3. Strukturierung von Web-Seiten (HTML)

Lernziele:

- Aufbau eines HTML-Dokumentes
- Block-Elemente und Inline-Elemente
- HTML-Entitäten und globale Attribute
- Einbinden von Multimediainhalte
- Kompositionsdiagramme und Sitemaps
- Erstellen von iFrames und Formularen
- Wichtige Beispiele für <meta>-Information

Überblick:

3.1 HTML- Basiswissen

WHATWG

HTML

Living Standard — Last Updated 24 November 2021

Geschichte von HTML

- 1990/1: Vorstellung der ersten Versionen von HTML.
- 1995: HTML 2.0 (RFC-Standard), basiert auf DTD.
- 1998: HTML 4.0 führt Cascading Stylesheets CSS ein.
- 1999: HTML 4.01 offizielle W3C Recommendation.
- 2000: XHTML 1.0. Reformulierung von HTML4 auf Basis von XML
- 2004: Gründung einer neuen Arbeitsgruppe Web Hypertext Application Technology Working Group (WHATWG), von Firmen die mit der schleppenden Entwicklung der W3C unzufrieden war.
- 2008: HTML5 W3C Recommendation basiert auf Zusammenarbeit von W3C mit WHATWG. Neue Struktur- und Multimedia-Elemente. Orientierung an aktuellen Browser-Technologien und der entstandenen „Tag-Soup“.

2011 Erneute Trennung von W3C und WHATWG

2017 W3C veröffentlicht HTML 5.2 –Recommendation

2018 W3C veröffentlicht HTML 5.3 –Recommendation

2019 WHATWG und W3C einigen sich auf die gemeinsame Entwicklung eines HTML – Living Standard

2021 HTML - Living Standard der WHATWG; W3C veröffentlicht Zwischenstände als Recommendation.

WHATWG

HTML

Living Standard — Last Updated 18 January 2024

https://html.spec.whatwg.org/multipage/

HTML und Webseiten

- Webseiten werden in HTML geschrieben.
- HTML (Hypertext Markup Language) ist wie XML eine Auszeichnungssprache (Markup Language), mit der Textdokumente erstellt und strukturiert werden.
- Die meisten XML-Regeln sind von daher auch für HTML gültig.
- Da HTML-Dokumente aus reinem Text bestehen, benötigen Sie keine spezielle Entwicklungs-Software, um Webseiten zu erstellen. Ein beliebiger Texteditor ist ausreichen (Beispiel: `notepad.exe`).
- Ein Browser kann ein HTML-Dokument analysieren (HTML-Parser) und zur Anzeige bringen (Rendering-Engine).

- Die HTML-Textdateien erhalten die Endung `.html`
- MIME-Type: `text/html`
- MIME steht für Multipurpose Internet Mail Extensions und ist ein Standard, der ursprünglich entwickelt wurde, das Simple Mail Transfer Protocol (SMTP) also E-Mailing mit nicht-textbasierten Inhalten zu unterstützen.
- Ein MIME-Typ besteht aus zwei Teilen: Typ/Subtyp.

```
text/plain: Unformatierter Text.
text/html: HTML-Inhalt.
```

- Der MIME-Typ und das gewählte Encoding wird durch den HTTP-Header Content-Type übermittelt.

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
```

HTML-Tags

- Ein HTML-Element besteht aus einem öffnenden Tag, einem schließenden Tag und dem Inhalt zwischen beiden.

```
<element attibutname="Wert">
  Das ist ein Textinhalt.
  <child-element>
    Das ist ein Child-Element
  </child-element>
  Das ist ein Textinhalt.
</element>
```

- Die Tags übermitteln dem Browser die Strukturinformation zum Aufbau der Webseite und den Inhalt. Die Tags werden selbst nicht anzeigt.
- HTML-Attribute sind zusätzliche Informationen oder Eigenschaften, die einem HTML-Element hinzugefügt werden, um dessen Verhalten, Darstellung oder Funktion zu beeinflussen.
- HTML-Tags und -Attribute sind nicht case sensitive (im Gegensatz zu XML).
- Best Practise in HTML: alles klein schreiben.
- Beispiel:
  - <article> : Signalisiert den Beginn eines Artikels.
  - </article> : Signalisiert das Ende eines Artikels
  - : Signalisiert den Beginn eines Absatzes.
  - : Signalisiert das Ende eines Absatzes

```
<article>
  Dieser Artikel beschreibt HTML.
  <p>HTML wird verwendet um…</p>
</article>
```

Prolog und HTML-Dokument

Das Bild auf der rechten Seite zeigt das HTML-Grundgerüst einer Webseite.

- Prolog: DOCTYPE-Definition teilt mit, dass der HTML Living Standard verwendet wird

  `<!DOCTYPE html>`

- Dokument: <html> ... </html>
  - Beginn und Ende des HTML-Dokuments wird mittels <html> und </html>.
  - Ist die Root von allen HTML-Dokumenten.
  - Verwendete Sprache des Dokuments wird über das Attribut lang spezifiziert:

    `lang="de"`
  - "de" ist ein standardisiertes Sprachkürzel (ISO 639-1)

```
<!DOCTYPE html>

<html lang="de">
```

Dokumentenkopf

```
  <head>
    <meta charset="utf-8"/>
    <title>Wetterdaten</title>
    <link rel="stylesheet" href="css/style.css"
          type="text/css"/>
    <link rel="shortcut icon" href="favicon.ico"
          type="image/x-icon" />
  </head>
```

Dokumentenrumpf

```
  <body>
    <p>Auf dieser Webseite können Sie die
       neuesten Wetterdaten einsehen.
    </p>
  </body>

</html>
```

HTML Dokumentenkopf

- Jedes HTML-Dokument besitzt einen Dokumentenkopf

```
<head> … </head>
```

- Im Dokumentenkopf befinden sich das HTML-Dokument beschreibende Informationen, die vom Browser, einer Bedienhilfe (z.B.: Screenreader) oder einer Suchmaschine ausgewertet werden (SEO: Search Engine Optimization).
- Was zwischen <head> und </head> steht, wird vom Browser auf der Webseite nicht angezeigt.
- Angabe der verwendeten Zeichencodierung

```
<meta charset="utf-8"/>
```

- Angabe des Dokumententitels. Der Titel wird in den Bookmarks gespeichert und als Überschrift in den Suchergebnissen von Search-Engines angezeigt.

```
<title>Wetterdaten</title>
```

```
<!DOCTYPE html>

<html lang="de">
```

Dokumentenkopf

```
  <head>
    <meta charset="utf-8"/>
    <title>Wetterdaten</title>
    <link rel="stylesheet" href="css/style.css"
          type="text/css"/>
    <link rel="shortcut icon" href="favicon.ico"
          type="image/x-icon" />
  </head>
```

Dokumentenrumpf

```
  <body>
    <p>Auf dieser Webseite können Sie die
       neuesten Wetterdaten einsehen.
    </p>
  </body>

</html>
```

HTML Dokumentenkopf

- Das HTML `<link>`-Element beschreibt die Beziehung zwischen dem aktuellen Dokument und einer externen Ressource an.
  - `rel`-Attribut: Beziehung (relationship) der Ressource zum Dokument.
  - `href`-Attribut: Hyper-Reference verweist auf die gewünschte Ressource.
- Beispiele:
  - Verweis auf ein externes CSS-Stylesheet im Unterverzeichnis `css`
  - Verweis auf ein Favicon[1]-Icon für die Repräsentation der Webseite in Browser-Tabs oder Bookmarks.

[1]**Favicon**: "**Fav**orite **icon**."

```
<!DOCTYPE html>

<html lang="de">
```

Dokumentenkopf

```
  <head>
    <meta charset="utf-8"/>
    <title>Wetterdaten</title>
    <link rel="stylesheet" href="css/style.css"
          type="text/css"/>
    <link rel="icon" href="favicon.ico"
          type="image/x-icon" />
  </head>
```

Dokumentenrumpf

```
  <body>
    <p>Auf dieser Webseite können Sie die
       neuesten Wetterdaten einsehen.
    </p>
  </body>

</html>
```

Favicon

- Das favicon (Kofferwort: **Fav**orite und **Icon**) selbst ist eine kleine Grafik, die im Windows-Icon-Format (Dateien *.ico) vorliegt.
- Mediatype: image/x-icon
- Mögliche Größen sind 16x16, 32x32 oder 48x48 Pixel.
- Das favicon wird im <head>-Bereich einer Webseite mittels dem <link>-Element eingebunden:

```
<link rel="icon" type="image/x-icon"
href="favicon.ico">
```

HTML-Dokumenten-Body

- Der Dokumentenrumpfs wird mittels des Elements <body> definiert.

```
<body> ... </body>
```

- Nur Inhalte, die innerhalb des body-Elementes definiert werden, erscheinen auf der Webseite.
- Ein Browser ignoriert im HTML-Body
  - Zeilenumbrüche,
  - Leerzeichen und Tabs (Tabulator-Sprung).
- Zur Formatierung des Dokumentenrumpfs verwendet man weitere HTML-Elemente.

```
<!DOCTYPE html>

<html lang="de">
```

Dokumentenkopf

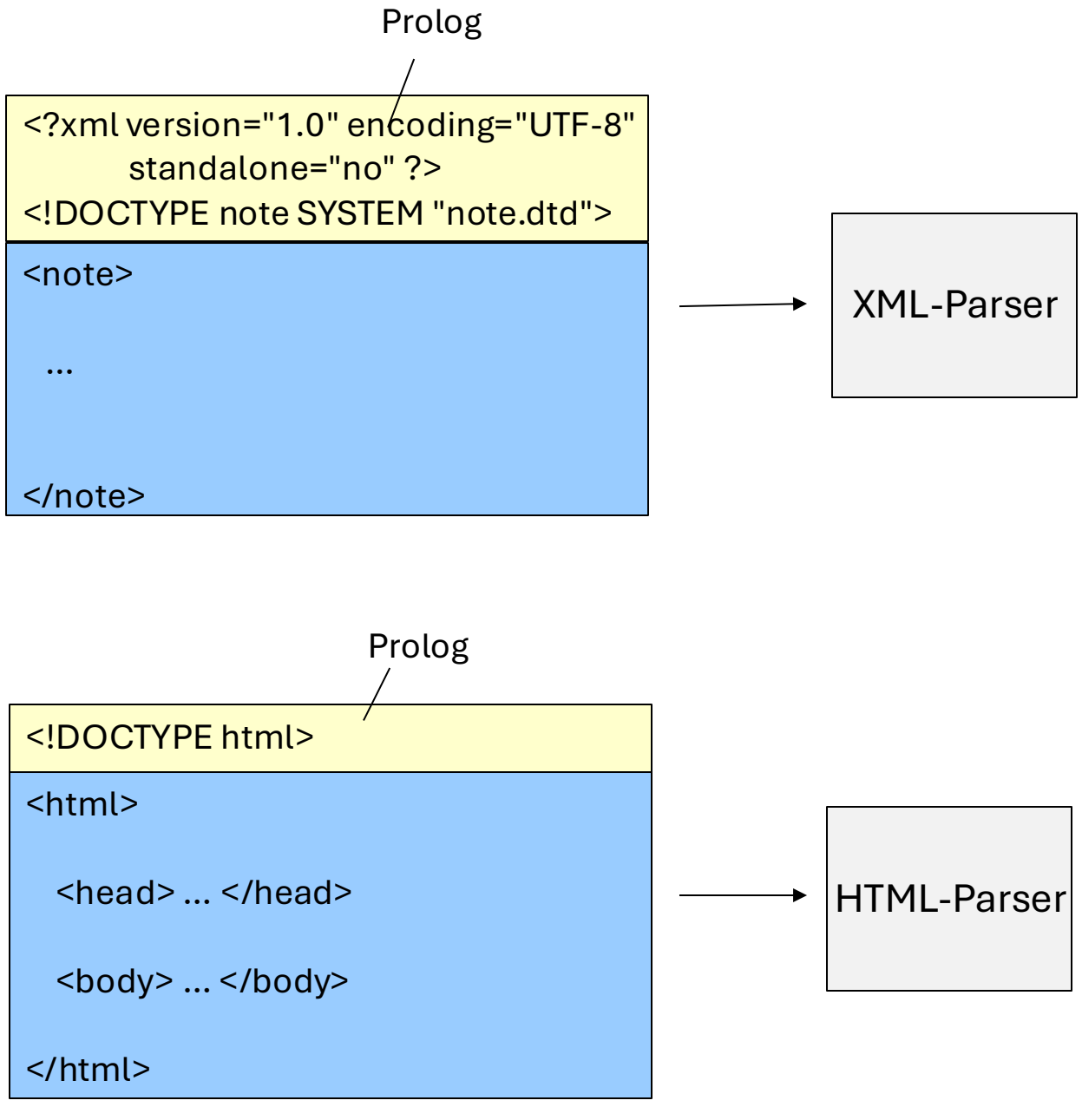
```
  <head>
    <meta charset="utf-8"/>
    <title>Wetterdaten</title>
    <link rel="stylesheet" href="css/style.css"
          type="text/css"/>
    <link rel="shortcut icon" href="favicon.ico"
          type="image/x-icon" />
  </head>
```

Dokumentenrumpf

```
  <body>
    <p>Auf dieser Webseite können Sie die
       neuesten Wetterdaten einsehen.
    </p>
  </body>

</html>
```

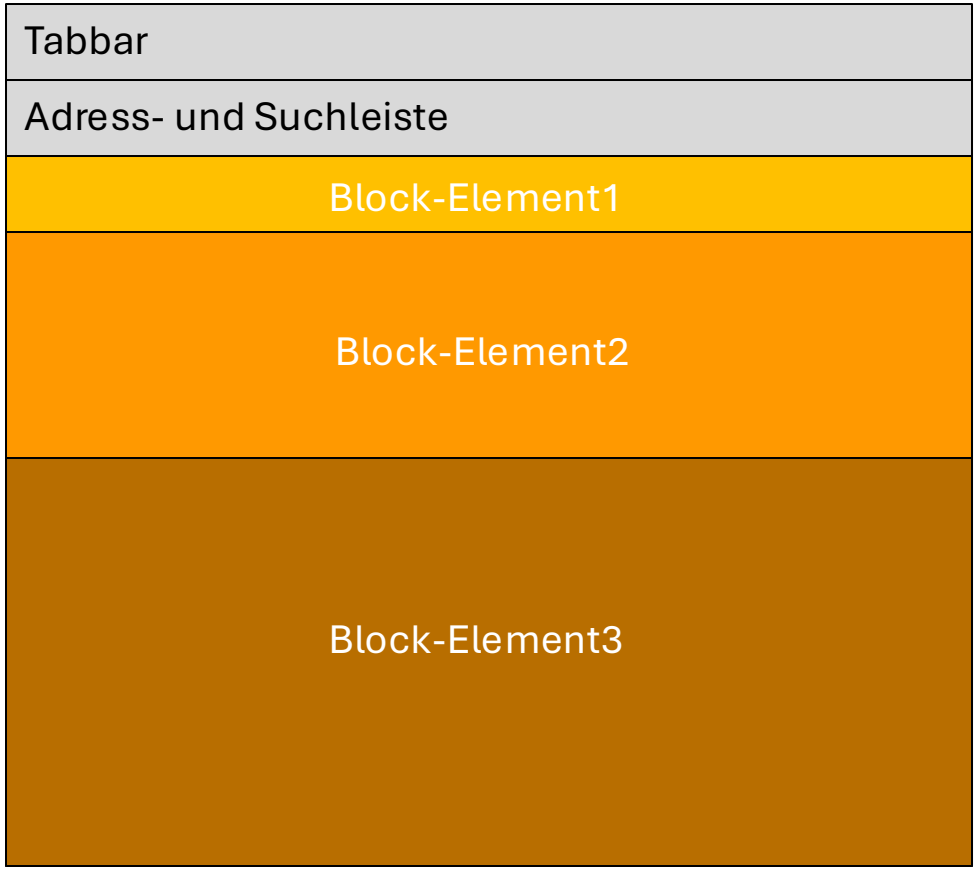
Vergleich XML und HTML

- Ein XML-und HTML-Dokument besitzen einen Prolog.
- Während in HTML der dokumentenunabhängige Living-Standard verwendet wird, kann pro XML-Dokument mittels einer eigenen Dokumententypdefinition (DTD) eine Syntax definiert werden.
- HTML besitzt zusätzlich einen Header-Bereich, der Meta-Daten über das Dokument enthält und einen Body-Bereich der die Strukturinformation und Inhaltsinformation enthält.



Blockelemente

- Mittels Blockelementen läßt sich der Body einer Webseite strukturieren.
- Blockelemente nehmen volle Fensterbreite ein und fügen einen Zeilenumbruch davor und danach ein.
- Bildlich: Ein Stapel an Boxen.

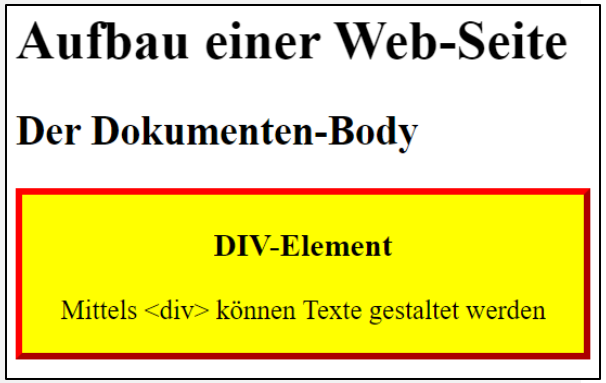


Blockelemente

- Beispiele für Blockelemente
  - Überschriften: 6 in der Schriftgröße abgestufte Überschriften
    <h1>, <h2>, ... <h6>
  - Textabsatz: <p>
  - Container: besitzt keine semantische Bedeutung, der Einsatzbereich von ist die Gruppierung von Elementen zum Zweck der Gestaltung durch CSS oder des programmatischen Zugriffs durch JS.

- Das im rechten Beispiel verwendete <style> - Element enthält CSS-Stilinformationen für das Dokument.
  - Das <style>-Element sollte nur für Testzwecke und nur im <head>-Element eines HTML-Dokumentes verwendet werden.

```
<head>
 <style>
   .divdemo {
      border: 5px outset red;
      background-color: yellow;
      text-align: center;
    }
 </style>
</head>

<body>
  <h1>Aufbau einer Web-Seite</h1>
  <h2>Der Dokumenten-Body</h2>
  <div class="divdemo">
    <h3>DIV-Element</h3>
    <p>Mittels &lt;div&gt; können Texte gestaltet
       werden</p>
  </div>
</body>
```



Listen

- Listen sind ebenfalls Blockelemente
- Es gibt geordnete Listen (ordered list <ol>) also mit Nummerierung
  `<ol> … </ol>`,
- und ungeordnete Listen (unordered list <ul>) also ohne Numerierung :
  `<ul> … </ul>`
- Das einzelne Listenelement wird mittels
  `<li> …<li>`
  getagt.
- Im Beispiel werden Entitäten verwendet, um im Text reservierte Zeichen verwenden zu können:
  `< >`

```
<body>

      <ol>
        <li>Überschriften: &lt; h1 &gt;</li>
        <li>Paragraphen: &lt; p &gt; </li>
        <li>Tabellen: &lt; table &gt;</li>
      </ol>

      <ul>
        <li>Überschriften: &lt; h1 &gt;</li>
        <li>Paragraphen: &lt; p &gt;</li>
        <li>Tabellen: &lt; table &gt;</li>
      </ul>

</body>
```

1. Überschriften: < h1 >
2. Paragraphen: < p >
3. Tabellen: < table >

- Überschriften: < h1 >
- Paragraphen: < p >
- Tabellen: < table >

Tabellen

- Auch Tabellen sind Blockelemente und werden mittels

  `<table> … </table>`

  erzeugt.
  - Mittels `<caption> … </caption>` wird eine Tabellenüberschrift erstellt werden.
  - Die Reihe (table row) einer Tabelle wird mittels dem `<tr> … </tr>` Tag angelegt.
  - Die Headerzelle (table header) einer Tabelle wird mittels dem `<th> … </th>` Tag angelegt.
  - Die Zellen (table data cell) innerhalb einer Reihe werden mit dem `<td> … </td>` Tag angelegt.

```
<table>
    <caption> <strong>Tabellen-Tags</strong> </caption>
     <tr>
       <th>Tag</th>
       <th>Description</th>
     </tr>
     <tr>
       <td>&lt;th&gt;</td>
       <td>Definition Header Zelle</td>
     </tr>
     <tr>
       <td>&lt;tr&gt;</td>
       <td>Definition einer Reihe</td>
    </tr>
    <tr>
       <td>&lt;td&gt;</td>
       <td>Definition einer Spalte</td>
    </tr>
  </table>
```

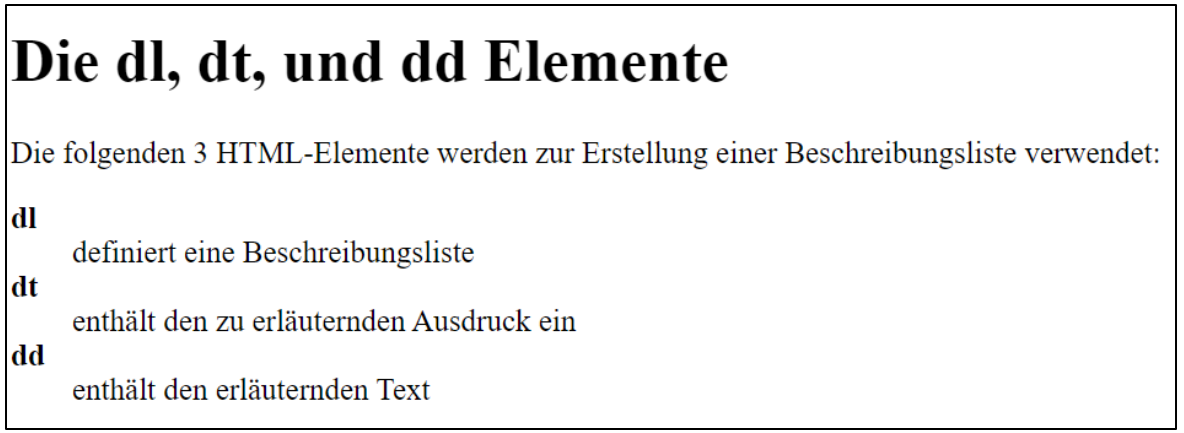
Tabellen-Tags

Tag	Description
<th>	Definition Header Zelle
<tr>	Definition einer Reihe
<td>	Definition einer Zelle

Beschreibungslisten

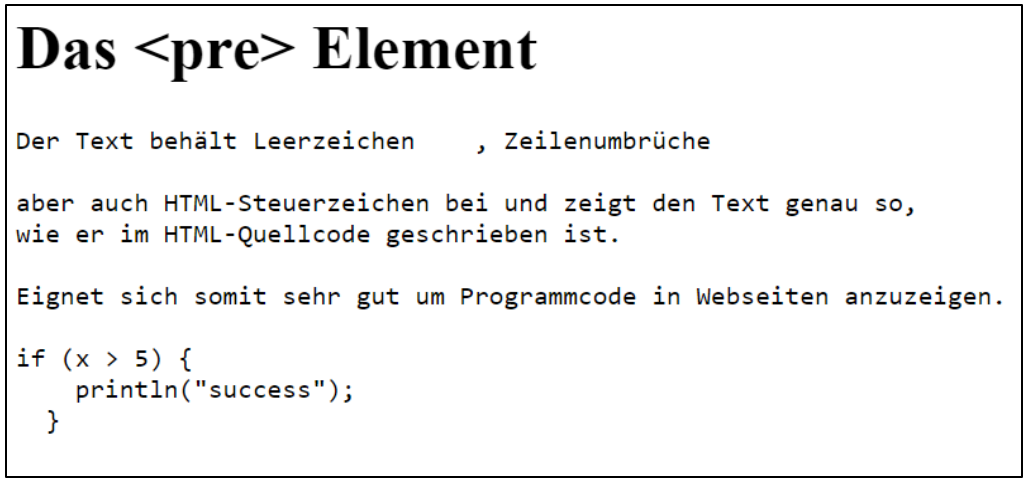
- `<dl>` (Description List) ist eine beschreibende Liste, deren Auzählungszeichen einem Begriff entsprechen, der pro Zeile definiert wird:
  - `<dl>` (Description List): Container für eine Beschreibungsliste
  - `<dt>` (Description Term): enthält den zu erläuternder Ausdruck
  - `<dd>` (Description Definition): enthält den erläuternden Text
- `<dl>` eignet sich für Glossare oder Literaturverzeichnisse.

```
<h1>Die dl, dt, und dd Elemente</h1>
<p>Die folgenden 3 HTML-Elemente werden zur Erstellung einer
Beschreibungsliste verwendet:</p>
<dl>
    <dt><strong>dl</strong></dt>
    <dd>definiert eine Beschreibungsliste</dd>
    <dt><strong>dt</strong></dt>
    <dd>enthält den zu erläuternden Ausdruck</dd>
    ...
</dl>
```



<pre> Tag

- Text in einem <pre>-Element wird in einer Schriftart mit fester Breite angezeigt (Monospace Schriftart).
- Das <pre>-Tag (preformated) behält Leerzeichen, Zeilenumbrüche aber auch HTML-Steuerzeichen bei und zeigt den Text genauso, wie er im HTML-Quellcode geschrieben ist.
- Eignet sich somit sehr gut um Programmcode (JavaScript, HTML, CSS) in Webseiten anzuzeigen.



Das <pre> Element

```
Der Text behält Leerzeichen    , Zeilenumbrüche

aber auch HTML-Steuerzeichen bei und zeigt den Text genau so,
wie er im HTML-Quellcode geschrieben ist.

Eignet sich somit sehr gut um Programmcode in Webseiten anzuzeigen.

if (x > 5) {
    println("success");
  }
```

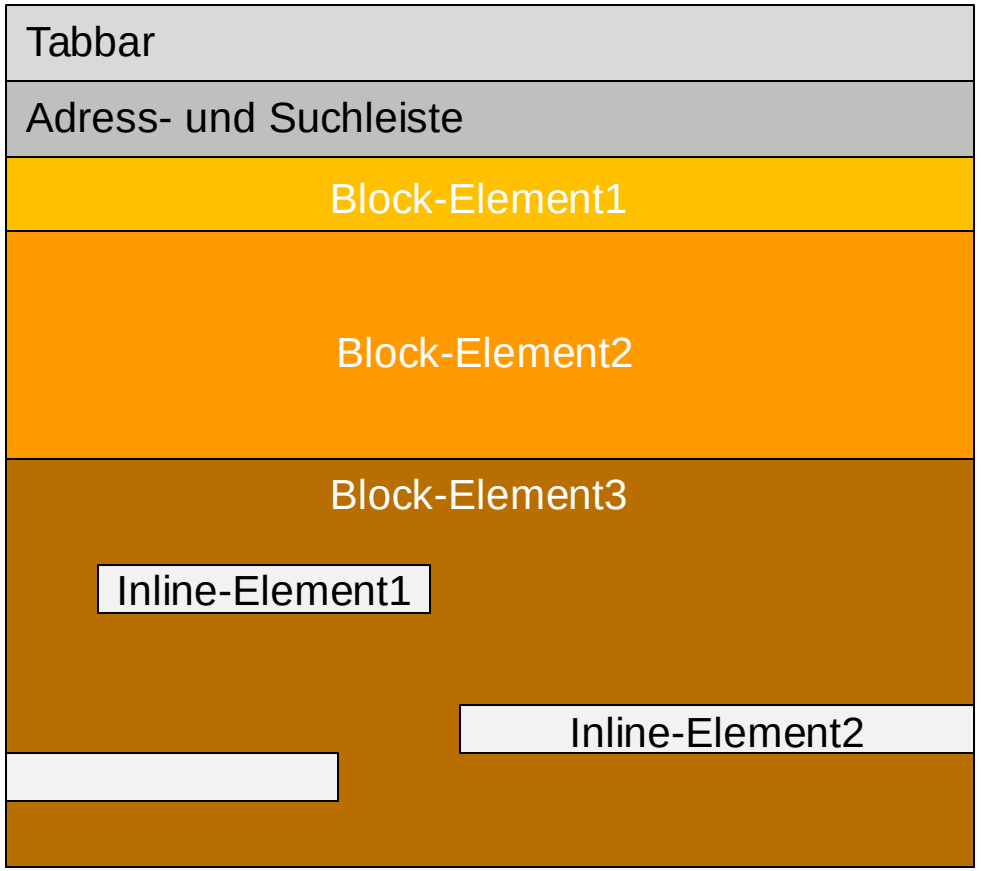
```
<h1>Das &lt;pre&gt; Element</h1>

<pre>
 Der Text behält Leerzeichen    , Zeilenumbrüche

 aber auch HTML-Steuerzeichen bei und zeigt den Text genau so,
 wie er im HTML-Quellcode geschrieben ist.

 Eignet sich somit sehr gut um Programmcode in Webseiten anzuzeigen.

 if (x > 5) {
     println("success");
  }

</pre>
```

Inlineelemente

- Inline-Elemente werden zur Formatierung des Textes in Block-Elementen verwendet und führen zu keinem Zeilenumbruch.
- Die Breite der Inline-Elemente richtet sich nach dem jeweiligen Inhalt.
- Der Inhalt selbst kann durch das Eltern-Blockelement umgebrochen werden.

Inline-Elemente dürfen Inline-Elemente enthalten.

Inline-Elemente dürfen keine Block-Elemente enthalten.



Inlineelemente

- Beispiele für Inlineelemente:
  - `<a>` : Link oder »Anker«
  - `<em>`: Kursive Schrift
  - `<strong>`: Fette Schrift
  - `<code>`: Darstellung als Monospace-Schriftart
  - `<kbd>`: Tastatureingaben hervorheben
  - `<samp>`: Hervorheben von Ausgaben eines Programmes
  - `<span>`: Ohne semantische Bedeutung, analog zu `<div>`
  - …

```
<p> Links sind ein <em>beliebtes</em> Mittel um
    <strong>Webseiten</strong> miteinander <br/> zu
    verknüpfen.
</p>
<code><a href="https://www.w3schools.com">Information zu
Links in der &lt; code &gt; - Darstellung</a></code>
```

```
<style>
    #span1 {
      color: #f00;
    }
    ...
</style>
```

rote Farbe

```
<p>Das Element <span id="span1">&lt;span&gt;</span> ist ein
<span id="span2">Inline Element</span> und dient zur Gestaltung
und programmatischen Veränderung von HTML-Seiten.</p>
```

Das Element ist ein Inline Element und dient zur Gestaltung und programmatischen Veränderung von HTML-Seiten.

Links sind ein *beliebtes* Mittel um **Webseiten** miteinander zu verknüfen.

Information zu Links in der <code>-Darstellung!

Hyperlinks: Externe URL

- `<a>`-Tag (anchor): definiert einen Hyperlink, der eine Web-Seite mit einer anderen verlinked.
- Das wichtigste Attribut des `<a>`-Elements ist das `href-Attribut`, das die URL des Links angibt.
- Standardmäßig werden Links in allen Browsern wie folgt angezeigt:
  - Ein nicht besuchter Link ist unterstrichen und blau.
  - Ein besuchter Link ist unterstrichen und lila.
  - Ein aktiver Link (Maustaste gedrückt) ist unterstrichen und rot.

```
<a href="https://www.w3schools.com"
   target="_blank" rel="noopener noreferrer">
   Visit W3Schools.com!
</a>
```

- Erklärung der Attribute
  - `target="_blank"`: Der Link öffnet die Webseite in einem neuen Tab
  - `rel="noopener"`: Attribut sorgt dafür, dass auf die aufrufende Seite via der JS-Funktion `window.opener` nicht zugegriffen werden kann.
  - `rel="noreferrer"`: verhindert, dass der Referrer (URL der ursprünglichen Seite) mitgeschickt wird.

Hyperlinks: Anker, Lokales Dokument

- Links können auf Bereiche innerhalb einer Web-Seite verweisen. Man spricht dann auch von sogenannten Anker-Links.
  Dabei wird das `Global-Attribut id` verwendet, das einem HTML-Element eine eindeutige Kennung zuweist und mit vorangestelltem `Hashtag #` referenziert

```
<h1 id="Kapitel1">Kapitel1: HTML-Elemente</h1>
<a href="#Kapitel1">Gehe zu Kapitel1</a>
```

- Links auf Web-Seiten im lokalen Filesystem des WWW-Servers, können mittels Dokumentenpfade der jeweiligen Datei referenziert werden.
  - Beispiel: Datei `index.html` befindet sich im Unterzeichnis `registrierung` in Bezug auf den Speicherort der aufrufenden Seite:

```
<a href="registrierung/index.html"> Registrierung
</a>
```

HTML-Dokumentenpfade

- Ein Dokumentenpfad definiert den Speicherort einer Datei in der Ordnerstruktur einer Website.
- Die Datei kann über den Dokumentenpfad in der Webseite verlinked werden.
- Beispiele für eingebundene Dateien
  - Webseiten: `<a href="Pfad" … >`
  - Bilder: `<img src="Pfad" … >`
  - JavaScripts: `<script src="Pfad" … >`
  - Stylesheets: `<link href="Pfad" … >`
- Es gibt absolute und relative Dateipfade.
- Bei absoluten Pfaden wird die vollständige URL der Datei angegeben

```
<img src="https://www.w3schools.com/images/picture.jpg">
```

- Bei relativen Dateipfaden bezieht man sich auf den Pfad der aktuellen Webseite oder auf das root-Verzeichnis „/“ der Web-Applikation (Verzeichnis der Homepage):

Pfad	Erklärung
`<img src="bild.jpg" …>`	Das Bild befindet sich im Verzeichnis der aktuellen Webseite.
`<script src="scripts/account.js" …>`	Das Script befindet sich im Unterverzeichnis “scripts” der aktuellen Webseite.
`<link href="/css/styles.css" …>`	Das Stylesheet befindet sich im Verzeichnis “css” unterhalb des root-Folders / der Web-Applikation.
`<a href="../product.html" …>`	Die Webseite “product.html” befindet sich im übergeordneten (“..”) Verzeichnis.

Multi-Media-Inhalte

- Text, Bilder, Video und Audio werden auch als »Multi-Media-Inhalte« bezeichnet und über ihren Dokumentenpfad (Ordner und Dateiname) in die Webseite gelinkt.

```
Text:  <a     href="produc/index.html" …> … </a>
Bild:  <img   src="images/handy.jpg" …> …   </img>
Video: <video src="video/animation.mp4" …   </video>
Audio: <audio src="audio/music.mp3" …>  …   </audio>
```

Bilder

- Bilder können sehr unterschiedliche Bedeutungen haben
  - Logos für die Wiedererkennung einer Produktmarke / einer Firma, …
  - Icons zur Erklärung der Bedeutung einer Schaltfläche
  - Fotos, zur emotionalen Bindung von Kunden(Story-Telling) oder zur Erklärung von Produkten
  - Abbildungen, zur Erläuterung von wiss. Inhalten

Bilder

- Bilder werden mit dem `<img> - Element` eingebunden bzw. referenziert.
- `<img>` ist ein Inline-Element.

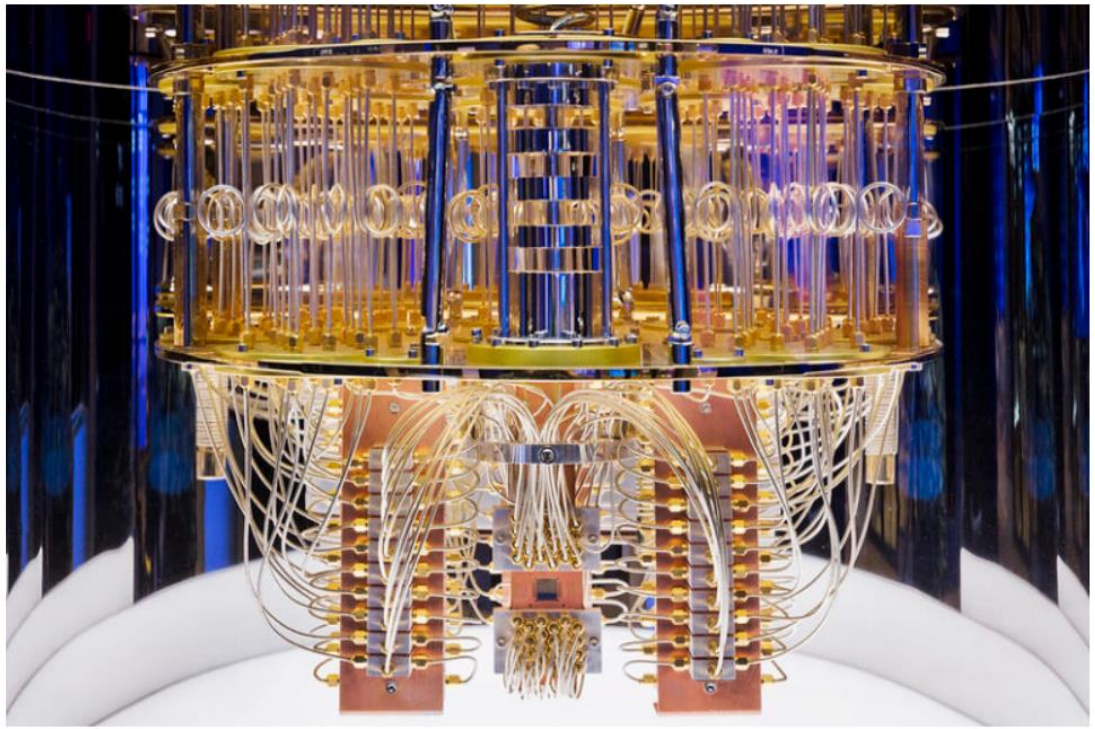
```
<img src="img/quantencomputer.jpg"
     alt="Hier finden Sie ein Bild"
     loading="lazy" height="400px"
     width="400px">
```

  - `src`: Pfad und Dateiname der Image-Datei
  - `alt`: Alternativer Text falls Image nicht geladen werden kann
  - `loading=eager|lazy;`
    `eager`: steht für das sofortige Laden und ist der Default-Wert;
    `lazy`: lädt alle Images im Anzeigebereich des Browsers sofort, den Rest nach Bedarf (Scrollen,...)

Einbinden von Multi-Media-Inhalten in Web Seiten

Bilder in HTML

Bilder, Video und Audio, sogenannte **Multi-Medien Inhalte** werden über den Pfad zu ihrem Ordner und ihren Dateinamen in die Webseite gelinkt. Wenn das Bild nicht geladen werden kann wird alternativ der Wert des Attribut alt angezeigt. Mittels dem loading Attribut kann die Ladegeschwindigkeit beeinflußt werden.



Abbildungen

- Wenn es sich um eine Abbildung handelt, also eine wissenschaftliche oder dokumentarische Darstellung, kann zusätzlich das Element `<figure>` verwendet.
- Das `<figure>` ermöglicht über das Child-Element `<figcaption>` das Anzeigen einer erklärenden Bildunterschrift.

```
<figure>
  <img loading="lazy" src="img/IBMQMcomp.jpg"
       alt="Quantencomputerbild" >
  <figcaption>Abb. 1: Ein Quantencomputer …
  </figcaption>
</figure>
```

Icons und Links

- Bilder wie zum Beispiel ein Icon, können in ein <a>-Element eingefügt werden, um ein klickbares Bild zu erhalten:

```
<a href="http://www.quantencomputer-info.de/">
   <img  src="img/ICON_QM.png"
         alt="Icon - clickme"
         height=40 width=40>
</a>
```

Bilder in HTML

Bilder, Video und Audio, sogenannte **Multi-Medien Inhalte** werden über den Pfad zu ihrem Ordner und ihren Dateinamen in die Webseite gelinkt. Wenn das Bild nicht geladen werden kann wird alternativ der Wert des Attribut alt angezeigt. Mittels dem loading kann die Ladegeschwindigkeit beeinflußt werden. 3 x 4 = 12

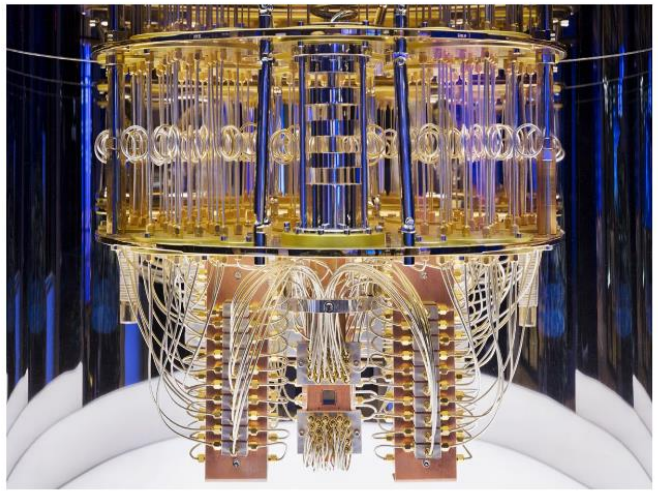


Abb. 1: Ein Quantencomputer der Firma IBM

Weitere Information zum Thema Quantencomputing erhalten Sie durch Klicken auf das folgende **Icon**

Audio und Video Elemente

- Das HTML-Element `<video>` bindet Videos in das Web-Dokument ein.
- `<video>` ist ein Blockelement
- `type="video/mp4"`
- Attribut `controls` blendet die Steuerelemente eines Mediaplayers ein.

```
<video src="video/animation.mp4" type="video/mp4"
preload="none" controls></video>
```

`preload="none"`: Video/Audio wird erst beim Abspielen geladen.

`preload="auto"`: Video/Audio wird beim Laden der Webpage geladen.

`preload="meta"`: Meta-Information (Dauer des Audios, ...) der Audio-Datei wird beim Laden der Webseite abgerufen

- Das HTML-Element `<audio>` wird verwendet, um Toninhalte
  - `<audio>` ist ein Inline-Element
  - `type="audio\mp3"`
  - `autoplay`: Audio/Video wird automatische abgespielt wenn die Datei geladen wurde.

```
<audio src="audio/music.mp3" type="audio/mp3"
controls preload="meta" autoplay></audio>
```

HTML-Entitäten

- Wie in XML gibt es in HTML vordefinierte Entitäten, um in HTML reservierte Zeichen auch in einem Text verwenden zu können (siehe Tabelle).
- So kann zum Beispiel das kleiner (<) und größer (>) Zeichen in einem HTML-Paragraphen wie folgt eingesetzt werden:

Die Zahl 3 ist > als 2 und < kleiner 4.“

```
<p> Die Zahl 3 ist &gt; als 2 und &lt; 4. </p>
```

- Die Non-Breaking Space Entity   fügt in einem Text Space ein Leerzeichen ein und sorgt dafür das innerhalb des Leerzeichens kein Zeilenumbruch erfolgt

3 x 4 = 12 soll nicht in der Formel umgebrochen werden

```
<p>3 x 4 = 12</p>
```

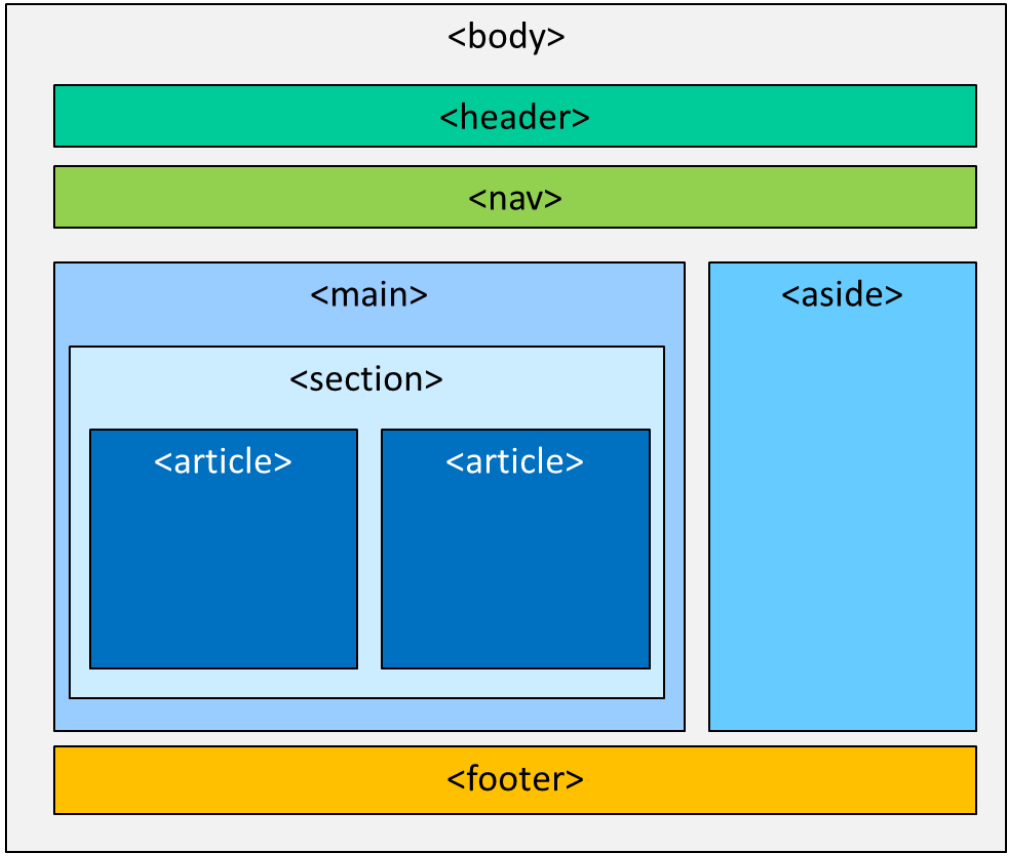
Unicode CodePoint als Dezimalzahl oder als Hexadezimalzahl.

```
&entity_name; oder &#CodePoint;
```

Für alle Unicode Zeichen verwendbar.

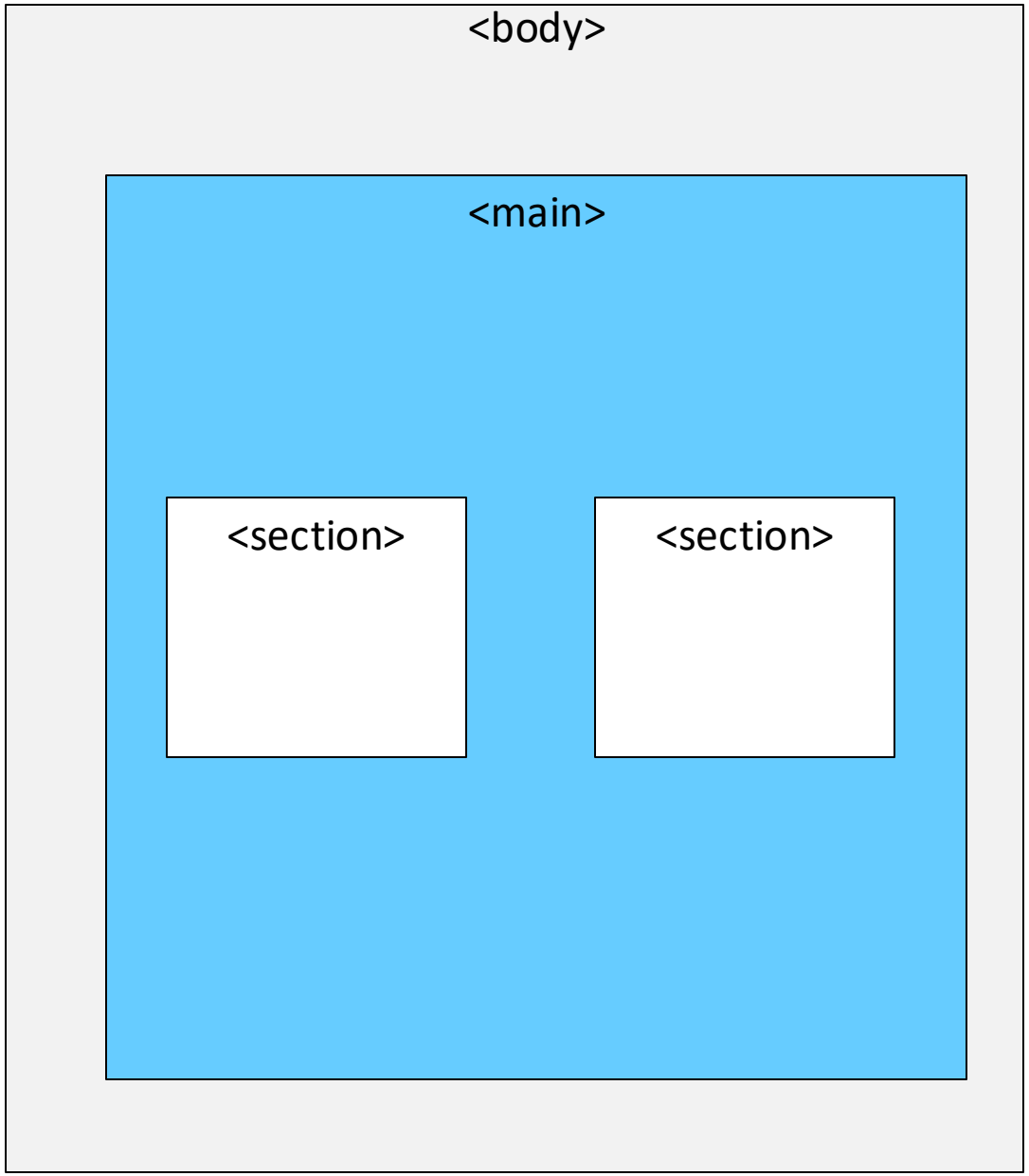
Result	Description	Entity Name	Entity Number
	non-breaking space		
<	less than	<	<
>	greater than	>	>
&	ampersand	&	&
"	double quotation mark	"	"
'	single quot	'	'
©	copyright	©	©
€	euro	€	€

3.2 Strukturierung einer Webseite



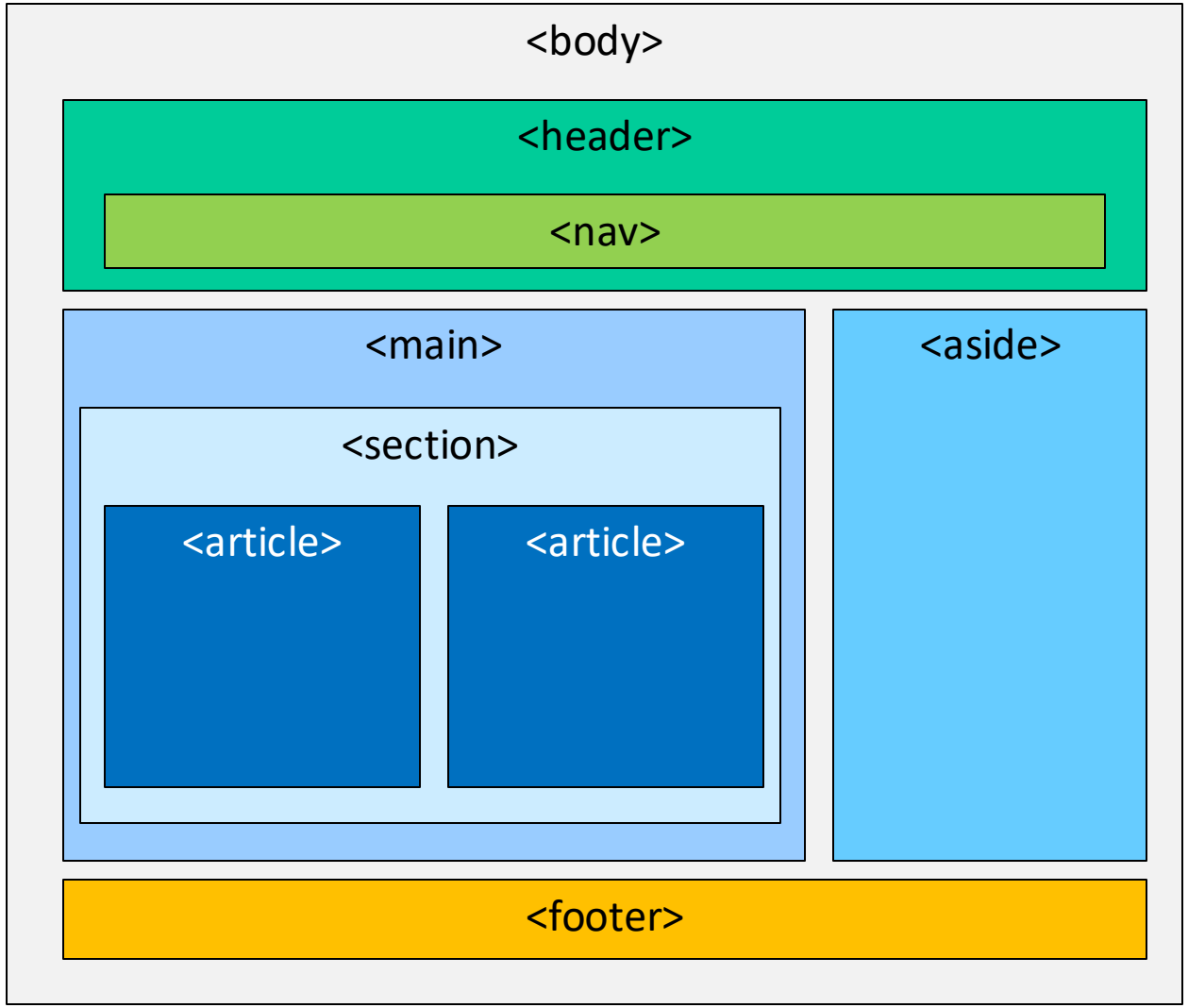
Komposition Diagramm

- Bevor Sie mit der Umsetzung einer Webseite beginnen, sollten Sie sich ein Kompositionsdiagramm erstellen, das den Inhalt und Aufbau ihrer WebSeite visualisiert.
- Dies können Sie auf einem Blatt Papier skizzieren ("scetching").
- Das Kompositionsdiagramm verwendet dabei eine sogenannte "has-a" Relation ("hat ein", "besteht aus"), die durch Container (Rechtecke) visualisiert wird.
  - Container (großes Rechteck) "besteht aus mehreren" Subcontainern (kleine Rechtecke)



Mehr Strukturelemente im HTML-<body> (Teil 1)

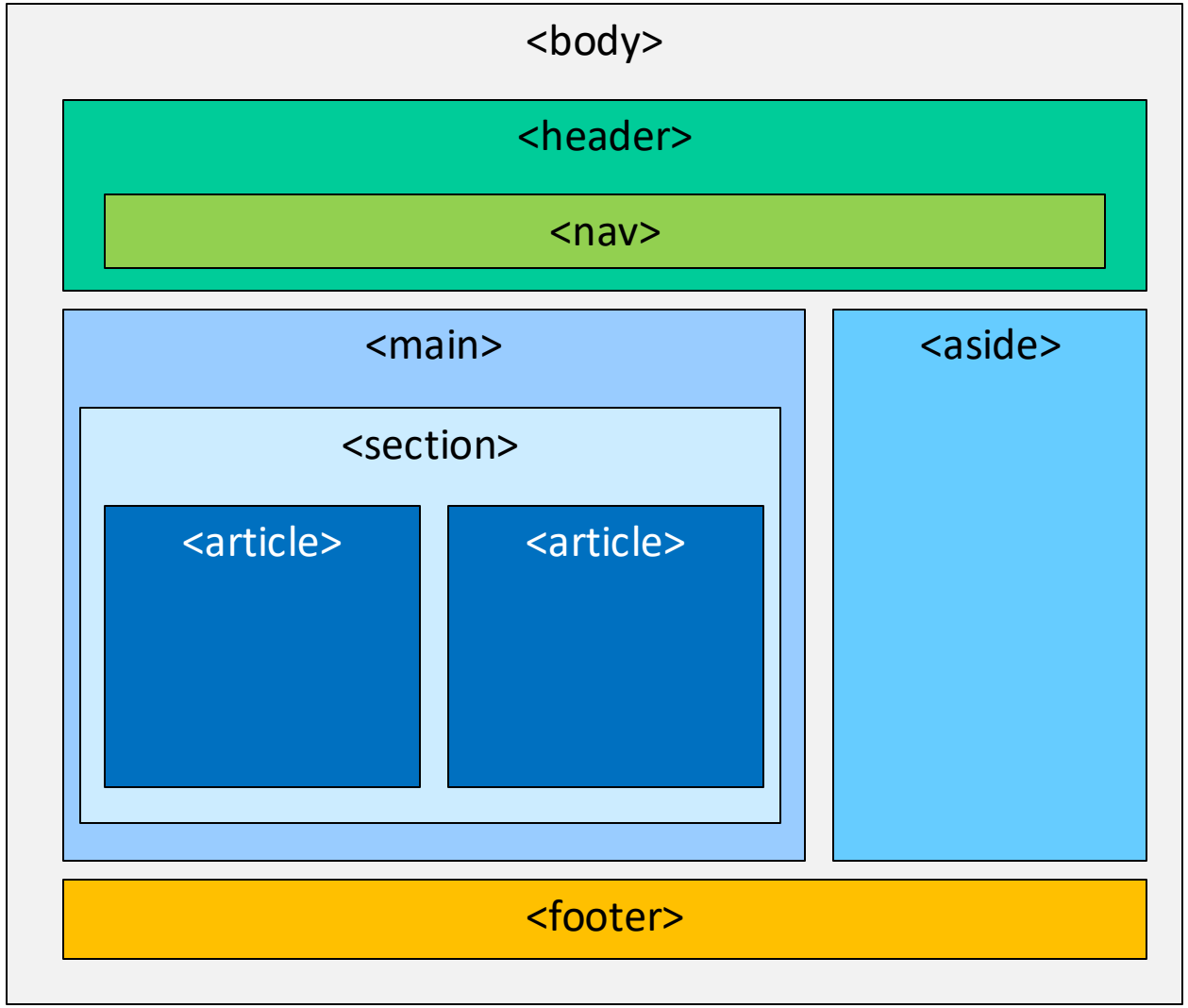
- In HTML5 wurden weitere neue Elemente definiert, um den HTML-Body übersichtlich zu strukturieren.
- Die verschiedenen Elemente haben die folgende Bedeutung:
  - `<header>`: Bereich für Überschriften (<h1> ...<h6>), Unterüberschrift, Logo, Navigationselemente.
  - `<nav>`: Navigationleiste der Webseite, enthält die Links (<a>) zum Navigieren innerhalb der Web-Seite oder auf andere Web-Seiten.
  - `<footer>`: Bereich für Informationen über den Autor, Copyright, Kontaktinformation, Sitemap, Links zu Social-Media-Kanälen, ...
  - `<main>`: umfasst alle inhaltstragenden Elemente innerhalb des <body>, es gibt nur ein <main>-Element pro HTML-<body.



Kompositionsdiagramm von <body>

Mehr Strukturelemente im HTML-<body> (Teil 2)

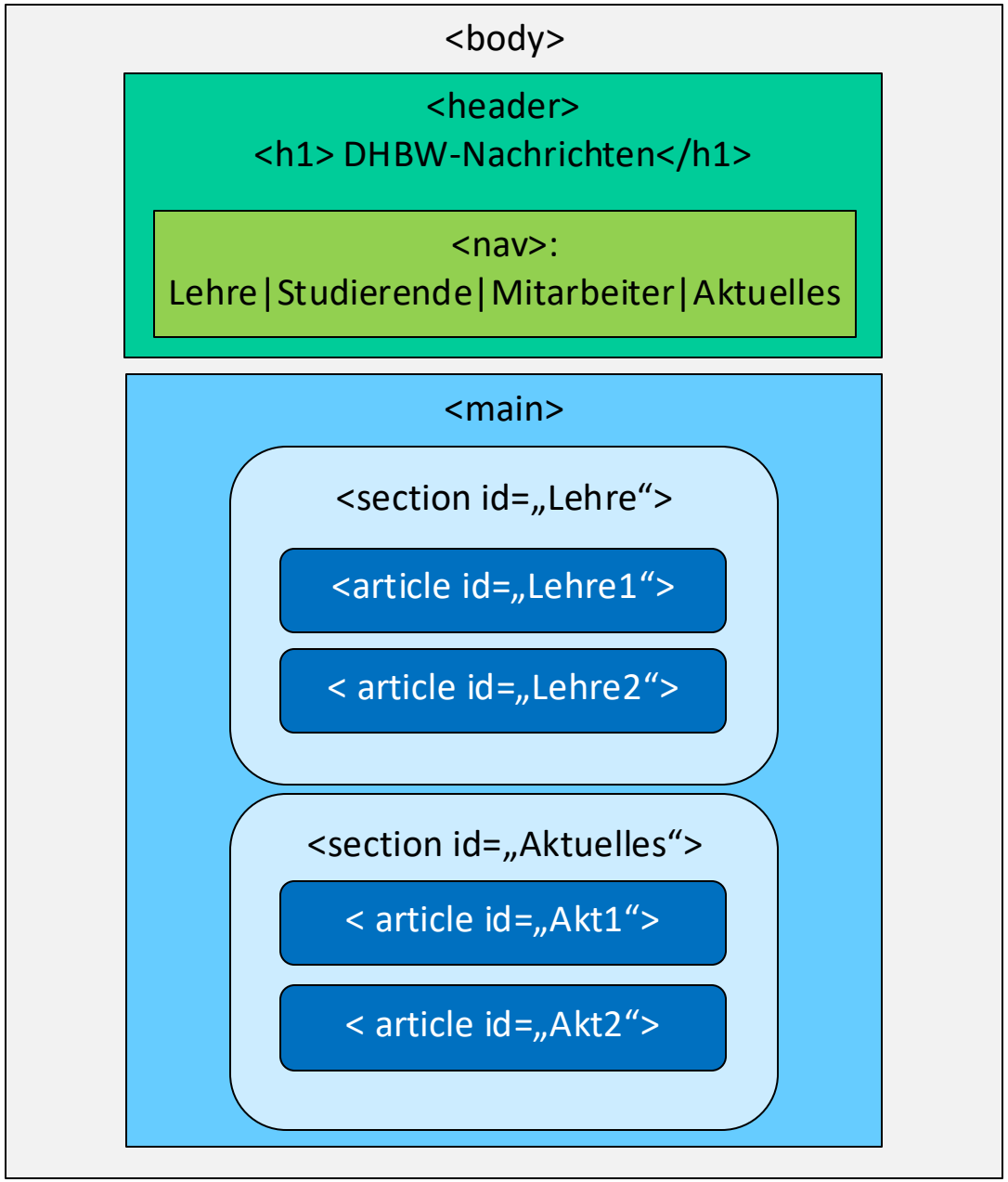
- `<section>:` fasst mehrere Inhaltselemente (`<article>`) zu einem Kapitel thematisch zusammen.
- `<article>`: eigenständiges Inhaltselement, z.B. eine News-Meldung, ein Blog-Post, ein Artikel zu einem Thema, ...
- `<aside>:` enthält Inhalte, die nur in einem indirekten, ergänzenden Zusammenhang mit dem Seiteninhalt stehen.
  - Beispiele: Randbemerkungen, Fußnoten oder Links zu weitergehenden Webseiten.



Kompositionsdiagramm von <body>

Komposition Diagramm

- Beispiel:
  - Webseiten-Body enthält eine Titelleiste (<header>), eine Navigationsbar (<nav>) und einen Inhaltbereich (<main>).
  - Der Inhaltsbereich enthält zwei Kapitel (<section>), die wiederum in zwei Inhaltselemente (<article>) unterteilt sind.



Address -Element

- `<address>`: ist für die Kontakt-Information über einen Autor eines Artikels/Beitrags einer Webseite gedacht:
  - seine E-Mail-Adresse,
  - seine Telefonnummer,
  - ein HTML-Link zur persönlichen Webseite des Autors,
  - die physikalische Adresse des Autors sein.
- Befindet sich ein `<address>`-Element im body-Tag, gelten die Kontaktinformationen für das ganze Dokument, im `<article>`-Tag als Kontaktinformationen zum konkreten Beitrag.
- Suchmaschinen im WWW suchen auf einer Web-Seite gezielt nach dem `<address>`-Element, um so die Autor-Information zu finden und anzuzeigen.
- `mailto-URI` startet einen E-Mail-Client

```
mailto:agent@mi6.uk?cc=moneypenny@mi6.uk&bcc=qbranch@mi6.uk&subject=Frage%20zu%20HTML
```

James Bond
Spezialist für geheime Aktionen
Thames House
London
james.bond@mi6.uk
+44 007 1234567

```
<address>
    James Bond<br>
    Spezialist für geheime Aktionen<br>
    Thames House<br>
    London<br>
    <a href="mailto:james.bond@mi6.uk">j.bond@mi5.uk</a>
    <br>
    <a href="tel:+44007123456">+44 (0)007 123456</a>
</address>
```

öffnet E-Mail-Client

öffnet Telefon-Client

HTML – Globale Attribute

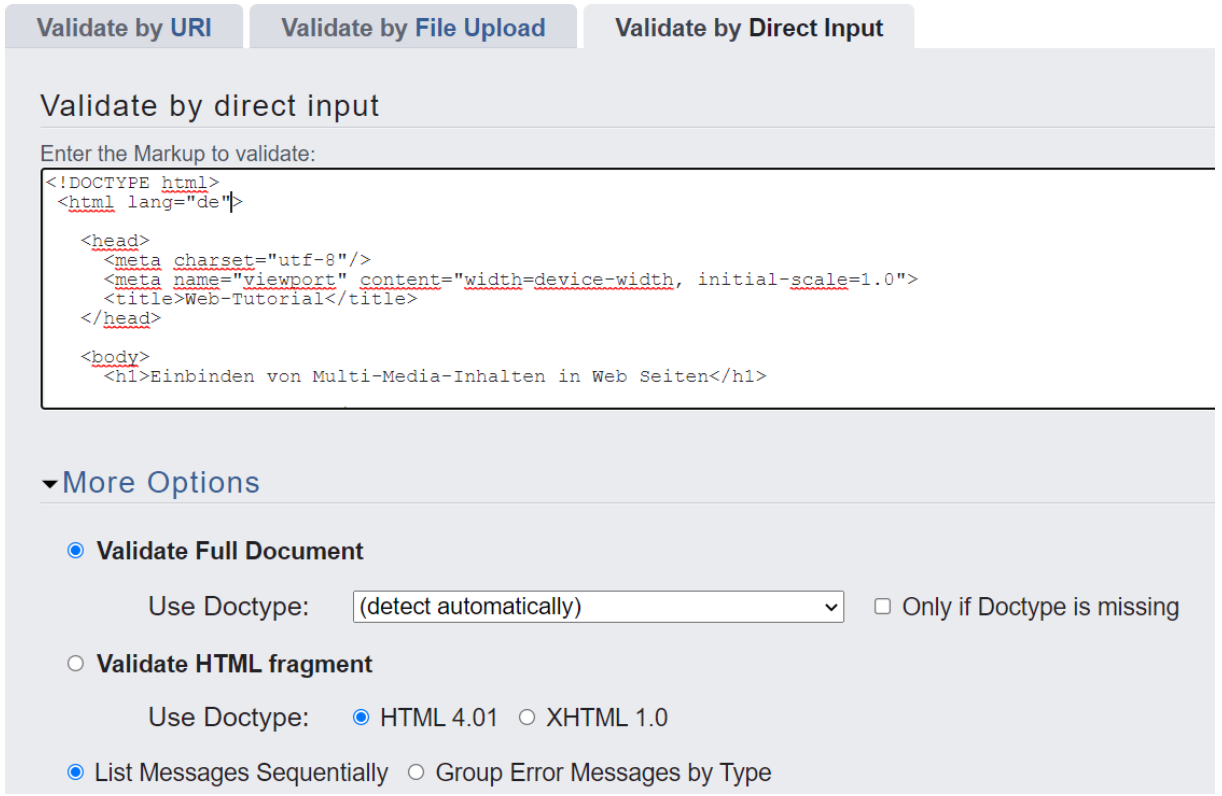
- Globale Attribute können in jedem Element verwendet werden.
- Sehr häufige verwendete globale Attribute sind class, id, lang, style, ...
  - `id`-Attribut: weist einem HTML-Element eine eindeutige Kennung zu.
  - `class`-Attribut: weist einem HTML-Element einen oder mehrere Klassennamen zuzuweisen.
  - `lang-Attribut`: definiert die Sprache des Element-Inhaltes. Es ist "Best Practise" für das gesamte Dokument die Sprache anzugeben
    <html lang="de">
  - `style-Attribut:` definiert CSS-Regeln direkt in HTML.

```
<!DOCTYPE html>
<html lang="de">

  <body>
    <h1 id="Kapitel1" style="text-align: center;">
         Kapitel1</h1>
      <!--   ToDo  -->
    <h2 id="Kapitel2">Kapitel2</h2>
    <p class="Text Wichtig">
        Ein wichtiger Aspekt stellt ….
    </p>

  </body>

</html>
```

HTML-Validation

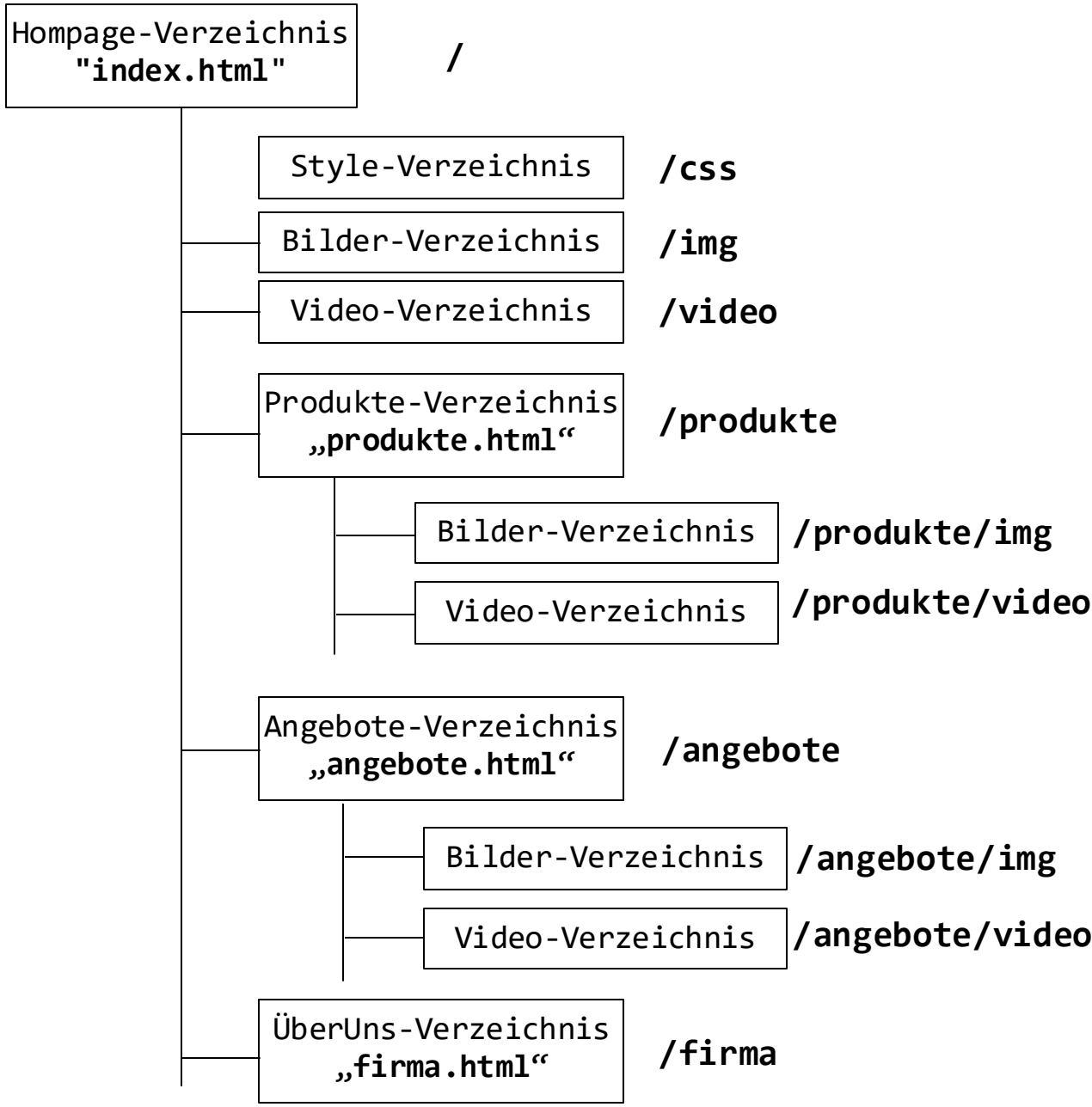
- Analog zu XML-Dokumenten lassen sich HTML-Dokumente auf ihre Gültigkeit überprüfen.
- Durch die Angabe von <!DOCTYPE html> teilen Sie einem Parser/Validator mit, das er ihr Dokument auf die richtige Umsetzung der HTML-Syntax prüfen soll.
- Das W3C stellt ein kostenfreies Tool zur Verfügung mit dem sich eine HTML-Seite auf Gültigkeit prüfen lässt.
- Viele IDEs (Integrated Development Environment) enthalten sogenannte HTML-Linter zur Überprüfung der Syntax des Quellcodes (z.B.: Webstorm).
- Wichtig: Da Browser je nach Hersteller unterschiedlich fehlertolerant sind und kleinere HTML-Fehler tolerieren, sollten Sie eine Validierung immer durchführen.

https://validator.w3.org/



Sitemaps

- Besteht ihr Webauftritt aus mehreren Webseiten, können Sie mittels einer Sitemap (dt. Seitenverzeichnis) ihren Webauftritt planen und visualisieren.
- In der Sitemap visualisieren Sie grafisch in Form eines hierarchischen Verzeichnisbaumes die Struktur ihrer Webseite.
- Ausgehende von der Hauptseite visualisieren Sie ihre Unterseiten und Webobjekte, die Sie verwenden möchten
- Pro Unterseite (Landings Page) legen Sie ein eigenes Verzeichnis mit den zugehörigen Unterverzeichnissen für externe Ressourcen an.
- Die Planung kann initial mittels Skizzen und final mittels einem graf. Tool visualisiert werden.



Sitemap XML Protokoll

- Damit Suchmaschinen eine Webseite einfach durchsuchen können, wird die Sitemap elektronisch in Form einer XML-Datei typischerweise im Root-Verzeichnis hinterlegt.
- Damit die XML-Tags eindeutig interpretiert werden können, werden diese durch das sogenannte Sitemap Protocols definiert (https://www.sitemaps.org/protocol.html).
- Die `sitemap.xml`-Datei wird typischerweise im root-Verzeichnis des Internetauftritts abgespeichert.
- Innerhalb der `robots.txt` – Datei können Sie den Speicherort ihrer Sitemap einer Suchmaschine bekanntmachen.

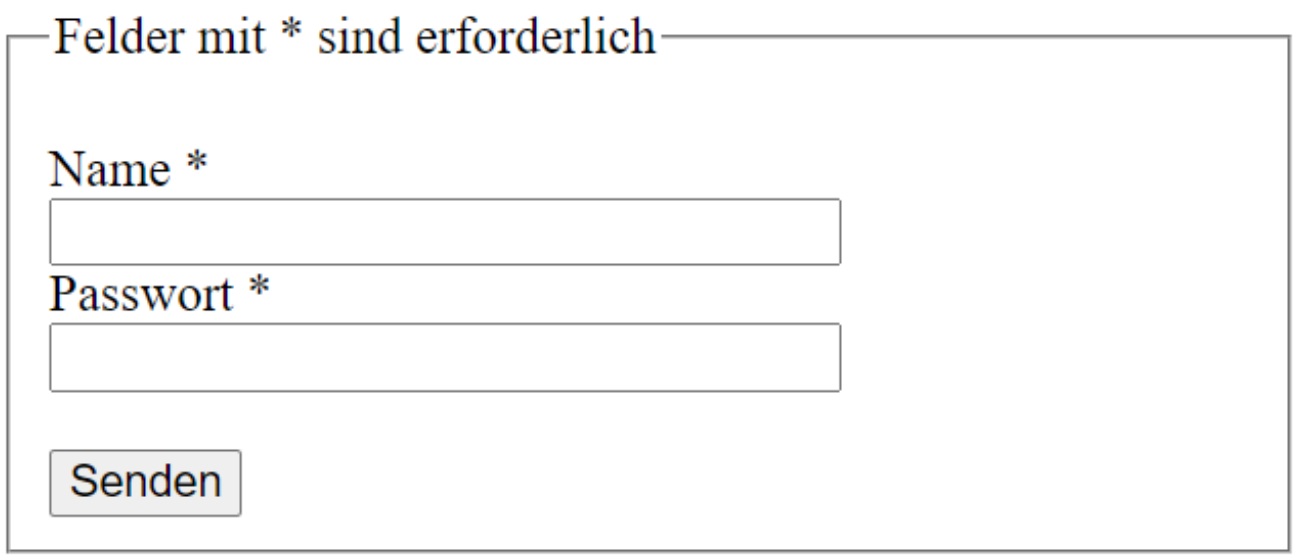
```
<?xml version="1.0" encoding="UTF-8"?>
<urlset
   xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">

  <url>
      <loc>http://www.example.de/</loc>
      <lastmod>2021-06-01</lastmod>
      <priority>0.9</priortity>
      <changefreq>monthly</changefreq>
  </url>
  <url>
      <loc>http://www.example.de/produkte/</loc>
      <lastmod>2021-07-15</lastmod>
      <priority>0.9</priortity>
      <changefreq>monthly</changefreq>
  </url>
…
</urlset>
```

robots.txt und sitemap.xml

- Mittels einer `robots.txt` Datei, wird einem Crawler mitgeteilt, auf welche Bereiche des Webauftritts der Crawler zugreifen darf.
- Die `robots.txt` Datei befindet sich im Root-Verzeichnis.
- Der Inhalt besteht aus Gruppen, die mit einer User-Agent-Zeile beginnt und mehrere Zugriffsregeln pro Crawler enthalten können.
- Der Crawler wendet die spezifischste Gruppe an, die für ihn anwendbar ist und zuerst in der Datei erscheint.
- Durch das Schlüsselwort Sitemap
  `Sitemap: http://www.example.de/sitemap.xml`
  wird dem Crawler die Position der sitemap.xml Datei mitgeteilt. Die URL muss vollständig qualifiziert sein.

```
# Example 1: Block Bingbot for the whole website
User-agent: Bingbot
Disallow: /

# Example 2: Googlebot is not allowed to read
# documents in /privacy
User-agent: Googlebot
Disallow: /privacy

# Example 3: All other Bots are allowed to read
# the complete Website
User-agent: *
Allow: /

Sitemap: https://www.example.com/sitemap.xml
```

3.3 <meta>-Informationen, <form> und <iframe>



Metadaten Content

- <meta>-Tags ermöglichen die Angabe von Informationen über eine Webseite im <head>-Bereich eines HTML-Dokuments.
- Diese werden auf der eigentlichen Webseite nicht angezeigt.
- <meta>-Tags liefern Zusatzangaben zu einer Webseite, die ein Webbrowser zur Darstellung der Seite oder eine Suchmaschine zur Klassifizierung der Webseite verwenden kann.

```
<meta name="Tagname" content="Information">
```

- Das <meta>-Tag
  - robots gibt einer Suchmaschine Verhaltensregeln vor
  - description beschreibt den Inhalt der Webseite.
  - keyword definiert Schlüsselworte für Suchmaschinen
  - author definiert den Author der Webseite
  - viewport definiert das Verhalten des Viewports auf dem Webclient („responsive Design“)

```
<head>
  <meta name="robots" content ="noindex, nofollow">
  <meta name="description" content="Free Web Tutorials">
  <meta name="keywords" content="HTML, CSS, JavaScript">
  <meta name="author" content="John Doe">
  <meta name="viewport" content="width=device-width,
                                 initial-scale=1.0">
</head>
```

Meta-Tag `viewport`

- Standardmäßig rendern mobile Geräte Webseiten in einem virtuellen Ansichtsfenster, das breiter (980px) als der Geräte-Bildschirm ist, und verkleinern dann das gerenderte Ergebnis, sodass die ganze Seite sichtbar wird.
- Meta-Tag `viewport`: Dient dazu den Browser beim Rendern der Webseite die Größe des initialen Viewports (Größe des initialen Fensters) anzugeben.

  Durch die Vorgabe von width=device-width wird die Bild-schirmbreite als initiale Größe für das Rendern des Webseiten-Inhaltes verwendet, und die Anzeige auf diese Bildschirmgröße optimiert.
- Die Angabe initial-scale=1.0 steht für den anfänglichen Zoomfaktor und besagt im Beispiel, dass die Seite komplett (als ohne Zoom) dargestellt werden soll.

```
<meta name="viewport"
      content="width=device-width, initial-scale=1.0">
```

ohne width-Angabe
(Anzeige auf Default Viewportsize wird verkleinert)

width=device-width
(Anzeige wird auf die tatsächliche Bild-schirmbreite optimiert)

Metadaten HTTP-EQUIV

- Mithilfe des Attributs `http-equiv` können HTTP-Headern im HTML-Dokument mitgegeben werden.
- Damit stellt der Entwickler einer Webseite sicher, das HTTP-Information auch dann zur Anwendung kommen, wenn der Webserver nicht entsprechend konfiguriert wurde und somit die korrespondierenden Werte im HTTP-Header fehlen.
- Es gilt die Regel, dass ein gleichnamiger HTTP-Header gegenüber einem HTML-Meta `http-equiv` Tags bevorzugt wird.

- Das `http-equiv`
  - `cache-control` kann mittels `no-cache` das Cachen einer Web-Seite im Browser unterbinden, mittels `private` darf die Web-Seite auf dem Browser des Benutzers gecached werden.
  - `refresh` automatische Aktualisierung der aktuellen Webseite nach 5s (ohne url=) oder Weiterleitung auf die Webseite www.example.de
  - `expires` definiert die Zeit in Sekunden, nachdem eine Webseite in einem Cache als abgelaufen gilt. Bei 0s muss der Inhalt der Webseite immer vom Originalserver geladen werden.

```
<meta http-equiv="cache-control" content="private">
<meta http-equiv="refresh" content="5; url=http://www.example.de/">
<meta http-equiv="expires" content="43200"/>
```

Formulare

- HTML-Formulare ermöglichen einem Benutzer mit einer Webseite zu interagieren. Beispiel dafür sind Blogs, Login-Seiten, Chats, Web-Shops,...
- Formulare werden mittels des `<form>`-Tags generiert.
- Der Datenversand erfolgt über die Attribute action (URL) und method (HTTP-Methode), die im <form>-Tag angegeben sind:
  - `method` – definiert die HTTP-Methode ("`post`" oder "`get`" (default) mittels der die Formdaten an den Web-Server übertragen werden sollen.
  - `action` – definiert die Adresse der Anwendung auf dem Web-Server (im Beispiel /login)
- Alternativ können die Daten per JavaScript gesendet werden. Dazu kann man das id-Attribut verwenden (siehe hinten).
  - `id` – um das Formular in CSS oder JavaScript eindeutig referenzieren zu können.

HTTP-POST

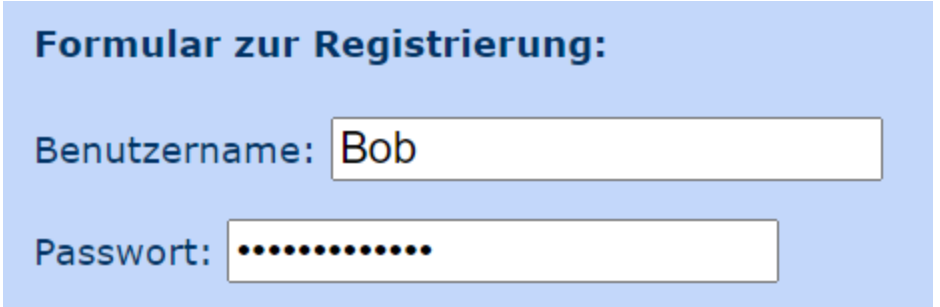
```
<form method="post" action="/login" id="login">
  …
  <button id="btn" type="submit">Senden</button>
</form>
```

HTTP-GET

```
<form action="/login" id="login">
  …
  <button id="btn" type="submit">Senden</button>
</form>
```

Eingabefelder in Formularen

- Das `<input>`-Tag erzeugt Eingabefelder, Checkboxen und Radiobuttons in einem Formular und ist ein Inline-Element.
- Funktion und Aussehen des `<input>`-Tag werden durch das Attribut type gesteuert:
  - `type="text"`: einfaches Text-Feld
  - `type="password`: Passwort-Feld Eingabezeichen werden maskiert
  - ...
- Das `name-Attribut` identifiziert das Eingabefeld bei der Übertragung zum Server.
- Das `value-`Attribut kann einen Default-Text in dem Eingabefeld vorbelegen (optional).



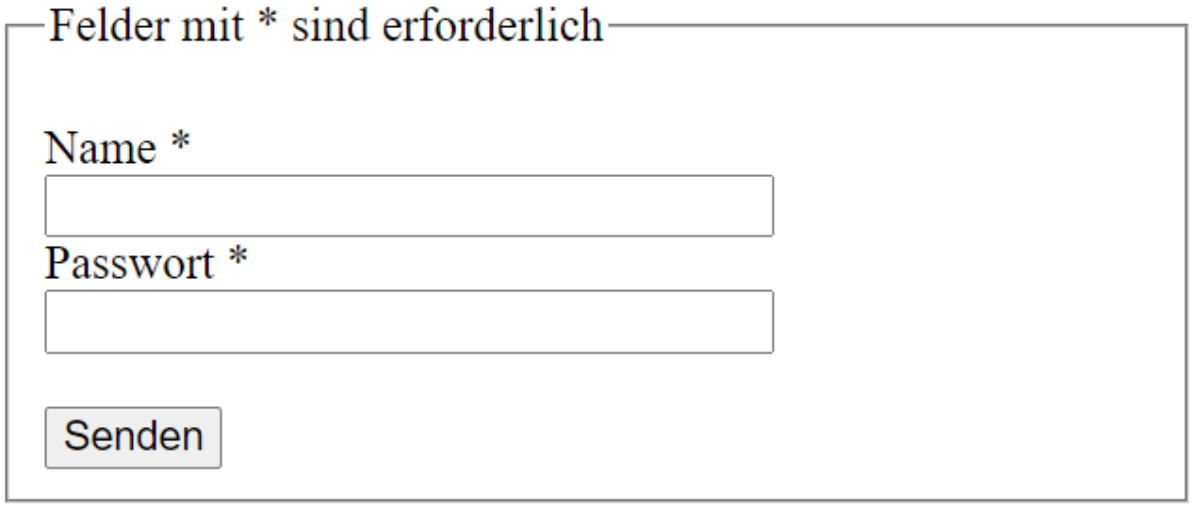
- Gesendete Daten im HTTP-Body bei POST

```
username=Bob&passwd=<eingegebenes Passwort>
```

```
<input id="name" type="text" name="username" value="Bob" size="30">
<input id="passwd" type="password" name="passwd" size="30">
```

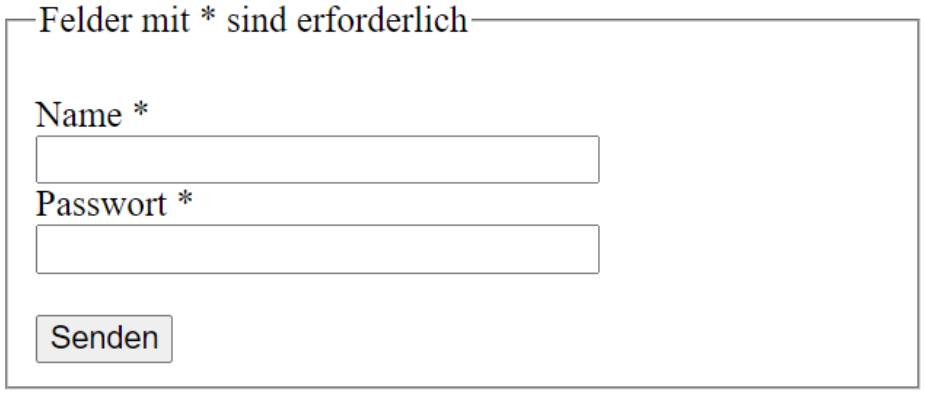
Darstellung eines Formulars

- Mittels den Formular-Tags `<fieldset>` und `<legend>` können zusammengehörende `<input>`-Eingabefelder optisch zusammengefasst werden und mit einer beschreibenden Information versehen werden
  - `<fieldset>` setzt einen Rahmen um eine Gruppe von Eingabefeldern
  - `<legend>` fügt dem Rahmen einen Text hinzu.



Beispiel: Login-Formular

```
<form method="post" action="/login" id="frmlogin">
  <fieldset>
      <legend> Felder mit * sind erforderlich</legend> <br>
      <label for="uname" size=30>Name *</label> <br>
      <input id="uname" type="text" name="uname" size="30"> <br>
      <label for="passwd">Passwort *</label> <br>
      <input id="passwd" type="password" name="passwd" size="30"> <br>
      <button id="btnlogin" type="submit">Senden</button>
    </fieldset>
 </form>
```

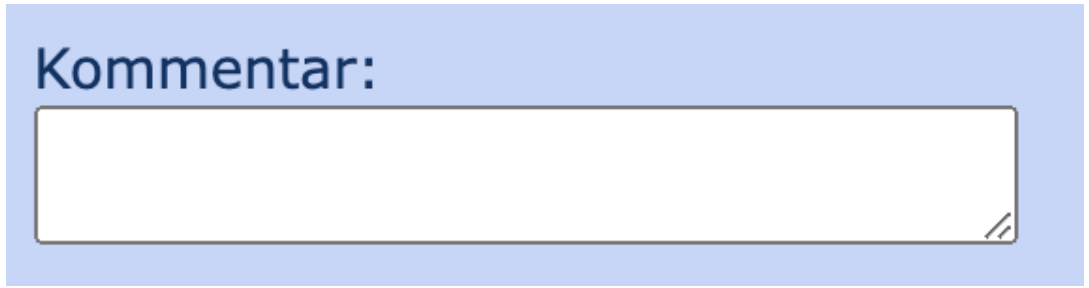


- Das Formular-Tag `<label>` verknüpft Eingabefelder, Button und Checkboxen eines Formulars mit einer Beschriftung („label“). Das `for`-Attribut referenziert das `id`-Feld des jeweiligen Eingabefeldes.
- Wenn das <input>-Formularfeld sich innerhalb des <label>-Tags befindet, wird das for-Attribut für die Zuordnung des Labels zum Eingabefeld nicht benötigt.

Eingabefelder in Formularen

- Wenn mehr als eine Textzeile für die Eingabe erwünscht ist,
  kann anstelle des <input>-Tags mit type="text"
  das <textarea>-Tag benutzt werden.

```
<label for="textarea">Kommentar:</label>
<br>
<textarea id="textarea" cols="30" rows="2" name="kommentar"> </textarea>
```



- Gesendete Daten mit POST

```
kommentar=1.+Zeile%0D%0A2.+Zeile
```

URL-Encoding:

+ ersetzt Leerzeichen.

%0D%0A steht für den Zeilenumbruch (CR+LF).

Radioboxen

- Mittels dem type-Attribut des <input>-Tag lassen sich auch Radioboxen anlegen.
  - `type="radio" :` Radioboxen, es kann immer nur eine Box ausgewählt werden (entweder oder), das `name-Attribut` muss innerhalb einer Radiobox-Gruppe gleich gesetzt werden.
  - `checked:` Attribut selektiert ein Checkbox-Element als Default-Auswahl .
  - `name="radiogroup":` Definiert den Namen der jeweiligen Radio-Gruppe

```
<form action="/setting" method="GET">
 <fieldset>
 <legend>Radioboxen</legend>
     <input type="radio" id="radio1" name="radiogroup" value="On" checked>
     <label for="radio1">On</label>
     <input type="radio" id="radio2" name="radiogroup" value="Off">
     <label for="radio2">Off</label>
 </fieldset>
 <button type="submit">Senden</button>
</form>
```


```
/setting?radiogroup=On
```

Checkboxen

- Mittels dem type-Attribut des <input>-Tag lassen sich auch Checkboxen anlegen.
  - `type="checkbox"` : Checkboxen, es kann eine oder mehrere Checkboxen ausgewählt werden.
  - `checked:` Attribut selektiert ein Checkbox-Element als Default-Auswahl .
  - `value="Zitrone"`: Definiert den Wert des oder der ausgewählten Input-Elemente.

- Beispiel: Auswahl beider Früchte und klicken des submit-buttons ergibt im POST-Body:

```
check[]=Zitronen&check[]=Birnen
```

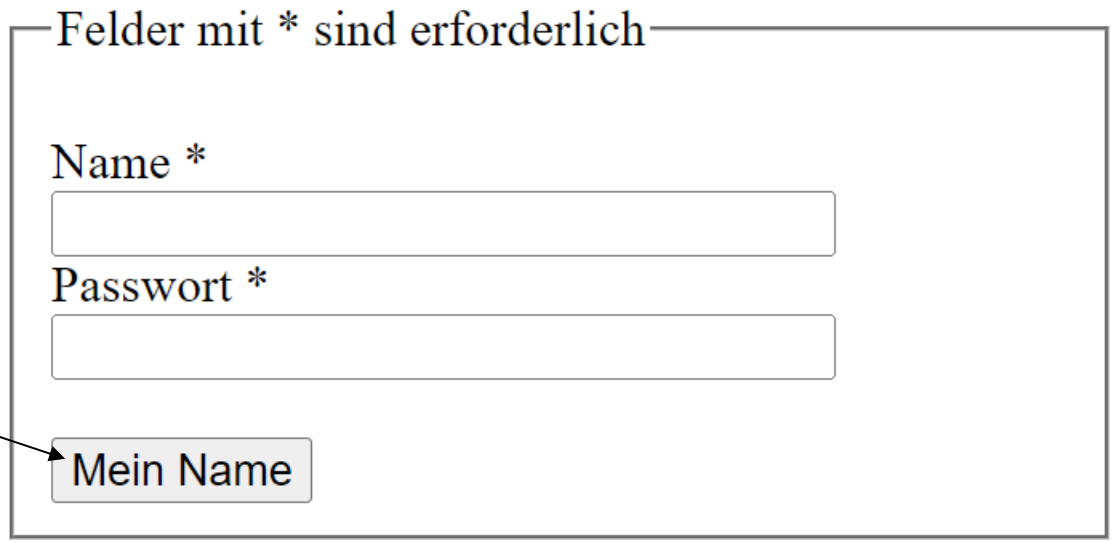

```
<fieldset>
<legend>Checkboxen</legend>
    <input type="checkbox" id="check1" name="check[]" value="Zitronen" checked>
    <label for="check1">Zitronen</label>
    <input type="checkbox" id="check2" name="check[]" value="Birnen">
    <label for="check2">Birnen</label>
</fieldset>
```

Formulare und Button

- Mittels dem `<button >`-Element kann die Form um eine Schaltfläche erweitert werden.

```
<button type="button|submit|reset">
  Mein Name
</button>
```

- In Abhängigkeit vom type-Attribut zeigt der Button ein unterschiedliches Verhalten:
  - `type="button"`: Button kann geklickt werden, kein Default-Verhalten wird zusammen mit Javascript verwendet.
  - `type="submit"`: Button übermittelt die Form-Daten, Default-Wert.
  - `type="reset"`: Button entfernt die Einträge in den Form-Eingabefeldern.



- Einem Button kann auch ein value- und name-Attribut zugeordnet werden. Ein submit-Button oder reset-Button überträgt dann zusätzlich das Name-Value-Paar an den Server.

HTML-<iframe>

- Das `<iframe>`- Inline Element ermöglicht die Einbettung eines beliebigen Inhalts (Webseiten, Formulare, Bilder, Videos, Tabellen und nicht zuletzt Werbung) in die aktuelle Webseite.
- Das `src`-Attribute enthält den `URI` für das einzubettende Webobjekt.
- Der Inhalt des iframe-Fensters ist unabhängig von der umgebenden Webseite. Die eingebettete Web-Seite verwendet ihre eigenen Style-Sheets und Skripte.
- `iframe`-Fenster verändern umgekehrt auch nicht das Layout der umgebenden Web-Seite.
- Vorteil: Inhalte aus verschiedenen Quellen können in einer einzigen Darstellung angezeigt werden.
- Nachteil: Externe Inhalte können schadhaften Code oder Links auf unsichere Verbindungen enthalten. Sicherheitsmaßnahme: sandbox-Attribut.

Zusätzliche Information zeigt der folgende Artikel (als iframe eingebunden!).


```
<iframe height="480" width="960"
        src="https://de.wikipedia.org/wiki/Quantencomputer">
</iframe>
```

Template für eine HTML-Webseite

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="css/style.css">
    <link rel="shortcut icon" href="favicon.ico">
    <meta name="viewport"
          content="width=device-width,
          initial-scale=1.0">
    <title> <!-- ToDo --> </title>
  </head>
  <body>
    <!-- ToDo -->
  </body>
</html>
```

```
<body>
        <header>
           <nav>
           </nav>
        </header>
        <main>
            <section>
                <article>
                </article>
            </ section >
            <aside>
            </aside>
        </main>
         <footer>
        </footer>
 </body>
```