restlicher Bereich
-------------------------------	--

die rows - Aufteilung erfolgt von oben nach unten gemäß der Reihenfolge der Attributwerte

Beispiel : Aufteilung für zwei Frames untereinander

```
<frameset rows="25%,75%">
```

Frame 1	Höhe = 25% der Gesamthöhe
Frame 2	Höhe = 75% der Gesamthöhe

Frames definieren :

Die Angabe, welche HTML-Dateien in den einzelnen Fenstern angezeigt werden ,
erfolgt mit <FRAME SRC = "Dateiname" >

Beispiel: horizontale und vertikale Aufteilung durch Einschachtelung

```
<frameset cols="200,*">
  <frame src="verweise.html" name="Frame1">
  <frameset rows="150,*">
    <frame src="titel.html" name="Frame2">
    <frame src="control.html" name="Frame3">
  </frameset>
</frameset>
```

Frame 1 zeigt „verweise.html“	Frame 2	zeigt „titel.html“
	Frame 3	zeigt „control.html“

Die gemischte Aufteilung in Reihen und Spalten kann durch Einschachtelung von frameset erfolgen, wobei jedes eingeschachtelte frameset anstelle eines frame-Tags eine weitere Unterteilung dieses Frames ausführt.

Namen für Frames sind mit dem Attribut name= "FrameName" festzulegen und dann sinnvoll, wenn die Ausgabe eines Verweises in einen anderen Frame umgeleitet werden soll.

Attribute zu FRAME :

SRC	URL - Verweis auf das darzustellende Dokument
NAME	Frame Name
FRAMEBORDER	Rahmendicke in Pixel (0 = unsichtbar)
MARGINWIDTH	Seitenränder in Pixel
MARGINHEIGHT	Rand oben und unten
SCROLLING	Scrollbars (Bildlaufleisten) [yes ,no,auto]
NORESIZE	Rahmengröße vom Anwender nicht veränderbar

Umleitung von Verweisen zu anderen Frames :

Beispiel : Frame-Aufteilung erfolgt in der Datei „index.html“

```
<frameset cols="200,*">
  <frame src="auswahl.html" name="Frame1">
  <frame src="inhalt.html" name="Frame2">
</frameset>
```

die Datei „auswahl.html“ enthält Verweise, die nicht im gleichen Frame sondern in den Frame2 ausgegeben (umgelenkt) werden sollen

```
<body>
  <a href="seite1.html" target="Frame2" > Seite 1 </a>
</body>
```

Mit dem **Attribut target="FrameName"** wird das Ausgabefenster für den Verweis definiert.

Groß- und Kleinschreibung werden für die internen Frame-Namen unterschieden !

Weiters kann auch mit <base target= > im Head-Teil ein Default-Frame für alle Links festgelegt werden.

Beispiel:

```
<head>
  <base target="Frame2">
</head>
```

Verwendung von Frames :

Vorteile :

- Darstellung mehrerer Seiten
- Verzeichnis mit Verweisen kann neben dem Informationsfenster angezeigt werden.
- Fenster mit immer ausführbaren Steuerverweisen einblenden

Nachteile :

- Frames werden nicht von allen WWW-Browsern ausgeführt
- Frames bewirken längere Ladezeiten
- problematisches Anwählen untergeordneter Seiten mit Lesezeichen

Formular Elemente :

Formulare können Eingabefelder, mehrzeilige Textfelder, Schalter u. Optionsfelder enthalten und die Eingaben clientseitig (mit JavaScript) weiterverarbeitet oder an den Server gesandt werden.

Definition :

```
<FORM action=[ ] method=[ ] >  
... Formular Elemente  
</FORM>
```

Beispiel :

```
<FORM action="anmeldung.php" method="post">  
...  
</FORM>
```

Mit dem Attribut **action= []** wird die Aktion nach dem Absenden (submit) der Formulardaten am Server festgelegt, mit **method= []** wird dabei die Übertragungsmethode bestimmt.

method = "get" Formulardaten werden in der URL mitgesandt

method = "post" Formulardaten werden in eigenem Datenblock zum Server übertragen

Definition der Formular Elemente :

Schalter	:	< INPUT TYPE = "button" >
Auswahl Schalter	:	< INPUT TYPE = "radio" >
Kontroll Schalter	:	< INPUT TYPE = "checkbox" >
Absenden	:	< INPUT TYPE = "submit" >
Rücksetzen	:	< INPUT TYPE = "reset" >
Textfelder	:	< INPUT TYPE = "text" >
Textbereiche	:	< TEXTAREA> Eingabetext </TEXTAREA >
Paßwortfelder	:	< INPUT TYPE = "password" >
Hidden Element	:	< INPUT TYPE = "hidden" value="">
Auswahl	:	< SELECT > </SELECT >
Optionen	:	< OPTION > </OPTION >

Eingabefeld - Beispiel:

```
<form>  
Name : <input type=text name="Name" size=50 maxlength=50> <br>  
</form>
```

Beispiel für mehrzeiliges Eingabefeld:

```
<textarea name=textfeld rows=5 cols=40></textarea>
```

Beispiel zu Auswahlliste :

```
<select name="Pizza" size=5>  
  <option value="P101"> Pizza Napoli  
  <option value="P102"> Pizza Funghi  
</select>
```

mit dem Attribut **value=Wert** kann ein bestimmter Wert anstatt des Listentextes übertragen werden.

Beispiel mit Button :

```
<input type=button value="Zurück" onClick="history.back()">  
einfache Buttons sind eigentlich nur sinnvoll in Verbindung mit JavaScript .
```

Beispiel zu Radiobuttons :

```
<input type=radio name="klasse" value="4AI"> 4AI <br>  
<input type=radio name="klasse" checked value="5AI"> 5AI <br>
```

eine Gruppe von Radiobuttons wird über einen gleichen Namen (mit Attribut **name=**) definiert, mit dem Attribut **value=** wird der Wert für die Formulardaten festgelegt, mit **checked** kann eine Option vorselektiert werden.

Allgemeine HTML - Datei Angaben :

werden **im Headteil** angegeben

Titel Element :

<TITLE> Kurzbeschreibung der HTML-Seite
</TITLE>

Jede HTML-Datei sollte einen Titel erhalten :

- der Titel wird bei der Anzeige im WWW-Browser in der Titelzeile angezeigt
- der Titel wird für die Kurzbeschreibung bei den Lesezeichen verwendet
- der Titel wird in der Liste der bereits besuchten Seiten angezeigt
- der Titel dient den Suchprogrammen im WWW als Suchinformation

Beispiel:

```
<head>
  <title> HTML Kurs Teil 1 </title>
</head>
```

Base Element :

< BASE HREF ="Datei" >

Dient zur Angabe von Basisadressen für Verweise dieses Dokuments

Beispiel:

```
<head>
  <base href="/grafiken">
</head>
```

Base Target Element :

< BASE TARGET = "FrameName" >

Definiert das Zielfenster der Verweise dieses Dokuments.

Beispiel:

```
<head>
  <base target="Inhalt">
</head>
```

META Elemente :

<META NAME ="Bedeutung" CONTENT="Inhalt">

Meta Elemente nehmen Verwaltungsinformationen auf.

Es können Angaben zu Autor und Inhalt eines Dokuments angegeben oder auch Anweisungen für WWW-Server, Browser und Suchprogramme ausgeführt werden.

Beispiel:

```
<head>
  <meta name="description" content="Beschreibung für Suchdienst ">
  <meta name="author"      content="Name des Verfassers">
  <meta name="keywords"   lang="de" content="Stichwörter für Suche">
  <meta name="date"       content="1998-09-30">
</head>
```

HTML - Dokument Typ : Verarbeitungsinformation an den Browser

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
<html>  ...
```

Bedeutung :

DOCTYPE HTML PUBLIC : öffentlich verfügbare HTML-Dokument Typ Definition (DTD)

W3C : W3-Konsortium ist der Herausgeber

DTD - HTML 4.0 : HTML-Sprachstandard 4.0

EN : Landeskürzel für Englisch

HTML Erweiterungen :

Style-Sheets :

Style-Sheets sind eine Ergänzung zu HTML zur Definition von Formatierungen.

Damit können zu HTML-Tags benutzdefinierte Formatierungsangaben festgelegt werden, wie Schriftgröße, Farben, Ränder, ...

Die Definitionen von Style-Sheets werden zentral im Headteil oder besser in separaten Dateien angeführt.

Beispiel :

```
<html>
<head>
  <style type="text/css">
    <!--
    body { color : white }
    p { color : blue;      /* Kommentar */
      font-size : 12pt;
      font-family : Arial
    }
    h1{ color : red;
      font-size : 18pt;
    }
    -->
  </style>
</head>
<body>
  ...
</body>
</html>
```

Dynamisches HTML :

Dynamisches HTML (DHTML) ist der Sammelbegriff für diverse Techniken, vor allem für den Einsatz von Scriptsprachen wie JavaScript.

Document Object Model (DOM) :

Das Document Object Model bildet alle Elemente einer angezeigten WWW-Seite als Objekthierarchie ab. Diese Objekthierarchie soll dann in Programmiersprachen wie JavaScript als Objektbibliothek zur Verfügung stehen.