

---

# **CSS-SCHULUNG**

---

**TEIL A:  
ALLGEMEINE EINFÜHRUNG IN CSS**

**TEIL B:  
CSS IM LOKALEN INFORMATIONSSYSTEM**

---

**Dozentin:  
Dipl. Ing.(FH) Nicole Hartmann  
Rhoen-Saale.net GmbH  
Oktober 2009**

---

# TEIL A: ALLGEMEINE EINFÜHRUNG IN CSS

## 1) Einführung – Was ist CSS?

CSS ist die Abkürzung für **C**ascading **S**tyle **S**heets.

CSS ist eine Sprache, die das Aussehen von HTML-Dokumenten definiert. CSS kann man z.B. zum Festlegen von Schriftarten, Farben, Rändern, Linien, Höhen, Breiten, Hintergrundbildern, für Positionierung und viele andere Sachen benutzen.

CSS hat viele Vorteile:

- Trennung von Layout und Inhalt zur leichteren Administration
- Kontrolle über das Layout vieler Webseiten aus einer einzigen Style-Sheet-Datei heraus
- präzisere Kontrolle über das Layout
- verschiedene Layouts für verschiedene Medientypen (Bildschirmanzeige, Druck etc.)
- Barrierefreie Gestaltung einer Webseite durch einfaches Austauschen des CSS

## 2) Wie wird CSS in ein HTML-Dokument eingebunden?

Um CSS-Formatierungen in einem HTML-Dokument einzubinden gibt es 3 Möglichkeiten.

### a) In-line als Style-Attribut

Die Style-Angabe kann direkt in das Style-Attribut des HTML-Tags eingetragen werden. Die Angaben gelten nur für diesen einen Tag und haben für die restlichen HTML-Tags keine Bedeutung.

```
<b style="background-color: red;">Text mit rotem Hintergrund</b>
```

### b) In-document

Die Style-Angabe kann innerhalb des Dokuments in einem eigenen style-Block definiert werden. Hierbei gilt die Style-Angabe für alle entsprechenden Elemente innerhalb des Dokuments.

```
<style type="text/css">
  b {background-color: green;}
</style>

<b>Text mit grünem Hintergrund</b>
<b>weiterer Text mit grünem Hintergrund</b>
```

### c) Extern in eigener Datei

Der gängigste Weg, Style-Angaben für ein HTML-Dokument zu speichern, ist eine externe css-Datei. Diese Methode wird dringend empfohlen, denn nur so kann man zentral das Layout eines kompletten Internetauftritts steuern ohne die einzelnen HTML- oder PHP-Dateien zu ändern. Eine css-Datei ist eine einfache Text-Datei mit der Datei-Endung .css. Die Datei kann an jeder beliebigen Stelle im Document-Root liegen, sie muss nur in der HTML- oder PHP-Datei richtig integriert sein. Dies wird mit folgender Zeile im HEAD-Bereich des HTML-Dokuments realisiert:

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

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

Der große Vorteil dieser Variante: Wenn das CSS in jedes Dokument des Internet-Auftritts eingebunden ist, dann kann durch eine Änderung in der CSS-Datei gleichzeitig das Layout aller HTML-

Dateien des Internet-Auftritts geändert werden. Ganz einfach ohne großen Aufwand kann so eine Neugestaltung des Internet-Auftritts erfolgen.  
Auf diesem Konzept basiert auch die Layout-Gestaltung des LIS.

#### Aufgabe

Erstellen Sie eine Datei namens 1\_beispiel.html und eine Datei namens 1\_style.css. Wenden Sie die drei oben genannten Methoden an, um folgendes zu realisieren:

- In-line: Fetter Text mit rotem Hintergrund (red)
- In-document: Fetter Text mit silbernem Hintergrund (silver)
- Extern: Absatz <p> mit hellgrünem Hintergrund (lime)

### 3) Grundlegende CSS-Syntax in einer CSS-Datei

Eine CSS-Datei besteht aus mehreren Blöcken, die jeweils das Aussehen eines HTML-Elements bestimmen. Es können beliebig viele solcher Blöcke in die CSS-Datei geschrieben werden. Der Aufbau eines solchen Blockes ist immer identisch entsprechend des folgenden Schemas:

```
selector{
  eigenschaft1: wert1;
  eigenschaft2: wert2;
  ...
}
```

Der „selector“ gibt an, welche HTML-Elemente von diesem Layout-Block formatiert werden sollen. Als selector können HTML-Tags (z.B. b, body, p, a), class-Namen und IDs aus dem HTML-Dokument verwendet werden.

#### a) HTML-Tag als selector

HTML-Tags können einfach direkt als selector eingetragen werden:

**html-Datei:**

```
<b>roter Hintergrund</b>
<a href="info.html">Link mit blauem Hintergrund</a>
```

**css-Datei:**

```
b{
  background-color:red;
}
a{
  background-color:blue;
}
```

#### b) Class-Name als selector

Class-Name bezieht sich auf das Klassen-Attribut im HTML-Tag. So können verschiedenen HTML-Tags gleiche Klassen zugeordnet werden.

Bei Class-Namen als selectoren muss ein Punkt vorangestellt werden, um den selector als Class-Name kenntlich zu machen.

**html-Datei:**

```
<b class=roterHintergrund>roter Hintergrund</b>
<a href="info.html" class=roterHintergrund>Link mit rotem Hintergrund</a>
```

**css-Datei:**

```
.roterHintergrund{
  background-color:red;
}
```

Der class-Name kann frei gewählt werden, dabei ist allerdings zu beachten, dass keine Leerzeichen und Sonderzeichen verwendet werden dürfen.

### c) ID als selector

Die ID bezieht sich auf das ID-Attribut im HTML-Tag. So kann gezielt ein einziger HTML-Tag formatiert werden. Eine ID kann nur einmalig für einen HTML-Tag vergeben werden, es können nicht mehrere HTML-Tags die gleiche ID verwenden.

Bei der ID als selector muss ein Hash (#) vorangestellt werden, um den selector als ID kenntlich zu machen.

**html-Datei:**

```
<b id=spezialFeld>roter Hintergrund</b>
<b>kein Hintergrund</b>
```

**css-Datei:**

```
#spezialFeld{
  background-color:red;
}
```

Die ID kann frei gewählt werden, dabei ist allerdings zu beachten, dass keine Leerzeichen und Sonderzeichen verwendet werden dürfen.

### d) Listen von selectoren

Selectoren können auch gelistet werden, d.h. eine Formatierung kann gleichzeitig für mehrere HTML-Tags angewendet werden. Die Selectoren werden dann mit Komma getrennt.

**html-Datei:**

```
<b id=spezialFeld>roter Hintergrund</b>
<b id=auchSpezialFeld>auch roter Hintergrund</b>
```

**css-Datei:**

```
#spezialFeld, #auchSpezialFeld{
  background-color:red;
}
```

### e) Kombination von selectoren

Selectoren können auch beliebig kombiniert und verknüpft werden, um gezielt Elemente im HTML-Dokument anzusprechen.

**html-Datei:**

```
<p>
  <b>Fetter Text in einem Absatz</b>
</p>
<b>Fetter Text außerhalb eines Absatzes</b>
```

**css-Datei:**

```
p b{
  background-color:red;
}
```

Hierbei können auch Klassen oder IDs in die Selector-Definition mit einbezogen werden. Die selectoren-Beschreibung kann dadurch recht komplex werden. Daher sollten möglichst aussagekräftige Klassen und IDs verwendet werden.

**html-Datei:**

```
<p class=absatz1>
  <b>Fetter Text in einem Absatz mit <a href="dummy.html" id=dummy>Link</a></b>
</p>
<b>Fetter Text außerhalb eines Absatzes</b>
```

**css-Datei:**

```
p.absatz1{
  background-color:lime;
}
p.absatz1 b{
```

```

background-color:silver;
}
p.absatz1 b a#dummy{
background-color:red;
}

```

An die selectoren können noch zusätzliche Bedingungen geknüpft werden, dies würde jedoch im Grundlagen-Tutorial zu weit führen, zudem kommen diese Techniken im LIS nicht zum Einsatz und sind daher im Moment nicht relevant.

## f) Eigenschaft und Wert

Für die Layout-Gestaltung der einzelnen HTML-Elemente, die durch die selectoren ausgewählt werden, gibt es nun zahlreiche Möglichkeiten. Grundsätzlich gibt es immer ein Paar aus Eigenschaft und Wert:

```

p{
[Eigenschaft]: [Wert];
background-color: red;
}

```

Im obigen Beispiel ist **background-color** die Eigenschaft und **red** der Wert. Eigenschaft und Wert werden immer durch Doppelpunkt getrennt. Und die gesamte Zeile muss immer mit einem Semikolon abgeschlossen werden.

Ein style-Block kann beliebig viele Eigenschaft-Werte-Paare beinhalten.

```

p{
background-color: red;
color: blue;
margin: 10px;
...
}

```

### Aufgabe

Erstellen Sie eine Datei 2\_beispiel.html und kopieren Sie den folgenden Code in die Datei.

```

<div id=block1>
  <div class=unterblock1>
    <H1>Erste Überschrift</H1>
    Dies ist der Text zur Überschrift.
  </div>
</div>
<div class=unterblock1>Ablenkungstext</div>

```

Erstellen Sie die Datei 2\_style.css und definieren Sie darin das Layout für das obige HTML. Geben Sie der Überschrift einen roten Hintergrund und dem Textblock zur Überschrift einen silbernen Hintergrund.

## 4) Farben und Hintergründe

Farben und Hintergründe sind ein wesentliches Gestaltungsmittel für Internetseiten.

Für Farben kann entweder der Farbname (red, green, blue, silver, ...) verwendet werden oder üblicherweise der Hexadezimal-Code eines Farbwertes (#FF0033). Beim Hexadezimal-Code stehen die ersten beiden Stellen für den Rot-Wert, die nächsten beiden für den Grün-Wert und die letzten beiden für den Blau-Wert. (siehe z.B. <http://www.ipsi.fraunhofer.de/~cruieger/farbe/farb-must.html>)

### a) Vordergrundfarbe (color)

Die Eigenschaft **color** bezeichnet die Vordergrundfarbe oder auch Textfarbe.

Möchte mal beispielsweise alle Überschriften rot schreiben, wird dies mit der folgenden Zeile realisiert:

```

h1{
color: #FF0000;
}

```

## b) Hintergrundfarbe (background-color)

Die Eigenschaft **background-color** bezeichnet die Hintergrundfarbe des selektierten Elements. Möchte man beispielsweise allen Überschriften einen silbernen Hintergrund geben, wird dies mit der folgenden Zeile realisiert:

```
h1{
  background-color: silver;
}
```

## c) Hintergrund-Bild (background-image)

Zusätzlich zur Hintergrundfarbe kann auch ein Hintergrund-Bild für das selektierte Element vergeben werden. Dazu muss das Bild auf dem Server vorliegen und entsprechend referenziert werden:

```
p{
  background-color: silver;
  background-image: url("bg_image.gif");
}
```

## d) Wiederholung des Hintergrund-Bildes (background-repeat)

Standardmäßig wird das Hintergrund-Bild sowohl vertikal als auch horizontal so oft wiederholt, bis das gesamte Element ausgefüllt ist. Dieses Verhalten möchte man manchmal jedoch ändern. Dazu gibt es die Eigenschaft **background-repeat**:

```
p{
  background-color: silver;
  background-image: url("bg_image.gif");
  background-repeat: no-repeat;
}
```

Die Eigenschaft **background-repeat** kann folgende Werte annehmen:

background-repeat: no-repeat;	Keine Wiederholung
background-repeat: repeat-x;	Nur horizontale Wiederholung
background-repeat: repeat-y;	Nur vertikale Wiederholung
background-repeat: repeat;	Wiederholung in alle Richtungen (=Standard)

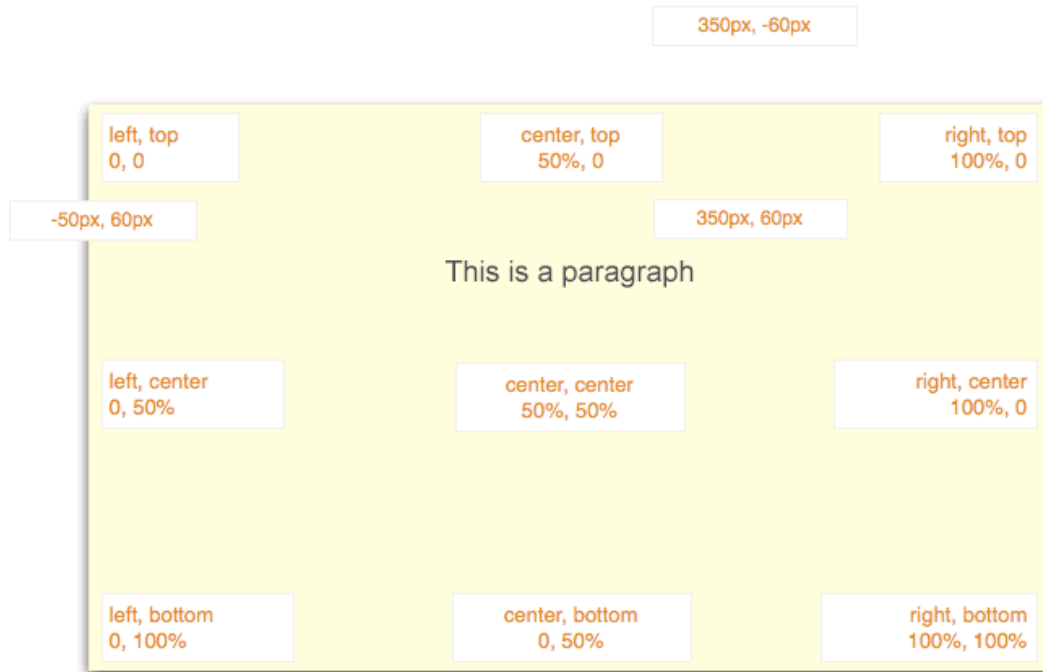
## e) Positionierung des Hintergrund-Bildes (background-position)

Das Hintergrund-Bild kann auch gezielt innerhalb des HTML-Elements positioniert werden. Dazu wird die Eigenschaft **background-position** verwendet:

```
p{
  background-color: silver;
  background-image: url("bg_image.gif");
  background-repeat: no-repeat;
  background-position: center center;
}
```

Der erste Wert bestimmt die horizontale und der Zweite die vertikale Position. Als Werte sind Längenangaben (in px, em, ...) , prozentuale Angaben (50%, ...) und vordefinierte Schlüsselwörter (top, bottom, left, right, center) erlaubt.

Ursprung des dafür angenommenen Koordinaten-Systems ist in der linken oberen Ecke des HTML-Elements.



## f) Verkürzte Schreibweise von background

Die Eigenschaft **background** ist eine Abkürzung für alle Eigenschaften aus dieser Lektion (außer **color**). Mit **background** können Sie mehrere Eigenschaften in eine zusammenfassen und so Ihr Stylesheet verkürzen und leichter lesbar machen:

```
background-color: silver;
background-image: url("bg_image.gif");
background-repeat: no-repeat;
background-position: right bottom;
```

Bei Verwendung von **background** kann der gleiche Effekt mit nur einer Codezeile erreicht werden:

```
background: silver url("bg_image.gif") no-repeat right bottom;
```

Die Reihenfolge ist dabei wie folgt:

**[background-color] | [background-image] | [background-repeat] | [background-attachment] | [background-position]**

Wird eine Eigenschaft ausgelassen, wird automatisch der Standardwert der jeweiligen Eigenschaft eingesetzt. Wird z.B. background-position aus dem Beispiel entfernt:

```
background: silver url("butterfly.gif") no-repeat;
```

wird diese Eigenschaften auf ihre Standardwerte **top** gesetzt.

### Aufgabe

Erstellen Sie eine Datei namens 3\_beispiel.html und 3\_style.css.

Erstellen Sie in der html-Datei 3 Textblöcke/Paragrafen (<p>) mit einem beliebigen mehrzeiligen Text.

Formatieren Sie den ersten Textblock mit einer blauen Schriftfarbe in Hexadezimal-Code und einem hell-roten Hintergrund.

Formatieren Sie den zweiten Textblock mit dunkel-grüner Schriftfarbe. Setzen Sie ein kleines Bild als sich wiederholenden Hintergrund ein.

Formatieren Sie den dritten Textblock mit dunkel-blauer Schriftfarbe. Setzen Sie ein kleines Bild einmalig in den Hintergrund. Das Bild soll horizontal mittig und mit 10px Abstand von oben positioniert sein.

## 5) Schriftarten

Mit der Gestaltung der Schrift lässt sich das Layout einer Seite wesentlich beeinflussen. Deshalb werden hier nur die wichtigsten Schrift-Eigenschaften erläutert.

### a) Zeichensatzfamilie: `font-family`

Die Eigenschaft **font-family** wird verwendet, um eine Liste der Zeichensätze anzugeben, in der ein bestimmtes Element oder eine Webseite angezeigt werden soll. Wenn der erste Zeichensatz nicht auf dem Computer, mit dem die Seite betrachtet wird, installiert ist, wird der nächstfolgende probiert, bis ein vorhandener bzw. passender Zeichensatz gefunden ist.

Wenn man die Zeichensätze für seine Webseite auflistet, beginnt man im Allgemeinen mit dem bevorzugten Zeichensatz und ergänzt dann ein paar Alternativen.

Ein Beispiel für solch eine Liste der Zeichensätze könnte wie folgt aussehen:

```
h1 {
  font-family: arial, verdana, sans-serif;
}
h2 {
  font-family: "Times New Roman", serif;
}
```

Überschriften, die mit `<h1>` markiert wurden, werden in der Schrift "Arial" angezeigt. Falls dieser Zeichensatz nicht auf dem Computer des Nutzers installiert ist, wird stattdessen "Verdana" verwendet. Sollten beide Schriften nicht verfügbar sein, wird ein Zeichensatz aus der sans-serif-Familie für die Überschriften benutzt.

Bitte beachten Sie, dass Schriftartnamen, die Leerzeichen enthalten, in Anführungsstrichen geschrieben werden müssen. Siehe "Times New Roman" im obigen Beispiel.

### b) Schriftstil: `font-style`

Die Eigenschaft `font-style` weist dem betreffenden Element die Stile **normal** (kein Stil), **italic** (kursiv) oder **oblique** (schräg) zu. Im folgenden Beispiel werden alle Überschriften, die mit `<h2>` markiert sind, kursiv geschrieben.

```
h2 {
  font-family: "Times New Roman", serif;
  font-style: italic;
}
```

Obwohl **italic** und **oblique** annähernd identisch sind, ist der Wert **italic** dem Wert **oblique** vorzuziehen, der der Browser automatisch auf **oblique** umschaltet, sollte die Schriftart **italic** beim Client nicht verfügbar sein.

### c) Schriftgewichtung: `font-weight`

Mit der Eigenschaft **font-weight** gibt man an, wie fett oder "schwer" eine Schriftart angezeigt werden soll. Eine Schrift kann entweder **normal** oder **bold** (fett) dargestellt werden. Einige Browser unterstützen auch die Angabe der Zahlen 100-900 (nur volle Hunderter) um eine feiner abgestufte Darstellung zu erreichen.

```
td {
  font-family: arial, verdana, sans-serif;
  font-weight: bold;
}
```

### d) Schriftgröße: `font-size`

Die Schriftgröße wird durch die Eigenschaft **font-size** festgelegt.



Man kann viele verschiedene Einheiten (z.B. Pixel und Prozentwerte) verwenden um die Größen zu beschreiben. Beispiele sind:

```
h1 {
  font-size: 30px;
}
h2 {
  font-size: 12cm;
}
h3 {
  font-size: 120%;
}
p {
  font-size: 1em;
}
```

Es gibt einen grundlegenden Unterschied zwischen den vier obigen Einheiten. **Px**, **cm** und **pt** bringen die Schrift auf eine absolute Größe, während es **%** und **em** als relative Größenangabe dem Nutzer erlaubt, die Größe so zu verändern, dass es ihm/ihr passend erscheint. Viele Internetnutzer sind behindert, älter, sehbehindert oder 'leiden' unter der schlechten Qualität Ihres Monitors. Um Ihre Webseite für jederman barrierefrei zugänglich zu machen, sollten Sie veränderbare Einheiten, wie **%** oder **em**, verwenden.

Im LIS wird als Größeneinheit hauptsächlich em verwendet, damit die Schriftgrößen vom Anwender beeinflusst werden können.

### e) Kombination der vorgenannten: font

Mit Hilfe der **font**-Eigenschaft können Sie alle Schrift-Eigenschaften in dieser einen Eigenschaft vereinen.

Stellen Sie sich z.B. die folgenden vier Codezeilen vor, die die Schrifteigenschaften für <p> angeben:

```
p {
  font-style: italic;
  font-weight: bold;
  font-size: 30px;
  font-family: arial, sans-serif;
}
```

Wenn man diese Kombination verwendet, kann der Code vereinfacht werden:

```
p {
  font: italic bold 30px arial, sans-serif;
}
```

Die Reihenfolge der Werte für font ist:

**[font-style] | [font-weight] | [font-size] | [font-family]**

#### Aufgabe

Erstellen Sie eine Datei 4\_beispiel.html und 4\_style.css.

In der html-Datei erstellen Sie eine Überschrift in h1, eine in h2, darunter einen beliebigen Text.

Formatieren Sie die h1-Überschrift in Times New Roman in einer Größe von 30px mit einem hell-roten Hintergrund.

Formatieren Sie die h2-Überschrift in Arial, kursiv und einer relativen Größe von 1.2em.

## 6) Textgestaltung

Für die Gestaltung von Texten gibt es noch weitere Möglichkeiten. In diesem Kapitel werden die wichtigsten, die auch im LIS zu Einsatz kommen, erläutert.

## a) Textausrichtung: text-align

Die CSS-Eigenschaft **text-align** ist mit dem Attribut **align** vergleichbar, welches in älteren HTML-Versionen verwendet wurde. Text kann entweder **left (links)**, **right (rechts)** oder **center (zentriert)** ausgerichtet werden. Außerdem steht der Wert **justify** für **Blocksatz** zur Verfügung, d.h. jede Zeile wird so gestreckt, dass alle gleich lang sind. Dies kennen Sie vielleicht aus Zeitungen und Zeitschriften.

Im unten stehenden Beispiel ist der Text in Tabellenköpfen <th> (table header) rechts ausgerichtet, während er in normalen Tabellenzellen <td> zentriert wird. Zusätzlich werden normale Absätze im Blocksatz geschrieben:

```
th {
  text-align: right;
}

td {
  text-align: center;
}

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

## b) Textdekoration: text-decoration

Mit der Eigenschaft **text-decoration** ist es möglich, Texte verschiedenartig zu "dekorieren" oder Ihnen "Effekte" zu verleihen. Sie können Texte unterstreichen (**underline**), durchstreichen (**line-through**) oder eine Linie darüber (**overline**) zeichnen lassen. Um die Formatierung zu entfernen, verwenden Sie den Wert **none**.

Im folgenden Beispiel sind <h1>-Überschriften unterstrichen, <h2>-Überschriften haben eine Linie über dem Text und <h3>-Überschriften sind durchgestrichen.

```
h1 {
  text-decoration: underline;
}

h2 {
  text-decoration: overline;
}

h3 {
  text-decoration: line-through;
}
```

Mit der Formatierung **underline** sollte sehr sparsam umgegangen werden, da unterstrichene Texte traditionell als Links wahrgenommen werden. Daher ist es geschickter, Text, der betont werden soll, in farbiger Schrift oder farbig hinterlegt zu gestalten.

### Aufgabe

Erstellen Sie eine Datei 5\_beispiel.html und 5\_style.css.

Erstellen Sie eine Tabelle (2x2) und füllen sie mit beliebigem Text. Formatieren Sie die erste Spalte linksbündig, die zweite Spalte rechtsbündig. Streichen Sie mit Hilfe eines span-Tags ein beliebiges Wort in Ihrem Text durch.

## 7) Links

Links oder aus Verweise sind spezielle Tags im Hinblick auf die CSS-Formatierung. Neu ist hier, dass CSS es Ihnen erlaubt, dies abhängig davon zu gestalten, ob der Link schon besucht (**visited**) oder noch nicht besucht (**link**) wurde, gerade angeklickt wird (**active**) oder nur der Mauszeiger über dem Link platziert ist (**hover**).

Dadurch kann man seiner Webseite tolle und nützliche Effekte hinzufügen. Um diese Effekte zu steuern, benutzt man sog. Pseudoklassen.

Eine Pseudoklasse erlaubt es Ihnen zwischen verschiedenen Zuständen oder Ereignissen zu unterscheiden, wenn Sie eine Eigenschaft für einen HTML-Tag festlegen.

Sehen wir uns ein Beispiel an. Wie Sie wissen, werden Links in HTML mit `<a>`-Tags markiert. In CSS nutzen wir wiederum `a` zum ansprechen der Links im HTML-Dokument:

```
a {
  color: blue;
}
```

Ein Link kann verschiedene Zustände haben. Er kann z.B. besucht oder noch nicht besucht worden sein. Sie können die folgenden Pseudoklassen verwenden, um dies darzustellen:

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

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

Die Pseudoklassen `link`, `visited`, `active` und `hover` werden durch Doppelpunkt getrennt an den Tag-selector angehängt.

#### a) Die Pseudoklasse `:link`

Die Pseudoklasse `:link` wird für Links zu Seiten benutzt, die der Nutzer vorher noch nicht besucht hat.

```
a:link {
  color: #6699CC;
}
```

#### b) Die Pseudoklasse `:visited`

Die Pseudoklasse `:visited` wird für Links zu bereits besuchten Seiten benutzt. Der nachfolgende Beispielcode färbt alle besuchten Links dunkelviolett:

```
a:visited {
  color: #660099;
}
```

#### c) Die Pseudoklasse `:active`

Die Pseudoklasse `:active` benutzt man für aktive Links, also solche, die gerade angeklickt werden. Das nachfolgende Beispiel gibt den aktiven Links eine gelbe Hintergrundfarbe:

```
a:active {
  background-color: #FFFF00;
}
```

#### d) Die Pseudoklasse `:hover`

Die Pseudoklasse `:hover` wird angewendet, wenn der Mauszeiger auf einem Link platziert wird.

Das kann verwendet werden, um interessante Effekte zu kreieren. Wenn man beispielsweise möchte, dass seine Links orange und kursiv geschrieben werden, wenn der Mauszeiger sie berührt, dann sollte das CSS so aussehen:

```
a:hover {
  color: orange;
  font-style: italic;
}
```

### Aufgabe

Erstellen Sie eine Datei 6\_beispiel.html und 6\_style.css.

Tragen Sie in die html-Datei 3 Paragraphen mit dem Text Link 1, Link 2, Link 3 ein. Verlinken Sie die Texte auf ‚#‘ (<a href=‘#‘).

Formatieren Sie die Texte in Arial. 1.1em mit hell-rottem Hintergrund und dunkel-roter Schrift. Fügen Sie einen Hover-Effekt hinzu, so dass der Text kursiv und fett wird, der Hintergrund dunkel-rot und die Schrift weiß.

Was passiert mit der Formatierung, wenn man auf einen Link klickt?

## 8) Gruppierung von Elementen mittels span und div

Die Elemente **<span>** und **<div>** werden zum Gruppieren und Strukturieren eines Dokumentes benutzt und sehr oft mit den Attributen **class** und **id** verwendet.

In dieser Lektion werfen wir einen genaueren Blick auf die Benutzung von **<span>** und **<div>**, da genau diese beiden HTML-Elemente von zentraler Bedeutung für den Aufbau von Webseiten mit Hilfe von CSS ist.

### a) Gruppieren mit **<span>**

Das Element **<span>** ist ein neutrales Element, welches dem Dokument eigentlich nichts hinzufügt, d.h. kein eigens definiertes Aussehen hat. Aber zusammen mit CSS, kann **<span>** dazu benutzt werden, um Texte innerhalb eines Fließtextes zu formatieren, ohne sonstigen Einfluss auf den Textfluss zu nehmen.

Der span-Tag ist ein **Inline-Element**, das bedeutet, dass er den Textfluss nicht unterbricht und sich in die Zeile eingliedert.

Natürliche Inline-Elemente sind z.B.: span, a, b.

```
<p>Elefanten, Giraffen und auch Leoparden leben in Afrika.</p>
```

Möchte man nun die Tiere in dem Satz hervorheben ohne den Satz an sich zu formatieren, kann und sollte man dies mit Hilfe des span-Tags umsetzen, indem man den Text folgendermaßen im HTML formatiert:

#### html-Datei

```
<p>
  <span class=tier>Elefanten</span>,
  <span class=tier>Giraffen</span> und auch
  <span class=tier>Leoparden</span> leben in Afrika.
</p>
```

#### css-Datei

```
span.tier {
  color:red;
}
```

Hier könnte man theoretisch auch das id-Attribut anstelle des class-Attributs verwenden, um den span-Elementen einen Stil zuzuordnen. Aber da man in diesem Fall jedem span-Element eine eigene eindeutige ID zuordnen müsste, wäre dies ein sehr umständlicher Weg.

## b) Gruppieren mit <div>

Im Gegensatz zum **span**-Tag handelt es sich bei dem **div**-Tag um ein Block-Element. Das bedeutet, dass ein div-Tag immer eine neue Zeile beginnt und diese in der vollen Breite ausfüllt, eben als kompletter Block. Das Ende des Blocks wird durch den Ende-div-Tag **</div>** kenntlich gemacht und auch hier wird automatisch ein Zeilenumbruch eingefügt.

Div-Elemente können beliebig verschachtelt werden. Daher eignen sie sich besonders für die Strukturierung einer HTML-Seite. Mit ihrer Hilfe können die Seiten-Elemente wie Navigation, Content, Fusszeile usw. auf einer HTML-Seite in Blöcken positioniert werden.

```
<div id=navigation>
  <a href=#>Link 1</a>
  <a href=#>Link 2</a>
  <a href=#>Link 3</a>
</div>

<div id=content>
  <p>Elefanten sind scheue Tiere. Sie bewegen sich in Herden durch die Savanne.</p>
  <p>Giraffen können durch ihren langen Hals Blätter von den Bäumen fressen.</p>
</div>

<div id=footer>
  Copyright 2009
</div>
```

Durch die Vergabe von eindeutigen IDs können wir die einzelnen Block-Elemente einer HTML-Seite gezielt definieren und ihr Layout im CSS gestalten.

### Aufgabe

Erstellen Sie eine Datei 7\_beispiel.html und 7\_style.css.

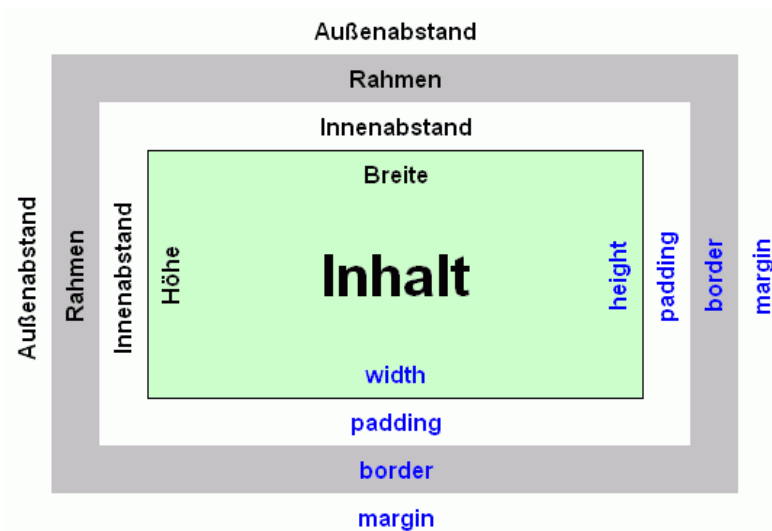
Tragen Sie Ihre komplette Büro-Anschrift mit Namen in die html-Datei ein. Hinterlegen Sie die Anschrift mit einem grauen Hintergrund. Formatieren sie Ihren Namen etwas größer als die Straße und den Ort.

Erstellen Sie einen zweiten Abschnitt mit Telefonnummer und E-Mail-Adresse. Hinterlegen Sie diesen Abschnitt mit einem hellen Gelb-Ton. Stellen Sie Ihrer Telefonnummer das Wort „Telefon“ voran und formatieren Sie es fett, kursiv und etwas kleiner als den restlichen Text. Gehen Sie entsprechend bei der E-Mail-Adresse vor.

## 9) Das Boxmodell

Das Box-Modell in CSS beschreibt die Boxen, die für HTML-Elemente generiert werden. Über das Box-Modell kann man Höhe und Breite von Inhalten, Innenabstände (padding), Ränder (border) und Außenabstände (margin) für jedes Element steuern.

Durch Addition der 4 Werte entsteht die tatsächliche Höhe und Breite eines Elements.



## 10) Breite & Höhe: width & height

Die Dimensionen eines Elements wie z.B. `p` und `div` lassen sich mit Hilfe der Eigenschaften `width` und `height` steuern.

```
div.box1{
  width:100px;
  height:100px;
}
```

Normalerweise wird die Eigenschaft `height` eher nicht verwendet, da die Höhe sich automatisch durch den Inhalt der `div`-Box ergibt. Dennoch kann es als Gestaltungs-Methode manchmal notwendig werden.

## 11) Rahmen: border

Um jedes Element kann man via CSS einen kompletten Rahmen ziehen. Für diesen Rahmen gibt es zahlreiche Gestaltungsmöglichkeiten über die diversen Eigenschaften.

Zu beachten ist, dass der Rahmen nur dann angezeigt wird, wenn alle der drei folgenden Eigenschaften definiert sind.

### a) border-width

Die Breite des Rahmens wird mit der Eigenschaft **border-width** eingestellt. Diese kann die Werte `thin` (dünn), `medium` (mittel) und `thick` (dick) oder einen numerischen Wert in Pixeln annehmen.

```
div.box2{
  border-width: 10px;
}

div.box3{
  border-width: thin;
}
```

### b) border-color

Die Eigenschaft **border-color** gibt an, welche Farbe der Rahmen hat. Werte können die normalen Farbwerte, z.B. über Hexadezimal-Code (`#123456`) oder auch über Farbnamen (`yellow`) definiert werden.

```
div.box2{
  border-color: #FF33EE;
}

div.box3{
  border-color: red;
}
```

### c) border-style

Es gibt verschiedene Umrandungsarten, aus denen Sie auswählen können: z.B. solid, dotted, dashed, double, groove, ridge, inset, outset.

Bitte beachten Sie, dass die Darstellungen in den einzelnen Browsern teilweise voneinander abweichen.

Die Werte none oder hidden können benutzt werden, wenn man keinerlei Umrandung wünscht.

```
div.box2{
  border-style: dotted;
}

div.box3{
  border-style: solid;
}
```

### d) Kombination: border

Wie auch bei vielen anderen Eigenschaften, können die Werte der Rahmeneigenschaften in der Eigenschaft border zusammengefasst werden. Schauen wir uns ein Beispiel an:

```
p {
  border-width: 1px;
  border-style: solid;
  border-color: blue;
}
```

kann kombiniert werden zu:

```
p {
  border: 1px solid blue;
}
```

Die Reihenfolge hierbei ist

**[border-width] [border-style] [border-color]**

## 12) Innenabstand: padding

Der Innenabstand **padding** kann auch als "Füllung" verstanden werden. Dies macht Sinn, da der Abstand zu anderen Elementen nicht verändert wird, sondern nur den inneren Abstand vom Rahmen (border) zum Inhalt des Elementes angibt.

```
h1 {
  background: yellow;
  padding: 10px;
}
```

## 13) Aussenabstand: margin

Der Aussenabstand **margin** gibt den Abstand vom Rahmen eines Elements bis zum nächsten Element an. So kann man pixelgenau Abstände definieren.

```
h1 {
  background: yellow;
```

```
padding: 10px;
margin: 20px;
}
```

## 14) Seitenbezogene Abstände

Die genannten Abstände des Boxmodells können auch seitenbezogen definiert werden, wenn man beispielsweise den linken Rand größer gestalten möchte als die rechte Seite.

Dazu kann man bei margin, border und padding jeweils den Zusatz

- -top
- -right
- -bottom und
- -left

verwenden.

```
h1 {
padding-top: 10px;
padding-right: 5px;
padding-bottom: 10px;
padding-left: 5px;
}
```

In der Kurzschreib-Variante kann man diese 4 Werte auch durch Leerzeichen getrennt aneinanderhängen. Dabei ist die Reihenfolge top, right, bottom und left, als im Uhrzeigersinn zu beachten.

```
h1 {
padding: 10px 5px 10px 5px;
}
```

### Aufgabe

Erstellen Sie eine Datei 8\_beispiel.html und 8\_style.css.

Formatieren Sie die beiden Texte wie in der Abbildung:

Aufgabe1: Der gepunktete Rahmen ist der Bezugsrahmen

Dies ist ein Beispiel-Text zu Aufgabe 1

Aufgabe2: Der gepunktete Rahmen ist der Bezugsrahmen

Dies ist ein Beispiel-Text zu Aufgabe 2

## 15) Allgemeine Hinweise und Informationen

Leider ist es noch immer so, dass nicht alle Browser einheitlich mit CSS-Formatierungen umgehen. Deshalb ist es immer ratsam, CSS-Formatierungen nach Möglichkeit in den aktuellen Versionen des Internet-Explorer, Firefox, Netscape, Opera und ggf. Safari (Linux) zu testen.

Die CSS-Schulung beinhaltet nur die wichtigsten CSS-Eigenschaften und Grundlagen zu CSS. CSS ist wesentlich mächtiger und umfangreicher als in der kurzen Zeit behandelt werden kann. Zur selbständigen Vertiefung des Themas gibt es zahlreiche Seiten im Internet, die sich mit dem Thema CSS beschäftigen.

Eine Auswahl sei hier genannt:

- <http://de.html.net/tutorials/css/>
- <http://www.css4you.de/>



<http://de.selfhtml.org/css/index.htm>

Manchmal kann es auch einfach reichen, z.B. „css kursiv“ in das Google Suchfeld einzugeben, um die gewünschten Formatierungs-Eigenschaften ausfindig zu machen.

# TEIL B: CSS IM LOKALEN INFORMATIONSSYSTEM

## 16) Vorbereitungen - Entwicklertools

Um das CSS einer bestehenden Seite zu bearbeiten, ist es zunächst einmal wichtig, sich einen Überblick über die Struktur der Seite zu verschaffen.

Dazu gibt es glücklicherweise verschiedene Tools, die den Seiten-Überblick erleichtern.

### a) Entwicklertools im Internet Explorer 8

Im Internet Explorer 8 sind von Haus aus die sogenannten Entwicklertools enthalten. Diese können mit F12 aktiviert und deaktiviert werden.

### b) Internet Explorer Developer Toolbar im Internet Explorer 7

In der älteren Version des Internet Explorer kann man sich die Internet Explorer Developer Toolbar von der Microsoft-Homepage nachinstallieren. Diese ist nicht ganz so komfortabel wie die Entwicklertools im IE 8, aber für das Auslesen der CSS-Struktur reichen die Funktionen allemal.

<http://www.microsoft.com/downloads/details.aspx?FamilyID=E59C3964-672D-4511-BB3E-2D5E1DB91038&displaylang=en>

### c) Plugin Firebug im Firefox

Für den Firefox gibt es das kostenlose Plugin „Firebug“, das man sich von der Mozilla-Seite herunterladen kann. Dies ist ebenfalls ein mächtiges und umfangreiches Werkzeug, um Internetseiten zu debuggen.

<https://addons.mozilla.org/en-US/firefox/addon/1843>

## Aufgabe

Betrachten Sie sich die Seite <http://www.rhoen-saale.net> und finden Sie mit einem der Entwickler-Tools heraus, welche CSS-ID der Menüpunkt „Über uns“ hat.

## 17) Offline testen

Bei Internetseiten ist es immer sehr unschön, wenn man an Online-System herum testet. Man kann nicht einfach mal was ausprobieren und muss extrem vorsichtig sein, um nichts kaputt zu machen. Denn jeder Fehler wäre sofort online und würde ein unschönes Bild für den Seitenbesucher abgeben.

Aus diesem Grund, ist es sinnvoll, größere Layout-Änderungen offline am lokalen PC durchzuführen. Dazu gehen wir folgendermaßen vor.

### a) Quelltext speichern

Auf der gewünschten Internetseite klicken wir mit der rechten Maustaste auf „Quelltext anzeigen“. Es öffnet sich eine Textdatei, die wir dann mit der Endung .html auf unserem lokalen Rechner abspeichern.

Die lokale html-Seite öffnen wir nun per Doppelklick im Internet Explorer.

Was stellen wir fest?

### b) Bilder und Dateien über base-Tag nachladen

Um die fehlenden Bilder und css-Dateien vom Internet-Server auf unserem lokalen Rechner nachzuladen, ergänzen wir im Head-Bereich den base-Tag:

```
<base href="http://www.rhoen-saale.net">
```

Diese Zeile führt dazu, dass der Browser externe Dateien nicht auf unserem lokalen PC sucht, sondern diese von der angegebenen Internet-Seite nachlädt.  
Nun sieht die Seite doch schon wie original aus.

### c) Lokales CSS verwenden

Nur haben wir nun das Problem, dass wir ja unser CSS lokal einbinden möchten und nicht das CSS vom Server verwenden möchten. Daher ändern wir eine weitere Zeile in der html-Datei ab:

```
<link rel="stylesheet" type="text/css" href="/css/LKRG/1/layout.css" media="screen">
```

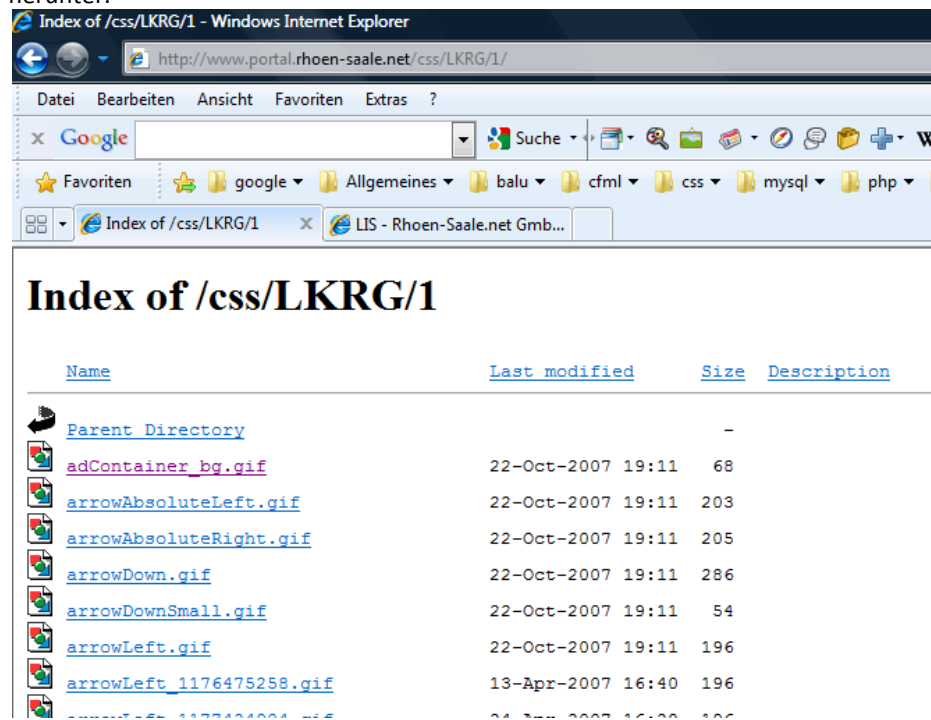
wird geändert in ...

```
<link rel="stylesheet" type="text/css" href="C:\[lokaler Pfad]\lis.css" media="screen">
```

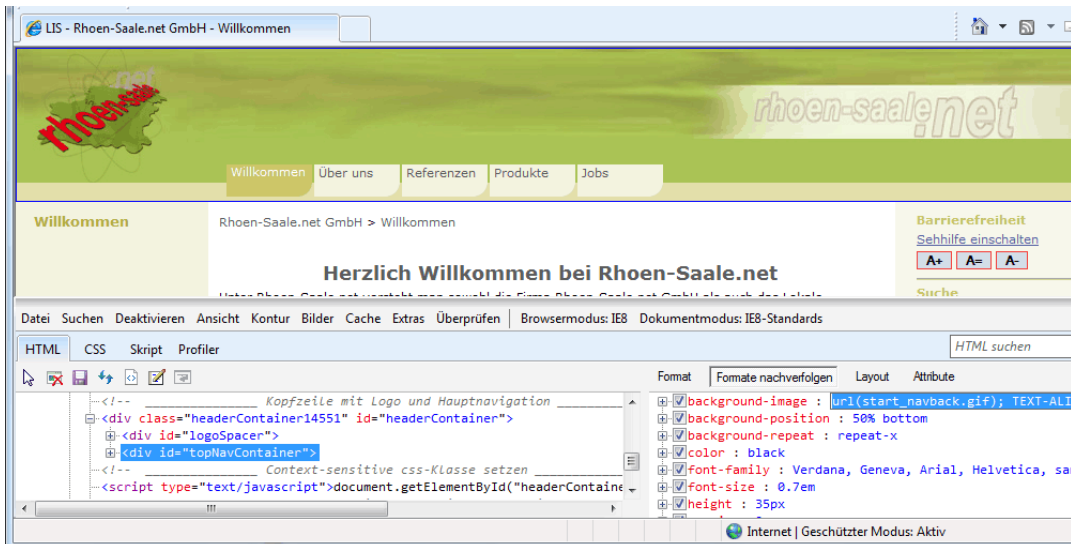
Wenn wir nun die Datei im Browser starten, stellen wir fest, dass die Formatierungen vorhanden sind, aber die im CSS integrierten Bilder nicht dargestellt werden. Woran liegt das?

### d) CSS-Bilder nachladen

Also müssen wir die im CSS referenzierten Bilder ebenfalls noch lokal speichern. Ein Blick in unser lis.css zeigt uns, dass die Bilder im selben Verzeichnis liegen wie die css-Dateien. Daher rufen wir das Verzeichnis im Browser auf und laden uns die notwendigen Bilder für die Darstellung der Seite herunter.



Welche Bilder notwendig sind, können wir mit Hilfe der Developer-Toolbar auf der LIS-Seite feststellen. Markieren Sie über die Toolbar das entsprechende Box-Element an, das ein Hintergrund-Bild enthält und lassen Sie sich in der Developer-Toolbar anzeigen, welche Bild-URL an dieser Stelle eingebunden ist.



Nachdem wir die notwendigen Bilder in das Verzeichnis einkopiert haben, sind die Vorbereitungen abgeschlossen und wir können mit der Bearbeitung des CSS beginnen.

### Aufgabe

Welche Schriftart wird im LIS bevorzugt verwendet?

Wie lautet die ID der div-Box, in der die Funktionen zur Barrierefreiheit enthalten sind?

## 18) CSS bearbeiten

Nun können Sie beginnen, Änderungen an Ihrer lokalen CSS-Datei vorzunehmen. Dazu suchen wir uns am einfachsten über die Developer-Toolbar das Element heraus, was wir bearbeiten möchten und ermitteln den betreffenden Tag oder die css-Klasse oder die ID. Über eines der drei Elemente versuchen wir dann die entsprechenden Formatierungen in der CSS-Datei ausfindig zu machen. Und diese dann zu bearbeiten oder ggf. neu anzulegen.

Wenn wir beispielsweise die Überschriften in dunkelrot darstellen möchten, ändern wir diese Angabe im CSS:

### Aktuelle Version im LIS

```
h1{
  font-size:1.6em;
  margin:30px 0 5px 0;
  color: #5D5D5D;
  text-align: center;
}
```

### Neue, bearbeitete Version

```
h1{
  font-size:1.6em;
  margin:30px 0 5px 0;
  color: #AA0000;
  text-align: center;
}
```

Wenn wir die Links in der Brotkrumen-Navigation fett darstellen möchten, dann können wir auch diese Angabe im CSS ändern bzw. eine neue Angabe hinzufügen:

### Aktuelle Version im LIS

```
div.breadCrumbBox a{
  font-size:0.95em;
  text-decoration:none;
  color:#5D5D5D;
}
```

### Neue, bearbeitete Version

```
div.breadCrumbBox a{
  font-size:0.95em;
  text-decoration:none;
  color:#5D5D5D;
  font-weight:bold;
}
```

### Aufgabe

Ersetzen Sie die Hintergrundgrafik im Kopfbereich durch eine beliebige neue Grafik.

## 19) Bearbeitetes CSS hochladen

Wenn Sie die Änderungen am Layout soweit abgeschlossen haben, dass Sie das neue Layout online stellen möchten, dann loggen Sie sich im LIS ein und wechseln in den **Administrationsbereich** und dort in den Unterpunkt **Layout**.

**Layoutverwaltung**

Hier können Layout-Dateien hochgeladen oder gelöscht werden.

### Layout-Dateien

Die Layout-Datei layout.css bestimmt das Aussehen Ihres Internet-Auftritts im Browser, die Layout-Datei handicapped.css bestimmt das Aussehen Ihres Internet-Auftritts ohne Layout für Seh-Behinderte Menschen, die Layout-Datei print.css bestimmt das Aussehen des Ausdrucks Ihres Internet-Auftritts. Die Dateien sollte nur von entsprechend qualifizierten Personen bearbeitet werden.

#### Neue Layout-Dateien hochladen

Normale Layout-Datei <a href="#">layout.css</a>	<input type="text"/>	Durchsuchen...
Druck-Layout-Datei <a href="#">print.css</a>	<input type="text"/>	Durchsuchen...
Barrierefrei-Layout-Datei <a href="#">handicapped.css</a>	<input type="text"/>	Durchsuchen...

#### Individuelle CSS-Dateien für die einzelnen Top-Menüs

Start & Service	<input type="text"/>	Durchsuchen...
Tourismus & Kultur <a href="#">topMenu_20124.css</a>	<input type="text"/>	Durchsuchen...
Wirtschaft & Arbeit <a href="#">topMenu_20138.css</a>	<input type="text"/>	Durchsuchen...
Landkreis & Verwaltung <a href="#">topMenu_23204.css</a>	<input type="text"/>	Durchsuchen...
Wellness & Gesundheit <a href="#">topMenu_23201.css</a>	<input type="text"/>	Durchsuchen...
Bildung & Soziales <a href="#">topMenu_23202.css</a>	<input type="text"/>	Durchsuchen...
Natur & Umwelt <a href="#">topMenu_23208.css</a>	<input type="text"/>	Durchsuchen...

### Layout-Bilder

Hier können Bilder hochgeladen werden, die über die Layout-Datei (layout.css) als grafische Elemente in das Layout integriert werden können. Erlaubt sind .jpg und.gif.

#### Neues Bild hochladen

	Dateiname ▲	Bildgröße (px)	Größe ▼	Datum ▼
<input checked="" type="checkbox"/>	<a href="#">adContainer_bq.qif</a>	780 x 1	68 Byte	24.04.2007 17:15
<input checked="" type="checkbox"/>	<a href="#">adContainer_bq_1176384448.qif</a>	780 x 1	68 Byte	12.04.2007 15:27
<input checked="" type="checkbox"/>	<a href="#">adContainer_bq_1176387628.qif</a>	780 x 1	68 Byte	12.04.2007 16:20

Hier sehen Sie, dass unterschiedliche CSS-Dateien hochgeladen werden können.

- **layout.css**: dies ist die normale Layout-Datei, die für die Gesamt-Darstellung verantwortlich ist.

- **print.css:** Diese Datei ist für das Layout des Drucks verantwortlich und sollte normalerweise nicht geändert werden. Druck-Layouts zeichnen sich dadurch aus, dass die Inhalte Schwarz-Weiß gehalten sind und nur der Content-Bereich gedruckt wird. Der aktuelle Navigationspunkt wird meist ebenfalls in die Druck-Version integriert.
- **handicapped.css:** Diese Datei ist für die barrierefreie Darstellung und sollte ebenfalls möglichst nicht bearbeitet werden. Die handicapped.css kann bei sehbehinderten Menschen aktiviert werden, damit die Schrift möglichst groß und frei von störenden Farben dargestellt werden kann.

Desweiteren gibt es noch die Möglichkeit, für die einzelnen Top-Navigations-Punkte eigene Layouts hochzuladen, um z.B. eine farbliche Trennung der einzelnen Bereiche zu realisieren. Diese css-Dateien sind nur ergänzend zum bestehenden layout.css, d.h. es müssen hier nicht noch einmal alle Definitionen enthalten sein.

Im unteren Teil der Seite können nun die entsprechenden Bilder für das Layout hochgeladen werden. Sobald ein neues Bild hochgeladen wird, wird ein Backup des alten Bildes erstellt, damit nicht aus Versehen eine Datei ungewollt überschrieben wird. Die gesamten Dateien werden hier aufgelistet und können auch durch rechten Mausklick heruntergeladen und bearbeitet werden.

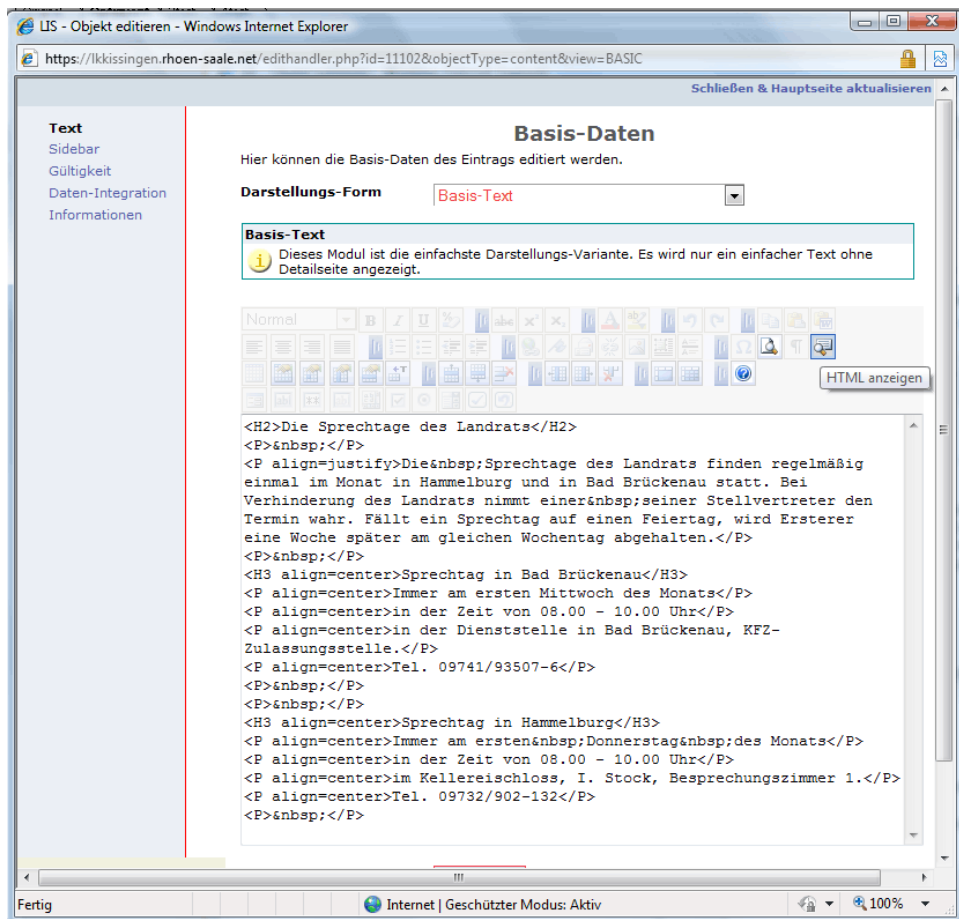
Sobald das neue Layout hochgeladen ist, kann das Ergebnis auf dem Work-Server begutachtet werden und ggf. noch einmal korrigiert werden. Erst wenn der Datenabgleich mit dem Live-Server erfolgt, ist das neue Layout online.

## 20) Layout-Änderung schnell testen

Für eine kleine Änderung des Layouts muss nicht der komplette Aufwand betrieben werden, die gesamte Seite offline zu speichern. Dafür gibt es in der Developer Toolbar ganz interessante Funktionen. Beispielsweise kann man in der Developer Toolbar die Werte einfach ändern und sieht sofort, wie sich das Layout verändert. Die Änderungen sind natürlich nur für diesen einen Aufruf und werden nicht gespeichert. Aber zum Testen einer Änderung, ist dies ein ganz nettes Feature.

## 21) Eigene Formate definieren

Für die hoch Motivierten gibt es auch noch die Möglichkeit, sich eigene Formate zu definieren. Dazu kann man im HTML-Editor im LIS auf die Quelltext-Ansicht umschalten und dem gewünschten Element beispielsweise eine eigene Klasse eintragen.



#### Aktuelle Version im LIS

```
<H2>Die Sprechstage des Landrats</H2>
```

#### Neue, bearbeitete Version

```
<H2 class="meineEigeneKlasse">Die Sprechstage des Landrats</H2>
```

Wenn Sie dieser Klasse nun einen eigenen Style im layout.css zuweisen, können Sie ganz gezielt Formatierungen vornehmen.

Sie könnten beispielsweise eine Klasse „textMarkieren“ erstellen und der Klasse die Hintergrundfarbe Gelb zuweisen. Diese Klasse könnte dann im Gesamten Internet-Auftritt verwendet werden.

## 22) Fragen, Kommentare und Anregungen

Wenn Sie noch Fragen rund um das Thema CSS haben, wäre jetzt der geeignete Zeitpunkt ;-)

Vielen Dank!

Nicole Hartmann  
Rhoen-Saale.net GmbH