

## 1.5 XHTML und CSS aus Sicht eines (In)Designers

Wenn Sie als (In)Designer bisher noch nicht mit XHTML und CSS in Berührung gekommen sind, sollten Sie dieses Kapitel durchlesen. Falls Sie schon Basiswissen darüber besitzen, was HTML und CSS ist, können Sie dieses Kapitel überspringen. Webentwicklern oder fortgeschrittenen Benutzern rate ich sogar dazu, denn sie wird nichts Neues erwarten!

Sollten Sie also hier nun weiterlesen, haben Sie noch keine Kenntnis von den Webtechnologien oder Sie sind einfach daran interessiert, was ich dazu zu sagen habe.

Unter Webtechnologien versteht man die Auszeichnungssprachen und Standards, die benötigt werden, um Webseiten oder Inhalte in Browsern darzustellen. Dazu gehören u.a. HTML, XHTML, HTML5, CSS, JavaScript etc. Für ePub interessant sind dabei XHTML, HTML5, CSS und JavaScript.

### Einleitung zum besseren Verständnis

Angefangen hat es mit **HTML** (**H**yper**T**ext **M**arkup **L**anguage). HTML ist die Grundlage für das World Wide Web, um Inhalte wie Texte, Bilder und Hyperlinks in einer genormten Struktur darzustellen (Version 4.01). HTML ist allerdings als Austauschformat zwischen Programmen und Geräten eher ungeeignet, da in der Praxis viele Ausnahmen und Unklarheiten in Daten toleriert werden. HTML ist zudem beschränkt in der Strukturierung.

Dann gibt es **XML** (**E**Xtensible **M**arkup **L**anguage). XML sieht auf den ersten Blick ähnlich aus, ist aber strenger und verzeiht keine Fehler im Code. XML bietet die Möglichkeit, verschiedene Sprachen für ganz unterschiedliche Anwendungen mit einer gemeinsamen Syntax zu definieren.

Um das Beste aus beiden Welten zu vereinen, wurden HTML und XML zu **XHTML** (**E**Xtensible **H**yper**T**ext **M**arkup **L**anguage) zusammengesetzt. Die Auszeichnungsmöglichkeiten stammen in dem Fall von HTML, die strenge Struktur kommt von XML.

HTML5 ist die Fortführung von HTML und XHTML und wird ergänzt durch die Möglichkeit, nun auch Video, Audio, Animationen etc. im Browser ohne Zusatzprogramme (wie z.B. Flash) darzustellen. Übrigens können HTML5-Dateien die Dateierweiterung .html sowie auch .xhtml besitzen. Lassen Sie sich davon nicht täuschen.

Sie haben also ein XHTML-Dokument, womit Ihre Daten schön strukturiert auf möglichst vielen Geräten dargestellt werden können. Doch weiß der Browser noch nicht, wie er die Struktur darstellen soll. Hier kommt **CSS** (**C**ascading **S**tyle **S**heets) ins Spiel. Im CSS werden die Formatierungen definiert, die im weitesten Sinne vergleichbar mit Absatz- und Zeichenformatvorlagen in InDesign sind. CSS sagt dem Browser also, welche Farben dargestellt werden sollen, um nur ein Beispiel zu nennen. CSS kann direkt in die

XHTML-Datei eingefügt werden oder verlinkt zu einer separaten CSS-Datei. (Es wird ein Link im XHTML-Dokument hinterlegt, wo die Formatierungsangaben zu finden sind.) Ähnlich wie HTML5 wird auch CSS immer weiterentwickelt und bietet in der aktuellen CSS3-Version viel mehr Darstellungsmöglichkeiten. Neben der Gestaltung hat CSS aber auch Einfluss auf die Darstellung der Struktur.

JavaScript ist ein optionaler Zusatz zu XHTML und CSS. Das heißt, um unseren Inhalt darzustellen, werden nur XHTML und CSS benötigt. Allerdings leben wir Menschen ja schon größtenteils in einer digitalen Welt und ein einfacher Inhalt ist uns manchmal zu langweilig. Ein Beispiel wäre eine animierte Slideshow verschiedener Bilder. Dabei ist uns JavaScript behilflich, da es erlaubt, gewisse Aktionen in ein XHTML-Dokument einzubauen. JavaScript ist allerdings keine leicht zu verstehende Programmiersprache, wird noch sehr wenig in E-Books verwendet und auch nicht von allen Lesegeräten unterstützt. Daher wird auch nicht weiter in diesem Buch auf JavaScript eingegangen.

## Ein Blick in eine (X)HTML-Datei

XHTML-Dateien bestehen aus reinem Text und können mit einem normalen Texteditor betrachtet werden. Einige Texteditoren beherrschen das sogenannte Syntax-Highlighting, was den Code in verschiedenen Farben darstellt und uns somit visuell die Unterschiede zwischen Inhalt und den Struktur-Tags anzeigt.

### Elemente

Durch Zuweisen von Absatz-, Zeichen- und Objektformaten definieren Sie in InDesign die Information, was ein Titel ist, was der Grundtext, was eine Bildunterschrift bzw. um welche Art von Rahmen es sich handelt. Diese Information wandelt InDesign beim ePub-Export in eine XHTML-Struktur um. Das geschieht mit Hilfe von Tags, welche Sie durch spitze Klammern `<>` erkennen können. Es gibt meist ein Start-Tag `<>` und ein End-Tag mit einem Schrägstrich nach der ersten Klammer `</>`. Die Inhalte werden zwischen beide Tags gesetzt. Das heißt, ein Element besteht im Normalfall aus Start- und End-Tag sowie dem Inhalt. Eine XHTML-Datei mit einem Absatz sieht so aus:

```
<html>
<head>
  <title>Ich bin ein HTML Dokument</title>
</head>
<body>
  <p>Hier steht mein Inhalt</p>
</body>
</html>
```

Die Basisstruktur einer HTML-Datei fängt mit `<html>` an. Gefolgt wird dies von `<head>`. Was innerhalb des `<head>`-Elements steht, sind Informationen zum HTML-Dokument. `<title>` ist dabei die Mindestangabe und `<head>` kann eine ganze Reihe an weiteren Angaben enthalten. Hier wird auch später die CSS-Datei angegeben, in der die Formatierungseigenschaften definiert sind. Die Angaben im `<head>`-Element werden nicht dargestellt.

Weiter geht es mit `<body>`. Dieses Element ist unsere Seite und enthält den darzustellenden Inhalt.

Unser Satz »Hier steht mein Inhalt« ist mit dem `<p>`-Element (`p` = Paragraph) ausgezeichnet, was die Mindestangabe zum Darstellen eines Absatzes ist. Wie Sie feststellen, wird ein angefangenes Element auch wieder geschlossen. Es gibt allerdings Elemente, die kein End-Tag besitzen und keinen Text darstellen bzw. eine andere Funktion haben. Die bekanntesten (leeren) Elemente, die Ihnen auch im ePub begegnen, sind `<br />` (break, harter Zeilenumbruch), `<hr />` (horizontal rule, Linie) oder `<img />` (image, Bild). Diese Elemente zeichnen sich aus, indem sie ein Leerzeichen zzgl. eines Schrägstrichs vor der abschließenden spitzen Klammer besitzen.

## Die Elemente aus dem InDesign-Export

- Ein Rahmen wird traditionellerweise mit einem `<div>` exportiert. HTML-Rahmen können Texte und Bilder enthalten, um Gruppierungen und Strukturierungen mit speziellen Auszeichnungen zu bilden. Anders als bei InDesign müssen Texte nicht in einem Rahmen in HTML definiert werden. In InDesign werden Rahmen und Objektformate als `<div>` ausgegeben.
- Titelzeilen werden mit `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` und `<h6>` ausgezeichnet, wobei `<h1>` die höchste Ebene eines Titels angibt und `<h6>` die niedrigste. In InDesign können Absätze mit Hilfe von Absatzformaten als `<h1>`–`<h6>` ausgezeichnet werden.
- Normale Texte werden mit dem `<p>`-Element angegeben.
- Bei Listen gibt es zwei Arten: Nummerierte Listen werden mit `<ol>` ausgezeichnet. Listen mit Aufzählungszeichen werden mit `<ul>` angegeben. Dabei besitzen beide noch eine Unterebene, in der die Texte angegeben werden. Diese werden dann allerdings nicht mit dem `<p>`-Element ausgezeichnet, sondern mit dem `<li>`-Element. Näheres dazu erfahren Sie im Kapitel 19 »Listen und Aufzählungen abändern« (siehe S. 176).
- Textformatierungen werden mit `<span>` ausgezeichnet und müssen innerhalb eines `<p>`-, `<h1>`–`<h6>`- oder `<li>`-Elementes stehen. Bei den Textformatierungen gibt es auch noch die Auszeichnungen `<strong>` für fette Texte und `<em>` für kursive Texte. In InDesign werden mit Zeichenformaten formatierte Textbereiche als `<span>`, `<strong>` und `<em>` ausgegeben.

- Bilder werden mit `<img />` angegeben.
- Links sind mit `<a href="http://www.epub4id.de">Link</a>` ausgezeichnet.
- Tabellen enthalten Elemente wie `<table>`, `<tr>`, `<td>` etc. und sind von der Strukturierung etwas aufwändiger. Die Details werden im Kapitel 17 »Tabellen im ePub« (siehe S. 168) erklärt.

Dies mag jetzt etwas verwirrend klingen, falls Sie noch nie etwas mit HTML zu tun hatten. In den weiteren Kapiteln gebe ich jeweils im Detail an, wie Sie diese Codezeilen bearbeiten können und sich auch darin zurechtfinden.

## Klassen und IDs

Jeder E-Reader, jede E-Book-App und jeder Browser hat bereits Formatangaben für die vorher genannten Elemente integriert. Das heißt, die Browser enthalten Standardwerte zu der Farbe und Schriftgröße, um z.B. Texte innerhalb eines `<p>`-Elementes anzuzeigen. Jedoch haben viele Browser unterschiedliche Werte und die Darstellung kann variieren. Zudem ist der Inhalt von Büchern ja nicht durchgehend gleich formatiert. Texte enthalten Abweichungen in Schrift, Größe etc., die in das E-Book übernommen werden sollen. An dieser Stelle kommen die sogenannten Klassen und IDs ins Spiel. Damit lassen sich Variationen in den Formatierungen erstellen, ähnlich dem Prinzip, mit Absatz- und Zeichenformaten zu arbeiten. Der Code würde wie folgt aussehen:

```
<p class="xyz">Hier steht mein Text</p>
<p id="abc">Hier steht mein Text</p>
```

Der Hauptunterschied zur Verwendung von Klassen und IDs ist, dass Klassen mehrfach genutzt werden können. Eine ID darf nur einmalig einem Element zugewiesen werden. Beispiel:

```
<p class="xyz">Hier steht mein Text</p>
<p class="xyz">Hier steht mein Text</p>
<p class="xyz">Hier steht mein Text</p>
<p id="abc1">Hier steht mein Text</p>
<p id="abc2">Hier steht mein Text</p>
<p id="abc3">Hier steht mein Text</p>
```

Klassen und IDs können auch kombiniert werden. Beispiel:

```
<p class="xyz" id="abc">Hier steht mein Text</p>
```

In HTML können mehrere Klassen (also Formatierungen) angegeben werden. Diese werden durch ein Leerzeichen getrennt:

```
<p class="format1 format2 format3">Hier steht mein Text</p>
```

Werden Klassen und IDs kombiniert oder sind mehrere Klassen angegeben, hat die letzte Angabe immer Priorität. Klassen und IDs sind allerdings nicht nur auf das `<p>`-Element begrenzt. Diese können nahezu jedem Element zugewiesen werden. Bisher haben sie allerdings nur leere Angaben. Den Klassen und IDs müssen nun die Eigenschaften zugewiesen werden. Dies wird mit CSS (Cascading Style Sheets) erreicht.

## Regeln bei der Strukturierung

Falls Sie den Strukturcode in den XHTML-Dateien bearbeiten, gilt es einige Regeln zu befolgen, damit keine Fehler auftreten:

**Alle Elemente müssen ein schließendes Tag besitzen:**

**falsch:** `<div><p>Ich bin ein Mustertext</div>`

**richtig:** `<div><p>Ich bin ein Mustertext</p></div>`

**Elemente müssen korrekt verschachtelt werden:**

**falsch:** `<div><p>Ich bin ein Mustertext</div></p>`

**richtig:** `<div><p>Ich bin ein Mustertext</p></div>`

**Selektoren-Namen (Klassen und IDs) dürfen nicht mit einer Ziffer beginnen:**

**falsch:** `<p class="123">Ich bin ein Mustertext</p>`

**richtig:** `<p class="x123">Ich bin ein Mustertext</p>`

**Bei Element-Namen muss die Groß-/Klein-Schreibweise beachtet werden:**

**falsch:** `<H1>Ich bin ein Mustertext</h1>`

**richtig:** `<h1>Ich bin ein Mustertext</h1>`

**Leere Elemente müssen in der Tag-Struktur geschlossen sein:**

**falsch:** `<br>`

**richtig:** `<br />`

**Element-Attribute (wie Klassennamen) müssen in Anführungszeichen stehen:**

**falsch:** `<p class=name>Ich bin ein Mustertext</p>`

**richtig:** `<p class="name">Ich bin ein Mustertext</p>`

## CSS (Cascading Style Sheets)

Neben den XHTML-Dokumenten, die für die Strukturierung der Inhalte zuständig sind, benötigt das E-Book noch die Eigenschaften für die Formatierungen. Dies wird mittels CSS erreicht.

Das CSS kann direkt in die XHTML-Datei geschrieben werden, wobei dann allerdings nur diese Datei Zugriff auf die Formatierungen hat. Eleganter ist es, die Formateigenschaften in einer separaten Datei zu definieren, auf welche die einzelnen XHTML-Dateien zugreifen. Dabei handelt es sich um eine reine Textdatei, die mit der Dateiendung `.css` abgespeichert wird.

Möchten Sie z.B. Ihre Überschrift rot setzen, in der Schriftart Minion Pro, 24 Punkt und fett? In InDesign werden diese Einstellungen in einem Absatzformat definiert. In der durch InDesign exportierten CSS-Datei steht dann Folgendes:

```
h1 {
  font-family: "Minion Pro", serif;
  font-weight: bold;
  font-size: 2em;
  color: #d90000;
}
```

Falls Sie etwas Englisch verstehen, werden Sie einige Eigenschaften wiedererkennen. Die Einheiten wie Punkt oder Millimeter, die Ihnen aus dem Print vertraut sind, werden hier zu Webeinheiten umkonvertiert: Aus 24 Punkt werden 2 em, aus 10 Millimeter werden 28 px usw.

In dem oben genannten Beispiel wurde der Standardwert des Browsers überschrieben und alle Texte, die das `<h1>`-Element besitzen, bekommen diese Eigenschaften zugewiesen.

Angenommen, das `<h1>`-Element soll mit einer Klasse »titel« ausgezeichnet und 36 pt anstelle von 24 pt groß sein. Dann würde das Beispiel so aussehen:

```
h1.titel {
  font-size: 3em;
}
```

Die Klasse wird dabei in dem CSS mit einem Punkt, gefolgt von dem Namen, angegeben: `.titel`. Möchten Sie eine ID angeben, ist es ein Doppelkreuz, gefolgt von dem Namen: `#titel`. Bei diesen Angaben spricht man in der Fachsprache von Selektoren:

```
h1#titel {
  font-size: 3em;
}
```

Es gibt eine ganze Reihe von Eigenschaften, die für einen Selektor definiert werden können. InDesign generiert die notwendigen Eigenschaften bei der ePub-Erstellung. Im Laufe des Buches wird allerdings auch noch auf Eigenschaften hingewiesen, welche nicht automatisch beim Export erstellt werden.

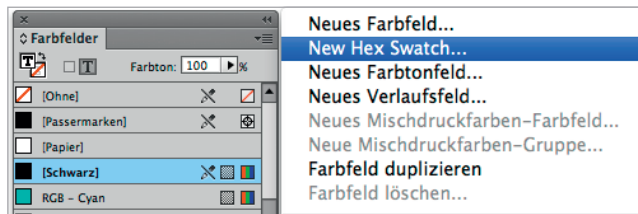
Sofern die Formatierungen in einer CSS-Datei deklariert wurden, muss diese Datei noch mit den XHTML-Dateien verbunden werden. InDesign erstellt diese Verknüpfungen bereits beim Export. Sollten Sie diese Verknüpfung manuell angeben oder abändern müssen, finden Sie diese im `<head>`-Bereich der XHTML-Dateien, wobei `idGeneratedStyles.css` in diesem Beispiel unsere CSS-Datei darstellt.

```
<head>
  <title>Ich bin ein HTML Dokument</title>
  <link href="css/idGeneratedStyles.css" rel="stylesheet"
    type="text/css" />
</head>
```

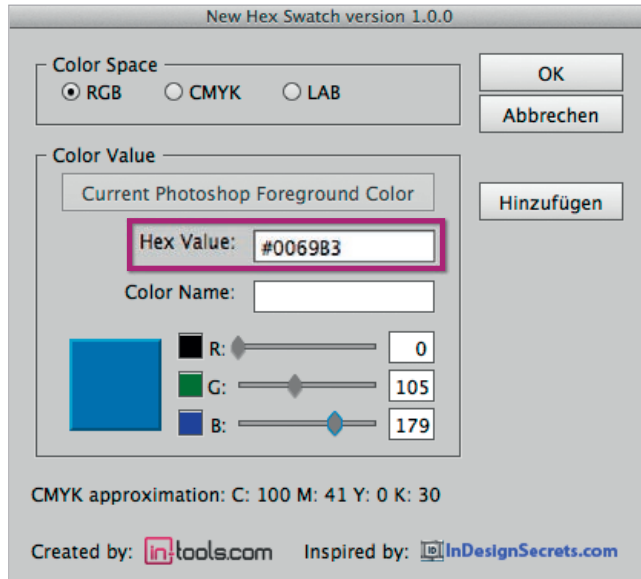
## Hexadezimalfarben definieren

Im Laufe des Buches komme ich manchmal auf Farben zu sprechen, die als Hexadezimalwert im CSS angegeben werden müssen. InDesign kann diesen Wert leider noch nicht von Haus aus bei den Farbfeldern anzeigen oder erstellen. InDesign konvertiert die bestehenden Farben beim Export in den Hexadezimalwert. Ein Skript mit dem Namen »Create Hex Swatch« (`ch_c1n7`) von der Firma In-Tools fügt diese Funktion in InDesign hinzu. Nachdem Sie das Skript heruntergeladen und entpackt haben, kopieren Sie den ganzen Ordner »Create Hex Swatch« (der im entpackten Ordner vorliegt) in das Skript-Verzeichnis im InDesign-Installationsverzeichnis (nicht »Scripts Panel«, wie es bei anderen Skripten der Fall ist). Starten Sie InDesign danach neu.

Im Kontextmenü des Farbfelder-Bedienfelds finden Sie nun »New Hex Swatch«:



Anschließend können Sie den RGB-Farbwert eingeben, um den gewünschten Hex-Wert herauszufinden, oder Sie geben den Hex-Wert ein, um den RGB-Wert herauszufinden.



Das Skript von In-Tools ermöglicht es, den Hex-Wert herauszufinden.

Durch *Hinzufügen* oder die Bestätigung mittels *OK* fügen Sie die Farbe der Liste im Farbfelder-Bedienfeld in InDesign hinzu.

Alternativ können Sie die Farbe in Photoshop beim Farbwähler eingeben, um den Hex-Wert für das CSS zu erhalten oder den RGB-bzw. CMYK-Wert des Hex-Wertes zu ermitteln:

