Modernes Webdesign

1. Vorwort

Ich werde in diesem Dokument das entstehens eines Webdesigns bzw. Entwurfes dokumentieren. Das Dokument soll den Entwurf, die grafische Umsetzung und die eigentliche Umsetzung in valides Markup dokumentieren. Geplant ist zur Zeit die grafische Umsetzung mit Gimp, und die eigentliche Umsetzung in Markup durch HTML5 und CSS3.

Das Design soll anhand von Webstandards geplant und später auch umgesetzt werden. Wir werden unter anderem Themen wie Farbenlehre, Typografie, Barrierefreiheit und natürlich Semantik durchkauen.

Kommentar des Autors:

Natürlich bin auch ich „nur“ ein Mensch, und auch ich habe Motivationsschwächen. Deswegen wird dieses Dokument Schrittweise erweitert. Ob das Dokument jemals fertig gestellt wird, weis keiner. Jedoch hoffe ich das inständig. Wenn ihr gefallen an dem Dokument habt, dann teilt es mir bitte mit, und zwingt mich zum weiter machen. Ein wenig Druck von Aussen sollte nich schaden. Naja, egal, ich wünsche viel Spass und gutes Gelingen :)

Kontakt:

Solltet ihr auf Fehler stoßen, oder Fragen haben, so kontaktiert mich doch bitte per Mail: xcr4cx [äät] googlemail [punkt] com.

So, nun aber viel Spass!

Inhalt

Die Themen im Überblick. (Anhand des aktuellen Umfangs)

1. Vorwort
2. Grundlegendes
3. Der Entwurf
4. Das Layout
5. Der Stylesheet
6. Die Ausgestaltung

2. Grundlegendes

Dieses Dokument soll Fortgeschrittenen, Profis, Anfängern, Hobbydesignern und allen Interessierten modernes Webdesign mit Webstandards näher bringen.

Aber um überhaupt erstmal an dieses riesige Themenfeld ranzugehen, müssen wir erstmal klären, was Webstandards sind.

Webstandards sind Technologien, die durch das World Wide Web Consortium (W3C)[1)] und anderen Organisationen, Institutionen und Unternehmen zusammen entwickelt worden, um Dokumente im Internet langlebiger zu machen, und gleichzeitig sicher zu stellen, dass möglichst viele Nutzer diese auch betrachten können – möglichst unter gleichen Bedingungen wie jeder andere.

Den Ursprung hat das Ganze Anfang der 1990er gefunden. Zu der Zeit gab es noch keine solcher Formatierungsmöglichkeiten wie heute. Web-Dokumente kamen meist trist und grau daher. Genau das wollte Netscape nun ändern, und schmiss den Netscape Navigator auf den Markt.
Dieser brachte diverse Formatierungen wie Tabellen, Frames, Script- und Multimediaelemente, sowie mehr Farben mit, die es bisher noch nicht gab.
1995 hielt Netscape somit 80% des Browsermarktes. Nun lief Microsoft gefahr, durch diese Entwicklung „technisch Irrelevant“ zu werden. Um das zu verhindern, veröffentlichte Microsoft im August '95 den Internet Explorer. Und der Wettkampf war hiermit nun feierlich eröffnet.

Jeder der beiden Hersteller versuchte nun, sein Angebot so attraktiv wie möglich zu gestalten. Jeder bastelte seine eigenen Erweiterungen zur HyperText Markup Language in sein Produkt, und lies es, so glaubten sie, besser aussehen, als die Konkurrenz. Dadurch wurden Standards fast unmöglich, da jeder der beiden Giganten seinen eigenen „Standard“ definierte.

Das Ende der Geschichte kennen wir wohl alle, siegreich zog Microsoft 1998 aus dieser Schlacht, und Netscape gab seinen Navigator frei ab.

[1)] Das W3C ist ein von Tim Berners-Lee gegründetes und bis heute geleitetes Gremium, welches die zur Standardtisierung des Webs entworfenen Technologien verwaltet und entwirft bzw. Verbessert.

Natürlich gab es Webstandards schon vor dem „Browserkrieg“. Hauptziele des W3C sind nämlich Plattformunabhängigkeit und Barrierefreiheit, schon seit seiner Gründung '94. Denn Standards ermöglichen es, dass Entwickler einen Richtwert haben, um ein Dokument immer gleich aussehen zu lassen.

Nun wissen wir ja, dass dem bis heute nicht so ist. Und genau das ist der Grund, sich als Designer an Standards zu halten, selbst wenn der am meisten verbreitetste Browser diese anders implementiert. Denn genau so zwingen wir die Monopolisten, ihre Stellung zu Standards zu ändern. Ein gutes Beispiel ist der Internet Explorer. Vieles beherrscht er nur teilweise oder gar nicht, ein aus unserer Sich grausiger Zustand, der sich nur durch die Massenbewegung in Richtung anderer Browser wie Firefox, Opera oder Chrome geändert hat. Genau so erreichen wir, dass der Internet Explorer dazu lernt, und somit jeder das Dokument gleichermaßen betrachten kann.

Webstandards sind also Bestimmungen, die durch das W3C ausgegeben werden, um Dokumente auf gleiche Weise vorzustellen. Beispiele hierfür wären der HTML-, CSS-, XHTML- oder der XML-Standard. So können die Browserentwickler sich nun an diese „Leitfäden“ richten, und Dokumente gleichermaßen verarbeiten.

Als Webstandards bezeichnet man nun aber nicht nur den HTML- oder CSS-Standard, sondern qualitative, sprich unmessbare und subjektive und nicht verpflichtende Standards, die durch die hohe Nachfrage einfach benötigt werden. Die Rede ist von Barrierefreiheit und Benutzbarkeit.

Diese Standards haben aber genau dasselbe Ziel wie das W3C. Sie wollen erreichen, dass möglichst viele Nutzer genau dasselbe zu sehen bekommen. Sie sind aber nicht durch Code umzusetzen, bzw. nur indirekt. Man muss für sie einen Gewissen Instinkt entwickeln, da es nirgends Leitfäden, sondern nur Tipps und Hinweise für Barrierearmes bzw. Barrierefreies und Benutzbares Webdesign gibt. Natürlich gibt es hier viele Ideen, die weit verbreitet sind, bspw. Skip-links. Dazu aber an anderer Stelle mehr.

Und Genau dieses Gespür soll uns von Anfang an zur Barrierearmen und ansehlichen Gestaltung unseres Designs helfen.

3. Der Entwurf

Damit wir unseren ersten Entwurf entwerfen können, müssen wir uns zunächst etwas Gedanken machen.

Worüber?, wirst du dich fragen. Nun wir denken zunächst über Inhalt und Struktur nach. Das sind unsere ersten Bausteine die wir setzen müssen.

Die Struktur bezeichnet den logischen Aufbau der Informationen, die wir präsentieren wollen. Die Faustregel dafür: „Der Nutzer muss den Aufbau verstehen können.“, er muss ihn nachvollziehen können. Komplexität ist hier in der Regel fehl am Platze. Denn wenn wir unsere Informationen komplex und unerkennbar gestalten, weis der Nutzer nicht, was er machen soll/muss. Und da ihn das womöglich überfordert – wer will schon großartig nachdenken – schliesst er unsere Seite, und kommt wahrscheinlich nie wieder. Aber sind unsere Informationen gut strukturiert dargestellt, bleibt er, erzählt es vielleicht sogar weiter. Also merken wir, dass gute Struktur eigentlich nur Vorteile bestitzt.

Für eine gute Struktur sollten wir die Informationen, die wir präsentieren möchten, grob vorgliedern. Wir müssen sie also einteilen, ihnen Kategorien zuweisen, bzw. übergeordneten Punkten, und sie in Verbindung setzen.

Man sollte sich folgende Gedanken dazu bereithalten:

Welche Kategorien verbinde/assoziiere ich miteinander?
Welche sind kombinierbar? Welche kann ich einander unterordnen?
Was würde ich als nächstes erwarten?
Wo würde ich das Element erwarten?

Solche Fragen müssen wir uns dazu stellen. Aus den Antworten haben wir dann unsere Grundgliederung vorliegen, und können daraus eine Navigationsstruktur ableiten.

Damit unsere Besucher sich aber auch im allgemeinen wieder finden, sollten wir immer einen Link auf den „Start“ bereitstellen, links also solche markieren, und den Benutzern immer den aktuellen Standort zeigen.

Letzteres wird meist durch sogenannte „Breadcrumb-navigations“ umgesetzt. Sie sehen wie folgt aus:

Home > Politik > Deutschland > Hessen

So findet sich der Nutzer immer zum letzen Ausgangspunk zurück.

Die Anordnung des Inhalts sollte der Struktur folgen. Hierarchisch angeordnete Informationen sollten im ganzen Projekt so umgesetzt werden.
So fühlt sich der Nutzer auch recht heimisch.

Wir haben jetzt schon einmal den elementaren Grundstein für unser Projekt gelegt. Als nächstes müssen wir uns aber der Zielgruppe zuwenden. Wir müssen uns fragen, An wen richtet sich das Angebot eingetlich?, denn wenn wir wissen, an wen sich das ganze richtet, wissen wir auch, was die Zielgruppe(n) erwarte(t/n). Man kann gewissen Zielgruppen bestimmte Verhaltensmuster zuordnen. Mit diesen Gewohnheiten müssen wir uns im Vorfeld beschäftigen, sie gar studieren. Passend währen Statistiken, Studien und Umfragen. All diese Quellen sollten uns ein umfangreiches Bild über die Erwartungen und Gewohnheiten der Zielgruppe geben.

Wenn wir jetzt auch den zweiten Grundstein gelegt haben, können wir mit ein paar Richlinien einen ersten Entwurf erstellen.

Wir sollten uns bei dem ersten Entwurf vorallem an die Effizienz, Wartung und Verständlichkeit halten. Das heisst, dass wir keine überladenen, anfälligen und komplexen Layouts erstellen sollten. Wir sollten alles so einfach wie möglich, jedoch immer an den Wünschen der Zielgruppe gestalten.

Es ist extremst wichtig, auf eine gute Wartbarkeit des Projektes zu achten, wir sollten alles wiedererkennbar und zukunftsorientiert gestalten.

Mit den vorhergehenden Betrachtungen im Unterbewusstsein legen wir nun los, und machen uns ein erstes Bild des Ergebnisses.

In unserem Beispiel wird das ganze eine Präsenz für eine Zeitung. Wir fragen uns zunächst, worüber handelt die Zeitung, welches Spektrum an Themen bietet sie überhaupt an? Schliesslich: Wer ist unsere Zielgruppe?

Diese Fragen sind in unserem Beispiel wie folgt zu beantworten:
Die Zeitung ist eine links-orientierte Zeitung mit Themen wie Politik, Wirtschaft, Kultur, Umwelt, Technik. Jetzt müssen wir uns fragen, was können wir kombinieren? Dazu sei noch angemerkt, wir müssen auch auf die Stärke des Themas achten. Ist zum Beispiel die Kategorie Umwelt recht umfangreich, wäre es nur umständlich, sie irgendeiner der anderen Kategorien unterzuordnen.

Ich komme zu dem Schluss, dass keine der Kategorien unterzuordnen ist.
Sie sind einfach an sich zu groß.

Jetzt fragen wir uns, was will der Leser zuerst lesen? Bei einem politisch-linken Blatt natürlich politische Artikel und Nachrichten.

Ich muss anmerken, dass man bei einem solchen Projekt erstmal etwas über die linke Politik lesen muss, bevor man prioritäten setzt.

Denn dann können wir auch folgende Prioritäten nachvollziehen: Politik, Wirtschaft, Umwelt, Kultur, Technik. Die ersten drei Rubriken sind recht gut mit linkem Gedankengut vereinbar und deswegen unser Hautpaugenmerk.

Die restlichen zwei Rubriken sind eher untypisch und nach der Interesse der Leser gerichtet. Technik ist meiner Ansicht nach zum Beispiel nur bedingt Interessant für politisch interessierte Leser.

Nun haben wir unsere Inhaltlichen Rubriken zusammen. Normalerweise sollte man nun unterkategorien festlegen. Da wir aber kein komplettes Blatt entwerfen wollen, überlassen wir das der „Redaktion“ der Zeitung. Das sollte in Auftragsfällen alles vorgegeben sein. Deswegen belassen wir es dabei. Dummykategorien werde ich trotzdem einfügen, also nicht wundern.

Wir gehen jetzt zu der Leseranalyse über. Da eine linke Zeitung wohl eher intelektuelle anspricht, möchten wir unser Angebot auch auf den Personenkreis optimieren. Da man den intelektuellen wohl kaum ein Surfverhalten zuordnen kann, nehmen wir die Standards. Das betrifft Auflösung, aktuell 1024x768, die Anordnung der layout-bestandteile, wie Navigation, und alles weitere.

Wir möchten den Lesern zudem auf der Startseite aus jeder Rubrik die 2 neusten Texte präsentieren und einen besonders relevanten Artikel hervorheben. Wir möchten den Lesern also folgendes Bild liefern. (siehe nächste Seite - Wegen der Übersicht digitalisiert). Für die größe des Entwurfes empfehle ich, die höhe zu vergrößern, als die tatsächliche Auflösung sein wird. Desweiteren sollte man bei den Maßen der Flächen die Scrollbars und Toolbars mit einbeziehen. So dass wir zu einer Breite von 960px kommen. So bleibt genug Platz für die Scrollbars.

Der erste Entwurf ist natürlich nicht endgültig, sondern erstmal eine Idee, woran man sich später orientieren kann. Der Entwurf ist erstmal ganz grob gestaltet.


Logo
suche
kategorien
subkategorien
artikel-highlight
sidebar
neusten texte aus den jeweiligen kategorien
footer

3. Das Layout

Dieser Entwurf zeigt uns nun, welche Elemente wir erstmal in Markup übernehmen können. Bevor wir dies tun, möchte ich aber noch ein äußerst wichtiges Thema zur Sprache bringen, Semantik.

Das Wort Semantik bedeutet soviel wie Bedeutungslehre, ziehlt also auf die richtige Verwendung von Wörtern bzw. in unserem Fall Tags ab. Das heisst also, dass wir Tags nur entsprechend ihrer vorgesehenen Bestimmung verwenden sollten, um sie semantisch zu verwenden.

Das wohl beste Beispiel für völlig ignorierte Semantik ist wohl jedem schon mal begegenet. Die rede ist von Tabellen. Diese wurden aufgrund ihrer Eigenschaften gerne auch mal benutzt, um ein Layout zu erzeugen. Nun kann man sich vorstellen, dass man das nicht tut. Das ist genau so, wie wenn du versuchst, deinen neuen HD-TV mit Scheuermilch zu putzen. Äusserlich mag es richtig und sauber sein. Unter der Oberfläche jedoch durchweg negativ.

Dieser Trend, der seinen Ursprung in den 90ern hat, ist bis heute leider immer noch weit verbreitet, sogar unter „Profis“.

Also versuchen wir, jeden tag nur mit seiner ürsprünglichen Bedeutung zu verwenden. Und putzen unseren Fernseher nur mit den extra dafür Entwickelten Produkten. Bei uns hieße das dann soviel, dass wir die Tags immer gewissenhatft und ihrer Bestimmung entsprechend verwenden.

Nun genug der Metaphern. Setzen wir unsere Struktur in valides, sprich nachweislich richtiges Markup um. Wie im Vorwort erwähnt, setzen wir das ganze in HTML5 um. Dazu sei gesagt, dass HTML in der Versionsnummer 5 noch experimentell und nicht als offizielle Spezifikation vorliegt. Deswegen wird es von dem W3C nicht im Produktiveinsatz empfohlen. Warum ich dies trotzdem tuhe, hat zwei simple Gründe. Zum einen mache ich mir das Leben durch die neu eingeführten bzw. optimierten Tags einfacher, und zum anderen will ich mit diesem Verhalten die Browserentwickler dazu bewegen, die neuen bzw. geplanten des W3C möglichst schnell und sauber zu übernehmen. Ich bin übrigens nicht der einzige, der so versucht, die neuen Standards schneller zu etablieren.

Das Markup werde ich ersmal blind „runterschreiben“ und dann im folgenden auf die einzelnen Tags eingehen.

```
<!DOCTYPE HTML>
<html lang="de">
<head>
    <meta charset="UTF-8">
    <title>Der Samstag – Politik, Wirtschaft ...</title>
    <meta name="description" content="Der Samstag">
    <meta name="keywords" content="Politik, Wirtschaft ...">
```

```
    <link rel="stylesheet" type="text/css" href="css/style.css">
</head>
<body>

<section id="head">
     <header>
          <h1>Der Samstag</h1>
          <div id="suche">
          </div>
     </header>
</section><!-- head ende -->

<section id="navigation">
     <nav id="haupt_nav">
     </nav>
     <nav id="sub_nav">
     </nav>
</section><!-- Navigation ende -->

<section id="main">
     <div id="hightlight">
     </div><!-- artikel hightlight ende -->
     <div id="zusammenfassung">
     </div><!-- zusammenfassung ende -->
     <div id="sidebar">
     </div><!-- sidebar ende -->
</section><!-- main ende -->

<section id="footer">
</section>
</body>
</html>
```

Das ist jetzt unser Grundlegendes Layout. Das Speichern wir als index.html ab. Jede Serversoftware, die was von sich hält, sucht in einem Verzeichnis nach einer Index-Datei, daher der Name.

Eingeführt wird das Layout mit `<!DOCTYPE HTML>` , dieser Tag ist nicht wirklich bestandteil der HTML-Sprache, sondern eine Angabe für den Browser, wo er die einzelnen Tagdefinitionen herbekommt. Da das ganze heutzutage unnötig ist, wird dieser Tag leer gelassen. Um den Browser aber nicht zu irritieren, wird er erstmal beibehalten.

Weiter geht es mit dem `<html lang="de"></html>` Tag. Dieser beginnt bzw. beendet das Dokument. Ein Tag folgt folgendem Schema:

```
<bezeichner attribut="wert">inhalt</bezeichner>
```

Ein Tag kann viele Attribute haben, nachzulesen unter:

http://w3schools.com/tags/ref_standardattributes.asp

Die einzelnen Tags sind dieser Addresse ebenfalls zu entnehmen. (Start > HTML > HTML Tag List)

In unserem Fall ist der Tag der `<html>` - Tag. Wir weisen dem Attribut lang (Sprache) den Wert „`de`“ zu. „`de`“ steht für Deutsch. So weiß der Browser und andere Programme, das dieses Dokument in deutscher Sprache verfasst ist.

Als nächstes folgt der `<head>` - Tag. Dieser leitet den Kopfbereich ein, der für die Benutzer eigentlich kaum eine Rolle spielt. Wichtiger sind die darin enthaltenen Elemente eher für die Software. Der `<head>` - Bereich gibt nämlich Suchwörter, Beschreibung, Titel des Dokuments, Kodierung und vieles mehr an.

Gerade weil es sehr viele dieser sog. Metatags gibt, werde ich im einzelnen nur auf die verwendeten Metatags eingehen.

Der erste Tag im Head-bereich ist der `<meta charset="UTF-8">` - Tag. Dieser verrät dem Browser, welches Charakterset, oder welchen Zeichensatz ich verwende. Es gibt nämlich je nach Region verschiedene Charakter, die nicht überall verwendet werden. Wir verwenden hier das UTF-8 Zeichenset. Dieses ist das Universelle Zeichenset schlechthin. Es enthält alle benötigten Zeichen, die wir brauchen, und ist zudem bei fast jedem Verfügbar bzw. vorinstalliert.

Der folgende `<title>` - Tag enthält den Dokument-titel. Dieser Tag ist Pflicht. Genau wie das <head> - Tag an sich, das `<html>` der `<!Doctype>` - und der `<body>` - Tag.

Jetzt kommen wir zu den Metatags. Diese Tags enthalten nicht benötigte Informationen, die wir dem Nutzer oder der Software zur Verfügung stellen können. Es gibt zahlreiche dieser Metatags, wir können sogar eigene Erfinden. Nachzulesen unter http://w3schools.com/tags/tag_meta.asp.

Der `<link>` - Tag gibt einen Pfad zu einer Ressource an, beispielsweise ein Script oder ein Stylesheet. In unserem Fall ist es ein CSS-Stylesheet. Es liegt im Ordner „CSS“.

Wir kommen nun zum `<body>` - Tag, dieser beinhaltet den eigentlichen Dokumentinhalt. Sprich, hier ist der eigentliche Inhalt zu finden. In ihm sind keine Pflichtinhalte zu finden. Wohl aber unsere Struktur, die wir dem Entwurf entnommen haben. Es gibt die Sektionen „*head*“, „*navi*“, „*main*“ und „*footer*“. Jeder dieser Bereiche steht für einen übergeordneten Container in unserem Layout. Größere Sektionen werden laut HTML5-Standard mit dem `<section>` - Tag umgesetzt. Dieses kann wiederum kleinere Bereiche enthalten. `<div>` - Tags. Die `<div>` – und `<section>` - Tags sind sogenannte Container. Container enthalten und gruppieren verschiedene Elemente und ermöglichen so eine gemeinsame Formatierung und weisen auf eine semantische

Zusammengehörigkeit hin.

Der `<header>` - Tag weist eine Kopfzeile aus. Sie kann diverse Angaben wie Überschriften und Zeitangaben enthalten.

Das `<nav>` - Element enthält Links, die der Navigation innerhalb des Dokumentes dienen. Diese können entweder durch Listen-Tags (ul,ol) oder direktes einfügen bereitgestellt werden:

Beispiel 1:

```
<nav>
     <ul>
          <li><a href=”#”>« Zurück</a></li>
          <li>|</li>
          <li><a href=”#”>Vor »</a></li>
     </ul>
</nav>
```

Beispiel 2:

```
<nav>
     <a href=”#”>« Zurück</a>
     |
     <a href=”#”>Vor »</a>
</nav>
```

Allgemein ist jedoch die erste Variante zu empfehlen, da auch ältere Browser mit ihr kompatibel sind. Mit dem obigen Code bekommen wir dann in beiden Fällen bei voller Kompatibilität folgendes Bild:

```
« Zurück | Vor »
```

Zu beachten ist, dass wir die Namen den IDs und Klassen eindeutige Namen zuweisen. Sonst finden wir uns nachher nicht mehr im Quelltext zurecht. Desweiteren findet sich unser Kunde dann auch besser durch den Quelltext und fühlt sich nicht übergangen. Um das ganze zu unterstützen sollten wir viel Kommentieren. Aber nicht nur uns oder der Kunden wegen, sondern weil das Web open-source ist. Jeder kann den Quelltext lesen. Und wenn jemand etwas ähnliches wie von uns verwendet haben möchte, dann kann er sich den Code anshauen. So hilft man sich sicherlich gegenseitig.

Kommentare sind folgendermaßen aufgebaut:

```
<!-- hier steht der Kommentar -->
```

Erwähnenswert wäre noch, dass wir darauf achten, alle Tags richtig zu schliessen usw. Leichter geht das mit einem Editor der Syntax-highlightning unterstützt. Für Windows wäre da Phase 5 oder ähnliches, und für die Linuxer sicherlich Interessant ist Gedit, Bluefish und Kate. Bluefish ist eine sogenannte IDE, eine Entwicklungsumgebung. Sie bietet Code-vervollständigung, eine Referenz und verschiedene Dialoge, die das Entwickeln leichter machen. Das KDE-Äquivalent wäre Quanta. Natürlich ist das keine Pflicht, wer mag kann auch mit vi oder notepad seinen Code schreiben.

4. Das Layout

Wir wollen uns nun um ein Grundlegendes Stylesheet kümmern. Wir haben das Stylesheet im Headbereich unter „ `css/style.css`“ angegeben. Man sollte auf die korrekte Ordnerangabe achten, um ewiges Fehlersuchen zu vermeiden. Wir erstellen also im selben Ordner wie die Hauptdatei „index.html“ den Ordner CSS. In dem CSS ordner erstellen wir entweder die Datei style.css, oder speichern den Stylesheet nachher unter dem Namen.

Da es, wie wir wissen, eine große Auswahl an Browsern gibt, und die nicht alles gleich darstellen, setzen wir erstmal die ganzen Elemente zurück. Sprich auf null-werte. Im Allgemeinen bekannt als CSS-Reset.

Hierzu verwenden wir den Universal-Selektor, dieser spricht alle Elemente an.

```
*
     {    margin: 0;
          padding: 0;
          text-decoration: none;
          list-style-type: none;
     }
```

Wir haben den Attributen bzw. Eigenschaften des Universalselektors * wieder Werte zugewiesen. Zuerst den Aussenabstand `margin` und den innenabstand `padding` des Elementes auf `0` gesetzt. Dann haben wir jegliche Textdekorationen wie Unterlinien und ähnliches entfernt, welche zum Beispiel bei Links zu finden ist. Als nächstes haben wir den Listen die „Bullets“, also die Aufzählungszeichen genommen. Wir sollten bei den letzen beiden nicht vergessen, diese später wie gewünscht zu stylen.

Es gibt natürlich, wie sollte es anders sein, diverse möglichkeiten seinen gesamten Style zu reseten. Es gibt diverse Stylesheets zum verwenden. Einige sind umfangreicher als meins, welches recht

grundlegend ist. Für nähere Informationen empfehle ich diesen Artikel auf Webmasterpro.de.