

**B1B**

# Back in Business HTML

Thomas Zenger  
mail@thomas-zenger.de  
23.11.2016

# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort</b>	<b>8</b>
1.1	B1B - Was ist das? . . . . .	8
1.2	Motivation . . . . .	8
1.3	Literatur und Grundlagen . . . . .	8
<b>2</b>	<b>Einführung und Grundlagen</b>	<b>9</b>
2.1	Grundgerüst . . . . .	9
2.2	Dokumententyp . . . . .	9
2.3	Wurzelement <html> . . . . .	9
2.4	Grammatik . . . . .	9
2.5	Kommentar . . . . .	10
2.6	DOM - Document Object Model . . . . .	10
<b>3</b>	<b>Kopfelement &lt;head&gt;</b>	<b>11</b>
3.1	Element <title> . . . . .	11
3.2	Element <base> . . . . .	11
3.3	Element <link> . . . . .	11
3.4	Element <style> . . . . .	12
3.5	Element <script> . . . . .	12
3.6	Element <meta> . . . . .	12
3.6.1	name/content Metadaten . . . . .	12
3.6.2	http-equiv/content Metadaten . . . . .	13
3.6.3	Zeichencodierung Metadaten . . . . .	13
3.6.4	Beispiel <head> . . . . .	13
<b>4</b>	<b>Körperelement &lt;body&gt; - Strukturierung</b>	<b>14</b>
4.1	Seitenstrukturierung . . . . .	14
4.1.1	Element <body> . . . . .	14
4.1.2	Element <section> . . . . .	14
4.1.3	Element <article> . . . . .	14
4.1.4	Element <aside> . . . . .	14
4.1.5	Element <nav> . . . . .	14
4.1.6	Element <h1> bis <h6> . . . . .	15
4.1.7	Element <header> . . . . .	15
4.1.8	Element <footer> . . . . .	15
4.1.9	Element <address> . . . . .	15
4.2	Textstrukturierung . . . . .	15
4.2.1	Element <p> . . . . .	15
4.2.2	Element   . . . . .	15
4.2.3	Element <wbr> . . . . .	15
4.2.4	Leerzeichen &nbsp; . . . . .	15

4.2.5	Element <code>&lt;hr&gt;</code> . . . . .	15
4.2.6	Element <code>&lt;blockquote&gt;</code> . . . . .	16
4.2.7	Element <code>&lt;div&gt;</code> . . . . .	16
4.2.8	Element <code>&lt;main&gt;</code> . . . . .	16
4.2.9	Element <code>&lt;figure&gt;</code> & <code>&lt;figcaption&gt;</code> . . . . .	16
4.2.10	Element <code>&lt;ul&gt;</code> . . . . .	16
4.2.11	Element <code>&lt;ol&gt;</code> . . . . .	16
4.2.12	Element <code>&lt;dl&gt;</code> . . . . .	16
4.2.13	Semantisches HTML . . . . .	17
<b>5</b>	<b>Körperelement <code>&lt;body&gt;</code> - Auszeichnung</b>	<b>18</b>
5.1	Element <code>&lt;abbr&gt;</code> . . . . .	18
5.2	Element <code>&lt;cite&gt;</code> . . . . .	18
5.3	Element <code>&lt;code&gt;</code> & <code>&lt;pre&gt;</code> . . . . .	18
5.4	Element <code>&lt;kdb&gt;</code> & <code>&lt;samp&gt;</code> . . . . .	18
5.5	Element <code>&lt;dfn&gt;</code> . . . . .	18
5.6	Element <code>&lt;var&gt;</code> . . . . .	19
5.7	Elemente <code>&lt;bdo&gt;</code> & <code>&lt;bdi&gt;</code> . . . . .	19
5.8	Elemente <code>&lt;em&gt;</code> & <code>&lt;i&gt;</code> bzw. <code>&lt;strong&gt;</code> & <code>&lt;b&gt;</code> . . . . .	19
5.9	Element <code>&lt;mark&gt;</code> . . . . .	19
5.10	Element <code>&lt;q&gt;</code> . . . . .	19
5.11	Elemente <code>&lt;u&gt;</code> & <code>&lt;s&gt;</code> . . . . .	19
5.12	Elemente <code>&lt;del&gt;</code> & <code>&lt;ins&gt;</code> . . . . .	19
5.13	Elemente <code>&lt;sub&gt;</code> & <code>&lt;sup&gt;</code> . . . . .	19
5.14	Element <code>&lt;time&gt;</code> . . . . .	20
5.15	Element <code>&lt;small&gt;</code> . . . . .	20
5.16	Elemente <code>&lt;ruby&gt;</code> , <code>&lt;rp&gt;</code> & <code>&lt;rt&gt;</code> . . . . .	20
5.17	Element <code>&lt;span&gt;</code> . . . . .	20
<b>6</b>	<b>Tabellen</b>	<b>21</b>
6.1	einfache Tabellenstruktur . . . . .	21
6.2	Attribute . . . . .	21
6.2.1	Attribute <code>&lt;colspan&gt;</code> & <code>&lt;rowspan&gt;</code> . . . . .	21
6.3	Struktur . . . . .	22
6.4	Gruppierung . . . . .	22
6.5	Beschriftung . . . . .	22
6.5.1	Element <code>&lt;caption&gt;</code> . . . . .	22
6.5.2	Element <code>&lt;figcaption&gt;</code> . . . . .	22
<b>7</b>	<b>Verweise</b>	<b>23</b>
7.1	interne Links . . . . .	23
7.2	externe Links . . . . .	23
7.3	E-Mail-Links . . . . .	23
7.4	Link zu anderen Dateitypen . . . . .	24
7.5	Download Links . . . . .	24
7.6	Links auf die gleiche Seite . . . . .	24
7.7	Attribute . . . . .	24

---

<b>8</b>	<b>Grafiken</b>	<b>25</b>
8.1	Element <code>&lt;img&gt;</code> . . . . .	25
8.1.1	Beschriftung . . . . .	25
8.1.2	Attribute . . . . .	25
8.2	Verweissensitive Grafiken (Image Maps) . . . . .	26
8.2.1	Attribute <code>&lt;area&gt;</code> Element . . . . .	26
8.3	Element <code>&lt;picture&gt;</code> . . . . .	27
8.4	Mehrere Bildquellen ohne <code>&lt;picture&gt;</code> Element . . . . .	27
8.5	Favicon . . . . .	28
8.6	Vektorgrafiken . . . . .	28
8.6.1	Einfügen mit <code>&lt;img&gt;</code> . . . . .	28
8.6.2	Direkt Einbinden mit <code>&lt;svg&gt;</code> . . . . .	28
8.7	Element <code>&lt;canvas&gt;</code> . . . . .	29
<b>9</b>	<b>Multimedia</b>	<b>30</b>
9.1	Element <code>&lt;video&gt;</code> . . . . .	30
9.1.1	Attribute . . . . .	30
9.1.2	Element <code>&lt;track&gt;</code> . . . . .	30
9.2	Element <code>&lt;audio&gt;</code> . . . . .	31
9.2.1	Attribute . . . . .	31
<b>10</b>	<b>Aktive Elemente</b>	<b>32</b>
10.1	Element <code>&lt;embed&gt;</code> . . . . .	32
10.2	Element <code>&lt;object&gt;</code> . . . . .	32
10.3	Element <code>&lt;iframe&gt;</code> . . . . .	32
10.3.1	Attribut <code>sandbox</code> . . . . .	33
10.3.2	Attribut <code>seamless</code> . . . . .	33
<b>11</b>	<b>Changelog</b>	<b>34</b>

# Abbildungsverzeichnis

# Tabellenverzeichnis

# Listings

2.1	"HTML Grundgerüst"	9
2.2	"Kommentare in HTML"	10
3.1	"<head> Beispiel"	13
6.1	"einfache Tabelle"	21
7.1	"Navigation"	23
7.2	"Anker"	24
8.1	"Image Map Beispiel"	26
8.2	"picture Beispiel"	27
8.3	"mehrere Quellen ohne picture Element"	27
8.4	"Favicon Beispiel"	28
9.1	"Video"	30
9.2	"Audio"	31
10.1	"Beispiel embed"	32
10.2	"Beispiel object"	32
10.3	"Beispiel iframe"	32

# 1 Vorwort

## 1.1 B1B - Was ist das?

Dieses Skript soll als Nachschlagewerk und Cheat Sheet dienen und eine Gedächtnisstütze zu einem bereits vertieften Thema bieten. Es soll keine Fachliteratur ersetzen und dient auch nicht zum Erlernen der Thematik, da auf ausführliche Erklärungen größtenteils verzichtet wird.

## 1.2 Motivation

Diese Idee zu diesem Kompendium entstand während meines Informatikstudiums. Da bereits erlernte Techniken, wie z.B. Programmier- oder Scriptsprachen, mit dem Fortschreiten des Studiums in den Hintergrund traten, zu einem späteren Zeitpunkt jedoch wieder benötigt wurden, war es unerlässlich sich diese wieder ins Gedächtnis zu rufen. Aus diesem Grund entstand dieses Werk als eine Art “erweiterte Zusammenfassung”.

## 1.3 Literatur und Grundlagen

Folgende Werke fanden bei der Erstellung dieses Dokuments Beachtung. An dieser Stelle soll ausdrücklich erwähnt werden, dass sich diese Arbeit nicht als Plagiat oder Kopie genannter Literatur verstanden werden soll, sondern als private Lernhilfe und Zusammenfassung.

- J. Wolf (2016) - HTML5 und CSS3



## 2 Einführung und Grundlagen

### 2.1 Grundgerüst

```
1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>Fenstertitel</title>
6   </head>
7   <body>
8     <h1>Hauptüberschrift</h1>
9     <p>Text</p>
10  </body>
11 </html>
```

Listing 2.1: "HTML Grundgerüst"

### 2.2 Dokumententyp

Erste Angabe im Dokument. Definiert den Typ des Dokuments. Bei älteren Dokumententypen muss eine Dokumenttypdefinition (DTD) erfolgen.

HTML5     <!doctype html>

### 2.3 Wurzelement <html>

Nach der Definition des Doctypes folgt das Wurzelement <html>. Dies umschließt alle anderen HTML Elemente. Dem Wurzelement kann ein Sprachattribut mitgegeben werden (lang="de").

### 2.4 Grammatik

Groß- und Kleinschreibung findet in HTML keine Beachtung. Allerdings ist auf einen einheitlichen Stil zu achten.

#### HTML Elemente

Ein HTML Element besteht aus einem öffnenden und einem schließenden Tag. Zugehörige Inhalt wird von diesen Tags eingeschlossen.

## Standalone Elemente

Alleinstehende Tags haben keinen Inhalt und benötigen deswegen keinen schließenden Tag.

## Optionale Tags

Manche Elemente erlauben ein Weglassen des Ende Tags.

## Verschachtelung

HTML Elemente können ineinander verschachtelt werden. Auf ein korrektes schließen der Tags ist zu achten.

## Attribute

Den Elementen können verschiedene passende Attribute zugeordnet werden. Folgende Arten von Attributen können zugewiesen werden:

- Attribut mit fester Wertzuweisung
- Attribut mit freier Wertzuweisung
- Attribut mit freier Wertzuweisung ohne Konvention
- allein stehende Attribute

## 2.5 Kommentar

```
1 < ! - - Kommentarzeile - - >
2
3 < ! - -
4     Kommentar
5     Kommentar
6 - - >
```

Listing 2.2: "Kommentare in HTML"

## 2.6 DOM - Document Object Model

Ein HTML-Dokument kann als Baumstruktur interpretiert werden. Jedes Objekt wird dabei als Knoten dargestellt und kann mit Hilfe von z.B. JavaScript manipuliert werden.

## 3 Kopfelement <head>

Mit Ausnahme des <title> Elements enthält das Kopfelement nur Elemente, welche nicht direkt für die Darstellung im Browser verwendet werden. <head> enthält neben dem Titel zusätzlich noch Metadaten, Pfade zu Stylesheets, Scripte, usw. .

### 3.1 Element <title>

Jedes HTML Dokument sollte einen Titel enthalten. Dieser wird vom Browser angezeigt und unter anderem auch von Suchmaschinen verwendet. Das Element ist einzigartig.

### 3.2 Element <base>

Mit dem <base> Element wird die Basis-URL für die relative oder absolute Referenzierung definiert. Das Element ist einzigartig. Folgende Attribute können vergeben werden:

Attribut	Beschreibung
href	Basis-URL
target	Ziel des Verweises _self (aktuelles Fenster), _blank (neues Fenster) _parent (Elternfenster), _top (Hierarchisch höchstes Fenster)

### 3.3 Element <link>

Mit dem <link> Element können externe Ressourcen (z.B. CSS-Dateien) eingebunden werden. Das Element darf mehrfach auftreten. Es existieren sehr viele verschiedene Werte für das rel Attribut. Diese sollten noch mit Vorsicht genutzt werden.

Attribut	Beschreibung
rel	Typ der Verlinkung stylesheet (externe CSS Datei), icon (Favicon)
type	MIME Typ
href	URL Adresse
size	nur bei icon sinnvoll
media	Optimiert für bestimmten Mediatyp (Responsive)

### 3.4 Element <style>

Mit dem <style> Element können Style-Informationen (CSS) direkt in das HTML Dokument geschrieben werden. Innerhalb des <body> Elements muss das Attribut `scoped` verwendet werden. Das Element darf mehrfach auftreten.

Attribut	Beschreibung
<code>media</code>	Optimiert für bestimmten Mediatyp (Responsive)
<code>scoped</code>	Style gilt nur im Eltern-Element und Kind-Elementen
<code>type</code>	MIME Type

### 3.5 Element <script>

Mit dem <script> Element können Skripte (JavaScript) in das HTML Dokument eingebunden werden. Das Skript kann direkt zwischen die Tags geschrieben werden oder es wird eine externe Datei referenziert. Dabei der Referenzierung darf das Element keinen Inhalt vorweisen. Das Element darf mehrfach auftreten und ebenfalls im <body> vorkommen. Wegen einer möglichen Verlängerung der Ladezeiten einer Seite, ist es sinnvoll, den Code erst am Ende der Seite einzufügen. Sollte JavaScript deaktiviert sein können mit dem <noscript> alternative Ausgaben gemacht werden.

Attribut	Beschreibung
<code>async</code>	asynchrone Ausführung (nur extern)
<code>charset</code>	Zeichencodierung (nur extern)
<code>defer</code>	Erst Website parsen, dann Skript ausführen (nur extern)
<code>src</code>	URL Adresse
<code>type</code>	MIME Typ

### 3.6 Element <meta>

Mit dem <meta> Element können zusätzliche Metainformationen zur Website angegeben werden. Viele Angaben folgen keinem Standard. Unter <https://wiki.whatwg.org/wiki/MetaE> können sinnvolle Erweiterungen nachgeschlagen werden.

#### 3.6.1 name/content Metadaten

Die Name/Content Paarungen sollten nicht für persönliche Informationen verwendet werden, sondern nur Informationen zum HTML Dokument enthalten.

```
<meta name=' 'name' ' content=' 'content' '>
```

Name	Beschreibung
<code>author</code>	Optimiert für bestimmten Mediatyp (Responsive)
<code>keywords</code>	Style gilt nur im Eltern-Element und Kind-Elementen
<code>description</code>	Beschreibung der Website

### 3.6.2 http-equiv/content Metadaten

Die http-equiv Angaben (Pragma-Direktive) waren für den Webserver zur Kommunikation gedacht, jedoch liegt es am Browser wie mit diesen Daten umgegangen wird, da Webserver eigentlich keine HTML-Dokumente parsen.

```
<meta http-equiv='refresh' content='5'>
```

Name	Beschreibung
refresh	Seite neu laden (Zeitangabe als Content)
refresh	Nach gewisser Zeit weiterleiten (Zeit; URL als Content)
expires	Seite von Originaladresse laden (Zeit als Content)

### 3.6.3 Zeichencodierung Metadaten

Die Zeichenkodierung sollte immer angegeben werden. Mit ihrer Hilfe wird sichergestellt, dass alle Zeichen korrekt dargestellt werden.

```
<meta charset="UTF-8">
```

Veraltet:

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

### 3.6.4 Beispiel <head>

```

1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <meta name='author' content='Thomas Zenger'>
6     <meta name='keywords' content='Beispiel'>
7     <meta http-equiv='refresh' content='10'>
8     <title>Fenstertitel</title>
9     <base href='www.thomas-zenger.de' target='_blank'>
10    <link rel='stylesheet' type='text/css' href='style.css'>
11    <style type='text/css'>
12      p {
13        background-color: #000000;
14      }
15    </style>
16    <script type='text/javascript'>
17      window.onload=alert("Ein JavaScript!")
18    </script>
19    <script type="text/javascript" src="script.js"></script>
20  </head>
21  <body>
22    <h1>Hauptüberschrift</h1>
23    <p>Text</p>
24  </body>
25 </html>

```

Listing 3.1: "<head> Beispiel"

## 4 Körperelement `<body>` - Strukturierung

### 4.1 Seitenstrukturierung

HTML5 stellt Elemente zur Verfügung, mit deren Hilfe eine sinnvolle Unterteilung einer Website möglich ist. Dies verringert den Einsatz von nicht semantischen `<div>` Elementen und `class` Attributen.

#### 4.1.1 Element `<body>`

Das `<body>` Element kennzeichnet den Dokumentenkörper. Alle Elemente zwischen den `body` Tags werden dargestellt. Das Element unterstützt keine Attribute.

#### 4.1.2 Element `<section>`

Mit dem `<section>` Element kann die Website in themenbezogene Abschnitte eingeteilt werden. Eine Gliederung kann z.B. nach Inhalt oder Sinn erfolgen.

#### 4.1.3 Element `<article>`

Das Element `<article>` verhält sich ähnlich wie das `<section>` Element. Allerdings sollte dieses Element für kleinere zusammengehörige Bereiche verwendet werden (z.B. einzelne Newsmeldungen/Blogeinträge). Charakteristisch für das Element ist die Mehrfache Verwendung für gleichartige Bereiche.

#### 4.1.4 Element `<aside>`

Das Element `<aside>` bietet die Möglichkeit zusätzliche Informationen bereitzustellen. Semantisch kann es z.B. für eine Seitenleiste oder zusätzliche Informationen verwendet werden.

#### 4.1.5 Element `<nav>`

Das Element `<nav>` dient zur Einteilung der Navigationselemente. Es wird empfohlen das Element in Kombination mit einer ungeordneten Liste für die Hauptnavigation zu verwenden. Es wird abgeraten innerhalb des `nav` Elements auf externe Seiten zu verlinken.

### 4.1.6 Element <h1> bis <h6>

Das Element stellt Überschriften unterschiedlicher Ordnung zur Verfügung. Diese Elemente dienen der inhaltlichen Struktur und nicht dem hervorheben von Text.

### 4.1.7 Element <header>

Mit dem Element <header> ist es möglich einen Kopfbereich zu erstellen. Dieser beeinflusst nicht die hierarchische Struktur der Website. Das Element darf nicht innerhalb von <footer>, <address> oder eines anderen <header> verwendet werden.

### 4.1.8 Element <footer>

Mit dem Element <footer> ist es möglich einen Fußbereich zu erstellen. Dieser beeinflusst nicht die hierarchische Struktur der Website. Das Element darf nicht innerhalb eines anderen <footer> Elements verwendet werden.

### 4.1.9 Element <address>

Das Element <address> ist für die Kontaktdaten des Autors gedacht. Es ist nicht für postalische Adressen zu verwenden.

## 4.2 Textstrukturierung

### 4.2.1 Element <p>

Das <p> Element umschließt einen Textabsatz mit Fließtext. Es können innerhalb dieses Elements keine Überschriften und Sektionselemente verwendet werden.

### 4.2.2 Element <br>

Das <br> Element erzwingt einen Zeilenumbruch. Das Element ist ein alleinstehender Tag.

### 4.2.3 Element <wbr>

Das <wbr> Element ermöglicht einen optionalen Zeilenumbruch an dieser Stelle. Das Element ist ein alleinstehender Tag.

### 4.2.4 Leerzeichen &nbsp;

Mehrere Leerzeichen werden mit &nbsp; erreicht. Außerdem wird ein Zeilenumbruch an dieser Stelle verhindert.

### 4.2.5 Element <hr>

Mit dem Element <hr> wird eine inhaltliche Trennung erzeugt. Es darf nicht innerhalb von Paragraphen oder Überschriften verwendet werden.

### 4.2.6 Element <blockquote>

Mit dem Element <blockquote> werden Zitate gekennzeichnet. Mit dem Attribut `cite` kann die Quelle des Zitats angegeben werden. Leider können aktuelle Browser diese Quelle noch nicht darstellen.

### 4.2.7 Element <div>

Mit dem <div> Element wird ein allgemeiner Bereich definiert. Dies bewirkt, dass eine neue Zeile begonnen wird. Mit Klassen und IDs kann der Bereich mit CSS verändert (gestylt) werden. HTML5 erlaubt keine Attribute für <div>.

### 4.2.8 Element <main>

Das Element <main> ist für den Hauptinhalt der Seite gedacht und darf im gesamten Dokument nur einmal vorkommen. Es ist ein reines Gruppierungselement und hat somit keinen Einfluss auf die Struktur. Für ältere Browser muss ein Workaround geschrieben werden.

### 4.2.9 Element <figure> & <figcaption>

Das Element <figure> dient als Elternelement für Inhalte wie Grafiken, Bilder, Videos, etc. und ermöglicht damit ein Absetzen dieses Elements gegenüber eines Fließtextes. Das optionale Element <figcaption> ermöglicht eine Beschriftung des Inhalts.

### 4.2.10 Element <ul>

Mit dem Element <ul> werden ungeordnete Listen realisiert. Diesen Listen können weitere Elemente mit Hilfe von <li> zugeordnet werden. Desweiteren können diese Listen verschachtelt werden.

### 4.2.11 Element <ol>

Mit dem Element <ol> werden geordnete Listen realisiert. Diesen Listen können weitere Elemente mit Hilfe von <li> zugeordnet werden. Desweiteren können diese Listen verschachtelt werden. Die Nummerierung der Liste kann mit dem Attribut `reversed="reversed"` umgekehrt werden. Mit dem Attribut `value="wert"` kann die Nummerierung eines Listenelements oder der kompletten Liste geändert werden.

### 4.2.12 Element <dl>

Mit dem Element <dl> werden Beschreibungslisten realisiert. Das Element <dt> fügt einen Ausdruck hinzu, während <dd> die zugehörige Beschreibung einfügt.



### 4.2.13 Semantisches HTML

Sematisch korrektes HTML ist Grundlage für eine ordentliche Struktur einer Website. Statt auf `<div>` Elemente sollte, wo es möglich ist, auf strukturierende HTML Elemente gesetzt werden.

## 5 Körperelement `<body>` - Auszeichnung

HTML Elemente zur Auszeichnung können innerhalb eines Textes auf Buchstaben, Wörter und Sätze angewandt werden. Diese Elemente erzeugen keinen Zeilenumbruch oder Absatz, sondern verhalten sich nur gemäß ihrer definierten Semantik. Trotz der leichten optischen Veränderungen durch diese Elemente erfolgt eine Formatierung nur durch CSS.

### 5.1 Element `<abbr>`

Mit dem `<abbr>` Element werden Abkürzungen gekennzeichnet. Das Attribut `title` kann angegeben werden. Der Browser stellt dann bei einem Mouse Over den übergebenen Titel dar.

### 5.2 Element `<cite>`

Mit dem `<cite>` Element werden Titel von Werken hervorgehoben. Das Attribut `title` kann angegeben werden. Der Browser stellt dann bei einem Mouse Over die Informationen dar.

### 5.3 Element `<code>` & `<pre>`

Mit dem Element `<code>` wird Computercode innerhalb eines Textes gekennzeichnet. Bei mehreren Zeilen Code sollte das `<code>` Element von einem `<pre>` Element umschlossen werden. Bei einem präformatierten Text werden keine Whitespace Zeichen zusammengefasst.

### 5.4 Element `<kdb>` & `<samp>`

Mit den Elementen `<kdb>` und `<samp>` werden Tastatureingaben bzw. Programmausgaben in einem Fließtext ausgezeichnet.

### 5.5 Element `<dfn>`

Mit dem Element `<dfn>` kann ein Ausdruck notiert werden, der nachfolgend definiert wird. Es ist nicht dafür gedacht komplette Definitionen zu markieren. Es wird empfohlen das Attribut `title` mit dem selben Inhalt zu verwenden.

## 5.6 Element <var>

Mit dem Element <var> wird eine Variable ausgezeichnet.

## 5.7 Elemente <bdo> & <bdi>

Mit den Elementen <bdo> und <bdi> sind zum Ändern der Textrichtung gedacht. <bdi> verlässt sich dazu auf Unicode fähige Browser, welche die Richtung selbstständig anpassen, wohingegen die Textrichtung beim Element <bdo> mit dem Attribut `dir` explizit angegeben werden muss.

## 5.8 Elemente <em> & <i> bzw. <strong> & <b>

Diese Elemente können zum Betonen und Hervorheben eines Textes genutzt werden. Das Element <em> sollte für Passagen, die beim Sprechen besonders betont werden, verwendet werden. <strong> hingegen dient zum Auszeichnen mit einer besonderen Wirkung. Mit dem Element <i> werden spezielle Fachausdrücke gekennzeichnet. Schlüsselwörter und Namen sollten mit dem <b> Element notiert werden.

## 5.9 Element <mark>

Das Element <mark> dient zum markieren von Text (vgl. Textmarker).

## 5.10 Element <q>

Das Element <q> dient dazu innerhalb eines Textes ein Zitat zu kennzeichnen.

## 5.11 Elemente <u> & <s>

Die Elemente <u> und <s> dienen zum Unterstreichen bzw. Durchstreichen von Text. Das Element <s> sollte für veralteten oder nicht mehr korrekten Inhalt verwendet werden. <u> dient in HTML5 zur Unterstreichung von Eigennamen (chin. Schrift) oder zum markieren von fehlerhaften Wörtern und Passagen.

## 5.12 Elemente <del> & <ins>

Die Elemente <del> und <ins> dienen zum Markieren von bearbeiteten Text. Das <del> Element wird für gelöschten Text verwendet, während mit dem Element <ins> neu eingefügte Inhalte notiert werden.

## 5.13 Elemente <sub> & <sup>

Die Elemente <sub> und <sup> dienen zum Hoch- bzw. Tiefstellen von Text.

## 5.14 Element <time>

Mit dem <time> Element werden Zeitangaben notiert. Die maschinenlesbare Form dieser Zeitangabe sollte mit dem Attribut `datetime` angegeben werden.

## 5.15 Element <small>

Das Element <small> dient zur Kennzeichnung von Informationen die im Kleingedruckten stehen.

## 5.16 Elemente <ruby>, <rp> & <rt>

Ruby Annotation mit Klammern <rp> und Text <rt>

## 5.17 Element <span>

Das Element <span> dient zum Auszeichnen von einzelnen Textpassagen. Das Element sollte nur verwendet werden, falls kein anderes Element semantisch besser passt.

# 6 Tabellen

Eine Tabelle in HTML dient nicht der Formatierung sondern nur der logischen Anordnung der Daten. Die Formatierung erfolgt wie üblich über CSS.

## 6.1 einfache Tabellenstruktur

Eine einfache Tabellenstruktur besteht aus folgenden Elementen:

- Tabelle `<table>`
- Reihe `<tr>`
- Kopfzeile `<th>`
- Zelle `<td>`

```
1 <table>
2   <tr>   <th>H1</th>   <th>H2</th>   <th>H3</th>   </tr>
3   <tr>   <td>C11</td>   <td>C12</td>   <td>C13</td>   </tr>
4   <tr>   <td>C21</td>   <td>C22</td>   <td>C23</td>   </tr>
5 </table>
```

Listing 6.1: "einfache Tabelle"

## 6.2 Attribute

In HTML5 wurden fast alle Tabellenattribute entfernt. Lediglich folgende Eigenschaften stehen zur Verfügung:

<code>&lt;table&gt;</code>	<code>border="1"</code> (Rahmen wird angezeigt)
<code>&lt;td&gt;</code> & <code>&lt;th&gt;</code>	<code>colspan</code>
	<code>rowspan</code>
	<code>scope="col"</code> (Überschrift für Spalte)
	<code>scope="row"</code> (Überschrift für Zeile)

### 6.2.1 Attribute `<colspan>` & `<rowspan>`

Mit Hilfe der Attribute `<colspan>` und `<rowspan>` können Spalten bzw. Zeilen zusammengefasst werden. Der übergebene Zahlenwert des Attributs beschreibt die Anzahl der Zellen.

## 6.3 Struktur

Zur Strukturierung der Tabelle stehen folgende drei Elemente zur Verfügung:

- Kopfbereich `<thead>`
- Datenbereich `<tbody>`
- Fußbereich `<tfoot>`

## 6.4 Gruppierung

Mit den Elementen `<colgroup>` & `<col>` können einzelne Bereiche einer Tabelle ausgezeichnet werden. Die Auszeichnung muss direkt hinter dem öffnenden `<table>` Element erfolgen. Spaltenübergreifungen können mit dem Attribut `span` erreicht werden.

## 6.5 Beschriftung

### 6.5.1 Element `<caption>`

Nach dem öffnenden `<table>` Tag erfolgt die Beschriftung der Tabelle mittels des `<caption>` Elements. Es kann nur eine Beschriftung pro Tabelle verwendet werden.

### 6.5.2 Element `<figcaption>`

Die Verwendung erfolgt äquivalent zu allen anderen Elementen, welche in ein `<figure>` Elternelement verpackt werden können.

# 7 Verweise

Hyperlinks dienen zur Strukturierung des Projektes und um auf andere Seiten (auch externe Seiten oder Dokumente sind erlaubt) zu verlinken. Außer interaktive Elemente, Links, Formularelemente, audio, video dürfen fast alle Elemente anstatt eines Textes als Link verwendet werden.

```
<a href =''www.thomas-zenger.de''> Startseite </a>
```

## 7.1 interne Links

Bei internen Links, d.h. links auf dem eigenen Webspace, reicht in der Regel eine relative Pfadangabe zur Verlinkung aus.

```
1 <nav>
2   <a href=''pages/about.html''> Über mich </a>
3   <a href=''pages/links.html''> Links </a>
4   <a href=''pages/impressum.html''> Impressum </a>
5 </nav>
```

Listing 7.1: "Navigation"

## 7.2 externe Links

Die Notierung eines externen Links erfolgt wie die eines Internen. Einziger Unterschied ist, dass eine absolute URL erforderlich ist. Mit Hilfe des `target` Attributs kann das Ziel des Links im Browser festgelegt werden.

```
<a target=''_blank'' href =''https://www.heise.de''> Startseite </a>
```

Attributwerte:

<code>_blank</code>	Neues Fenster oder Tab
<code>_self</code>	aktuelles Fenster
<code>_parent</code>	Eltern-Fenster
<code>_top</code>	oberste Fenster-Ebene

## 7.3 E-Mail-Links

Ein Link auf eine E-Mail-Adresse wird mit der Standardanwendung für eMails geöffnet.

```
<a href =''mailto:mail@thomas-zenger.de''>Kontakt aufnehmen</a>
```

## 7.4 Link zu anderen Dateitypen

Ob ein Link zu einem anderen Dateityp funktioniert und dieser Dateityp korrekt dargestellt wird, ist browserabhängig. Als Empfehlung gilt, dass weit verbreitete Dateitypen verwendet werden sollten.

## 7.5 Download Links

Mit HTML5 ist es möglich Links unabhängig vom Inhalt als Downloadverweise auszuzeichnen. Das Attribut `download` kann als alleinstehendes Attribut verwendet werden oder es kann ein Name angegeben werden, unter welchem die Datei gespeichert werden soll.

```
<a href =''/download/antrag.pdf'' download>Antrag herunterladen</a>
```

## 7.6 Links auf die gleiche Seite

Um auf bestimmte Bereiche einer Website zu Verweisen müssen Anker gesetzt werden. Diese können über einen Hyperlink angesprochen werden. Ebenso ist es möglich einen Anker einer anderen Website zu referenzieren.

```
1 <h1 id="top">Uberschrift</h1>
2 ...
3 <a href="#top">Nach oben</a>
```

Listing 7.2: "Anker"

## 7.7 Attribute

Attribut	Beschreibung
<code>download</code>	Ziel zum Download anbieten
<code>href</code>	URL
<code>hreflang</code>	Sprache des verlinkten Dokuments
<code>media</code>	Medientyp
<code>rel</code>	Typ der Verlinkung
<code>target</code>	Verweisziel wo öffnen
<code>type</code>	MIME-Type



# 8 Grafiken

## 8.1 Element `<img>`

Die Attribute `source` und `alt` müssen beim Standalone Tag `<img>` angegeben werden. `src` gibt die Quelle (Pfad) des Bildes an, während mit `alt` eine Beschreibung des Bildes angegeben wird. Falls das Bild nicht geladen werden konnte, wird stattdessen die Beschreibung angezeigt.

```
<img src='' /pic/sample.jpg'' alt=''Ein Beispielbild''>
```

### 8.1.1 Beschriftung

#### Attribut `title`

Wird das `title` Attribut gesetzt, zeigt der Browser bei einem Mouse Over den angegebenen Text.

#### Elemente `<figure>` & `<figcaption>`

Mit den beiden Elementen `<figure>` & `<figcaption>` kann eine Bildunterschrift gesetzt werden. Das Bild wird mit `<figure>` eingeschlossen und `<figcaption>` direkt nach dem öffnenden Tag angegeben (Beschriftung erscheint über dem Bild) oder direkt vor dem schließenden Tag (Beschriftung erscheint unter dem Bild). Bilder können auch mit Hilfe von `<figcaption>` zusammengefasst werden. Allerdings ist pro `<figcaption>` Element nur eine Caption erlaubt.

### 8.1.2 Attribute

Attribut	Beschreibung
<code>alt</code>	alternativer Text
<code>height</code>	Höhe der Grafik
<code>ismap</code>	boolscher Wert, serverseitige Image Map
<code>src</code>	Verweis zur Quelldatei
<code>usemap</code>	Name einer Image Map (clientseitig)
<code>width</code>	Breite der Grafik

#### Anmerkung:

`height` und `width` sollten immer angegeben werden. Der Browser reserviert beim Laden des HTML Codes den Platz für die Grafiken, so dass es nachträglich nicht zu Verschiebungen kommt. Es ist auch möglich Grafiken mithilfe der Attribute zu skalieren. Dies sollte allerdings vermieden werden, da dies nur die Darstellung im Browser beeinflusst und keinen Einfluss auf die Originaldatei hat.

## 8.2 Verweissensitive Grafiken (Image Maps)

Eine Image Map besteht aus drei Teilen:

- Einem Bild `<img>` mit dem Attribut `usemap="#mapname"`
- Image Map Element mit angegebenen Namen `<map name="mapname">`
- Koordinaten der verweissensitiven Bereiche. Für jeden Bereich wird ein `area` Element notiert.

```

1 <img src='../pic/direction.jpg' alt="ImageMap" width="100" height="
  50" usemap="#direc">
2
3 <map name="direc">
4   <area shape="rect" coords="0,0,50,50"
5     href="left.html" alt="links" title="links">
6   <area shape="rect" coords="50,0,100,50"
7     href="right.html" alt="rechts" title="rechts">
8   </map>

```

Listing 8.1: "Image Map Beispiel"

### 8.2.1 Attribute `<area>` Element

#### Attribute `shape` & `cord`

Mit dem Attribut `shape` wird die Form des Bereiches deklariert. Die Koordinaten der Bereiche werden im `coords` Attribut aufgeführt. Zur Auswahl stehen:

- `rect` (Rechteck)  
Es müssen die linke obere und rechte untere Ecke angegeben werden (`x1`, `y1`, `x2`, `y2`).
- `circle` (Kreis)  
Es müssen der Mittelpunkt und der Radius angegeben werden (`x`, `y`, `r`).
- `poly` (Polygon)  
Es können beliebig viele Punkte angegeben werden. Diese werden durch gerade Linien miteinander verbunden. Zum Schließen des Polygons müssen die Koordinaten des ersten Elements nochmals zum Schluss angegeben werden.

#### Attribute `href` & `alt`

Die Attribute `href` & `alt` sollten immer paarweise auftreten. Mit diesen Attributen wird die Verlinkung realisiert.

**weitere Attribute**

Attribut	Beschreibung
download	Ziel zum Download anbieten
hreflang	Sprache des verlinkten Dokuments
media	Medientyp
rel	Typ der Verlinkung
target	Verweisziel wo öffnen
type	MIME-Type

**8.3 Element <picture>**

Das Element <picture> dient als Container Element für mehrere Bidler. Die einzelnen Quellangaben enthalten eine Abfrage, sodass das richtige Bild zugeordnet werden kann. Das Lesen der Quellangaben erfolgt von oben nach unten. Am Ende sollte für ältere Browser noch ein <img> Element notiert werden.

```

1 <picture>
2   <source media="(min-width: 1024px)" srcset="pic/1024.jpg">
3   <source media="(min-width: 640px)" srcset="pic/640.jpg">
4   <source media="(min-width: 480px)" srcset="pic/480.jpg">
5   <!-- Fallback alte Browser -->
6   <img src='pic/480.jpg' alt="title">
7 </picture>

```

Listing 8.2: "picture Beispiel"

**Attribute**

Attribut	Beschreibung
srcset	Quellangabe (kann auch mehrere Quellen enthalten)
media	Media Query
type	MIME-Type
sizes	Breite des Bildes

**8.4 Mehrere Bildquellen ohne <picture> Element**

```

1 
5 </p>

```

Listing 8.3: "mehrere Quellen ohne picture Element"

## 8.5 Favicon

```

1 <head>
2   <link rel="icon" sizes="16x16" href="pic/favicon.ico">
3   <link rel="apple-touch-icon" sizes="120x120" href="pic/touch.png"
      type="image/png">
4 </head>

```

Listing 8.4: "Favicon Beispiel"

## 8.6 Vektorgrafiken

### 8.6.1 Einfügen mit <img>

```
<img src='../pic/sample.svg' width='50' height='50' alt='Vektrografik'>
```

### 8.6.2 Direkt Einbinden mit <svg>

Mit Hilfe des <svg> Elements können SVG Grafiken direkt im Browser gezeichnet werden. Dabei müssen die XML Regeln beachtet werden. Der Ursprung des Koordinatensystems liegt dabei in der oberen linken Ecke. Die Größe wird im öffnenden Tag angegeben.

#### Grafische Elemente

Attribut	Beschreibung
<path .../>	Pfad
<circle .../>	Kreis, beschrieben durch Mittelpunkt und Radius (cx, cy, r)
<ellipse .../>	Ellipse, beschrieben durch Mittelpunkt, vertikalen und horizontalen Radius (cx, cy, rx, ry)
<rect .../>	Rechteck, beschrieben durch linke obere Ecke, Länge und Breite (x, y, width, height)
<line .../>	Linie, beschrieben durch zwei Punkte (x1, y1, x2, y2)
<polygon .../>	Polygon, Eckpunkte werden dem point Attribut übergeben (points='0,0 10,10 20,0')
<polyline .../>	Wie Polygon, jedoch werden Start- und Endpunkt nicht miteinander verbunden.
<text .../>	Text, beschrieben durch Position, Schriftfamilie, Schriftgröße (x="5" y="80" fill="red" font-size="60")
<g>...</g>	Zusammenfassen von graphischen Elementen

**Attribute**

Attribut	Beschreibung
stroke-width	Linienstärke Rahmen
stroke	Rahmenfarbe
stroke-linecap	Aussehen
stroke-dasharray	Gestrichelte/Gepunktete Linie
opacity	Transparenz
fill	Füllfarbe
transform	Transformation der Grafik

**weitere Attribute und Informationen auf:** <https://wiki.selfhtml.org/wiki/SVG>

## 8.7 Element <canvas>

Das <canvas> Element stellt eine “Leinwand” zur Verfügung. Auf dieser kann mit Hilfe von JavaScript gezeichnet werden.

```
<canvas id="canvas1" width="400" height="200">  
  canvas element not supported  
</canvas>
```

# 9 Multimedia

## 9.1 Element <video>

Mit dem <video> Element ist es möglich Videos ohne Plug-In im Browser abzuspielen. Unterstützt werden die Formate **MP4**, **WebM** und **OGG**. Allerdings werden nicht alle Formate von jedem Browser unterstützt (außer MP4).

```
1 <video width="640" controls autoplay>
2   <source src="sample.webm" type="video/webm">
3   <source src="sample.mp4" type="video/mp4">
4   <source src="sample.ogv" type="video/ogg">
5   Not Supported
6 </video>
```

Listing 9.1: "Video"

### 9.1.1 Attribute

Attribut	Beschreibung
auto-play	Startet Video sobald geladen
controls	Blendet Steuerelemente ein
height	Höhe
loop	Endlosschleife
muted	ohne Ton
poster	Vorschaubild
preload	alles (auto) nur Metadaten (meta) Nichts (none)
src	URL
type	Videoformat
width	Breite

### 9.1.2 Element <track>

Gibt eine Untertiteldatei an.

```
<track src="untertitel_en.vtt" kind="subtitles" srclang="uk" label="Englisch">
```

## 9.2 Element <audio>

Mit dem <audio> Element ist es möglich Audiodateien ohne Plug-In im Browser abzuspielen. Unterstützt werden die Formate **MP3**, **WAV** und **OGG**.

```
1 <audio controls>
2   <source src="sound/tusch.ogg" type="audio/ogg">
3   <source src="sound/tusch.mp3" type="audio/mpeg">
4   Not Supported
5 </audio>
```

Listing 9.2: "Audio"

### 9.2.1 Attribute

Attribut	Beschreibung
auto-play	Startet Audio sobald geladen
controls	Blendet Steuerelemente ein
loop	Endlosschleife
muted	ohne Ton
preload	alles (auto) nur Metadaten (meta) Nichts (none)
src	URL
type	Videoformat

# 10 Aktive Elemente

## 10.1 Element <embed>

Das stand alone Element <embed wird für Inhalte verwendet, für die ein Plug In benötigt wird.

```
1 <embed width="200" height="50"  
2   type="application/x-shockwave-flash"  
3   src="animation.swf" quality="high">
```

Listing 10.1: "Beispiel embed"

## 10.2 Element <object>

Das <object> Element ermöglicht ebenfalls das Einbinden von aktiven Inhalten. Im Gegensatz zum <embed> Element hat es einen schließenden Tag. Dies ermöglicht die Angabe eines alternativen Inhalts, falls das Element nicht unterstützt wird. Parameter werden mithilfe des <param> Elements angegeben.

```
1 <object width="200" height="50" type="application/x-shockwave-flash"  
  ">  
2   <param name='movie' src="animation.swf">  
3   <param name='quality' quality="high">  
4   Not Supported  
5 </object>
```

Listing 10.2: "Beispiel object"

## 10.3 Element <iframe>

Das <iframe> Element ermöglicht ebenfalls das Einbinden von Inhalten. In der Praxis wird es überwiegend dazu genutzt externe HTML Dokumente in das aktuelle Dokument einzubinden.

```
1 <iframe height="320" width="680"  
2   src="schleichwerbung.html">  
3   Not Supported  
4   <a href="schleichwerbung.html">Schleichwerbung</a>  
5 </iframe>
```

Listing 10.3: "Beispiel iframe"



### 10.3.1 Attribut `sandbox`

Das Attribut `sandbox` verhindert das Ausführen von Skripten, Links, etc. welche aus dem Frame hinausgehen, auf Cookies zugreifen oder Formulare absenden.

### 10.3.2 Attribut `seamless`

Das Attribut `seamless` veranlasst, dass eine Ressource nicht eingebettet sondern inkludiert ist. Die Ressource verhält sich wie ein Block Element.

# 11 Changelog

**V1.0 - 2016.11.23**

- Erstveröffentlichung