

HTML5

Inhalt:

Grundlegende Infos zu HTML5

DOCTYPE – und seine Bedeutung

Wichtigste Änderungen

Anpassungen in der css-Datei

Grundlegende Infos zu HTML5

Ältere Browser (z. B. IE8, Safari3...) unterstützen HTML5 noch nicht, so dass man nicht HTML5 verwenden sollte, wenn man davon ausgeht, dass der größte Teil der angestrebten Zielgruppe noch ältere Browser verwendet. Mobilbrowser von Smartphones und neuere Browser unterstützen HTML5 jedoch.

DOCTYPE – und seine Bedeutung

Betrachten wir nun mal genauer die erste Zeile in den HTML-Dokumenten, die wir bisher völlig ignoriert haben:

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

Diese Zeile enthält KEIN html-Element!!! Dies erkennt man daran, dass der < ein ! folgt. Mit dieser Zeile wird dem Browser der Dokumententyp angegeben. **HTML** gibt dabei dem Browser an, dass das erste Element der Seite <html> ist. **Public** gibt an, dass der HTML-4.01-Standard öffentlich zugänglich ist.

-//W3C//DTD HTML 4.01 Transitional//En sagt, dass HTML in der Version 4.01 genutzt wird und HTML-Markup in Englisch geschrieben ist. Zum Schluss wird die Quell angegeben, die den spezifischen Standard identifiziert.

Vereinfachung für HTML5 (und folgende):

```
<!doctype html>      oder      <!DOCTYPE html>
```

Das ist doch mal einfach, so dass man sich das auch merken kann. Und es kommt nichts weiter hinzu! Man muss keine Versionsnummern oder Ressourcenorte mehr angeben. Es genügt, wenn der Browser weiß, dass nun html folgt. In Zukunft soll es sich um einen lebenden Standard handeln, das bedeutet, dass alle Neuerungen hinzukommen können, aber ältere Versionen lesbar bleiben. Dies nennt man Rückwärtskompatibilität. Seiten, die man heute mit html schreibt, werden also auch lesbar bleiben, wenn sich html weiter entwickelt.

Somit können wir ab sofort immer diese einfache Zeile verwenden!

Wichtigste Änderungen

Die Standardisierungsorganisationen haben sich angeschaut, wie Webseiten mit Boxen gestaltet werden und dabei festgestellt, dass einige Boxen immer für bestimmte Bereiche verwendet werden. Daher hat man nun in HTML5 für bestimmte Boxen eigene html-Elemente vorgesehen. Dies bedeutet jedoch nicht, dass man nicht auch weiterhin mit dem div-Element arbeiten kann und soll. Hier eine Auflistung der wichtigsten Elemente:

<header>-Element

Inhalt, der am Anfang der Seite oder am Anfang eines Abschnittes der Seite steht. Das <header>-Element ersetzt somit die Box, die wir bisher mit <div id="header"> als Kopfzeile erstellt haben.

<footer>-Element

Inhalt, der sich am Ende der Seite oder am Ende eines Abschnittes der Seite befindet. Das <footer>-Element ersetzt somit die Box, die wir bisher mit <div id="footer"> als Fußzeile erstellt haben.

<nav>-Element

Enthält den Inhalt, der für die Navigationslinks in der Seite gedacht ist. Das <nav>-Element ersetzt somit die Box, die wir bisher mit <div id="menu"> als Kopfzeile erstellt haben.

<section>-Element

Mit diesem Element wird ein übergeordneter Abschnitt im Dokument definiert, so lassen sich miteinander verwandte Inhalte gruppieren. Dieses Element können wir beispielsweise für unsrern Inhalt verwenden. Um verschiedene Bereiche zu gruppieren, kann man die einzelnen section-Elemente klassifizieren, z. B. <section id="inhalt">

<article>-Element

Repräsentiert eine in sich abgeschlossene Abhandlung in der Seite, beispielsweise einen Blögeintrag, einen Forenbeitrag oder einen Zeitungsartikel. Hiermit lassen sich also bestimmte Textabschnitte gezielter formatieren als mit dem <p>-Element.

<aside>-Element

Enthält Inhalt, der den Seiteninhalt ergänzt, beispielsweise eine Anmerkung oder eine Seitenleiste. Dies Element beinhaltet also nichts, was zum Hauptinhalt gehört.

<video>-Element

Wird eingesetzt, um der Webseite Videos hinzuzufügen.

<audio>-Element

Wird eingesetzt, um der Webseite Musik oder andere Tonaufnahmen hinzuzufügen.

Anpassungen in der css-Datei

```
/*#header { */  
header {  
    background-color: #FFBF00;  
    margin: 10px 10px 0px 10px;  
    height: 108px;  
}
```

Den bisherigen Quelltext erkennst du noch im Kommentar. Also eigentlich muss man in der css-Datei nur # vor dem jeweiligen Boxennamen entfernen, da wir hier nun mit html-Elementen arbeiten.