


1 HTML-Grundlagen

1.1 Einführung in HTML

1.1.1.1 HTML als "lingua franca" des Web

- HTML = „Hypertext Markup Language“
- HTML als "lingua franca" = "Brot- und Butter-Sprache",
 - eine Sprache, die jeder kennt, jeder spricht, jeder leicht erlernen kann und jeder braucht.
 - erfolgreichste und verbreitetstes Dateiformat der Welt.
 - Sprache zur Strukturierung von Texten, wobei aber auch die Möglichkeit besteht, Grafiken und multimediale Inhalte in Form einer Referenz einzubinden und in den Text zu integrieren.
 - Mit HTML können Sie Überschriften, Textabsätze, Listen und Tabellen erzeugen.
 - Sie können anklickbare Verweise auf beliebige andere Web-Seiten oder Datenquellen im Internet erzeugen.
 - Nicht-textuelle Inhalte können Sie wie bereits erwähnt referenzieren.
 - Sie können Formulare in den Text integrieren.
 - Schnittstellen für Erweiterungssprachen wie [CSS Stylesheets](#) oder [JavaScript](#) an, mit deren Hilfe Sie HTML-Elemente nach Wunsch gestalten und formatieren oder Interaktion mit dem Anwender realisieren können.
- Ergänzungen zu HTML
 - Als "lingua franca" ist HTML allerdings mit diversen spezielleren Anforderungen überfordert.
 - HTML allein kann weder Grafik-Designer befriedigen, die jeden Pixel am Bildschirm kontrollieren wollen, noch Daten-Designer, die aus der Welt der relationalen Datenbanken kommen und sich anwendungs-spezifische Lösungen wünschen.
 - Deshalb gibt es heute Style-Sprachen wie CSS, und es gibt Lösungen wie [XML](#), um anwendungsspezifisches Daten-Design zu ermöglichen.

1.1.1.2 HTML als software-unabhängiges Klartextformat

- HTML ist ein so genanntes Klartext-Format.
 - HTML-Dateien können mit jedem beliebigen Texteditor bearbeitet werden
 - Es gibt also keine bestimmte Software, die man zum Erstellen von HTML-Dateien benötigt.
 - Da HTML ein Klartextformat ist, lässt es sich auch hervorragend mit Hilfe von Programmen generieren.
 - Von dieser Möglichkeit machen beispielsweise  [CGI](#)-Scripts Gebrauch.
 - Wenn Sie im Web zum Beispiel einen Suchdienst benutzen und nach einer Suchanfrage die Ergebnisse präsentiert bekommen, dann ist das, was Sie am Bildschirm sehen, HTML-Code, der von einem Script generiert wurde.

1.1.1.3 HTML als Auszeichnungssprache

- HTML ist eine so genannte Auszeichnungssprache (Markup Language).
 - Sie hat die Aufgabe, die logischen Bestandteile eines textorientierten Dokuments, wie Überschriften, Textabsätze, Listen, Tabellen oder Grafikreferenzen zu beschreiben.
 - Einige dieser Elemente haben wiederum Unterelemente.
 - So enthält ein Textabsatz zum Beispiel eine als betont oder fett markierte Textstelle, eine Aufzählungsliste besteht aus einzelnen Listenpunkten, und eine Tabelle gliedert sich in einzelne Tabellenzellen.
 - Die meisten dieser Elemente haben einen fest definierbaren Erstreckungsraum.
 - So geht eine Überschrift vom ersten bis zum letzten Zeichen, eine Aufzählungsliste vom ersten bis zum letzten Listenpunkt, oder eine Tabelle von der ersten bis zur letzten Zelle.
 - Auszeichnungen markieren Anfang und Ende von Elementen.
 - Bsp. Überschrift→
[Überschrift] Text der Überschrift [Ende Überschrift]
 - Bsp. mit Unterelement→
[Liste]
[Listenpunkt] Text des Listenpunkts [Ende Listenpunkt]
[Listenpunkt] Text des Listenpunkts [Ende Listenpunkt]
[Ende Liste]
- Web-Browser, die HTML-Dateien am Bildschirm anzeigen, lösen die Auszeichnungsmarkierungen auf und stellen die Elemente dann in optisch gut erkennbarer Form am Bildschirm dar.

1.1.1.4 HTML-Versionen

- unterschiedliche Versionen mit unterschiedlichem Sprachumfang
 - HTML 1.0 enthielt Auszeichnungen für Standard-Elemente wie Überschriften, Textabsätze, für Grafikreferenzen und natürlich für Verweise.
 - HTML 2.0 → 1995 offizieller Sprachstandard.
 - Die Version 2.0 von HTML gilt heute als der allerkleinste gemeinsame Nenner, der auch von älteren Browsern interpretiert werden kann.
 - HTML 3.2 → 1997 offizieller Sprachstandard
 - Tabellen offizieller Bestandteil von HTML.
 - HTML 4.0 → 1998 als Sprachstandard verabschiedet.
 - HTML 4.0 stellt den Versuch dar, sich auf die Kernaufgaben von HTML zurückzubedenken. → aktueller Stand 4.01
 - Einbindung von CSS Style Sheets und von Scriptsprachen wie JavaScript in HTML.
- HTML im Zuge der Etablierung von **XML** in Gestalt von **XHTML** neu definiert wurde.
 - Es existiert heute neben dem SGML-basierten, klassischen HTML das XML-basierte XHTML. (SGML Standard Generalized Markup Language) (HTML ist eine Anwendung von SGML, XML ist eine Untermenge von SGML)

1.2 Aufbau einer HTML-Seite

1.2.1 Notierung von Elementen und Tags in HTML

- Der Inhalt von HTML-Dateien steht in **HTML-Elementen**.
- HTML-Elemente werden durch so genannte **Tags** markiert.
- Fast alle HTML-Elemente werden durch ein einleitendes und ein abschließendes Tag markiert.
- Der Inhalt dazwischen ist der "Gültigkeitsbereich" des entsprechenden Elements. Tags werden in spitzen Klammern notiert.

- Bsp:

```
<h1>Hello World</h1>
```

- Beispiel zeigt eine Überschrift 1. Ordnung.
 - Das einleitende Tag <h1> signalisiert, dass eine Überschrift 1. Ordnung folgt (h = heading = Überschrift).
 - Das abschließende Tag </h1> signalisiert das Ende der Überschrift. Ein abschließendes Tag beginnt mit einem Schrägstrich "/".
- Verschachtelung von Elementen
 - Elemente können ineinander verschachtelt werden. Auf diese Weise entsteht eine hierarchische Struktur.
 - Bsp:

```
<h1><i>Hello</i> World</h1>
```

 - *i*-Element steht für italic (= kursiver Text). Der Text zwischen <i> und </i> wird als kursiv interpretiert
 - Attribute in Tags
 - Einleitende Tags und Standalone-Tags können zusätzliche Angaben in Form von Attributen enthalten.
 - Attribute sind je nach Tag unterschiedlich
 - Bsp:

```
<h1 align="center">Hello World</h1>
```

 - bewirkt, dass der Text zentriert ausgerichtet wird.

1.2.2 Grundgerüst einer HTML-Datei

- Eine HTML-Datei besteht grundsätzlich aus folgenden Teilen:
 - Genutzte Dokumenttypdefinition = DTD (Angabe zur verwendeten HTML-Version)
 - Header (Kopfdaten. z.B. Angaben zu Titel u.ä.)
 - Body (Körper - anzuzeigender Inhalt, also Text mit Überschriften, Verweisen, Grafikreferenzen usw.)

- Bsp.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Text des Titels</title>
</head>
<body>

</body>
</html>
```

1.2.3 Spezielle Zeichen in HTML

- Deutsche Umlaute und scharfes S

- Wenn in der HTML-Datei keine Angaben zum verwendeten Zeichensatz notiert sind, sollten im Hinblick auf das Internet und die internationale Verwendung deutsche Umlaute und scharfes S durch spezielle, dafür vorgesehene **benannte Zeichen** ersetzen.

- ä → `ä`
- Ä → `Ä`
- ö → `ö`
- Ö → `Ö`
- ü → `ü`
- Ü → `Ü`
- ß → `ß`

- Weitere Sonderzeichen:

- Es gibt eine Menge weiterer Sonderzeichen, die durch solche Umschreibungen dargestellt werden sollten.

- Bsp.: Euro-Zeichen

- € → `€`

- HTML-eigene Zeichen

- bei Zeichen, die in HTML eine bestimmte Bedeutung haben, müssen diese maskieren werden.

- Bsp.

- < → `<`
- > → `>`
- & → `&`
- " → `"`

1.2.4 wichtige Elemente und Tags

1.2.4.1 Kommentare

- an beliebigen Stellen innerhalb einer HTML-Datei können Kommentare eingefügt werden.
- Kommentare werden von Web-Browsern ignoriert, d.h. bei der Präsentation nicht angezeigt.
- sinnvoll, um interne Anmerkungen zu bestimmten Textstellen zu machen oder um verwendete HTML-Befehle intern zu kommentieren.
- Bsp:

```
<!-- Kommentartext -->
```

1.2.4.2 Header

- `<head>`
 - kennzeichnet den Kopfbereich einer HTML-Datei
- `<title>`
 - wird bei der Anzeige im Web-Browser in der Titelzeile des Anzeigefensters angezeigt
 - wird vom Web-Browser beim Setzen von Lesezeichen (Bookmarks, Favoriten) auf die Datei verwendet
 - dient im Web vielen automatischen Suchprogrammen als wichtiger Input
- weitere Meta-Angaben innerhalb des `<head>`-Tags möglich z.B. Anweisungen für Web-Server, Web-Browser und automatische Suchprogramme

1.2.4.3 dateiweite Einstellungen → `<body>`

- dateiweite Einstellungen werden als Attribute des `<body>`-Tags spezifiziert
- Hintergrundfarbe
 - `<body bgcolor="#CCFFFF">`
- Hintergrundbild
 - `<body background="background.jpg">`
- Hinweis:
deprecated → gehören noch zum HTML-Standard, sollen in Zukunft jedoch entfallen, da es mit CSS realisiert werden kann

1.3 logische Textformatierung

→ dient der Strukturierung von Text

→ Elemente bilden eigene Absätze

- Überschriften

- HTML unterscheidet 6 Überschriftenebenen

- `<h1>Überschrift 1. Ordnung</h1>`

- `<h2>Überschrift 2. Ordnung</h2>`

- Überschriften ausrichten → Attribute

- `<h1 align="center">Überschrift 1. Ordnung zentriert</h1>`

- `<h2 align="right">Überschrift 2. Ordnung`

- `rechtsbündig</h2>`

- Hinweis:

- Ausrichtungsattribute sind *deprecated* → gehören noch zum HTML-Standard, sollen in Zukunft jedoch entfallen, da es mit CSS realisiert werden kann

- Textabsätze

- `<p>` (*p = paragraph = Absatz*) leitet einen Textabsatz ein. `</p>` beendet den Textabsatz und steht am Ende des Absatztextes

- `<p>Hier beginnt ein Absatz, und hier ist er zu Ende.</p>`

- `<p>Hier beginnt ein neuer Absatz, und hier ist er zu Ende.</p>`

- jeder neue Textabsatz bewirkt einen Zeilenumbruch

- Textabsätze ausrichten → Attribute

- wie Überschriften ausrichten

- Zeilenumbruch

- Text innerhalb von normalen Absätzen, Listen, sowie in Überschriften oder Tabellenzellen wird vom Web-Browser bei der Anzeige automatisch umgebrochen.

- Sie können jedoch an einer gewünschten Stelle einen Zeilenumbruch erzwingen.

- `
` (*br = break = Umbruch*) fügt an der gewünschten Stelle einen Zeilenumbruch ein

- Geschützte Leerzeichen

- verhindern, dass bei einem Leerzeichen ein automatischer Zeilenumbruch erfolgen darf

- Die Zeichenfolge ` ` erzeugt ein geschütztes Leerzeichen (*nbsp = nonbreaking space = nicht umbrechbares Leerzeichen*).

- Bsp. → [HTML 4.0](#)

- Listen/Aufzählungen

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

- unnummerierte Liste

```
<ul>
  <li>   </li>
  <li>   </li>
  <li>   </li>
</ul>
```

- leitet eine Aufzählungsliste
- Mit beginnt ein neuer Punkt innerhalb der Liste
- beendet den Listeneintrag.
- beendet die Liste
- Verschachteln von Listen ist möglich → Liste in Listeneintrag

- nummerierte Liste

```
<ol>
  <li>   </li>
  <li>   </li>
  <li>   </li>
</ol>
```

1.3.1 physischen Auszeichnung von Text

→ dient der **Formatierung** von Text (direkte Angaben zur Schriftformatierung)

→ Elemente bilden keine eigenen Absätze

- fett → `...`
- kursiv → `<i>...</i>`
- unterstrichen → `<u>...</u>`
 - Hinweis:
`<u>` ist *deprecated* → gehören noch zum HTML-Standard, sollen in Zukunft jedoch entfallen, da es mit CSS realisiert werden kann

- Bsp:

```
<p>Das Schwein ist <b>fett</b> und der Turm von Pisa ist  
<i>schief</i>.<br>  
Und was ist <b><i>fett und schief</i></b>?</p>
```

• Schriftart, Schriftgröße und Schriftfarbe

- sind allesamt *deprecated* → gehören noch zum HTML-Standard, sollen in Zukunft jedoch entfallen, da es mit CSS realisiert werden kann
 - `Ziemlich riesiger Text`
 - `Knallroter Text`

- ``

1.4 Tabellen

- Tabellen, um tabellarische Daten darzustellen, oder um Text und Grafik am Bildschirm zu verteilen

- genereller Aufbau

```
<table>
  <tr>
    <th> </th>
    <th> </th>
    <th> </th>
  </tr>
  <tr>
    <td> </td>
    <td> </td>
    <td> </td>
  </tr>
  <tr>
    <td> </td>
    <td> </td>
    <td> </td>
  </tr>
</table>
```

- `<table>` leitet eine Tabelle ein
- `<tr>` leitet eine neue Tabellenzeile ein
- Tabelle kann Kopfzellen und gewöhnliche Datenzellen enthalten
 - `<th>` leitet eine Kopfzelle ein → kann entfallen
 - `<td>` eine normale Datenzelle
- Anzahl der Zellen sollte bei jeder Zeile gleich sein, so dass die Tabelle durchweg die gleiche Anzahl Spalten pro Zeile hat.
- In der ersten Zeile wird durch die Anzahl der dort definierten Zellen die Anzahl der Spalten festgelegt.
- Tabellenzellen dürfen auch leer sein → `<td></td>`.

- Zellenabstand und Zelleninnenabstand

- `cellpadding`

- Innenabstand einer Zelle, also den Abstand zwischen Zellenrand und Zelleninhalt in Pixeln

- `<table cellpadding="10">`

- `cellspacing`

- Abstand der Zellen untereinander in Pixeln

- `<table cellspacing="10">`

- border
 - Rahmen und sichtbare Gitternetzlinien in der Tabelle
 - `<table border="3">` → 3 Pixel
 - `<table border="0">` → kein Rahmen

- Breiten- und Höhenangaben
 - Es können die gesamte Tabelle, sowie für einzelne Spalten und Zeilen Angaben zu Breite und Höhe notieren
 - `<table width=...>`
 - Bestimmt die Breite der gesamten Tabelle
 - `<table height=...>`
 - Höhe der gesamten Tabelle in Bezug auf das Anzeigefenster
 - `<th width=...>`
 - Bestimmt die Breite einer Tabellenzelle und damit die Breite aller Zellen der gleichen Spalte
 - → *deprecated*
 - `<td height=...>`
 - Bestimmt die Höhe einer Tabellenzelle und damit die Höhe aller Zellen der gleichen Tabellenzeile
 - → *deprecated*

 - Werte als
 - absolute Zahlenangaben → px
 - prozentuale Angaben → %
 - Angaben zu Breite und Höhe sind nur wirksam, wenn der Tabellen- bzw. Zelleninhalt kleiner ist, so dass leerer Raum entsteht. Wenn der Tabelleninhalt größer ist, werden die Angaben zu Breite und Höhe außer Kraft gesetzt. Die Tabelle bzw. Zelle (und mit ihr Spalte und Zeile) werden dann so dimensioniert, dass der gesamte Inhalt angezeigt wird.

- Ausrichtung von Zellen
 - orientiert am Zellenrand
 - horizontal (linksbündig, zentriert, rechtsbündig)
 - vertikal (obenbündig, mittig, untenbündig)

 - Angaben sind sowohl auf einzelne Zellen (also `th`- und `td`-Elemente) anwendbar, als auch auf Tabellenbereiche wie Tabellenzeilen (`tr`),

 - `align="left" / "center" / "right"`
 - `valign="top" / "middle" / "bottom"`

- Hintergrundfarben und -bilder
 - können sowohl für eine gesamte Tabelle als auch für einzelne Zeilen oder Zellen eine definiert werden

- bgcolor="#xxxxxx"
- style="background-image:url(background.jpg) "

- Zellen verbinden

- colspan

- eine Zelle erstreckt sich über mehrere Spalten hinweg
- nur wirksam, wenn die Tabelle mindestens so viele Spalten besitzt wie angegeben.
- Wenn Sie in einer Tabellenzeile eine Zelle beispielsweise über vier Spalten hinweg "spannen", sollten Sie in den anderen Zeilen der Tabelle anstelle dieser einen Zelle vier eigene Zellen definieren.
- Bsp:

```
<table>
  <tr>
    <th colspan="2">Studenten sind</th>
  </tr>
  <tr>
    <td> dies </td>
    <td> das </td>
  </tr>
</table>
```

- rowspan

- eine Zelle in einer Spalte erstreckt sich über mehrere Zeilen hinweg
- nur wirksam, wenn die Tabelle mindestens so viele Zeilen besitzt wie angegeben
- Wenn Sie in einer Tabellenzeile eine Zelle beispielsweise über vier Zeilen hinweg "spannen", sollten Sie in der gleichen Zeile den Inhalt der ersten danebenstehenden Zelle definieren. In den drei folgenden Zeilen sollten Sie an gleicher Stelle die drei übrigen Zellen definieren, die neben der gestreckten Zelle stehen sollen.
- Bsp.

```
<table>
  <tr>
    <th rowspan="2">Dozenten sind </th>
    <td> dies </td>
  </tr>
  <tr>
    <td> das </td>
  </tr>
</table>
```

1.5 Verweise

- Alle Verweise in HTML haben einen einheitlichen Aufbau, egal ob sie zu einem Verweisziel in der gleichen Datei, zu einer anderen Datei im eigenen Projekt, zu einer beliebigen WWW-Adresse oder zu einer beliebigen Datei eines anderen Dateityps im Internet oder lokal auf dem eigenen Rechner führen
- genereller Aufbau (projektexterne Verweise)
 - `<a>`
 - Attribut `href=` erforderlich
 - Wert des `href`-Attributs → Verweisziel
 - Inhalt des `a`-Elements, also zwischen `<a>` und ``, → Text, der dem Anwender als Verweis angeboten wird (oder Grafik, z.B. Button)
 - Bsp.
`Universität Trier`
- Zielfenster für Verweise bestimmen
 - Per Voreinstellung werden alle Verweise im aktuellen Fenster geöffnet
 - Attribut `target=` im `<a>`-Tag kann ein Zielfenster für einen Verweis festgelegt werden
 - Verweis in neuem Fenster öffnen → `target="_blank"`
 - Bsp.
` ... `
 -
- projektinterne Verweise
 - relative Angaben zum Verweisziel machen
 - Projekt bleibt dadurch flexibler, und die Verweise funktionieren auch in anderen Umgebungen (z.B. solange Sie das Projekt lokal auf Ihrem PC erstellen und austesten wollen, oder wenn Sie es mal auf CD-ROM präsentieren möchten).
 - Bsp.
`zur Startseite`
`zur 1. Seite`
`zum PDF-Dokument`
`zum Bild1`
- Anker definieren und Verweise zu Ankern
 - Anker → Sprungziele die direkt von Verweisen angesprungen werden können
 - besonders bei langen Seiten sinnvoll
 - Anker definieren
 - Anker wird genau wie ein Verweis mit Hilfe des `a`-Elements erzeugt.
 - kein Attribut `href=`-Attribut, sondern stattdessen ein Attribut `name=`.
 - Bsp.
`...`
 - Namen dürfen keine Leerzeichen und keine deutschen Umlaute enthalten

- Verweise zu Ankeren
 - innerhalb einer Datei
 - `Verweistext.`
 - Verweis zu einem Anker in einer anderen Datei
 - `Verweistext.`
- Email-Verweise
 - `T. Bräuninger`
 - `A. Müller`
- Download-Verweise
 - `ZIP-Datei`

1.6 Grafiken

- Dateiformate für Web-gerechte Grafiken
 - GIF, JPEG, PNG
- Dateigröße
 - Ladezeiten sollten kurz gehalten werden
 - Bildschirmauflösung i.d.R. zwischen 72 und 96 dpi (für Print z.B. 300 dpi)
 - Bilder auf diese Größe anpassen
 - ggf. Anzahl der Farben verringern oder Bild verkleinern
- ``-Tag
 - Standalone-Tag ohne Elementinhalt und ohne End-Tag
 - Attribut `src=` → bestimmt die Grafikdatei
 - unbedingt auf Groß-/Kleinschreibung von Dateinamen und Verzeichnisnamen achten
 - Die meisten Server-Rechner laufen mit Betriebssystemen, bei denen streng zwischen Groß- und Kleinschreibung unterschieden wird.
 - Am einfachsten und sichersten ist es, wenn Sie alle Datei- und Verzeichnisnamen kleinschreiben
 - Attribut `alt=` → Alternativtext an für den Fall, dass die Grafik nicht angezeigt werden kann
 - Bsp.:
 - ``
- Breite und Höhe von Grafiken
 - Breite und Höhe der Grafik sollte stetes angegeben werden
 - Dadurch entnimmt der Web-Browser bereits der HTML-Datei, wie groß die Grafik ist, und muss nicht warten, bis er die entsprechende Header-Information der Grafikdatei ausgelesen hat.

- So kann er die gesamte Web-Seite bereits am Bildschirm aufbauen und bei noch nicht eingelesenen Grafiken erst mal eine entsprechend große Freifläche anzeigen.
- Bsp.:

```

```

- Rahmen um Grafiken
 - `border=` → definiert Rahmen um die Grafik

1.7 Frames

- zur Aufteilung des Anzeigebereichs des Browsers in verschiedene, frei definierbare Segmente aufteilen.
 - Jedes Segment kann eigene statische oder wechselnde Inhalte enthalten.
 - Verweise in einem Frame können Dateien aufrufen, die dann in einem anderen Frame angezeigt werden.

- Grundgerüst einer HTML-Datei mit Framesets
 - Um Frames zu definieren, brauchen Sie eine spezielle HTML-Datei, in der ein Frameset definiert wird
 - bestimmt die Fensteraufteilung
 - anderes Grundgerüst erforderlich als das sonst übliche Grundgerüst von HTML-Dateien
 - [Dokumenttyp-Angabe](#)
 - kein `body`-Element

- Bsp.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<html>
<head>
<title> ... </title>
</head>
<frameset ...> <!-- Frameset-Definition -->
  <frame ...> <!-- Framefenster-Definition -->
  <noframes>
    angezeigt, wenn der Browser keine Frames anzeigen kann
  </noframes>
</frameset>
</html>

```

- Der Titel (`<title>...</title>`), den Sie in der Datei mit der Frameset-Definition angeben, wird während der gesamten Dauer des Frame-Sets angezeigt

- Framesets definieren
 - Aufteilung des Anzeigefenster des Browsers festlegen

- Bsp.1:

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

- `cols=` → teilt das Anzeigefenster in Spalten auf
- im Beispiel → zwei Spalten
- linke Spalte → 200 Pixel des Anzeigefensters
- rechte Spalte → Rest.

- Bsp.2:

```
<frameset rows="20%,*,60">
  ...
</frameset>
```

- `rows=` → teilt das Anzeigefenster in Zeilen auf
- im Beispiel → drei Zeilen
- linke Zeile → 20% der Höhe des Anzeigefensters
- rechte Zeile → 60 Pixel Höhe
- mittlere Zeile → Rest

- Frames zu einem Frameset definieren

- Definition von Framesets → Aufteilung des Anzeigefensters bestimmen
- Definition von Frames → Inhalte in den einzelnen Framefenstern bestimmen

- Bsp.

```
<frameset cols="250,*">
  <frame src="verweise.htm" name="Navigation">
  <frame src="startseite.htm" name="Daten">
  <noframes>
    Ihr Browser kann diese Seite leider nicht anzeigen!
  </noframes>
</frameset>
```

- `<frame src=" ... ">` → definiert den Inhalt der im Fenster angezeigt werden soll
- jede "Zelle" des Framesets wird hier ein Framefenster definiert
- `name=` → Namen des Framefensters
 - erforderlich um Verweise zu definieren, die in diesem Framefenster angezeigt werden sollen. (keine Leerzeichen, Sonderzeichen oder deutsche Umlaute)
 - Bsp. → die oben spezifizierte Datei `verweis.html`

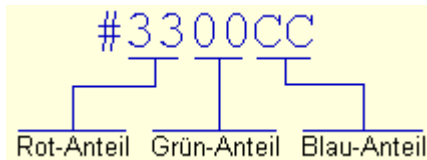
```
<h1>Navigation</h1>
<p>
<a href="kapitel1.htm" target="Daten">Kapitel 1<br>
<a href="kapitel2.htm" target="Daten">Kapitel 2<br>
<a href="kapitel3.htm" target="Daten">Kapitel 3<br>
</p>
```

1.8 Positionierung von Elementen mit Containern

- HTML-Elemente über Container anordnen: Der neue Weg anstelle von Frames mit `<div>` und ``

1.9 Farbe in HTML-Seiten

- Grundsätzlich gibt es zwei Möglichkeiten, Farben in HTML zu definieren:
 - durch Angabe der RGB-Werte der gewünschten Farbe in Hexadezimalform (RGB = Rot/Grün/Blau-Wert der Farbe)
 - Browser-unabhängig, 16,7 Millionen Farben.
 - durch Angabe eines Farbnamens
 - Farbnamen umgehen die etwas schwierige Definition einer Farbe im Hexadezimal-Modus.
 - Derzeit nur 16 Farbnamen offiziell standardisiert. Weitere Farbnamen sind Browser-abhängig.
- hexadezimalen Farbsystem
 - Farbcode wird aus Angaben zu den drei Grundfarben **Rot**, **Grün** und **Blau** (RGB-Werte) zusammengestellt
 - Jede hexadezimale Farbdefinition ist 6stellig und hat das Schema: `#xxxxxx`
 - Die ersten beiden Stellen geben den Rot-Wert der Farbe an,
 - die zweiten beiden Stellen den Grün-Wert,
 - und die letzten beiden Stellen den Blau-Wert.



- 0 (entspricht dezimal 0)
 - 1 (entspricht dezimal 1)
 - ...
 - 9 (entspricht dezimal 9)
 - A (entspricht dezimal 10)
 - B (entspricht dezimal 11)
 - C (entspricht dezimal 12)
 - D (entspricht dezimal 13)
 - E (entspricht dezimal 14)
 - F (entspricht dezimal 15)
-
- Umrechnung in RGB-Farbwerte
 - 1. Stelle x 16 + 2. Stelle
 - Bsp.: 33 → 3 x 16 + 3 = 051
 - Bsp.: C8 → 12 x 16 + 8 = 200
 - Eine hexadezimale Ziffer kann also 16 Zustände haben.
 - Für jeden Farbwert (Rot, Grün, Blau) stehen 2 Ziffern zur Verfügung.

- Das macht 16 x 16 (= 256) mögliche Zustände pro Farbwert.

- Bsp.

```
<body bgcolor="#808080"> <!-- dunkelgrauer Dateihintergrund -->  
<table bgcolor="#00C0C0"> <!-- blaugrüner Tabellenhintergrund -->  
<hr color="#CC00CC"> <!-- violette Trennlinie -->
```

- Links

- HTML: <http://aktuell.de.selfhtml.org/>
- <http://www.planethtml.de/html/>
- Farben: www.colorblender.com