

HTML - HyperText Markup Language

Gute Online-Unterlagen zum Selbstlernen:

<https://wiki.selfhtml.org/> resp. <https://selfhtml.org/>

<https://www.w3schools.com/>

<https://developer.mozilla.org/de/docs/Learn>

HyperText Markup Language

HTML ist die Beschreibungssprache des WWW

- Hypertext: Dokumente, die Verweise auf andere Dokumente enthalten
- universelle, plattformunabhängige Markup-Sprache
- Markup steht dabei für Auszeichnung und bedeutet, dass Teile eines Textes als besondere strukturelle Elemente ausgezeichnet werden können. Beispiele: Überschriften, Tabellenstrukturen, ...
- Bei HTML wird die Auszeichnung mit einer Semantik verknüpft
- Auszeichnungselemente können einen Inhalt haben

Eigenschaften von HTML

- HTML dient der Definition des Aufbaus einer Internetseite: Elemente, Strukturen, Verweise (Hyperlinks), referenzierte Elemente (Grafiken, Multimedia...)
- Seit 1992 standardisiert und zunehmend fortentwickelt
- Im Jahr 2014 wurde HTML 5 standardisiert

HyperText Markup Language

HTML, und was gibt es noch?

- HTML: Inhalt und Aufbau/Grundstruktur der Seite
- CSS: Design/Erscheinungsbild des Dokumentes
- JavaScript: Logik, Programmierung beim Verarbeiten des Dokumentes

Hierarchische Gliederung der Auszeichnungselemente, z.B.

- Auszeichnungselemente haben einen festen Erstreckungsraum
- Sie haben eine Semantik und ggf. einen Inhalt
- Elemente können verschachtelt sein
- Verschachtelte Elemente müssen vollständig in den Erstreckungsraum der übergeordneten Elemente enthalten sein
- Repräsentation als Baum möglich (**DOM-Baum**)

[Überschrift] *Text der Überschrift* **[Ende Überschrift]**

[Liste]

[Listenpunkt] *Text des Listenpunkts* **[Ende Listenpunkt]**

[Listenpunkt] *Text des Listenpunkts* **[Ende Listenpunkt]**

[Ende Liste]

HTML-Elemente werden durch sogenannte „Tags“ markiert

- Tags werden in spitzen Klammern (<, >) notiert
- Die meisten HTML-Elemente werden durch ein einleitendes (<>) und ein abschließendes (</>) Tag markiert, außer der Erstreckungsraum ist auch ohne schließendem Tag klar
- Mit diesen Elementen ist über den Erstreckungsraum ein Inhalt assoziiert, der auch weitere Auszeichnungselemente beinhalten kann. Hier spielt die Reihenfolge eine Rolle!
- Es gibt Elemente, sogenannte **empty-Elements**, die grundsätzlich keinen Inhalt haben und von daher **keinen schließenden Tag benötigen** (Beispiel:
, <col>, <link>,)

https://developer.mozilla.org/de/docs/Glossary/empty_element

- Tags können mittels Attributen (Key-Value-Paaren) präzisiert werden (Notation: name="wert"). Diese sind spezifisch zum Tag. Die Reihenfolge spielt hierbei keine Rolle

HTML

Auszeichnungselemente: Tags

`<h1>Überschrift 1. Ordnung</h1>`

`Wichtiger Text in Fettschrift`

Ein Zeilenumbruch wird durch den Tag `
`
eingeleitet

`<h1>HTML - die Sprache des Web</h1>`

``

Vereinfachte Darstellung eines HTML-Dokuments

```
<!DOCTYPE ...>
```

```
<html>
```

```
<head>
```

Grundsätzliches zum Dokument

(z.B. <title>Fenstertitel</title>)

```
</head>
```

```
<body>
```

Eigentliches Dokument mit Auszeichnungselementen

```
</body>
```

```
</html>
```

HTML

Grundlegender Aufbau

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Titel der Webseite</title>
</head>
<body>

<h1>Überschrift</h1>
Text mit Zeilenumbruch am Ende<br>
An dieser Stelle erfolgt kein Umbruch:
Sag ich doch!

</body>
</html>
```

→ Kommentar, keine HTML-Auszeichnung
Wird als (empfohlene!) Anweisung
für den Browser genutzt, der weiß, was
zu verarbeiten ist

→ Der Inhalt des h1-
Elements ist „Überschrift“

HTML

Darstellung als DOM-Baum

Darstellung des HTML-Dokumentes als DOM-Baum

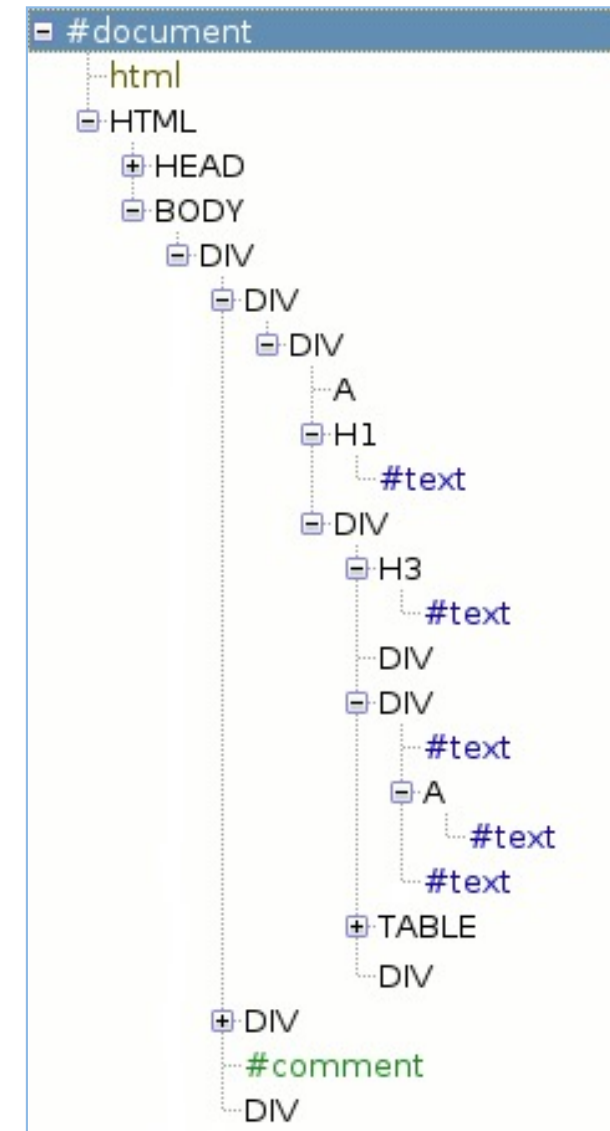
- Darstellungsform bei der Verarbeitung
- Tags und Elementinhalte werden durch Knoten repräsentiert
- Werkzeuge zum Navigieren durch den Baum erlauben Zugriff auf alle Tags und Attribute der Webseite

Beispiel (rechte Seite)

- Teilweise ausgeklappter DOM-Baum
- Text ist ein eigener DOM-Knoten (#text), der den Inhalt repräsentiert
- Reihenfolge wichtig

Tag-Attribute

- Zusätzlich kann jeder Knoten Eigenschaften durch Attribute erhalten

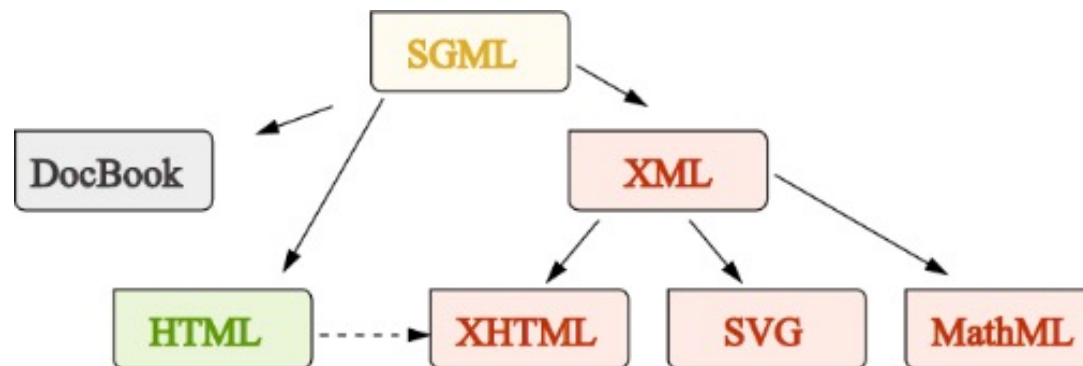


XML (Extensible Markup Language) und HTML

- HTML ursprünglich unabhängig von XML entwickelt
- Beide entstanden aus Standard Generalized Markup Language (SGML)
 - > Ähnlicher Aufbau (Tags, Baumstruktur)
- XML erlaubt beliebige Tags, HTML hat vorgegebene Tags
- HTML-Auszeichnungen implizieren eine Semantik, z.B. Überschrift

XHTML

- Vereinigung von HTML und XML
- Version 2.0 zu Gunsten von HTML 5 eingestellt



Quelle: <https://wiki.selfhtml.org/wiki/Datei:Auszeichnungssprachen.svg>

ISO 8879 (SGML): <https://www.iso.org/standard/16387.html>

HTML 4 (1997), HTML 4.01 (1999)

- Teils mit unspezifischen Vorgaben
- Einfach HTML zu schreiben
- Schwierig browserübergreifend das gleiche Layout zu erzeugen

XHTML: Extensible HTML (~2000)

- Strikter Standard sollte die Fehler von HTML 4 beheben
- Durchsetzung von XHTML scheitert auf Grund der Striktheit
 - > Minimale Fehler im XML können zum kompletten Ausfall der Seite führen, der Browser zeigt einen XML-Parse-Error

HTML 5 (Entwicklung seit ca. 2008, Recommendation 2014)

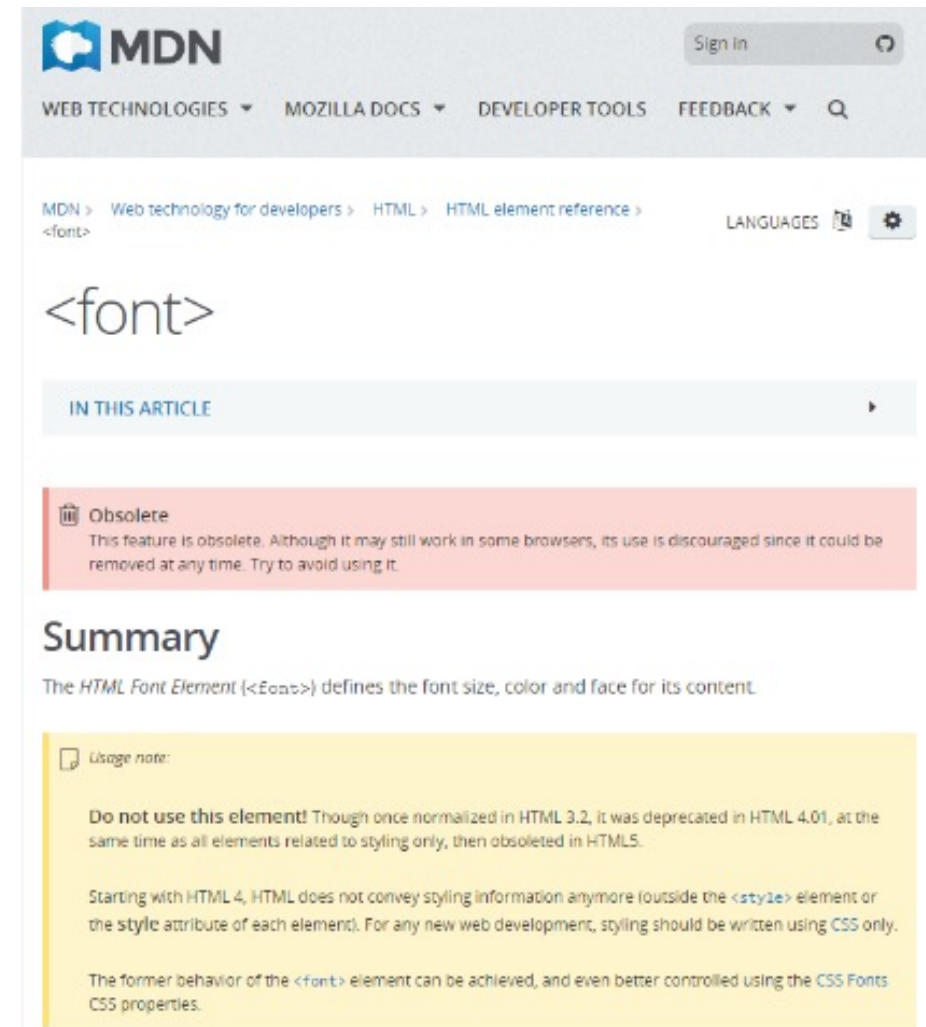
- Ein browserübergreifender einheitlicher Parser soll die Unterschiede im Anzeigen verhindern und gleichzeitig Flexibilität erlauben
- Zahlreiche moderne HTML-Erweiterungen (video, audio)
- Viele „Altlasten“ entfernt
- Oft als **Sammelbegriff für moderne Technologien** genutzt

Unsere Beispiele sind HTML 5

Weitere Informationen: <https://html.spec.whatwg.org/#history-2>

HTML Altlasten

- Styling im HTML-Code
 - > Veränderung von Farben, Textgrößen, Abständen, usw. im HTML (z.B. mittels font-Tags)
 - > Sollte seit HTML 4 (1997) mittels **CSS** gemacht werden!
- Escaping von Umlauten (und anderen Zeichen)
 - > Bei der Wahl des richtigen Zeichenstatzes ist das nicht nötig
- JavaScript im HTML-Code
 - > HTML, JavaScript und CSS sollten durch eigene Dateien getrennt sein



Quelle: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/font>

Beispiele für schlechten Code

```
<body bgcolor="red">
<font size="+2" onclick="alert('Hi');">Gro&szlig;er Text</font>
<b>Das hier ist wichtig!</b>

<table align="center">
  <tr>
    <td height="50" width="50">Text</td>
    <td height="50" width="100">Mehr Text</td>
  </tr>
</table>
...
```

Browserunterstützung

- Trotz Auszeichnung als „obsolete“ unterstützen moderne Browser aus Gründen der Kompatibilität i.d.R. immer noch alle veralteten Tags und Attribute

Viele Internetseiten enthalten immer noch schlechte Codebeispiele

- Im Zweifel MSDN oder MDN aufsuchen

Weitere Informationen: <https://www.w3.org/TR/html5/obsolete.html>

HTML

Mozilla Developer Network



The screenshot shows the MDN website in a browser window. The address bar displays the URL <https://developer.mozilla.org/de/docs/Web/HTML>. The page title is "HTML: HyperText Markup Language". The navigation bar includes links for "Technologien", "Referenzen & Leitfäden", "Feedback", a search bar, and "Anmelden". The main content area is titled "HTML: HyperText Markup Language" and includes a breadcrumb trail "Webtechnologien für Entwickler > HTML: HyperText Markup Language". The page is in German, as indicated by the "Deutsch" language selector in the top right. The content is organized into sections: "Verwandte Themen" (related topics), "Tutorials" (HTML basics, Introduction to HTML, Multimedia and embedding), "References" (HTML elements, Global attributes, <input> types), and "Documentation" (Useful lists, Contribute). The main text describes HTML as the fundamental building block of the web, defining the content and layout of web pages. It mentions that HTML is used to structure content, and that other technologies like CSS and JavaScript are used for styling and interactivity. The text also explains that HTML uses "markup" to format content, and that tags are case-insensitive. At the bottom of the page, there are three boxes: "HTML Einführung" (HTML Introduction), "HTML Tutorials", and "HTML Referenz" (HTML Reference). The "HTML Einführung" box contains a link to the "HTML Grundlagen" article. The "HTML Tutorials" box contains a link to the "HTML Lernbereich" (HTML Learning Area). The "HTML Referenz" box contains a link to the "HTML Referenz" section.

HTML: HyperText Markup Language

Webtechnologien für Entwickler > HTML: HyperText Markup Language

Deutsch

Verwandte Themen

HTML

Tutorials:

HTML basics

- Introduction to HTML
- Multimedia and embedding

References:

- HTML elements
- Global attributes
- <input> types

Documentation:

- Useful lists
- Contribute

HTML (HyperText Markup Language) ist der grundlegendste Baustein des Webs. Es beschreibt und definiert den *Inhalt* einer Webseite zusammen mit dem grundlegenden Layout der Webseite. Neben HTML werden im Allgemeinen andere Technologien verwendet, um das Erscheinungsbild (CSS) oder die Funktionalität/Verhalten (JavaScript) einer Webseite zu beschreiben.

"HyperText" bezieht sich auf Links, die Webseiten miteinander verbinden, entweder innerhalb einer einzelnen Webseite oder zwischen mehreren Webseiten. Links sind ein grundlegender Bestandteil des Webs. Indem Sie Inhalte in das Internet hochladen und mit Seiten verlinken, die von anderen Personen erstellt wurden, werden Sie ein aktiver Teilnehmer im World Wide Web.

HTML verwendet "Markup", um Texte, Bilder und andere Inhalte für die Anzeige in einem Webbrowser zu kommentieren. HTML-Markup beinhaltet spezielle "Elemente" wie `<head>`, `<title>`, `<body>`, `<header>`, `<footer>`, `<article>`, `<section>`, `<p>`, `<div>`, ``, ``, `<aside>`, `<audio>`, `<canvas>`, `<datalist>`, `<details>`, `<embed>`, `<nav>`, `<output>`, `<progress>`, `<video>` und viele andere.

Die Tags in HTML sind case-insensitive. Das heißt, sie können in Groß- und Kleinbuchstaben oder in einer Mischung geschrieben werden. Beispiel `<title>` tag kann als `<Title>`, `<TITLE>` oder auf andere Weise geschrieben werden.

Die folgenden Artikel helfen Ihnen, mehr über HTML zu erfahren.

HTML Einführung

Wenn Sie neu in der Webentwicklung sind, lesen Sie unbedingt unseren [HTML Grundlagen](#) Artikel, um zu erfahren, was HTML ist und wie man es verwendet.

HTML Tutorials

Artikel über die Verwendung von HTML sowie Tutorials und vollständige Beispiele finden Sie in unserem [HTML Lernbereich](#).

HTML Referenz

In unserem umfangreichen Abschnitt [HTML Referenz](#) finden Sie die Details zu jedem Element und Attribut in HTML.

Tutorials für Anfänger

Referenzen

Zeichensatz des Dokumentes

- Bestimmt welche Zeichen für die übermittelten Bytes angezeigt werden
- Beispiele:
 - > ASCII-Byte: 65 (alternativ: 0x41 oder 01000001)
 - Entspricht: A
 - > UTF-8 Byte: 65 (alternativ: 0x41 oder 01000001)
 - Entspricht: A
- Die ersten 128 Zeichen (0-127) des ASCII und UTF-8-Standards sind gleich!
 - > Tatsächlich besteht ein ASCII-Code nur aus 7 Bits

UTF-8 ist abwärtskompatibel zu ASCII

UTF-8 ist abwärtskompatibel zu ASCII

- Problem: Es fällt evtl. nicht auf, wenn ein Dokument mit einem falschen Zeichensatz ausgeliefert wurde.

Mögliche Lösung: Andere Zeichen kodieren

- Beispiele:
 - > € statt €
 - > ä statt ä
 - > © statt ©
- Unsaubere Lösung!
 - > Vor allem historisch bedingt, da Browser sich in manchen Fällen über den übermittelten Zeichensatz hinwegsetzten
 - > Besser: Korrekten Zeichensatz angeben!
 - > **Im <head>-Element: <meta charset="UTF-8">**

When not to use escapes: <https://www.w3.org/International/questions/qa-escapes#not>

Nur wenige Zeichen müssen tatsächlich „escaped“ werden:

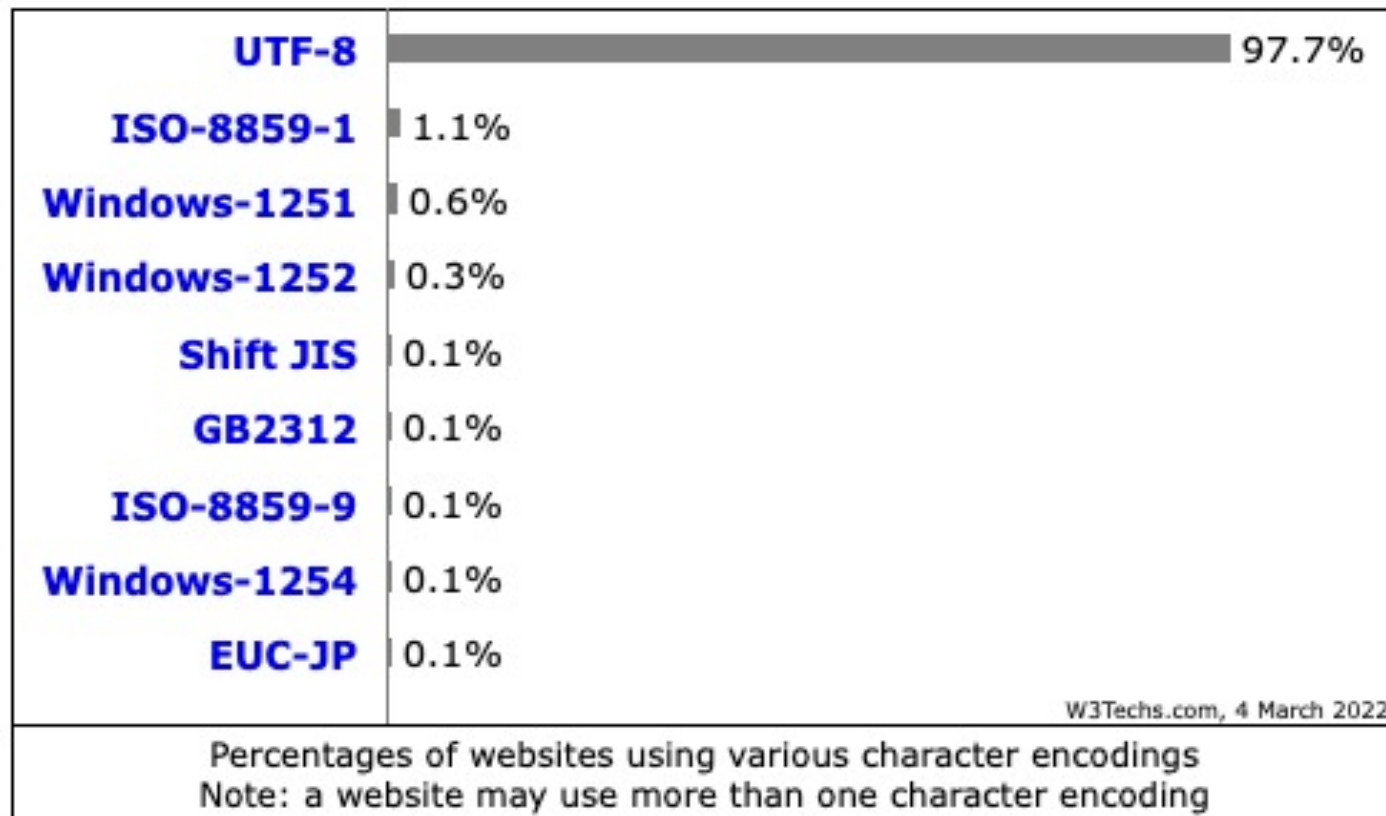
- Verwendung eckiger Klammern ohne Interpretation als Auszeichnung
 - > Beispiel: Ein h1-Tag: `<h1>`
- „Ampersand“, das sonst als Einleitung eines Escape-Characters interpretiert wird
- Anführungszeichen innerhalb von Attributen
 - > Beispiel: `title="Er sagte "Hallo";."`

Zeichen	Bedeutung	HTML-Code
<	Öffnende Klammer (Auswertung als Tag verhindern)	<code>&lt;</code>
>	Schließende Klammer	<code>&gt;</code>
&	„Ampersand“, Käufmännisches Und	<code>&amp;</code>
"	Doppelte Anführungszeichen	<code>&quot;</code>
'	Einfaches Anführungszeichen	<code>&#39;</code>

When to use escapes: <https://www.w3.org/International/questions/qa-escapes#use>

Den korrekten Zeichensatz wählen

- Immer UTF-8 wählen!
 - > Weitverbreitetste Implementierung von UNICODE-Zeichen



Quelle: https://w3techs.com/technologies/overview/character_encoding/

UTF-8 nutzen

1. Editor ggf. auf UTF-8 stellen

2. Encoding im HTML angeben

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8"/>
...
```

3. Prüfen, welche Einstellungen der Server übermittelt

- > Einstellung im Apache:
 - httpd.conf / .htaccess: `AddDefaultCharset UTF-8` oder `AddCharset UTF-8`
- > Bei widersprüchlichen Angaben nutzt der Browser das Server-Encoding
HTTP-Response Header: `Content-Type: text/html; charset=utf-8`

Weitere Informationen: <https://www.w3.org/International/questions/qa-html-encoding-declarations>

Enthält Informationen über das Dokument

- Beispiel

```
<head>  
  <meta charset="utf-8">  
  <title>Titel der Webseite</title>  
  <meta name="author" content="Autor der Seite">  
  <meta name="description" content="Beschreibung der Webseite,  
    die u.a. von Google verwendet wird">  
  <meta name="keywords" content="Schlüsselworte,HTML,lernen">  
  <link rel="stylesheet" href="style.css" type="text/css">  
</head>
```

- Die Informationen im <head> sind größtenteils für eine automatisierte Verarbeitung von Bedeutung (Browser, Bots)
- **Oftmals werden hier auch wesentlich Ressourcen wie CSS und JavaScript eingebunden**

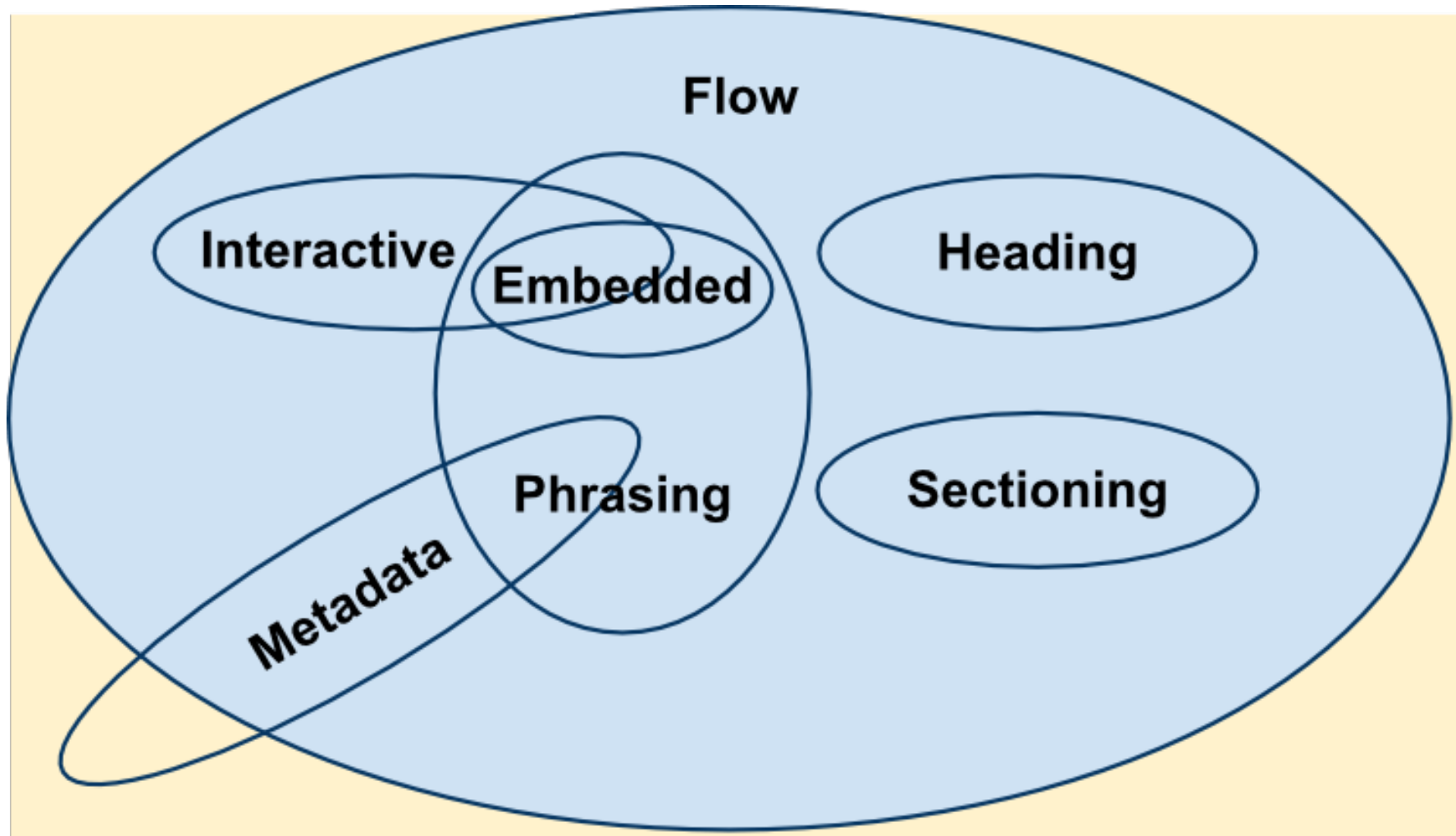
Enthält den eigentlichen Teil des Dokuments

- Beispiel

```
<body>
<h1>Überschrift</h1>

<h2>Tabelle</h2>
<table>
  <tr>
    <th>Land</th>
    <th>BIP</th>
  </tr>
  <tr>
    <td>Deutschland</td>
    <td>3,73 Billionen USD</td>
  </tr>
  <tr>
    <td>UK</td>
    <td>2,81 Billionen USD</td>
  </tr>
</table>
</body>
```

HTML Elemente



Vollständige Liste: https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Content_categories

Metadaten

- `<meta>`, `<title>`, `<base>`, `<link>`, `<script>`, `<style>`

Flow/Fluss-Elemente

- `<a>`, `<article>`, `
`, `<button>`, `<form>`, **`<div>`**, `<header>`,
``, `<input>`, `<nav>`, ``, `<p>`, `<script>`, `<h[1-6]>`,
`<section>`, ``

Elemente zur Aufteilung (HTML5 Semantics-Tags)

- `<article>`, `<aside>`, `<nav>`, `<section>`

Überschriften

- `<h1>`, `<h2>`, ...

Stil-Elemente

- ``, `<button>`, `<code>`, `<input>`, `<label>`, ``, `<time>`

Elemente zur Einbettung von Inhalten

- `<audio>`, `<canvas>`, `<iframe>`, ``, `<object>`, `<svg>`, `<video>`

Interaktive Elemente

- `<a>`, `<button>`, `<iframe>`, `<label>`, `<select>`, `<textarea>`

Vollständige Liste: https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Content_categories

Wichtige Unterscheidung

- In HTML4 gab es logische und physische Elemente zur Auszeichnung des Inhaltes (Text)
- Physische Elemente definierten direkt das Aussehen eines Elementes (z.B. unterstrichen, kursiv), treffen aber keine Aussage über die Bedeutung
- Logische Elemente machten eine Aussage über die Bedeutung des Elementes (z.B. um was für Text es sich handelt), die Formatierung hängt vom Browser ab, kann aber auch mit Hilfe von CSS verändert werden

Physische Auszeichnungselemente wurden faktisch mit HTML5 verworfen (auch wenn die Tags noch unterstützt werden)

- Faktisch sollte man möglichst mit **logischen** Auszeichnungen arbeiten
- **Mit CSS kann das Aussehen beliebig verändert werden**

HTML Elemente

Logisch vs. Physisch

Beispiele für logische Auszeichnungen

`stark betont`
`betont (emphasized)`
`<code>Quellcode</code>`
`<samp>Beispiel</samp>`
`<cite>Zitat</cite>`
`<dfn>Definition</dfn>`
`<abbr>abgekürzte Schreibweise</abbr>`

Beispiele für alte physische Auszeichnungen

`fett`
`<i>kursiv (italic)</i>`
`<u>unterstrichen (underlined)</u>`
`<strike>durchgestrichen</strike>`
`<big>größer als normal</big>`
`<small>kleiner als normal</small>`
`^{hochgestellt}`
`_{tiefgestellt}`

Weitere Informationen: https://developer.mozilla.org/de/docs/Web/HTML/Element/strong#Bold_vs._Strong

HTML unterscheidet 6 Überschriftenebenen

`<h1>Überschrift 1. Ordnung</h1>`

`<h2>Überschrift 2. Ordnung</h2>`

`<h3>Überschrift 3. Ordnung</h3>`

`<h4>Überschrift 4. Ordnung</h4>`

`<h5>Überschrift 5. Ordnung</h5>`

`<h6>Überschrift 6. Ordnung</h6>`

Nicht zu verwechseln mit: `<header>...</header>`

- Keine Überschrift!
- Element mit Einführungsinformationen zur Navigationshilfe
 - > Logo der Seite
 - > Gruppierung von Überschriften
 - > Autor
- `<footer>...</footer>` entspricht dem Gegenstück dazu

Weitere Informationen: <https://developer.mozilla.org/docs/Web/HTML/Element/h1-h6>, <https://developer.mozilla.org/docs/Web/HTML/Element/header>

HTML Elemente

Textabsätze und Zeilenumbrüche

<p>Textabsatz</p>

- Repräsentiert einen Absatz

`<p>Erster Absatz.</p>`

`<p>Hier beginnt der zweite Absatz.</p>`

`<p>Und hier ist noch ein Absatz.</p>`

**
**

- Erzwingt einen Zeilenumbruch
- Dies sollte nicht als Ersatz für `<p>` gesehen werden

HTML Elemente

Listen

Unsortierte Listen (Unordered List)

```
<ul>
```

```
  <li>Apfel</li>
```

```
  <li>Birne</li>
```

```
  <li>Kiwi</li>
```

```
</ul>
```

- Apfel
- Birne
- Kiwi

Sortierte Listen (Ordered List)

```
<ol>
```

```
  <li>Apfel</li>
```

```
  <li>Birne</li>
```

```
  <li>Kiwi</li>
```

```
</ol>
```

1. Apfel
2. Birne
3. Kiwi

Weitere Informationen: <https://developer.mozilla.org/docs/Web/HTML/Element/ol>, <https://developer.mozilla.org/docs/Web/HTML/Element/ul>

HTML Elemente

Tabellen

Formatierung

- `<table>` um die Tabelle einzuleiten
- `<tr>` beginnt eine Zeile (table row)
- `<th>` beginnt eine Header-Zelle
- `<td>` beginnt eine normale Zelle (table data)

Beispiel

```
<table>
  <tr>
    <th>Land</th>
    <th>BIP (in Billionen USD)</th>
  </tr>
  <tr>
    <td>Deutschland</td>
    <td>3,73</td>
  </tr>
  <tr>
    <td>UK</td>
    <td>2,81</td>
  </tr>
</table>
```

Land	BIP (in Billionen USD)
Deutschland	3,73
UK	2,81

Weitere Informationen: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/table>

HTML Elemente

Tabellen

<code><table></code>				
<code><tr></code>	<code><th></code> <code></th></code>	<code><th></code> <code></th></code>	<code><th></code> <code></th></code>	<code></tr></code>
<code><tr></code>	<code><td></code> <code></td></code>	<code><td></code> <code></td></code>	<code><td></code> <code></td></code>	<code></tr></code>
<code><tr></code>	<code><td></code> <code></td></code>	<code><td></code> <code></td></code>	<code><td></code> <code></td></code>	<code></tr></code>
				<code></table></code>

Formatierung

- Es ist zusätzlich möglich, dass eine Zelle mehrere Spalten oder Zeilen umfasst
- `colspan="2"` zum horizontalen Umfassen von zwei Zellen
- `rowspan="2"` zum vertikalen Umfassen von zwei Zellen

Beispiel (mit CSS-Klassen)

```
<table>
  <tr>
    <td colspan="2" class="red">1</td>
    <td rowspan="2" class="green">2</td>
  </tr>
  <tr>
    <td class="blue">3</td>
    <td class="yellow">4</td>
  </tr>
  <tr>
    <td colspan="3" class="red">5</td>
  </tr>
</table>
```

1		2
3	4	
5		

Bilder einbinden

- ``
 - > Das Attribut „src“ enthält den (relativen) Pfad (die URL) zur Bilddatei
 - > Das Attribut „alt“ enthält einen Alternativtext, der angezeigt wird, wenn das Bild nicht geladen werden kann oder von Screenreadern (barrierefrei) verarbeitet wird
- Die meist unterstützten Grafikformate sind
 - > JPEG, GIF, PNG, BMP, SVG
 - > https://en.wikipedia.org/wiki/Comparison_of_web_browsers#Image_formats_at_support
 - > Empfehlungen:
 - JPEG für Fotos, Bilder mit vielen Farben oder Farbverläufen
 - GIF für Animationen
 - PNG für Logos, kleinere Bilder oder Bilder mit wenig Farben

Beispiel

```

```

Weitere Informationen: <https://developer.mozilla.org/docs/Web/HTML/Element/img>

Verbindung zwischen zwei Dokumenten per Link

- Relative und absolute Angaben sind erlaubt
- Beispiele

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

```
<a href="https://example.com/datei.html">Ein absoluter Link</a>
```

```
<a href="mailto:test@example.com">Email versenden</a>
```

Verweise innerhalb des Dokumentes

- Ein Anker innerhalb des Dokumentes kann auch als Ziel eines Links angegeben werden
- Das Browserfenster springt dann an die jeweilige Stelle
- Beispiel

```
<h2 id="Anker">Überschrift</h2>
```

```
<a href="#Anker">Link zur Überschrift</a>
```

Weiterführend: <https://developer.mozilla.org/docs/Web/HTML/Element/a>, <http://w3c.github.io/html/browsers.html#navigating-to-a-fragment-identifier>

<div>...</div>

- Generischer Container, der nichts repräsentiert
- Häufig genutzt um **Elemente** zu gruppieren oder **zu stylen** (per **CSS**)
- Wenn semantisch aussagekräftigere Alternativen verfügbar sind, sollten diese verwendet werden:
 <nav>, <header>, <footer>, <article>, ...
- „**block**“-Element, d.h. der Container beginnt in einer neuen Zeile, was i.d.R. zu einem Zeilenumbruch führt

...

- Generischer Container wie div
- „inline“-Element, d.h. der Container kann auch innerhalb von Text genutzt werden (kein Zeilenumbruch)
- Beispiel (mit CSS-Klasse highlight)

```
<span class="highlight">Ich</span> bin hervorgehoben.
```

Übermittlung von Daten

- Ein Formular enthält Eingabefelder, die das Versenden von Informationen an einen Server erlauben

Beispiel

```
<form action="register_post.php" method="post">  
  <label for="l-user">Name:</label>  
  <input id="l-user" name="user" type="text"><br>  
  <label for="l-pw">Passwort:</label>  
  <input id="l-pw" name="pw" type="password"><br>  
  <input type="submit" value="Absenden!">  
</form>
```

Name:

Passwort:

<form>...</form>

- `action="..."` Wohin sollen die Daten gesendet werden
- `method="..."` Methode zur Übermittlung der Daten
 - > POST oder GET (wird noch näher behandelt)

Weitere Informationen: <https://developer.mozilla.org/docs/Web/HTML/Element/form>

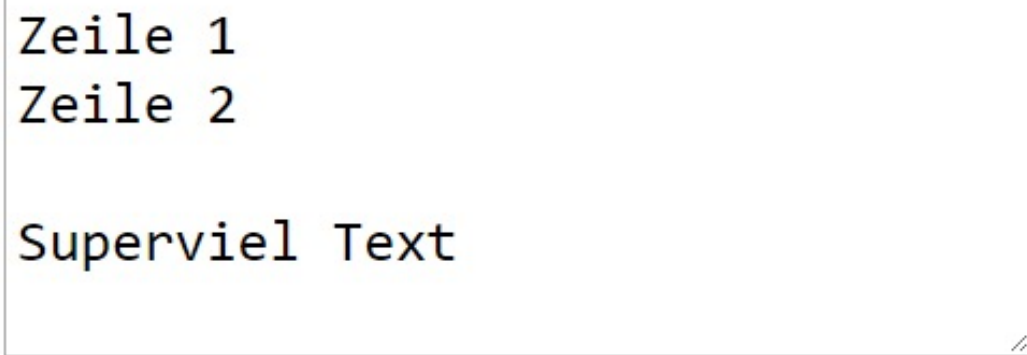
<textarea>...</textarea>

- Erlaubt die Eingabe von längerem und mehrzeiligem Text
- cols-Attribut bestimmt die Breite, rows die Höhe
 - > Besser über CSS festlegen
- Beispiel

```
<textarea name="textarea" rows="5" cols="30">Zeile 1
```

Zeile 2

Superviel Text</textarea>

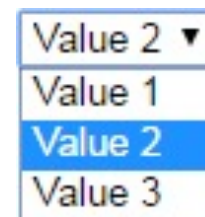


Weitere Informationen: <https://developer.mozilla.org/de/docs/Web/HTML/Element/textarea>

<select>...</select>

- Erlaubt die Auswahl eines Wertes (ähnlich zu Radiobuttons)
- <option>-Tags enthalten die Möglichkeiten
- Beispiel

```
<select name="select">  
  <option value="value1">Value 1</option>  
  <option value="value2" selected>Value 2</option>  
  <option value="value3">Value 3</option>  
</select>
```



Weitere Informationen: <https://developer.mozilla.org/docs/Web/HTML/Element/select>

HTML Elemente

Weitere nützliche HTML Elemente

- Seitenstrukturierung
 - > header, nav, aside, main, section, article, footer
- Textstrukturierung
 - > pre, figure, dt, dd, hr
- Textauszeichnung
 - > cite, var, time, del, ins
- Multimedia
 - > canvas, svg, iframe, audio, video
- Formulare
 - > fieldset, legend, progress

Weitere Informationen: <https://wiki.selfhtml.org/wiki/HTML>

HTML Elemente

Dokumentstruktur

- html, head, body

Kopfdaten

- title, meta, link, style, base

Seitenstrukturierung

- body, header, nav, aside, main, section, article, footer, address
- h1, h2, h3, h4, h5, h6 (Überschriften)

Textstrukturierung

- h1, h2, h3, h4, h5, h6 (Überschriften)
- p, pre, blockquote, figure, figcaption
- ol, ul, dl, li, dt, dd (Listen)
- hr
- div

Textauszeichnung

- a (Verweise)
- b, em, i, kbd, mark, s, small, strong, sub, sup, u
- cite, q (Zitate)
- dfn, abbr
- code, var, samp
- time
- ruby, rt, rp
- bdi, bdo
- br, wbr (Zeilenumbrüche)
- del, ins (Änderungsmarkierungen)
- span

Links (Verweise)

- a
- map, area

Tabellen

- table
- caption
- col, colgroup
- thead, tbody, tfoot
- tr
- th, td

Multimedia und Grafiken

- img, picture, map, area,
- canvas, svg, math
- iframe, embed, object, param
- audio, video, source, track

Formulare

- form, fieldset, legend, label, datalist
- input, button, select, optgroup, option, textarea, keygen
- output, progress, meter

Skripte

- script, noscript,
- canvas
- content, decorator, element, shadow, template

Interaktive Elemente

- details, summary
- dialog
- menu, menuitem, command

Besondere Attribute

- Universalattribute
- Eventhandler

Quelle: <https://wiki.selfhtml.org/wiki/HTML>

HTML5 – Event-Attribute

Einer der wichtigen Einsatzszenarien werden durch die Ereignis-Attribute in HTML erreicht

- Basiswerkzeug, um Dynamik in Web-Seiten einzubauen
- Fest verknüpft mit der später noch behandelten Fähigkeit, JavaScript-Code im Browser auszuführen
- Code wird bei bestimmten Handlungen ausgeführt
- Hinweis: Es gibt auch native JavaScript-Möglichkeiten
- Vielfältige Event-Attribute
 - > Fenster-Attribute sind mit dem Body-Element verknüpft (z.B. onunload)
 - > Formular-Attribute unterstützen Ereignisse in Formularen (z.B. onfocus)
 - > Tastatur-Attribute reagieren auf Tastenaktionen (z.B. onkeydown)
 - > Maus-Attribute auf Mausoperationen (z.B. onclick)
 - > Drag-Attribute sind mit Drag-Operationen der Maus verknüpft (z.B. ondrag)
 - > Clipboard-Attribute sind mit Markierungsaktionen verknüpft (z.B. oncopy)
 - > Media-Attribute auf Ereignisse bei der Behandlung von Bildern, Audio und Videos (z.B. onended)