Modernes Webdesign mit HTML und CSS

Webseiten sind nicht aus Papier!

Das klassische Printmedium baut auf bekannten Größen auf, z. B. dem Seitenverhältnis und den Abmessungen definierter Blattformate. Auch wenn man verschiedene **Tageszeitungen** in Bezug auf das Layout vergleicht, so verwenden alle den Spaltensatz, bestimmte Schriftgrößen, Zeilenlängen und Zeilenabstände. Diese **unterscheiden sich eigentlich nur wenig von einander**. Und trotz aller Gemeinsamkeiten hat jede Zeitung ihr eigenes Gesicht. Die Seitenaufteilung, die Gestaltung der Überschriften und Bilder und vieles andere beiten unzählige Gestaltungsmöglichkeiten, die in Summe das individuelle Erscheinungsbild prägen.

Im Internet als elektronisches Medium eröffnet sich eine völlig neue Welt. Zwar gelten prinzipiell ähnliche Gesetzmäßigkeiten für ein ausgewogenes Layout wie bei den Zeitungen, aber die Rahmenbedingungen unterscheiden sich völlig.

Im Internet gibt es nicht diese konstanten Größen für eine vollständige Kontrolle über das Layout – schon allein die große Anzahl **verschiedener BROWSER** ergeben eine Unterschied in der Darstellung einer Website.

- Jeder Browser besitzt eigene Stylesheets um Inhalte darzustellen, die sich aber teilweise stark unterscheiden.

- Das Ausgabemedium kann sehr unterschiedlich sein: Welches Medium verwendet der Nutzer? Einen Monitor, mobile Endgeräte wie z.B. Smartphones oder Tablets?

- Druckt der Nutzer die Website aus? Gerade längerer Text mag fast niemand am Bildschirm lesen. Die Aufbereitung des Layouts und der Seiteninhalte für den Druck erfordert wiederum besondere Sorgfalt, damit keine Information verloren geht.

- Bildschirmauflösung und Monitorgröße: der verfügbare Platz, den eine Webseite beansprucht ist unterschiedlich, je nachdem ab neben bei andere Programme am PC laufen und das Browserfenster vielleicht nicht den kompletten Bidlschirm belegten kann. Hat der Nutzer zusätzliche Werkzeugleisten (z.B. Google Toolbar) installiert, schrumpft der verfügbare Platz. Ebenso bei seitlichen Scrollbalken.

- Schriftarten und Schriftgrößen: auch bei der Schriftgestaltung sind dem Einfluss des Webdesigners hinsichtlich des Erscheinungsbildes auf dem Monitor Grenzen gesetzt. Ein Text in Schriftgröße 12 px mag auf einem 15 Zoll großen Monitor bei geringer Auflösung sehr gut lesbar sein. Auf einem 21 Zoll Flachbildschirm mit einer Standardauflösung von 1600 x 1200 Pixel wird er jedoch unter Umständen zur Qual.

Jeder Browser hat eigene Standard-CSS Vorgaben, die sich leider manchmal zwischen den Browsern unterscheiden.
Für den Webdesigner ergibt sich daraus ein Problem, da sich diese Vorgaben von Browser zu Browser zum Teil erheblich unterscheiden.

a)HTML – Beschreibung der Struktur

HTML wird oft als „Programmiersprache“ bezeichnet, was allerdings nicht zutrifft. Es handelt sich um eine „Auszeichnungs- oder Markup-Sprache“. Während eine Programmiersprache (wie z.B. JavaScript) Abläufe und Vorgänge beschreibt, dient eine Auszeichnungssprache dazu, die STRUKTUR von in Textform vorliegender Information inb einer semantischen (ihrer Bedeutung entsprechenden) Form zu formulieren.
Hierfür wird ein Informationsblock in Marken eingeschlossen, die seine Aufgabe oder Bedeutung bennenen und gleichzeitig seinen Anfang und sein Ende bestimmen. Diese Marken werden als „Tags“ bezeichnet. Man unterscheidet einen Start- und einen End-Tag.
Beispiel:

```
<h1>Dies ist eine Überschrift</h1>
```

b)CSS – Beschreibung der Präsentation

Arbeitet ohne Layouttabellen, ohne feste Breite und passt sich so den Gewohnheiten des Nutzers an. CSS gestattet die Berücksichtigung medienspezifischer Besonderheiten auf einfache Weise und ermöglicht eine optimale Präsentation der Inhalte.
Fixe Breiten sind keineswegs ein Ausschlusskriterium für modernes Webdesign, sie sind sogar deutlich in der Überzahl. Der Umgang mit variablen Breiten erfordert Sicherheit im Umgang mit dem CSS-Box-Modell und den Grenzen der verschiedenen Positionierungs-varianten die CSS anbietet.

Strukturiertes Arbeiten mit CSS
CSS wird allgemein als Formatierungssprache bezeichnet.

- Es ist jedoch keine Programmiersprache.
- Mit CSS wird lediglich beschrieben, welche Elemente auf welche Weise dargestellt werden und in welchen Zusammenhang eine definierete Darstellungsanweisung eingesetzt wird.
- Man bezeichnet CSS daher auch als Präsentationssprache.
- Jede Deklaration beginnt mit der Formulierung eines Selektors, mit dem die zu formatierenden Elemente des (X)HTML-Quellcodes spezifiziert werden. Innerhalb von geschweiften Klammern werden bestimmte Eigenschaften für die selektierten Elemente vergeben:

  mein_selektor { eigenschaften:wert; }

Kaskade

Die Kaskade ist ein **Ablauf**, nach dem der Browser entscheidet, wie er eine CSS-Deklaration zu behandeln hat.

Eine Kaskade kann man sich gut als mehrteiligen Brunnen vorstellen, in dem das Wasser von der obersten Schüssel in die untere usw. fließt. Stufe für Stufe. Oder in der Chemi als hintereinandergeschaltete Gefäße.

„Cascading Style Sheets“ funktioniert genau nach diesem Prinzip, indem mehrere CSS-Dateien hintereinandergeschaltet werden.

Die Kaskade ist eine Hilfe für den Browser. Dabei muss der **Browser** in Bruchteilen einer Sekunde die Befehle abarbeiten und sich an Regeln halten. Treffen nämlich mehrere CSS-Deklarationen für ein Element innerhalb eines oder mehrerer Stylesheets eines Dokuments aufeinander, so muss der Browser entscheiden, welche der ihm vorliegenden Deklarationen er zu beachten hat.

Beispiel: ein <p>-Element kann viele Eigenschaften haben, wie Schriftfarbe, HIntergrundfarbe, usw., eventuell eine Veränderung durch ein id-Element oder eine Klasse. Was gilt nun?

Daher gilt für das Abarbeiten der Angaben in der CSS-Datei folgende generelle Kaskade, damit der Browser die richtigen Werte ausgeben kann:

1.)Browser-Stylesheet:
„Browserinternes CSS“, das immer verwendet wird, auch wenn keine CSS-Datei eingebunden ist. Dort sind die Grundeigenschaften aller HTML-Elemente definiert.
2.)Autoren-Stylesheet:
CSS des Entwicklers, das er verwendet.
3.)User-Stylesheet:
Persönliches Stylesheet eines Nutzers. Diese ermöglicht es dem Benutzer selbst, gewisse Darstellung zu wählen, wie z.B. größere Schrift oder stärkere Kontraste. Wird über den Browser eingebunden.

Die Reihenfolge entspricht auch der, in welcher die CSS-Dateien im Browser eingebunden werden. Jede einzelne Stufe wird immer von oben und vom Beginn her abgearbeitet. Und am Schluss kann es nur einen Wert geben, der angezeigt wird.
Wesentlich ist, dass sich mehrere Stylesheets nicht ersetzen, sondern ergänzen. Bevor eine Webseite angezeigt wird, werden alle CSS-Regeln aus allen Dateien ausgelesen und ergeben ein „berechnetes Style“, das dann die Darstellung steuert.

Beispiel:
1.)Innerhalb einer CSS-Datei erhält die jeweils letzte Deklaration für ein Element, welches direkt beim Element steht, das höchste Gewicht. Frei nach dem Motto „Der Letzte gewinnt!“ Wird z.B. eine „margin: 20px“ vom Autor verwendet, aber in der Browser-Definition würde

stehen „margin: 0px“, dann würde natürlich die 2. Stufe, hier die Autoren-Definition umgesetzt werden. („Letztes gewinnt.“)
2.)Inline-CSS über dem style-Attribut sind höherwertig als Deklarationen im Header des Dokuments.

Link: http://www.thestyleworks.de/basics/cascade.shtml
Link: Standard-CSS des Firefox: www.h5c3.de/link-37-1

Die Vererbung (inheritance)

Die Vererbung bezeichnet ein weiteres Prinzip von CSS, das ohne Zutun der Webentwickler gegeben ist. Es bedeutet, dass HTML-Elemente bestimmte CSS-Eigenschaften der Elternelemente „erben“, wenn diese für das Kind Element nicht ausdrücklich definiert sind.

Vererbung bedeutet somit, dass bestimmte Eigenschaften (zum Beispiel font-family) von Vorfahren (zum Beispiel body) an Nachfahren (zum Beispiel div#navibereich ul) weitergegeben werden.

Vererbung macht ein Stylesheet übersichtlicher. Ein wichtiges Prinzip der Vererbung ist: von oben herab!
Definiere alles Mögliche weit oben (ab dem body-Element) und man wird dank Vererbung eine einheitliche Website erhalten. Ändere dann jeweils nur die Ausnahmen der Elemente, die man besonders hervorheben möchte.

Beispiel
Bei der Deklaration der Schriftart und Schriftgröße: Eine der ersten Regeln im Stylesheet für die Beispielseiten sieht – etwas verkürzt – so aus:

```
body {
font-family: Verdana, Arial, Helvetica, sans-serif;
}
```

Erklärung:
Alle Nachfahren von »body« erben die Schriftart.

Diese Deklaration gilt nicht nur für body, sondern auch für alle Nachfahren. Da im Prinzip alle Elemente einer Webseite Nachfahren von body sind, gilt die Schriftart für alle Elemente dieser Webseite, sofern im Rahmen der Kaskade nicht bereits etwas anderes definiert wurde.
Wenn es das Prinzip der **Vererbung nicht geben würde,** müsste man alle Elemente namentlich erwähnen, und die gleiche Deklaration könnte ungefähr so aussehen:

```
body, h1, h2, p, ul, li, a, strong, em, address {
font-family: Verdana, Arial, Helvetica, sans-serif;
}
```

Ohne Vererbung müssten alle Selektoren explizit aufgeführt werden.
Das ist deutlich umständlicher als vorher. Der geschickte Einsatz von Vererbung macht ein Stylesheet übersichtlicher.

Info:

Wird eine vererbte Eigenschaft vom Elternelement vererbt, wird diese Eigenschaft bei allen Kind-Elementen automatisch übernommen. Ausnahme: Es wird ein eigener Wert für die gleiche Eigenschaft definiert. Die Kaskade bricht hier ab und schaut nicht in den Standardwerten des Browsers nach.

Vererbung aushebeln (überschreiben)

Soll z.B. eine andere Überschrift als die, welche im <body> festgelegt wurde für die Überschrift gelten, so legt man einfach die gewünschte Schriftart für das untergeordnete Element fest und überschreibt somit die Vererbung:

```
body { font-family: Arial; }

h1, h2 { font-family: Courier; }
```

Beispiel für Vererbung: „Kombinierte Selektoren“

- **Nachfahrenselektoren (descendant)**
  Beispiel:
  ```
  p  span {
      font-style: italic;
  }
  ```
  Hier werden **alle** Nachfahren von p den Wert „italic“ erhalten. Und zwar nicht nur in der ersten Verschachtelungstiefe, sondern bis zum letzten Element.
  WICHTIG: der Abstand (Leerzeichen) zwischen p und span!

- **Kindselektoren (child)**
  Der Kindselektor ist die eingeschränkte Version des Nachfahreselektors.
  Beispiel:
  ```
  h1>span {
      font-weight: bold;
  }
  ```
  Hier erhält nur die erste Verschachtelungstiefe ab <h1> die Eigenschaft. Aber auf dieser Verschachtelungstiefe erhalten alle span die Eigenschaft „bold“.
  WICHTIG: das Größer-Zeichen bestimmt einen Kindselektor.

- **Mehrere Selektoren**
  Wenn mehreren Selektoren die gleiche Eigenschaften zugeteilt werden sollen, dann werden diese Selektoren per Komma getrennt.
  Beispiel:
  ```
  h1, h2, div, p {
      font-style: italic;
  }
  ```

Fragen:

- Woraus besteht eine Kaskade?
- Beschreibe die drei Stufen der (großen) Kaskade.
- Beschreibe die vier Stufen der kleinen Kaskade.
- Wann werden Standardwerte des Browsers übernommen? (Wenn keine Eigenschaften ermittelt werden konnten.)
- Wo ist es am sinnvollsten, grundlegende Einstellungen für die ganze Website zu erstellen, an welchen Platz der CSS-Datei?
  (Nach dem Prinzip „von oben nach unten“ sind globale Einstellungen im <body> gut.)
- Was sind Kombinierte Selektoren wie z.B. Nachfahren- und Kindselektoren?
  (Nachfahrenselektoren vererben ihre Eigenschaften allen nachfolgenden Elementen, egal wie tief die Verschachtelungstiefe geht. Es gilt: von oben einfach abwärts. Nachfahrenselektoren werden mit einem Abstand zwischen den zwei Selektoren definiert. Z.B. p span
  Kindselektoren sind den Nachfahrenselektoren sehr ähnlich, der einzige Unterschied: Kindselektoren wirken nur eine Verschachtelungsebene tiefer. Die Anwendung erfolgt mit einemn „Größer-Zeichen“. Z.B. h1>span.)