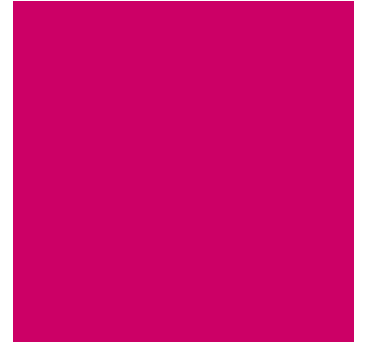




**HTW** Chur

Hochschule für Technik und Wirtschaft  
University of Applied Sciences



# Interaktive Medien 1. Semester

## Martin Vollenweider

### Web Entwicklung I

### Videsequenz: Layouten im HTML und CSS

HTML = Struktur

CSS = Formatierung

# Cascading Style Sheets Grundlagen

- CSS ist für das Layouten, Formatieren der Seite, Absätze und Zeichen zuständig.
- Die Dateien lassen sich extern (oder im Dokument) ablegen.

## CSS-Regeln: Was lässt sich mit CSS formatieren? = Selektor

Mit CSS können folgende Objekte formatiert werden:

- HTML-Tags: den im HTML fix gegebenen Formatierungen lassen sich Schriften, Farben etc. zuweisen. Beispiele für HTML-Tags: `h1`, `h2`, `p`, `table` und `body`. Hyperlinks sind ein Spezialfall!

Dies ist ein Typoblindtext. An ihm kann man sehen, ob alle Buchstaben da sind und wie sie aussehen. Manchmal benutzt man Worte wie Hamburgfont, Raifenduks oder Handgloves, um Schriften zu testen. Manchmal Sätze, die alle Buchstaben des Alphabets enthalten

HTML: `<p>`

CSS: `p`

- Klassen: eigene Stilnamen. Sie haben nichts mit HTML zu tun, sondern gehören zum CSS. Klassen haben im CSS einen Punkt vor dem Namen.

Dies ist ein Typoblindtext. An ihm kann man sehen, ob alle Buchstaben da sind und wie sie aussehen. Manchmal benutzt man Worte wie Hamburgefonts, Rafigenduks oder Handgloves, um Schriften zu testen. Manchmal Sätze, die alle Buchstaben des Alphabets enthalten

**HTML: <p>**

**CSS: p**

Dies ist ein Typoblindtext. An ihm kann man sehen, ob alle Buchstaben da sind und wie sie aussehen. Manchmal benutzt man Worte wie Hamburgefonts, Rafigenduks oder Handgloves, um Schriften zu testen. Manchmal Sätze, die alle Buchstaben des Alphabets enthalten

**HTML: <p class='farbe'>**

**CSS: .farbe**

Dies ist ein Typoblindtext. An ihm kann man sehen, ob alle Buchstaben da sind und wie sie aussehen. Manchmal benutzt man Worte wie Hamburgefonts, Rafigenduks oder Handgloves, um Schriften zu testen. Manchmal Sätze, die alle Buchstaben des Alphabets enthalten

**HTML: <p class='rahmen'> CSS: .rahmen**

- ID: Wie Klassen, dürfen aber nur einmal pro Datei oder Seite vorkommen. Die IDs haben im CSS eine Raute # vor dem Namen.

Dies ist ein Typoblindtext. An ihm kann man sehen, ob alle Buchstaben da sind und wie sie aussehen. Manchmal benutzt man Worte wie Hamburgefonts, Raifgenduks oder Handgloves, um Schriften zu testen. Manchmal Sätze, die alle Buchstaben des Alphabets enthalten	HTML: <p>	CSS: p
Dies ist ein Typoblindtext. An ihm kann man sehen, ob alle Buchstaben da sind und wie sie aussehen. Manchmal benutzt man Worte wie Hamburgefonts, Raifgenduks oder Handgloves, um Schriften zu testen. Manchmal Sätze, die alle Buchstaben des Alphabets enthalten	HTML: <p id='farbe'>	CSS: #farbe
Dies ist ein Typoblindtext. An ihm kann man sehen, ob alle Buchstaben da sind und wie sie aussehen. Manchmal benutzt man Worte wie Hamburgefonts, Raifgenduks oder Handgloves, um Schriften zu testen. Manchmal Sätze, die alle Buchstaben des Alphabets enthalten	HTML: <p id='rahmen'>	CSS: #rahmen
Dies ist ein Typoblindtext. An ihm kann man sehen, ob alle Buchstaben da sind und wie sie aussehen. Manchmal benutzt man Worte wie Hamburgefonts, Raifgenduks oder Handgloves, um Schriften zu testen. Manchmal Sätze, die alle Buchstaben des Alphabets enthalten	HTML: <p id='rahmen'>	CSS: #rahmen

## Regel

- IDs nur sparsam einsetzen. Nur wenn ein Element pro Seite nur einmal vorkommt: <body id='deutsch'>

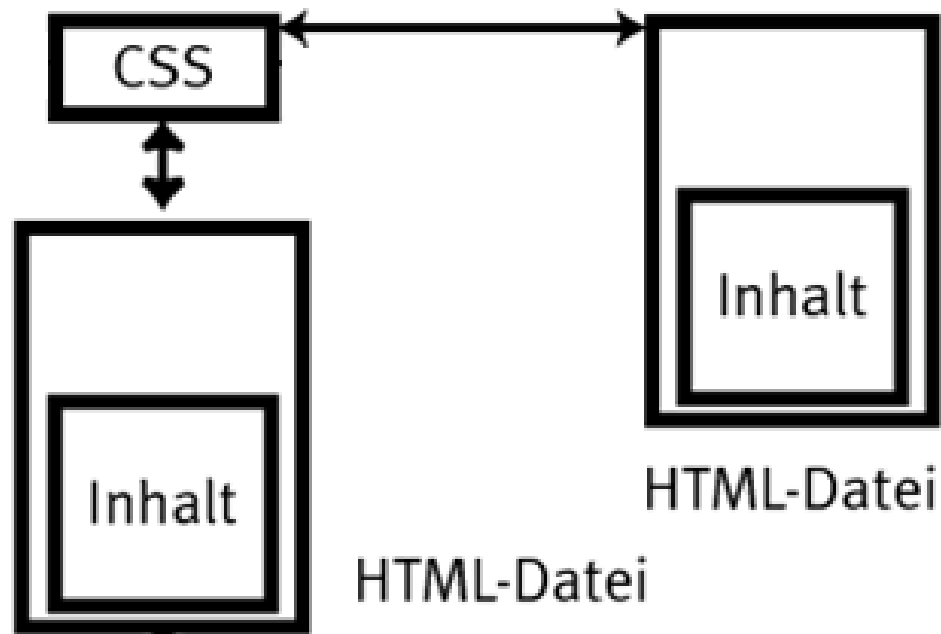
- Zusammengesetzter Ausdruck: Kombination obiger Möglichkeiten und Formatieren von Hyperlinks.

body article p em: Formatierung des Tags `<em>` nur innerhalb `<p>`, welches sich innerhalb `<article>` befinden muss.

```
<body>
  <article>
    <p>Dieses <em>Wort</em> im Artikel...</p>
  </article>
</body>
```

# Wie hängen HTML und CSS zusammen?

Bei uns ist die externe CSS-Datei mit der Vorlage verknüpft:

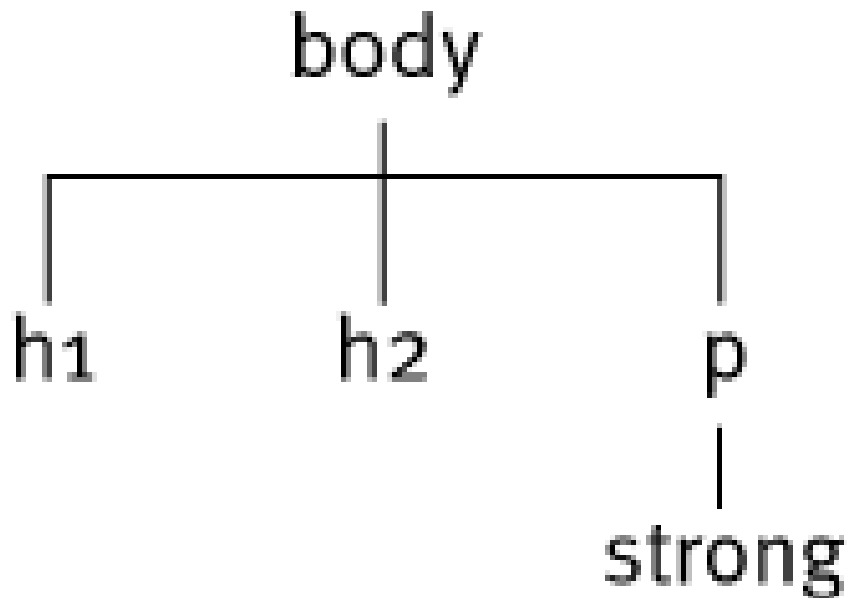


- Per Dreamweaver oder per Code

# Vererbung

Die meisten Formatierungen, welche `body` zugewiesen werden vererben sich auf die untergeordneten Tags:

- Aus diesem Grunde muss zuerst `body` definiert, resp. angepasst werden!



`strong` erbt von `p`

`p` erbt von `body`

`body` vererbt an `h1`, `h2` und `p`

`p` vererbt an `strong`

Warum erben wir so selten? :-)



# Wie herausfinden? Antwort: Referenz

## Zeilenhöhe: **line-height**

Diese Eigenschaft gibt es seit Version: CSS 1

Durch **line-height** wird die Zeilenhöhe festgelegt. Sind mehrere Zeilen untereinander vorhanden, so wird der Durchschuß festlegen.

Ist der Wert von line-height dabei kleiner als die aktuelle Schriftgröße (font-size), so überläßt es die Zeilenhöhe. Erlaubte Werte sind **Zahlen**, **Längenangaben** und **Prozentwerte**. Im folgenden Beispiel wird

### Code-Beispiel für **line-height**:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>Beispiel für Zeilenhöhe</title>
    <style type="text/css">
      .zahl {line-height: 1.5;}
      .prozent {line-height: 150%;}
      .em {line-height: 1.5em;}
    </style>
  </head>
  <body>
    <p class="zahl">Ein Absatz mit einer Zeilenhöhe von 1.5.</p>

    <p class="prozent">Ein Absatz mit einer Zeilenhöhe von 150%.</p>
    <p class="em">Ein Absatz mit einer Zeilenhöhe von 1.5em. </p>
  </body>
</html>
```

### Beispiel ansehen

Initialwert: **normal**  
Vererbbar: **Ja**

## CSS **line-height** Property

[« Previous](#)

[Complete CSS Reference](#)

### Example

Set the line height in percent:

```
p.small {
  line-height: 90%;
}

p.big {
  line-height: 200%;
}
```

[Try it yourself »](#)

More "Try it Yourself" examples below.

### Definition and Usage

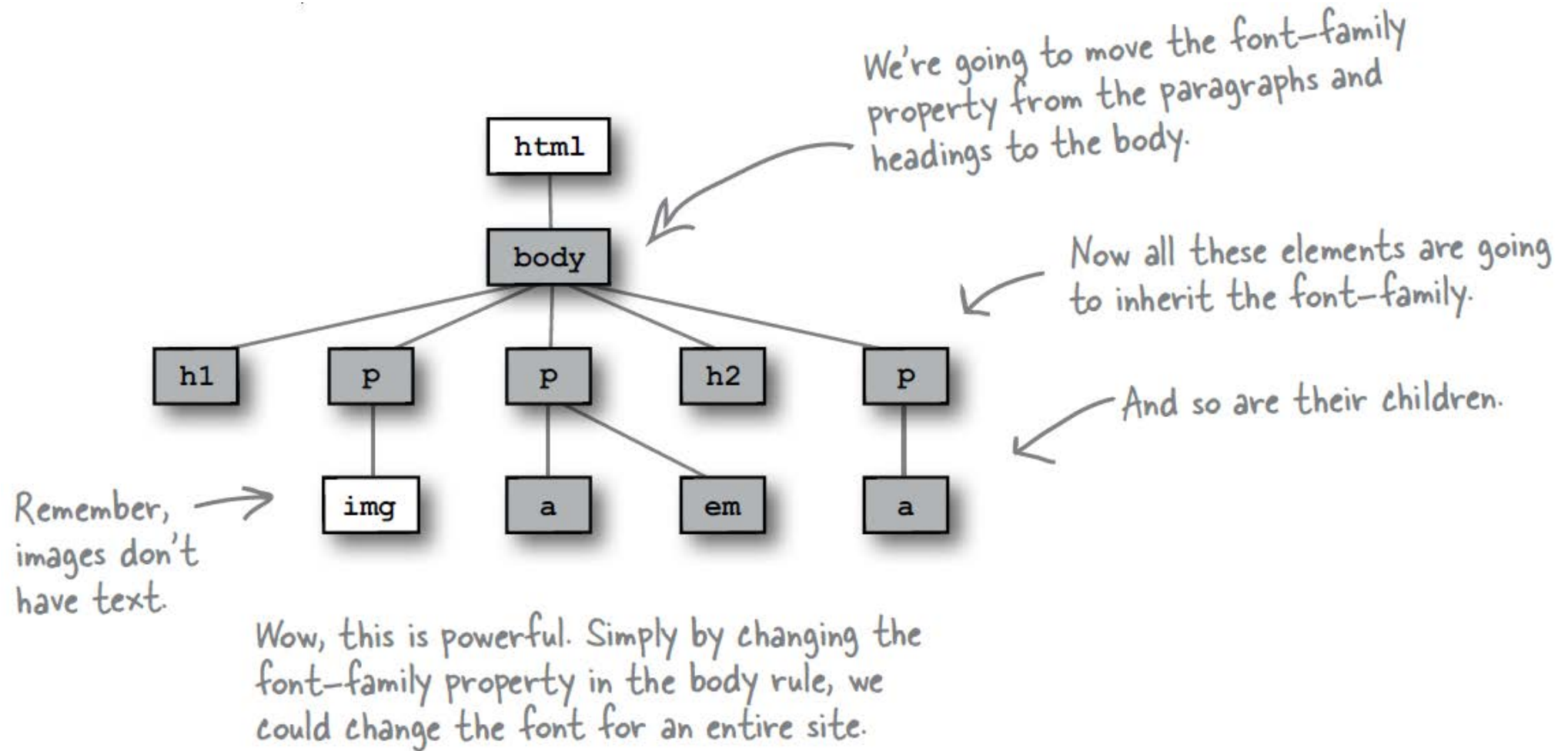
The line-height property specifies the line height.

**Note:** Negative values are not allowed.

Default value:	normal
Inherited:	yes

[CSS4You](#)

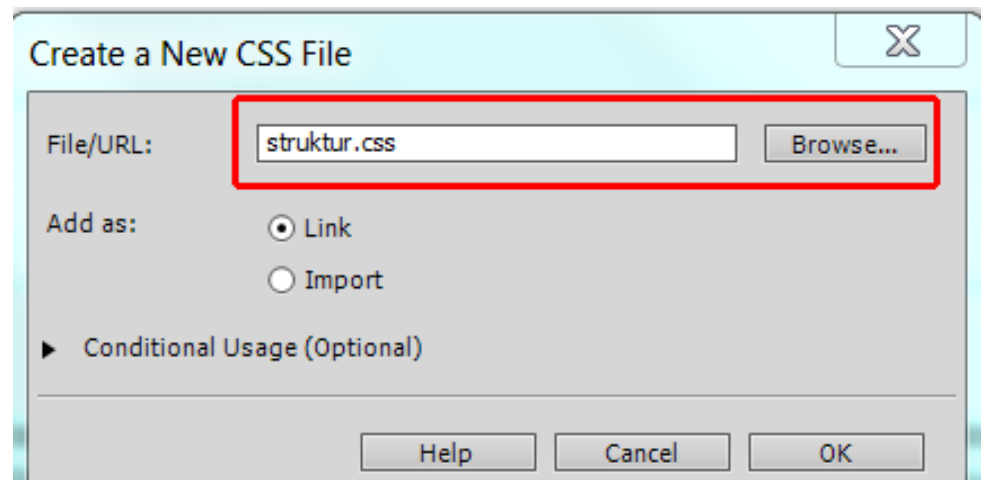
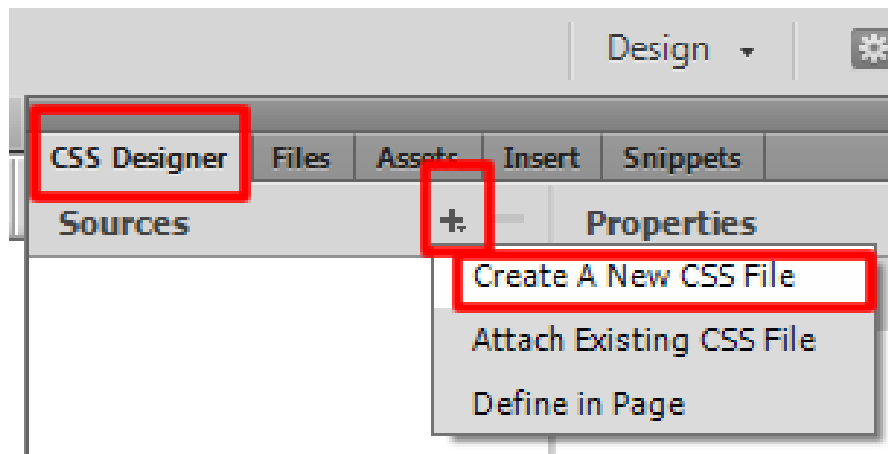
[W3Schools](#)



Fallstrick: font-size ist vererbbar, nicht aber zum Beispiel background-color

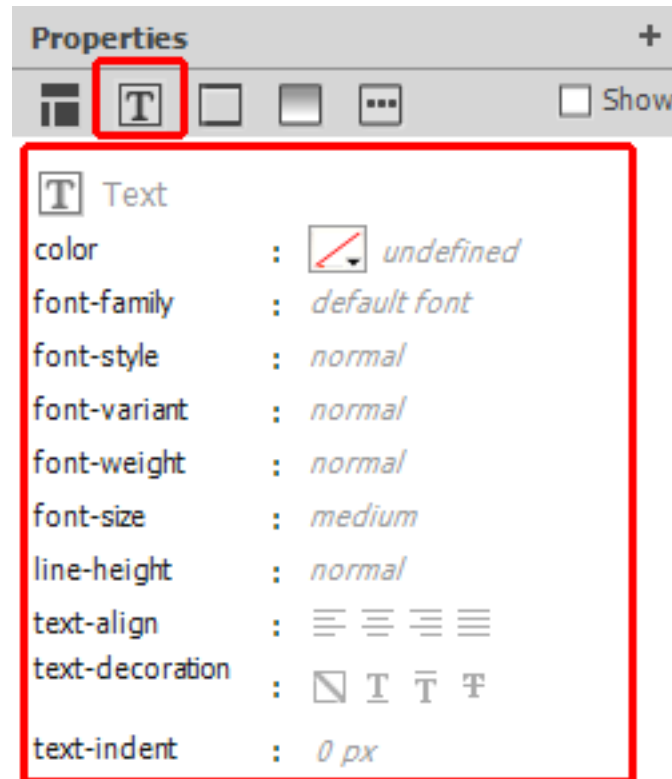
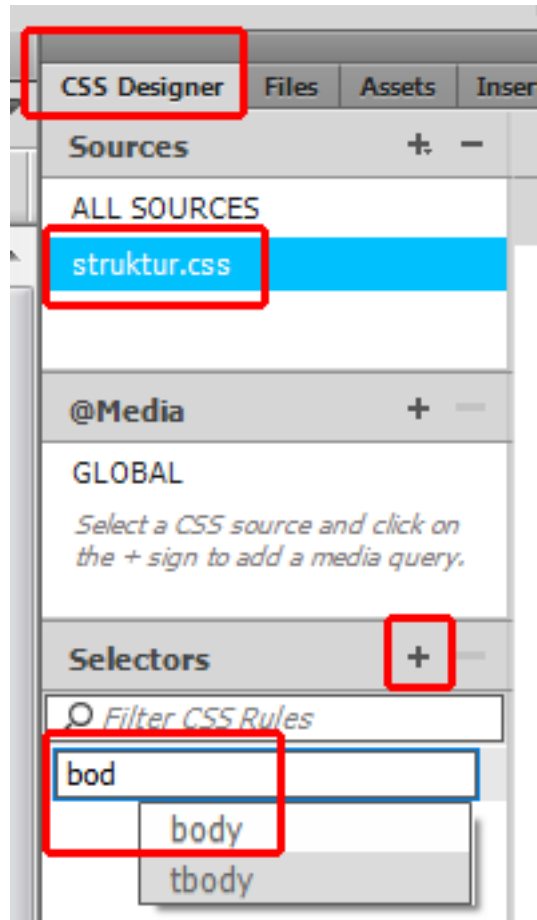
# Vorgehen in Dreamweaver

1. Site erstellen oder bestehende Site öffnen
2. HTML Dokument in Site speichern
3. Neue CSS Datei erstellen:



```
<head>  
  <meta charset="utf-8">  
  <title>Semantische Website</title>  
  <link href="struktur.css" rel="stylesheet" type="text/css">  
</head>
```

# body formatieren



```
body {  
    color: #000000;  
    font-size: 16px;  
}
```

Details der  
Formatierung  
später ... und im  
Visualisieren!!

# CSS-Code

## Die drei Grundelemente der CSS Regeln

```
body { background:#cccccc; }
```

**Selektor** { **D e k l a r a t i o n** }  
**Eigenschaft** **Wert**

Nach Wolfgang Bock

- Vergessen Sie den ; (Strichpunkt) nicht!

[Liste](#)

Mac:

{ = ALT + 8

} = ALT + 9

Windows:

{ = ALT GR + ä

} = ALT GR + \$

# Aufbau



# Tag



Nach Wolfgang Bock

# Klassen

Classes: wirkt sich auf alle, als Class definierten Tags aus

```
<p class="note"> Inhalte </p>
```

HTML

```
.note { background:#ffffff; }
```

CSS



# IDs

IDs: wirkt sich nur auf ein einzelnes Tag, das die ID besitzt aus. Eine ID darf nur ein Mal innerhalb einer HTML-Datei vergeben werden.

```
<div id="fusszeile"> Inhalte </div>
```

HTML

```
#fusszeile { color:#ffffff; }
```

CSS

Nach Wolfgang Bock

# Verboten im Code

```
body {  
  width: 400 px;  
}
```

**Leerschlag zwischen Zahl  
und Einheit!**

```
body {  
  width: 400px  
}
```

**Kein Strichpunkt am  
Zeilenende!**

```
body {  
  width: 400;  
}
```

**Keine Einheit!**

# Jetzt geht es um das Layouten!

Unser erstes Beispiel

**Hallo Welt, das ist HTML**

**Untertitel**

Das ist Fliesstext

Gewünschte Formatierung



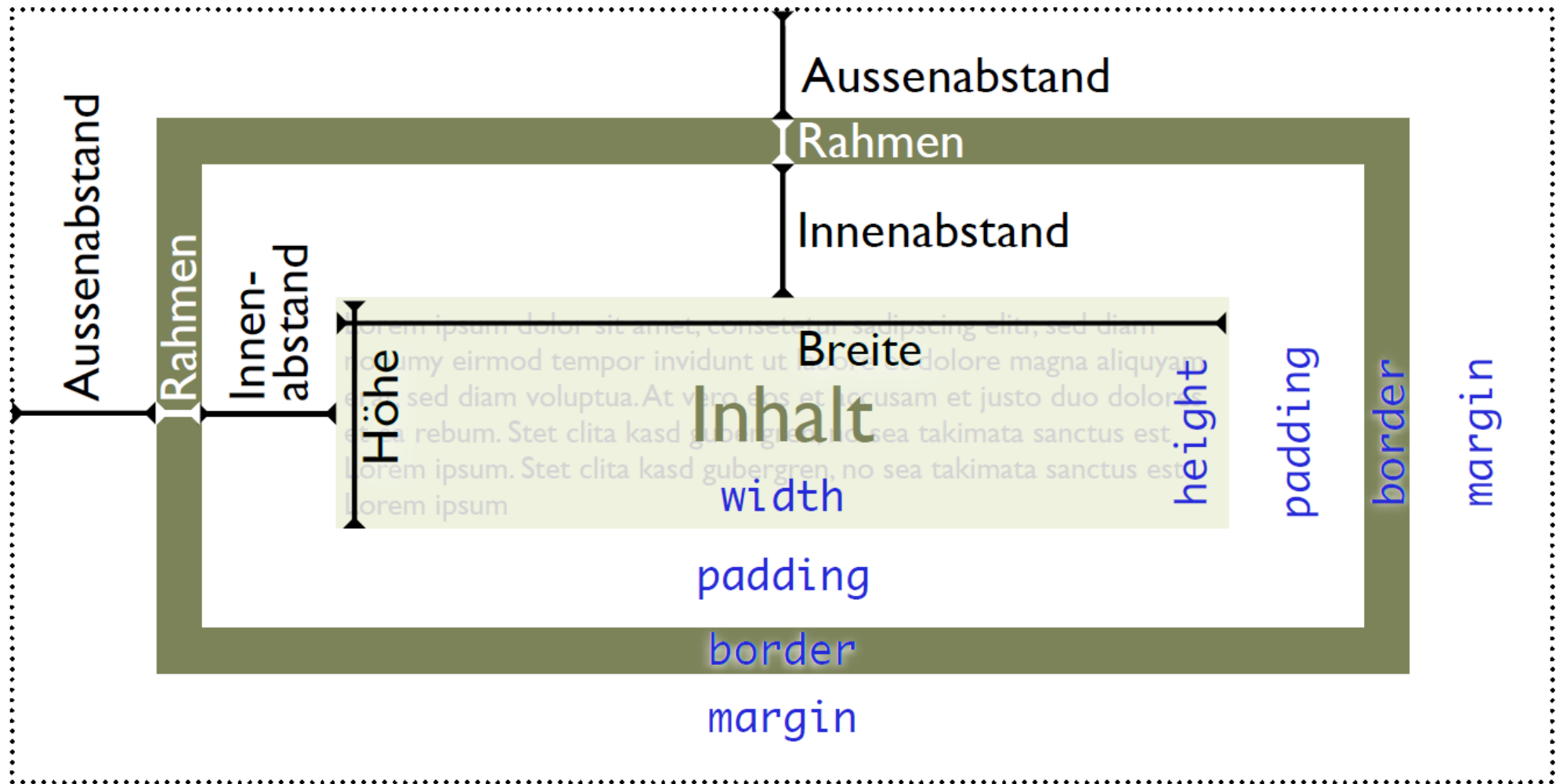
## Fragen

Warum schreibt HTML alles untereinander?

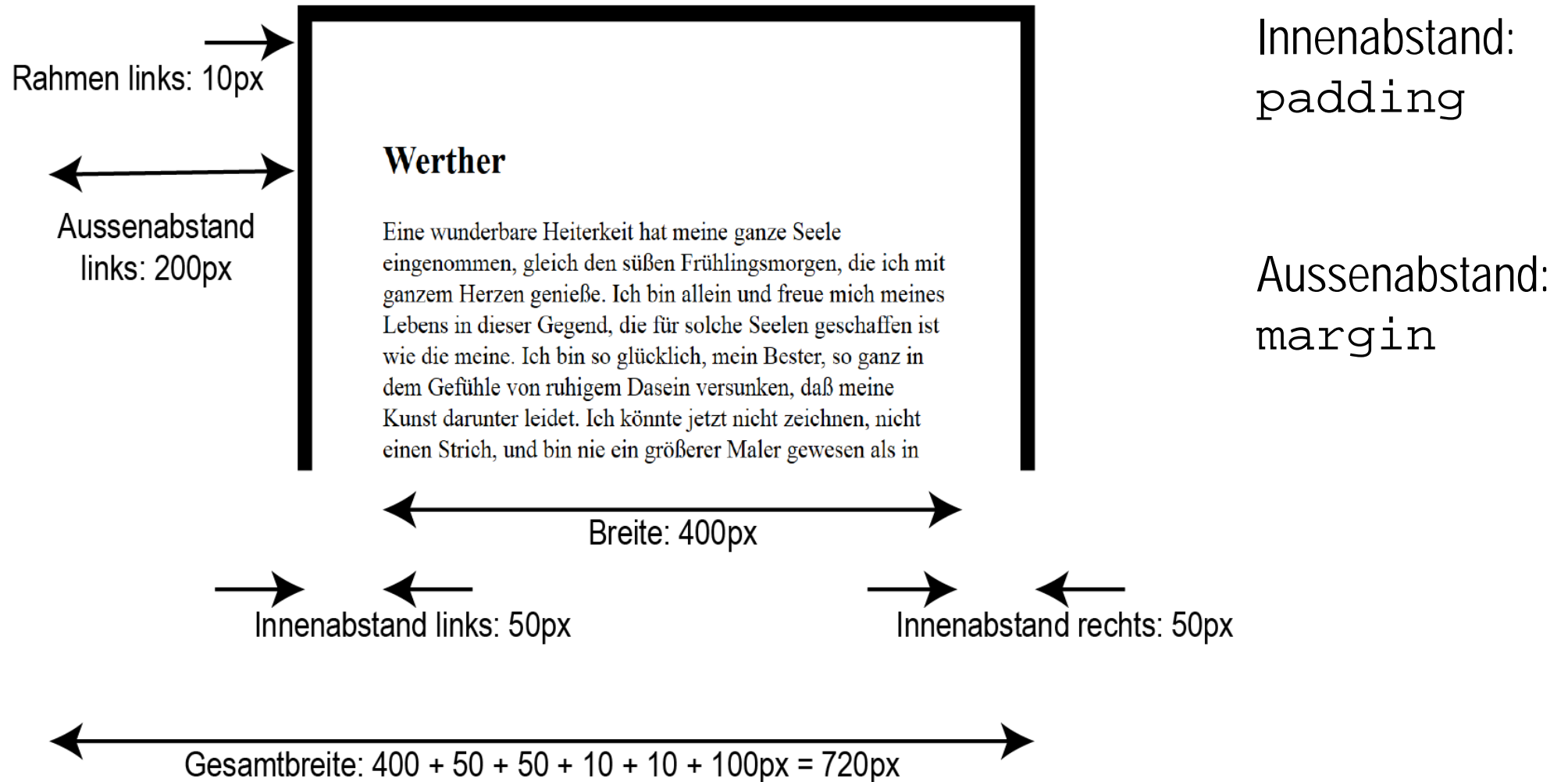
Wie kann ich Elemente nebeneinander positionieren?

Wie kann ich layouten?

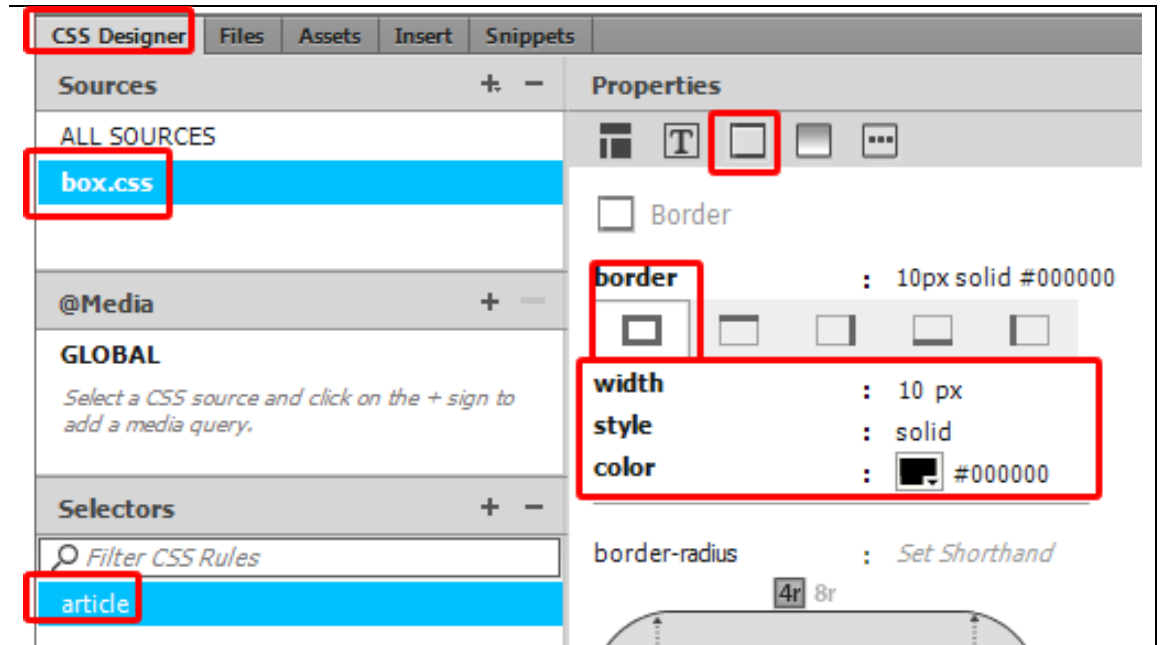
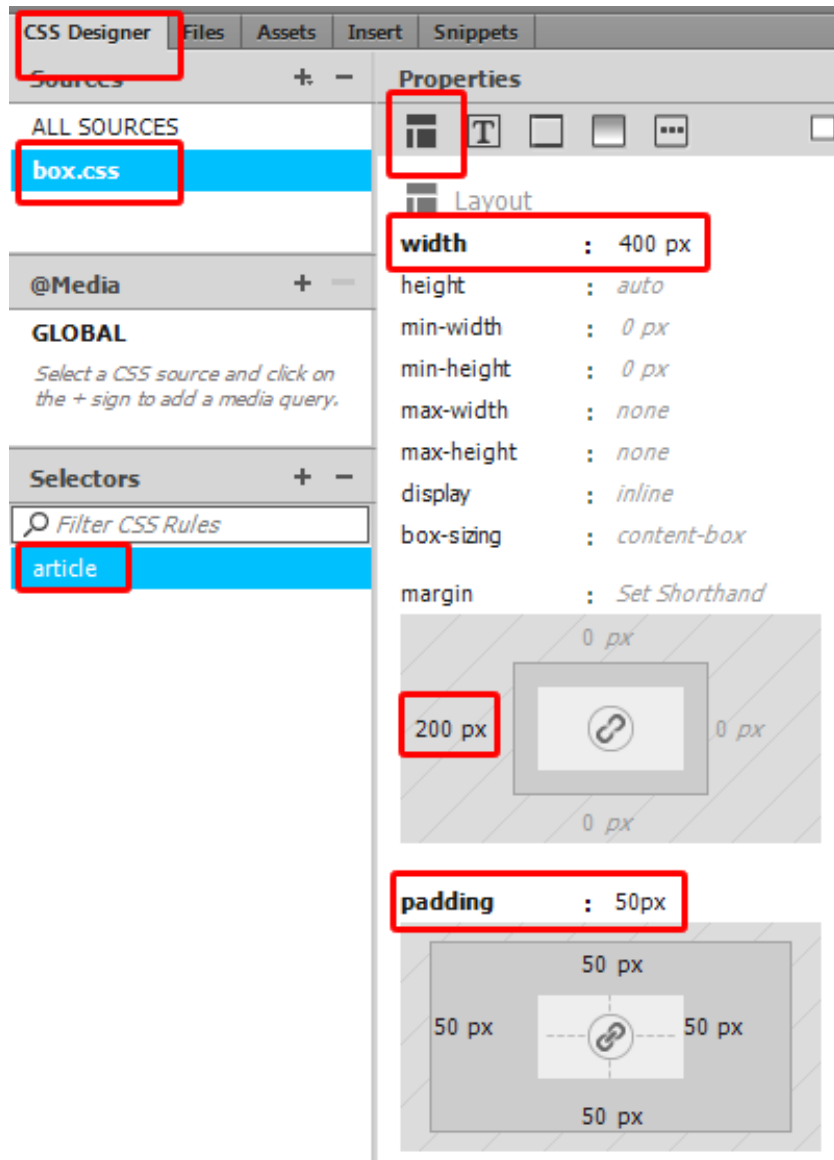
# Abstände her => Boxmodell 1 «content»



# Beispiel



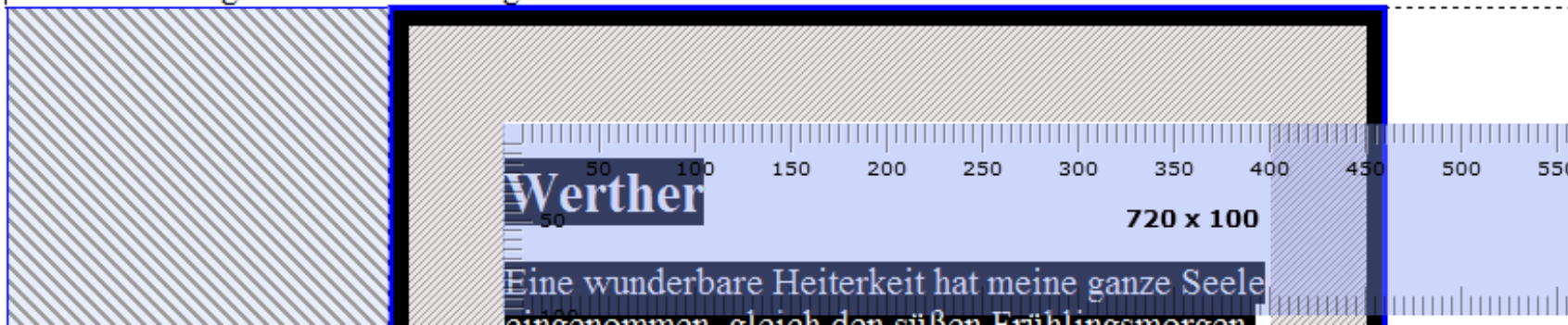
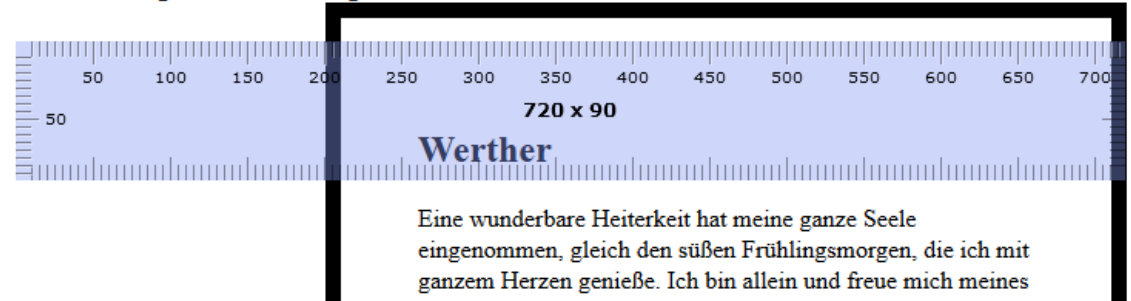
Die Gesamtbreite ergibt sich aus der Breite, den Innenabständen links und rechts, den Rahmendicken links und rechts sowie den Aussenabständen links und rechts.



# Code

```
article {  
    padding: 50px;  
    width: 400px;  
    border: 10px solid  
#000000;  
    margin-left: 200px;  
    margin-right: 0px;  
}
```

Header für Logo und Bezeichnung

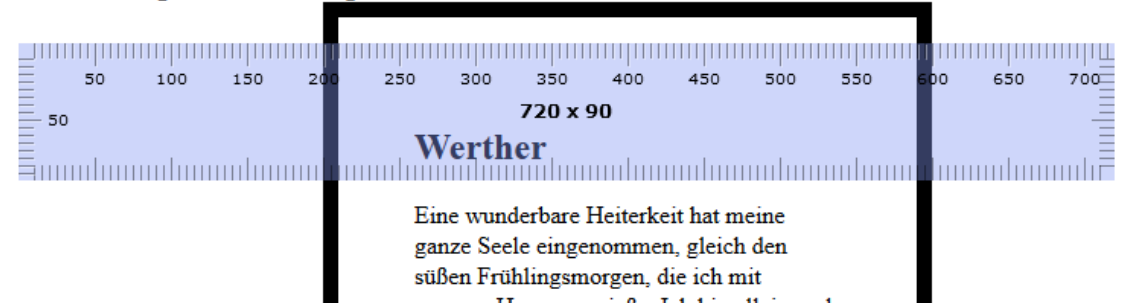


- Dieses Modell heisst auch `content-box`, weil die Breite des Contents festgelegt wird.

# Abstände her => Boxmodell 2 «border»

```
article {  
  padding: 50px;  
  width: 400px;  
  border: 10px solid #000000;  
  margin-left: 200px;  
  margin-right: 0px;  
  box-sizing: border-box;  
}
```

Header für Logo und Bezeichnung



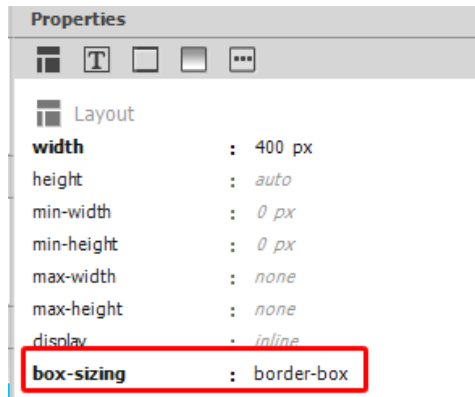
Header für Logo und Bezeichnung



- Dieses Modell heisst auch border-box, weil die Breite der Box inklusive Rand festgelegt wird.



# Unterschiede der Boxmodelle



- Keine Einstellung von `box-sizing` (oder im Code weglassen):  
`content-box`  
=> `width` bezieht sich auf die Breite des Inhalts.  
=> Die Breite des Kastens muss von Hand berechnet werden.
- Einstellung von `box-sizing`: `border-box: border-box`  
=> `width` bezieht sich auf die Breite des Kastens (inklusive Rahmen).  
=> Die Breite des Inhalts muss von Hand berechnet werden.

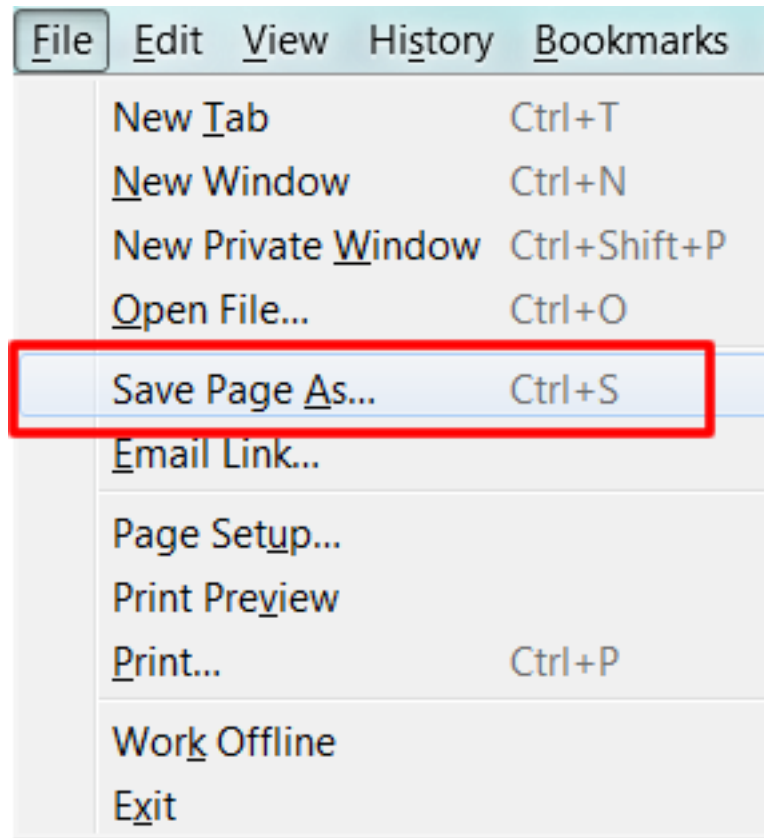
# Interaktive Medien 1. Semester

## Martin Vollenweider

### Web Entwicklung I

### HTML-Tags

# Code analysieren



Am Beispiel <http://www.watson.ch/>

# Tags

## Unterscheidung

- Physische, Layout oder Struktur Tags
- Semantische Tags

# Physische, Layout oder Struktur Tags

- Struktur, Präsentation
- `<h1>` gibt Struktur an; `<b>` macht einen Teil in fett; `<i>` in kursiv; `<sup>` setzt einen Text hochgestellt; `<sub>` setzt einen Text tiefgestellt

Element	HTML4 definition	HTML5 definition (taken from the spec on May 12, 2010)
<i>	Renders as italic text style	“The i element represents a span of text in an alternate voice or mood, or otherwise offset from the normal prose, such as a taxonomic designation, a technical term, an idiomatic phrase from another language, a thought, a ship name, or some other prose whose typical typographic presentation is italicized.”
<b>	Renders as bold text style	“The b element represents a span of text to be stylistically offset from the normal prose without conveying any extra importance, such as key words in a document abstract, product names in a review, or other spans of text whose typical typographic presentation is boldened.”

- Spüren Sie die Verunsicherung der HTML5 Definition?

# Semantische Tags

- Semantik = «von Bedeutung / in Bezug auf die Bedeutung»
- Semantik = Bedeutung eines Elements
- Wie Element den Inhalt beschreibt, den es enthält
- Sinn und Zweck: Suchmaschinen, andere Ausgabegeräte, Zusatzinformationen
- `<em>` meint emphasis

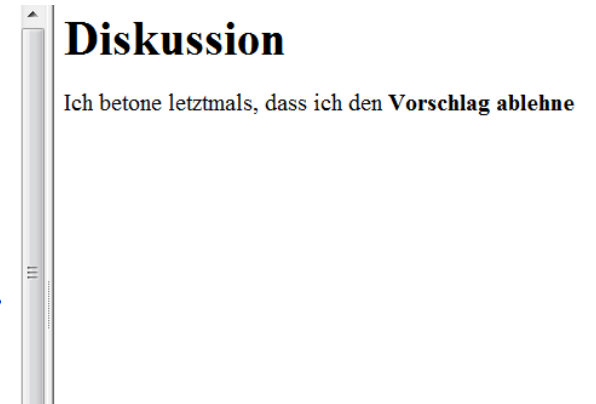
```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Website</title>
  </head>
  <body>
    <h1>Diskussion</h1>
    <p>Sie sagte, sie komme <em>nicht</em> mit</p>
  </body>
</html>
```

## Diskussion

Sie sagte, sie komme *nicht* mit

- `<strong>` meint starke Auszeichnung

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Website</title>
  </head>
  <body>
    <h1>Diskussion</h1>
    <p>Ich betone letztmals, dass ich den <strong>Vorschlag ablehne</strong></p>
  </body>
</html>
```



- Es geht um die Sprache!
- Ein zukünftiges Gerät betont `<i>`, `<em>` und `<b>`, `<strong>` unterschiedlich





<q> und <blockquote> für Zitate:

Er sagte, «ich denke, also bin ich» und trank eine Tasse Kaffee

Er sagte, "ich denke, also bin ich" und trank eine Tasse Kaffee

Er sagte, “ich denke, also bin ich” und trank eine Tasse Kaffee

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Website</title>
  </head>
  <body>
    <h1>Diskussion</h1>
    <p>Er sagte, <q>ich denke, also bin ich</q> und trank eine Tasse Kaffee</p>
  </body>
</html>
```

## Diskussion

Er sagte, ich denke, also bin ich und trank eine Tasse Kaffee

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Website</title>
  </head>
  <body>
    <h1>Diskussion</h1>
    <p>Er sagte, </p><blockquote>ich denke, also bin ich</blockquote><p>und trank Kaffee</p>
  </body>
</html>
```

## Diskussion

Er sagte,  
ich denke, also bin ich  
und trank Kaffee

# Beispiel Unterschiede <i> und <em>

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Website</title>
  </head>
  <body>
    <h1>Diskussion</h1>
    <p>Sie sagte, sie komme <em>nicht</em> mit</p>
    <p>Sie sagte, sie reise mit der <i>Titanic</i></p>
  </body>
</html>
```

## Diskussion

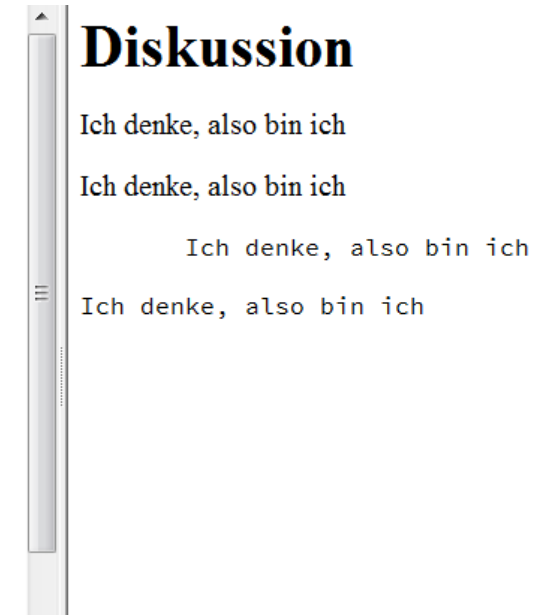
Sie sagte, sie komme *nicht* mit

Sie sagte, sie reise mit der *Titanic*

# Weissraum

**Gewünschter Weissraum muss mit `<pre>` formatiert werden.**

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Website</title>
  </head>
  <body>
    <h1>Diskussion</h1>
    <p>      Ich denke, also bin ich</p>
    <p>Ich denke, also bin ich</p>
    <pre>      Ich denke, also bin ich</pre>
    <pre>Ich denke, also bin ich</pre>
  </body>
</html>
```



- There's no such thing as a free lunch  
=> There's no such thing as a tab or several spaces

# Parsing und Rendering

## Rendering: Text als grafische Darstellung

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Website</title>
  </head>
  <body>
    <h1>Diskussion</h1>
    <p>Er sagte, <q>ich denke, also bin ich</q> und trank eine Tasse Kaffee</p>
  </body>
</html>
```

### Diskussion

Er sagte, ich denke, also bin ich und trank eine Tasse Kaffee

## Parsing: Text lesen, analysieren

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Website</title>
  </head>
  <body>
    <h1>Diskussion</h1>
    <p>Er sagte, <q>ich denke, also bin ich</q> und trank eine Tasse Kaffee</p>
  </body>
</html>
```

# Listen

## Ungeordnete Listen

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Website</title>
  </head>
  <body>
    <h1>Diskussion</h1>
    <ul>
      <li>Ein Modul ist Bestandteil eines Semesters</li>
      <li>Ein Kurs ist Bestandteil eines Moduls</li>
      <li>Eine Lektion ist Bestandteil eines Kurses</li>
    </ul>
  </body>
</html>
```

## Diskussion

- Ein Modul ist Bestandteil eines Semesters
- Ein Kurs ist Bestandteil eines Moduls
- Eine Lektion ist Bestandteil eines Kurses

# Geordnete Listen

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Website</title>
  </head>
  <body>
    <h1>Diskussion</h1>
    <ol>
      <li>Ein Modul ist Bestandteil eines Semesters</li>
      <li>Ein Kurs ist Bestandteil eines Moduls</li>
      <li>Eine Lektion ist Bestandteil eines Kurses</li>
    </ol>
  </body>
</html>
```

## Diskussion

1. Ein Modul ist Bestandteil eines Semesters
2. Ein Kurs ist Bestandteil eines Moduls
3. Eine Lektion ist Bestandteil eines Kurses

# Es kann auch schön kompliziert werden: Verschachtelung

```
<body>
  <h1>Diskussion</h1>
  <ol>
    <li>Einkaufen
      <ol>
        <li>Laden evaluieren
          <ol>
            <li>Öffnungszeiten
              <ol>
                <li>Bedingung: Am Freitagabend offen</li>
              </ol>
            </li>
            <li>Lage
              <ol>
                <li>Mit ÖV erreichbar</li>
              </ol>
            </li>
          </ol>
        </li>
        <li>Einkaufsliste erstellen</li>
      </ol>
    </li>
    <li>Kochen</li>
    <li>essen</li>
  </ol>
</body>
```

## Diskussion

1. Einkaufen
  1. Laden evaluieren
    1. Öffnungszeiten
      1. Bedingung: Am Freitagabend offen
    2. Lage
      1. Mit ÖV erreichbar
  2. Einkaufsliste erstellen
2. Kochen
3. essen

# Navigation

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Website</title>
  </head>
  <body>
    <h1>Diskussion</h1>
    <p>Sie sagte, sie komme <em>nicht</em> mit</p>
    <p>Die <a href="http://www.htwchur.ch">HTW</a> bietet viel</p>
  </body>
</html>
```

## Diskussion

Sie sagte, sie komme *nicht* mit

Die [HTW](http://www.htwchur.ch) bietet viel

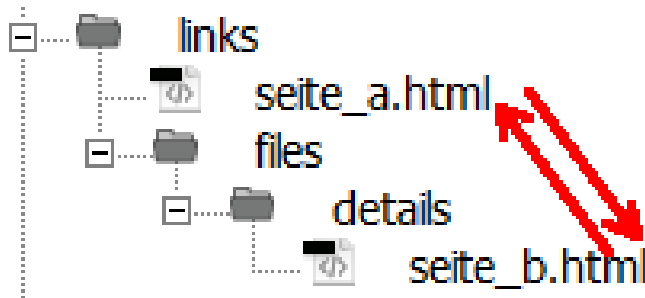
## Absolute und relative Links

- Absolute Links mit URL: *http://www....*
- Relative Links mit Pfad und Dateiname: *logo.gif*



## Relative Links: A.html <-> B.html

Wie gelange ich von A nach B und von B nach A?



*Code Seite A:*

```
<a href="files/details/seite_b.html">Seite B</a>
```

*Code Seite B:*

```
<a href="../../seite_a.html">Seite A</a>
```

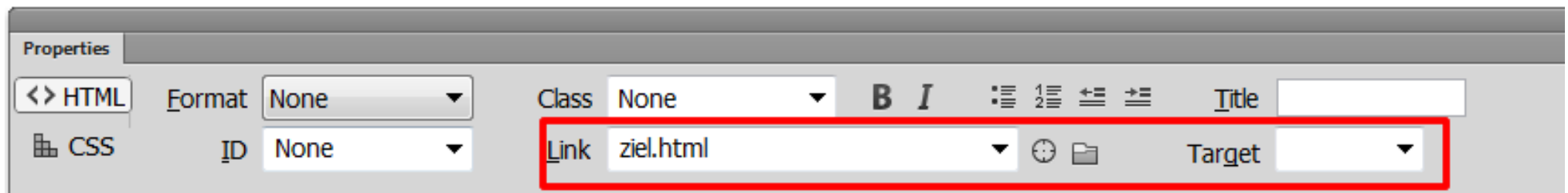
## Link in neuem Fenster / neuer Registerkarte öffnen

Attribut: target, Attributwert: \_blank

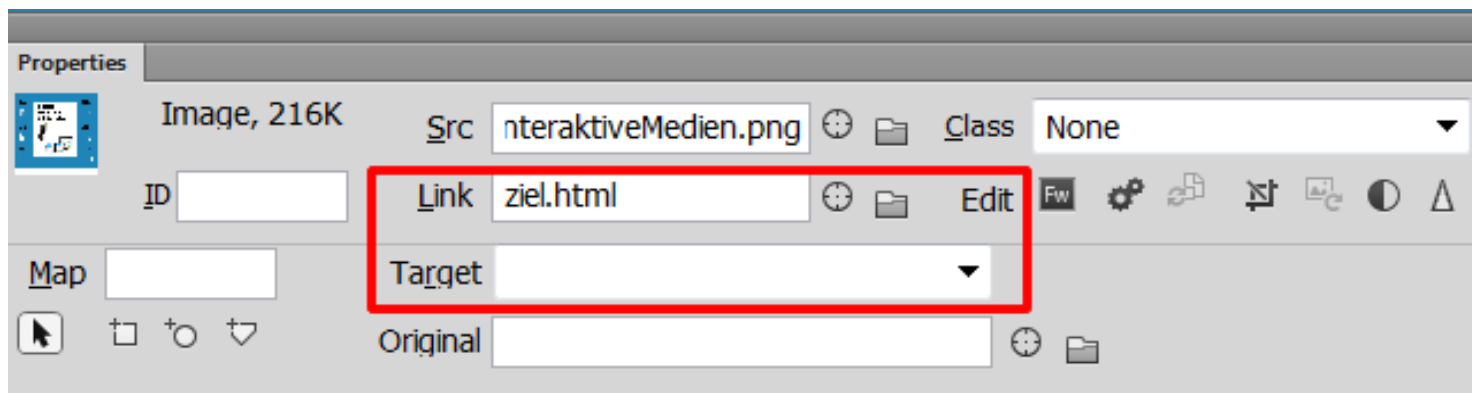
```
<a href="seite_b.html" target="_blank">Seite B</a>
```

## Dreamweaver

Hyperlinks lassen sich auf markierte Texte oder Bilder legen:

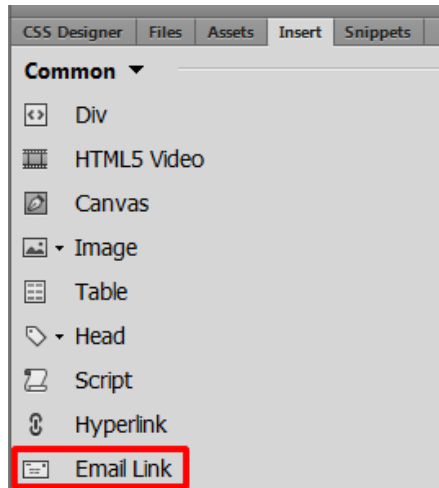


Bei markierten Bildern ist der Hyperlink in der Mitte der Properties:

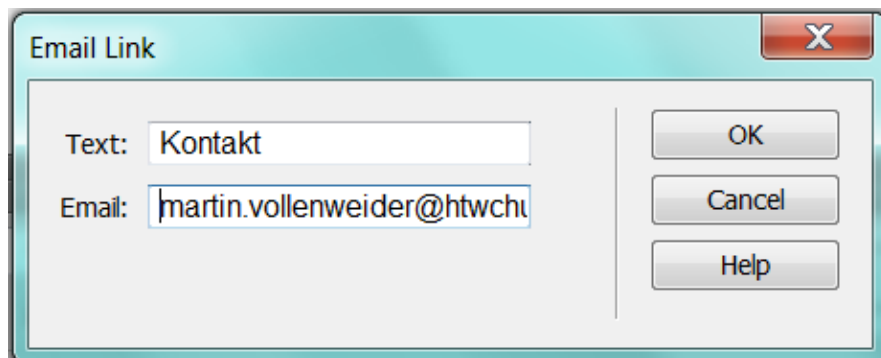


# Mail Verknüpfung

Registerkarte **Common** > **Email Link**:



Es öffnet sich folgendes Fenster:



```
<a href="mailto:martin@timerate.ch">Kontakt</a>
```

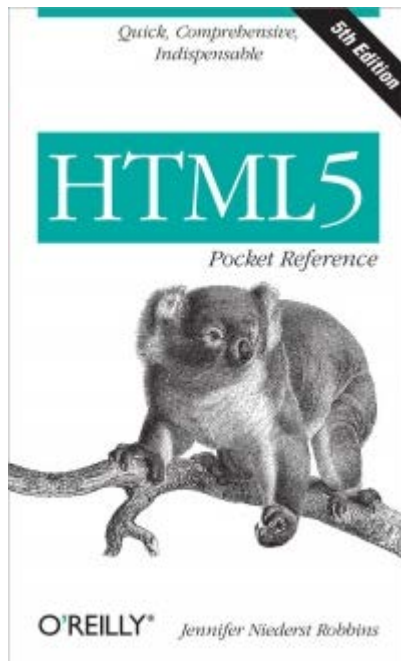
# Referenzen



HTML Tutorial

HTML Tag Reference

<http://www.w3schools.com/>



<http://it-ebooks.info/book/2516/>

# HTML5 = HTML

- HTML5-[Spezifikationen](#) + diverse Technologien, die unter HTML5 zusammengefasst werden, damit aber nichts zu tun haben (Beispiel: Geolocation)

## Neue Regelung

HTML5 => HTML
---------------

# Browser Unterstützung HTML5

- [Caniuse.com](http://caniuse.com)

The screenshot shows the Caniuse.com website. At the top, it says "Can I use..." with links for "Suggestions", "Feed", and "Twitter". A counter shows "881" and a "Flattr" button. Below this is a banner for "Latest update: New site features available for preview (October 15, 2013)". Navigation links include "AdChoices", "HTML5 Browser", "HTML5 SlideShow", "HTML5 for Mobile", and "HTML5 Development". A search bar contains the text "border-radius, WebGL, woff, etc". On the left, there are tabs for "Index" and "Tables". On the right, there are links for "Import stats", "FAQ", "Resources", and "Embed". The main content area is divided into three columns: "CSS", "HTML5", and "SVG". The "CSS" column lists features like "@font-face Web fonts", "calc() as CSS unit value", "2.1 selectors", and "Counters". The "HTML5" column lists "Audio element", "Canvas (basic support)", "Canvas blend modes", and "Color input type". The "SVG" column lists "Inline SVG in HTML5", "SVG (basic support)", "SVG effects for HTML", and "SVG filters".

- [Hazio](http://haz.io)