



Werkzeuge der Informatik

Übung 2: Einführung in Cascading Stylesheets (CSS)*

Wintersemester 2008/ 2009

Institut für Informatik

05.12.2008

*) <http://de.selfhtml.org/>

Übersicht: Einführung in CSS

▪ **Aufgabenstellung**

- I: Stylesheets in HTML einbinden
 - Formate zentral in HTML-Datei definieren
 - Formate zentral in separater CSS Datei definieren
 - Formate innerhalb eines HTML-Elements definieren
- II: CSS-Eigenschaften
 - Beispiel 1: Zentrale Formate für HTML-Elemente definieren
 - Hintergrundfarbe
 - Schriftfarbe
 - Einschub: Farben
 - Schriftart
 - Schriftstil und Schriftgröße
 - Außenrand und -abstand
 - Einschub: Numerische Angaben
 - Rahmen
 - Beispiel 2: Individualformate definieren
 - Innenabstand
 - Positionierung und Anzeige von Elementen
 - Zusatz 1: Formate für verschachtelte HTML-Elemente definieren
 - Zusatz 2: Formate für Klassen definieren
- III: CSS-basierte Layouts
 - Beispiel 3: Zweispaltiges CSS-basiertes Layout
 - Pseudoklassen für Verweise
 - Textumfluss

Aufgabenstellung: Überblick

- In dieser Aufgabe erstellen Sie eine geeignete CSS-Definition für ein gegebenes HTML-Dokument (anschrift.html), welches in einer Website die erste Seite Ihres Lebenslaufs abbilden soll. Die Unterseiten (Schulausbildung, Studium,...) sind für diese Übung uninteressant, können aber von Ihnen zur Übung gerne erstellt werden. Orientieren Sie sich an dem Beispiel unten (verwenden Sie aber in Ihrer Abgabe Ihre eigenen Daten!):

The screenshot shows a Windows Internet Explorer browser window displaying a resume page. The browser's address bar shows the URL `http://winf.in.tu-clausthal.de/foalem/ws0809/anschrift.html`. The page title is "Mein Lebenslauf". The main content area is titled "Mein Lebenslauf" and contains the following information:

Dipl.-Wirt.-Inf. Olivier Foalem
Julius-Albert-Str. 4
38678 Clausthal-Zellerfeld
Telefon: 05323 - 72 71 55

To the right of the contact information is a portrait photo of a man. Below the photo is a vertical list of menu items: Anschrift und Foto, Schulausbildung, Studium, Sprachkenntnisse, and Private Interessen. At the bottom of the page, a footer indicates: "Dieser Lebenslauf wurde erstellt am: 17.11.2008".

Aufgabestellung: Details

- **Arbeiten Sie zunächst die Beispiele in diesem Skript durch, um die grundlegenden Prinzipien von CSS zu erlernen. Mit Hilfe der Beispiele können Sie Ideen zur Bearbeitung der Aufgabe sammeln.**
- Betrachten Sie erst **nach Durcharbeiten dieses Skripts** die HTML-Seite des Lebenslaufs (anschrift.html) genauer. Sie entdecken ein dreispaltiges Layout.
 - Links befindet sich eine Navigation.
 - Es gibt insgesamt fünf Navigationsflächen, die grau hinterlegt sind:
 - „Anschrift und Foto“,
 - „Schulbildung“,
 - „Studium“,
 - „Sprachkenntnisse“ und
 - „Private Interessen“.
 - Die Navigationsflächen verändern Ihre Hintergrundfarbe, wenn Sie Ihren Mauszeiger darüber bewegen und nehmen wieder die Ursprungsfarbe an, wenn Sie den Link mit der Maus verlassen.
 - Nach Klick auf einen Navigationspunkt wird dieser dauerhaft in einer anderen Farbe dargestellt.

Aufgabestellung: Details (2)

- In der Mitte ist ein Textfeld für welches die durch das Menü ausgewählte Unterseite anzeigt. Dort soll Ihr Name als Überschrift und Ihre Anschrift als Standard-Text dargestellt werden.
 - Rechts befindet sich Ihr persönliches Bild. Die Bildfläche ist grau hinterlegt.
 - Das dreispaltige Layout ist umgeben von einer Kopf- und einer Fußzeile. Beides sind ebenfalls farblich hinterlegt.
- Als Schriftart verwenden Sie bitte generell Helvetica, Arial oder eine Schriftart im Stil „sans-serif“
 - Ihr Stylesheet soll in jedem Fall mit Hilfe von „Individualformaten“ erstellt werden. In diesem Fall empfiehlt es sich drei ID's zu definieren: „Navigation“, „Inhalt“ und „Bild“ (vgl. Folie 29 ff.).
 - Das Stylesheet definieren Sie bitte zentral in der HTML-Datei (vgl. Folie 11).

Das zugehöriges HTML-Dokument (anschrift.html)

```
<body>

  <h1>Mein Lebenslauf</h1>

  <ul id="Navigation">
    <li><a href="anschrift.html">Anschrift und Foto</a></li>
    <li><a href="schule.html">Schulausbildung</a></li>
    <li><a href="studium.html">Studium</a></li>
    <li><a href="sprachen.html">Sprachkenntnisse</a></li>
    <li><a href="privat.html">Private Interessen </a></li>
  </ul>

  <div id="Bild">
    <p></p>
  </div>

  <div id="Inhalt">
    <h2>Dipl.-Wirt.-Inf. Olivier Foalem </h2>
    <p>Julius-Albert-Str. 4</p>
    <p>38678 Clausthal-Zellerfeld</p>
    <p>Telefon: 05323 - 72 7155</p>
  </div>

  <p id="Fusszeile">Dieser Lebenslauf wurde erstellt am: 17.11.2008</p>

</body>
```

Aufgabenstellung: Das Stylesheet erstellen (1)

- **Als erstes definieren Sie das Format für den „body“:**
 - Innenabstand allgemein: 1em
 - Aussenabstand allgemein: 0
 - Schriftfamilie: Helvetica,Arial,sans-serif
 - Schriftfarbe: schwarz
 - Hintergrundfarbe: weiss

- **Jetzt formatieren Sie die Überschrift „h1“ (Mein Lebenslauf) wie folgt:**
 - Schriftgrösse: 1.5 em
 - Aussenabstand nach oben, rechts und links: 0
 - Aussenabstand nach unten: 0.7em
 - Innenabstand allgemein: 0.3em
 - Textausrichtung: zentriert
 - Hintergrundfarbe: #fed
 - Rahmen: „2px ridge silver“

- **Nun formatieren Sie das erste Individualformat: „Navigation“**
 - **Sie beginnen mit der Aufzählungsliste „ul“ und definieren:**
 - Schriftgrösse: 0.83 em
 - Textumfluss: Element steht links
 - Breite: 18em
 - Aussenabstand nach oben, rechts und links: 0
 - Aussenabstand nach unten: 1.2em
 - Innenabstand allgemein: 0
 - Rahmen: „1px dashed silver“
 - **Für die Aufzählung „ul“ definieren Sie die Listenpunkte „li“:**
 - Listenformatierung: kein Bullet-Zeichen
 - Aussenabstand allgemein: 0
 - Innenabstand allgemein: 0.5em

Aufgabenstellung: Das Stylesheet erstellen (2)

- **Für die Aufzählung „ul“ definieren Sie die Links „a“:**
 - Innenabstand allgemein: 0.2em
 - Anzeige des Elements als (absatzerzeugenden) Block
 - Schriftgewicht: fett
 - **Für die Aufzählung „ul“ definieren Sie die Pseudoklasse „a:link“:**
 - Schriftfarbe: schwarz
 - Hintergrundfarbe: #eee
 - **Dann die Pseudoklasse „a:visited“:**
 - Schriftfarbe: #666
 - Hintergrundfarbe: #eee
 - **Danach die Pseudoklasse „a:hover“:**
 - Schriftfarbe: schwarz
 - Hintergrundfarbe: weiss
 - **Und zuletzt die Pseudoklasse „a:active“:**
 - Schriftfarbe: weiss
 - Hintergrundfarbe: grau
-
- **Jetzt folgt das Individualformat „Bild“**
 - **Definieren Sie für das Tag „div“ folgendes Format**
 - Schriftgröße: 0.9em
 - Textfluss: Element steht rechts
 - Breite: 12em
 - Aussenabstand nach oben, rechts und links: 0
 - Abstand nach unten: 1.1em
 - Innenabstand: 0
 - Hintergrundfarbe: #eee
 - Rahmen: „1px dashed silver“

Aufgabenstellung: Das Stylesheet erstellen (3)

- **Für einen Absatz „p“ im div-Bereich von „Bild“ gilt folgendes**
 - Abstand allgemein: 1.0em

- **Als vorletztes definieren Sie das Individualformat „Inhalt“**
 - **Es gilt im gesamten Bereich:**
 - Abstand oben: 0, Abstand rechts: 12em, Abstand links: 16em, Abstand unten: 1em
 - Innenabstand oben und unten: 0, Innenabstand rechts und links: 1em
 - Rahmen: „1px dashed silver“
 - **Für die Überschrift „h2“ im „Inhalt“-Bereich gilt speziell:**
 - Schriftgröße: 1.2em
 - Abstand oben und unten: 0.2em, Abstand rechts und links: 0
 - **Für den Absatz p“ im „Inhalt“-Bereich gilt:**
 - Schriftgröße: 1em
 - Abstand oben und unten: 1em, Abstand rechts und links: 0

- **Zuletzt definieren Sie das Individualformat „Fusszeile“**
 - Den Textfluss links und rechts unterbrechen und die Fortsetzung unterhalb erzwingen
 - Schriftgröße: 0.9em
 - Abstand allgemein: 0
 - Innenabstand allgemein: 0.1em
 - Textausrichtung: zentriert
 - Hintergrundfarbe: #fed
 - Rahmen: „1px solid silver“

Übersicht: Einführung in CSS

- Aufgabenstellung
- **I: Stylesheets in HTML einbinden**
 - **Formate zentral in HTML-Datei definieren**
 - **Formate zentral in separater CSS Datei definieren**
 - **Formate innerhalb eines HTML-Elements definieren**
- II: CSS-Eigenschaften
 - Beispiel 1: Zentrale Formate für HTML-Elemente definieren
 - Hintergrundfarbe
 - Schriftfarbe
 - Einschub: Farben
 - Schriftart
 - Schriftstil und Schriftgröße
 - Außenrand und Abstand
 - Einschub: Numerische Angaben
 - Rahmen
 - Beispiel 2: Individualformate definieren
 - Innenabstand
 - Positionierung und Anzeige von Elementen
 - Zusatz 1: Formate für verschachtelte HTML-Elemente definieren
 - Zusatz 2: Formate für Klassen definieren
- III: CSS-basierte Layouts
 - Beispiel 3: Zweispaltiges CSS-basiertes Layout
 - Pseudoklassen für Verweise
 - Textumfluss

Formate zentral in HTML-Datei definieren

- Sie können im head-Element einer HTML-Datei einen Bereich für CSS-Formate definieren.
- Mit `<style...> ... </style>` notieren Sie einen Bereich für Stylesheet-Formatdefinitionen (*style = Stil, Format*).
- Im einleitenden `<style>`-Tag müssen Sie den MIME-Typ der Stylesheet-Sprache angeben. Für CSS ist das die Angabe `type="text/css"`.
- Zwischen dem einleitenden Tag und dem abschließenden `</style>` können Sie dann zentrale CSS-Formate definieren.

```
<html>

<head>
<title>Seitentitel</title>
<style type="text/css">
  <!-- /* ... Hier werden die Formate definiert ... */ -->
</style>
</head>

<body>
</body>
</html>
```

Formate zentral in separater CSS Datei definieren

Im Dateikopf einer HTML-Datei können Sie mit `<link...>` eine CSS-Datei referenzieren, die CSS-Formatdefinitionen enthält (*link = Verweis*).

Innerhalb des `<link>`-Tags müssen die Angaben `rel="stylesheet" type="text/css"` stehen (*rel = relation = Bezug, type = MIME-Typ*).

Beim Attribut `href` geben Sie die gewünschte Datei an (*href = hyper reference = Hypertext-Referenz*).

Wenn sich die CSS-Datei in einem anderen Verzeichnis oder auf einem anderen Server befindet, müssen Sie an dieser Stelle Pfadangaben oder absolute URIs notieren.

```
<html>
<head>
  <title>Seitentitel</title>
  <link rel="stylesheet" type="text/css" href="„stylesheet.css">
</head>
<body>
</body>
</html>
```

Formate innerhalb eines HTML-Elements definieren

- Sie können Formate für ein HTML-Element definieren, indem Sie innerhalb des einleitenden HTML-Tags das Attribut `style` und dahinter die gewünschten Formatdefinitionen notieren.
- Innerhalb der Formatdefinitionen sind CSS-Eigenschaften erlaubt. Dabei gilt die gleiche Syntax wie innerhalb der geschweiften Klammern bei zentralen Formaten.
- Jede Eigenschaft besteht aus einem Namen, z.B. `color` (Schriftfarbe) und einem zugehörigen Wert, z.B. `#D8FD02`, getrennt durch einen Doppelpunkt.
- Schließen Sie jede Formatdefinition jeweils durch einen Strichpunkt ab. Nur bei der letzten Formatdefinition vor dem abschließenden Anführungszeichen darf der Strichpunkt entfallen.

```
<html>
<head>
  <title>Seitentitel</title>
  <style type="text/css"> body { background-color:#000000 } </style>
</head>
<body>
  <p style="background-color:#808040; color:#D8FD02;...">Hier steht Text.</p>
</body>
</html>
```

Übersicht: Einführung in CSS

- Aufgabenstellung
- I: Stylesheets in HTML einbinden
 - Formate zentral in HTML-Datei definieren
 - Formate zentral in separater CSS Datei definieren
 - Formate innerhalb eines HTML-Elements definieren
- **II: CSS-Eigenschaften**
 - **Beispiel 1: Zentrale Formate für HTML-Elemente definieren**
 - **Hintergrundfarbe**
 - **Schriftfarbe**
 - **Einschub: Farben**
 - **Schriftart**
 - **Schriftstil und Schriftgröße**
 - **Außenrand und Abstand**
 - **Einschub: Numerische Angaben**
 - **Rahmen**
 - **Beispiel 2: Individualformate definieren**
 - **Innenabstand**
 - **Positionierung und Anzeige von Elementen**
 - **Zusatz 1: Formate für verschachtelte HTML-Elemente definieren**
 - **Zusatz 2: Formate für Klassen definieren**
- III: CSS-basierte Layouts
 - Beispiel 3: Zweispaltiges CSS-basiertes Layout
 - Pseudoklassen für Verweise
 - Textumfluss

Beispiel 1: Zentrale Formate für HTML-Elemente definieren

- Um ein zentrales Format für alle HTML-Elemente eines bestimmten Typs zu definieren, notieren Sie als Selektor den Namen des HTML-Elementtyps, und zwar ohne spitze Klammern.
- Im nachfolgenden Beispiel (1) werden
 - body (Dokument),
 - h1 (Überschriften 1. Ordnung),
 - p (Textabsätze) und
 - li (Listeneinträge) auf diese Weise notiert.
- Über den Universalselektor * haben Sie die Möglichkeit, Eigenschaften für *alle* Elemente zu definieren, wobei jedem Element natürlich nur die jeweils zulässigen Eigenschaften zugewiesen werden.

Beispiel 1: Code und Ergebnis (1)

```
<html>
<head>
<title>Seitentitel</title>
<style type="text/css">
  body { background-color:#FFFFCC; margin-left:100px; }
  * { color:blue; }
  h1 { font-size:300%; color:#FF0000; font-style:italic; border-bottom:solid thin
black; }
  p,li { font-size:140%; font-family:Helvetica,Arial,sans-serif;
letter-spacing:0.3em; }
</style>
</head>
<body>
<h1>&U
<p>ein normaler Textabsatz</p>
<ul>
  <li>Ein Listenpunkt</li>
  <li>Ein anderer Listenpunkt</li>
</ul>
</body>
</html>
```

Dem body-Element wird eine hellgelbe Hintergrundfarbe (background-color:#FFFFCC;) und ein linker Randabstand von 100 Pixeln (margin-left:100px;) zugewiesen.

Überschrift 1. Ordnung

ein normaler Textabsatz

- Ein Listenpunkt
- Ein anderer Listenpunkt

[zurück](#)

Beispiel 1: Code und Ergebnis (2)

```
<html>
<head>
<title>Seitentitel</title>
<style type="text/css">
  body { background-color:#FFFFCC; margin-left:100px; }
  * { color:blue; }
  h1 { font-size:300%; color:#FF0000; font-style:italic; border-bottom:solid thin
black; }
  p,li { font-size:100%; line-height:140%; font-family:Helvetica,Arial,sans-serif;
letter-spacing:0.1em; }
</style>
</head>
<body>
<h1>&U
<p>ein normaler Textabsatz</p>
<ul>
  <li>Ein Listenelement</li>
  <li>Ein anderer Listenelement</li>
</ul>
</body>
</html>
```

Über den Universalselektor wird für alle Elemente zunächst blaue Schriftfarbe (color:blue;) definiert.

Überschrift 1. Ordnung

ein normaler Textabsatz

- Ein Listenelement
- Ein anderer Listenelement

[zurück](#)

Beispiel 1: Code und Ergebnis (3)

```
<html>
<head>
<title>Seitentitel</title>
<style type="text/css">
  body { background-color:#FFFFCC; margin-left:100px; }
  * { color:blue; }
  h1 { font-size:300%; color:#FF0000; font-style:italic; border-bottom:solid thin
  black; }
  p,li { font-size:100%; line-height:140%; font-family:Helvetica,Arial,sans-serif;
  letter-spacing:-0.1em; text-indent:0.3em; }
</style>
</head>
<body>
<h1>&U
<p>ein normaler Textabsatz</p>
<ul>
  <li>Ein Listenelement</li>
  <li>Ein anderer Listenelement</li>
</ul>
</body>
</html>
```

Überschriften 1. Ordnung (h1) erhalten eine Schriftgröße von 300 Prozent (font-size:300%;), die abweichende Schriftfarbe rot (color:#FF0000;) und den Schriftstil kursiv (font-style:italic;).

Überschrift 1. Ordnung

ein normaler Textabsatz

- Ein Listenelement
- Ein anderer Listenelement

[zurück](#)

Beispiel 1: Code und Ergebnis (4)

```
<html>
<head>
<title>Sei
<style type="text/css">
  body { background-color: #e0e0e0; }
  * { color: black; }
  h1 { font-size: 1.2em; }
  p,li { font-size:110%; line-height:140%; font-family:Helvetica,Arial,sans-serif;
        letter-spacing:0.1em; word-spacing:0.3em; }
</style>
</head>
<body>
<h1>&Uuml;berschrift 1. Ordnung</h1>
<p>ein normaler Textabsatz</p>
<ul>
  <li>Ein Listenpunkt</li>
  <li>Ein anderer Listenpunkt</li>
</ul>
</body>
</html>
```

Textabsätze (p) und Listenpunkte (li) sollen in 110 Prozent Schriftgröße (font-size:110%;) mit einer Zeilenhöhe von 140 Prozent (line-height:140%;) in Schriftart Helvetica, oder wenn nicht verfügbar, dann in Arial, und wenn ebenfalls nicht verfügbar, in irgendeiner Sans-Serif-Schrift dargestellt werden (font-family:Helvetica,Arial,sans-serif;).

Überschrift 1. Ordnung

ein normaler Textabsatz

- Ein Listenpunkt
- Ein anderer Listenpunkt

[zurück](#)

Details zum Beispiel 1: Hintergrundfarbe

■ **background-color**

- Sie können für jedes HTML-Element eine eigene Hintergrundfarbe definieren.
- Sie können Farbwerte **hexadezimal** nach dem **Schema #RRGGBB** angeben, **aber auch die verbreiteten Farbnamen verwenden**. R ist der Rotwert, G der Grünwert und B der Blauwert der Farbe.
- Eine weitere Möglichkeit ist das **Schema rgb(rrr,ggg,bbb)**. Innerhalb der Klammern müssen drei Dezimalwerte stehen, durch Kommata voneinander getrennt.
- Für alle drei Werte sind entweder absolute Zahlen zwischen 0 (kein Anteil der entsprechenden Farbe) und 255 (maximaler Anteil der entsprechenden Farbe) erlaubt.

```
<style type="text/css">
body { background-color:rgb(51,0,102) }
h1 { color:#FFCCDD; background-color:black }
</style>
```

Details zum Beispiel 1: Schriftfarbe

black	#000000	gray	#808080
maroon	#800000	red	#FF0000
green	#008000	lime	#00FF00
olive	#808000	yellow	#FFFF00
navy	#000080	blue	#0000FF
purple	#800080	fuchsia	#FF00FF
teal	#008080	aqua	#00FFFF
silver	#C0C0C0	white	#FFFFFF

■ color

- Mit dieser Angabe können Sie Textvordergrundfarbe bzw. Schriftfarbe bestimmen.
- Im Beispiel 1 wurde für die Überschrift die Farbe rot gewählt:

```
h1 { font-size:300%; color:#FF0000; ... }
```

■ Einschub: „Farben“

- In CSS gibt es wie in HTML die Möglichkeit, mit hexadezimalen Ziffern die 16 Grundfarben, die von jedem VGA-kompatiblen Bildschirm angezeigt werden können, zu definieren.
- Die Beschriftungen rechts zeigen noch einmal, welchen Hexadezimalnotationen entsprechend der Syntax von HTML und CSS die jeweiligen Farben entsprechen.
- Um eine Farbe mit Hilfe eines Farbnamens zu definieren, geben Sie anstelle des hexadezimalen RGB-Werts einfach den gewünschten Farbnamen an.

Details zum Beispiel 1: Schriftart

■ font-family

- **Schriftarten** sind beispielsweise Arial, Helvetica, Times Roman usw.
- Auch **Schriftfamilien** wie *Sans Serif* usw. gehören dazu.
- Bei der hier beschriebenen Eigenschaft können Sie Schriftarten angeben, ohne sich darum kümmern zu müssen, ob und wie die Schriftart beim Anwender angezeigt werden kann.
- Mit font-family können Sie **eine oder mehrere Schriftarten** bestimmen. Bei mehreren angegebenen Schriftarten ist die Reihenfolge der Angabe entscheidend: Ist die erste angegebene Schriftart verfügbar, wird diese verwendet. Ist sie nicht verfügbar, wird die zweite Schriftart verwendet, falls diese verfügbar ist usw.
- Im Beispiel sollen Textabsätze und Listenpunkte auf jeden Fall eine serifenlose Schriftart (sans-serif) verwendet werden (vgl. Folie 13):

```
p,li { ... font-family:Helvetica,Arial,sans-serif; ... }
```

<small>kleiner Beispieltext in Times New Roman</small> großer Beispieltext in Times New Roman
<small>kleiner Beispieltext in Verdana</small> großer Beispieltext in Verdana
<small>kleiner Beispieltext in Wide Latin</small> großer Beispieltext in Wide Latin
<small>kleiner Beispieltext in Comic Sans MS</small> großer Beispieltext in Comic Sans MS
<small>kleiner Beispieltext in Avantgarde</small> großer Beispieltext in Avantgarde
<small>kleiner Beispieltext in Balloon</small> großer Beispieltext in Balloon
<small>kleiner Beispieltext in Bodoni</small> großer Beispieltext in Bodoni
<small>kleiner Beispieltext in Garamond</small> großer Beispieltext in Garamond
<small>kleiner Beispieltext in Impact</small> großer Beispieltext in Impact
<small>kleiner Beispieltext in Helvetica</small> großer Beispieltext in Helvetica

Details zum Beispiel 1: Schriftstil und Schriftgröße

■ font-style

- Schriftstil bedeutet die Neigung der Schrift.
 - **italic** = kursiver Schriftstil
 - **oblique** = schräggestellter Schriftstil
 - **normal** = normaler Schriftstil

kleiner Beispieltext mit Schriftstil italic

großer Beispieltext mit Schriftstil italic

■ font-size

- Erlaubt sind
 - numerische Angaben (z.B. pt, px, em)
 - Prozentangaben (Prozentwerte beziehen sich auf die Schriftgröße des Elternelements.)
 - Absolute Schlüsselwerte
 - xx-small = winzig
 - x-small = sehr klein
 - small = klein
 - medium = mittel
 - large = groß
 - x-large = sehr groß
 - xx-large = riesig
- Beispiel:
 - `font-size: 300%` (Schriftgröße von 300 Prozent)

Details zum Beispiel 1: Rahmen

- Ein Rahmen ist anwendbar bei HTML-Elementen, die einen eigenen Absatz erzeugen bzw. einen Block bilden, also etwa für **h[1-6], p, div, table, usw.**
- **border[-top, -right, -bottom, -left]-width** (Rahmendicke)
 - Sie können für ein Element die Rahmendicke bestimmen. Sinnvollerweise sollten Sie diese Angabe zumindest mit einer Angabe zum **Rahmentyp** kombinieren, gegebenenfalls aber auch mit einer Festlegung der **Rahmenfarbe**.
 - Mit **border-width**: können Sie die Dicke des Rahmens um ein Element bestimmen. Erlaubt ist eine numerische Angabe (mit Ausnahme von Prozentwerten) für die Rahmendicke oder einer der folgenden Werte: **thin** = dünn, **medium** = mittelstark oder **thick** = dick.
 - **border-top-width** definiert die Rahmendicke für oben, **border-right-width** die Rahmendicke für rechts, **border-bottom-width** die Rahmendicke für unten und **border-left-width** die Rahmendicke für links.

Details zum Beispiel 1: Rahmen (2)

```
<p style="border-width:medium; border-color:#FF9999;  
border-style:dotted; padding:5px;">  
Textabsatz mit Rahmendicke medium und Typ dotted.</p>
```

Textabsatz mit Rahmendicke medium und Typ dotted.

```
<p style="border-width:medium; border-color:#9999FF;  
border-style:dashed; padding:5px;">  
Textabsatz mit Rahmendicke medium und Typ dashed.</p>
```

Textabsatz mit Rahmendicke medium und Typ dashed.

```
<p style="border-width:medium; border-color:#66CC66;  
border-style:solid; padding:5px;"> Textabsatz mit  
Rahmendicke medium und Typ solid.</p>
```

Textabsatz mit Rahmendicke medium und Typ solid.

```
<p style="border-width:medium; border-color:#FF9999;  
border-style:double; padding:5px;"> Textabsatz mit  
Rahmendicke medium und Typ double.</p>
```

Textabsatz mit Rahmendicke medium und Typ double.

```
<p style="border-width:12px; border-color:#9999FF;  
border-style:groove; padding:5px;"> Textabsatz mit  
Rahmendicke 12px und Typ groove.</p>
```

Textabsatz mit Rahmendicke 12px und Typ groove.

```
<p style="border-width:12px; border-color:#66CC66;  
border-style:ridge; padding:5px;"> Textabsatz mit  
Rahmendicke 12px und Typ ridge.</p>
```

Textabsatz mit Rahmendicke 12px und Typ ridge.

```
<p style="border-width:12px; border-color:#FF9999;  
border-style:inset; padding:5px;"> Textabsatz mit  
Rahmendicke 12px und Typ inset.</p>
```

Textabsatz mit Rahmendicke 12px und Typ inset.

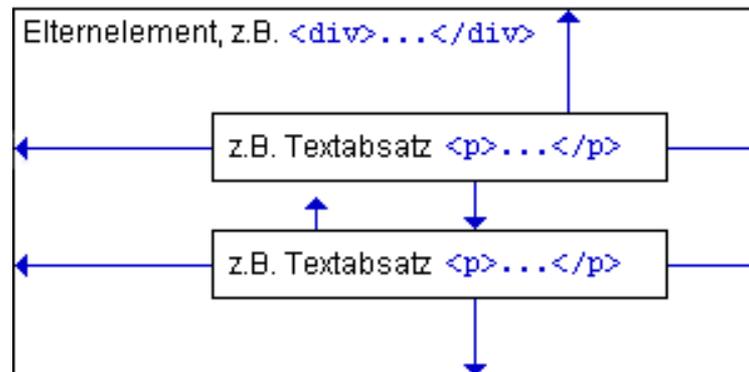
```
<p style="border-width:12px; border-color:#9999FF;  
border-style:outset; padding:5px;"> Textabsatz mit  
Rahmendicke 12px und Typ outset.</p>
```

Textabsatz mit Rahmendicke 12px und Typ outset.

Details zum Beispiel 1: Außenrand und -abstand

■ margin(-top/-right/-bottom/-left)

- Außenrand oder Außenabstand bedeutet:
Erzwungener Leerraum zwischen dem aktuellen Element und seinem Elternelement oder Nachbarelement.
- Für ein p-Element, also einen Textabsatz etwa, der direkt innerhalb des body-Elements notiert ist, markieren Angaben zu *linkem und rechten* Außenrand seinen Abstand zu den Elementgrenzen des body-Elements.
- Wenn mehrere solcher p-Absätze aufeinander folgen, markieren Angaben zum Außenrand *oben und unten* den Abstand zwischen den Absätzen. Das folgende Bild veranschaulicht die Möglichkeiten, Außenränder und Abstände zu definieren:

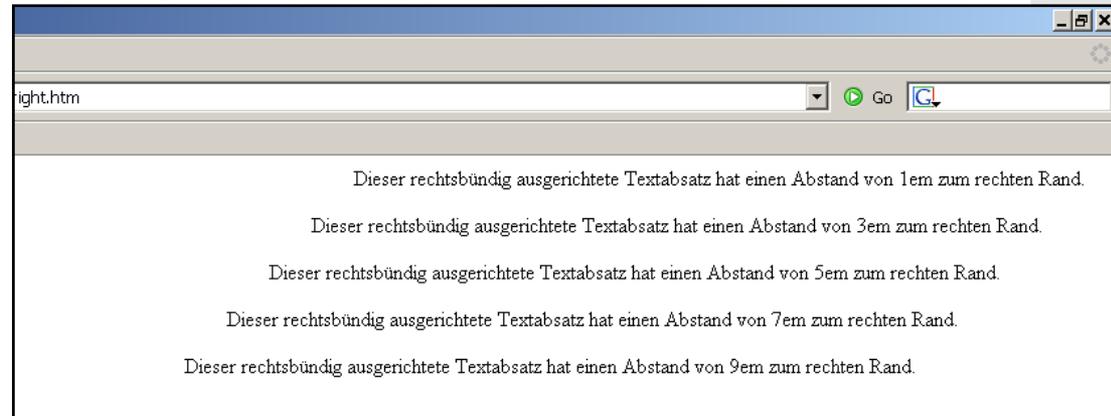


Details zum Beispiel 1: Außenrand und -abstand (2)

■ **margin right:**

```
<p align="right" style="margin-right:1em">  
Dieser rechtsbündig ausgerichtete  
Textabsatz hat einen Abstand von  
1em zum rechten Rand.</p>
```

```
<p align="right" style="margin-right:3em">  
Dieser rechtsbündig ausgerichtete  
Textabsatz hat einen Abstand von  
3em zum rechten Rand.</p>
```



■ **Einschub: „Numerische Angaben“**

- **pt (absolut):** Steht für **Punkt**. Typografische Maßeinheit. 1 Punkt entspricht 1/72 Inches. [1in = 0,0254 m]
- **px (absolut/ relativ):** Steht für **Pixel**. Abhängig von der Pixeldichte des Ausgabegeräts, relativ also von Ausgabegerät zu Ausgabegerät, absolut dagegen auf ein und das selbe Ausgabegerät bezogen.
- **em (relativ):** Steht für **Ems**. 1em entspricht der Größe des Buchstabens M in der gewählten Schriftart, basierend auf der Schriftgröße des gesamten Dokuments bzw. des Eltern-Elements. Sie können Angaben in em durch Dezimalzahlen ausdrücken, wobei 1.5em 150% der Größe des Buchstabens M entspricht.

Beispiel 2: Individualformate

- Sie können Formate definieren, die über das Universalattribut **id** angesprochen werden.
- Da die Wertzuweisung an ein solches Attribut ein dokumentweit eindeutiger Name sein sollte, handelt es sich also einfach um eine zentrale Formatdefinition für das eine Element mit diesem id-Namen.

Beispiel 2: Code und Ergebnis

```
<html>
<head>
<title>Seitentitel</title>
  <style type="text/css">
    #roterBereich {
      position:absolute;
      top:130px;
      left:30px;
      width:320px;
      padding:10px;
      margin:0px;
      border:4px solid #EE0000; }
    #blauerBereich {
      position:absolute;
      top:130px;
      left:400px;
      width:320px;
      padding:10px;
      margin:0px;
      border:4px solid #0000EE; }
    #Titel {
      font-family:Arial,sans-serif;
      font-size:2em;
      font-weight:normal;
      color:green; }
  </style>
</head>
...
```

Diese Überschrift und zwei positionierte Bereiche

Der rote Bereich

Der blaue Bereich

In dem Beispiel werden drei Individualformate definiert:
Zunächst einmal zwei Individualformate mit den Namen
#roterBereich und #blauerBereich.

Solche Formate beginnen also mit dem Gatterzeichen #,
gefolgt von dem Namen.

Ein HTML-Element, das diesen Namen als Wertzuweisung
an das id-Attribut benutzt, bekommt dann die
entsprechenden Formate zugewiesen.

Dieses Beispiel zeigt einen typischen Anwendungsfall für
Individualformate, nämlich das Definieren von Formaten
für absolut positionierte Bereiche (position:absolute;).

```
...
<body>

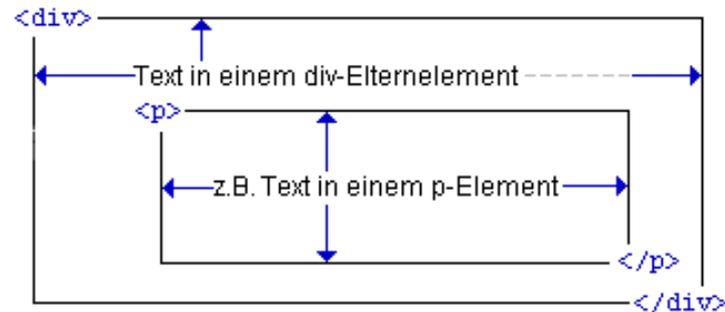
<h1 id="Titel">Diese &Uuml;berschrift und zwei
  positionierte Bereiche</h1>
<div id="roterBereich"><h1>Der rote Bereich</h1></div>
<div id="blauerBereich"><h1>Der blaue Bereich</h1></div>

</body>

</html>
```

Details zum Beispiel 2: Innenabstand

- Innenabstand bedeutet: erzwungener Leerraum zwischen dem Inhalt eines Elements und seinem eigenen Elementrand, also z.B. zwischen dem Text eines Elements und dem Rand dieses Elements.
- Das folgende Bild zeigt, wie Innenabstände zu verstehen sind (die blauen Pfeile stehen für definierte Innenabstände):



Details zum Beispiel 2: Innenabstand (2)

■ padding

- Damit bestimmen Sie den Innenabstand zwischen Elementinhalt und den vier Elementgrenzen oben, rechts, unten und links.
- Die Angabe bedeutet: **Alle vier Ränder des Elements erhalten den gleichen Innenabstand.**

- Weitere Möglichkeiten: **padding-top, padding-bottom, padding-left oder padding-right** für entsprechend Zwischenraum zwischen Elementinhalt und oberer, unterer, linker oder rechter Elementgrenze.

```
<html>
<head>
<title>Seitentitel</title>
<style type="text/css">
  #p1 { padding-left: 100px; background-color:#FFFF99; font-size:120%; }
  #p2 { padding-left: 200px; background-color:#FFFF99; font-size:120%; }
  #p3 { padding-left: 300px; background-color:#FFFF99; font-size:120%; }
</style>
</head>
<body>
<p id="p1">der erste Absatz p1</p>
<p id="p2">der zweite Absatz p2</p>
<p id="p3">der dritte Absatz p3</p>
</body>
</html>
```

der erste Absatz p1

der zweite Absatz p2

der dritte Absatz p3

Details zum Beispiel 2:

Positionierung und Anzeige von Elementen (1)

- Es gibt verschiedene Eigenschaften, um Elemente einer WWW-Seite exakt im Anzeigefenster des WWW-Browsers zu positionieren und die Position von Elementen untereinander genau zu kontrollieren.
- Dazu gehören Angaben zum **absoluten und relativen Positionieren** von Elementen, Angaben zur **Ausdehnung** von Elementen, Angaben zum **Umfließen** von Elementen und Angaben zum **Überlappen und Anzeigen** von Elementen.
- Absolutes und relatives Positionieren von Elementen bietet Ihnen die Möglichkeit, das Erscheinungsbild von Web-Seiten stärker medienorientiert zu gestalten. So können Sie für einzelne Bereiche festlegen, wo genau diese Bereiche beginnen sollen. Bereiche können sich überlappen usw.
- Das Positionieren von Elementen ist eine wichtige Voraussetzung für viele Anwendungsfälle von dynamischem HTML. Mit Hilfe entsprechender Script-Unterstützung können Sie positionierte Elemente bewegen, ein-/ausblenden, anders überlappen usw.

Details zum Beispiel 2: Positionierung und Anzeige von Elementen (2)

- Mit „position:“ können Sie die Positionsart bestimmen.
- Folgende Angaben sind erlaubt:
 - **absolute** = absolute Positionierung, gemessen am Rand des nächsthöheren Elternelements. Scrollt mit.
 - **fixed** = absolute Positionierung, gemessen am "Viewport", d.h. am Browserfenster. Bleibt beim Scrollen stehen.
 - **relative** = relative Positionierung, gemessen an der Normalposition oder Anfangsposition des Elements selbst.
- Die Angabe position: legt noch nicht fest, wo genau ein Element beginnen soll. **Die Angabe ist nur sinnvoll, wenn Sie zugleich die gewünschte Startposition angeben.**
- Startposition können sein:
**top (von oben), left (von links),
bottom (von unten) oder right (von rechts)**

Details zum Beispiel 2: Positionierung und Anzeige von Elementen (3)



```
<div style="position:absolute; top:180px">
  
</div>
```

```
<div style="position:absolute;
  top:136px;
  width:208px;
  background-color:#FFFFE0;
  border:1px solid #804000;
  padding:10px">
  Dies ist ein absolut positionierter Text
  &uuml;ber einer absolut positionierten
  Grafik.
</div>
```



```
<div style="position:absolute; top:100px; left:100px;">
  
</div>
```

```
<div style="position:absolute;
  top:140px;
  left:350px;
  width:200px;
  background-color:#FFFFE0;
  border:1px solid #804000;
  padding:10px">
  Dies ist ein absolut positionierter Text neben einer
  absolut positionierten Grafik.
</div>
```

Details zum Beispiel 2: Positionierung und Anzeige von Elementen (4)

■ **width (Breite)**

- Sie können bestimmen, wie breit ein Element oder Bereich sein soll. Um zu bestimmen, was mit dem Inhalt passieren soll, wenn er mehr Breite einnimmt als die hier angegebene Breite, können Sie zusätzlich die Eigenschaft „overflow“ verwenden.

```
<h1 style="width:200px;  
border:10px solid red;  
padding:10px;  
font-size:3em;  
overflow:hidden;">  
Eine Überschrift, die möglicherweise breiter ist als erlaubt  
</h1>
```

- Entsprechend können Sie mit **height (Höhe)** bestimmen, wie hoch ein Element oder Bereich sein soll.

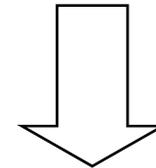


Zusatz 1:

Formate für verschachtelte HTML-Elemente definieren

- Wenn Sie nichts anderes angeben, übernimmt ein HTML-Element, das innerhalb eines anderen HTML-Elements vorkommt, dessen Eigenschaften und fügt seine eigenen Eigenschaften nur hinzu.
- Wenn Sie beispielsweise für Überschriften 1. Ordnung die Schriftart Times und die Farbe rot definieren, erscheint Text, der innerhalb einer solchen Überschrift mit `<i>...</i>` formatiert wird, ebenfalls rot und in Times, aber zusätzlich kursiv.
- Sie können mit Hilfe von Stylesheets z.B. bestimmen, dass ein HTML-Element bestimmte Eigenschaften nur dann hat, wenn es innerhalb eines bestimmten anderen HTML-Elements vorkommt.
→ **So können Sie etwa bestimmen, dass `<i>...</i>` innerhalb von Überschriften nicht kursiv, stattdessen aber in blauer Farbe dargestellt wird, während das gleiche Element innerhalb anderer HTML-Tags nach wie vor nichts anderes als eine kursive Darstellung bewirkt.**

```
<html>
<head>
<title>Titel der Datei</title>
<style type="text/css">
    h1 { color:red; }
    h1 i { color:blue; font-style:normal; }
</style>
</head>
<body>
    <h1>Wir lernen <i>Stylesheets</i></h1>
    <p>Wir lernen <i>Stylesheets</i></p>
</html>
```



Wir lernen Stylesheets

Wir lernen Stylesheets

Zusatz 2: Formate für Klassen definieren

- Sie können Formate für Klassen definieren. Anwenden können Sie solche Klassen in HTML mit dem Universalattribut **class**.
- So können Sie beispielsweise für Überschriften 1. Ordnung zwei Varianten erzeugen. Dazu vergeben Sie Namen für die Klassen:

```
<html>
<head>
<title>Titel der Datei</title>
<style type="text/css">
    h1 { font-family:Arial,sans-serif; font-size:2em; font-weight:normal; }
    h1.hinterlegt { background-color:#FFFF00 }
</style>
</head>
<body>
<h1>H1 ganz normal nur etwas formatiert</h1>
<h1 class="hinterlegt">H1 knallgelb hinterlegt</h1>
</body>
</html>
```

- Es gibt zwei Möglichkeiten, Klassen für HTML-Elemente zu notieren:
 - Entweder für einen bestimmten HTML-Elementtyp,
 - oder für keinen bestimmten.
- In jedem Fall bestimmen Sie eine Formatdefinition für eine Klasse, indem Sie im Selektor einen Punkt notieren und unmittelbar dahinter einen Namen für die Klasse angeben.
- **Im obigen Beispiel wird mit `h1.hinterlegt` eine Klasse namens „hinterlegt“ angesprochen, die aber nur für HTML-Elemente vom Typ `h1` gilt: also z.B. auf HTML-Elemente wie `<h1 class="hinterlegt">` anwendbar ist.**

Übersicht: Einführung in CSS

- Aufgabenstellung
- I: Stylesheets in HTML einbinden
 - Formate zentral in HTML-Datei definieren
 - Formate zentral in separater CSS Datei definieren
 - Formate innerhalb eines HTML-Elements definieren
- II: CSS-Eigenschaften
 - Beispiel 1: Zentrale Formate für HTML-Elemente definieren
 - Hintergrundfarbe
 - Schriftfarbe
 - Einschub: Farben
 - Schriftart
 - Schriftstil und Schriftgröße
 - Außenrand und Abstand
 - Einschub: Numerische Angaben
 - Rahmen
 - Beispiel 2: Individualformate definieren
 - Innenabstand
 - Positionierung und Anzeige von Elementen
 - Zusatz 1: Formate für verschachtelte HTML-Elemente definieren
 - Zusatz 2: Formate für Klassen definieren
- **III: CSS-basierte Layouts**
 - **Beispiel 3: Zweispaltiges CSS-basiertes Layout**
 - **Pseudoklassen für Verweise**
 - **Textumfluss**

Was sind CSS-basierte Layouts?

- Ein CSS-basiertes Layout setzt das CSS zugrunde liegende Konzept der Trennung des Inhalts und der Struktur einer Seite von deren Präsentation um. Layoutänderungen sind dadurch bereits mit nur geringer Modifikation der CSS-Definitionen zu realisieren und sofern diese in separaten Dateien notiert sind, können Sie das Layout für das gesamte Projekt hierüber einheitlich bestimmen.
- Im Gegensatz zu Tabellen-Layouts wird die Seitenaufteilung nicht mehr ausschließlich per HTML-Auszeichnung festgelegt. Sie können die Anordnung der Elemente auf einer Seite über CSS flexibel definieren und sind nicht auf eine unflexible Einteilung in Zeilen und Spalten beschränkt. CSS-basierte Layouts eröffnen Ihnen Möglichkeiten der Seitengestaltung, die mit Tabellen nicht umsetzbar sind.
- Durch eine sinnvolle Auswahl der Elemente und schlüssige Anordnung der Inhalte mit Formatierung und Positionierung per CSS wird die Zugänglichkeit verbessert. Die Erstellung von barrierefreien Seiten gewinnt zunehmend an Bedeutung. Diese Entwicklung wird nicht zuletzt von der Gesetzgebung (betreffend Internetpräsenzen, die unter staatlicher Aufsicht stehen) gefördert. Darüber hinaus erleichtern CSS-Layouts die Pflege der Seiten und reduzieren den Umfang des HTML-Quelltextes und damit die Ladezeiten deutlich.

Beispiel 3: Zweispaltiges CSS-basiertes Layout (HTML-Code)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>Zweispaltiges Layout ohne CSS</title>
  </head>
  <body>
    <ul id="Navigation">
      <li><a href="einfuehrung.htm">Einführung</a></li>
      <li><a href="seite1.htm">1. Seite</a></li>
      <li><a href="seite1.htm">2. Seite</a></li>
      <li><a href="seite1.htm">3. Seite</a></li>
    </ul>

    <div id="Inhalt"> #
      <h1>CSS-basierte Layouts</h1>
      <h2>HTML-Struktur ohne CSS</h2>
      <p>Diese Seite enthält zunächst nur die Struktur und den Inhalt.
        Das Aussehen wird erst im nachfolgenden Beispiel über CSS
        definiert.</p>
      <p>Dennoch ist diese Seite schon nutzbar und wird vom Browser
        entsprechend dessen Voreinstellungen bereits sinnvoll
        angezeigt.</p>
    </div>
  </body>
</html>
```

Beispiel 3: Zweispaltiges CSS-basiertes Layout (HTML-Code 2)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>Zweispaltiges Layout ohne CSS</title>
  </head>
  <body>
    <ul id="Navigation">
      <li><a href="einfuehrung.htm">Einführung</a></li>
      <li><a href="seitenuebersicht.htm">Seitenübersicht</a></li>
      <li><a href="kontakt.htm">Kontakt</a></li>
      <li><a href="faq.htm">FAQ</a></li>
    </ul>
    <div id="Inhalt">
      <h1>CSS-basierte Layouts</h1>
      <h2>HTML-Struktur ohne CSS</h2>
      <p>Diese Seite enthält zunächst nur die Struktur und den Inhalt.
        Das Aussehen wird erst im nachfolgenden Beispiel über CSS
        definiert.</p>
      <p>Dennoch ist diese Seite schon nutzbar und wird vom Browser
        entsprechend dessen Voreinstellungen bereits sinnvoll
```

Für die Navigationspunkte wurde eine Aufzählungsliste verwendet, deren ID „Navigation“ später als Selektor für die CSS-Definitionen dient.

Die Elemente des Inhaltsbereichs werden mit einem „div“-Element gruppiert, das ebenfalls eine ID („Inhalt“) erhält und die spätere Formatierung des Inhaltsbereichs und seiner Elemente über CSS erleichtert.

Beispiel 3: Browser Ansicht (nur HTML)

- [Einführung in CSS-basierte Layouts](#)
- [1. Seite](#)
- [2. Seite](#)
- [3. Seite](#)

CSS-basierte Layouts

HTML-Struktur ohne CSS

Diese Seite enthält zunächst nur die Struktur und den Inhalt. Das Aussehen wird erst im folgenden Beispiel über CSS definiert.

Dennoch ist diese Seite schon nutzbar und wird vom Browser entsprechend dessen Voreinstellungen bereits sinnvoll angezeigt.

- Um die **Navigation links neben dem Inhaltsbereich** zu platzieren, aber auch die Aufzählungspunkte der Listenelemente zu entfernen und die Präsentation der übrigen Elemente selbst zu definieren und nicht den Voreinstellungen der Browser zu überlassen, wird das folgende Stylesheet in die HTML-Datei eingebunden.



Lösungsvorschlag

```
body {
  color: black;
  background-color: white;
  font-family: Helvetica,Arial,sans-serif;
  margin: 0;
  padding: 1em; }
```

...

```
ul#Navigation {
  font-size: 0.90em;
  float: left;
  width: 22em;
  margin: 0;
  padding: 0;
  border: 1px dashed silver; }
```

```
ul#Navigation li {
  list-style: none;
  margin: 0;
  padding: 0.5em; }
```

```
ul#Navigation a {
  display: block;
  padding: 0.2em;
  font-weight: bold; }
```

```
ul#Navigation a:link {
  color: black;
  background-color: #999999; }
```

```
ul#Navigation a:visited {
  color: white;
  background-color: #333333; }
```

```
ul#Navigation a:hover {
  color: black;
  background-color: #0099CC; }
```

```
ul#Navigation a:active {
  color: white;
  background-color: #0099CC; }
```

```
div#Inhalt {
  margin-left: 22em;
  padding-top: 0;
  padding-bottom: 0;
  padding-right: 1em;
  padding-left: 1em;
  border: 1px dashed silver;
}
```

```
div#Inhalt h1 {
  font-size: 1.5em;
  margin-top: 0;
  margin-bottom: 1em;
  margin-right: 0;
  margin-left: 0;
}
```

```
div#Inhalt h2 {
  font-size: 1.2em;
  margin-top: 0;
  margin-bottom: 1em;
  margin-right: 0;
  margin-left: 0;
}
```

```
div#Inhalt p {
  font-size: 1em;
  margin-top: 1em;
  margin-bottom: 1em;
  margin-right: 0;
  margin-left: 0;
}
```

Beispiel 3: Browser Ansicht (HTML und CSS)



float:

Sie können für einen Bereich oder ein Element bestimmen, dass nachfolgende Elemente diesen Bereich bzw. dieses Element umfließen.

(vgl. Folien 46 u. 47)

Mittels **Pseudoklassen** können Sie Deklarationen für HTML-Bestandteile definieren, die sich nicht durch ein eindeutiges HTML-Element ausdrücken lassen, z.B. ein „noch nicht besuchter Verweis“.

Zum Definieren solcher Pseudoklassen gibt es eine spezielle Syntax.

(vgl. Folie 45)

```
color: black;
background-color: white;
font-family: Helvetica,Arial,sans-serif;
margin: 0;
padding: 1em; }
```

...

Hier wird ein Individualformat mit dem Selektor **div#Inhalt h1** definiert. Das ist eine Syntax, die erst seit CSS 2.0 zur Verfügung steht.

Dabei wird vor dem Gatterzeichen, das ein Individualformat kennzeichnet, ein HTML-Elementtyp notiert, im Beispiel also div (divison) für ein allgemeines Blockelement. Hinter dem Gatterzeichen dann wieder der Name des Individualformats.

Im Beispiel wird die Formatdefinition nur dann angewendet, wenn es in der HTML-Datei eine Überschrift h1 in einem div-Element mit dem Attribut id=„Inhalt“ gibt.

```
ul#Navigation {
font-size: 0.9em;
float: left;
width: 22em;
margin: 0;
padding: 0;
border: 1px dashed silver; }
```

```
ul#Navigation li {
list-style: none;
margin: 0;
padding: 0.5em; }
```

```
ul#Navigation a {
display: block;
padding: 0.2em;
font-weight: bold; }
```

```
ul#Navigation a:link {
color: black;
background-color: #999999; }
```

```
ul#Navigation a:visited {
color: white;
background-color: #333333; }
```

```
ul#Navigation a:hover {
color: black;
background-color: #0099CC; }
```

```
ul#Navigation a:active {
color: white;
background-color: #0099CC; }
```

```
border: 1px dashed silver; }
```

```
div#Inhalt h1 {
font-size: 1.5em; }
```

```
div#Inhalt h2 {
font-size: 1.2em;
margin-top: 0; }
```

```
div#Inhalt p {
font-size: 1em;
margin-top: 1em;
margin-bottom: 1em;
margin-right: 0;
margin-left: 0; }
```

list-style: none
Bewirkt, dass keine Aufzählungszeichen verwendet werden. (vgl. Folie 48)

display: block
Erzwingt einen neuen Block für jeden Link in der Navigationsleiste (vgl. Folie 49)

Details zum Beispiel 3: Pseudoklassen für Verweise

- Sie können das Erscheinungsbild von Verweisen zu noch nicht besuchten Seiten, zu bereits besuchten Seiten und zu Verweisen, die gerade mit der Maus überfahren oder angeklickt werden, bestimmen.
- Pseudoklassen werden zentral in einem style-Bereich notiert. Die hier beschriebenen Pseudoklassen gelten für das a-Element in HTML, daher wird vor dem Doppelpunkt das a notiert. In den Formatdefinitionen für die einzelnen Pseudoklassen können Sie beliebige, geeignete CSS-Eigenschaften zuweisen. Die folgenden Pseudoklassen bedeuten:
 - **:link** für Verweise zu noch nicht besuchten Seiten
 - **:visited** für Verweise zu bereits besuchten Seiten
 - **:focus** für Verweise, die den Fokus erhalten, z.B. durch "Durchsteppen" mit der Tabulator-Taste (CSS 2.0)
 - **:hover** für Verweise, während der Anwender mit der Maus darüber fährt (CSS 2.0)
 - **:active** für gerade angeklickte Verweise

Details zum Beispiel 3: Textumfluss

■ float

- Sie können für einen Bereich oder ein Element bestimmen, dass nachfolgende Elemente diesen Bereich bzw. dieses Element umfließen.
- Folgende Angaben sind möglich:
 - left = Element steht links und wird rechts davon von nachfolgenden Elementen umflossen.
 - right = Element steht rechts und wird links davon von nachfolgenden Elementen umflossen.

Überschrift mal anders

Eine Überschrift muss nicht immer einen langweiligen, über die ganze Breite des Anzeigefensters gehenden Block erzeugen. Sie kann auch mal seitlich vom Text stehen, und der Text fließt um die Überschrift. Auch so kann man lange Texte mal anders gestalten als sonst üblich.

Eine Überschrift muss nicht immer einen langweiligen, über die

```
<h1 style="width:400px; float:left; color:red">
&Uuml;berschrift mal anders</h1>
```

```
<p style="font-size:120%">Eine &Uuml;berschrift
muss nicht immer einen langweiligen, &uuml;ber die
ganze Breite des Anzeigefensters gehenden Block
erzeugen. Sie kann auch mal seitlich vom Text
stehen, und der Text flie&szlig;t um die
&Uuml;berschrift. Auch so kann man lange Texte mal
anders gestalten als sonst &uuml;blich.</p>
```

Details zum Beispiel 3: Fortsetzung bei Textumfluss

- Wenn Sie mit float einen Textumfluss definieren, möchten Sie vielleicht nur zwei oder drei Zeilen kommentierenden Text neben einem umflossenen Element haben und den folgenden Text dann unterhalb des umflossenen Elements fortsetzen.
- Mit clear können Sie einen Umfluss abbrechen und die Fortsetzung unterhalb des umflossenen Elements oder Bereichs erzwingen. Folgende Angaben sind möglich:
 - **left** = Erzwingt bei float:left die Fortsetzung unterhalb.
 - **right** = Erzwingt bei float:right die Fortsetzung unterhalb.
 - **both** = Erzwingt in jedem Fall die Fortsetzung unterhalb.

```

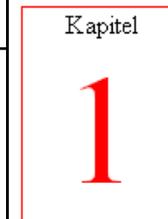
<div style="float:left;
margin-right:20px;
margin-bottom:20px;
border:solid 1px red;
text-align:center"> Kapitel

  <span style="font-family:Algerian,serif;
font-size:96px;
color:red"> 1

</span>

</div>
<p>In diesem Kapitel geht es um Einf&uuml;hrendes.</p>
<p style="clear:left">Dann kommen wir mal zur Sache: ...</p>

```



In diesem Kapitel geht es um Einführendes.

Dann kommen wir mal zur Sache: ...

Details zum Beispiel 3: Listenformatierung

- Die Stylesheet-Angaben für Listen beziehen sich speziell auf die HTML-Elemente ul (Aufzählungslisten) und ol (nummerierte Listen).
- Für Aufzählungslisten können Sie das Erscheinungsbild des Bullet-Zeichens bestimmen. Für nummerierte Liste können Sie die Darstellung der Durchnummerierung beeinflussen.
- Mit **list-style-type**: können Sie das Aussehen von Listenzeichen kontrollieren. Erlaubt ist eine der folgenden Angaben.
 - **decimal** = für ol-Listen: Nummerierung 1.,2.,3.,4. usw.
 - **lower-roman** = für ol-Listen: Nummerierung i.,ii.,iii.,iv. usw.
 - **upper-roman** = für ol-Listen: Nummerierung I.,II.,III.,IV. usw.
 - **lower-alpha** oder **lower-latin** = für ol-Listen: Nummerierung a.,b.,c.,d. usw.
 - **upper-alpha** oder **upper-latin** = für ol-Listen: Nummerierung A.,B.,C.,D. usw.
 - **disc** = für ul-Listen: gefüllter Kreis als Bullet-Zeichen
 - **circle** = für ul-Listen: leerer Kreis als Bullet-Zeichen
 - **square** = für ul-Listen: rechteckiges Bullet-Zeichen
 - **none** = kein Bullet-Zeichen, keine Nummerierung
- *Im Beispiel 3 bewirkt die Angabe **ul#Navigation li** { list-style: none; ...}, dass vor den Aufzählungslisten der Navigation keine Bullet-Zeichen erscheinen.*

Details zum Beispiel 3: Anzeigeart

■ **display**

- Sie können mit der hier beschriebenen Angabe bestimmen, wie ein Element angezeigt werden soll.
 - Sie können bestimmen, dass es einen eigenen (absatzerzeugenden) Block bilden soll, oder dass es mitten im Text (inline) angezeigt werden soll.
 - So können Sie beispielsweise absatzerzeugende Elemente wie Überschriften in Elemente umdefinieren, die keinen eigenen Absatz erzeugen.
-
- Folgende Angaben sind u.a. möglich:
 - **block** = Erzwingt einen Block - das Element erzeugt eine neue Zeile.
 - **inline** = Erzwingt die Anzeige im Text - das Element wird im laufenden Textfluss angezeigt.
-
- *Im Beispiel 3 erzwingt die Angabe **ul#Navigation a** { display: block; ...} jeweils einen Block für einen neuen Link <a href...>.*