Kirsten Schelper

ebooks für Webworker

Schnelleinstieg HTML5 & Wordpress

Juli 2011

Kirsten Schelper
Dipl. Kommunikationsdesignerin (FH)

kontakt@die-netzialisten.de
www.die-netzialisten.de

Sie dürfen gern aus meinen Texten zitieren.
Bitte machen Sie die Zitate aber als solche kenntlich und geben Sie dazu meinen Namen an.

1. Auflage Juli 2011

Dieses e-book ist die überarbeitete und erweiterte Fassung einer Artikelserie aus meinem Blog www.die-netzialisten.de

Das Buch wendet sich an Webdesigner, Blogger und Wordpress-Nutzer, die sich schnell einen Überblick über die Funktionsweise von HTML5 verschaffen möchten.

Inhalt

1. Wozu HTML5?

Eigentlich kommen wir mit HTML4 ja ganz gut zurecht, wieso also schon wieder etwas Neues? Was ist der Sinn hinter HTML5?

HTML5 wurde aus der Praxis heraus entwickelt und orientiert sich daran, wie Webseiten tatsächlich funktionieren bzw. wie sie genutzt werden. Die neuen Funktionen sollen »echte« Probleme lösen, und alle, die mit dem Web als Werkzeug arbeiten, in ihrer Arbeit unterstützen. Die »reine Lehre« steht ganz bewusst nicht im Mittelpunkt. Auch wollen die Entwickler von HTML5 das Rad nicht neu erfinden. Was gut funktioniert, bleibt erhalten. HTML5 ist deshalb auch kompatibel mit älteren HTML-Versionen.

Am bekanntesten sind wahrscheinlich die Tags, mit denen man Audio– und Video-Dateien einbetten kann. Mit dem HTML5-Element **<video>**kann man einen Film so in die Website einbauen, dass die Besucher kein PlugIn mehr brauchen, um das Video anzuschauen. Ich freue mich jedenfalls über jedes in HTML5 eingebettete Video, das ich mit meinem iPad anschauen kann.

Aber HTML5 kann natürlich noch viel mehr als Videos abspielen.

2. Das Seitenlayout

Der Aufbau einer Webseite

Werfen wir mal einen Blick in die Praxis.
Das übliche HTML-Template sieht bisher ungefähr so aus:

„Klassischer"Aufbau eines HTML-Templates:

Die einzelnen Bereiche des Layouts werden von -Containern umschlossen, die mit einer id versehen sind

Mit HTML5-Elementen wird daraus das:

Aufbau eines Templates in HTML5:

Neue Elemente mit eindeutigen Bezeichnungen umschließen die Bereiche

» Das Problem

Über ein ** **-Element kann man einen Bereich im Layout nicht eindeutig bezeichnen. Die Container haben zwar in der Regel eine ID, den Namen dafür kann sich aber jeder Entwickler je nach Gusto ausdenken. Das ** ** für die Navigation hat in unserem Beispiel oben die ID »navi«.
Es könnte aber auch »main« oder »menu« heissen.

» Die Folge

Weder Suchmaschinen noch Screen-Reader können einschätzen, welche Art von Inhalt in einem ** ** zu finden ist.

Bessere Orientierung über eindeutige Elemente

Mit HTML5 wird das anders. Hier gibt es das semantische Element **<nav>** und dorthinein packt man die Navigation der Seite. Suchrobots und Screen-Reader, die auf der Suche nach Inhalten sind, können diesen Bereich einfach überspringen. Denn die Inhalte liegen woanders, nämlich in den Bereichen **<section>** und/oder **<article>**.

Es geht also um Einfachheit und Funktionalität. HTML5 führt neue strukturelle Elemente ein, die man sich in HTML4 mehr oder weniger umständlich zusammenbauen musste.

Im Grunde hat HTML5 das Ziel, die Struktur des Codes schlanker und besser lesbar zu machen. Adressaten sind dabei nicht nur die Entwickler, die sich Arbeit und Mühe sparen können, sondern auch lesende Maschinen.

Smarte Formulare

Aber nicht nur der Seitenaufbau wird übersichtlicher, HTML5 macht auch an anderen Stellen das Leben des Webworkers leichter. Ein einfaches Formular ist zwar schnell zusammengetippt, aber wenn es an die Überprüfung der Eingaben geht, läuft nichts ohne Javascript.

Bei HTML5 übernimmt in Zukunft der Browser die Überprüfung. Für Formularfelder gibt es neue Zusätze für input-Felder. Will man zum Beispiel ein Formularfeld anlegen, in das eine E-Mail-Adresse eingegeben werden soll, so schreibt man es einfach so: **input type='email'**.

Der Browser weiß nun, dass in diesem Feld eine Mailadresse stehen soll und überprüft, ob eine gültige Adresse eingegeben wurde. Wenn nicht, gibt er eine Warnmeldung aus. Das alles passiert ganz ohne Javascript.

3. Aufbau eines Artikels in Wordpress

HTML5 bringt eine Reihe von neuen Elementen mit, die anzeigen, welche Art von Inhalt sie enthalten. Eine Navigations-Leiste wird beispielsweise nicht mehr von einem ** ** umschlossen, sondern sie bekommt ihr eigenes Tag: **<nav>**. Dasselbe gilt für den Header einer Seite und für eine Reihe von anderen Bereichen.

Struktur eines Blog-Artikels

Um einen Blog-Artikel zu strukturieren, müssen wir passende Tags für folgende Inhalte finden:

- Die Überschrift
- Das Erscheinungs-Datum
- Den eigentlichen Artikeltext
- Anmerkungen, Verweise, Metaangaben

In HTML5 sieht ein Blog-Artikel dann so aus:

```
<article>
    <header>
        <h2>Überschrift</h2>
    <time>
        Erscheinungsdatum
    </time>
    </header>
        Inhalt des Artikels
    <footer>
        Links zu Kommentaren etc.
    </footer>
</article>
```

Auch innerhalb eines Blog-Artikels kommen also die neuen Strukturelemente

zum Einsatz. Die Überschrift und das Erscheinungsdatum werden von einem header-Tag umschlossen. Danach kommt der Artikel-Text, der eigentliche Inhalt. Folgen darunter noch weitere Informationen wie zum Beispiel Hinweise auf Kommentare, eine Kurzbiografie des Autors oder Verweise auf thematisch verwandte Themen, dann stehen diese Inhalte in einem footer-Tag.

Der ganze Artikel wird also nicht mehr von einem ** ** umschlossen, sondern ist als **<article>** ausgewiesen.

Die Startseite im Blog

Was passiert nun, wenn mehrere Artikel aufeinander folgen?
In einem Blog ist das die übliche Darstellungsweise, der neueste Artikel steht oben und darunter kommen weitere Artikel in zeitlicher Reihenfolge.

Die Struktur sieht in diesem Fall so aus:

```
<div>

      <article>
      </article >

      <article>
      </article >

      <article>
      </article >

</div>
```

Der Bereich, in dem die einzelnen Artikel untereinander angezeigt werden, wird in ein ** ** gepackt. Das gute alte ** ** also.

In HTML5 gibt es noch ein Tag, das dem **<article>**-Element ähnelt. Dieses Tag heißt <section>. Einige Autoren verwenden nun **<section>** anstatt eines ** **, um mehrere **<article>**-Elemente auf einer Blog-Homepage zusammenzufassen.

Article versus Section

Das ist nicht ganz korrekt. Der Grund: Das **<section>**-Element ist nicht als Container gedacht. Diese Rolle behält nach wie vor das ** **.

Eine **<section>** ist ein in sich geschlossenes Element, das z.B. eine Headline und einen Inhalt enthält. Als Faustregel kann man sich folgendes merken:

Ein ** ** ist immer dann die richtige Wahl, wenn man etwas zusammenfassen möchte. Ein **<section>**-Element steht für sich und hat immer eine eigene Headline bzw. könnte sinngemäß eine haben.

Das <section>-Element steht damit quasi eine Stufe tiefer als das **<article>**-Element, das relativ komplex strukturierte Inhalte aufnehmen kann und ausdrücklich für Blog-Posts konzipiert wurde.

Auch wenn es darum geht, einen Text »verlinkbar« (z.B. per Twitter) zu machen, ist das **<article>**-Tag die richtige Wahl. Die WHATWG-Community weist ausdrücklich auf diesen Punkt hin:
„Authors are encouraged to use the article element instead of the section element when it would make sense to syndicate the contents of the element."

ZUM WEITERLESEN

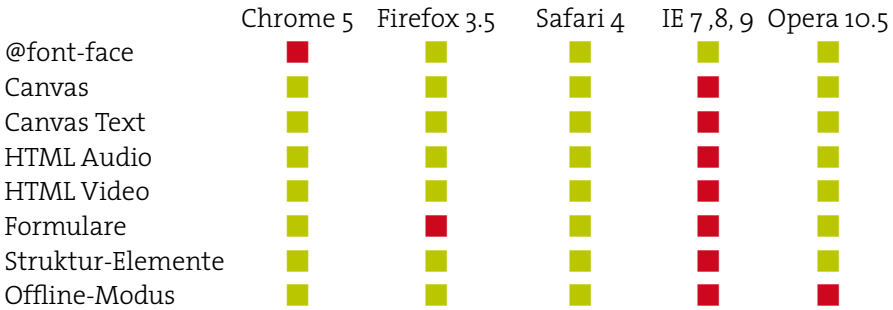
- Smashingmagazine: Learning to Love HTML5
- diveintohtml5.org: Dive Into HTML5
- threestyles.com: HTML5 Rocks My Socks Off

3. Die Browser

Die gute Nachricht: Alle modernen Browser unterstützen das Format. Firefox, Opera, Safari und Chrome sind mit im Boot. Eine Ausnahme macht wie üblich der Internet-Explorer. Bis inklusive der Version 8 ist der Internet Explorer aussen vor, mit der Version 9 wird es ein bisschen besser, aber immer noch nicht überzeugend.

Aber es gibt Workarounds über Javascript, mit denen man den Internet Explorer dazu bringen kann, HTML5 zu verstehen.

Überblick Browser-Kompatibiltät



focus.com | Stand Mai 2011

Nachhilfe für den IE

Wie man sieht, hat der IE noch Nachholbedarf. In der Zwischenzeit kann man auf Javascript-Hacks zurückgreifen. Das Prinzip: Über Javascript werden die neuen Elemente einfach »nachgebaut«:

```
document.createElement(„article“);

document.createElement(„aside“);

document.createElement(„audio“);

(...)
```

Eine sehr umfassende und mächtige Lösung dafür bietet **Modernizr**. Modernizr ist eine Open-Source Javascript-Library, die ältere Browser für HTML5 und CSS3 fit macht.

Stylesheet anpassen

Auch im CSS-File müssen die neuen Elemente natürlich bedacht werden. Eine gute Basis dafür ist das Projekt normalize.css. In diesem Stylesheet werden alle HTML5-Elemente »auf Linie« gebracht, so dass alle Browser die Elemente gleich und konsistent darstellen. Diese CSS-Datei ist sozusagen die Weiterführung des berühmten reset.css von Eric Meyer.

Ein sehr aufgeräumter Doctype

Eine sehr nette Neuerung bei HTML5 ist die überarbeitete Doctype-Angabe. Bisher sah der Doctype etwa so aus:

```
<!DOCTYPE HTML PUBLIC „-//W3C//DTD HTML 4.01 Transitional//EN“
„http://www.w3.org/TR/html4/loose.dtd“>
```

Die korrekte Syntax lautet ab sofort:

```
<!DOCTYPE HTML>
```

Das war’s schon.

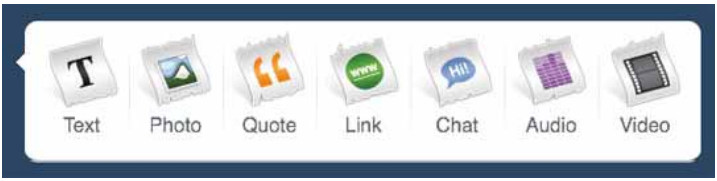
Gerüchte, diese Angabe würde im Explorer 8 für Verwirrung sorgen, haben sich nicht bestätigt.

5. WordPress Post Formats

Seit der Version 3.1 unterstützt WordPress eine Reihe von HTML5-Eigenschaften über die so genannten **Post Formats**.

Post Formats sind spezielle Formatierungen, sehr ähnlich den Custom Post Types. Post Formats unterscheiden sich von Custom Post Types dadurch, dass sie eng an den Artikel gebunden sind. Damit sind sie sind eher Formatierungen *innerhalb* eines Artikels und keine eigenständige Artikel-Form.

Wer **Tumblr** oder **Posterous** kennt, weiß wie das aussieht:



Diese Formate werden von WordPress unterstützt

- **aside** Ein kleiner Content-Schnipsel, üblicherweise ohne Überschrift
- **gallery** Eine Bildergalerie
- **link** Ein Link zu einer anderen Seite
- **image** Ein einzelnes Bild
- **quote** Ein Zitat
- **status** Eine kurze Statusmeldung, ähnlich wie ein Tweet bei Twitter
- **video** Eine Video-Datei
- **audio** Eine Audio-Datei
- **chat** Ein kurzer Chat-Dialog

Ist das WordPress-Theme entsprechend vorbereitet, erscheint eine Auswahl-Liste mit den Formaten in der Bearbeitungsansicht des Artikels im Backend.

Post Formats ins Theme einbinden

Ist das Theme noch nicht vorbereitet, kann man die Funktion aber auch manuell in die **functions.php** eintragen:

```
add_theme_support('post-formats', array( 'aside', 'chat',
'gallery', 'image', 'link', 'quote', 'status', 'video',
'audio'));
```

In das array kann man alle Formate eintragen oder nur eine Auswahl. Die Elemente, die man nicht braucht, lässt man einfach weg.

Damit ist die Funktion freigeschaltet.
In der Bearbeitungsansicht des Artikels erscheint jetzt das oben gezeigte Menü und man kann seine Inhalte entsprechend formatieren.

Damit man davon aber auch etwas sieht und das Element auch auf der Website angezeigt wird, muss man das Element noch **dort aufrufen**, wo es erscheinen soll. Man fügt an der entsprechenden Stelle – z.B. in index.php, single.php oder archive.php – noch eine Code-Zeile ein.
Soll beispielsweise ein Bild-Element angezeigt werden, sieht das so aus:

```
if ( has_post_format( 'image' )) {
 the_post_thumbnail('large');
 echo the_title();
}
```

Ein anderer, etwas eleganterer Weg ist der Einsatz von **template_parts.**

Dieses Tag wurde mit der WordPress-Version 3.0 eingeführt und man kann damit eine Datei (Template) aufrufen und in den Code einbauen.

Ein Beispiel aus dem Codex:

```
get_template_part('nav'); // Navigation bar (nav.php)

get_template_part('nav', '2'); // Navigation bar #2 (nav-2.php)
```

Die erste Zeile ruft ein Template auf, das **nav.php** heißt.
Die zweite Zeile ruft ein weiteres Template namens **nav-2.php** auf.

In der Klammer nach dem Tag **get_template_part** steht im Prinzip der Name des Templates, das aufgerufen werden soll. Besteht dieser Name aus zwei Teilen, die mit einem Bindestrich verbunden sind, werden die beiden Namensteile hintereinander notiert, dazwischen steht ein Komma.

Im Zusammenhang mit den Post Formats setzt man die Funktion get_template_part folgendermaßen ein:

1. Template anlegen

Für jedes Post Format-Element legt man im Theme-Ordner eine eigene Datei an. Im Ordner sieht das dann so aus:

2. Template aufrufen

Dieser Code sorgt dafür, dass z.B. in der index.php die richtigen Inhalte angezeigt werden:

```
if (have_posts()) :
     while (have_posts()) : the_post();
          if(!get_post_format()) {
               get_template_part('content', 'standard');
          } else {
               get_template_part('content', get_post_format());
          }
        endwhile;
endif;
```

WordPress schaut zuerst nach, ob überhaupt ein **Post Format** ausgewählt wurde. Wenn ja, wird die Datei mit dem entsprechenden Namen geladen (z.B. content-aside.php). Wenn nicht, wird das Standard-Format (content.php) geladen.

Das Theme **„Toolbox“** arbeitet mit dieser Methode (s. nächste Seite).

6. WordPress-Themes für den Einsatz mit HTML5

A. HTML5 Blank-Themes für Entwickler

Toolbox
http://themeshaper.com/2010/07/02/toolbox-html5-starter-theme/

Starkers
http://nathanstaines.com/archive/starkers-html5

H5
http://digwp.com/2009/07/free-html-5-wordpress-theme/

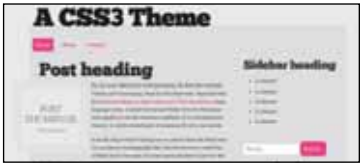
constellationtheme
http://constellationtheme.com/

html5press
http://www.longren.org/wordpress/html5press/

B. Fertige HTML5 Themes

Graphite
Link zum Theme

Ari
Link zum Theme



A CSS3 Theme
Link zum Theme

Spectacular
Link zum Theme

Yoko
Link zum Theme

Grey
Link zum Theme

7. Was HTML5 sonst noch so kann

HTML5 bringt ein ganzes Bündel an neuen Eigenschaften und Möglichkeiten mit. Dabei sind nicht nur die neuen strukturierenden Elemente interessant, sondern auch die Schnittstellen (APIs) zu anderen Techniken und Anwendungen.

A. Elemente und Tags

Struktur Elemente section, nav, article, aside & Co.

Nicht nur die neuen Tags machen die Struktur eines Templates übersichtlicher, auch die Hierarchie der Überschriften **<h1>** bis **<h6>** wird dadurch erweitert. Bisher wurde einfach nur „durchgezählt" (**<h1>** ist wichtig, **<h6>** ist unwichtig). Jetzt kann man eindeutig bestimmen, welche Bedeutung eine Überschrift hat. Die Position der Überschrift innerhalb des Elements gibt dazu den Hinweis: Eine Überschrift in einem **<aside>** Element hat eine geringere Bedeutung als eine Überschrift ein einem header-Element eines **<article>**-Tags.

Multimedia-Elemente

Audio- und Videodateien konnte man zuvor nur über ein Media-Player-PlugIn einbauen (am bekanntesten ist Adobe® Flash®).
Jetzt reicht ein simples **<video>** oder **<audio>**-Tag. Für die Dateien können mehrere Quellen bzw. verschiedene Formate hinterlegt werden, aus denen der Browser dann das Format auswählt, das er darstellen kann.

Canvas

Das Canvas-Element ist eine Zeichenoberfläche, auf der man mit Maus oder Stift zeichnen kann. Canvas ist eine sehr mächtige Eigenschaft, hierüber lassen sich Animationen und Spiele realisieren.

Formulare mit intelligenten Feldern

Musste man bisher die Eingaben in Formularen über Javascripts überprüfen, ist diese Überprüfung jetzt „eingebaut" und wird vom Browser übernommen. Das input-Element wurde um verschiedene Typen erweitert. Man kann jetzt vorgeben, ob ein Feld einen Suchbegriff, eine Telefonnummer, eine E-Mail-Adresse oder ein Datum enthalten soll.

Interaktive Elemente

Der Inhalt des **<summary>**-Elements wird permanent angezeigt, der Inhalt des **<details>**-Elements kann ein- und ausgeblendet werden.

Für das Erstellen von Werkzeugleisten und (Kontext-)Menüs ist das **<menu>**-Element die strukturierende Basis. **<command>** definiert einen Button (z.B. einen Radiobutton oder eine Checkbox) innerhalb des <menu>-Bereichs.

B. Schnittstellen (API)

HTML5 hält Schnittstellen für anderen Anwendungen und Techniken bereit. Die Grenze zwischen der klassischen Website, die im Browser angezeigt wird und der mobilen App verschwimmt mit HTML5 immer mehr.

WebApps für Smartphones können in Zukunft in HTML5 entwickelt werden. Sie werden damit zur vollwertige Alternative zur klassisch codierten App.

Daten lokal speichern

Es ist zum Beispiel möglich, Daten lokal auf dem jeweiligen Gerät zu speichern. Eine solche Website funktioniert auch dann noch, wenn gerade keine Verbindung zum Internet besteht.
Für die Entwicklung von mobilen Apps ist das ein zentraler Punkt.

Drag & Drop

Über Javascript kann man Drag & Drop-Funktionalitäten realisieren.

Video & Audio

Video- und Audio-Dateien können ohne externes PlugIn direkt über ein Tag in den Code eingebettet werden. Alles andere übernimmt der Browser.

Geolocation

Über diese Schnittstelle kann man – so der Nutzer es erlaubt – auf die GPS-Daten eines Geräts zugreifen. Ideal für die Entwicklung von Smartphone-Applikationen.

Glossar Die neuen HTML5-Tags

<article> In sich geschlossenes Content-Element

<aside> Content-Element, das Teil eines Artikels ist

<audio> Musikdatei oder Podcast

<canvas> Definiert eine Grafik

<command> Definiert einen Button (nur sichtbar, innerhalb eines Menu-Elements)

<datalist> Definiert eine Dropdown-Liste

<details> Details eines Elements, die auf Klick sichtbar werden

<dialog> Zeichnet eine Konversation aus, z.B. in einem Chat oder Forum

<embed> Einbinden von Plug-Ins wie z.B. Java oder Flash

<figure> Gruppiert mehrere Elemente, die den Content ergänzen, z.B. eine Bildbeschreibung, und ein Foto

<footer> Der footer-Bereich eines HTML-Templates

<header> Der header-Bereich eines HTML-Templates

<mark> Zeichnet Teile eines Textes aus (Hervorhebung)

<meter> Gibt ein relatives Maß an (der minimale und maximale Wert müssen bekannt sein)

<nav> Der Navigations-Bereich eines HTML-Templates

<output> Ergebnis einer Auswertung oder Berechnung (z.B. die Summe aller Artikel in einem Warenkorb)

<section> Eine Zusammenfassung verschiedener Elemente, z.B. ein Kapitel
<source> Beschreibt eine Media-Datei (z.B. Video oder Audio)

<time> Zeit und Datum

<video> Video-Datei

www.die-netzialisten.de
Blog über Webdesign, Webentwicklung und WordPress

www.schelperdesign.net
Ideen sichtbar machen –
Wir gestalten und programmieren klare, elegante Webseiten