2 Cascading Style Sheets CSS

HTML bietet nur sehr rudimentäre Möglichkeiten das Aussehen von Internetseiten zu beeinflussen. Die heute üblichen komplexen Layouts moderner Internetseiten gehen weit über das hinaus, was das ursprünglich standardisierte HTML leisten sollte. Über die Jahre wurden immer mehr Funktionalitäten hinzu definiert, was zu einem schwer les- und wartbarem Code führte. Dieses Durcheinander machte es Web–Entwicklern schwer strukturierten Seiten–Code zu erstellen. Strukturierter Code besteht aus standardisierten HTML Elementen die es ermöglichen sollen die Elemente einer Seite zu Klassifizieren. Insbesondere Suchmaschinen können aus einem strukturierten Seitenaufbau bessere Ergebnisse liefern, da sie Überschriften (z.B. <h1>) anders bewerten können als beispielsweise folgenden Ausdruck:

```
<font size=”+3” face=”Helvetica” color=”red”>Seitentitel</font>
```

Dieser Code kann im Browserfenster einen für das menschliche Auge klar als Überschrift zu erkennenden Schriftzug erzeugen, für die Struktur der Seite ist das <font> - TAG jedoch gänzlich ohne Bedeutung. Eine Suchmaschine gewichtet demnach den eingeschlossenen Text nicht stärker. Bei der Suchmaschine Google erhalten gut strukturierte Seiten einen höheren Rang.

Zusammenfassend lassen sich folgende Probleme benennen, die eine unstrukturierte Seitenprogrammierung mit sich bringen:

- Indizierung unstrukturierter Seiten ist unglaublich schwierig
- Zugänglichkeit der Internetseite ist nicht, oder nur schwer gewährleistet. Z.B. fehlt sehbehinderten Menschen die Möglichkeit sich sinnvoll den Kontent der Seite vorlesen zu lassen.
- Das Warten von strukturierten Seiten ist wesentlich einfacher. Änderungen sind schneller und mit einer geringeren Fehlerhäufigkeit erledigt.

Natürlich ist das Problem der Verschmutzung von HTML mit Präsentationselementen dem W3C nicht entgangen. 1995 veröffentlichte das Konsortium deshalb ein Konzept mit dem Arbeitstitel CSS.

2.1 Stilvorlagen in CSS

CSS bietet umfassende Möglichkeiten der Gestaltung eines Dokumentes. Mehr als es HTML jemals konnte. Jedes HTML-Element kann mit einer eigenen Hintergrundfarbe, Textfarbe, Schriftart u.v.m. versehen werden. Man kann den Platz um Elemente frei Definieren und Rahmen und Dekorationen angeben.

Nehmen wir als Beispiel die erste Hauptüberschrift auf einer Seite, die normalerweise dem Seitentitel entspricht. Die korrekte HTML-Bezeichnung lautet:

```
<h1>Seitentitel</h1>
```

Nehmen wir weiter an Sie möchten nun den Überschriftentext folgendermaßen Definieren:

- Farbe: rot
- Schrift: Times, kursiv, unterstrichen
- Hintergrund: gelb

Möchten Sie diese Einstellungen mit reinem HTML vornehmen müssen Sie eine ganze Reihe Klimmzüge machen. Sie müssen die Überschrift in eine eigens dafür definierte Tabelle packen, hierein das <h1>-TAG setzen und dieses mit einer ganzen Reihe anderer Elemente, wie z.B. *font* und *u* voll laden.

Mit CSS erschlagen Sie das ganze Problem mit nur einer einzigen Regel:

```
h1 {color: red; font: italic 2em Times, serif; text-decoration:
underline; background: yellow;}
```

Es müssen nicht zwangsläufig Formatierungen vorgenommen werden die schon in HTML definiert sind. Es gibt eine riesige Menge an Möglichkeiten, von denen in reinem HTML keine Rede ist:

```
h1 {color: red; font: italic 2em Times, serif; text-decoration:
underline; background: yellow url(tux.png) repeat-x; border:
1px solid red; margin-bottom: 0; padding: 5px;}
```

Hier wird zusätzlich ein Hintergrundbild hinter den Text gelegt und dieses nur horizontal wiederholt. Umrandet wird der Text von einem Rahmen mit einem Pixel breite, der mindestens einen Abstand von 5 Pixeln zum Text haben soll. Außerdem haben wir den unteren Rand des Elementes entfernt. Dies alles sind Dinge, die HTML nicht mal Ansatzweise beherrscht, trotzdem sind sie nur ein Vorgeschmack auf das, was mit CSS möglich ist.

Die in CSS definierten Angaben gelten für das gesamte Dokument. Das bedeutet, dass sich Änderungen in einer bereits definierten CSS-Regel immer auf das gesamte Dokument auswirken. Möchte man beispielsweise das Aussehen aller Überschriften des Typs <h1> in einem Dokument ändern, so reicht es aus die für das TAG h1 definierten Einstellungen im CSS entsprechend abzuändern. Ohne CSS müsste man das gesamte Dokument nach <h1>-TAGs durchsuchen und die Änderungen von Hand vornehmen.

Des Weiteren ist es möglich Eigenschaften auf andere HTML-TAGs abzubilden, indem man bestimmte CSS-Regeln für mehrere TAGs gültig macht.

```
h1, h2 {color: blue; background: grey;}
```

Diese Regel ist sowohl für das TAG <h1> als auch für das TAG <h2> gültig. Alle Änderungen und Ergänzungen wirken sich auf beide TAGs aus. Generell lassen sich beliebig viele TAGs mit einer CSS-Regel einstellen.

2.2 Kaskadierung

CSS kennt Regeln, die bei Konflikten zwischen verschiedenen Stildefinitionen zur Anwendung kommen. Diese Regeln bezeichnet man zusammen als Kaskadierung. Kaskadierung ist immer dann interessant, wenn Sie generelle Regeln auf vielen Ihrer Seiten benutzen, auf bestimmten Seiten jedoch zusätzlich andere Regeln definieren wollen. Nehmen wir als Beispiel die Regel aus dem vorherigem Kapitel:

```
h1, h2 {color: blue; background: grey;}
```

Möchten wir, dass diese Regel für alle Seiten gültig ist, jedoch in einer einzigen Seite Änderungen an dem Regelwerk gelten sollen, reicht es aus diese eine Regel mit einer anderen zu Überschreiben.

```
h1, h2 {color: white; background: black;}
```

Alle Regeln haben weiterhin ihre Gültigkeit, nur die Überschriften <h1> und <h2> werden von nun an in diesem Dokument weiß auf schwarz dargestellt.

2.3 Elemente

Elemente sind das Fundament der CSS-gesteuerten Darstellung. Generell lassen sich zwei Typen von Elementen unterscheiden:

- Ersetzte Elemente – Ersetzte Elemente sind alle die Elemente ohne eigenen Inhalt. Dies ist z.B. das Element <img>. Hier wird das eigentliche Element komplett durch ein Bild ersetzt. Diese Element stellt nur etwas dar, wenn im Attribut src an einen externen Inhalt verweist.

- Nicht-ersetzte Elemente – Nicht ersetzte Elemente machen den Großteil der in HTML definierten Elemente aus. Der von den Elementen erzeugte Inhalt wird in der Regel direkt durch den Browser dargestellt.

2.4 Darstellungsrollen der Elemente

Zusätzlich zu den ersetzten und nicht-ersetzten Elementen verwendet CSS zwei weitere grundsätzliche Elementtypen:

- Block-Elemente – Block-Elemente erzeugen standardmäßig eine Box, die den Inhaltsbereich des Elternelements ausfüllt und dabei nicht von anderen Elementen umschlossen sein kann. Ein Blockelement erzeugt immer einen Zeilenumbruch über und unterhalb der Box. Die gängigsten Block-Elemente sind <p> und .

- Inline-Elemente – Inline Elemente erzeugen eine Box innerhalb einer Textzeile, deren Fluss nicht unterbrochen wird. Ein gutes Beispiel für ein Inline-Element ist das <a>-TAG.

Generell können in CSS Darstellungsrollen von Block-Elementen und Inline-Elementen beliebig ineinander Verschachtelt werden, was in HTML nicht erlaubt ist.

Hierzu wurde in CSS folgende Eigenschaft definiert:

```
display                                          ! neue CSS-Eigenschaft
Werte:          none | inline | block | inline-block | list-
                item | run-in | table | inline-table |
                table-row-group | table-header-group |
                table-footer-group | table-row | table-
                column-group | table-column | table-cell |
                table-caption | inherit
Anfangswert:    inline
Anwendung auf: alle Elemente
Vererbt:        nein
```

Betrachten wir nun folgenden Code:

```
<body>
<p>Linux ist ein freies und <em>plattformunabhängiges
Mehrbenutzer-Betriebssystem</em> für Computer, das Unix ähnlich
ist.</p>
</body>
```

Sowohl <body> als auch <p> sind Block-Elemente, ist ein Inline-Element. Nach der HTML-Spezifikation ist diese Verschachtelung erlaubt, denn darf in <p> enthalten sein, umgekehrt wäre dies jedoch nicht erlaubt.

CSS hat dagegen keinerlei Einschränkungen dieser Art. Sie können den Code unverändert lassen, aber die Darstellungsrolle der zwei Elemente auf folgende Weise ändern:

```
p {display: inline;}
em{display: block;}
```

Jetzt würde innerhalb eines Inline-Elementes ein Block-Element erzeugt werden. Dieses Vorgehen ist erlaubt. Weiterhin nicht erlaubt ist es jedoch die Verschachtelung umzukehren:

```
<body>
<em>Linux ist ein freies und <p>plattformunabhängiges
Mehrbenutzer-Betriebssystem</p> für Computer, das Unix ähnlich
ist.</em>
</body>
```

2.5 CSS und HTML

Nun wollen wir uns die Möglichkeiten anschauen, wie CSS-Regeln in HTML-Dokumente eingebunden werden können.

Das <link>-TAG

Das <link>-TAG ermöglicht es laut HTML-Spezifikation Dokumente in andere Dokumente einzubinden. CSS nutzt dieses TAG um Stylesheets mit dem Dokument zu verbinden.

```
<link rel=”stylesheet” type=”text/css” href=”sheet.css”
media=”all” />
         rel   Beziehung oder Verknüpfung. Hier bezieht sich
               der Link auf ein Stylesheet
         type  Beschreibung der Daten, die über das TAG geladen
               werden sollen (hier immer text/css)
         href  URL zum Speicherort des Stylesheets
               (z.B. http://www.meineseite.de/mein.css)
         media Definition des Präsentationsmediums. Folgende
               Möglichkeiten stehen zur Auswahl:
               all         alle Medien (Standard)
               aural       auf Sprachsynthesizern
               braille     auf Braille-Geräten
               embossed    auf Braille-Druckern
               handheld    auf PDAs oder Mobiltelefonen
               print       zum Druck / zur Druckvorschau
               projection  auf Projektoren
               screen      auf Bildschirmen
               tty         auf Nichtproportionalschrift-
                           Umgebung
               tv          auf TV-Gerät
```

Stylesheets die sich nicht im HTML-Dokument befinden, sondern von außen eingebunden werden, nennt man externe Stylesheets. Wichtig ist es das <link>-TAG innerhalb des <head>-TAGs des Dokumentes zu definieren.

Der Aufbau des externen Stylesheets ist sehr einfach. Generell enthält es nur eine Liste von Regeln. Beachten Sie, dass keinerlei HTML-Code im Stylesheet eingetragen werden

darf.

Es ist ohne Weiteres möglich mehrere Stylesheets nacheinander zu laden. Der Browser wird in diesem Fall die Kombination der beiden Dokumente anwenden, soweit dies möglich ist.

```
<link rel="stylesheet" type="text/css" href="sheet1.css"
media="all" />
<link rel="stylesheet" type="text/css" href="sheet2.css"
media="all" />
```

Das <style>-Element

Das Element <style> ist ein neues HTML-Element. Es erlaubt das direkte Einbetten von Stylesheets in das Dokument und ist deshalb sehr beliebt bei Webentwicklern.

```
<style type="text/css">
h1 { color: blue; font-size: 24pt; }
</style>
```

<style> sollte immer zusammen mit dem Attribut type verwendet werden. Im Falle von CSS ist "text/css" immer der richtige Wert, genau wie bei dem <link>-Element.

Es ist möglich ein media-Attribut anzugeben um die weiter oben beschriebenen Funktionalität zu gewährleisten.

Diese Form der Einbindung von CSS-Anweisungen sollte nur sehr sparsam und gezielt einsetzt werden. Styles können die Pflege von HTML-Dokumenten erheblich erschweren. Das Verwenden des <style>-Tags kann u.U. sinnvoll sein, wenn die CSS-Eigenschaften aus einer Datenbank kommen sollen.

inline Styles

Schließlich soll als dritte Möglichkeit der Einbindung von CSS-Anweisungen noch kurz auf inline Styles eingegangen werden.

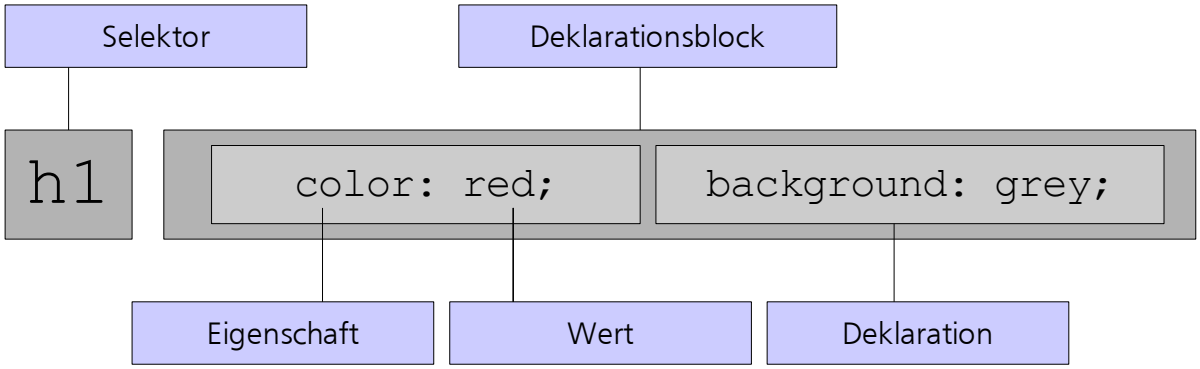
Auch diese Stildefinitionen werden in unmittelbarem Kontext eines HTML-Elements angegeben. Möchten Sie beispielsweise erreichen, dass eine bestimmte Überschrift ersten Grades die Schriftgröße `30pt` erhält und rot erscheint, benötigen Sie das `style`-Attribut, das wie folgt Verwendung findet:

```
<h1 style="font-size: 30pt; color: red;">
```

Sie sollten diese Form der Einbindung von CSS-Anweisungen nur sehr sparsam und gezielt einsetzen. Inline Styles können die Pflege von HTML-Dokumenten erheblich erschweren. Benutzen Sie sie also nur, wenn tatsächlich ein einzelnes Element abweichend formatiert werden soll. Des Weiteren wird das Attribut von der Spezifikation von XHTML 1.1 als veraltet angesehen.

2.6 Struktur

Jede Regel besteht generell aus zwei wesentlichen Teilen, dem Selektor und dem Deklarationsblock. Der Deklarationsblock besteht seinerseits aus mehreren Deklarationen. Jede Deklaration ist unterteilt in eine Eigenschaft und einer den dazugehörigen Wert. Jedes Stylesheet besteht aus einer Reihe von Regeln.



2.7 Spezifität - Gewichtung der Selektoren

In den Fällen, in denen CSS-Blöcke mehrfach vorkommen, muss entschieden werden, welcher Block für das darzustellende Element zuständig ist. Folgendes Beispiel zeigt diese Situation:

```
body h1 { color: blue; }
h1 { color: red; }
```

In diesem Fall würden wir für alle <h1>-Elemente erwarten, dass diese rot dargestellt werden. Richtig ist in diesem Fall, dass ausschließlich eine Regel treffen kann. Es ist also erforderlich, das der Browser eine Auswahl treffen muss, um das richtige Element auszuwählen. Dies geschieht anhand der Spezifität der CSS-Eigenschaften.

Im obigen Beispiel würde der HTML-Code

```
<h1>Überschrift</h1>
```

folgende Ausgabe erzeugen:

Überschrift

Schauen wir uns an warum dies so ist. In CSS sind folgende Regeln für die Spezifität von Elementen definiert:

Selektor	A	B	C	D
`style="..."`	1	0	0	0
`a`	0	0	0	1
`a.extern`	0	0	1	0
`a h1`	0	0	0	2
`#action`	0	1	0	0
`a.extern b`	0	0	1	1
`a.extern b h1`	0	0	1	2
`a:first-line`	0	0	1	1

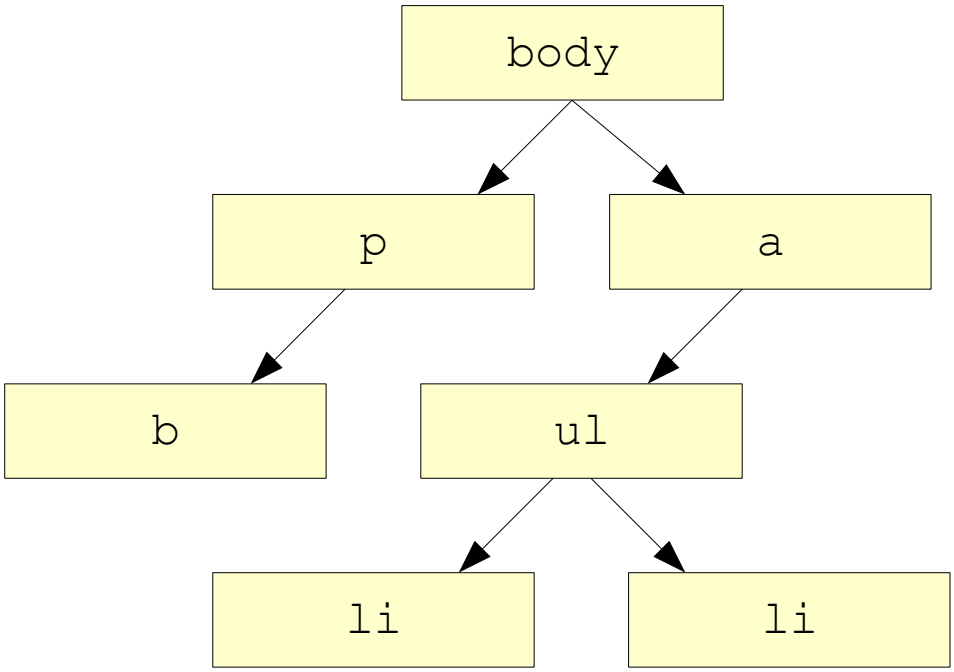
- **A** enthält den Wert **1**, wenn CSS-Deklarationen dem Element direkt per style-Element zugewiesen wurde (höchste Wertigkeit)
- **B** entspricht der Anzahl der selektierten ID-Attribute
- **C** entspricht der Anzahl der selektierten anderen Attribute und Pseudoklassen
- **D** entspricht der Anzahl der selektierten Elementnamen und Pseudoklassen (niedrigste Wertigkeit)

Spezialfällen in obigem Beispiel:

```
body h1 { color: blue; }          // 0 0 0 2  (Gewonnen!)
h1 { color: red; }                // 0 0 0 1
```

2.8 Vererbung

Jedes HTML-Dokument ist in einer Baumstruktur aufgebaut. Jedes Element (mit Ausnahme von body/html) hat ein Vaterelement. In CSS ist nun Definiert, dass jedes Vaterelement allen seinen Söhnen einige CSS-Eigenschaften vererbt.



Zu beachten ist, dass die folgenden Eigenschaften border, margin und padding nicht vererbt werden.

Beispiel:

CSS-Eigenschaften:

```
body { background: yellow; }
p { color: red; }
b { font: italic; }
```

HTML-Code:

```
<body>
   <p>
         Die Bezeichnung <b>Linux</b> wurde von Torvalds
         anfänglich nur für den von ihm geschriebenen
         <b>Kernel</b> genutzt.
   </p>
</body>
```

Ergebnis:

Die Bezeichnung **Linux** wurde von Torvalds anfänglich nur für den von ihm geschriebenen **Kernel** genutzt.

Die Hintergrundfarbe wurde von allen weiteren Elementen geerbt. Das <b>-Tag hat die rote-Schriftfarbe vom <p>-Tag geerbt.

2.9 Deklarationen gruppieren

Im Folgenden sollen einige Möglichkeiten der Gruppierung von Stylesheet-Regeln gezeigt werden.

Möglichkeit 1:

```
h1 { color: blue;}
h1 { font-size: 24pt; }
h1 { background: aqua; }
```

Möglichkeit 2:

```
h1 { color: blue; font-size: 24pt; background: aqua; }
```

Möglichkeit 3:

```
h1 {
   color: blue;
   font-size: 24pt;
   background: aqua;
}
```

2.10 Klassenselektoren

Die häufigste Methode Styles zu verwenden, ohne sich dabei um die beteiligten Elemente kümmern zu müssen, besteht darin, Klassendeklarationen zu benutzen. Wichtig ist es im Vorhinein festzulegen, welche Klassendeklarationen benutzt werden sollen, da ansonsten im gesamten Dokument Änderungen vorgenommen werden müssten. Folgendes Beispiel zeigt das Benutzen von Klassendeklarationen:

```
<p class="alert">Ein Virus wurde gefunden!</p>
```

Der Wert des Attributs class muss mit dem im Stylesheet vergebenen Klassenselektor übereinstimmen. Im obigen Beispiel könnte das Stylesheet folgerndermaßen aussehen:

```
p.alert {color: red; font-weight: bold;};
```

Besonders praktisch sind allgemeine Klassenselektoren die folgendermaßen ausschauen:

```
.alert {color: red;}
p.alert {font-weight: bold;}
```

Von nun an werden alle Fehlermeldungen auf der Seite in rot dargestellt. Sobald nun in einem <p>-TAG die alert-Klasse angegeben wird, wird die Fehlermeldung nicht nur rot, sondern zusätzlich auch noch fett angezeigt.

2.11 Mehrfache Klassen

In HTML ist es erlaubt einem einzelnen Element mehrere Klassen zuzuordnen. Diese Klassen müssen mit Space getrennt angegeben werden. Praktisch ist dies beispielsweise, wenn Sie ein Element als *wichtig* und als *Warnung* darstellen wollen:

```
<p class="urgent warning">Ihr System läuft instabil!</p>

<p class="urgent">Der Kaffee ist fertig!</p>
```

Die im Stylesheet definierten Regeln könnten dann so aussehen:

```
.urgent {font-weight: bold;}
.warning {font-style: italic;}
```

Dieses Beispiel hätte folgendes Ergebnis zur Folge:

Ihr System läuft instabil!

Der Kaffee ist fertig!

2.12 ID-Selektoren

Wie zuvor gezeigt, können Sie Klassen einer beliebigen Anzahl von Elementen zuweisen. So haben Sie den Klassennamen sowohl einem <p>-TAG als auch einem -TAG zugeordnet. IDs werden jedoch nur genau einmal innerhalb eines HTML-Dokuments vergeben. Wenn also ein Element bereits eine bestimmte ID zugewiesen bekommen hat kann kein anderes zusätzlich diese ID bekommen.

Auch Kombinationen von IDs sind nicht erlaubt. War es bei Klassen erlaubt einem Element mehrere, durch Kommata getrennte Werte zu übergeben, so ist dies bei IDs nicht erlaubt.

Nützlich kann die Verwendung von IDs sein, wenn klar ist das eine bestimmte Passage nur genau einmal pro Dokument vorkommt, jedoch nicht klar ist in welchem Element diese Passage vorkommt (z.B. <p>, ,...). Die Deklaration würde im Stylesheet folgendermaßen aussehen:

```
#wichtig {color: red; background: yellow;}
```

Im HTML-Dokument wäre diese Regel dann Folgendermaßen nutzbar:

```
<h1 id="wichtig">Wichtige Ankündigung!</h1>
<em id="wichtig">Wichtige Ankündigung!</em>
<p id="wichtig">Wichtige Ankündigung!</p>
```

Natürlich darf nur eines dieser Elemente in einem Dokument genutzt werden, sofern sie die gleiche ID besitzen.

Vorteile von IDs:

- Sie kann als Sprungziel für Hyperlinks verwendet werden.
- IDs können mit Javascript angesprochen werden. Das funktioniert über getElementById()
- IDs haben in CSS ein höhere Spezifität als Klassen.
- Quellcodes können genauer strukturiert werden, als mit den mehrfach vorkommenden Klassen.

Nachteile von IDs:

- Eine ID darf auf einer Seite nur einmal verwendet werden.
- IDs können nicht wie Klassen kombiniert werden.

2.13 Werte und Einheiten

2.13.1 Zahlen

In CSS gibt es zwei Arten von Zahlen, Integerwerte (ohne Nachkommastellen), und reelle Zahlen (mit Nachkommastellen). Die Nachkommastellen werden mit einem Dezimalpunkt von dem ganzzahligen Anteil getrennt. Manche Eigenschaften schränken den Wertebereich den die Zahlen einnehmen dürfen ein, generell gilt jedoch das sowohl negative als auch positive Werte erlaubt sind.

2.13.2 Farben

Man unterscheidet zwischen benannten Farben und Farbangaben anhand von RGB-Werten. Es stehen genau 17 benannte Grundfarben zur Verfügung:

aqua (Wasser)	green (Grün)	orange (Orange)	yellow (Gelb)
black (schwarz)	lime(Limone)	purple (Lila)	white (Weiß)
blue (Blau)	maroon (Kastanie)	red (Rot)	fuchsia (Fuchsie)
navy (Marine)	silver (Silber)	gray (Grau)	olive (Oliv)
teal (Krickentengrün)			

Glücklicherweise gibt es wesentlich bessere Möglichkeiten Farben in CSS zu definieren. Es ist möglich beliebige Farben aus dem 8Bit Farbraum auszuwählen. Folgende Arten der Deklaration stehen zur Verfügung:

```
p.eins {color: rgb(100%,100%,100%);}
p.zwei {color: rgb(0%,0%,0%);}
p.drei {color: rgb(255,255,255);}
p.vier {color: rgb(0,0,0);}
p.fuenf {color: #FF0000;}
```

2.14 Längeneinheiten

Man unterscheidet generell zwischen zwei Arten von Längenangaben in CSS:

- Absolute Längeneinheiten – Absolute Längeneinheiten hängen stark mit der Auflösung und der Größe des Monitors des Betrachters zusammen. Da diese Werte vom Autor einer Seite nicht beeinflussbar sind und diese Art der Längendefinition häufig zu Darstellungsfehlern führt ist es ratsam diese Art der Längendefinition zu umgehen.

  Mögliche Werte sind:

Inch (Zoll)(in)	
Zentimeter (cm)	Ein Inch entspricht 2,54cm
Millimeter (mm)	
Punkte (pt)	Ein Inch hat typischerweise 72pt
Pica (pc)	Ein Inch ist typischerweise 6pc

- Relative Längeneinheiten – Relative Längeneinheiten werden im Verhältnis zu anderen Dingen berechnet. Die folgenden Maßeinheiten stehen in CSS zur Verfügung:

em, ex	Wert von font-size für einen gegebenen Zeichensatz. Ist der Wert von font-size 14px, so entspricht die em=1. ex bezieht sich auf die Größe eines kleingeschriebenen Buchstabens ”x” eines Zeichensatzes.
px	Größenangabe in Pixeln

2.15 Zeichensätze

Die folgende Regel veranlasst den Browser eine angegebene Schriftart oder eine angegebene Schriftfamilie auszuwählen. Es können mehrere Angaben durch Kommata getrennt werden, wobei der erste Wert die höchste Priorität hat.

font-family	! neue CSS-Eigenschaft
Werte	Schriftartenname, [serif | sans-serif | cursive | fantasy | monospace] | inherit
Standardwert	abhängig vom Benutzerprogramm

Mit Hilfe folgender CSS-Eigenschaft läßt sich die Wichtung der Schrift angeben.

font-weight	! neue CSS-Eigenschaft
Werte	normal | bold | bolder | lighter | 100 | 200 | 300 | 400(=normal) | 500 | 600 | 700(=bold) | 800 | 900 | inherit
Standardwert	normal

Die folgende CSS-Eigenschaft dient der Festlegung der Schriftgröße. Entweder Sie geben die Werte absolut an (xx-small, x-small, ...), oder wie in Kapitel *Längeneinheiten* beschrieben.

font-size	! neue CSS-Eigenschaft
Werte	xx-small | x-small | small | medium | large | x-large | xx-large | larger | smaller | <Prozentwert> | <Höhe>
Standardwert	medium

Die folgende Eigenschaft ermöglicht kursive oder geneigte Schrift.

font-style	! neue CSS-Eigenschaft
Werte	italic | oblique | normal | inherit
Standardwert	normal

Die folgende Eigenschaft ermöglicht das Setzen der Schriftvariante.

font-variant	! neue CSS-Eigenschaft
Werte	small-caps | normal | inherit
Standardwert	normal

Beispiele:

```
h1 {
   font-family: serif;
}
```

font-family: serif

```
h1 {
   font-weight: bold;
}
```

font-weight: bold

```
h1 {
   font-size: 12px;
}
```

font-size: 12px

```
h1 {
   font-style: italic;
}
```

font-style: italic

2.16 Texteigenschaften

Einrückung der ersten Zeile

text-indent	! neue CSS-Eigenschaft
Werte	<Entfernung> | <Prozentwert> | inherit
Standardwert	0

Horizontale Ausrichtung

text-align	! neue CSS-Eigenschaft
Werte	left | center | right | justify | inherit
Standardwert	Abhängig vom Benutzerprogramm

Zeilenhöhe

line-height	! neue CSS-Eigenschaft
Werte	<Höhe> | <Prozentwert> | <Zahl> | normal | inherit
Standardwert	normal

Vertikale Ausrichtung

vertical-align	! neue CSS-Eigenschaft
Werte	baseline | sub | super | top | text-top | middle | bottom | text-bottom | <Höhe> | <Prozentwert> | inherit
Standardwert	baseline
Anwendung auf	Inline-Elemente und Tabellenzellen

Wortabstände

```
word-spacing                                    ! neue CSS-Eigenschaft
Werte           <Breite> | normal | inherit
Standardwert    normal
```

Buchstabenabstände

```
letter-spacing                                  ! neue CSS-Eigenschaft
Werte           <Breite> | normal | inherit
Standardwert    normal
```

Text-Umwandlungen

```
text-transform                                  ! neue CSS-Eigenschaft
Werte           uppercase | lowercase | capitalize | none |
                inherit
Standardwert    none
```

Text-Ausschmückung

```
text-decoration                                 ! neue CSS-Eigenschaft
Werte            none | underline | overline | line-through
                 | blink | inherit
Standardwert     none
```

Beispiele:

```
p {
   text-intent: 20px;
}
```

Linux (oder auch GNU/Linux, siehe auch GNU/Linux Namensstreit) ist ein freies und plattformunabhängiges Mehrbenutzer-Betriebssystem für Computer, das Unix ähnlich ist. Die Entwicklung von Linux begann 1991 durch die Veröffentlichung des ersten Linux-Kernels durch Linus Torvalds.

```
p {
   text-align: justify;
}
```

Linux (oder auch GNU/Linux, siehe auch GNU/Linux Namensstreit) ist ein freies und plattformunabhängiges Mehrbenutzer-Betriebssystem für Computer, das Unix ähnlich ist. Die Entwicklung von Linux begann 1991 durch die Veröffentlichung des ersten Linux-Kernels durch Linus Torvalds.

```
p {
   line-height: 20px;
}
```

Linux (oder auch GNU/Linux, siehe auch GNU/Linux Namensstreit) ist ein freies und plattformunabhängiges Mehrbenutzer-Betriebssystem für Computer, das Unix ähnlich ist. Die Entwicklung von Linux begann 1991 durch die Veröffentlichung des ersten Linux-Kernels durch Linus Torvalds.

```
p {
   letter-spacing: 5px;
}
```

Linux (oder auch GNU/Linux, siehe auch GNU/Linux Namensstreit) ist ein freies und plattformunabhängiges Mehrbenutzer-Betriebssystem.

```
p {
   text-transform: uppercase;
}
```

LINUX (ODER AUCH GNU/LINUX, SIEHE AUCH GNU/LINUX NAMENSSTREIT) IST EIN FREIES UND PLATTFORMUNABHÄNGIGES MEHRBENUTZER-BETRIEBSSYSTEM.

Textschatten

Schatten werden angegeben indem man eine Farbe definiert und mit den ersten beiden Parametern den Abstand zum Text angibt. Der dritte Parameter ist optional und kann einen Weichzeichnenradius angeben.

text-shadow	! neue CSS-Eigenschaft
Werte	none | <Farbe> <Länge> <Länge> [<Länge>] | inherit
Standardwert	none

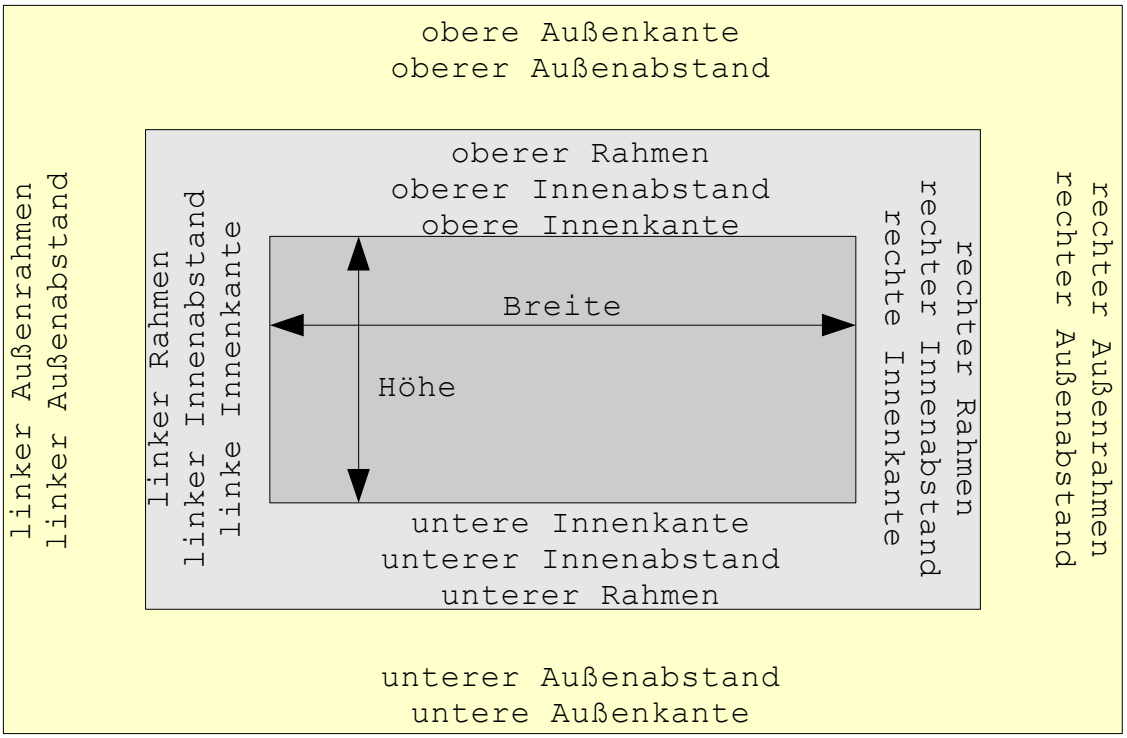
Leerzeichen umgehen

Im Normalfall reduziert ein Browser hintereinander gestellte Leerzeichen und Umbrüche auf ein/einen einzelnes/n. Dieses Verhalten kann mittels der folgenden Eigenschaft genau Definiert werden.

white-space	! neue CSS-Eigenschaft
Werte	normal | nowrap | pre | pre-wrap | pre-line | inherit
Standardwert	normal

2.17 Innenabstände, Rahmen und Außenabstände

Das CSS-Box-Modell



Die Höhe ist der Abstand zwischen der unteren und der oberen Innenkante

height	! neue CSS-Eigenschaft
Werte	<Höhe> | <Prozentwert> | auto | inherit
Standardwert	auto

Die Breite ist der Abstand zwischen der linken und der rechten Innenkante

width	! neue CSS-Eigenschaft
Werte	<Breite> | <Prozentwert> | auto | inherit
Standardwert	auto

Definition des Außenabstands

margin	! neue CSS-Eigenschaft
Werte	<Breite> | <Prozentwert> | auto | inherit
Standardwert	0

Definition der einzelnen Außenabstände

margin-top	! neue CSS-Eigenschaft
Werte	<Breite> | <Prozentwert> | auto | inherit
Standardwert	0
margin-right	! neue CSS-Eigenschaft
Werte	<Breite> | <Prozentwert> | auto | inherit
Standardwert	0
margin-left	! neue CSS-Eigenschaft
Werte	<Breite> | <Prozentwert> | auto | inherit
Standardwert	0
margin-bottom	! neue CSS-Eigenschaft
Werte	<Breite> | <Prozentwert> | auto | inherit
Standardwert	0

Rahmen

border-style	! neue CSS-Eigenschaft
Werte	none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset | inherit
Standardwert	nicht definiert

Stile für einzelne Rahmenseiten

border-top-style	! neue CSS-Eigenschaft
Werte	none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset | inherit
Standardwert	nicht definiert
border-right-style	**! neue CSS-Eigenschaft**
Werte	none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset | inherit
Standardwert	nicht definiert
border-left-style	**! neue CSS-Eigenschaft**
Werte	none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset | inherit
Standardwert	nicht definiert
border-bottom-style	**! neue CSS-Eigenschaft**
Werte	none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset | inherit
Standardwert	nicht definiert

Rahmenbreiten

border-width	! neue CSS-Eigenschaft
Werte	thin | medium | thick | <Breite> | inherit
Standardwert	nicht definiert

Stile für einzelne Rahmenseiten

border-top-width	! neue CSS-Eigenschaft
Werte	thin | medium | thick | <Breite> | inherit
Standardwert	nicht definiert
border-right-width	**! neue CSS-Eigenschaft**
Werte	thin | medium | thick | <Breite> | inherit
Standardwert	nicht definiert
border-left-width	**! neue CSS-Eigenschaft**
Werte	thin | medium | thick | <Breite> | inherit
Standardwert	nicht definiert
border-bottom-width	**! neue CSS-Eigenschaft**
Werte	thin | medium | thick | <Breite> | inherit
Standardwert	nicht definiert

Rahmenfarben

border-color	! neue CSS-Eigenschaft
Werte	<Farbwert> | transparent | inherit
Standardwert	nicht definiert

Stile für einzelne Rahmenfarben

border-top-color	! neue CSS-Eigenschaft
Werte	<Farbwert> | transparent | inherit
Standardwert	nicht definiert
border-right-color	**! neue CSS-Eigenschaft**
Werte	<Farbwert> | transparent | inherit
Standardwert	nicht definiert
border-left-color	**! neue CSS-Eigenschaft**
Werte	<Farbwert> | transparent | inherit
Standardwert	nicht definiert
border-bottom-color	**! neue CSS-Eigenschaft**
Werte	<Farbwert> | transparent | inherit
Standardwert	nicht definiert

Innenabstände

Zwischen dem Rahmen und der Inhaltsbox befinden sich die sogenannten Innenabstände der Elementbox.

padding	! neue CSS-Eigenschaft
Werte	<Breite> | <Prozentwert> | inherit
Standardwert	nicht definiert

Einseitige Innenabstände

padding-top	! neue CSS-Eigenschaft
Werte	<Breite> | <Prozentwert> | inherit
Standardwert	nicht definiert

padding-right	! neue CSS-Eigenschaft
Werte	<Breite> | <Prozentwert> | inherit
Standardwert	nicht definiert

padding-left	! neue CSS-Eigenschaft
Werte	<Breite> | <Prozentwert> | inherit
Standardwert	nicht definiert

padding-bottom	! neue CSS-Eigenschaft
Werte	<Breite> | <Prozentwert> | inherit
Standardwert	nicht definiert

Beispiel:

```
p {
   border-width: 1px;
   padding-top: 10px;
   padding-bottom: 10px
   padding-right: 30px
   padding-left: 30px
}
```

Linux (oder auch GNU/Linux, siehe auch GNU/Linux Namensstreit) ist ein freies und plattformunabhängiges Mehrbenutzer-Betriebssystem.

```
p {
   border-width: 1px;
   border-style: double;
   border-color: green;
}
```

Linux (oder auch GNU/Linux, siehe auch GNU/Linux Namensstreit) ist ein freies und plattformunabhängiges Mehrbenutzer-Betriebssystem.

```
p {
   border-width: 3px;
   border-style: solid;
   margin-left: 10px;
   margin-right: 10px;
   border-color: blue;
}
```

Linux (oder auch GNU/Linux, siehe auch GNU/Linux Namensstreit) ist ein freies und plattformunabhängiges Mehrbenutzer-Betriebssystem.

2.18 Farben und Hintergründe

2.18.1 Vordergundfarben

Die folgende Eigenschaft ändert primär die Farbe des Vordergrundes. Dies ist im Normalfall die Schrift. Rahmen gehören streng genommen zum Vordergrund, weshalb diese mit dieser Eigenschaft auch geändert wird. Möchte man dies nicht, muss man die Rahmenfarbe explizit setzen.

```
color                                                    ! neue CSS-Eigenschaft
Werte           <Farbwert> | inherit
Standardwert    abhängig von Benutzerprogramm
```

```
p {
   color: #FF0000;
}
```

Linux (oder auch GNU/Linux, siehe auch GNU/Linux Namensstreit) ist ein freies und plattformunabhängiges Mehrbenutzer-Betriebssystem.

2.18.2 Hintergründe

Hintergrundfarben

```
backgound-color                                    ! neue CSS-Eigenschaft
Werte              <Farbwert> | transparent | inherit
Standardwert       transparent
```

Hintergrundbilder

```
background-image                                   ! neue CSS-Eigenschaft
Werte              <URL> | none | inherit
Standardwert       none
```

Wiederholungen in eine bestimmte Richtung

```
background-repeat                                  ! neue CSS-Eigenschaft
Werte               repeat | repeat-x | repeat-y | no-repeat
                    | inherit
Standardwert        repeat
```

Positionierung von Hintergrundbildern

```
background-position                                ! neue CSS-Eigenschaft
Werte                 [<Prozentwert> | <Entfernungsangabe> |
                      left | center | right] |
                      [<Prozentwert> | <Entfernungsangabe> |
                      top | center | bottom] | inherit
Standardwert          0% 0%
```

Hintergrundbilder verankern

background-attachment	! neue CSS-Eigenschaft
Werte	scroll | fixed | inherit
Standardwert	scroll

Beispiele:

```
p {
   background-color: silver;
}
```

Linux (oder auch GNU/Linux, siehe auch GNU/Linux Namensstreit) ist ein freies und plattformunabhängiges Mehrbenutzer-Betriebssystem.

```
p {
   background-image: url(back.png);
}
```

Linux (oder auch GNU/Linux, siehe auch GNU/Linux Namensstreit) ist ein freies und plattformunabhängiges Mehrbenutzer-Betriebssystem.

```
p {
   background-image: url(back.png);
   background-repeat: no-repeat;
}
```

Linux (oder auch GNU/Linux, siehe auch GNU/Linux Namensstreit) ist ein freies und plattformunabhängiges Mehrbenutzer-Betriebssystem.

2.19 Pseudoklassen

Die folgenden Pseudoklassen sind ausschließlich für Links definiert.

Name	! neue CSS-Eigenschaft
:link	Bezieht sich auf Hyperlinks die noch nicht besucht wurden
:visited	Bezieht sich auf Hyperlinks die bereits besucht wurden
:focus	Bezieht sich auf Elemente die gerade den Eingabefokus besitzen
:hover	Bezieht sich auf Elemente über denen sich gerade der Mauszeiger befindet
:active	Bezieht sich auf Elemente die gerade angeklickt werden (Maustaste ist gedrückt)

Beispiele:

```
a:link {
   color: green;
}
```

Bitte klicken Sie HIER

```
a:visited {
   color: red;
}
```

Bitte klicken Sie HIER

2.20 Selektoren für Pseudoelemente

Name	! neue CSS-Eigenschaft
:first-letter	Setzt die Eigenschaften für den ersten Buchstaben eines Block-Elements
:first-line	Setzt die Eigenschaften der ersten Zeile eines Elements

Beispiele:

```
p:first-letter {
   font-size: 200%;
}
```

Linux (oder auch GNU/Linux, siehe auch GNU/Linux Namensstreit) ist ein freies und plattformunabhängiges Mehrbenutzer-Betriebssystem.

```
a:first-line {
   color: red;
}
```

Linux (oder auch GNU/Linux, siehe auch GNU/Linux Namensstreit) ist ein freies und plattformunabhängiges Mehrbenutzer-Betriebssystem. Linux (oder auch GNU/Linux, siehe auch GNU/Linux Namensstreit) ist ein freies und plattformunabhängiges Mehrbenutzer-Betriebssystem.

2.21 Layouts mit

Betrachtet man das in HTML-Definierte Tag unabhängig von CSS, so wird man sich sicher fragen, wie man mit einem derart schwachen Tag komplexe Layouts realisieren soll. Das -Tag ist in HTML als Gruppierungselement definiert. Alle HTML-Elemente, welche sich zwischen und befinden sind Bestandteil dieser -Gruppe. Alle Einstellungen bzgl. der Ausrichtung der Elemente werden von den Gruppenelementen übernommen. Wesentlich mehr ist in HTML (ohne CSS) mit diesem Element nicht vorgesehen.

Mit CSS hat sich diese Situation allerdings grundlegend geändert. CSS bietet vielfältige Möglichkeiten der Manipulation von -Elementen. Einige Möglichkeiten werden im Folgenden vorgestellt.

Betrachten wir zunächst das folgende Beispiel:

```
<div>
   Robert Gernhardt<br>
   Ein Gleichnis
</div>
<div>
   Wie wenn da einer, und er hielte<br>
   ein frühgereiftes Kind, das schielte,<br>
   hoch in den Himmel und er bäte:<br>
   Du hörst jetzt auf den Namen Käthe! -<br>
   Wär dieser nicht dem Elch vergleichbar,<br>
   der tief im Sumpf und unerreichbar<br>
   nach Wurzeln, Halmen, Stauden sucht<br>
   und dabei stumm den Tag verflucht,<br>
   an dem er dieser Erde Licht ...<br>
   Nein? Nicht vergleichbar? Na, dann nicht!
</div>
```

In diesem Beispiel sind die -Tags nicht per CSS definiert worden. Dem entsprechend wird die Ausgabe im Browser folgendermaßen aussehen:

Robert Gernhardt
Ein Gleichnis
Wie wenn da einer, und er hielte
ein frühgereiftes Kind, das schielte,
hoch in den Himmel und er bäte:
Du hörst jetzt auf den Namen Käthe! -
Wär dieser nicht dem Elch vergleichbar,
der tief im Sumpf und unerreichbar
nach Wurzeln, Halmen, Stauden sucht
und dabei stumm den Tag verflucht,
an dem er dieser Erde Licht ...
Nein? Nicht vergleichbar? Na, dann nicht!

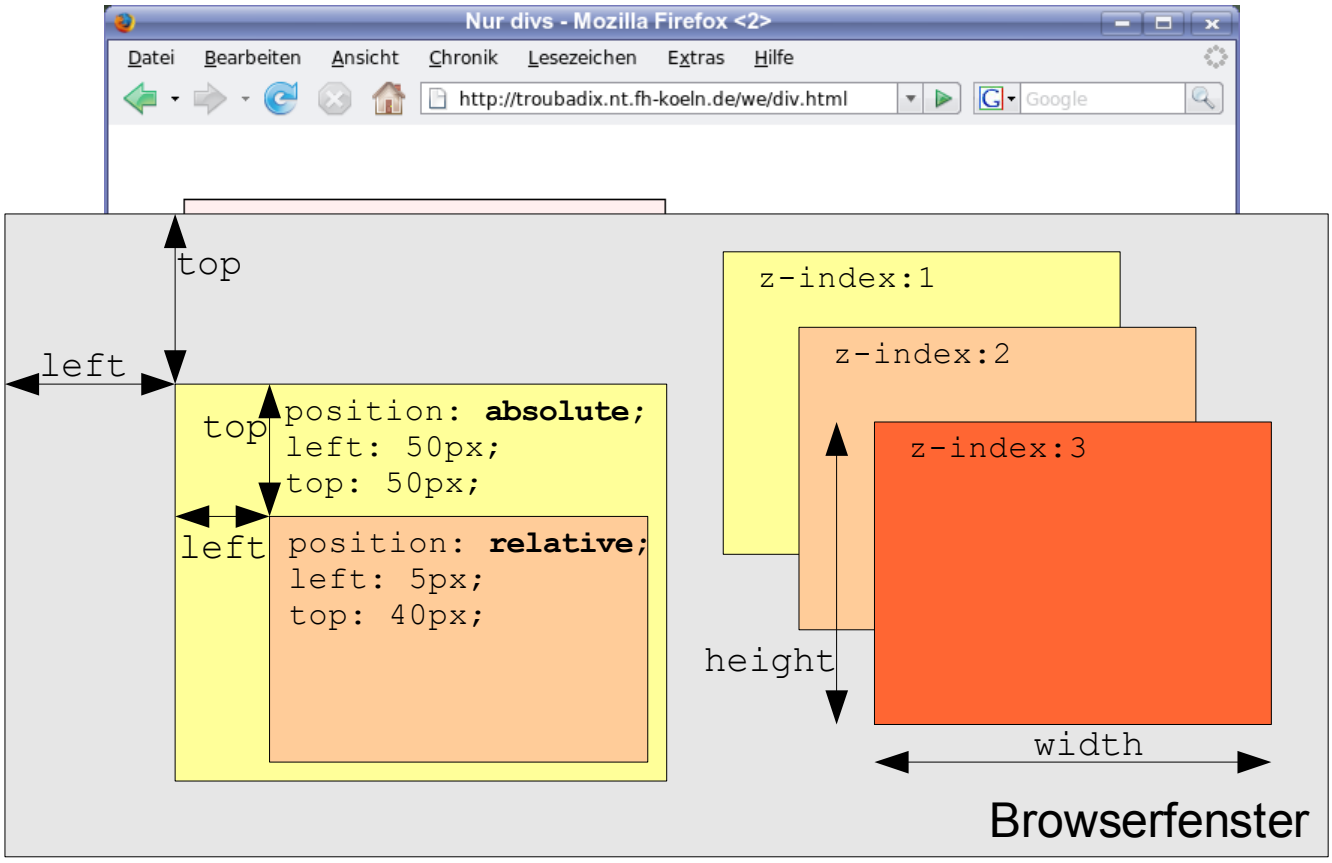
Nun wollen wir die CSS-Boxen per CSS beeinflussen. Dazu definieren wir die folgenden Regeln:

```
#head {
   position:absolute;
   left:50px;
   top:50px;
   width:300px;
   height:auto;
   border:1px solid #000000;
   background-color:#ffefef;
   padding: 10px;
}
#text {
   position:absolute;
   left:50px;
   top:120px;
   width:500px;
   height:auto;
   border:1px solid #000000;
   background-color:#efffef;
   padding: 3px;
}
```

Nun müssen noch die IDs in den HTML-Code integriert werden:

```
<div id="head">
   Robert Gernhardt<br>
   Ein Gleichnis
</div>
<div id="text">
   Wie wenn da einer, und er hielte<br>
   ...
   Nein? Nicht vergleichbar? Na, dann nicht!
</div>
```

Diese Änderungen wirken sich nun folgendermaßen im Browser aus:



Generell können Sie alle Formatierungen wie gewohnt auch an -Tags in CSS definieren. Es kommen jedoch für die Layoutgestalltung einige CSS-Eigenschaften hinzu, welche ausschließlich auf -Tags Verwendung finden.

Name	! neue CSS-Eigenschaft
top	Abstand von der oberen Begrenzung des Elternelements oder des oberen Browserrands
left	Abstand von der linken Begrenzung des Elternelements oder des linken Browserrands
position	top und left werden gemessen vom: - Browserrand: absolute - Elternelement: relative
z-index	„Überlappungsindex". Das Element mit dem höheren Index überdeckt das mit dem Kleineren.