

O'REILLY®

5. Auflage

# HTML5

kurz & gut

---

O'REILLYS TASCHENBIBLIOTHEK



Jennifer Niederst Robbins  
Übersetzung von Jørgen W. Lang



5. AUFLAGE

---

# HTML5

*kurz & gut*

*Jennifer Niederst Robbins*

*Deutsche Übersetzung von Jørgen W. Lang*

O'REILLY®

Beijing · Cambridge · Farnham · Köln · Sebastopol · Tokyo

Die Informationen in diesem Buch wurden mit größter Sorgfalt erarbeitet. Dennoch können Fehler nicht vollständig ausgeschlossen werden. Verlag, Autoren und Übersetzer übernehmen keine juristische Verantwortung oder irgendeine Haftung für eventuell verbliebene Fehler und deren Folgen.

Alle Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt und sind möglicherweise eingetragene Warenzeichen. Der Verlag richtet sich im Wesentlichen nach den Schreibweisen der Hersteller. Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Alle Rechte vorbehalten einschließlich der Vervielfältigung, Übersetzung, Mikroverfilmung sowie Einspeicherung und Verarbeitung in elektronischen Systemen.

Kommentare und Fragen können Sie gerne an uns richten:

O'Reilly Verlag

Balthasarstr. 81

50670 Köln

E-Mail: [kommentar@oreilly.de](mailto:kommentar@oreilly.de)

Copyright der deutschen Ausgabe:

© 2014 O'Reilly Verlag GmbH & Co. KG

Die Originalausgabe erschien 2013 unter dem Titel *HTML5 Pocket Reference, 5th Edition* bei O'Reilly Media, Inc.

Die Darstellung eines Koalas im Zusammenhang mit HTML ist ein Warenzeichen von O'Reilly Media, Inc.

Bibliografische Information Der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Lektorat: Inken Kiupel, Köln

Korrektorat: Sibylle Feldmann, Düsseldorf

Produktion: Karin Driesen, Köln

Umschlaggestaltung: Karen Montgomery, Boston & Michael Oreal, Köln

Satz: Reemers Publishing Services GmbH, Krefeld, [www.reemers.de](http://www.reemers.de),

Druck: fgb freiburger graphische betriebe; [www.fgb.de](http://www.fgb.de)

ISBN: 978-3-95561-656-4

Dieses Buch ist auf 100% chlorfrei gebleichtem Papier gedruckt.

---

# Inhalt

<b>HTML5 – kurz &amp; gut</b>	<b>1</b>
HTML5-Übersicht	2
Globale HTML5-Attribute	9
Alphabetische Liste der Elemente	14
Nach Funktion geordnete Elemente	156
<b>Anhang A: HTML-Entities</b>	<b>159</b>
<b>Anhang B: Übersicht über die XHTML-Syntax</b>	<b>167</b>
<b>Index</b>	<b>171</b>



---

# HTML5 – kurz & gut

HTML (*HyperText Markup Language*) ist die Markup-Sprache, mit der Textdokumente in Webseiten und -applikationen umgewandelt werden. Die Hauptaufgabe von HTML als Auszeichnungssprache besteht darin, eine *semantische Beschreibung des Inhalts* (die Bedeutung) zu liefern und dem Dokument eine *Struktur* (eine Hierarchie der Elemente) zu geben.

Diese Kurzreferenz enthält eine knappe, aber umfassende Liste der HTML-Elemente und -Attribute, wie sie zum Entstehungszeitpunkt dieses Buchs in den aktuellen Spezifikationen festgelegt wurden: in der HTML5 Candidate Recommendation des World Wide Web Consortiums (W3C, Stand: 17. Dezember 2012), im HTML 5.1 Editor's Draft (Stand: 15. Juni 2013) und in der sogenannten »lebendigen« HTML-Spezifikation, die von der Web Hypertext Application Technology Working Group (WHATWG) gepflegt wird (Stand: 15. Juni 2013).

Elemente und Attribute aus der HTML 4.01-Spezifikation, die in HTML5 verworfen (*deprecated*) wurden, sind aus dieser Auflage entfernt worden. Die in diesem Buch genannten Elemente und Attribute können in HTML 4.01-, XHTML 1.0- und XHTML 1.1-Dokumenten verwendet werden, es sei denn, sie sind ausdrücklich als »Nicht in HTML 4.01« gekennzeichnet. In diesem Fall macht ihre Verwendung das Dokument ungültig.

HTML5-Dokumente können ebenfalls in der XHTML-Syntax (früher auch als »XML-Serialisierung von HTML5« bezeichnet) geschrieben werden. Gibt es für die Verwendung eines Elements mit

XHTML etwas zu beachten, haben wir das entsprechend vermerkt. Die Details zu den Anforderungen der XHTML-Syntax finden Sie in Anhang B am Ende dieser Kurzreferenz.

Dieses Buch besteht aus den folgenden Abschnitten:

- HTML5-Übersicht
- Globale HTML5-Attribute
- Alphabetische Liste der Elemente
- Nach Funktion geordnete Elemente
- Anhang A, *HTML-Entities*
- Anhang B, *Übersicht über die XHTML-Syntax*

## HTML5-Übersicht

HTML5 besitzt eine Reihe neuer Merkmale (Elemente, Attribute, Event-Handler und APIs), mit denen die Programmierung von Web-Applikationen und die Verarbeitung von Formularen erleichtert werden sollen.

Die Spezifikation von HTML5 basiert auf HTML 4.01 Strict. Im Gegensatz zu früheren Versionen wird in HTML5 jedoch keine Dokumenttyp-Definition (DTD) benutzt. Als Basis bestimmter Syntaxregeln dient nun das Document Object Model (DOM, der »Baum«, der die Struktur des Dokuments abbildet). Ein weiterer Unterschied zu früheren Versionen besteht außerdem darin, dass HTML5 eine Reihe von Anweisungen dazu enthält, wie Browser mit falsch formatiertem und nicht mehr unterstütztem Markup-Code umgehen sollen.

## W3C und WHATWG

Beim Schreiben dieses Buchs existierten zwei Organisationen, die zwei leicht voneinander abweichende Versionen der HTML-Spezifikation pflegen.

HTML5 wurde ursprünglich von der Web Hypertext Application Technology Working Group (WHATWG) entwickelt. 2003 gründeten Mitglieder von Apple, Mozilla und Opera die WHATWG, um

die Entwicklung des HTML-Standards auf eine Art voranzutreiben, die sich an den aktuellen Entwicklungspraktiken und dem tatsächlichen Browserverhalten orientierte. Die ursprünglichen Dokumente, Web Applications 1.0 und Web Forms 1.0, wurden im neuen HTML5-Standard zusammengefasst. Dieser wird unter der Leitung von WHATWG-Autor Ian Hickson beständig weiterentwickelt. Nach einiger Zeit wurden die Versionsnummern weggelassen. Stattdessen unterhält die WHATWG unter der Adresse *whatwg.org* jetzt eine »lebendige« (nicht nummerierte) Spezifikation.

Im Jahr 2006 gründete das World Wide Web Consortium (W3C) seine eigene HTML5-Arbeitsgruppe, die auf die Arbeit der WHATWG aufbaute. 2009 wurde dann die Arbeit am XHTML 2.0-Standard eingestellt, um sich auf HTML5 zu konzentrieren. Das W3C unterhält eine (nummerierte) »Snapshot«-Version der HTML5-Spezifikation (*www.w3.org/TR/html5/*). Es wird erwartet, dass diese Variante der Spezifikation noch im Jahr 2014 den Status einer offiziellen W3C-Empfehlung (»Recommendation«) erhält. Gleichzeitig ist bereits HTML5.1 in Arbeit, das 2016 den Empfehlungsstatus erreichen soll. Nightly Builds des Entwurfs (»Editor's Draft«) finden Sie unter *www.w3.org/html/wg/drafts/html/master/*.

Die Unterschiede zwischen der HTML5 Candidate Recommendation der W3C- und der WHATWG-Version sind eher gering. Beide Spezifikationen sind noch im Wandel begriffen. Die folgende Liste zeigt die Unterschiede während des Schreibens dieses Buchs:

## Nur WHATWG

- ping-Attribut für die Elemente a und area
- srcset-Attribut für das img-Element

## Nur WHATWG und W3C HTML5.1

- data (Element)
- menuitem (Element)
- dialog (Element)
- main (Element)
- inert (globales Attribut)
- itemid, itemprop, itemref, itemscope und itemtype (globale Attribute)

onclose und onsort (globale Event-Handler)  
download-Attribut für die Elemente a und area  
sortable-Attribut für das table-Element  
sorted-Attribut für das th-Element

## Nur W3C HTML5

command (Element, ersetzt durch menuitem)  
media-Attribut für das a-Element  
pubdate-Attribut für das time-Element

## Neue semantische Elemente in HTML5

HTML5 enthält eine Reihe neuer semantischer Elemente zum Auszeichnen des Seiteninhalts. Details zu den einzelnen Elementen finden Sie im Abschnitt »Alphabetische Liste der Elemente« auf Seite 14:

article	figcaption	output
aside	figure	progress
audio	footer	rp
bdi	header	rt
canvas	hgroup*	ruby
command*	keygen	section
data**	main**	source
datalist	mark	time
details	menuitem**	track
dialog**	meter	video
embed	nav	wbr

\* Nicht mehr Teil von HTML5.1.

\*\* Nur WHATWG und HTML5.1.

## Neue Eingabetypen

Mit HTML5 wird eine Reihe neuer Typen für Eingabeelemente in Formularen definiert. Diese werden innerhalb des input-Elements als Werte für das type-Attribut angegeben. Folgende neue Werte

sind möglich: color, date, datetime, datetime-local, email, month, number, range, search, tel, time, url und week.

## Obsoleete HTML 4.01-Elemente

Die folgenden HTML 4.01-Elemente wurden nicht in die HTML5-Spezifikation übernommen, weil sie entweder lediglich Darstellungszwecken dienten oder verwirrend bzw. nur schlecht unterstützt waren: acronym, applet, basefont, big, center, dir, frame, frameset, font, isindex, noframes, strike und tt.

## HTML5-APIs

Um dem wachsenden Bedarf nach interaktiven Webinhalten gerecht zu werden, führt HTML5 eine Reihe neuer APIs (Application Programming Interfaces, Schnittstellen) für die Erstellung von Webapplikationen ein. Die APIs standardisieren Aufgaben, für die normalerweise proprietäre Plugins oder eigenhändig programmiertes notwendig waren. Für einige dieser APIs existieren entsprechende Markup-Bausteine wie beispielsweise audio, video und canvas; andere APIs verwenden JavaScript oder serverseitige Komponenten.

Die folgenden APIs sind Teil der HTML5-Spezifikation des W3C:

- **Media API:** Zum Abspielen von Video- und Audiodateien mittels der Elemente video und audio ([bit.ly/17EC8HT](http://bit.ly/17EC8HT) und [bit.ly/13xE0Br](http://bit.ly/13xE0Br)). Die API bietet Multimedia-Synchronisierung und die Möglichkeit, zeitgesteuert Untertitel einzublenden.
- **TextTrack:** Zum Einbinden zeitgesteuerter Untertitel und Beschriftungen für die Elemente video und audio ([bit.ly/16ZQm6S](http://bit.ly/16ZQm6S)).
- **Session History API:** Für die Arbeit mit dem Browserverlauf (»browser history«, [bit.ly/12iDo0Q](http://bit.ly/12iDo0Q)).
- **Offline Web Applications API:** Über die Webressourcen genutzt werden können, auch wenn keine Verbindung zum Internet besteht ([bit.ly/108K1mp](http://bit.ly/108K1mp)).
- **Editing API:** Enthält das neue globale Attribut contenteditable ([bit.ly/11Tw0Tv](http://bit.ly/11Tw0Tv)).

- **Drag and Drop API:** Enthält das neue Attribut `draggable` ([bit.ly/18sdPhs](http://bit.ly/18sdPhs)).

Weitere APIs aus dem HTML5-Umfeld haben eigene Spezifikationen. Hierzu gehören beispielsweise:

- **Canvas API:** Für die zweidimensionale Darstellung mit dem neuen `canvas`-Element ([bit.ly/17ftf5F](http://bit.ly/17ftf5F)).
- **Web Storage API:** Ermöglicht das Speichern von Daten im Browsercache für die spätere Verwendung durch Webapplikationen ([bit.ly/168vZjW](http://bit.ly/168vZjW)).
- **Geolocation API:** Ermöglicht die Weitergabe der geografischen Position des Benutzers (Länge und Breite) und stellt sie für Skripts und Webapplikationen zur Verfügung ([bit.ly/168w5rQ](http://bit.ly/168w5rQ)).
- **Web Workers API:** Ermöglicht das Ausführen von Skripts im Hintergrund, um beispielsweise die Performance von Webapplikationen zu erhöhen ([bit.ly/17ECwWI](http://bit.ly/17ECwWI)).
- **Web Sockets API:** Unterhält eine beständige Verbindung zwischen Client und Server, wodurch Daten in Echtzeit ausgetauscht werden können. Das ist beispielsweise für Multiplayer-Spiele, Chat-Anwendungen oder das Livestreaming nützlich ([bit.ly/13UfWUK](http://bit.ly/13UfWUK)).
- **File API:** Ermöglicht den Zugriff auf per Formular hochgeladene Dateien. Hierdurch wird es möglich, Dateien per Drag-and-Drop hochzuladen und eine Vorschau dieser Dateien anzuzeigen ([bit.ly/11sTkro](http://bit.ly/11sTkro)).

Dies ist nur eine kleine Auswahl der APIs, die sich gegenwärtig in der Entwicklung befinden. Unter den folgenden Quellen finden Sie leicht handhabbare Listen der verfügbaren APIs und anderer Webtechnologien:

- *The Web Platform: Browser Technologies*, unterhalten von Mike Smith, W3C ([platform.html5.org](http://platform.html5.org))
- *HTML5 Landscape Overview*, von Erik Wilde ([dret.typepad.com/dretblog/html5-api-overview.html](http://dret.typepad.com/dretblog/html5-api-overview.html))
- *Web Platform Docs* ([docs.webplatform.org/wiki/apis](http://docs.webplatform.org/wiki/apis))

## HTML5-Dokumentenstruktur

Von HTML5 gibt es nur eine Version. Dabei muss keine Referenz auf eine DTD mehr angegeben werden. Stattdessen benötigen HTML5-Dokumente eine vereinfachte DOCTYPE-Deklaration, damit die Dokumente im Browser standardkonform dargestellt werden. Unten sehen Sie die Empfehlung einer minimalen HTML5-Dokumentenstruktur:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titel des Dokuments</title>
  </head>
  <body>
    Inhalt des Dokuments ...
  </body>
</html>
```

In XML-Syntax geschriebene HTML5-Dokumente benötigen keine DOCTYPE-Angabe. Stattdessen können sie eine XML-Deklaration enthalten. Zudem sollte für Dokumente mit dieser Syntax der MIME-Typ `application/xhtml+xml` oder `application/xml` verwendet werden. Im Folgenden sehen Sie ein einfaches HTML5-Dokument in der HTML5-Schreibweise:

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Titel des Dokuments</title>
  </head>
  <body>
    Inhalt des Dokuments ...
  </body>
</html>
```

## Browserunterstützung für HTML5

Die meisten semantischen Elemente und Attribute von HTML5 werden von allen aktuellen Desktop- und Mobilbrowsern unterstützt (Chrome, Safari, Opera, Firefox und Internet Explorer 9+). Für Browser, die die neuen Elemente und APIs nicht unterstützen, gibt es in der Regel JavaScript-Polyfills (Patches), die die Unterstüt-

zung simulieren. Eine umfassende Liste der existierenden Polyfills finden Sie in der Modernizr-Sammlung unter [www.github.com/Modernizr/Modernizr/wiki/HTML5-Cross-Browser-Polyfills](http://www.github.com/Modernizr/Modernizr/wiki/HTML5-Cross-Browser-Polyfills).

Ein Beispiel für einen einfachen Polyfill ist das »HTML5 Shiv«-Skript von Remy Sharp. Damit auch ältere Browser wie Internet Explorer 8 (und frühere Versionen) die neuen HTML5-Elemente erkennen (und sie so für Stylesheets und Skripts zugänglich zu machen), könnten Sie die Elemente anhand von JavaScript einzeln im DOM erzeugen.

Dieser Befehl erzeugt beispielsweise ein `section`-Element:

```
document.createElement("section");
```

Das HTML5-Shiv-Skript von Remy Sharp erzeugt die neuen Elemente dagegen alle auf einmal für Sie. Es reicht, das bei Google gehostete Skript einfach in Ihr Dokument einzubinden, wie hier gezeigt:

```
<!--[if lt IE 9]>  
<script src="http://html5shiv.googlecode.com/svn/trunk/  
html5-els.js"></script>  
<![endif]-->
```

Die folgenden Quellen können helfen, die Unterstützung und Verwendung von HTML5 in »freier Wildbahn« herauszufinden:

*Can I Use...* (<http://caniuse.com>)

Ein Vergleich der Browserunterstützung für HTML5, CSS3 und andere Webtechnologien von Alexis Deveria.

*HTML5 Please* (<http://www.html5please.com>)

Empfehlungen für die Verwendung von HTML5- und CSS3-Merkmalen, die bereits benutzt werden können, bzw. welche Fallback-Möglichkeiten es gegebenenfalls gibt.

*Wikipedia-Artikel »Comparison of Layout Engines (HTML5)«* ([http://www.en.wikipedia.org/wiki/Comparison\\_of\\_layout\\_engines\\_\(HTML\\_5\)](http://www.en.wikipedia.org/wiki/Comparison_of_layout_engines_(HTML_5)))

Diagramme, die die HTML5-Unterstützung durch die verschiedenen großen Layout-Engines der Browser zeigen.

HTML5 Readiness (<http://www.html5readiness.com/>)

Eine interessante Visualisierung der Unterstützung für HTML5 und CSS3 von 2008 bis heute.

## Die Gültigkeit von HTML5-Dokumenten überprüfen

Mithilfe des Onlinevalidators unter [validator.w3.org](http://validator.w3.org) können Sie überprüfen, ob Ihre HTML5-Dokumente gültig sind (die Unterstützung für HTML5 befindet sich im Moment noch im Betastadium). Einen weiteren Validator finden Sie unter [html5.validator.nu](http://html5.validator.nu).

## Globale HTML5-Attribute

Hierbei handelt es sich um eine Reihe von Attributen, die mit sämtlichen HTML5-Elementen verwendet werden können. Diese werden auch als *globale Attribute* bezeichnet:

`accesskey="Zeichen"`

Definiert ein Tastaturkürzel (*access key*), das ein Element aktiviert oder ihm den Fokus zuteilt. Der Wert ist ein einzelnes Zeichen. Benutzer können mit der Tastenkombination `Alt-<Taste>` (PC) bzw. `Ctrl-<Taste>` (Mac) auf das Element zugreifen.

`class="Textstring"`

Weist dem Element einen oder mehrere Klassennamen zu.

`contenteditable="true|false"`

**Nicht in HTML 4.01.** Besagt, dass der Inhalt des Elements durch den Benutzer verändert werden kann. Dieses Attribut wird von aktuellen Browserversionen bereits gut unterstützt.

`contextmenu="ID des Menüelements"`

**Nicht in HTML 4.01.** Weist dem Element ein Kontextmenü zu. Das Kontextmenü muss vom Benutzer aufgerufen werden, zum Beispiel durch einen Rechtsklick.

`dir="ltr|rtl|auto"`

Gibt die Schreibrichtung eines Elements an. Der Wert `ltr` bedeutet »von links nach rechts«, `rtl` steht für »von rechts

nach links«. auto überlässt die Ermittlung der Schreibrichtung dem Browser.

`draggable="true|false"`

**Nicht in HTML 4.01.** Gibt an, dass ein Element per Drag-and-Drop an eine neue Position im Fenster verschoben werden kann.

`dropzone="copy|move|link|string:Textstring-Typ|file:Dateityp"`

**Nicht in HTML 4.01.** Gibt an, was passiert, wenn Daten auf ein Element gezogen werden, und welche Art von Daten akzeptiert werden. Der Wert `copy` legt eine Kopie der auf das Element gezogenen Daten an, `move` verschiebt die Daten an einen neuen Ort, `link` gibt einen Link auf die Originaldaten zurück. Die Angabe von `string:text/plain` ermöglicht die Verwendung eines beliebigen Textstrings. Der Wert `file:` gibt an, welche Art von Datei akzeptiert wird (z. B. `file:image/png`). Sowohl die auszuführende Aktion als auch die Art der akzeptierten Daten können angegeben werden, zum Beispiel `dropzone="copy string:text/plain"`.

`hidden` (`hidden="hidden" in XHTML`)

**Nicht in HTML 4.01.** Verhindert die Darstellung des Elements und seiner Nachkommen im Browser. Sämtliche Skripts oder Formularelemente werden weiterhin ausgeführt, aber nicht angezeigt.

`id="Textstring"`

Weist dem Element eine einmalige ID zu.

`inert` (`inert="inert" in XHTML`)

**WHATWG & Nur HTML5.1.** Markiert ein Element als inaktiv (*inert*). Ein so gekennzeichnetes Element kann weder ausgewählt noch gesucht oder von Benutzerinteraktionen angesprochen werden.

`itemid="Text"`

**WHATWG & Nur HTML5.1.** Teil des Mikrodatensystems zum Einbetten maschinenlesbarer Daten. Das Attribut `itemid` enthält einen global anerkannten Identifier (wie beispielsweise den ISBN-Code für ein Buch). Es wird zusammen mit dem

Attribut `itemtype` im selben Element verwendet, das das dazugehörige `itemscope`-Attribut enthält.

`itemprop="Text"`

**WHATWG & Nur HTML5.1.** Teil des Mikrodatensystems zum Einbetten maschinenlesbarer Daten. Das Attribut `itemprop` definiert den Namen der Eigenschaft. Der Inhalt des Elements definiert seinen Wert. Der Wert kann auch eine vom `href`-Attribut bereitgestellte URL eines `a`-Elements oder das `src`-Attribut eines `img`-Elements sein.

`itemref="Durch Leerzeichen getrennte Liste mit IDs"`

**WHATWG & Nur HTML5.1.** Teil des Mikrodatensystems zum Einbetten maschinenlesbarer Daten. Hierbei definiert das `itemref`-Attribut eine Liste der Elemente aus dem aktuellen Dokument (durch Angabe der Werte ihrer `id`-Attribute), die in einem Item enthalten sein sollen. Hierbei muss das `itemref`-Attribut im gleichen Element verwendet werden wie das `itemscope`-Attribut, mit dem das Item angelegt wurde.

`itemscope`

**WHATWG & Nur HTML5.1.** Teil des Mikrodatensystems zum Einbetten maschinenlesbarer Daten. `itemscope` erzeugt ein neues Item, sprich, eine Gruppe von Eigenschaften (Name/Wert-Paare).

`itemtype="URL oder umgekehrtes DNS-Label"`

**WHATWG & Nur HTML5.1.** Teil des Mikrodatensystems zum Einbetten maschinenlesbarer Daten. Das Attribut `itemtype` kennzeichnet einen standardisierten Item-Typ anhand einer URL (z. B. `http://vocab.example.net/book`) oder ein umgekehrtes DNS-Label (z. B. `com.example.person`). Das `itemtype`-Attribut wird im selben Element verwendet wie das, das auch das `itemscope`-Attribut enthält.

`lang`

Gibt anhand eines Sprachcodes an, welche (natürliche) Sprache für das Element verwendet wurde.

`xml:lang`

**Nur XHTML.** Gibt in XHTML-Dokumenten an, welche (natürliche) Sprache für das Element verwendet werden soll.

`spellcheck="true/false"`

**Nicht in HTML 4.01.** Gibt an, ob für das Element Rechtschreibung und Grammatik überprüft werden sollen.

`style="CSS-Stile"`

Weist einem Element eine oder mehrere Inline-Stildefinitionen zu.

`tabindex="Zahl"`

Gibt an, welche Position das gegenwärtige Element in der Tabulatorreihenfolge des Dokuments hat. Der Wert muss zwischen 0 und 32.767 liegen. Durch Verwendung dieses Attributs kann sich der Benutzer durch mehrmaliges Drücken der Tabulatortaste z. B. durch die Links (oder Formularfelder) einer Seite bewegen.

`title="Textstring"`

Mit diesem Attribut kann ein Element mit einem Titel oder kurzen zusätzlichen Informationen versehen werden.

`translate="yes/no"`

**Nicht in HTML 4.01.** Gibt an, ob der Textinhalt bei einer Lokalisierung des Dokuments übersetzt werden soll. `yes` ist der Standardwert; der Wert `no` lässt den Inhalt unverändert.

Die folgenden Attribute sind nicht offiziell Teil der globalen Attribute, sind aber in HTML5-Elementen erlaubt:

`aria-*="Textstring oder Zahl"`

**Nicht in HTML 4.01.** Hiermit kann einem Element eine beliebige Zustands- oder Eigenschaftsdefinition nach dem WAI-ARIA-Standard zugewiesen werden, um so die Zugänglichkeit des Elements zu erhöhen, zum Beispiel `aria-hidden="true"`. Der Wert des Attributs hängt von der jeweiligen Eigenschaft ab. Weitere Informationen zu den unterstützten Zustands- und Eigenschaftsdefinitionen finden Sie unter [www.w3.org/TR/wai-aria/states\\_and\\_properties](http://www.w3.org/TR/wai-aria/states_and_properties).

`data-*`="Textstring oder numerische Daten"

**Nicht in HTML 4.01.** Hiermit können Webautoren eigene datenbezogene Attribute erzeugen, zum Beispiel `data-length`, `data-duration`, `data-speed` usw. Auf diese Weise können für den Benutzer unsichtbare Daten direkt in das Dokument eingebettet und von eigenen Applikationen oder Skripts weiterverwendet werden.

`role`="standardisierte WAI-ARIA-Rolle"

**Nicht in HTML 4.01.** Weist dem Element eine der standardisierten WAI-ARIA-Rollen zu, um seinen Zweck für Benutzer mit Einschränkungen leichter erkennbar zu machen. Die folgenden Rollen (`application`, `banner`, `complementary`, `contentinfo`, `form`, `main`, `navigation` und `search`) haben eine wichtige Bedeutung für die Navigation in Hilfstechnologien. Eine vollständige Liste der erlaubten Werte für das `role`-Attribut finden Sie unter [www.w3.org/TR/wai-aria/roles#role\\_definitions](http://www.w3.org/TR/wai-aria/roles#role_definitions).

## HTML5-Event-Handler

Falls nicht anders angegeben, können die folgenden Event-Handler in Form von Attributen für alle HTML-Elemente verwendet werden:

<code>onabort</code>	<code>ondragleave</code>	<code>onload</code> *	<code>onratechange</code>
<code>onblur</code> *	<code>ondragover</code>	<code>onloadeddata</code>	<code>onreset</code>
<code>oncancel</code>	<code>ondragstart</code>	<code>onloadedmetadata</code>	<code>onscroll</code>
<code>oncanplay</code>	<code>ondrop</code>	<code>onloadstart</code>	<code>onseeked</code>
<code>oncanplaythrough</code>	<code>ondurationchange</code>	<code>onmousedown</code>	<code>onseeking</code>
<code>onchange</code>	<code>onemptied</code>	<code>onmousemove</code>	<code>onselect</code>
<code>onclick</code>	<code>onended</code>	<code>onmouseout</code>	<code>onshow</code>
<code>onclose</code> **	<code>onerror</code> *	<code>onmouseover</code>	<code>onsort</code> **
<code>oncontextmenu</code>	<code>onfocus</code> *	<code>onmouseup</code>	<code>onstalled</code>

---

\* Event-Handler für das `Window`-Objekt bei Verwendung am `body`-Element.

\*\* WHATWG, Nur HTML5.1. Nicht in der HTML5 Candidate Recommendation.