

ZUSAMMENFASSUNG HTML Grundkurs – Wichtige HTML-Elemente

ÜBERSCHRIFTEN UND PARAGRAPHEN

Mithilfe von Überschriften kann **Text** besser **strukturiert** werden. Dafür werden HTML-Elemente verwendet. Das h1-Element stellt die größte Überschrift dar, das h6-Element die Überschrift mit der kleinsten Schriftgröße.

Beispiele:

`<h1>Das ist Überschrift 1</h1>`

`<h2>Das ist Überschrift 2</h2>`

`<h3>Das ist Überschrift 3</h3>`

`<h4>Das ist Überschrift 4</h4>`

`<h5>Das ist Überschrift 5</h5>`

`<h6>Das ist Überschrift 6</h6>`

Best Practice Hinweis

Die H1-Überschrift sollte man auf jeder HTML-Seite nur **einmal** vorfinden.

p-Tag

Mithilfe des p-Tags können einzelne Textabschnitte, also Paragraphen definiert werden.

Beispiel: `<p>In diesem Kurs lernst du wie man C# programmiert.</p>`
`<p>Wir beginnen komplett bei 0.</p>`

Auf der Website werden diese beiden Sätze nun nicht mehr in eine Zeile geschrieben, sondern als separate Paragraphen dargestellt.

Emmet Shortcut für Beispieltext

Dafür schreibt man an der gewünschten Stelle „Lorem“ und die Anzahl an Wörtern, die generiert werden soll.

Beispiel: **Lorem5** Generierter Text: **Lorem ipsum dolor sit amet.**

WICHTIG

Zwischenüberschriften sollten **nicht übersprungen** werden. Nach der Überschrift h2 sollte beispielsweise nicht direkt eine h4-Überschrift folgen, sondern stattdessen eine h3-Überschrift.

KOMMENTARE

Kommentare bieten eine Möglichkeit, innerhalb der HTML-Datei Notizen zu machen. So können bestimmte Stellen dokumentiert werden, damit man sich auch nach einer längeren Pause wieder darin zurechtfindet oder andere Entwickler sich einen Überblick verschaffen können.

Beispiel: `<!-- Das ist ein Kommentar -->`

Der Kommentar wird auf der Website nicht angezeigt, sondern ist nur für den Entwickler sichtbar.

Kommentare können auch verwendet werden, um vorübergehend bestimmte Abschnitte auszukommentieren, also auszublenden.

Text auskommentieren (Shortcut)

Dafür markiert man den gewünschten Text und drückt SHIFT + ALT + A auf Windows oder CMD + SHIFT + 7 auf dem MacOS. Mit der gleichen Tastenkombination kann der Kommentar dann auch wieder entfernt werden.

LINKS

a-Element

Mithilfe des a-Elements können in einer HTML-Datei Links eingefügt werden. Ein Link kann durch Emmet schneller erzeugt werden, indem man im Body den Buchstaben A eintippt und anschließend die Enter-Taste drückt. Dieser sieht folgendermaßen aus:

```
<a href=""></a>
```

href

Das href-Attribut gibt das Ziel des Links an.

Beispiel: ``

Zwischen die beiden Tags kann ein Linktext geschrieben werden.

Beispiel: `Bitte hier klicken!`

target-Attribut

Mithilfe des target-Attributs kann man den Link so gestalten, dass sich die Zielseite in einem **neuen Tab öffnet**. So verschwindet der Nutzer mit einem Klick auf den Link nicht von seiner ursprünglichen Seite.

Dem target-Attribut muss der Wert `_blank` zugewiesen werden.

Beispiel:

```
<a href="https://programmieren-starten.de/" target="_blank">Bitte hier klicken!</a>
```

Interne Links

Da ein HTML-Dokument aus mehreren Seiten bestehen kann, besteht die Möglichkeit durch interne Links auf diese zu verlinken. Man muss dafür also nicht mehr die vollständige URL angeben, sondern schreibt stattdessen ein Slash (/) und wählt eines der vorgeschlagenen HTML-Dokumente aus.

Beispiel:

```
<a href="/09_headings.html" target="_blank">Bitte hier klicken</a>
```

BILDER

img-Element

Mithilfe des `img`-Elements können Bilder in eine Website eingebunden werden.

Beispiel: ``

src=""

Bei `src` handelt es sich um die Abkürzung für Source, also Quelle. Dort weisen wir den Pfad zum Bild zu, das geladen werden soll.

alt=""

Über das Attribut `alt` kann man einen alternativen Text festlegen, der angezeigt werden soll, falls das bei source spezifizierte Bild nicht geladen werden kann.

LISTEN

Unordered Lists (ul-Element)

Dabei handelt es sich um Listen, die keine Ordnung beinhalten. Um so eine Liste zu erstellen, benötigt man das `ul`-Element.

Beispiel:

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

li-Element

`li` steht für list item. Je nach Anzahl der Einträge einer Liste benötigt man entsprechend viele `li`-Elemente.

Beispiel:

```
<ul>
  <li>Eintrag 1</li>
  <li>Eintrag 2</li>
  <li>Eintrag 3</li>
</ul>
```

Nützlicher Emmet Shortcut

Um `li`-Elemente schneller zu erzeugen, schreibt man `li` und ein `*`-Zeichen gefolgt von der Anzahl der Elemente, die man hinzufügen möchte und der Enter-Taste.

Beispiel: `li*4` -> erzeugt vier `li`-Elemente

Ordered Lists (ol-Element)

Bei den Ordered Lists unterliegen die List Items einer Ordnung. Ihre Reihenfolge wird durch Zahlen gekennzeichnet.

Beispiel:

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

type-Attribut

Die Art der Ordnung einer Ordered List kann selbst festgelegt werden über das `type`-Attribut. Dem `type`-Attribut können unterschiedliche Werte zugewiesen werden, die die Ordnung verändern.

Beispiel:

<ol type="1"> -> Der Standardfall mit den aufsteigenden Zahlen

1. Punkt Nummer 1
2. Punkt Nummer 2
3. Punkt Nummer 3

<ol type="i"> -> Elemente werden mit i aufgezählt

- i. Punkt Nummer 1
- ii. Punkt Nummer 2
- iii. Punkt Nummer 3

<ol type="a"> -> Elemente werden alphabetisch in Kleinbuchstaben aufgezählt

- a. Punkt Nummer 1
- b. Punkt Nummer 2
- c. Punkt Nummer 3

<ol type="A"> -> Elemente werden alphabetisch in Großbuchstaben aufgezählt

- A. Punkt Nummer 1
- B. Punkt Nummer 2
- C. Punkt Nummer 3

Description Lists (dl-Element)

Damit kann man eine Liste von Begriffspaaren und deren Beschreibung anlegen. Man kann diese beispielsweise nutzen, wenn man ein Glossar implementieren möchte.

Beispiel:

```
<dl>
</dl>
```

dt-Element

Damit legt man den Begriff eines Begriffspaares fest.

Beispiel:

```
<dl>
  <dt>HTML</dt>
</dl>
```

dd-Element

Damit legt man eine Beschreibung für ein Begriffspaar fest.

Beispiel:

```
<dl>
  <dt>HTML</dt>
  <dd>Erklärung was HTML ist</dd>
</dl>
```

TABELLEN

table-Element

Das table-Element benötigt man, um eine Tabelle zu erzeugen.

Beispiel:

```
<table>
</table>
```

tr-Element

Mit dem tr-Element erzeugt man eine Zeile in HTML.

Beispiel:

```
<table>
  <tr>

  </tr>
</table>
```

td-Element

Mit dem td-Element, welches für „table data“ steht, kann man Spalten einer Tabelle erzeugen.

Beispiel:

```
<table>
  <tr>
    <td>Zeile 1/Spalte 1</td>
  </tr>
</table>
```

Info: Jede Spalte ist so breit wie der breiteste Inhalt. Je länger also das Wort oder die Wörter einer Spalte sind, desto breiter fällt dementsprechend die Spalte aus.

th-Element

Mit dem th-Element kann man eine Überschriftszeile erzeugen.

Beispiel:

```
<tr>
  <th>Spalte 1</th>
  <th>Spalte 2</th>
  <th>Spalte 3</th>
</tr>
```

Wenn man eine Tabelle sauber erstellen möchte, sollte man diese auf zweiter Ebene noch mal grob in Head und Body untergliedern.

thead-Element

Der Inhalt, der im thead-Element steht, wird beim Drucken beispielsweise auf jeder ausgedruckten Seite wiederholt.

Beispiel:

```
<table>
  <thead>
    ...
  </thead>
  <tbody>
    ...
```

```
</tbody>
</table>
```

tfoot-Element

Das tfoot-Element stellt eine Footer-Zeile dar. Auch diese Zeile würde beim Drucken auf jeder Seite ganz unten platziert werden.

Beispiel:

```
<tfoot>
  <tr>
    <td>Fuß Spalte 1</td>
    <td>Fuß Spalte 2</td>
    <td>Fuß Spalte 3</td>
  </tr>
</tfoot>
```

HTML ENTITIES

< und >

Bestimmte Zeichen, die vom Browser verwechselt werden könnten, sollten mit HTML-Entities dargestellt werden. Zu diesen Zeichen gehören zum Beispiel das < und das >-Zeichen.

Beispiel 1: 3 < 4 => 3 < 4

Beispiel 2: 3 > 4 => 3 > 4

Non breaking space entity ()

Mit dieser Entity kann man beispielsweise zusätzliche Leerzeichen einfügen, die auch im Browser angezeigt werden.

Beispiel:

<p>Das ist ein Beispieltext</p> => <p>Das ist ein Beispieltext</p>

Copyright-Zeichen (©)

Mit dieser Entity kann ein Copyright-Zeichen im HTML-Dokument erzeugt werden.

Beispiel:

<p>Das ist ein Beispieltext © </p>

Link zur Übersicht weiterer HTML-Entities:

<https://www.key-shortcut.com/html-entities/alle-entitaeten>

UNTERSCHIED ZWISCHEN BLOCK-LEVEL UND INLINE ELEMENTEN

Jedes HTML-Element hat standardmäßig einen Display-Value. Dieser kann entweder auf **block** oder auf **inline** gesetzt sein.

Block-Level-Elemente

Beispiele

Überschriften (h1, h2, h3, etc.), Paragraphen, Listen

Eigenschaften

Block-Level-Elemente starten immer in einer neuen Zeile und der Browser fügt automatisch etwas Platz vor und nach dem Element ein. Sie versuchen also immer, so viel Platz wie möglich nach rechts und links zu allokalieren.

Inline-Elemente

Beispiele

Links, image-Elemente, b-Element

Eigenschaften

Inline-Elemente starten **nicht** in einer neuen Zeile. Sie allokalieren nur so viel Platz, dass der Inhalt des Elements sauber dargestellt werden kann.

DIVS UND SPANS

Bei den Div- und Span-Elementen handelt es sich um sogenannte Container. Innerhalb dieser kann man zusammengehörende Inhalte einer Website gruppieren und sie anschließend gemeinsam via CSS stylen.

Div

Divs sind Container, mithilfe derer man eine Website in bestimmte Sektionen unterteilen kann, um diese dann beispielsweise mit CSS zu stylen.

Sie sind block-level Elemente, was wiederum bedeutet, dass ein div Container immer in einer neuen Zeile steht und versucht so viel Platz in die Breite wie nur möglich zu allokalieren.

Beispiel:

```
<div style="background-color: yellow;">
  <h1>Willkommen</h1>
  <p>Hier lernst du Schritt für Schritt das Programmieren!</p>
</div>
```

Span

Wenn man einen gewissen Inhalt inline stylen möchte, verwendet man das Span-Element. Da span ein inline-Element ist, steht der modifizierte Text **nicht** in einer extra Zeile, sondern weiterhin mitten im Text.

Beispiel:

```
<p>Lorem ipsum, dolor <span style="font-size: 30px;">sit amet consectetur adipisicing elit.</span> Necessitatibus, sit?</p>
```

Im obigen Beispiel wird die Schriftgröße des Textes geändert, der zwischen den Tags steht.

SEMANTISCHE HTML-ELEMENTE

Man sollte versuchen den Einsatz von Divs und Spans weitestgehend zu limitieren. Das liegt daran, dass es sich bei diesen um generische Elemente handelt und keines dieser beiden Elemente etwas über den Inhalt oder den Zweck aussagt, den es enthält.

Aus diesem Grund gibt es auch semantische Elemente, die sich aus Tags zusammensetzen, die die Bedeutung bzw. den Zweck des Elements kommunizieren.

Beispiele:

Das p-Element kommuniziert durch seine Abkürzung, dass es sich um einen Paragraphen handelt. Das h1-Element kommuniziert durch seine Abkürzung, dass der Inhalt die wichtigste Überschrift dieser Seite ist.

Weitere semantische Elemente

header-Element

Header ist ein Container und zugleich ein Block-Level-Element.

Beispiel des header-Elements:

```
<header style="background-color: yellow;">
  <h1>Willkommen</h1>
  <p>Hier lernst du Schritt für Schritt das Programmieren!</p>
</header>
```

footer-Element

Im semantischen Footer-Element können beispielsweise Links zu Impressum und Datenschutzerklärung enthalten sein.

Beispiel des footer-Elements:

```
<footer style="background-color: yellow;">
  <a href="#">Impressum</a>
  <a href="#">Datenschutzerklärung</a>
</footer>
```

main-Element

Das Main-Element verwendet man für den Hauptinhalt der Seite. Alles, was zwischen Header und Footer liegt, kann in das Main-Element geschrieben werden.

Beispiel des main-Elements:

```
<main>
  <h2>Produkte</h2>
  <p>Lorem ipsum dolor sit amet.</p>

  <h2>Über uns</h2>
  <p>Lorem ipsum <span style="font-size: 30px;">sit amet.</span> sit?</p>
```



```
<p>Lorem ipsum dolor.</p>  
</main>
```

nav-Element

Das navigation-Element ist ein weiteres semantisches Element, mit welchem man Menüs umschließen kann.

Beispiel des nav-Elements:

```
<header style="background-color: yellow;">  
  <nav>  
    <ul>  
      <li>Startseite</li>  
      <li>Blog</li>  
    </ul>  
  </nav>  
  
  <h1>Willkommen</h1>  
  <p>Hier lernst du Schritt für Schritt das Programmieren!</p>  
</header>
```