

Grundlagen der Informatik 2

- Dipl.-Inf., Dipl.-Ing. (FH) Michael Wilhelm
- Hochschule Harz
- FB Automatisierung und Informatik
- mwilhelm@hs-harz.de
- Raum 2.202
- Tel. 03943 / 659 338

Inhalt

1. **HTML und CSS**
2. Formulare
3. PHP
4. PHP-OOP
5. PHP-Datenbanken
6. Unix
7. Unix Shellprogrammierung

HTML Überschriften:

<html>

<head> <title> alle sechs Überschriften </title>

</head>

<body>

<h1>Text der 1. Überschrift</h1>

<h2>Text der 2. Überschrift</h2>

<h3>Text der 3. Überschrift</h3>

<h4>Text der 4. Überschrift</h4>

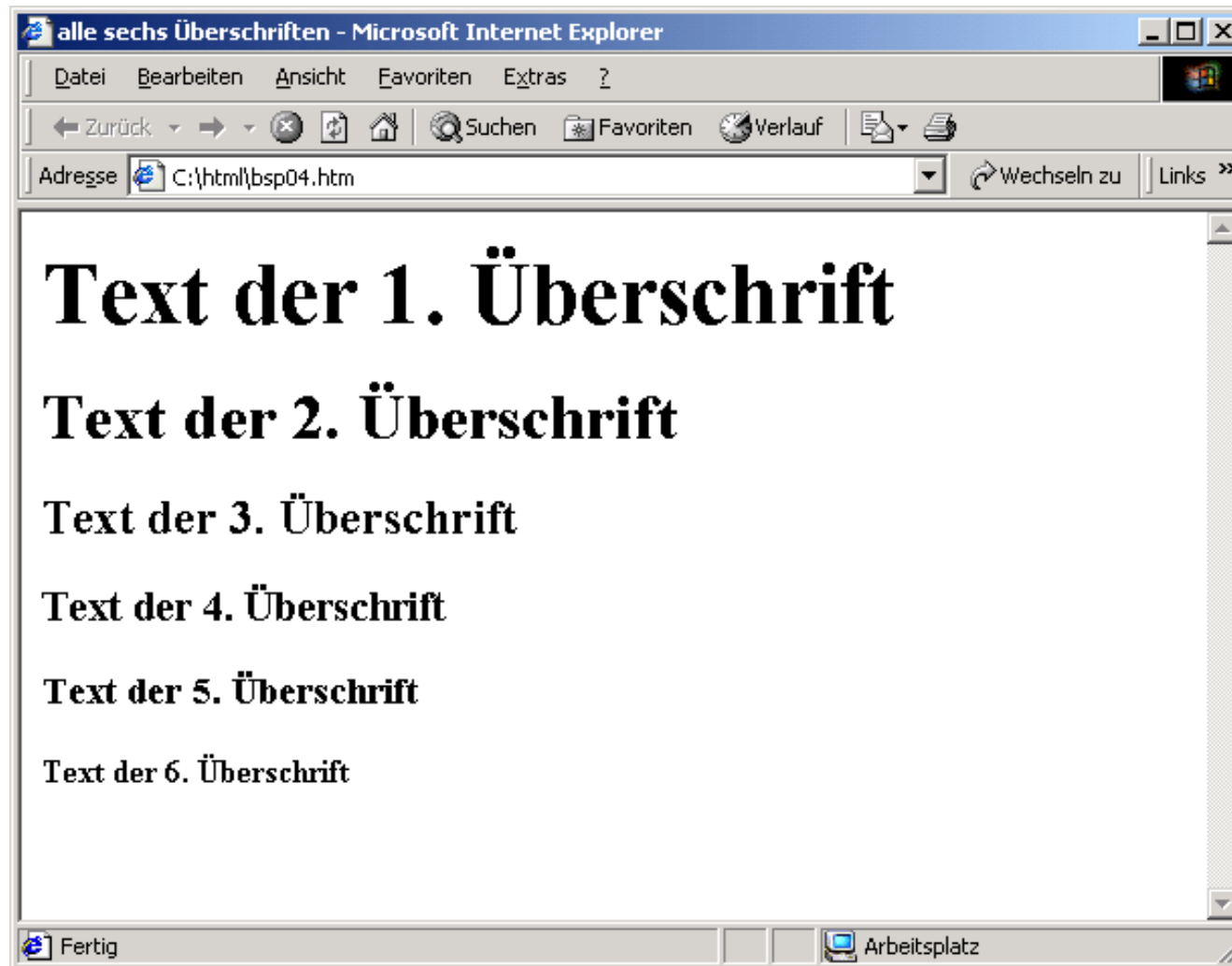
<h5>Text der 5. Überschrift</h5>

<h6>Text der 6. Überschrift</h6>

</body>

</html>

HTML Überschriften:



Sonderzeichen Unicode

 Leerzeichen, sp;
   Leerzeichen Breite n
   Leerzeichen Breite m
   Leerzeichen, schmal
 < <
 > >
 £ £ Pfundzeichen
 € € Eurozeichen
 ¢ ¢ Centzeichen
 § § Paragraphenzeichen
 © © Copyright
 ® ® eingetragene Marke
 ™ ™ Trademark-Zeichen
 ³ ³ hochgestellte 3
 ² ² hochgestellte 2
 ¹ ¹ hochgestellte 1
 ½ ½
 ¼ ¼
 ¾ ¾

μ μ Microzeichen
 α α alpha
 β β beta
 λ λ lambda (z.B. Wellenlänge)
 ω ω omega (z.B. Kreisfrequenz)
 Ω Ω Omega (z.B. Widerstand)
 π π Kreiszahl
 Πi Π
 ¶ ¶ Absatz-Zeichen
 ä ä deutsche Umlaute
 Ä Ä
 ö ö
 Ö Ö
 ü ü
 Ü Ü
 ß ß
 ± ± plusminus
 ­ Man. Trennung: shy

HTML Absätze:

<html>

<head>

<title> Absatzdefinition </title>

</head>

<body>

Dies ist die erste Zeile des Tests,

Dies ist die zweite Zeile des Tests,

<p>

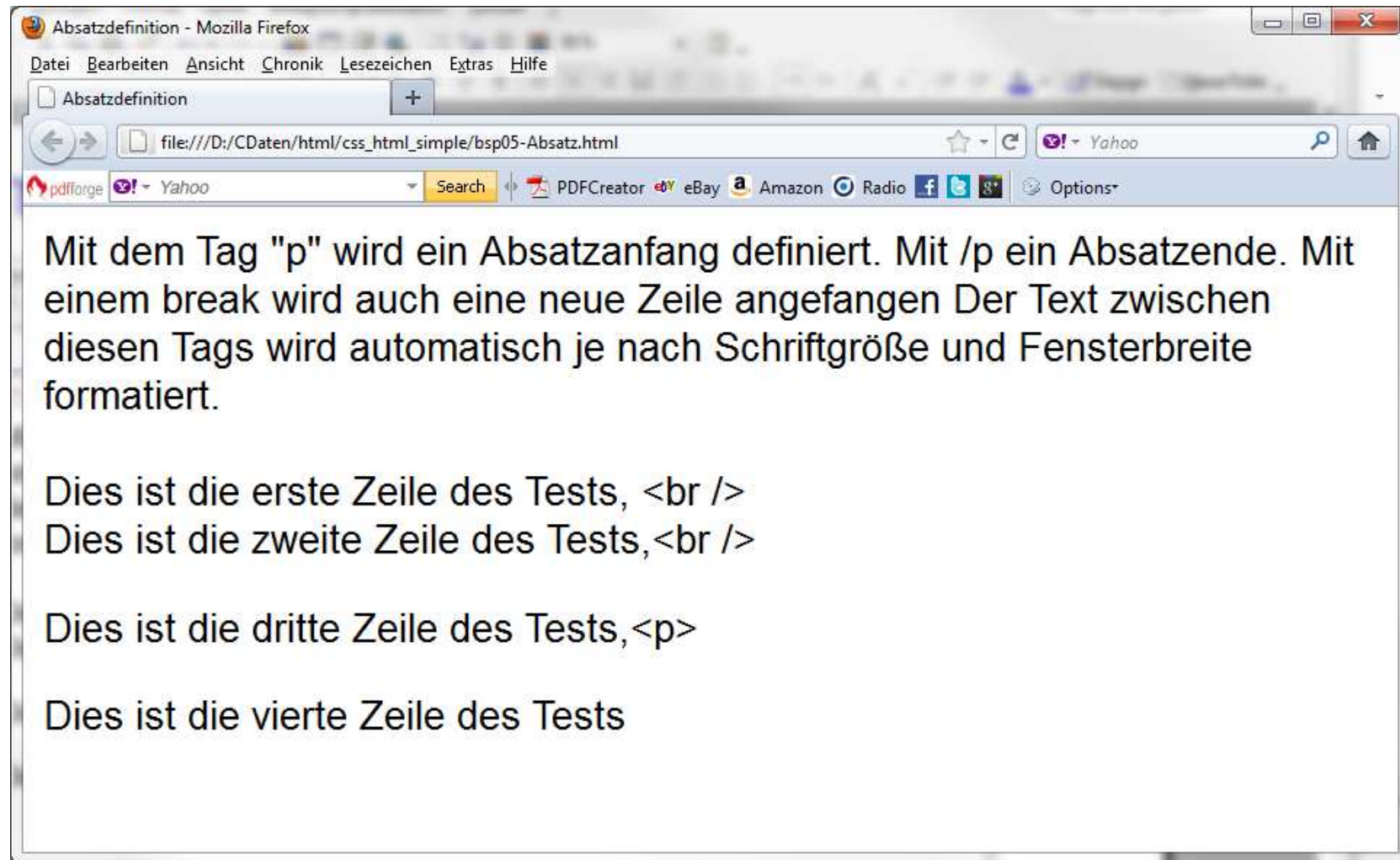
Dies ist die dritte Zeile des Tests,<p>

</p>

Dies ist die vierte Zeile des Tests </body>

</html>

HTML Absätze:



Text

<p> Text </p>: Leerzeile vorher und nachher

Cascading Stylesheets

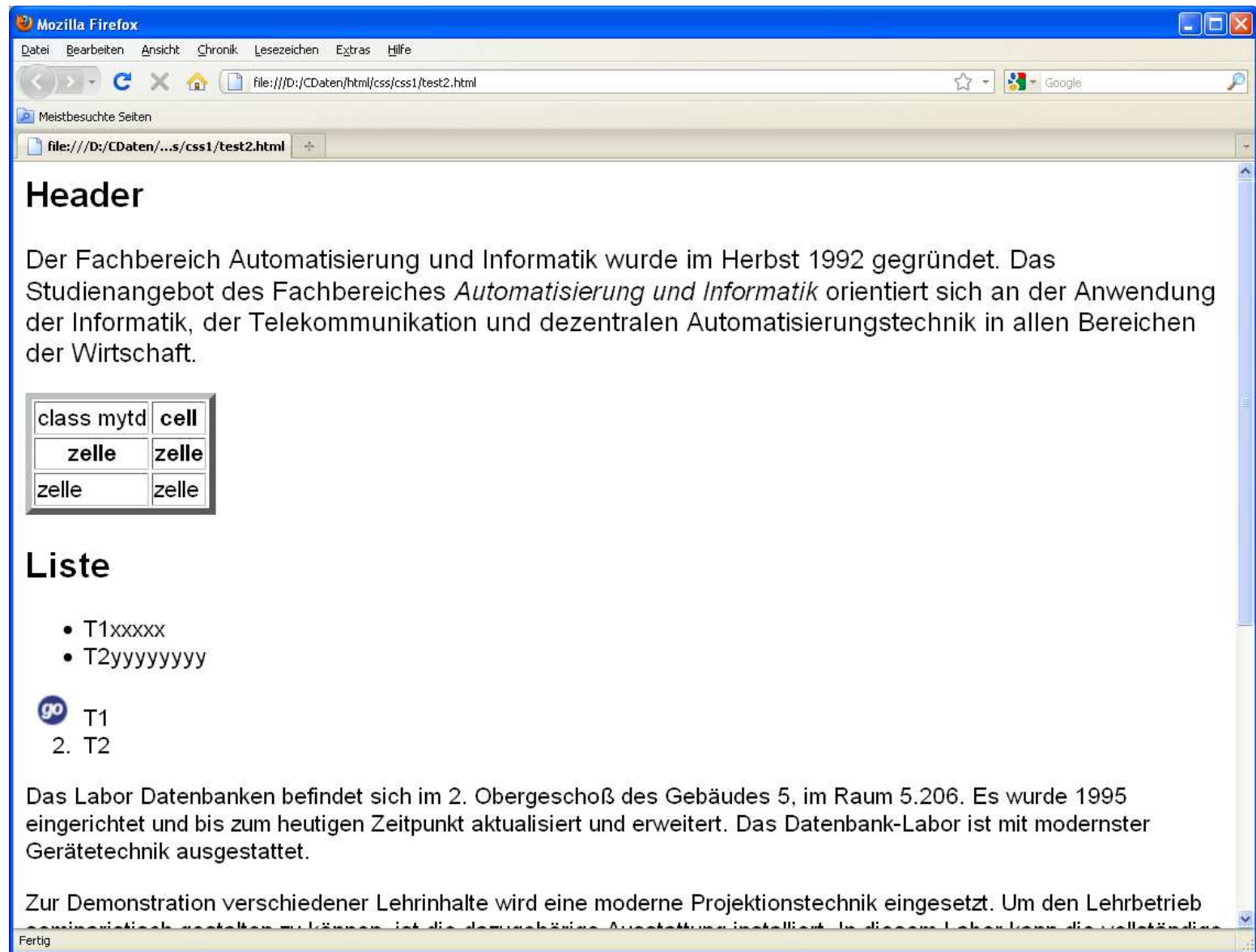
Eigenschaften von CSS:

- Ergänzung zu HTML
- Definition von Formateigenschaften von HTML-Elemente (Styles)
 - Format Überschrift
 - Abstand eines Absatzes
 - Tabellenkopf
- Eigene Hintergrundfarbe pro Absatz
- Eigener Rahmen
- Umfangreiche Styles und Maßeinheiten
- Definition von Styles mittels Ort (Cascade)
- Definition von Absätzen, auch übereinander (Sheets)
- Akustische Wiedergabe von Texten (SMIL)
- Adresse: <http://de.selfhtml.org/css/intro.htm>
<http://de.selfhtml.org/css/eigenschaften/index.htm>
<http://www.w3.org/TR/REC-CSS2/>

Einheitliches Aussehen: Firmenwebsite ohne CSS

Formate:

- Header
- Tabelle
- Listen
- Absätze





CSS-Begriffe:

- CCS steuert das Aussehen von HTML-Elementen
- Die „Konstanten“ stehen in der HTML-Datei
- Die „Konstanten“ stehen in einer externen CSS-Datei
- Der Selektor bestimmt das Aussehen
 - Selektor {
Eigenschaft1:Wert1;
Eigenschaft2:Wert2;
}

Beispiel:

```
h2 {  
    font-size: 14pt;  
    color: #0000FF;  
    font-weight: small;  
    background-color: yellow;  
}
```

CSS-Referenz:

Messeinheiten:

- in inch
- cm centimeter
- mm millimeter
- em Höhe des aktuellen Fonts
- ex Höhe des Buchstaben „x“ des aktuellen Fonts
- pt point, 1/72 in, entspricht 0,35278 mm
- pc pica, entspricht 12 points: 4,23 mm
- px Pixel

Farbeinheiten:

- #rrggbb Hexadezimale Darstellung
- rgb(x,y,z) Farbanteil pro Farbe, jeweils 0 bis 255
- rgb(x%,y%,z%) Farbanteil in Prozent pro Farbe, jeweils 0 bis 100%

Farbnamen:

aqua	black	blue	fuchsia	gray	green	lime	maroon
navy	olive	purple	red	silver	teal	white	yellow

CSS-Referenz:

Font

- font-family
- font-style
- font-variant
- font-weight
- font-size

Text

- word-spacing
- letter-spacing
- text-decoration
- vertical-align
- text-transform
- text-align
- text-indent
- line-height

Color Background

- color
- background-color
- background-image
- background-repeat
- background-attachment
- background-position

Classification

- display
- white-space
- list-style-type
- list-style-image
- list-style-position
- list-style

CSS-Referenz:

Font

- font-family:** Die Fonts werden nach ihrer Reihenfolge ausgewählt.
- a) feste Namen wie
Garamond, Palatino, Serif, Times, Courier New,
 - b) generische Namen
 - Serif
 - sans-serif
 - Kursiv (Italic ?)
 - Fancy
 - Monospace
- font-style:** Gibt die Stil an (normal oder kursiv)
- normal
 - italic
 - oblique// ähnlich italic

CSS-Referenz:

Font

font-size: Gibt die Größe der Schrift an. Vier Möglichkeiten

a) Absolute Größe

- | | | |
|------------|--------------|---------|
| • xx-small | x-small | small |
| • medium | large | x-large |
| • xx-large | | |
| • 18pt | 18in | 18cm |
| • 18mm | 18pc | 18px |

b) Relative Größe entsprechend des aktuellen Fonts

- smaller
- larger

c) Relative Höhe entsprechend des aktuellen Fonts, **Multiplikation**

- 1,5 em = 150%

d) Prozentwert, Verhältnis zum aktuellen Font

- 300%

CSS-Referenz:

Font

font-variant: Setzt den Text in KAPITÄLCHEN.

- normal
- small-caps

font-weight: Dicke der Buchstaben (normal, fett etc).

a) Nummern

- 100 // Dünn, lighter
- 200 bis
- 900 // Fett, bold

b) Name

- lighter // Verringerung zur Elternschrift
- normal
- bold
- bolder // Vergrößerung zur Elternschrift

CSS-Definition Beispiele:

Externes Format:

Datei: bsp1.xhtml

<head>

<title>Link mit CSS</title>

<link rel="stylesheet" type="text/css" href="../../bsp1.css">

</head>

Datei: bsp1.css

body {

background-color: red;

font-size: 1.2em;

font-family: Times;

}

CSS-Definition:

Internes Format:

<head>

<title> Test mit interner Definition</title>

<style type="text/css">

h1 {

background-color: #00FF00;

font-family: Helvetica; Arial; sans-serif;

}

</style>

<body>

...

</body>

CSS-Beispiele:

body {

font-size: 14pt;

color: rgb(20,100,100);

font-family:Arial,Garamond, Times, "Gill Sans", Serif;

font-weight:400;

}

ul{

list-style-type:circle; // square, discs, none

list-style-position:outside;

}

ol{

list-style-type:lower-alpha; // decimal, upper-roman,

// decimal-leading-zero

}

HTML Body-Attribute

■ CSS-„Attribute“ zu body sind:

- background-color *Hintergrundfarbe*
- link, alink, vlink *Farbe der Links*
- color *Textfarbe*
- background-image *Hintergrund mit einem Bild*
- background-repeat

■ Farben werden hexadezimal, in Prozent oder als 0 bis 255 eingetragen:

- #C0C0C0
- rgb(100,200,255);

■ Dabei bezeichnet je eine zweistellige hexadezimale Zahl die Intensität von rot, grün und blau (RGB-Wert)

HTML Body-Attribute

```
<style type="text/css">
```

```
body {
```

```
    color: rgb(0, 0, 255); /* Textfarbe */
```

```
    background-color: yellow;
```

```
    background-image: url("Virus.jpg");
```

```
    background-repeat: no-repeat;
```

```
    background-attachment: fixed; /* scroll fixed */
```

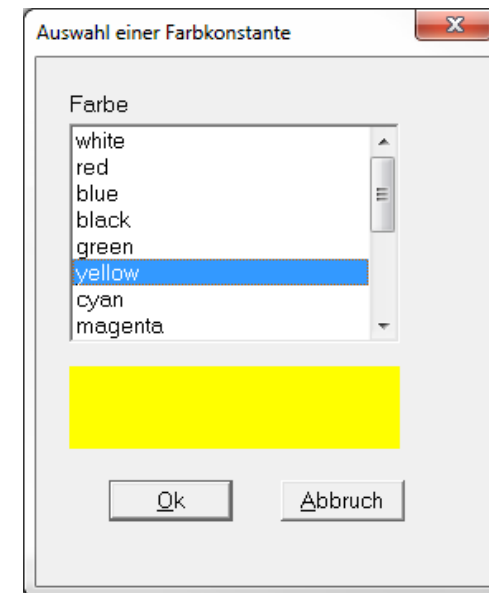
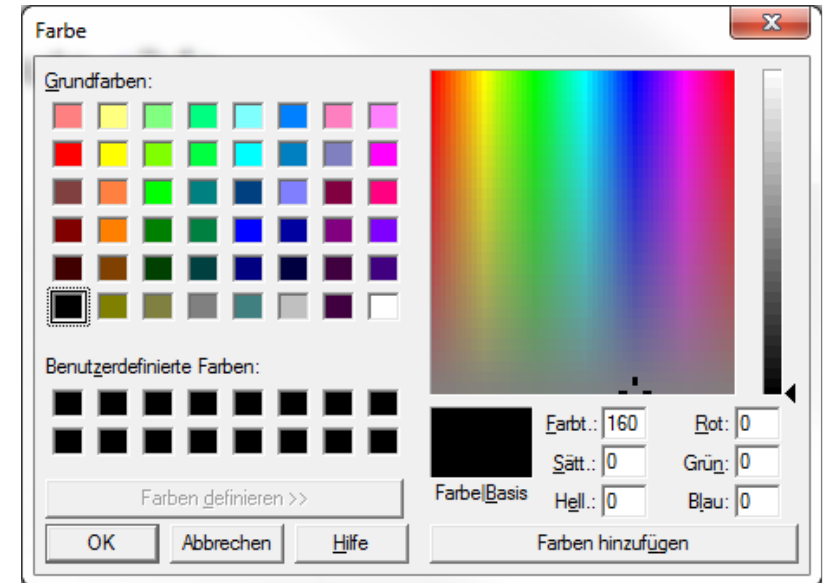
```
    background-position: top; /* left center right top bottom */
```

```
}
```

```
</style>
```

Farben in HTML

- rot/grün/blau (RGB) Angaben zum body Tag, meist hexadezimal
- #C0C0C0 für grau
- #FFFFFF für weiß
- #FF69B4 für pink
- in Angaben von 0 bis 255
- In Prozentangaben
- getrennte Farben für Hintergrund, Text, Verweise, besuchte Verweise



Websichere Farben

Firefox browser window showing the website www.css4you.de/refcolor216.html. The page title is "CSS 4 You - The Finest in Stylesheets: Websichere Farben - Mozilla Firefox". The page content includes a navigation bar with links like "Download", "Kontakt", "Forum", "Suche", and "Hilfe". The main section is titled "Websichere Farben" and contains a table of 216 color swatches arranged in a 6x6 grid. The colors are organized by hex code, ranging from #000000 to #FFFFFF. The page also features a sidebar with sections like "Übersichten", "Eigenschaften", "Workshops", "Referenzen", and "Tipps und Tricks".

Websichere Farben

Für Systeme, die nur 256 Farben anzeigen, ist diese Liste interessant. Die folgenden 216 Farben werden auf Windows- und Mac-Rechnern gleich angezeigt. Die restlichen 40 sind zwischen beiden Systemen unterschiedlich

#000000	#000033	#000066	#000099	#0000CC	#0000FF
#330000	#330033	#330066	#330099	#3300CC	#3300FF
#660000	#660033	#660066	#660099	#6600CC	#6600FF
#990000	#990033	#990066	#990099	#9900CC	#9900FF
#CC0000	#CC0033	#CC0066	#CC0099	#CC00CC	#CC00FF
#FF0000	#FF0033	#FF0066	#FF0099	#FF00CC	#FF00FF
#003300	#003333	#003366	#003399	#0033CC	#0033FF
#333300	#333333	#333366	#333399	#3333CC	#3333FF
#663300	#663333	#663366	#663399	#6633CC	#6633FF
#993300	#993333	#993366	#993399	#9933CC	#9933FF
#CC3300	#CC3333	#CC3366	#CC3399	#CC33CC	#CC33FF
#FF3300	#FF3333	#FF3366	#FF3399	#FF33CC	#FF33FF
#006600	#006633	#006666	#006699	#0066CC	#0066FF
#336600	#336633	#336666	#336699	#3366CC	#3366FF
#666600	#666633	#666666	#666699	#6666CC	#6666FF
#996600	#996633	#996666	#996699	#9966CC	#9966FF
#CC6600	#CC6633	#CC6666	#CC6699	#CC66CC	#CC66FF
#FF6600	#FF6633	#FF6666	#FF6699	#FF66CC	#FF66FF
#009900	#009933	#009966	#009999	#0099CC	#0099FF
#339900	#339933	#339966	#339999	#3399CC	#3399FF
#669900	#669933	#669966	#669999	#6699CC	#6699FF
#999900	#999933	#999966	#999999	#9999CC	#9999FF
#CC9900	#CC9933	#CC9966	#CC9999	#CC99CC	#CC99FF
#FF9900	#FF9933	#FF9966	#FF9999	#FF99CC	#FF99FF
#00CC00	#00CC33	#00CC66	#00CC99	#00CCCC	#00CCFF
#33CC00	#33CC33	#33CC66	#33CC99	#33CCCC	#33CCFF
#66CC00	#66CC33	#66CC66	#66CC99	#66CCCC	#66CCFF
#99CC00	#99CC33	#99CC66	#99CC99	#99CCCC	#99CCFF
#CCCC00	#CCCC33	#CCCC66	#CCCC99	#CCCCCC	#CCCCFF
#FFCC00	#FFCC33	#FFCC66	#FFCC99	#FFCCCC	#FFCCFF
#00FF00	#00FF33	#00FF66	#00FF99	#00FFCC	#00FFFF
#33FF00	#33FF33	#33FF66	#33FF99	#33FFCC	#33FFFF
#66FF00	#66FF33	#66FF66	#66FF99	#66FFCC	#66FFFF
#99FF00	#99FF33	#99FF66	#99FF99	#99FFCC	#99FFFF
#CCFF00	#CCFF33	#CCFF66	#CCFF99	#CCFFCC	#CCFFFF
#FFFF00	#FFFF33	#FFFF66	#FFFF99	#FFFFCC	#FFFFFF

↑ nach oben

©2004 - 2006 css4you.de
Letzte Änderung: 20.10.2006
XHTML 1.0 Strict
CSS 2.1

HTML CSS-Body-Attribute:

#rrggbb	Hexadezimale Darstellung
rgb(x,y,z)	Farbanteil pro Farbe, jeweils 0 bis 255
rgb(x%,y%,z%)	Farbanteil in Prozent pro Farbe, jeweils 0 bis 100%

body {	
background-color:#ff0000;	zeigt eine rote Farbe an.
background-color:#0000ff;	zeigt eine blaue Farbe an.
background-color:#00ff00;	zeigt eine grüne Farbe an.
}	

Alternative können auch definierte Konstanten verwendet werden.

white	black
red	blue
green	yellow
cyan	magenta
purple	navy
olive	lime
aqua	teal
silver	maroon

Die sieben HTML CSS-Schriftgrößen:

`` für die kleinste Schrift `xx-small` ``

`
`

`` für eine kleine Schrift `x-small` ``

`
`

`` für die Schrift `small` ``

`
`

`` für die Schrift `medium` ``

`
`

`` für die Schrift `large` ``

`
`

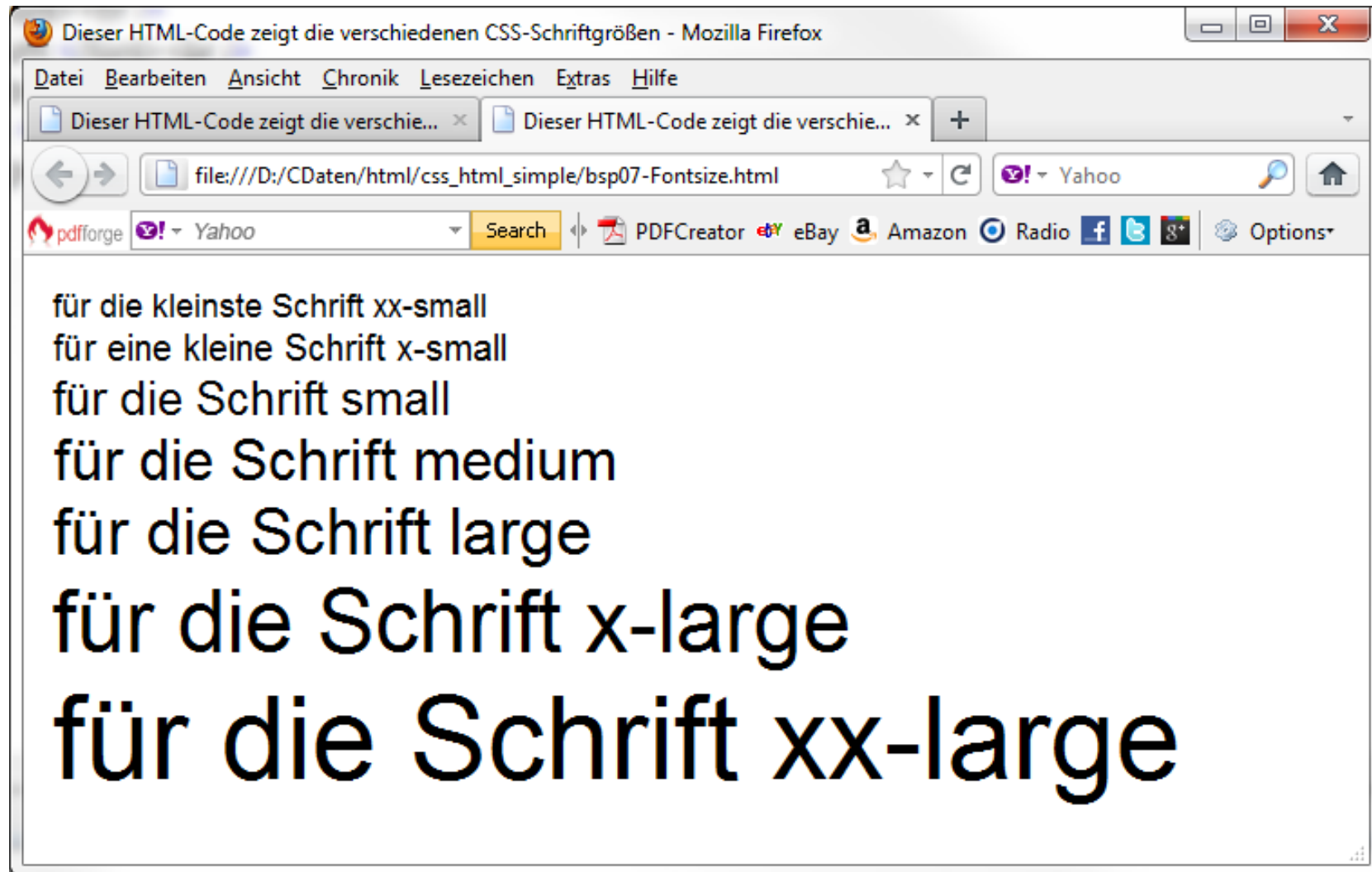
`` für die Schrift `x-large` ``

`
`

`` für die Schrift `xx-large` ``

`
`

HTML Schriften:



Text und Farbe

- Farbe kann auch für einzelne Schriften festgelegt werden:

`` für die kleinste Schrift `xx-small` ``

`` für eine kleine Schrift `x-small` ``

`` für die Schrift `small` ``

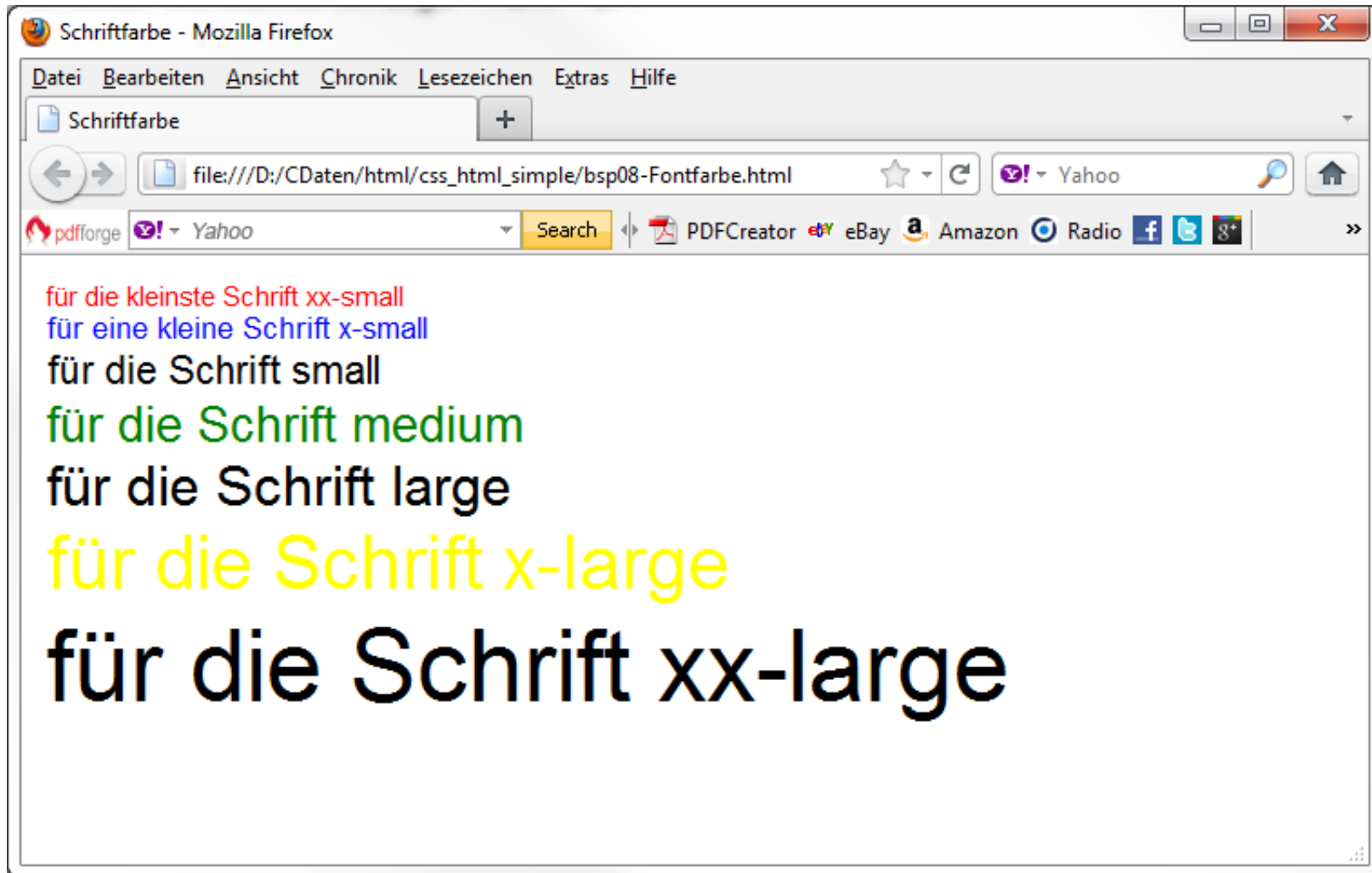
`` für die Schrift `medium` ``

`` für die Schrift `large` ``

`` für die Schrift `x-large` ``

`` für die Schrift `xx-large` ``

Schriftgrößen und Farbe in CSS



Inhalte strukturieren

- Das <div>-Tag ist ein Container für mehrere HTML-Elemente, denen durch die Kernattribute des <div>-Tags Stylesheet-Eigenschaften zugewiesen werden.
- div-Elemente sind Blockelemente, da das öffnende und das schließende <div>-Tag jeweils zu **Zeilenumbrüchen** – äquivalent zum
-Tag – führen.
- <div>-Tags lassen sich ineinander verschachteln und bilden einen leistungsfähigen Mechanismus
- Man verwendet das <div>-Tag dazu, eine Menge von logisch zusammengehörigen HTML-Elementen mit der Hilfe von Cascading Stylesheets zu formatieren, positionieren oder mit JavaScript zu animieren. Dazu werden die Kernattribute *class*, *id* und *style* benutzt.
- Äquivalent zum <div>-Tag gibt es das **-Tag**, das benutzt wird, wenn einer Gruppe von HTML-Elementen Inlinestile – also Stile ohne Zeilenumbruch – zugewiesen werden sollen.

Text und Ausrichtung

CSS-Ausrichtungs-Attribute

- text-align:left;
- text-align:center;
- text-align:right;
- text-align:justify;

```
<style type="text/css">
    .align_l {
        text-align:left;
        color:#0000ff;
    }
    .align_r {
        text-align:right;
        color:#ff0000;
    }
    .align_c {
        text-align:center;
        color:#ff00ff;
    }
    .align_j {
        text-align:justify;
        color:#00ff00;
    }
</style>
```

`<div class="align_l">`

L: Die Texte können mit dem Attribut ALIGN innerhalb eines Blockes ausgerichtet werden. Als mögliche Werte stehen "LEFT, CENTER, RIGHT, JUSTIFY" zur Verfügung.

`</div>
`

`<div class="align_c">`

C: Die Texte können mit dem Attribut ALIGN innerhalb eines Blockes ausgerichtet werden. Als mögliche Werte stehen "LEFT, CENTER, RIGHT, JUSTIFY" zur Verfügung.

`</div>
`

`<div class="align_r">`

R: Die Texte können mit dem Attribut ALIGN innerhalb eines Blockes ausgerichtet werden. Als mögliche Werte stehen "LEFT, CENTER, RIGHT, JUSTIFY" zur Verfügung.

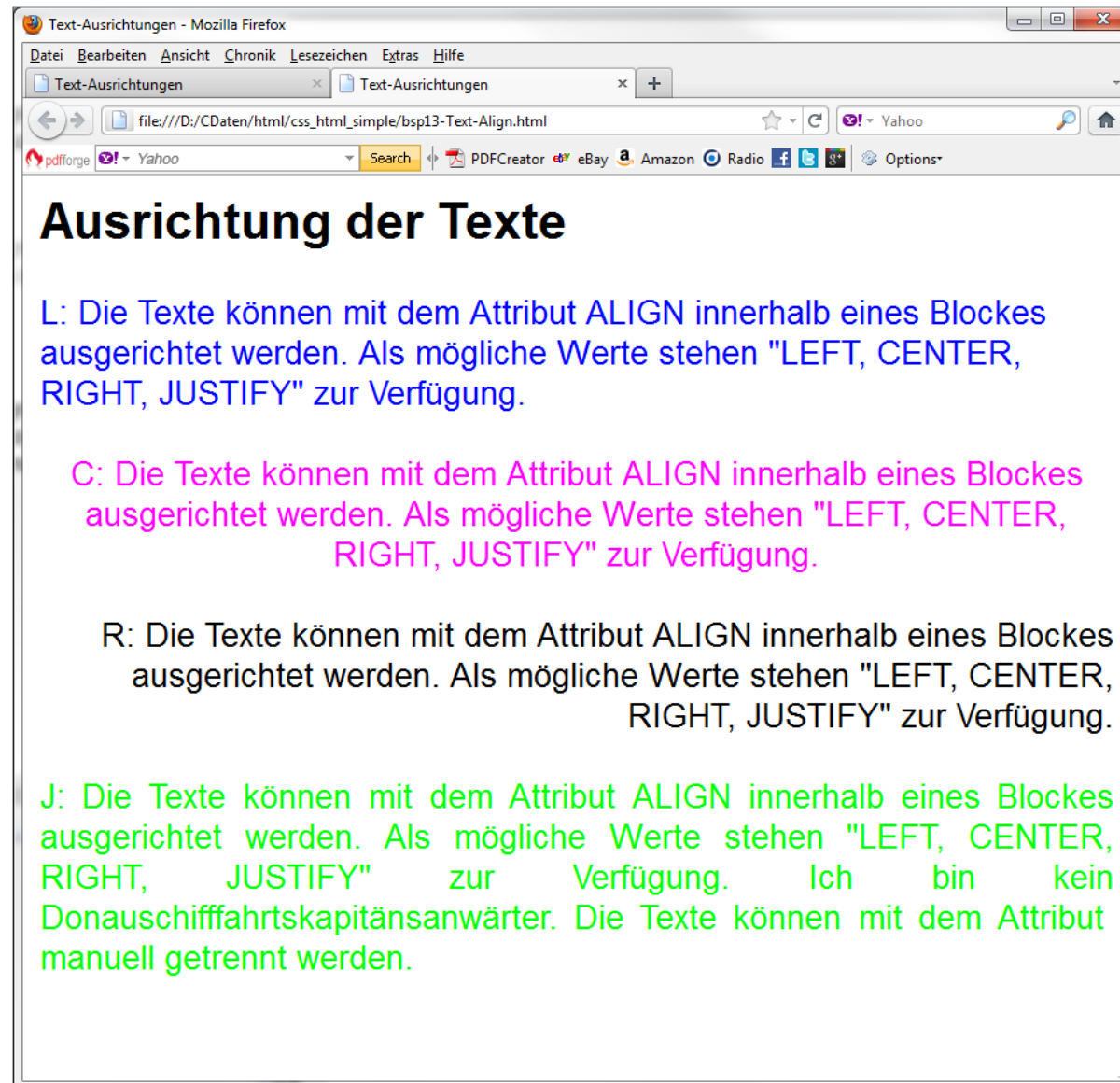
`</div>
`

`<div class="align_j">`

J: Die Texte können mit dem Attribut ALIGN innerhalb eines Blockes ausgerichtet werden. Als mögliche Werte stehen "LEFT, CENTER, RIGHT, JUSTIFY" zur Verfügung.

Ich bin kein Donauschiffahrtsskapitänsanwärter. Die Texte können mit dem Attribut `­` manuell getrennt werden.

`</div>`



Beispiel css_bsp1.xhtml

```
.div1 {  
    color:blue;  
}
```

```
p {  
    color:green;  
}
```

```
.over {  
    text-decoration: overline;  
}
```

<p>

Diese Zeile sollte in grün angezeigt werden

</p>

<div class="div1">

3. Zeile, im zweiten Beispiel, div

</div>

css_bsp1.xhtml

Beispiel css_bsp1.xhtml

```
.div1 {  
    color:blue;  
}
```

```
.over {  
    text-decoration: overline;  
}
```

```
p {  
    color:yellow;  
    background-color:blue;  
}
```

<p>

Diese Zeile sollte in grün angezeigt werden

<div class="div1">

3. Zeile, im zweiten Beispiel, div

</div>

</p>

css_bsp1.xhtml

Beispiel css_bsp1.xhtml



Font-Attribute: tt gibt es nicht mehr

```
.tt {
```

```
    font-family: "courier new",courier,monospace;
```

```
}
```

```
<body>
```

```
    iiiii mmmmm<body>
```

```
<div class="tt">
```

```
<tt>iiiiii mmmmm</tt>
```

```
</div>
```

```
<div style="font-style:italic;">
```

```
&lt;i> liefert Kursive schrift<br />
```

```
</div>
```

```
<div style="font-weight:900;"> // 100 bis 900
```

```
    Test: Nun Fett
```

```
</div>
```

Font-Attribute

`<div style="text-decoration:underline;">`

Test: Nun Unterstrichen

`</div>`

Test: ``Nun Überstrichen``

`<div style="text-decoration: line-through;">`

Test`<u>`Nun Durchgestrichen`</u>`

`</div>`

`<div style=" text-decoration: blink;">`

Test`<u>`Nun Blinkend`</u>`

`</div>`

Test ``Nun Kapitälchen``

Font-Attribute

Test H20

Test: A2+B2=C2

<div>

Test Normale Schrift:imnpjEtwas kleiner:

imnpjNoch kleiner:

imnpj

</div>

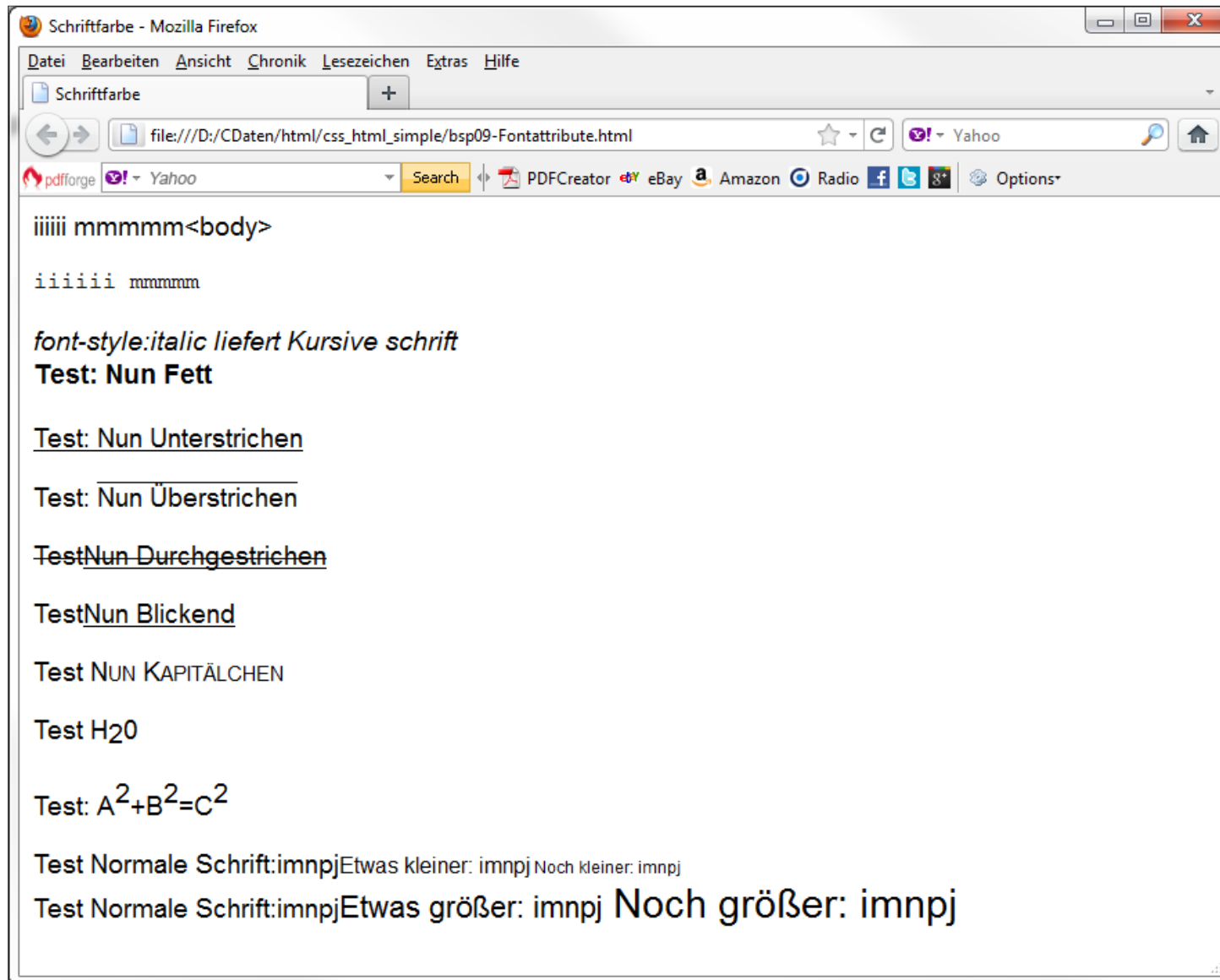
<div>

Test Normale Schrift:imnpjEtwas größer:

imnpjNoch größer:

imnpj

</div>



Horizontale Linien

In HTML können horizontale Linien auf einfache Art erzeugt werden:

Das Tag `<hr>` erlaubt die Definition einer Linie

Weitere CSS-Attribute:

- width: Pixel Länge in Pixel
- width: % Länge in Prozent
- height: Pixel Höhe in Pixel, oder ä.
- color: rgb Farbe, Randfarbe
- background-color: Hintergrundfarbe, nur sichtbar bei großer Höhe
- border 1 Wert top, left, bottom, right
- border 2 Werte top/left und bottom/right
- border 4 Wert top, left, bottom, right
- text-align left, center, right nur mit margin-right etc.

Horizontale Linien

Linie mit einer Breite von 300 px, Farbe yellow

```
<hr style="width:300px; color:yellow;" />
```

Linie mit einer Breite von 200 px, Farbe rot, Ausrichtung rechts, Höhe 5px

```
<hr style="width:300px; color:red; height:5px; margin-right:0px; text-align:right;" />
```

Linie mit einer Breite von 50%, Farbe blue, BGColor: green, Ausrichtung rechts, Höhe 15px

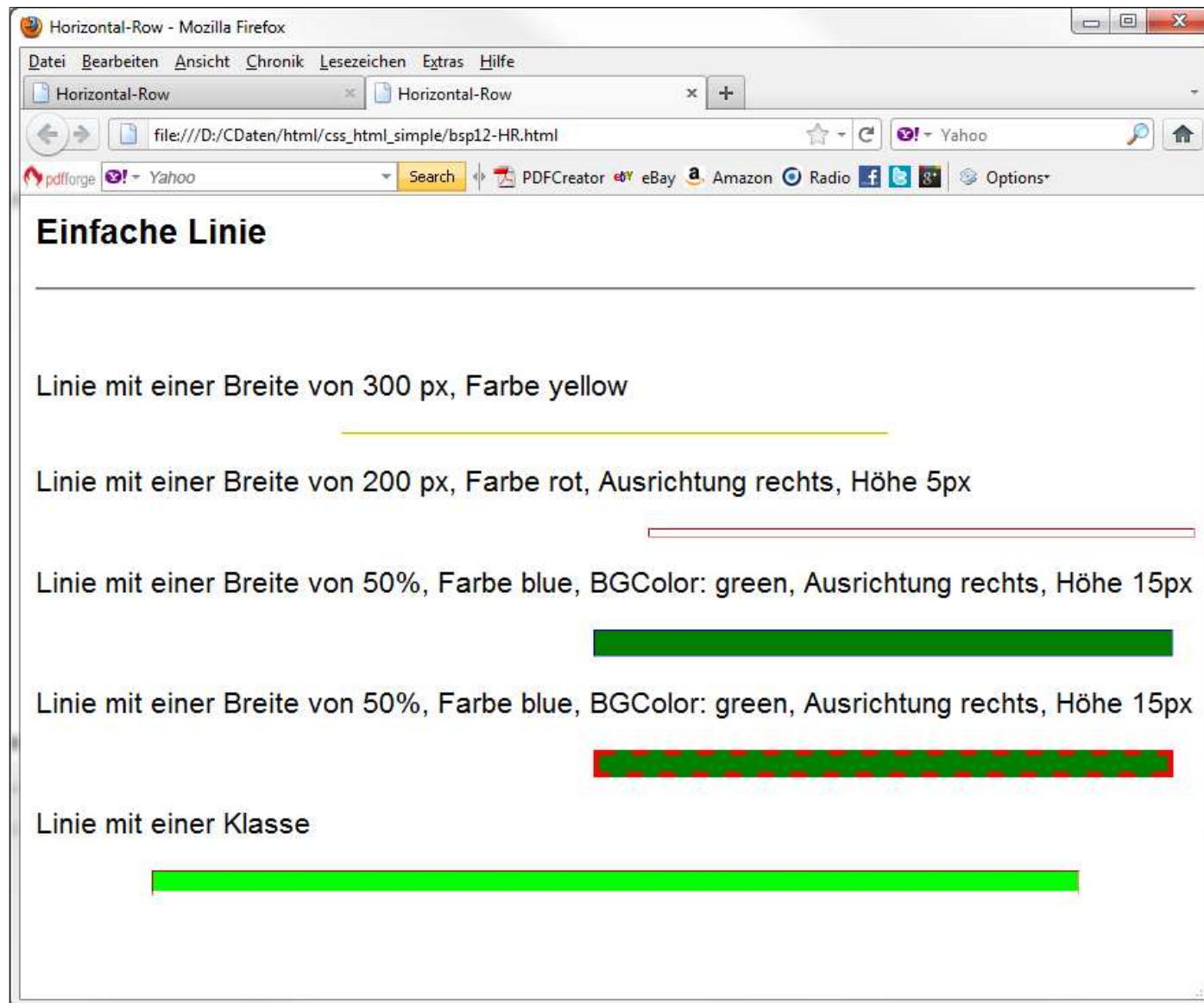
```
<hr style="width:50%; color:blue; background-color:green; height:15px; margin-right:12px; text-align:right;">
```

Linie mit einer Breite von 50%, Farbe blue, BGColor: green, Ausrichtung rechts, Höhe 15px, Border

```
<hr style="width:50%; color:blue; background-color:green; height:15px; margin-right:12px; text-align:right; border:3px dashed red;">
```

```
</p>
```

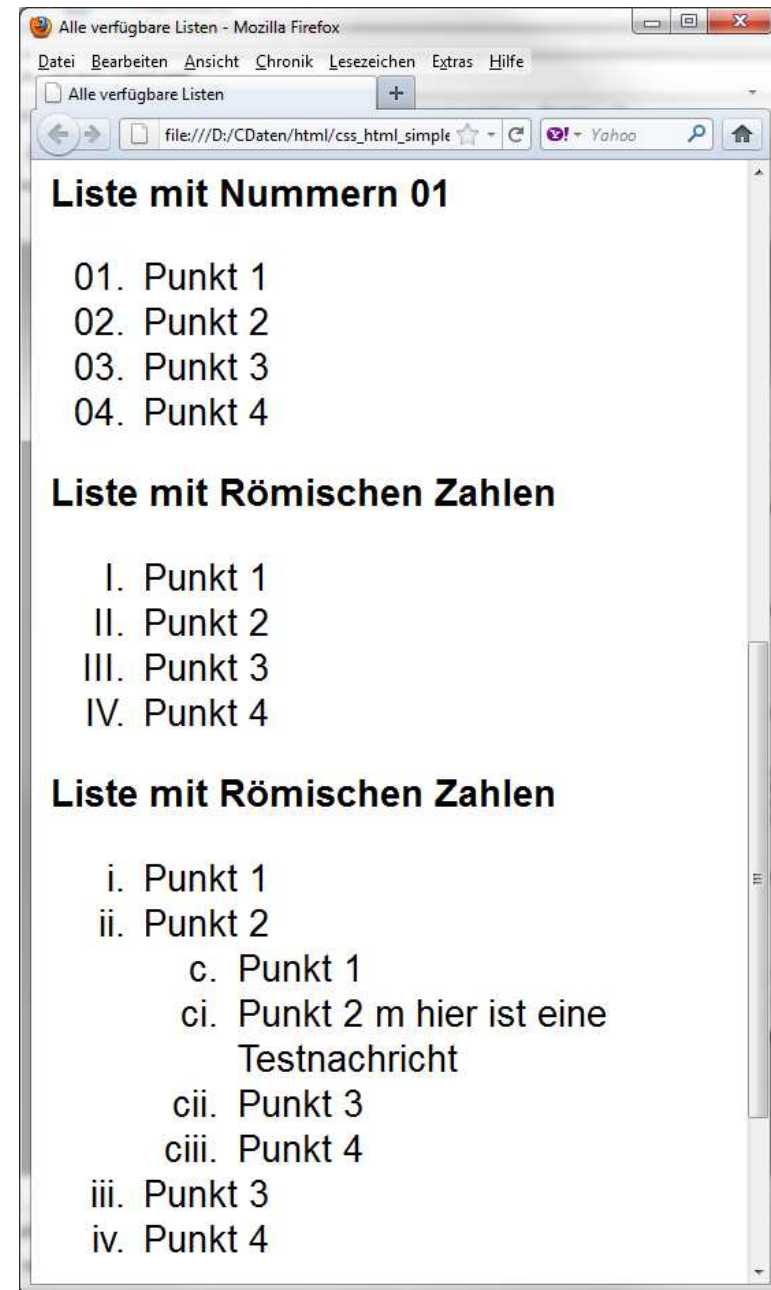
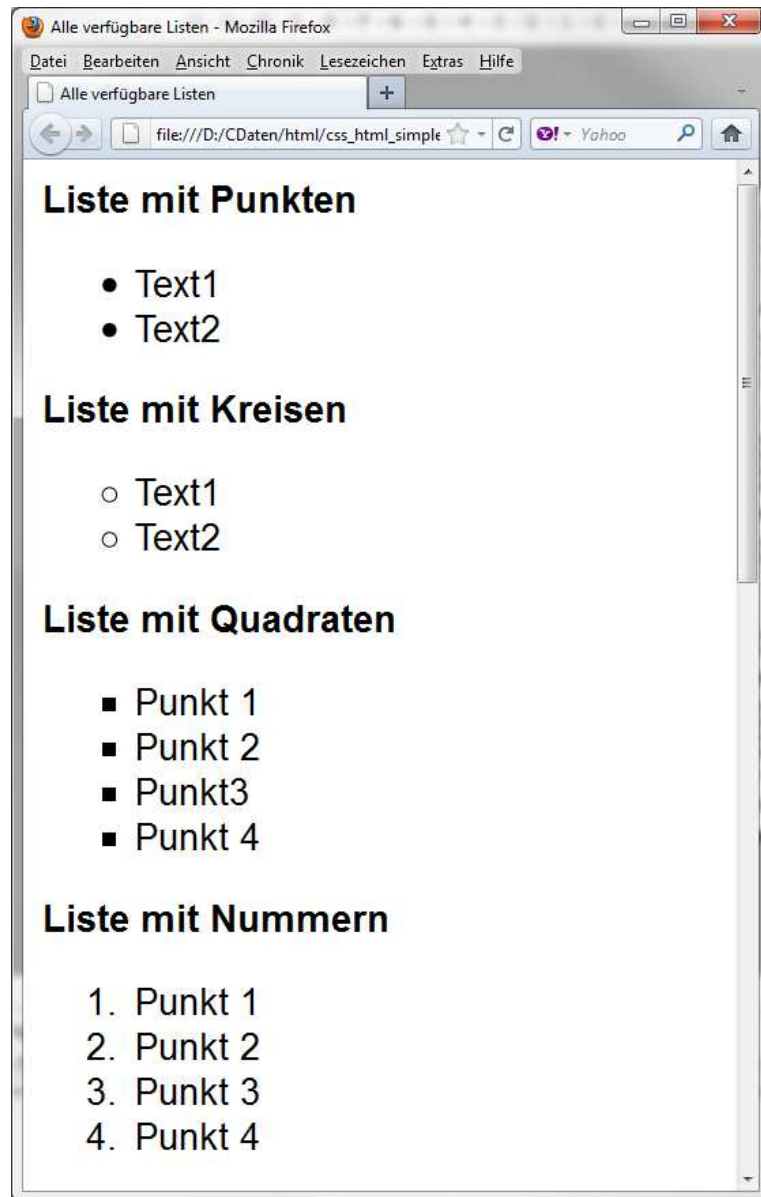
Horizontale Linien



Listen in CSS

- none Kein Aufzählungszeichen
- Circle Kreis, nur Rahmen
- square Quadrat
- disc Gefüllter Kreis
- decimal Dezimalzahlen (1. ,2. , 3. , ...)
- lower-roman Kleine römische Zahlen (i. ,ii. ,iii. , ...)
- upper-roman Grosse römische Zahlen (I. ,II. , III., ...)
- decimal-leading-zero Dezimalzahlen mit führender 0 (01. ,02. , 03. , ...)
- lower-greek Kleine griechische Nummerierung alpha, beta, gamma,...
- lower-latin Kleine Ascii-Zeichen (a. ,b., c. , ...)
- upper-latin Große Ascii-Zeichen (A., B. ,C. , ...)
- armenian Armenische Nummerierung
- georgian Georgische Nummerierung

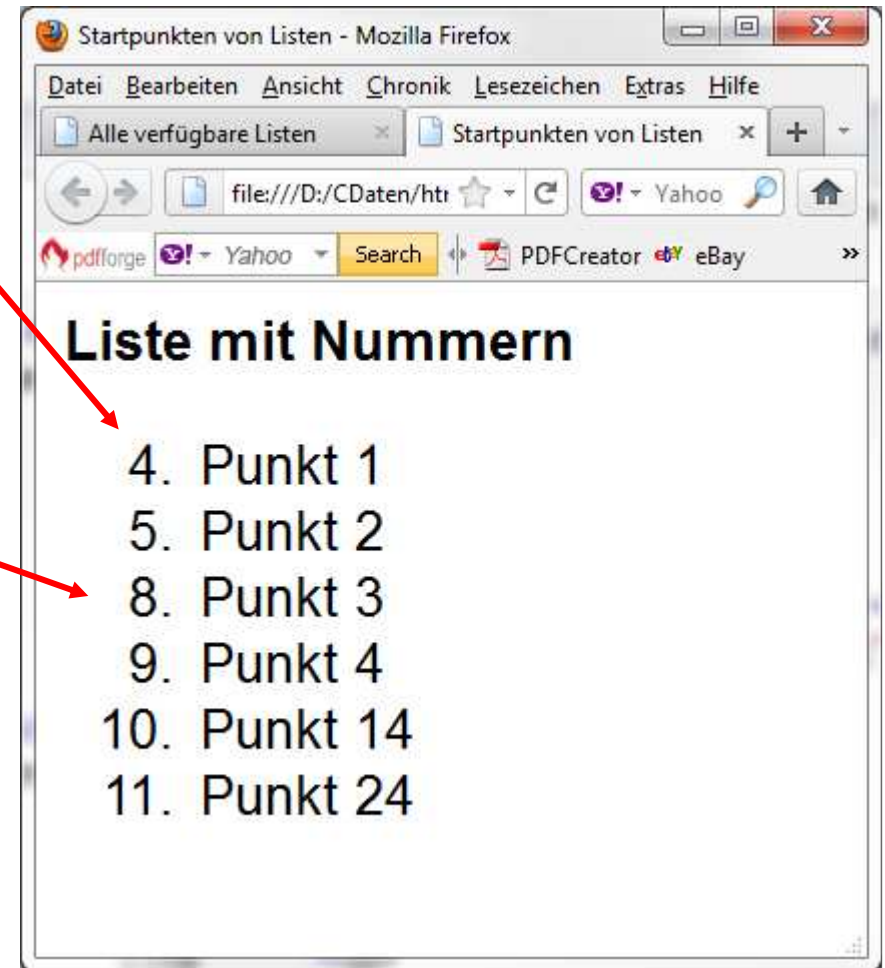
HTML Listen:



HTML Listen:

Liste mit Nummern

```
<ol style="list-style-type:decimal;" start="4">  
<li>Punkt 1</li>  
<li>Punkt 2</li>  
<li value="8">Punkt 3</li>  
<li>Punkt 4</li>  
<li>Punkt 14</li>  
<li>Punkt 24</li>  
</ol>
```



HTML Verknüpfungen

Eine Verknüpfung kann auf eine beliebige Datei verweisen. Dabei kann diese Datei ein HTML-Seite oder eine beliebige andere Datei (z. B. Test.pdf). HTML-Seiten werden direkt angezeigt, andere Dateien werden über einen Viewer geladen oder direkt gestartet.

Angabe des Pfades:

Möglichkeit	Beispiel
Datei als URL-Angabe	http://www.hs-harz.de
Datei lokal mit relativen Pfad	images\bild.gif
Datei lokal mit absoluten Pfad	/images/bild.gif

HTML-Verknüpfungen

- Text-Link
- Anker
- Link eines Bildes
- Mehrere Links in einem Bild (Bereiche)
- Links mit Schalter
- Frames
- Target
- Mail-To
- Formulare (Button)
- Event (onload=....)

HTML Verknüpfungen / Anker: / ist das Trennzeichen unter Linux bzw. Unix

<p> <h2> Links </h2> </p>

<p><h4 >
Verweis zum Beispiel 05
</h4></p>

<p><h4 >
Bsp06 mit neuem Frame
</h4></p>

<p><h4 >
Aufruf eines Bildes im Unterverzeichnis in
einem neuem Fenster
</h4></p>

<p><h4 >
Aufruf eines Bildes (absolut)
</h4></p>

HTML Anker-Verknüpfungen

Verweis auf einer Stelle in der Datei, also **nicht** am Anfang

<h2>Anker, Link in selben Datei

 Go to Liste1

</h2>

 1. Anker

Datei: bsp06.html

<h2>Anker, Link in einer anderen Datei

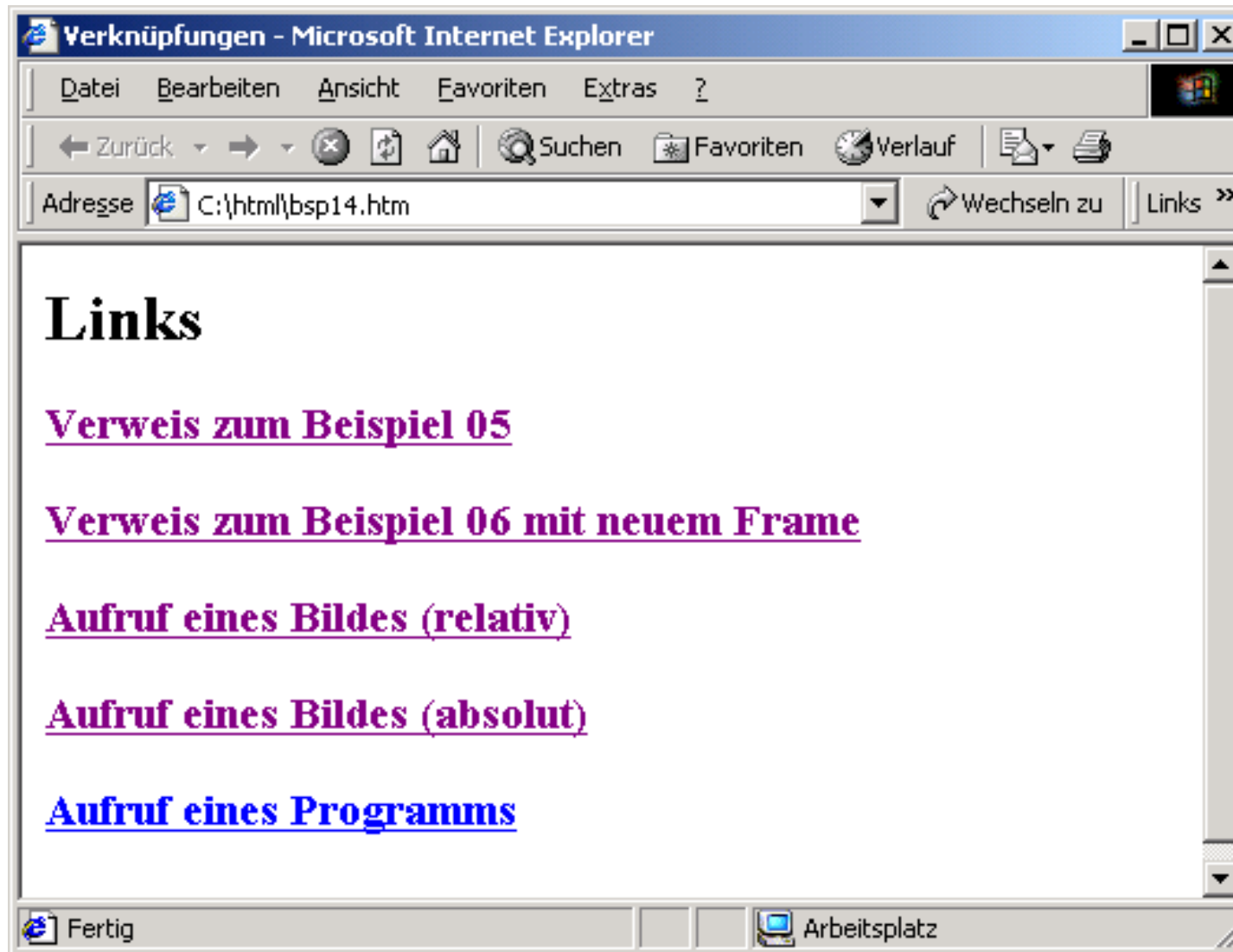
 Go to Liste1 in bsp07

</h2>

Datei: bsp07.html

 1. Anker

HTML Verknüpfungen

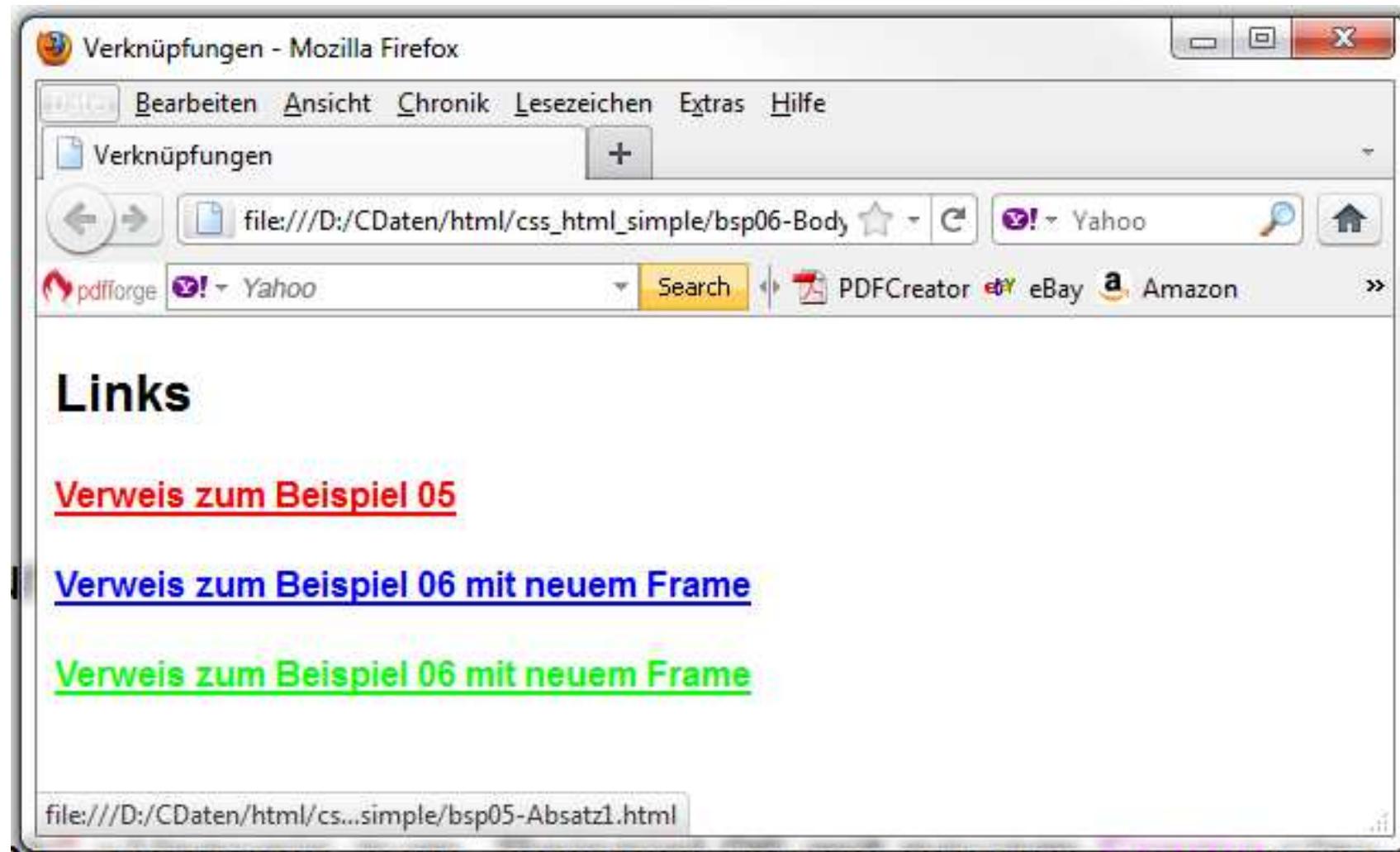


HTML CSS-Body-Attribute: Link-Attribute

link

alink

vlink



HTML CSS-Body-Attribute: Link-Attribute

```
a:link{  
  color: #0000FF;  
}
```

```
a:visited{  
  color: #0000FF;  
}
```

```
a:active{  
  color: #FF0000;  
}
```

```
a:link, a:visited, a:active{  
  color: #FF0000;  
}
```

HTML Bilder:

Bilder:

Anzeigen eines Bildes

```

```

Anzeigen eines Bildes mit Link

```
<a href="bsp1.htm">
```

```

```

```
</a>
```

Besser wäre es, nur „width“ oder „height“ anzugeben

Tabellen in HTML

- regelmäßige Anordnung des Zelleninhalts
- mit Rand/ohne Rand: mehrspaltige Dokumente
- beliebige Zellenelemente (andere Tabellen, Bilder usw.)
- keine absolute Ausrichtung, nur relative Ausrichtung:
- tr: TableRow
- th Table-Header, fett
- td: Table-Data
- colspan: Spalten verbinden
- rowspan: Zeilen verbinden

Tabellen in HTML

<html>

<head><title> 1. Tabellenbeispiel **</title> </head>**

<body>

<table>

<tr>

<th>Überschrift**</th>**

<td>Zelle(1,1)**</td>**

<td>Zelle(1,2)**</td>**

</tr>

</table>

</body>

</html>



th: Table Header

td: Table Data

Tabellen in HTML mit CSS

```
<html>
<head><title> 2. Tabellenbeispiel </title> </head>
<body>
<table>
<tr>
  <th>Überschrift</th>
  <td>Zelle(1,1)</td>
  <td>Zelle(1,2)</td>
</tr>
</table>
</body>
</html>
```

```
/* Aussenrand */
table {
  border: 1px solid;
}
/* einfacher Rand in der Tabelle */
td, th {
  border: 1px solid;
  border-color: black;
}
```




```
/* Aussenrand */
```

```
table {  
    border: 1px solid;  
}
```

```
/* einfacher Rand in der Tabelle */
```

```
td, th {  
    border: 1px solid;  
    border-color: black;  
}
```

- Rand ist doppelt

```
/* Aussenrand */
```

```
table {  
    border: 1px solid;  
    border-collapse: collapse;  
}
```

```
/* einfacher Rand in der Tabelle */
```

```
td, th {  
    border: 1px solid;  
    border-color: black;  
}
```

- Rand ist einfach
- Border-collapse

Tabellen in CSS-HTML

<html>

<head><title> 3. Tabellenbeispiel </title> </head>

<body>

<table>

<tr>

<th>1. Überschrift</th>

<th>2. Überschrift</th>

</tr>

<tr>

<td>Zelle(1,1)</td>

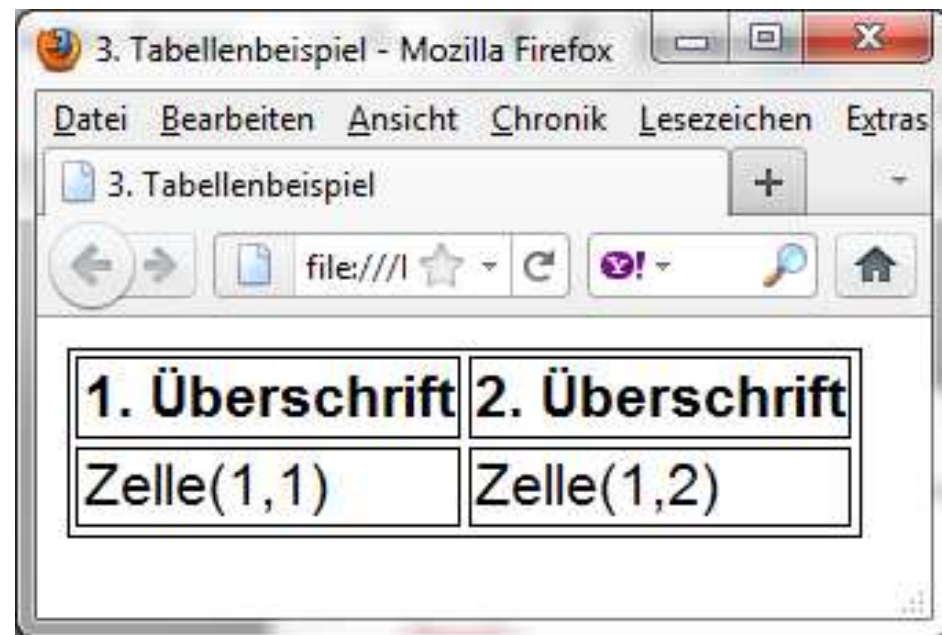
<td>Zelle(1,2)</td>

</tr>

</table>

</body>

</html>



Tabellen in CSS

- | | |
|--|---------------------|
| ■ border | 1px solid #000 |
| ■ border-collapse | collapse, separate |
| - Darstellung untersch. Rahmen | |
| ■ border-spacing | Länge |
| ■ caption-side | top, bottom |
| ■ empty-cells | show, hide |
| - Leere Zelle, aber mit td,td | |
| ■ table-layouts | auto, fixed |
| - Breite der Zellen | |
| ■ vertical-align: | top, center, bottom |
| ■ overflow: | hidden |
| - Elementbereich mit übergroßem Inhalt, Breite=200, Bild mit 250 | |
| ■ speak-header | always, once |

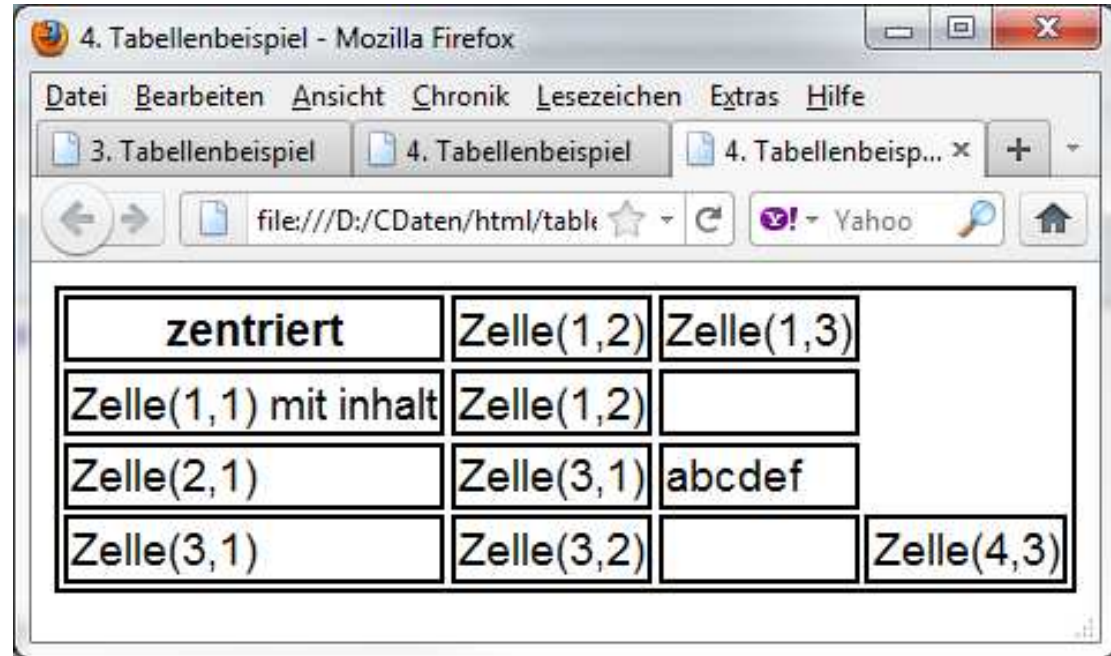
table, tr, colgroup, col, thead, tbody, tfoot, th und td

Tabellen in HTML: Leerzellen

```

<tr>
  <th>zentriert</th>
  <td>Zelle(1,2)</td>
  <td>Zelle(1,3)</td>
</tr>
<tr>
  <td>Zelle(1,1) mit inhalt</td>
  <td>Zelle(1,2)</td>
  <td> <br /> </td>
</tr>
<tr>
  <td>Zelle(2,1)</td>
  <td>Zelle(3,1)</td>
  <td>abcdef </td>
</tr>
<tr>
  <td>Zelle(3,1)</td>
  <td>Zelle(3,2)</td>
  <td> &#160;&#160;&#160;&#160; </td>
  <td>Zelle(4,3)</td>
</tr>

```



 oder

Tabellenüberschrift: caption-side:top/bottom

<table>

<caption> Noten in der Klausur </caption>

<tr>

<th>Name</th>

<th>Note</th>

</tr>

<tr>

<td> Müller, Peter</td>

<td> 1.0</td>

</tr>

<tr>

<td> Meyer, Hans Hogo</td>

<td> 2- </td>

</tr>

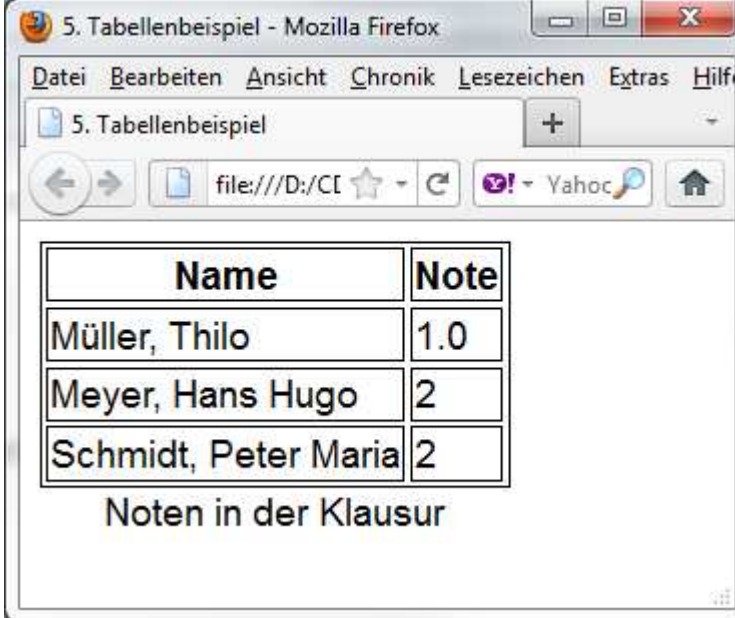
<tr>

<td> Schmidt, Peter Maria</td>

<td> 2 </td>

</tr>

</table>



5. Tabellenbeispiel - Mozilla Firefox

5. Tabellenbeispiel

file:///D:/CI

Yahoo

Name	Note
Müller, Thilo	1.0
Meyer, Hans Hugo	2
Schmidt, Peter Maria	2

Noten in der Klausur

Tabellen in HTML: Horizontale Ausrichtung

```
<tr class="center" >
  <th>Name</th>
  <th>Hobbies</th>
  <th>Bemerkung</th>
</tr>
<tr class="center">
  <td> Müller, Peter</td>
  <td> Tauchen, Segeln, Fahrradfahren</td>
  <td class="left" > keine </td>
</tr>
<tr>
  <td> Meyer, Hans Hugo</td>
  <td> Tennis, Sportschießen </td>
  <td> keine </td>
</tr>
<tr class="center">
  <td> Schmidt, Peter Maria</td>
  <td> Fußball </td>
  <td class="right" > keine </td>
</tr>
```

Tabellen in HTML: Ausrichtung, Ergebnis



6. Tabellenbeispiel vertikale zellenausrichtung - Mozilla Firefox

6. Tabellenbeispiel vertikale zellenausric...

file:///D:/CDaten/html/table/table06.html

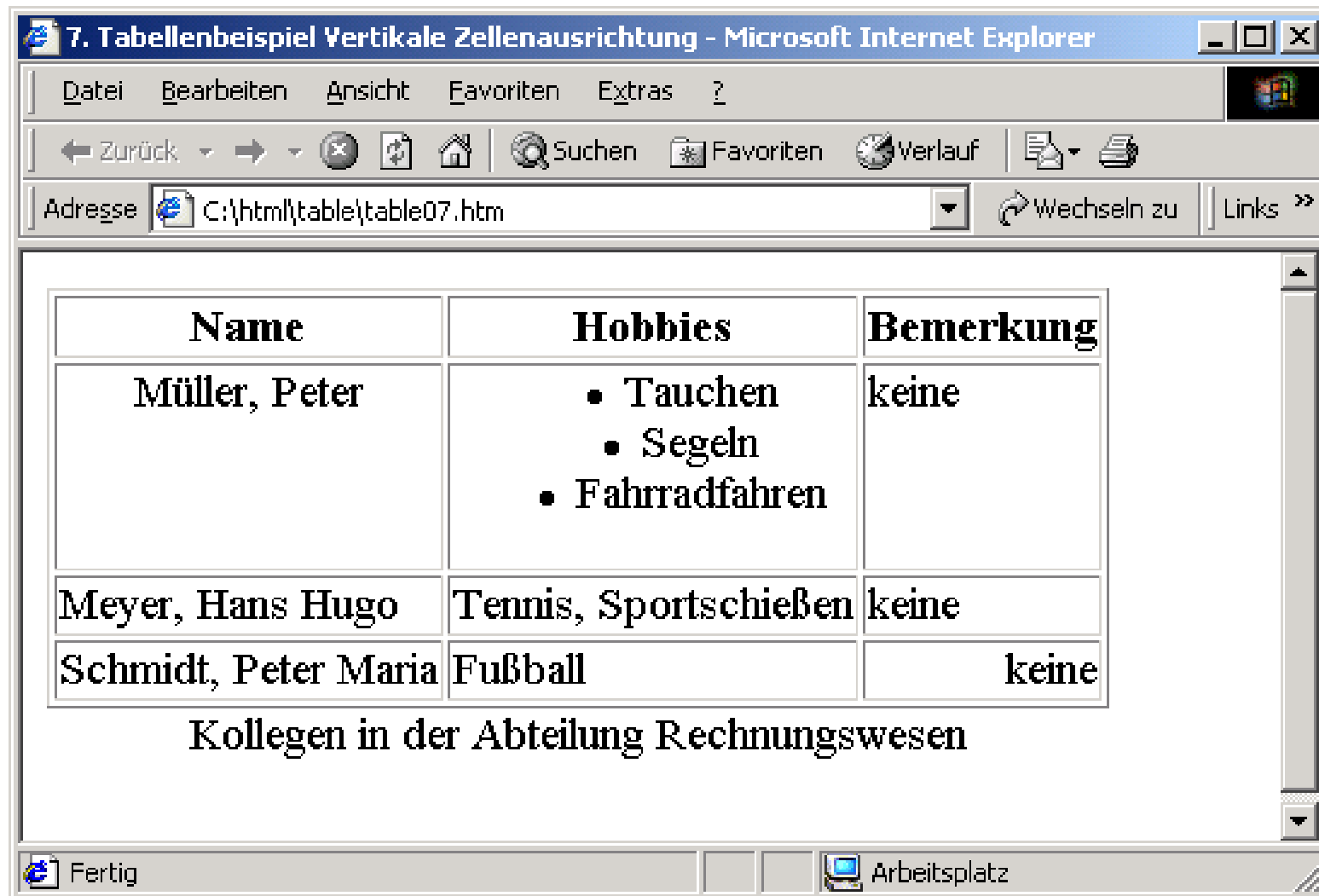
Name	Hobbies	Bemerkung
Müller, Peter	<ul style="list-style-type: none">• Tauchen• Segeln• Fahrradfahren	keine
Meyer, Hans Hugo	Tennis, Sportschießen	keine
Schmidt, Peter Maria	Fußball	keine
Kollegen in der Abteilung Rechnungswesen		

Tabellen in HTML: Vertikale Ausrichtung+Listen

```
<table>
<caption> Kollegen in der Abteilung Rechnungswesen </caption>
<tr >
  <th>Name</th>
  <th>Hobbies</th>
  <th>Bemerkung</th>
</tr>
<tr class="top" class="center">
  <td> Müller, Peter</td>
  <td>
    <ul>
      <li> Tauchen</li>
      <li> Segeln</li>
      <li> Fahrradfahren</li>
    </ul>
  </td>
  <td class="center" > keine </td>
</tr>
```

Abhilfe: vertical-align: top;

Tabellen in HTML: Vertikale Ausrichtung+Listen



Tabellen in HTML: Zellen über mehrere Bereiche

```
<table>
<caption> Firmenfahrzeuge und Nutzung der privaten PKWs </caption>
<tr >
  <th> Name </th>
  <th colspan="2" > Geschlecht </th>
  <th colspan="3" > Fahrzeug  </th>
</tr>
<tr class="center">
  <th> <br /> </th>
  <th> Weiblich</th>
  <th> Männlich</th>
  <th> Firmen-PKW</th>
  <th> Eigener PKW</th>
  <th> PKW des Partners</th>
</tr>
<tr>
  <td> Meyer, Hans Hugo</td>
  <td> <br /> </td>
  <td> X </td>
  <td> Jaguar </td>
  <td> Porsche </td>
  <td> <br /> </td>
</tr>
<tr >
  <td> Schulze, Petra </td>
  <td> X </td>
  <td> <br /> </td>
  <td> Ford Focus </td>
  <td> Camaro </td>
  <td> Golf </td>
</tr>
<tr >
  <td> Vega, Suzanne </td>
  <td> X </td>
  <td> <br /> </td>
  <td> <br /> </td>
  <td> Lancia </td>
  <td> Fahrrad </td>
</tr>
```

Tabellen in HTML: Zellen über mehrere Bereiche

7. Tabellenbeispiel vertikale zellenausrichtung - Mozilla Firefox

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

7. Tabellenbeispiel vertikale zellenaus... x 7. Tabellenbeispiel vertikale zellenaus... x +

file:///D:/CDaten/html/table/table08.html

☆ ↻ Yahoo 🔍 🏠

Name	Geschlecht		Fahrzeug		
	weiblich	männlich	Firmen-pkw	eigener Pkw	Pkw des Partners
Meyer, Hans Hugo		x	Jaguar	Porsche	
Schulze, Petra	x		Ford Focus	Camaro	Golf
Vega, Suzanne	x			Lancia	Fahrrad

Firmenfahrzeuge und Nutzung der privaten Pkws

Tabelle: Zellen über mehrere vertikale Bereiche

The screenshot shows a Mozilla Firefox browser window titled "Beispiel 9. Tabelle - Mozilla Firefox". The address bar displays the file path "file:///D:/CDaten/html/table/table09.html". The page content features a green heading "Tabelle mit vertikalen Zellen und separaten td und th". Below the heading is a table with two columns and multiple rows. The first row contains two header cells labeled "Kopf spalte 1" and "Kopf spalte 2". The second row is empty. The third row contains a cell labeled "1. Testzeile" in the first column and a cell labeled "2. testzeile" in the second column. The fourth row contains a cell labeled "Zelle(n,1)" in the first column and a cell labeled "Zelle(n,1)" in the second column. The fifth row contains a cell labeled "Zelle(n,1)" in the first column and a cell labeled "Zelle(n,1)" in the second column. The sixth row contains a cell labeled "Zelle(n,1)" in the first column and a cell labeled "Zelle(n,2)" in the second column. The table is captioned "Tabellentitel" below it.

Kopf spalte 1	Kopf spalte 2
1. Testzeile	2. testzeile
	Zelle(n,1)
	Zelle(n,1)
Zelle(n,1)	Zelle(n,2)

Tabellentitel

```

/* Aussenrand */
table {
    border: 2px solid;
    caption-side: bottom;

    border-collapse: collapse;
    width: 100%;
}
/* einfacher Rand in der Tabelle */
td, th {
    border: 2px solid;
    border-color: black;
}
td {
    color: rgb(100%, 0%, 100%);
}
th {
    color: rgb(0, 100, 255);
}

```

```

.center {
    text-align: center;
}
.left {
    text-align: left;
}
.right {
    text-align: right;
}
.top {
    vertical-align: top;
}
h2 {
    color: green;
}

```

<table>

<caption>Tabellentitel</caption>

<tr>

<th>Kopf spalte 1</th>

<th>Kopf spalte 2</th>

</tr>

<tr>

<td colspan="2" class="center">...</td>

</tr>

<tr>

<td rowspan="3" class="top">1. Testzeile</td>

<td >2. testzeile</td>

</tr>

Tabelle: Zellen über mehrere vertikale Bereiche

```
<tr>  
  <td>Zelle(n,1)</td>  
</tr>  
<tr>  
  <td>Zelle(n,1)</td>  
</tr>  
<tr>  
  <td>Zelle(n,1)</td>  
  <td>Zelle(n,2)</td>  
</tr>  
</table>
```

CSS-Referenz: white-space

Legt fest, wie Textumbrüche, Leerzeichen und Tabulatoren im Quellcode vom Browser angezeigt werden.

Normal

- Zeilenumbrüche im Quellcode werden nicht dargestellt und mehrere Leerzeichen im Quellcode zu einem zusammengefügt. Der Browser fügt einen Zeilenumbruch am Zeilenende ein.

Pre

- Zeilenumbrüche und Leerzeichen werden, wie im `<pre>`-Tag, so dargestellt, wie sie im Quellcode eingegeben sind.

Nowrap

- Zeilenumbrüche im Quellcode werden nicht dargestellt, mehrere Leerzeichen im Quellcode zu einem zusammengefügt und der Browser fügt am Zeilenende keinen Zeilenumbruch ein. Dieser kann nur mit dem `
`-Tag erzeugt werden.

CSS-Referenz: white-space (2)

Legt fest, wie Textumbrüche, Leerzeichen und Tabulatoren im Quellcode vom Browser angezeigt werden.

pre-wrap

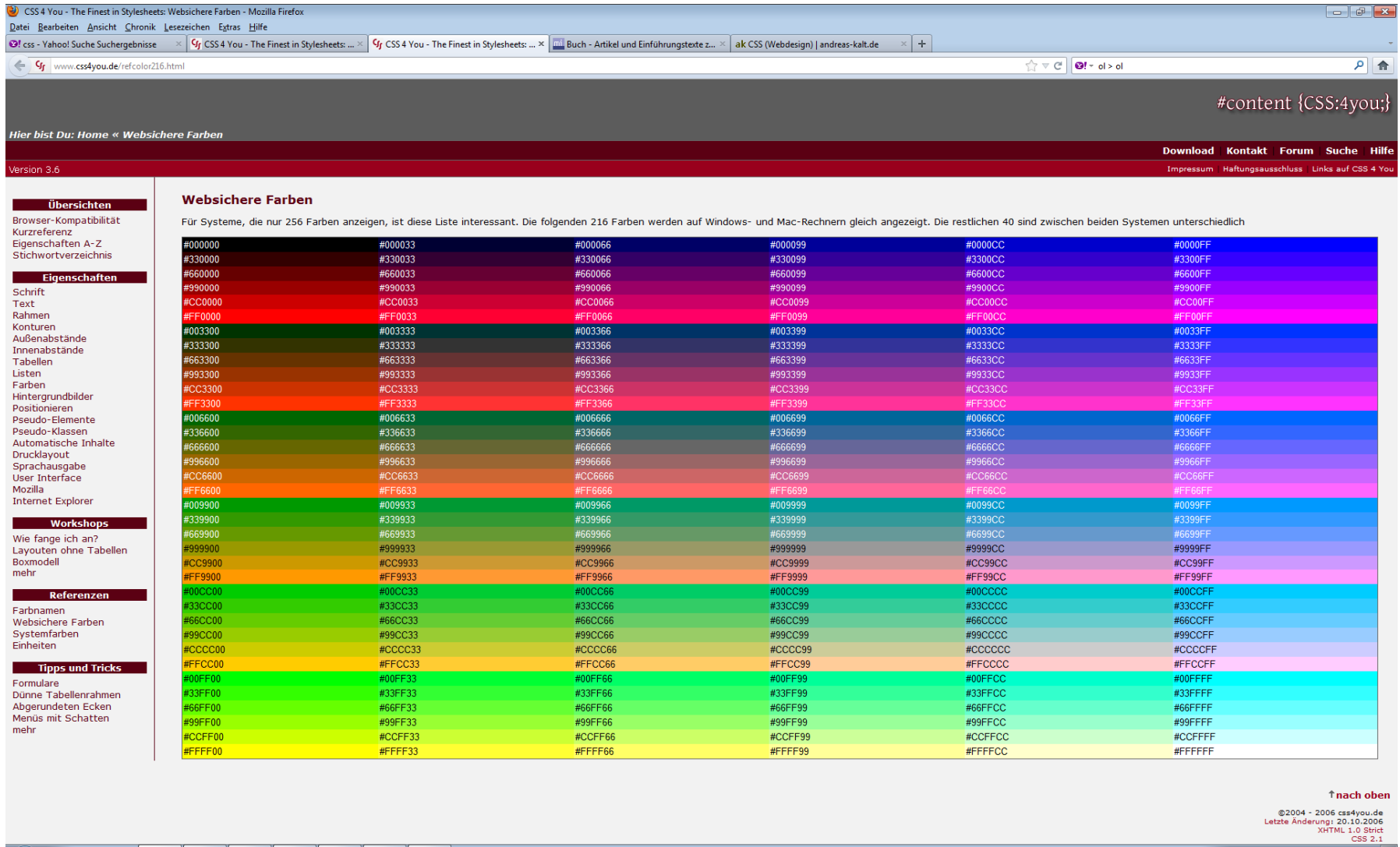
- Wie im `<pre>`-Tag stellt der Browser Zeilenumbrüche und Leerzeichen dar wie sie im Quellcode eingegeben sind, jedoch erfolgt am Zeilenende ebenfalls ein automatischer Zeilenumbruch.

pre-line

- Mehrere Leerzeichen im Quellcode werden zu einem zusammengefügt und Zeilenumbrüche fügt der Browser am Ende der Zeile, durch das `
`-Tag und dort, wo sie im Quellcode eingegeben sind, ein.

`/* Kommentare werden in CSS so geschrieben */`

Websichere Farben

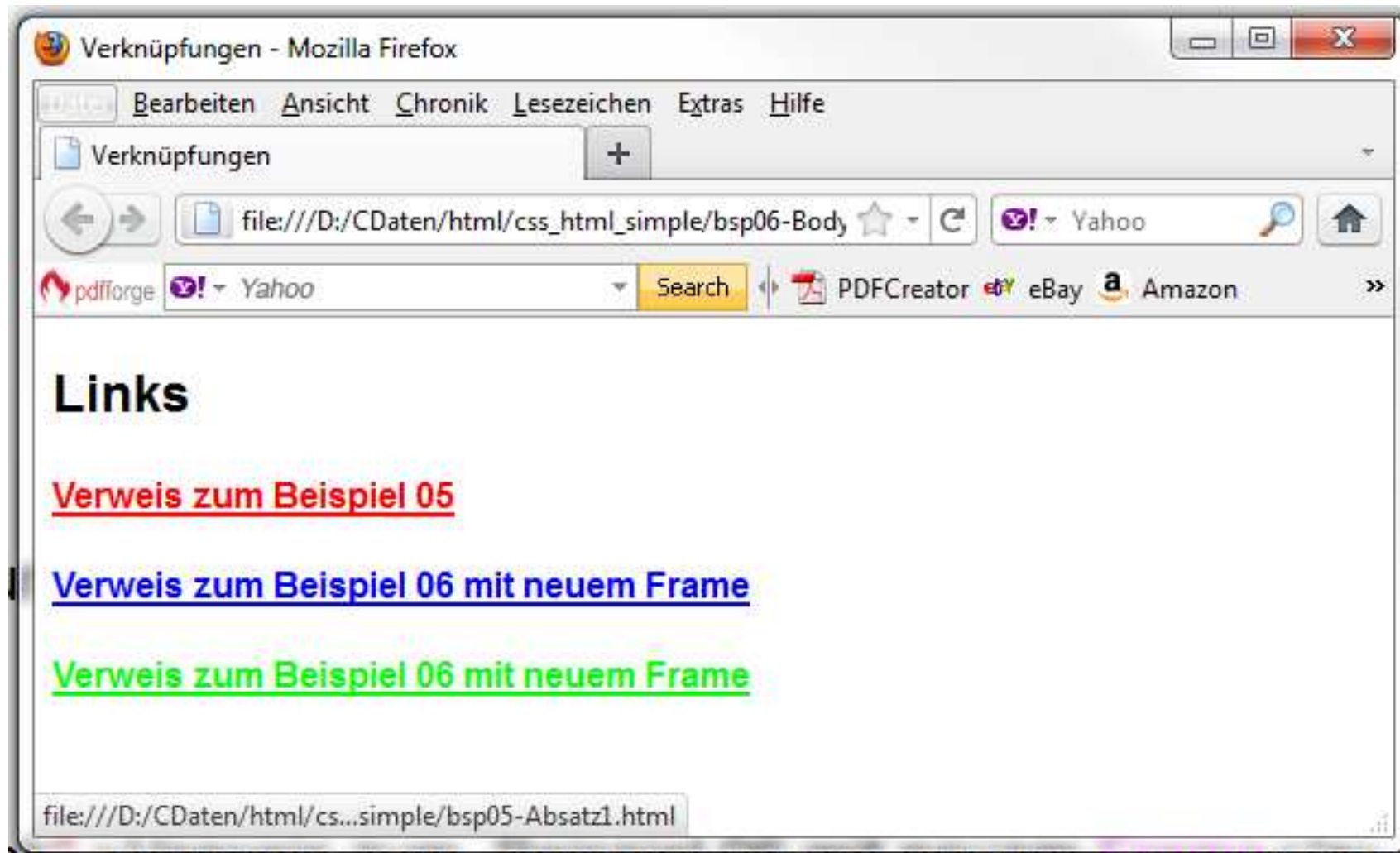


HTML CSS-Body-Attribute: Link-Attribute

link

alink

vlink



HTML CSS-Body-Attribute: Link-Attribute

```
a:link{  
  color: #0000FF;  
}
```

```
a:visited{  
  color: #0000FF;  
}
```

```
a:active{  
  color: #FF0000;  
}
```

Inhalte strukturieren

- Das <div>-Tag ist ein Container für mehrere HTML-Elemente, denen durch die Kernattribute des <div>-Tags Stylesheet-Eigenschaften zugewiesen werden.
- div-Elemente sind Blockelemente, da das öffnende und das schließende <div>-Tag jeweils zu **Zeilenumbrüchen** – äquivalent zum
-Tag – führen.
- <div>-Tags lassen sich ineinander verschachteln und bilden einen leistungsfähigen Mechanismus
- Man verwendet das <div>-Tag dazu, eine Menge von logisch zusammengehörigen HTML-Elementen mit der Hilfe von Cascading Stylesheets zu formatieren, positionieren oder mit JavaScript zu animieren. Dazu werden die Kernattribute *class*, *id* und *style* benutzt.
- Äquivalent zum <div>-Tag gibt es das **-Tag**, das benutzt wird, wenn einer Gruppe von HTML-Elementen Inlinestile – also Stile ohne Zeilenumbruch – zugewiesen werden sollen.

Text und Ausrichtung

CSS-Ausrichtungs-Attribute

- text-align:left;
- text-align:center;
- text-align:right;
- text-align:justify;

```
<style type="text/css">
    .align_l {
        text-align:left;
        color:#0000ff;
    }
    .align_r {
        text-align:right;
        color:#ff0000;
    }
    .align_c {
        text-align:center;
        color:#ff00ff;
    }
    .align_j {
        text-align:justify;
        color:#00ff00;
    }
</style>
```

Beispiel css_bsp1.xhtml

```
.div1 {  
    color:blue;  
}
```

```
p {  
    color:green;  
}
```

```
.over {  
    text-decoration: overline;  
}
```

<p>

Diese Zeile sollte in grün angezeigt werden

</p>

<div class="div1">

3. Zeile, im zweiten Beispiel, div

</div>

css_bsp1.xhtml

Beispiel css_bsp1.xhtml



Klassen-Selektoren in CSS

Klassen werden in den Definitionen verwendet. Es gibt:

- **vordefinierte** Selektoren
- Pseudo-Selektoren
- eigene Selektoren mit beliebigen Namen.

Beispiele für vordefinierte Klassen:

- h1
- h6
- body
- td
- th
- p

Klassen-Selektoren in CSS

Beispiel:

```
body {  
    background: yellow;  
    font-size: 16pt;  
}
```

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

Klassen-Selektoren in CSS: Beispiel rahmen

Ein Style kann man nun in die CSS-Datei eintragen, um spezielle Formate zu definieren, **ohne** in die HTML-Datei weitere Attribute einzufügen:

```
.rahmen {  
  border: 1px solid green;  
}
```

allgemeine Regel

```
p.rahmen {  
  border: 1px solid red;  
}
```

gilt nur in einem p-Abschnitt

```
h1.rahmen {  
  border: 1px solid blue;  
}
```

gilt nur in einem h1-Abschnitt

Klassen-Selektoren in CSS

Anwendung:

`<h1 class="rahmen">Eine h1-Überschrift mit Rahmen</h1>`

`<h2 class="rahmen">Eine h2-Überschrift, allgm. Regel</h2>`

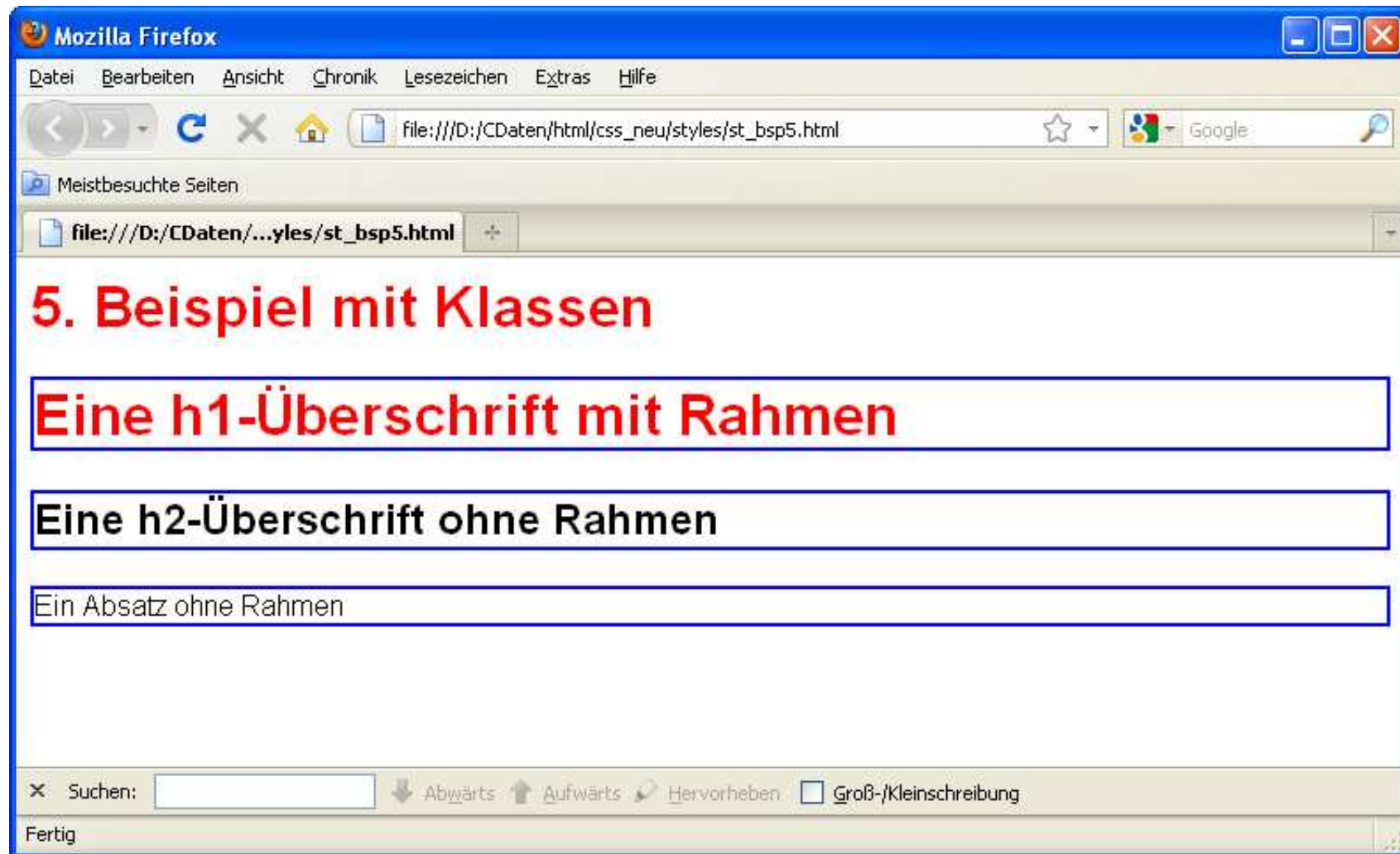
`<p class="rahmen">Ein Absatz, p-Regel</p>`

`<div class="rahmen">Ein Absatz, allgm. Regel</div>`

CSS-Beispiele: st_bsp4.html



CSS-Beispiele: st_bsp5.html



Klassen-Selektoren in CSS

Beispiele:

```
h1, h2 {  
    color:red;  
}
```

```
h1 {  
    font-size: 2em;  
}
```

Beispiele:

```
h2 {  
    font-size: 1.5em;  
}  
.under {  
    text-decoration:underline;  
}  
.blue {  
    color:blue;  
}
```

Anwendung:

<p class="under blue">Ein Text in blauer Schrift und unterstrichen</p>

Pseudo-Elemente und -Klassen in CSS

:after	Gilt nach einem Element
:before	Gilt Vor einem Element
:first-letter	Das erste Zeichen in einer Zeile
:first-line	Die erste Zeile in einem Absatz
:first-child	Das erste "Kind" des Abschnittes (body,ol)
:last-child	Das letzte "Kind" des Abschnittes (body,ol)
:active	benutzt in Hyperlink
:focus	wenn Element den Fokus erhält
:hover	Maus über Elemente, (Hyperlink)
:lang	Sprache
:link	Normaler Link (Hyperlink)
:visited	Besuchter Link

st_bsp6.html und st_bsp6.css

<p>

A.B Zeile

</p>

Dies ist die 2. Zeile

Dies ist die 3. Zeile

 Text1

 Text2

 Text3

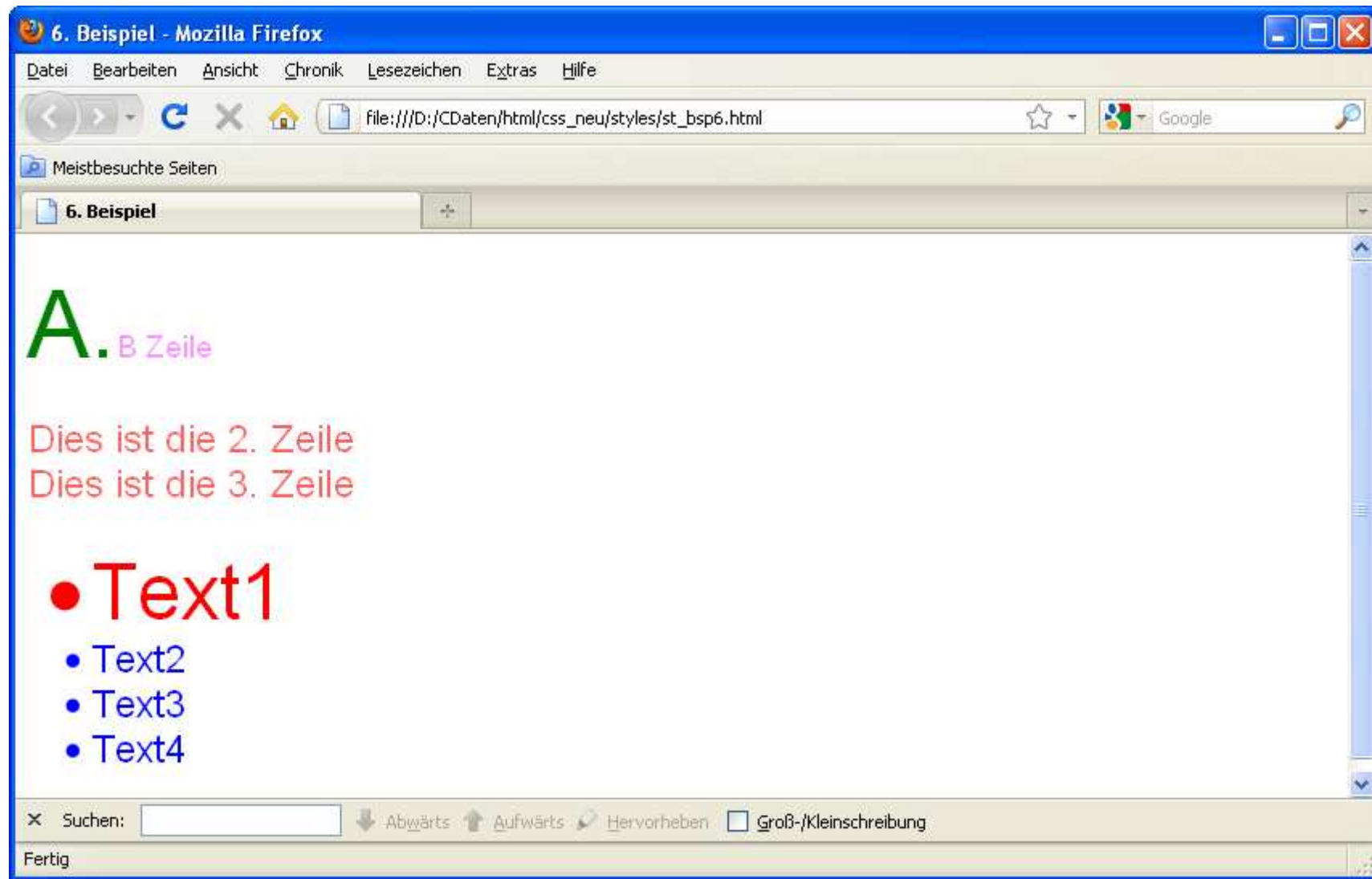
 Text4


```
p:first-letter{  
    font-size:3em;  
    color:green;  
}
```

```
p:first-line{  
    font-size:80%;  
    color:#FF96FF;  
}
```

```
li:first-child{  
    color:red;  
    font-size:2em;  
}
```

st_bsp6.html und st_bsp6.css



Id-Selektor in CSS

- Die Klassen-Styles können an **mehreren** Stellen benutzt werden. Dies ist bei der Benutzung von Java-Script bzw. bei der Layout Definition von Bereichen (Sheets) nicht immer gewünscht.
- Deshalb existieren ID-Selektoren
- Die Style-Definition und die Adresse eines ID-Selektors sollte nur einmal existieren. Jedes Element besitzt eine eindeutige ID, mit der dieses adressiert werden kann.
- ID-Selektoren werden häufig in Verbindung mit Ajax benutzt

Beispiel

```
.div1 {  
    color:blue;  
}
```

```
#iddiv1 {  
    color:green;  
}
```

```
.over {  
    text-decoration: overline;  
}
```

<div id="iddiv1">

2. Zeile, im Beispiel, iddiv1, sollte nur einmal vorhanden sein

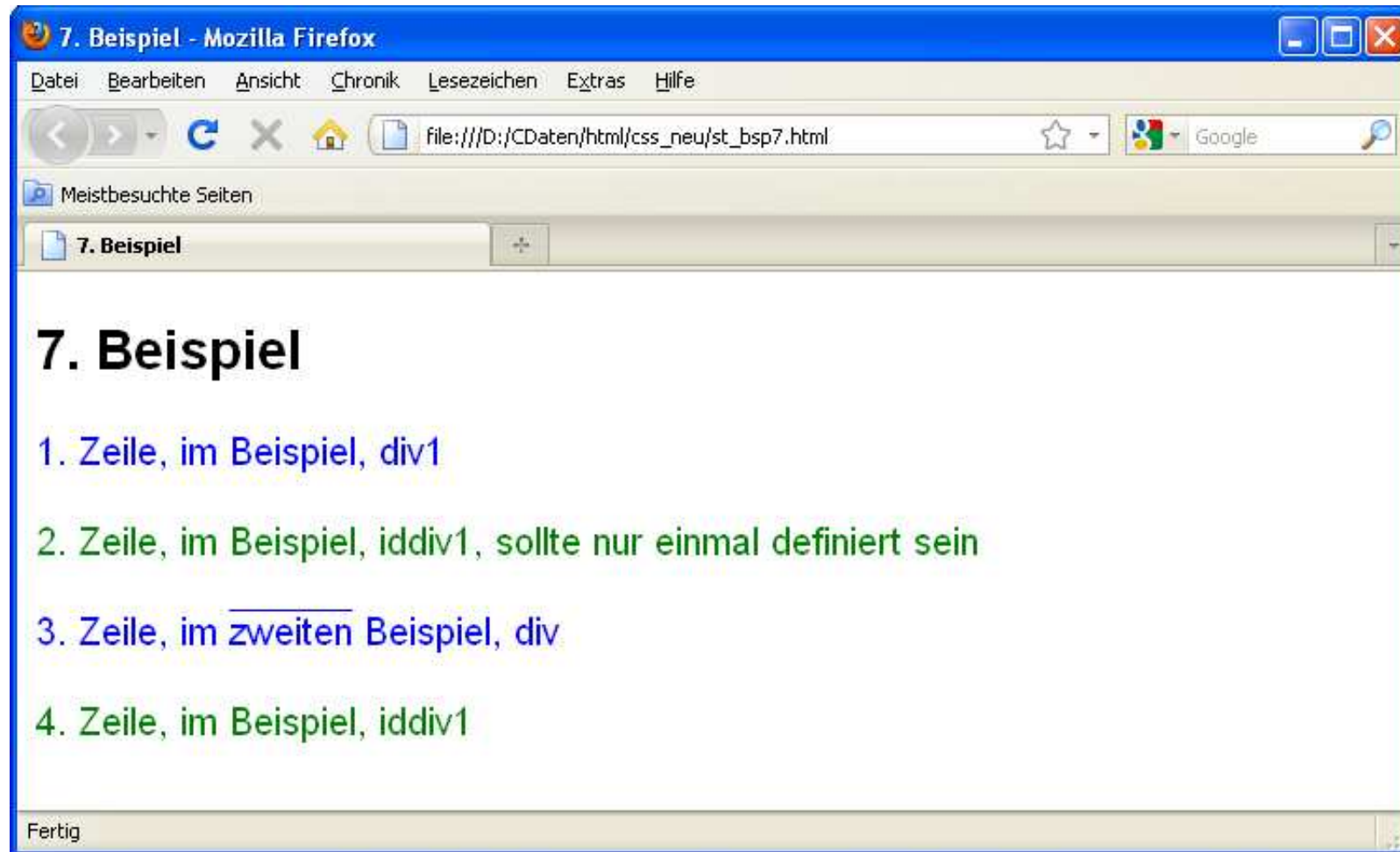
</div>

<div class="div1">

3. Zeile, im zweiten Beispiel, div

</div>

st_bsp7.html



Font-Attribute

`<div style="text-decoration:underline;">`

Test: Nun Unterstrichen

`</div>`

Test: ``Nun Überstrichen``

`<div style="text-decoration: line-through;">`

Test`<u>`Nun Durchgestrichen`</u>`

`</div>`

`<div style=" text-decoration: blink;">`

Test`<u>`Nun Blinkend`</u>`

`</div>`

Test ``Nun Kapitälchen``

Font-Attribute

Test H20

Test: A2+B2=C2

<div>

Test Normale Schrift:imnpjEtwas kleiner:

imnpjNoch kleiner:

imnpj

</div>

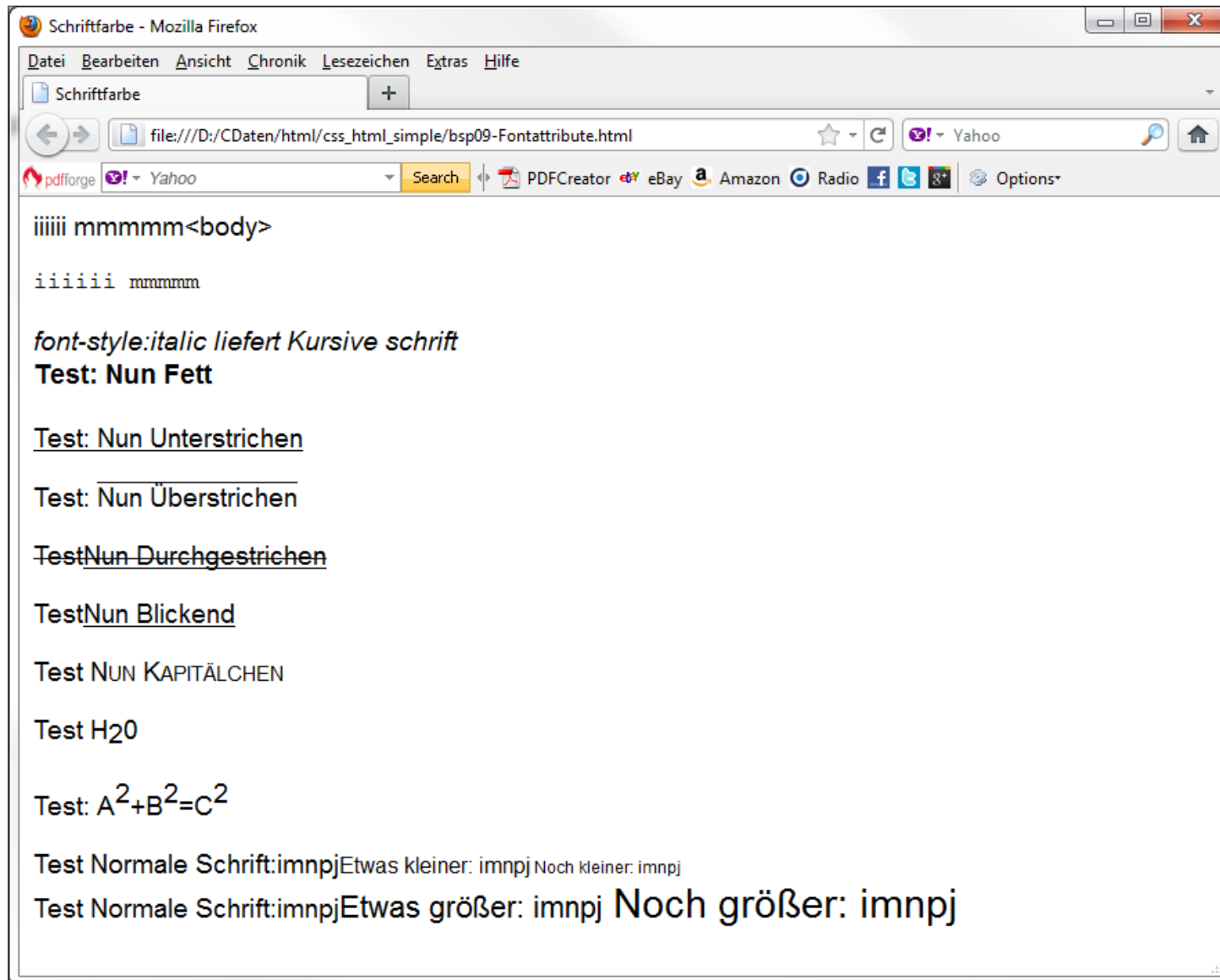
<div>

Test Normale Schrift:imnpjEtwas größer:

imnpjNoch größer:

imnpj

</div>



Cascade und Nachbarschaft in CSS

- Alle Elemente erben alle Eigenschaften des Parent-Element (außer body)
- Um genauer die Vererbung steuern zu können, existieren mehrere Selektor-Typen.
- Es stehen jeweils zwei Elemente in Beziehung
- Direkter Nachfahr-Selektor (*Element ist in parent*)

- ol ol td li

- Nachfahr-Selektor, Tiefe > 1

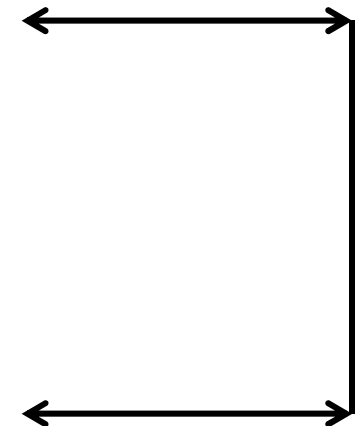
- body * p div * h2

- Nebeneinander-Selektor (*Hintereinander*)

- div + h2 ul + ul p + h2

- Direkter Nachfahr oder Hintereinander -Selektor

- body > p p > h2 div > h2



Cascade mit cs_bsp01.html

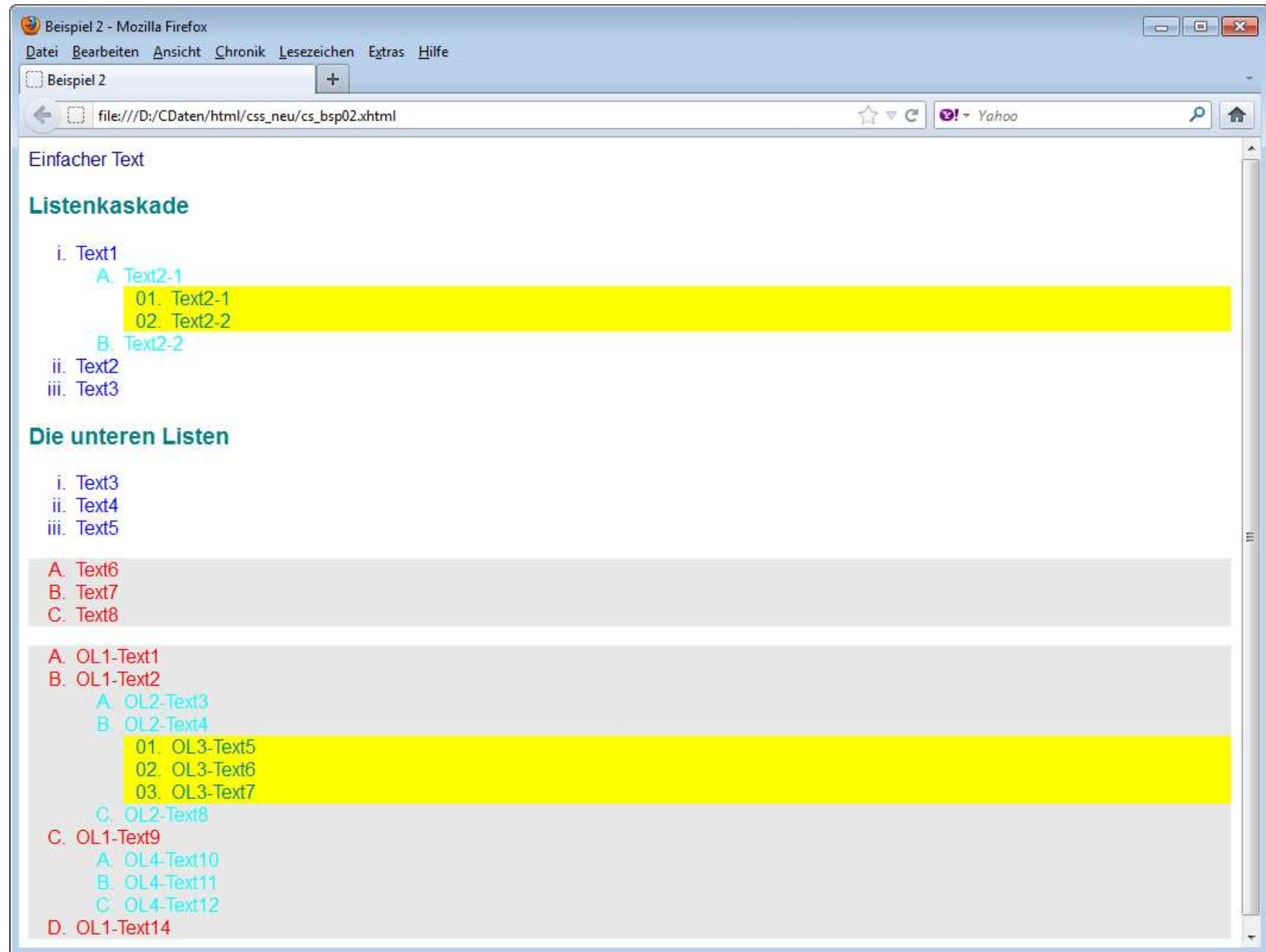
```
ul {  
    color:blue;  
    list-style-type: circle;  
}  
  
ul ul {  
    color:green;  
    list-style-type: disc;  
}  
  
ul + ul {  
    color:yellow;  
    background-color:black;  
    list-style-type: disc;  
    list-style-position: inside;  
}
```



cs_bsp02.css

```
ol {  
    color:blue;  
    list-style-type: lower-roman;  
}  
  
/* Element liegt in einem anderen Element  
*/  
  
ol ol{  
    color:green;  
    list-style-type: lower-greek;  
}  
  
ol + ol{  
    color:red;  
    background-color:#E8E8E8;  
  
    list-style-type: upper-latin;  
}
```

```
/* Element liegt hinter- oder ineinander eines  
anderen Element */  
  
ol > ol{  
    color: cyan;  
    list-style-type: upper-latin;  
}  
  
/* Element liegt ineinander eines anderen  
Element */  
  
ol * ol{  
    color:teal;  
    background-color:yellow;  
    list-style-type: decimal-leading-zero;  
}
```



```

<ol>
  <li> Text1 </li>
  <ol >
    <li> Text2-1 </li>
    <ol >
      <li> Text2-1 </li>
      <li> Text2-2 </li>
    </ol>
    <li> Text2-2 </li>
  </ol>
  <li> Text2 </li>
  <li> Text3 </li>
</ol>
<h3> Die unteren Listen</h3>
<ol>
  <li> Text3 </li>
  <li> Text4 </li>
  <li> Text5 </li>

```

```

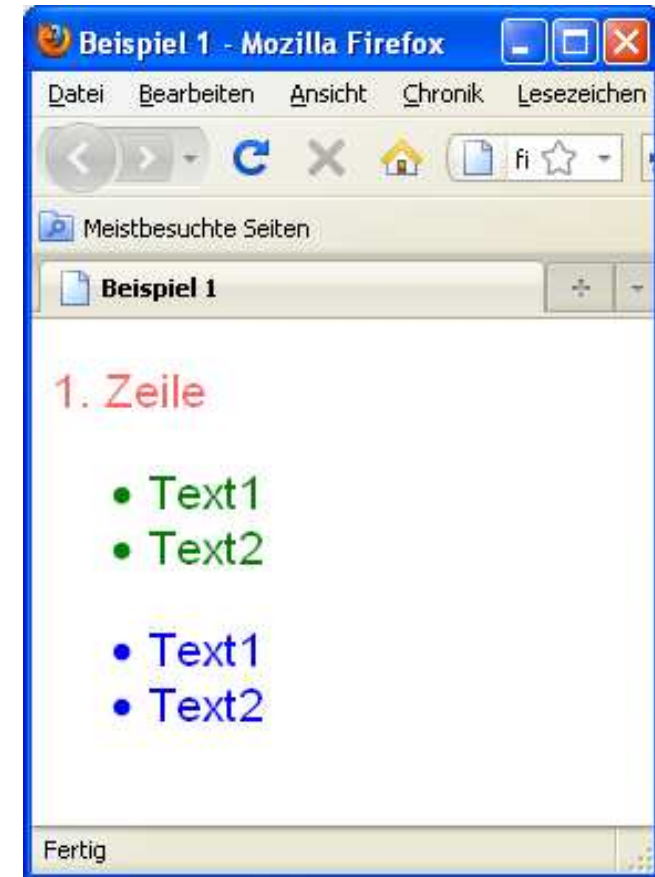
<ol >
  <li> Text6 </li>
  <li> Text7 </li>
</ol>

<ol >
  <li> OL1-Text1 </li>
  <ol>
    <li> OL2-Text3 </li>
  <ol >
    <li> OL3-Text7 </li>
  </ol>
    <li> OL2-Text8 </li>
  </ol>
  <li> OL1-Text9 </li>
  <ol >
    <li> OL4-Text10 </li>
  </ol>
  <li> OL1-Text14 </li>

```

cs_bsp03.html: Cascade mit ID-Selektor

```
ul {  
  color:blue;  
}  
  
#nav ul {  
  color:green;  
}  
  
<div id="nav">  
  <ul >  
    <li> Text1 </li>  
    <li> Text2 </li>  
  </ul>  
</div>  
  
<div>  
  <ul >  
    <li> Text1 </li>  
    <li> Text2 </li>  
  </ul>  
</div>
```



cs_bsp04.html: Komplettes Beispiel



weine.html:

```
a {  
    color: blue;  
}  
a.weisswein {  
    color: #FFBB00;  
}  
a.rotwein {  
    color: #FF00FF;  
}  
remark {  
    font-size=2pt;  
}
```



weine.html:

```
<ul>
```

```
  <li><a href="ri.htm" class="weisswein">Riesling</a></li>
```

```
  <li><a href="ch.htm" class="weisswein">Chardonnay</a></li>
```

```
  <li><a href="pb.htm" class="weisswein">Pinot Blanc</a></li>
```

```
</ul>
```

<p>Trauben für Rotwein:</p>

```
<ul>
```

```
  <li><a href="cs.htm" class="rotwein">Cabernet Sauvignon</a></li>
```

```
  <li><a href="me.htm" class="rotwein">Merlot</a></li>
```

```
  <li><a href="pn.htm" class="rotwein">Pinot Noir</a></li>
```

```
</ul>
```

CSS-Beispiele:

<-- Farbe eines nicht besuchten Links ->

a:link{

color: #FF0000;

}

<-- Farbe eines besuchten Links ->

a:visited{

color: #00FF00;

}

<-- Farbe, wenn Maustaste über einen Link gedrueckt wird ->

a:active{

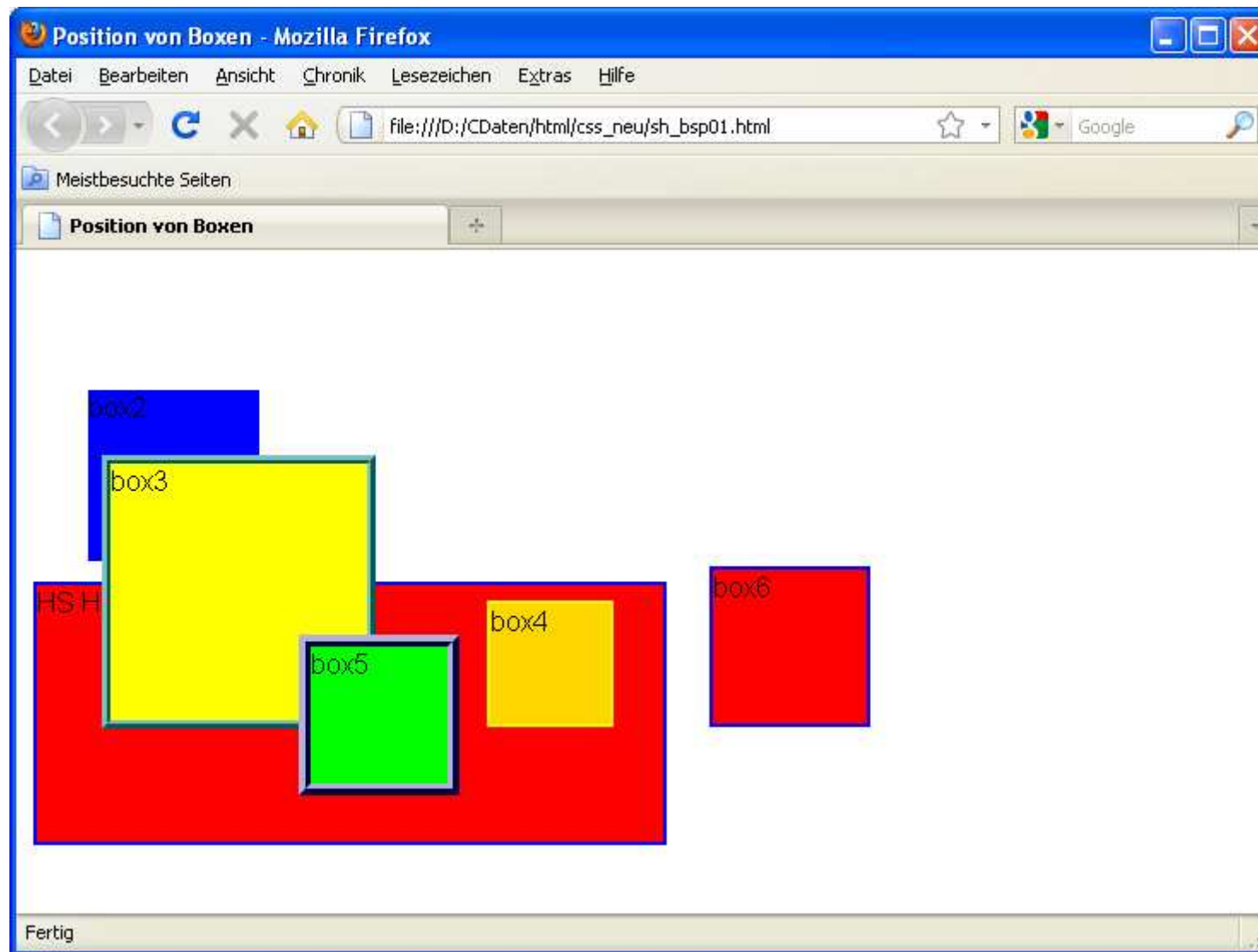
color: #FFFF00;

}

Sheets in CSS

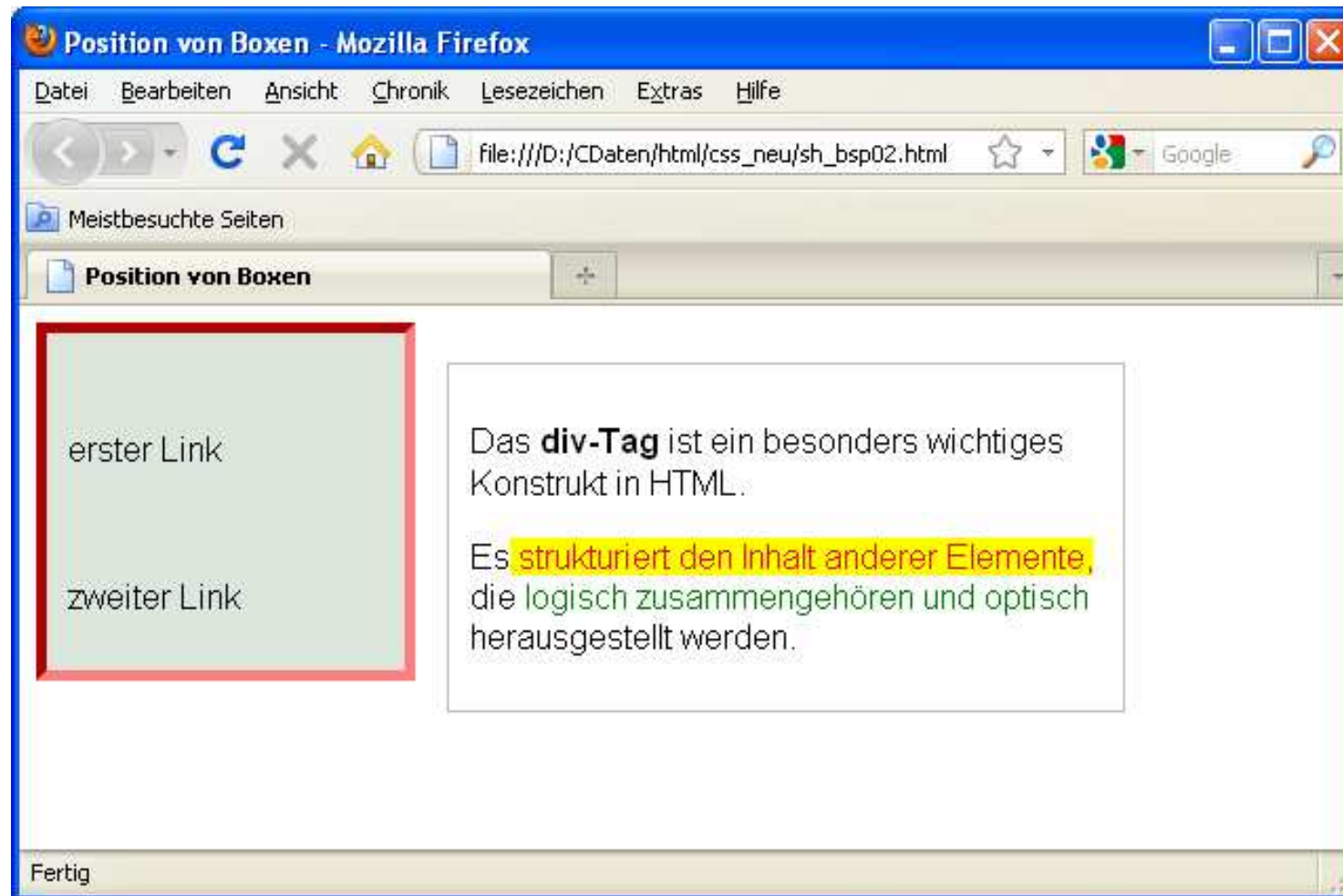
- Bereiche werden mit dem "div"-Tag erstellt
- Jeder Bereich hat einen "parent" und erbt alle Eigenschaften
 - Ränder, Schriftgrößen, Fonts
 - Problem Multiplikation 2em
- Man kann diese Bereiche beliebig positionieren
 - top, left, width, height
 - Ränder, Abstände (margin)
 - float, none, links und rechts
- Das "parent"-Element eines Bereiches kann sich beim Vergrößern oder Verkleinern ändern

sh_bsp01.html: Bereiche über- und nebeneinander

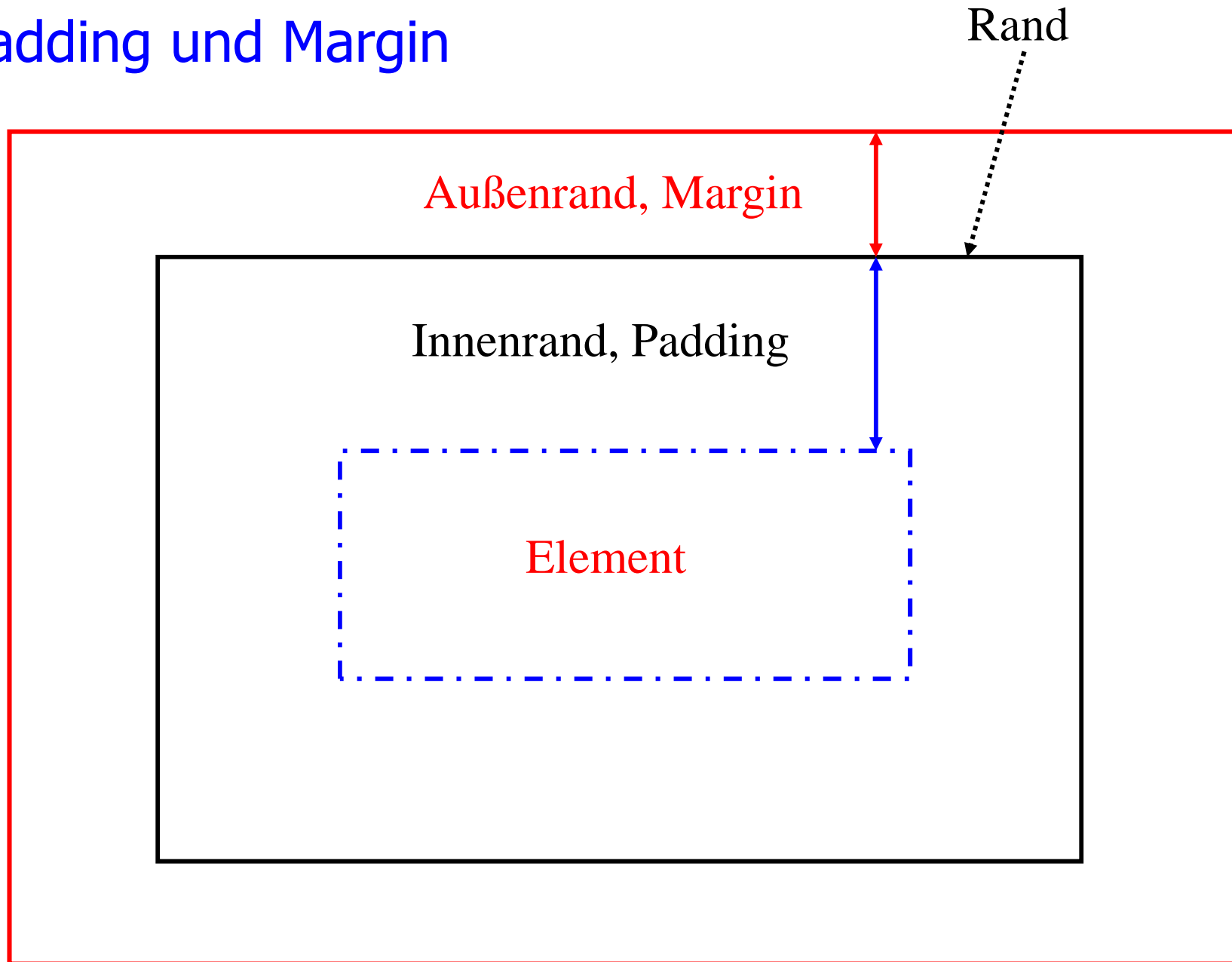


```
#box1 { position:absolute; top:50%;  
        left:10px; width:50%; height:150px;  
        z-index:1;  
}  
#box2 { position:absolute; top:80px;  
        left:40px; width:100px; height:100px;  
        z-index:2;  
}  
#box3 { position:absolute; top:120px;  
        left:50px; width:150px; height:150px;  
        z-index:3;  
}  
#box4 { position:absolute; top:80px;  
        left:220px; width:70px; height:70px;  
        z-index:1;  
}
```

sh_bsp02.html: Bereiche nebeneinander



Padding und Margin



Attribute für die Positionierung und Größe

bottom

- Position unten, Längen- oder Prozentangabe, **auto**

clear

- Text umfließen beenden, **left, right, both, none**

clip

- Ausschnitt, **rect()**, **auto**

display

- Anzeigeart, **none** und viele weitere

float

- Text umfließen, **left, right, none**

height

- Höhe, Längen- oder Prozentangabe, **auto**

Attribute für die Positionierung und Größe

left

- Position links, Längen- oder Prozentangabe, **auto**

max-height

- Maximale Höhe, Längen- oder Prozentangabe, **none**

max-width

- Maximale Breite, Längen- oder Prozentangabe, **none**

min-height

- Minimale Höhe, Längen- oder Prozentangabe

min-width

- Minimale Breite, Längen- oder Prozentangabe

overflow

- Überlauf in Tabellen, visible, hidden, scroll, auto

Attribute für die Positionierung und Größe

Position

- Positionsart, **absolute, relative, static, fixed**
- right**
- Position rechts, Längen- oder Prozentangabe, **auto**
- top**
- Position oben, Längen- oder Prozentangabe, **auto**
- visibility**
- Unsichtbar, **visible, hidden, collapse**
- width**
- Breite, Längen- oder Prozentangabe, **auto**
- z-index**
- Überlappung, Zahl, **auto**

Attribut position: Zeigt an, wie die Box positioniert wird

static

- keine spezielle Positionierung, normaler Elementfluss (Normal-Einstellung)

relative

- relative Positionierung (Verschiebung), gemessen an der Normalposition oder Anfangsposition des Elements selbst
- fließt mit den anderen Elementen

absolute

- absolute Positionierung, gemessen am Rand des **nächsthöheren** Vorfahrenelements, das nicht die Normaleinstellung position:static aufweist. Scrollt mit
- Unabhängig von anderen Elementen

fixed

- absolute Positionierung, gemessen am "Viewport", d.h. am Browserfenster. Bleibt beim Scrollen stehen

float

- Attribut zeigt an, wie ein Element andere umfließt
 - left Aktive Element fließt links von anderen
 - right
 - none
- zeigt an, an welche Seite das Element fließt
- Es muss immer eine Breite, width, angegeben werden
- Ersetzt das Attribut align

Wichtige display-Attribute

none

- Keine Anzeige (Ausschalten der Listensymbole)

block

- Das Element wird als Block-Element dargestellt

inline

- Das Element wird als Inline-Element dargestellt

list-item

- Sichtbar. Das Element erzeugt einen Absatz und hat ein Aufzählungszeichen

Positionieren von Blöcken: float mit umbrechen

- float:left
- width:14em;
- height:auto;

- float:left
- width:600 px;
- height:auto;

Beispiel: sh_bsp02.html

Positionieren von Blöcken: float ohne umbrechen

- float:left
- position:relative;
- width:14em;
- top:0;
- left:0;
- height:auto;

- margin-left:15em;
- width:600 px;
- height:auto;

Beispiel: sh_bsp03.html

Positionieren von Blöcken: absolute

- `position:absolute;`
- `width:14em;`
- `top:0;`
- `left:0;`
- `height:auto;`

- `position:absolute;`
- `width:500 px;`
- `top:0;`
- `left:15em;`
- `(left:auto);`
- `height:auto;`

Beispiel: sh_bsp04.html

Positionieren von Blöcken: absolute

- position:absolute;
- width:14em;
- top:0;
- left:0;
- height:auto;

- position:absolute;
- top:0;
- left:200px;
- right:10px;
- height:auto;
- Ohne width

Beispiel: sh_bsp04a.html

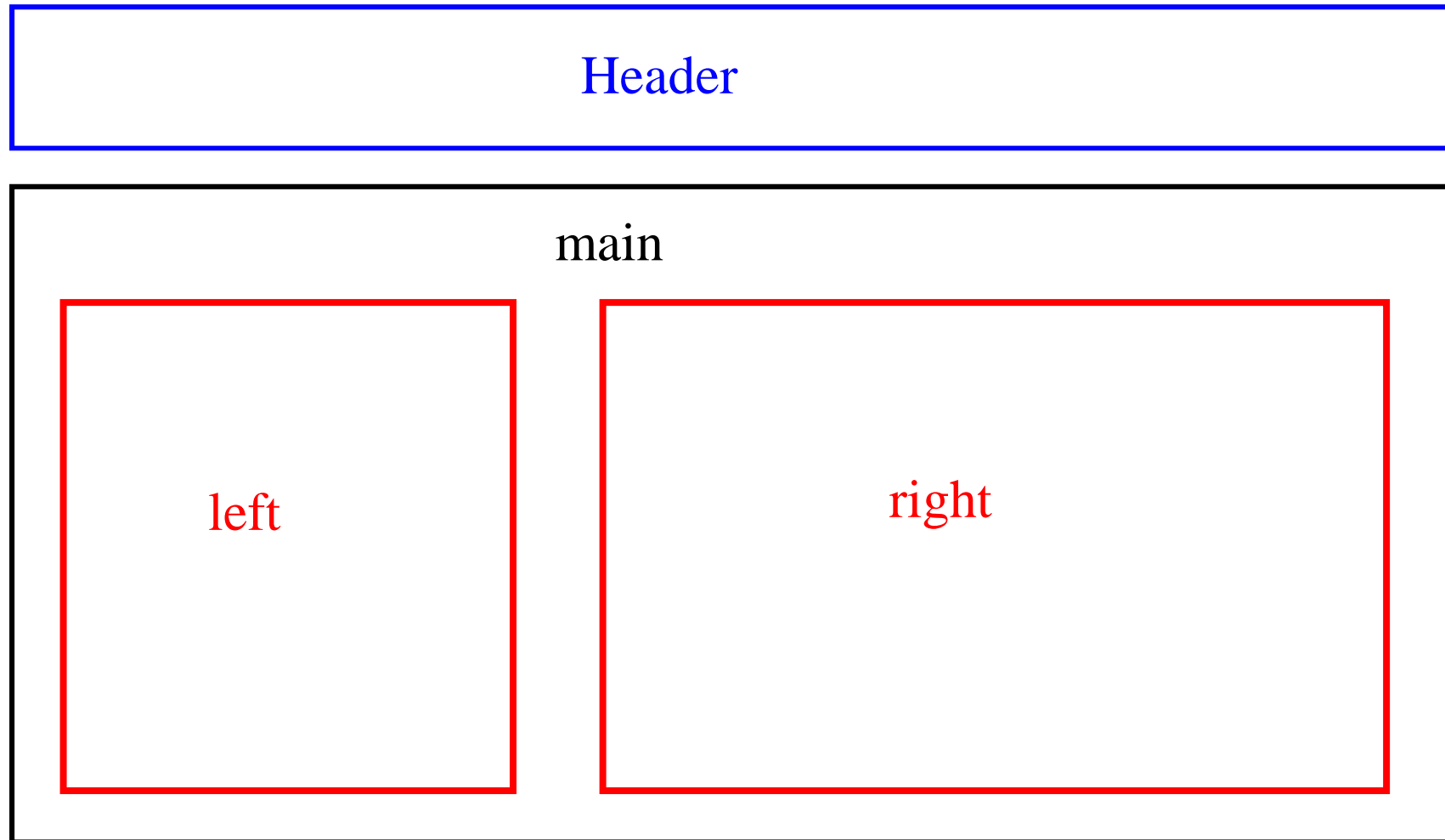
Positionieren von Blöcken: absolute

- position:absolute;
- top:0;
- left:0;
- bottom:0px;
- right:400px;

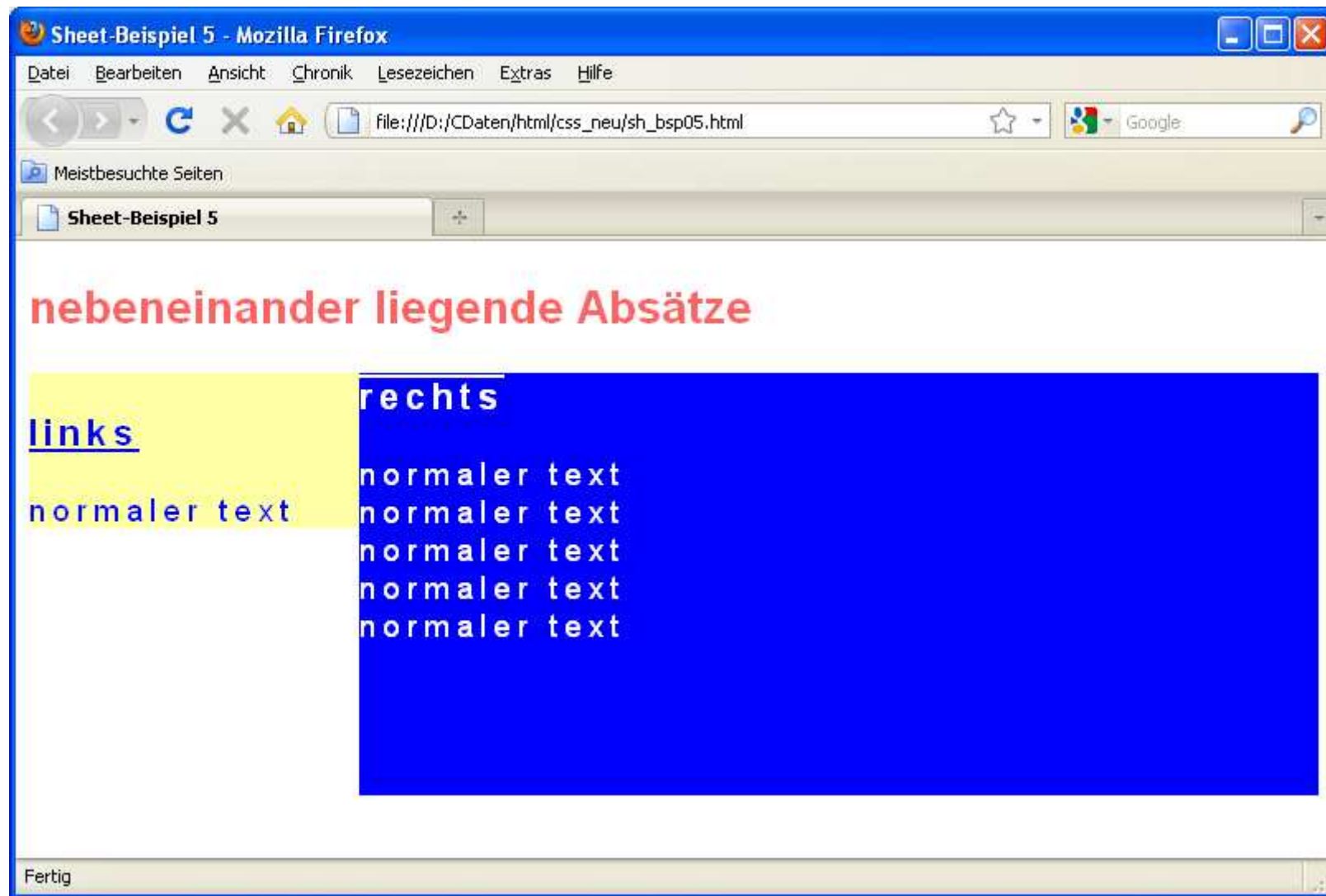
- position:absolute;
- top:0;
- left:400px;
- bottom:10px;
- right:10px;

Beispiel: sh_bsp04b.html

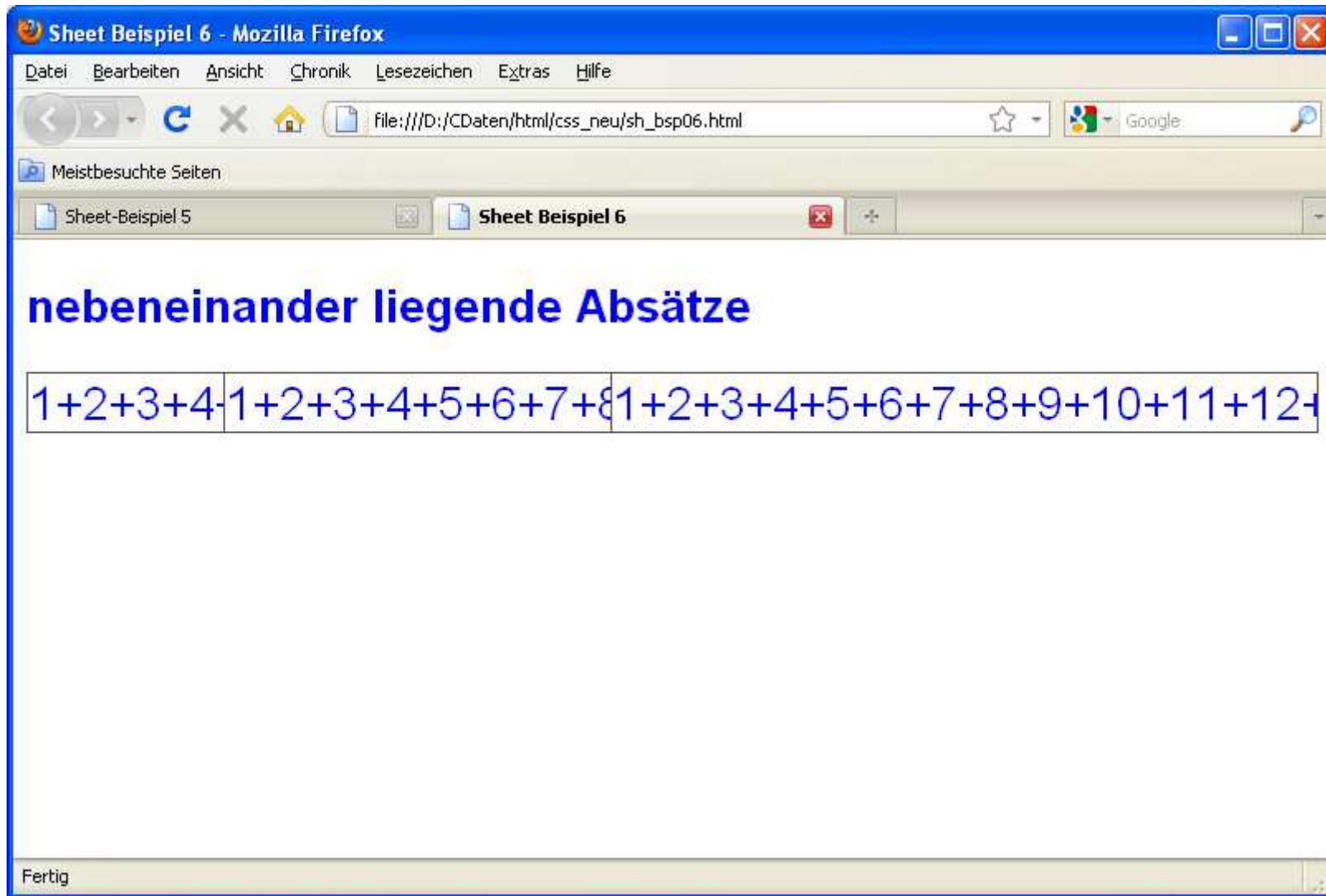
Modernes Layout



sh_bsp05.html: Bereiche nebeneinander



sh_bsp06.html: table-layout:fixed und overflow



```
<table border="1" style="width:100%; table-layout:fixed; border-collapse:collapse">
```

```
<tr>
```

```
<td style="width:15%; overflow:hidden; font-size:1.5em;">
```

```
1+2+3+4+5+6+7+8+9+10
```

```
</td>
```

```
<td style="width:30%; overflow:hidden; font-size:1.5em;">
```

```
1+2+3+4+5+6+7+8+9+10+11+
```

```
</td>
```

```
<td style="width:55%; overflow:hidden; font-size:1.5em;">
```

```
1+2+3+4+5+6+7+8+9+10+11+12+13+14
```

```
</td>
```

```
</tr>
```

```
</table>
```

sh_bsp06.html: table-layout:fixed und overflow

table-layout:fixed

- Problem ist der Text ohne White-Spaces
- Der Eintrag "fixed" verhindert, dass die Spalten den Text komplett anzeigen
- Die Breite muss aber pro Spalte angegeben werden
- Der Eintrag "border-collapse" zeichnet nur einen Rand pro Zelle. Beeinflusst also die Darstellungen von Zellen mit unterschiedlichen Rahmen
- Das overflow-Attribut schneidet die Einträge am Ende der Spalte ab

sh_bsp06.html: nebeneinander liegende Spalten

