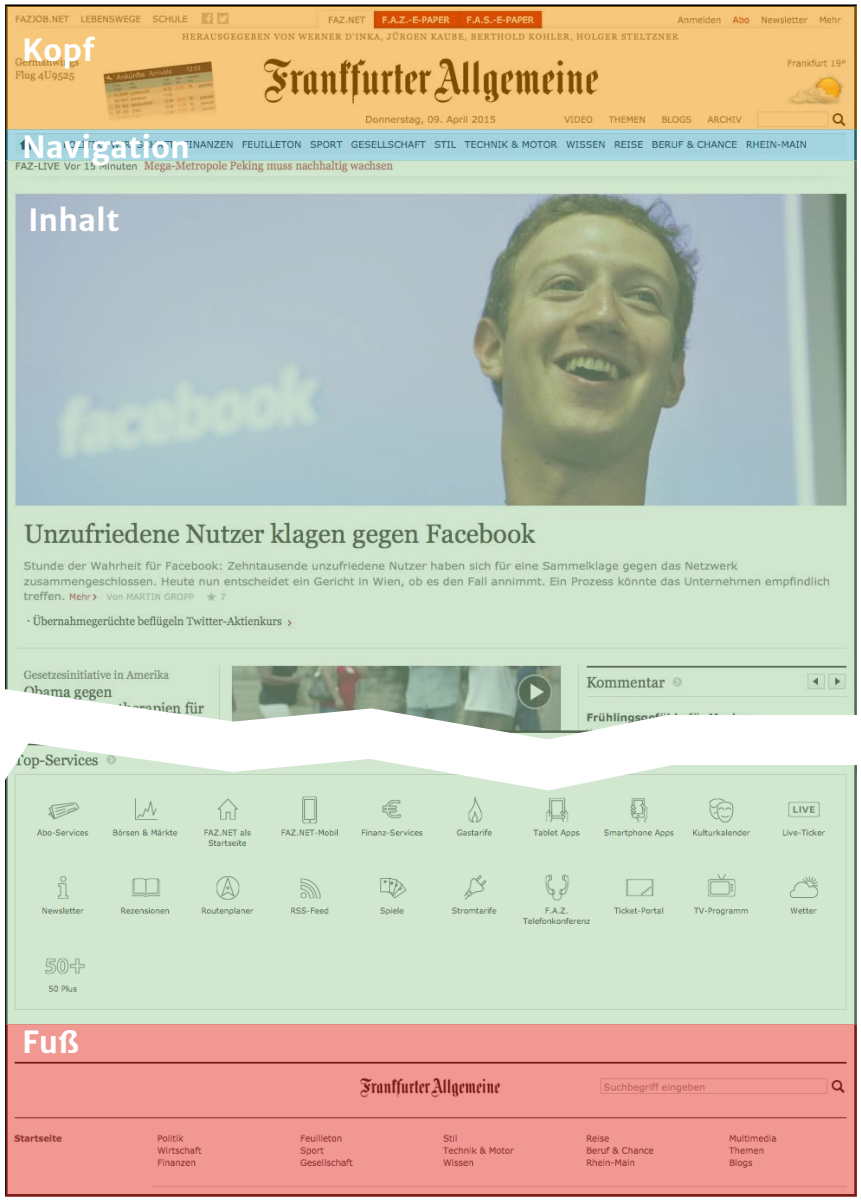
HTML

Was ist das ?

HTML = Hyper Text Markup Language

- ist eine Auszeichnungssprache,
  also KEINE Programmiersprache
- erstellbar mit jedem beliebigen Texteditor
- dient der Strukturierung einer Webseite
- wird vom Browser interpretiert
- dient NICHT zur Gestaltung einer Seite
  (strikte Trennung von Inhalt und Gestaltung)
- Schnittstelle zu Javascript und CSS
- wird von Browsern unterschiedlich interpretiert



Tags

Beispiel:

<h1> Das ist eine Überschrift **</h1>**

Standalone Tags

Beispiel:

** ** Das hier ist **
** ein Beispieltext ** **

Verschachtelung der Elemente

Beispiel:

** ** Hier geht es zurück zur **** Startseite **. **

Attribute

- nur in Start- sowie Standalone-Tags

Beispiel:

<a href="index.html">

Wertzuweisung

Beispiel:

```
<input type="button">
```

Freie Wertzuweisung mit Konvention

Beispiel:

```
<style type="text/css">
```

Freie Wertzuweisung ohne Konvention

Beispiel:

```
<input type="button" value="Senden">
```

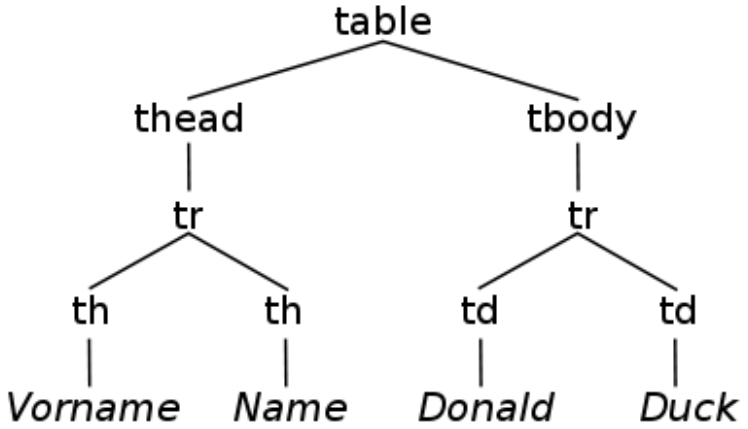
Alleinstehende Attribute

Beispiel:

```
<input type="checkbox" checked>
```

DOM – Die hierarchische Struktur von HTML – Dokumenten:

```
<table>
  <thead>
    <tr>
      <th>Vorname</th>
      <th>Name</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Donald</td>
      <td>Duck</td>
    </tr>
  </tbody>
</table>
```



Grundlegender Aufbau:

```
<!DOCTYPE html>
<html>
      <head>
            <meta charset="UTF-8">
            <title> Startseite </title>
      </head>
      <body>
            <h1> Überschrift </h1>
            <p> Text </p>
      </body>
</html>
```

Sinnvolle Teilbereiche durch Standardkontainer:

```
<header>   </header>
<nav>      </nav>
<section>  </section>
<article>  </article>
<footer>   </footer>
```

Kommentare:

```
<!-- Ein Kommentar der vom Parser nicht gelesen wird -->
```

Metaangaben

= nützliche Anweisungen für Browser, Webserver oder Suchmaschinen (z.B.: Angaben zum Autor oder zum Inhalt der Seite).

Zeichenkodierung:

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

Seitenbeschreibung:

<meta name="description" content="Portfolio von Max Mustermann">

Autor:

<meta name="author" content="Max Mustermann">

Suchmaschine:

<meta name="robots" content="noindex, nofollow">

Der Seiteninhalt wird nicht ausgelesen und die Links (auch zu Unterseiten werden nicht verfolgt).

Textstrukturierung

<section>...</section>
Zusammengefasster Bereich, auch mehrere Section-Tags möglich

<header>...</header>
Kopfleiste, darf keinen header/footer enthalten

<footer>...</footer>
Abschlussleiste, darf keinen header/footer enthalten

<nav>...</nav>
Bereich für die Navigation meist in Form einer ungeordneten Liste

<aside>...</aside>
Seitenleiste oder Bereich für Marginaltext

...
markiert einen allgemeinen Bereich (Zur CSS Formatierung gedacht)

<article>...</article>
Logisch definierter Bereich für Artikel (für Blogs, Benutzerkommentare, Magazinartikel, usw.)

Wird oft innen nochmal mit header und footer etc. unterteilt um z.B. den Autor eines Benutzerkommentars im footer einzuschließen

<address>...</address>
Angaben zum Verfasser eines Artikels oder ähnliches

Anmerkung: Optisch erzeugen diese Elemente lediglich eine neue Zeile im Textfluss!

Überschriften

<h1>...</h1> Überschrift 1. Ordnung
<h2>...</h2> Überschrift 2. Ordnung usw.

Anstatt immer das h1-Tag zu verwenden und dann die Schriftgröße bzw. den Schriftstil zu verändern, werden für Überschriften unterschiedlicher Rangordnung die oben stehenden Tags benutzt.

So entsteht eine einheitliche Hierarchie unter den Überschriften im HTML.

<hgroup>...</hgroup>

Dient der Zusammenfassung mehrerer Überschriften. Da oft mehrere Überschriften nacheinander kommen, lohnt es sich diese zu gruppieren.

Geordnete und ungeordnete Listen

Geordnete Listen
sind Listen, bei denen jeder Listeneintrag mit einer Nummer versehen wird.

```
<ol reversed>          zählt von unten nach oben (durch Attribut reversed)
    <li>...</li>       wird durchnummeriert und darf andere HTML-Elemente enthalten
    <li>...</li>
</ol>
```

Ungeordnete Listen
listen Einträge nacheinander auf ohne sie durchzunummerieren

```
<ul>
    <li>...</li>       wird nicht durchnummeriert und darf andere HTML-Elemente enthalten
    <li>...</li>
</ul>
```

Anmerkung: Listen werden oft ineinander verschachtelt, z.B. für Dropdown-Menüs

```
<ul>
    <li>
        <ol start="2">          beginnt bei 2 zu zählen
            <li value="4">...</li>      bekommt trotzdem, aufgrund des Attributs,
                                        die Nummer 4
        </ol>
    </li>
    <li>...</li>
</ul>
```

Textauszeichnungen

... dienen der Hervorhebung von Textpassagen mit besonderer Bedeutung.

Es geht hierbei nicht um die Gestaltung des Textes, das ist Sache von CSS. Es geht vielmehr darum, den Text mit semantisch wichtigen Informationen zu versehen und die **Grundlage für die spätere Textformatierung zu schaffen.**

Beispiele:

Stilistische Hervorhebung:
...

Wörtliche Rede:
<q>...</q>
mit dem Attribut cite="http://...." kann eine Quelle für ein Zitat angegeben werden

Tabellen

Besteht aus Zeilen- und Spalten-Tags die von einem Table-Tag umschlossen werden.

```
<table>
        <thead>
                <tr>  Tabellenzeile, darf nur th-Tags beinhalten
                        <th>...</th>  Tabellenüberschrift, Text und Inhalt ohne Zeilenumbruch
                        <th>...</th>
                </tr>
        </thead>
        <tfoot>
                <tr>
                        <td>...</td>  Tabellenzelle mit Inhalt, auch weitere HTML-Elemente
                        <td>...</td>  Selbst weitere Tabellen dürfen verschachtelt werden
                </tr>
        </tfoot>
        <tbody>
                <tr>
                        <td>...</td>  Tabellenzelle mit Inhalt, auch weitere HTML-Elemente
                        <td>...</td>  Selbst weitere Tabellen dürfen verschachtelt werden
                </tr>
        </tbody>
</table>
```

Die Tabelle passt sich standardmäßig an die Breite des Inhalts an.
Sinnvoll ist es, die Anzahl der Spalten in jeder Zeile gleich zu halten, aber nicht zwingend erforderlich.

Die Unterteilung in head, foot und body ist nicht zwingend erforderlich, jedoch muss die Reihenfolge beachtet werden!
Reihenfolge: 1. thead, 2. tfoot, 3. tbody

Das Universal-Attribut id

Mithilfe des id-Attributs kann man HTML-Elementen sogenannte **Ids also einmalige Bezeichnungen** zuweisen über die diese Elemente dann z.B. für CSS Zwecke adressiert werden können. Das id-Attribut kann nahezu jedem HTML-Element zugewiesen werden, muss jedoch in einem HTML-Dokument einmalig sein.

```
<p id="besonderer_textabschnitt">...</p>
```

Erlaubte Zeichen: Buchstaben, Zahlen und _ (keine Sonderzeichen!)

Das Universal-Attribut class

Entsprechend der Funktion der id kann HTML-Elementen auch das class-Attribut zugewiesen werden. Ein class-Attribut kann jedoch mehrfach im Dokument verwendet werden!

```
<p id="besonderer_textabschnitt">
    Hier steht <span class="important">
    ein Text </span> der <span
    class="important"> mehrfach </span>
    an Stellen gleich formatiert <span
    class="important">hervorgehoben </
    span> werden soll.
</p>
```

Erlaubte Zeichen: Buchstaben, Zahlen und _ (keine Sonderzeichen!)

Hyperlinks

Hyperlinks bestehen grundlegend aus einem <a> Element und einem Attribut für das Verweisziel.

```
<a href="startseite.html">...</a>
```
Verweis auf eine andere Seite im aktuellen Projekt

```
<a href="unterseiten/aktuelles.html">...</a>
```
Verweis auf eine Seite in einem anderen Ordner

```
<a href="http://www.google.de">...</a>
```
Link zu einer externen Seite

TIPP:
Auch andere Inhalte, neben Text, sind erlaubt. So z.B. Bilder oder andere HTML-Elemente.

Links zu Ressourcen

Eine Notation um Dateien beim Anklicken zum Download anzubieten existiert in HTML nicht. Lediglich der Browser sollte bei bestimmten Dateiendungen diese Funktion bieten. Vorallem das ZIP-Format (.zip) hat sich hier als Standard herausgebildet.

```
<a href="datei.zip" type="application/zip">
    Download der Datei
</a>
```
Das Type-Attribut gibt einen Hinweis für den Browser auf den Datei-Typ.

Grundsätzlich ist es möglich einen Verweis auf jede beliebige Datei zu erstellen. Je nach Browser kann diese Datei dann sogar direkt im Webbrowser angezeigt werden, z.B. PDF-Dateien.

Die Verwendung des Type-Attributs ist zwar sinnvoll aber nicht zwingend erforderlich.

E-Mail Links

Hierdurch können direkt Mails an E-Mailadressen gesendet werden. Jedoch hängt es vom Webbrowser und evtl. installierten PlugIns ab ob das funktioniert.

```
<a href="mailto:max.mustermann@web.de"
title="max.mustermann@web.de">
        Mail an Max
</a>
```
mailto: wird benötigt

Es ist sinnvoll die gesamte E-Mailadresse nochmal zu hinterlegen, um es dem User zu ermöglichen auch selbst eine E-Mail zu verfassen.

ACHTUNG: Das hinterlegen von E-Mailadressen im Internet führt generell zu sehr viel mehr Spam, daher ist ein sehr guter Spamfilter nötig!

Links und Anker

Da HTML-Dokumente auch sehr umfangreich werden können, gibt es so genannte Anker, die direkt zu bestimmten Stellen auf einer Seite springen ohne, dass der Benutzer Scrollen muss.
Beim Anker handelt es sich um ein normales a-Attribut, jedoch ohne ein href-Attribut. Stattdessen wird eine id vergeben zu der dann verlinkt wird.

Kapitel 1

Um dann zu einem solchen Anker springen zu können wird erneut ein a-Attribut verwendet.

Zu Kapitel 1

Auch Anker innerhalb anderer HTML-Dokumente sind adressierbar.

Zu Kapitel 1 der Anleitung

Pixelgrafik

Um Pixelgrafiken einzubinden müssen diese nur referenziert werden.
Geeignete Formate :
GIF, JPEG und PNG Dateien.

Es sollte immer darauf geachtet werden, dass die Dateigröße möglichst gering ist um lange Ladezeiten zu verhindern.

<img src="bild.jpg" alt="Bild" width="240" height="180">
img erzeugt keine neue Zeile

Der Text des alt-Attributs wird angezeigt, wenn die Bilddatei nicht gefunden wurde. Anstelle des Bildes wird dann ein Rahmen in der per width- und height-Attribut angegebenen Größe dargestellt. Das alt Attribut ist auch für die Barrierefreiheit sinnvoll, jedoch lohnt es sich nicht dort auführliche Erklärungen zu platzieren.

Durch die Angabe der Höhe und Breite wird es für den Browser einfacher die Bilder schnell einzubauen, da er nicht erst die Größeninformationen aus der Datei auslesen muss. Man sollte die original Größe verwenden, zwar kann der Browser die Bilder strecken, dies verlangsamt jedoch den Bildaufbau.
Es ist auch möglich Bilder direkt von anderen Internetseiten zu referenzieren.
<img src="http://www.example.org/grafiken/datei.gif"> **Vorsicht Urheberrecht!**

Audio- und Video-Ressourcen

Bei Audio Ressourcen unterstützen die Browser unterschiedliche Formate:
IE: MP3, WAV
FF: OGG, WAV
GC: MP3, OGG
Safari: MP3, WAV, AU/SND, AIF
Opera: OGG, WAV, AU/SND

Daher ist es nötig verschiedene Versionen zu hinterlegen. Mindestens MP3 und OGG.

<audio preload controls> Das src-Attribut kann hier verwendet werden (bevorzugtes Format)
    <source src="datei.mp3">
    <source src="datei.ogg">
</audio>

Attribute:
controls Der Browser zeigt Steuerelemente an
autoplay Startet automatisch
preload Lädt bereits beim Seitenaufbau
loop Die Datei wird immer wieder wiederholt

Bei Video-Ressourcen läuft es ähnlich ab.
Hier muss jedoch, um alle Browser zu unterstützen die Datei als OGV (OGG-Video) und als MPEG-4 hinterlegt sein.

<video src="datei.mp4" width="640" height="480" controls preload poster="datei.jpg">
  <source src="datei.ogv" type="video/ogg">
  <source src="datei.mp4" type="video/mp4">
</video>

Attribute:
die gleichen wie bei audio
poster Vorschau-Grafik
width, height Größe des Videos, es sollten die tatsächlichen Maße sein

Formulare

Mit Formularen wird dem Anwender die Möglichkeit geboten, Daten einzugeben und an den Server zu schicken. Hierbei kann es sich um Persönliche Daten des Nutzers, um eine Bestellung in einem Onlineshop oder auch um „user generated content“ handeln.
Das gesamte Formular befindet sich innerhalb des öffnenden und des schließenden form-Tags.

Attribute:

action	gibt an, wohin das Formular beim Absenden geschickt werden soll
method	legt die Art der HTTP-Übertragung fest

```
<form action="verarbeitung.php"
method="post" name="Mein Formular"> ...
</form>
```

Beim Absenden wird ein php-Skript aufgerufen. Dieses Skript muss gleichzeitig zur Verarbeitung der Eingaben auch eine HTML-Seite zur Verfügung stellen die beim Absenden des Formulars aufgerufen wird.

HTTP-Request-Methoden

Beim Attribut method wird die Art der Übertragung festgelegt. Wird kein method Attribut verwendet wird **als Standard die GET-Methode** eingesetzt. In diesem Fall werden die eingegebenen Daten einfach an die bei action angegebene URL-Adresse angehängt.

method="get"
Beispiel:
http://www.beispiel.de/verarbeitung.php?name=Max+Muster&mail=max@example.de&text=Text

Diese URL erscheint beim Absenden des Formulars auch in der Adresszeile des Browsers. Die GET-Methode eignet sich nur für geringe Datenmengen.

Für größere Datenmengen und wenn man nicht will, dass die Daten in der Adresszeile angezeigt werden, eignet sich **die POST-Methode.**

method="post"

Es ist nicht zwingend erforderlich ein action bzw. ein method-Attribut zu verwenden. Wenn Daten direkt via Javascript verarbeitet und nicht gespeichert werden sollen, genügt <form></form>.

Zeichenkodierung

Mit dem Attribut **accept-charset="utf-8"** wird die Übertragung der eingegebenen Daten mit UTF-8 kodiert. Dies erlaubt Sonderzeichen oder spezielle Schriften in den Eingabefeldern.

Strukturierung

Größere Formulare sollten zur besseren Übersichtlichkeit strukturiert werden. Dafür bietet HTML das fieldset-Tag. Zwischen **<fieldset>** und **</fieldset>** können beliebige Teile des Formulars notiert werden. Mit **<legend></legend>** können Gruppenüberschriften übergeben werden. Innerhalb des legend-Tags sind auch weitere HTML-Tags wie Auszeichnungen erlaubt. Das **label**-Tag ermöglicht das Beschriften von Formularfeldern. Dabei gibt es zwei unterschiedliche Herangehensweisen:

Explizite Variante:
<label for="formularfeld_id">Beschriftung:</label>
Das Formularfeld muss als id die gleiche Bezeichnung haben wie das for-Attribut des label-Tags.

Implizite Variante:
<label>Beschriftung:
<!-- Formularelement --></label>
Das Formularelement auf das sich das label-Tag bezieht, muss sich zwischen dem öffnenden und dem schließenden label-Tag befinden.

Formularfeld-Typen

Das Aussehen von Eingabefeldern richtet sich standardmäßig nach dem Betriebssystem und Browser des Anwenders. Allerdings lassen sich viele Eingabelemente mit CSS stark manipulieren.

```
<input type="text" name="vorname" value=
"Name eingeben" size="32" maxlength="32">
```

Einzeilige Eingabefelder

Mit dem Standalone-Tag input lassen sich verschiedene einzeilige Eingabefelder definieren.
Attribute:

type	legt fest um welche Art von Eingabe es sich handelt, z.B. Text.
name	dem Eingabefeld wird eine Bezeichnung zugewiesen. Über diese Bezeichnung wird später bei der Auswertung das Feld identifiziert.
size	legt die angezeigte Länge des Feldes in Zeichen fest
maxlength	legt fest wie viele Zeichen eingegeben werden können
value	gibt an welcher Text als Vorschau angezeigt wird.

TIPP: Bei Verwendung von type="password" statt type="text" werden die eingegebenen Zeichen nur als Punkte angezeigt.

Radiobuttons und Checkboxen

Radiobuttons sind beschriftete Knöpfe von denen jeweils immer nur einer ausgewählt sein kann.

```
<input type="radio" name="radiobutton"
value="auswahl1">Auswahl1<br>
<input type="radio" name="radiobutton"
value="auswahl2">Auswahl2</br>
```

Checkboxen sind kleine Vierecke die angekreuzt werden können. Hierbei ist eine Mehrfachauswahl möglich.

```
<input type="checkbox" name="checkbox"
value="auswahl1">Auswahl1</br>
<input type="checkbox" name="checkbox"
value="auswahl2">Auswahl2</br>
```

TIPP: Mit dem Attribut checked in einem input-Tag kann man eine Vorauswahl realisieren.

Schaltflächen (Buttons)

Es gibt 3 Arten von Schaltflächen:
- **button** zum auslösen von Javascript
- **submit** zum abschicken eines Formulars
- **reset** zum zurücksetzen eines Formulars

```
<input type="button" value="Beschriftung">
Alternativ: <button type="button></button>
```

```
<input type="submit" value="Beschriftung">
Ruft die unter action angegeben Adresse auf
```

```
<input type="reset" value="Beschriftung">
Das Formular wird zurückgesetzt
```

Auswahllisten

... geben eine vordefinierte Auswahl an Einträgen an. Der Ausgewählte wird dann an das bei action im form-Tag angegebene Script geschickt.

```
<select name="Beilagen">
        <optgroup label="Beilage">
        definiert eine Gruppe mit Überschrift
                <option>Salat</option>
                legt die einzelnen Auswahl-
                elemente fest
                <option>Pommes</option>
        </optgroup>
</select>
```

Mit **size="1"** wird nur ein Eintrag gezeigt. Nach dem Anklicken werden die Anderen Optionen angezeigt. Das Attribut **multiple** lässt eine Mehrfachauswahl zu.

Mehrzeilige Eingabefelder

Ein großes, mehrzeiliges Eingabefeld. Die Attribute cols und rows legen die Größe in Zeichen fest, wrap="soft" bewirkt einen automatischen Zeilenumbruch. Bei wrap="hard" erfolgt kein Zeilenumbruch.

```
<textarea name="eingabefeld" cols="50"
rows="15" maxlength="100000"
wrap="soft">
Hier steht ein Vorschau-Text
</textarea>
```