

CSS zur Gestaltung von HTML-Seiten nutzen

Handbuch und Literatur

- Handbuch des RRZNs: CSS Cascading Style Sheet; Level 2
- Charles Wyke-Smith: Stylin with CSS; Addison-Wesley
- Tommy Olsson & Paul O'Brien: The Ultimate CSS Reference; O'Reilly
- Daniel Koch: CSS: schnell & kompakt; entwickler.press
- Jens Kilgenstein: CSS einfach erklärt; Books on demand

Informationen im Web

- <http://www.w3.org/Style/CSS/>
- <http://de.selfhtml.org/css/index.htm>
- <http://www.thestyleworks.de/basics/index.shtml#basics1>
- <http://www.css4you.de/>
- <http://www.de.html.net/tutorials/css/introduction.asp>
- <http://jendryschik.de/wsdev/einfuehrung/>
- <http://www.html-seminar.de/style-sheet.htm>
- <http://barrierefrei.e-workers.de/workshops/tricks/menues.php>

C(ascading) S(tyle) S(heets)

- Formatierung und Positionierung von HTML und XHTML-Elementen.
- Formatvorlagen für den Inhalt von Webseiten.
- Wie wird das Element dargestellt?
- Regeln für das Layout bestimmter Elemente.

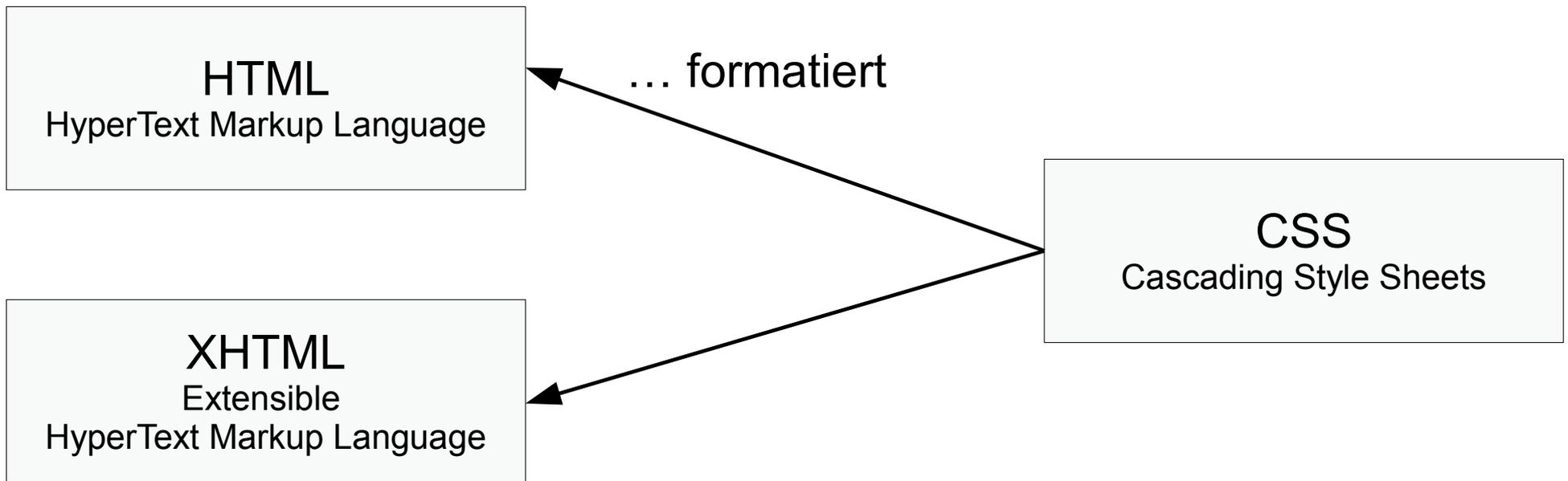
Version

- Version 2 (CSS2) seit 12.05.1998
 - Internet Explorer 6.x bindet CSS2 vollständig ein, liefert aber manchmal merkwürdige Ergebnisse.
 - Ab Internet Explorer 7.x wird CSS2 vollständig eingebunden.
 - Firefox 3.x bindet CSS2 vollständig ein.
- Version 3 (CSS3) wurde am 23.05.2001 verabschiedet und wird von allen Browsern nicht vollständig unterstützt.
- Informationen zur Browser-Kompatibilität finden Sie auf folgenden Webseiten:
 - <http://www.css-hack.de/>
 - <http://www.webdevout.net/browser-support-css>
 - <http://www.quirksmode.org/css/contents.html>

Vorteile

- Formatierung und Strukturierung des Dokuments werden getrennt.
- Formatierungsanweisungen können für verschiedene Ausgabemedien oder Browser genutzt.
- Bessere Wartbarkeit von Webseiten.
 - Formatierungen werden schneller gefunden.
 - Formatierungen müssen nur an einer Stelle geändert werden.
- Wiederverwendbarkeit der Formatierungen.

HTML und CSS



... strukturieren das Dokument

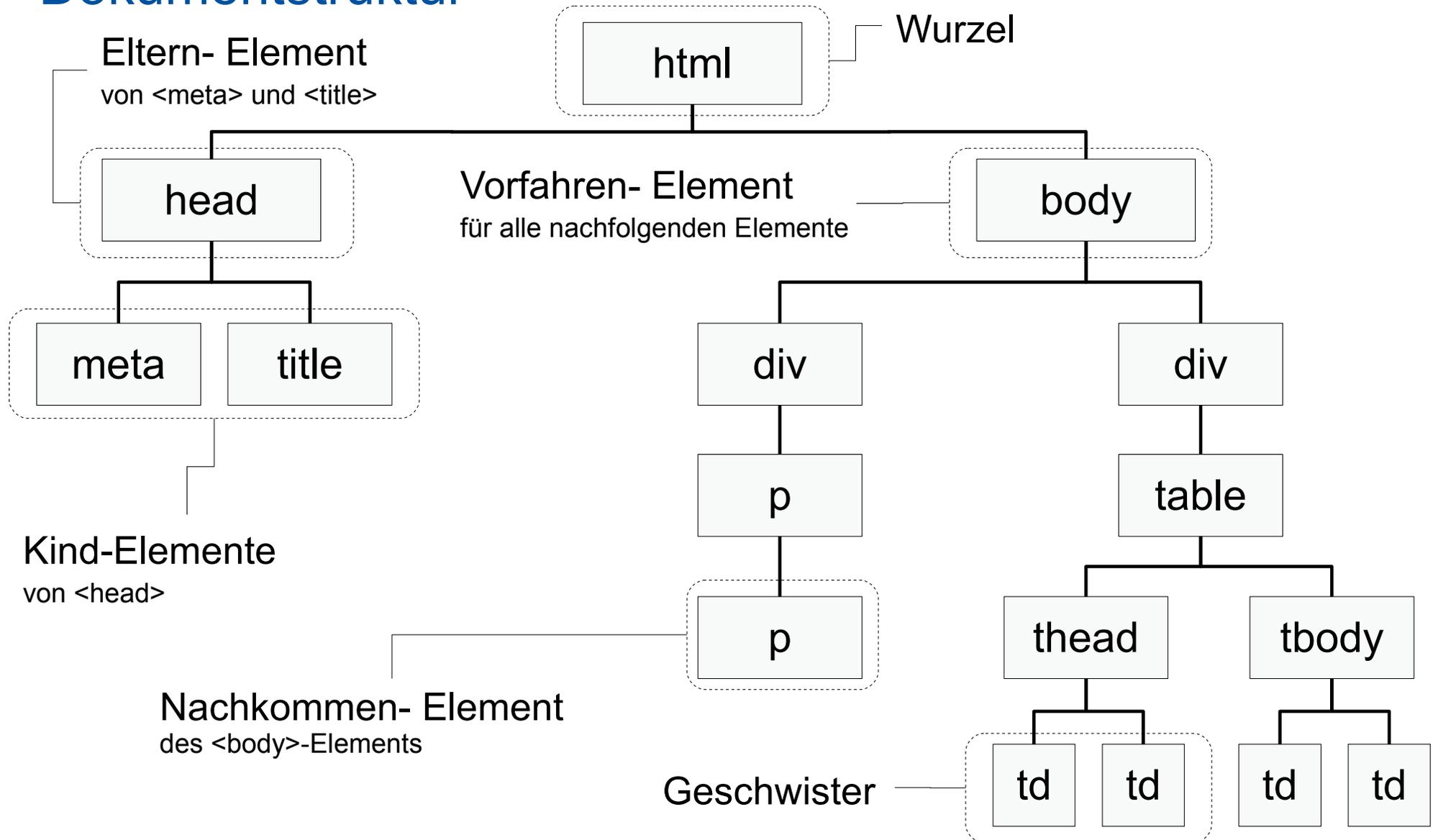
Struktur eines HTML-Elements

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
2 <html lang="de">
3 <head>
4 <title></title>
5 <meta name="author" http-equiv="content-type" content="text/html; charset=iso-8859-1">
6 </head>
7 <body>
8 <div id="main">
9 <p>
10 Autor: Joachim Ringelnatz
11 
12 </p>
13 <table cellpadding="5" cellspacing="10">
14 <thead style="font-family: Arial, Trebuchet, sans-serif; font-size: 24px; color: blue;">
15 <tr>
16 <td>Bumerang</td>
17 <td>Die Ameisen</td>
18 </tr>
19 </thead>
20 <tbody>
21 <tr>
22 <td margin="0.5">
23 War einmal ein Bumerang;<br>
24 War ein Weniges zu lang.<br>
25 Bumerang flog ein Stück,<br>
26 Aber kam nicht mehr zurück.<br>
27 Publikum - noch stundenlang -<br>
28 Wartete auf Bumerang.
29 </td>
30 <td>
31 In Hamburg lebten zwei Ameisen,<br>
32 Die wollten nach Australien reisen.<br>
33 Bei Altona auf der Chaussee,<br>
34 Da taten ihnen die Beine weh,<br>
35 Und da verzichteten sie weise<br>
36 Dann auf den letzten Teil der Reise.
37 </td>
38 </tr>
39 </tbody>
40 </table>
41 </div>
42 </body>
43 </html>
44
```

HTML-Tags

- ... kennzeichnen bestimmte Textbestandteile.
- ... beschreiben Elemente eines Textes mit Hilfe von Attributen.
- Ein Starttag beginnt mit `<` und endet mit `>`.
- Eine Endetag beginnt mit `</` und endet mit `>`.
- Start- und Endetag bilden ein Paar.

Dokumentstruktur



Block-Elemente

- ... erzwingen eine neue Zeile.
- ... ordnen sich untereinander auf eine Seite an.
- ... haben einen gewissen Abstand zu den vorherigen und nachfolgenden Block-Elementen.
- ... können Text, Block-Elemente oder inline-Elemente enthalten. Ausnahme: Der Tag `<p>` enthält keine Block-Elemente.
- ... sind zum Beispiel: `<div>`, `<h1>`-`<h6>`, `<p>`, `<table>`, ``, ``.

inline-Elemente

- ... sind nebeneinander über die Seite verteilt.
- ... besitzen zu den vorherigen und nachfolgenden Elemente keinen vordefinierten Abstand.
- ... werden laut Norm von Block-Elementen eingeschlossen.
- ... dürfen Text und andere Inline-Elemente einschließen.
- ... sind zum Beispiel ``, `<a>`, `
` oder ``.
- ... können nicht in der Höhe und Breite vom Entwickler angepasst werden. Ausnahme: ``.

Meta-Tag

```
<meta name="author" http-equiv="content-type"  
      content="text/html; charset=iso-8859-1">
```

- ... bietet versteckte Informationen zu der Datei für Browser, Web-Server und Suchmaschinen.
- Die Angaben `http-equiv` geben Auskunft darüber,
 - ... welche Art von Information `content` enthält.
 - ... wie die Seite vom Browser oder Web-Server verarbeitet wird.
- Mit Hilfe des Attributs `content` wird die Struktur des Inhalts beschrieben. *text/html* strukturiert den Inhalt mit Hilfe von HTML-Elementen. *application/xhtml+xml* wird für XHTML genutzt werden.
- *charset* gibt Auskunft über den genutzten Zeichensatz. Sonderzeichen etc. können korrekt dargestellt werden. *iso-8859-1* wird als Zeichensatz für westeuropäische Sprachen genutzt.

Dokumenttyp

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"  
    "http://www.w3.org/TR/html4/strict.dtd">
```

- ... legt die Syntax der Datei fest.
- ... wird für die Überprüfung der Gültigkeit eines Dokuments genutzt.
- ... sollte in der ersten Zeile der Datei stehen.
- Die Deklaration beginnt mit *<!DOCTYPE*.
- Das Wurzelelement (html) der Datei wird genannt.
- In Anführungszeichen wird der Dokumenttyp definiert. Dem Dokumenttyp wird ein PUBLIC voran gestellt. Der Dokumenttyp legt die verwendete HTML-Version fest.
- Als letzter Parameter wird der Speicherort der Dokumenttyp-Definition in Anführungsstrichen angegeben.

... legt folgende Regeln fest

- Wie lautet das Wurzelement der Datei?
- Welche Elemente darf das Dokument enthalten?
- Welche Attribute sind für welche Elemente erlaubt?
- Welche Werte dürfen für die Attribute gesetzt werden?

Dokumenttypen für HTML-Dateien

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"  
    "http://www.w3.org/TR/html4/strict.dtd">
```

- CSS wird zur Formatierung genutzt.
- Nutzung bei Erstellung einer neuen Webseite.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
    "http://www.w3.org/TR/html4/loose.dtd">
```

- Nutzung von älteren Browsern wie zum Beispiel Netscape 3.x etc.
- Attribute der HTML-Elemente werden für die Formatierung genutzt.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"  
    "http://www.w3.org/TR/html4/frameset.dtd">
```

- Frameset erlauben.
- Diese Deklaration ist heute nicht mehr erwünscht.

Dokumenttypen für XHTML-Dateien

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

- Alle Elemente sind XHTML 1.0 - konform.
- CSS wird für die Formatierung genutzt.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

- Es können Abweichungen zum Standard vorkommen. Beispiel:
Innerhalb des Bodys kann Text ohne Tags wie `<p>` gesetzt werden.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml11.dtd">
```

- Die Datei wird in XHTML 1.1 geschrieben.

Namensraum in XHTML

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="de" xml:lang="de">
```

- Elemente und Attribute werden in einer Schublade zusammengefasst.
- ... legt fest, wie sich Elemente in dem Kontext XHTML verhalten.
- ... vermeiden Konflikte zwischen gleichen Bezeichnungen.
- `xmlns` ist die Abkürzung für XML Namespace. Als Wert wird ein Link zur Definition des Namensraums übergeben.
- `lang` und `xml:lang` definieren den Sprachraum für die Datei.

Formatierung mit Hilfe von CSS

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
2 <html lang="de">
3 <head>
4 <title></title>
5 <meta name="author" http-equiv="content-type" content="text/html; charset=iso-8859-1">
6 <link rel="stylesheet" type="text/css" href="format.css">
7 <style type="text/css">
8     p { font-family: Arial, Trebuchet, sans-serif;
9         font-size: 16px;
10        color: #000000;
11        text-indent: 1em
12    }
13 </style>
14 </head>
15 <body>
16 <div id="head">
17 <p>
18     Autor: Joachim Ringelnatz
19     
20 </p>
21 </div>
22 <div id="main">
23 <table cellspacing="5" cellpadding="10">
24 <thead style="font-family: Arial, Trebuchet, sans-serif; font-size: 24px; color: blue;">
25 <tr>
26 <td>Bumerang</td>
27 <td>Die Ameisen</td>
28 </tr>
29 </thead>
```

Für die Formatierung wird eine externe Datei genutzt.

Hier wird CSS in den HTML-Code eingebettet. Der Absatz wird formatiert.

Hier wird das Element direkt formatiert.

Stylesheet mit dem Element verbinden

```
<h1 style="font-family: Arial, Trebuchet, sans-serif;  
font-size: 24px;  
color: blue;">
```

Die Ameisen

```
</h1>
```

- Innerhalb des Tags wird das Element formatiert.
- Harte Formatierung des Elements mit Hilfe des Attributs *style*.
 - Mit Hilfe des Gleichheitszeichen wird dem Attribut ein Wert zugewiesen.
 - Der Wert wird durch Anführungszeichen begrenzt.
- Die Formatierung ist nicht von der Struktur des Dokuments getrennt.
- Die Formatierung sollte nur in der Testphase genutzt werden.

Stylesheet in HTML einbetten

```
<head>  
<style type="text/css">  
  
    h1 { font-family: Arial, Trebuchet, sans-serif;  
        font-size: 24px;  
        color: blue;  
    }  
    p  { font-family: Arial, Trebuchet, sans-serif;  
        font-size: 16px;  
        color: #000000;  
        text-indent: 1em  
    }  
  
</style>  
</head>
```

<style>-Tag

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

- Die Formatierungen beginnen mit `<style>` und endet mit `</style>`.
 - Das Attribut `type` beschreibt die Verarbeitung der angegebenen Regeln. In diesem Kurs wird CSS für die Definition der Formate genutzt.
 - Der Attributwert wird durch die Anführungszeichen begrenzt.
 - Mit Hilfe des Gleichheitszeichen wird dem Attribut ein Wert zugewiesen.
- Die Formatierung
 - ... haben immer als Eltern-Element `<head>`.
 - ... werden im Kopf einer HTML-Datei gesetzt.
- Die Formatierung haben nur innerhalb der Datei, in der sie definiert sind, Gültigkeit.
- Die Struktur und die Formatierungen befinden sich in einem Dokument.

CSS-Regeln

```
h1 { font-family: Arial, Trebuchet, sans-serif; color: blue; }
```

- ... beginnen immer mit der Bezeichnung des Elements. In diesem Beispiel wird die Überschrift der ersten Ebene formatiert.
- Die Format-Anweisungen
 - ... legen den Wert für ein Attribut fest. In diesem Beispiel die Schriftfarbe und die Schriftart. Eine Aufzählung von Attributwerten wird durch Kommata getrennt.
 - ... werden mit Hilfe des Semikolons abgeschlossen.
 - ... werden mit Hilfe der geschweiften Klammern zusammengefasst

Wohlgeformte Dokumente in Bezug auf Tags

- ... halten sich an die Syntaxregeln.
- Die Struktur des Dokuments entspricht dem angegebenen Dokumenttyp.
- `<html>` wird als Wurzel-Element genutzt.
- Jedes Element besitzt ein Start- und Ende-Tag in einer XHTML-Datei.
 - `<p>Text</p>`
 - Leeres Element-Tag für XHTML: ``
- Die Elemente sind korrekt ineinander verschachtelt. Untergeordnete Elemente müssen vor den übergeordneten Elemente geschlossen werden.
 - `` wird nur innerhalb von Listen verwendet.
 - Zitate, die mit `<blockquote>` gekennzeichnet sind, müssen in ein `<p>`-Element verschachtelt werden.

Wohlgeformte Dokumente in Bezug auf Attribute

- Elementnamen und Attributnamen sollten klein geschrieben werden.
- Die Attributwerte werden durch Anführungszeichen begrenzt.
- Das img-Tag besitzt einen Alternativtext.
- Attribute werden nicht minimiert.
 - Falsch: `<input typed="checkbox" checked>`
 - Richtig: `<input typed="checkbox" checked="checked">`

Gültige Dokumente

- In dem Dokument werden nur Elemente genutzt, die der Dokumenttypdefinition (DTD) entsprechen.
- Entspricht das Dokument dem, in der DTD, angegebenen Standard?
- ... sind wohlgeformt.
- ... besitzen eine Baumstruktur.
- Die Syntax des Dokuments ist korrekt.
- Überprüfungstools:
 - Validation von HTML: <http://www.validator.w3.org>.
 - Validation von CSS: <http://jigsaw.w3.org/css-validator/>
 - Add-On für Firefox: <http://chrispederick.com/work/web-developer/>

Externe Datei für die Formatierung nutzen

- Eine Datei enthält alle Formatierungen.
- Die Formatierung wird von der Struktur des Dokuments getrennt.
- Die Formatierung
 - ... sind ähnlich wie Formatvorlagen in einer Textverarbeitung wieder verwendbar.
 - ... können sehr schnell verändert werden.

Verweis auf die externe Datei

```
<head>
```

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

```
</head>
```

- Mit Hilfe von `<link>` im Kopf einer HTML-Datei wird auf eine externe Datei verwiesen.
 - `type` legt die Struktur des Inhalts der Datei fest.
 - `rel` gibt den Hinweis, dass auf eine Datei mit Formatierungen verwiesen werden soll.
 - `href` (hyper reference) gibt den Pfad zur Datei an.
- Der Tag kann beliebig oft vorkommen.
- Standardformatierung.

Ausgabemedium angeben

```
<head>  
<link type="text/css" rel="stylesheet" href="styleScreen.css" media="screen">  
<link type="text/css" rel="stylesheet" href="styleMobil.css" media="handheld">  
</head>
```

- Mit Hilfe des Attributs `media` wird das Ausgabemedium angegeben.
- Einige Beispiele für Attributwerte:

<i>screen</i>	Bildschirm
<i>projection</i>	Beamer
<i>handheld</i>	Mobile Geräte
<i>print</i>	Druckausgabe
<i>all</i>	Alle Geräte

Bevorzugte Format-Stile

```
<head>  
<link rel="stylesheet" type="text/css" href="format.css">  
<link title="blue" rel="stylesheet" type="text/css" href="format01.css">  
</head>
```

- Dem Attribut title wird eine Bezeichnung für die Formatierung übergeben.
- ... ist nur einmal in einem Web-Dokument vorhanden.
- ... wird genutzt, wenn kein anderes Format vorhanden ist.
- Der Nutzer kann mit Hilfe des Menüs
 - Ansicht – Format im Internet Explorer 8 eine Formatierung auswählen.
 - Ansicht – Webseiten-Stil im Firefox 3.5.2 eine Vorlage auswählen.

Alternative Formatierungen

```
<head>  
<link rel="stylesheet" type="text/css" href="format.css">  
<link title="blue" rel="stylesheet" type="text/css" href="format01.css">  
<link title="red" rel="alternate stylesheet" type="text/css" href="format02.css">  
<link title="blGr" rel="alternate stylesheet" type="text/css" href="format03.css">  
</head>
```

- ... werden mit dem Typ *alternate stylesheet* gekennzeichnet.
- ... können in beliebiger Anzahl vorhanden sein.
- ... müssen vom Benutzer ausgewählt werden.
- Die Standardformatierung wird trotzdem ausgewertet.
- Der Nutzer kann mit Hilfe des Menüs
 - Ansicht – Format im Internet Explorer 8 eine Formatierung auswählen.
 - Ansicht – Webseiten-Stil im Firefox 3.5.2 eine Vorlage auswählen.

CSS-Datei

```
h1 { font-family: Arial, Trebuchet, sans-serif;
      font-size: 24px;
      color: blue;
    }
p   { font-family: Arial, Trebuchet, sans-serif;
      font-size: 16px;
      color: #000000;
      text-indent: 1em
    }
```

- ... werden in Textdateien mit der Dateiendung ".css" gespeichert.
- ... enthalten die Formatdefinitionen für die verschiedenen Elemente.
- ... enthalten keine HTML-Tags.

Kommentare

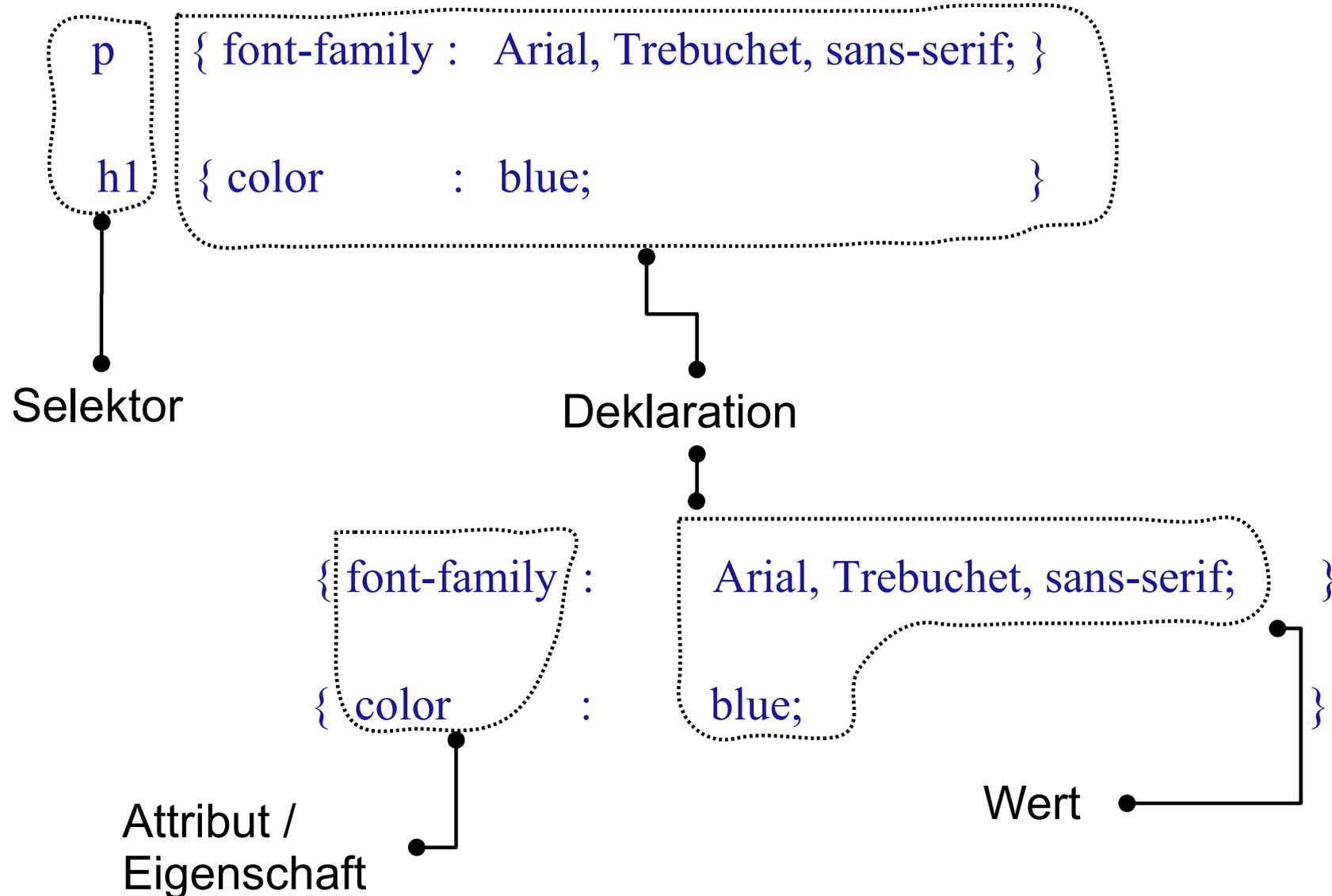
<!-- Kommentar -->

- In einer HTML-Datei wird Text innerhalb der Zeichen <!-- und --> ignoriert.
- Der Kommentar kann mehrzeilig sein.
- Informationen wie zum Beispiel Name des Entwicklers, Herstellungsdatum etc. werden an den Anfang einer HTML-Datei gesetzt.

/* Kommentar */

- In CSS-Dateien wird Text innerhalb der Zeichen /* und */ ignoriert.
- Der Kommentar kann mehrzeilig sein.
- Kommentare beschreiben die Formatierung.

CSS-Regeln bestehen aus



Selektor

- Auf wen wird die Regel angewendet?
- Welches Element wird mit Hilfe der Regel formatiert?
- Selektoren können
 - ... Elementnamen wie `<p>` etc. sein.
 - ... mit Hilfe von Klassen vielen Elementen eine Formatierung zuweisen.
 - ... mit Hilfe einer ID einem bestimmten Element eine Vorlage zuweisen.
- Selektoren können mit Hilfe von Nachkommen und Kindern zu formatierende Elemente auswählen.

Selektoren

```
h3    { font-family: Arial, Trebuchet, sans-serif;
        font-size: 20px;
        color: black;
      }
```

- Als Selektor wird ein Tag-Name aus HTML genutzt.
- Der Tag wird ohne die eckigen Klammern als Selektor genutzt.
- Wähle Element Überschrift3 aus und formatiere es mit Hilfe der angegebenen Regel.

Selektoren gruppieren

```
h1, h2 { font-family: Arial, Trebuchet, sans-serif;
         font-size: 24px;
         color: blue;
       }
h3     { font-family: Arial, Trebuchet, sans-serif;
         font-size: 20px;
         color: black;
       }
```

- Verschiedene Selektoren nutzen eine Formatvorlage.
- Verschiedene Elemente werden gleich formatiert.
- Die Selektoren in der Gruppe werden durch ein Komma getrennt.

Beispiel: HTML- Text

```
<body>
```

```
<h1>Arbeitsweise von <em>Kontextselektoren</em></h1>
```

```
<p>Dieses <em>Beispiel</em> zeigt wie man Text mit Hilfe  
der Hierarchie der Elemente formatieren kann.</p>
```

```
<p>Tags müssen nur <span>in Abhängigkeit ihrer <em>  
Ordnung</em> formatiert </span> werden.</p>
```

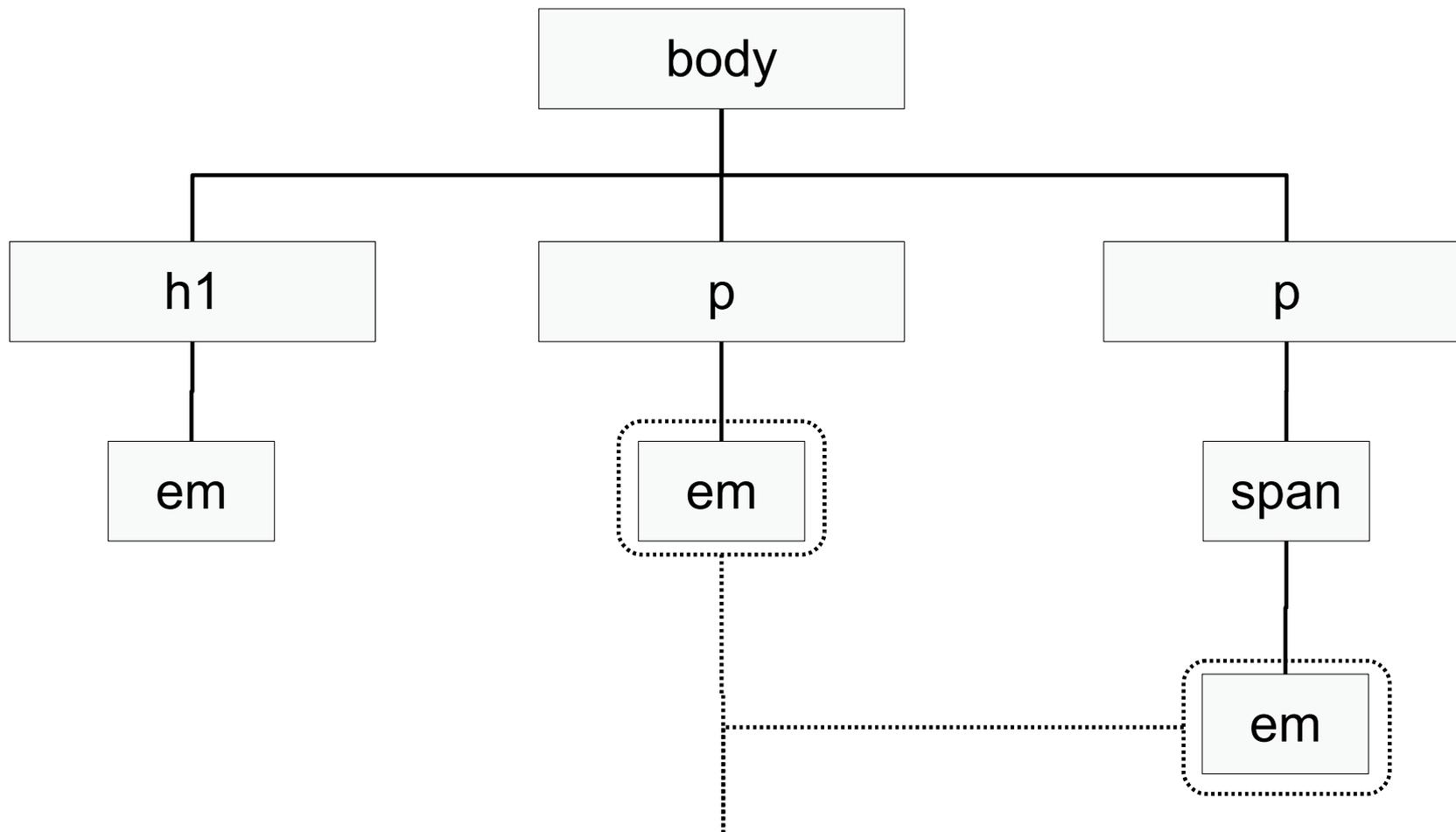
```
</body>
```

Kontextselektoren

```
p      { font-family: Arial, Trebuchet, sans-serif;
        font-size: 16px;
        color: #000000;
        }
p em   {
        color : #00FF00;
        }
```

- ... beschreiben die Dokumentstruktur.
- ... bilden die Hierarchie der Elemente untereinander ab.
- <p> ist ein Vorfahren-Element von .
- Vorfahren und Nachkommen werden durch ein Leerzeichen getrennt.
- Die Formatierung wird von den Nachkommen genutzt.
- Zwischen Vorfahren und Nachkommen können beliebig viele andere Elemente liegen.

Beispiel: Dokumentstruktur



... werden mit dem
Kontextselektor formatiert.

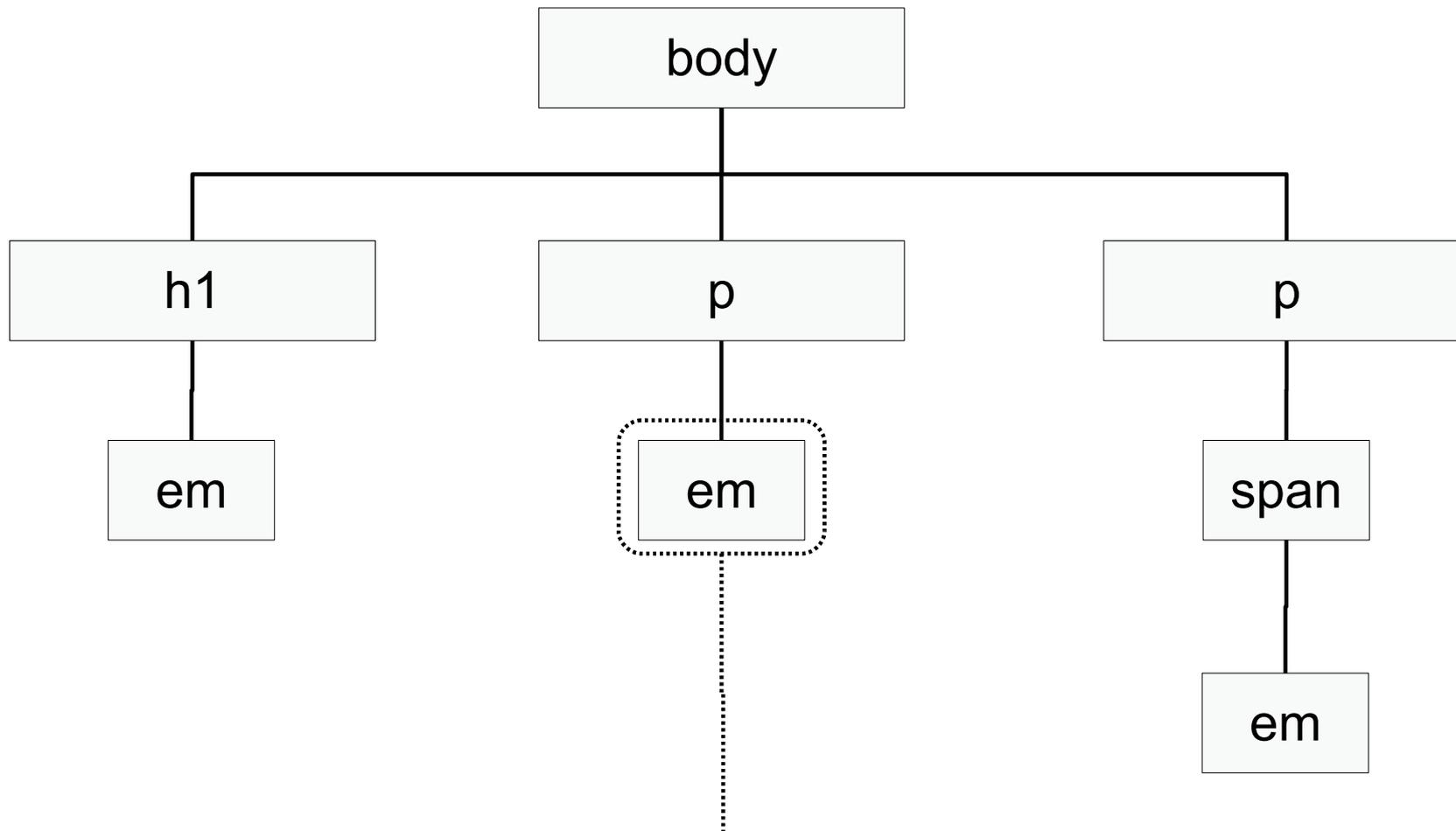
Kindselektoren

```
p      { font-family: Arial, Trebuchet, sans-serif;
        font-size: 16px;
        color: #000000;
        }

p > em { color : #00FFFF;
        }
```

- `` ist ein Kind-Element von `<p>`.
- Die Elemente werden mit Hilfe des Größer-Zeichens verbunden.
- Das Kind-Element wird formatiert.
- Internet Explorer 6 unterstützt keine Kind-Selektoren!

Beispiel: Dokumentstruktur



... wird mit dem Kindselektor formatiert.

Klassen von Elementen

```
<div class="absatz">
```

```
  Fläche: <div class="absatz formel">A = a * b</div></div>
```

```
<div class="absatz">
```

```
  Umfang: <div class="absatz formel">u = 2 * (a + b)</div></div>
```

- Dem Attribut `class` wird der Name einer Formatvorlage übergeben.
- Unterschiedlichen Elementen wird die gleiche Vorlage zugewiesen.
- Verschiedenen HTML-Elemente besitzen das gleiche Layout.
- Einem Element kann mehr als eine Klasse zugewiesen werden. Die Klassennamen werden durch ein Leerzeichen getrennt.

Klassen definieren

```
<style type="text/css">
  .absatz {
    font-family: Arial, Trebuchet, sans-serif;
    font-size: 18px;
    color: #000000;
  }
  .formel {
    font-style:italic;
    font-size: 90%;
    text-indent: 2em
  }
</style>
```

Klassen in CSS

- ... beginnen immer mit einem Punkt.
- ... haben immer einen benutzerdefinierten Namen, der dem Punkt ohne Leerzeichen folgt.
- ... können gruppiert werden. Die Klassen werden durch Kommata getrennt.

Klassennamen

- ... können aus den Zeichen a..z, A..Z, 0..9, dem Unterstrich und dem Trennstrich gebildet.
- ... können Zeichen ab dem ASCII-Wert 161 wie zum Beispiel ä enthalten. Diese Zeichen sollten aber nicht verwendet werden.
- ... dürfen nicht mit dem Trennstrich oder einer Zahl beginnen.
- ... können Leerzeichen enthalten.
- ... beachten nicht die Groß- und Kleinschreibung.

Klassen in Abhängigkeit eines Elements

```
*.absatz {  
    font-family: Arial, Trebuchet, sans-serif;  
    color: #000000;  
}  
div.absatz {  
    font-family: Arial, Trebuchet, sans-serif;  
    font-size: 18px;  
    color: #000000;  
}
```

- Element.Klasse.
- Die Klasse bezieht sich immer auf ein bestimmtes Element.
- Es werden Formatvorlagen für bestimmte Elemente erstellt.
- Ein Sternchen steht für alle Elemente, die diese Klasse nutzen. Das Sternchen muss nicht gesetzt werden.

Klassen und Nachfahren

```
div.absatz {  
    font-family: Arial, Trebuchet, sans-serif;  
    font-size: 18px;  
    color: #000000;  
}  
  
div.absatz p {  
    font-style:italic;  
    font-size: 90%;  
    text-indent: 2em  
}
```

Erläuterung

- `element.klasse nachkomme`
 - Die Klasse bezieht sich auf ein bestimmtes Element. Dieses Element hat einen Nachkommen, der mit Hilfe der Vorlage formatiert wird.
 - Es wird der Nachkomme eines Elements mit Hilfe der angegebenen CSS-Regel formatiert.
 - Dem Klassennamen folgt der Name des Nachkommen. Klassename und Nachkomme werden durch Leerzeichen getrennt.
- `.klasse nachkomme`
 - Die Formatierungen eines Kindes von Elementen einer bestimmten Klasse wird festgelegt.
 - Das Layout von Nachkommen-Elementen wird in Abhängigkeit einer Klasse festgelegt.

Ids für Elemente vergeben

```
<div id="head">
  <p>
    Autor: Joachim Ringelnatz
    
  </p>
</div>
<div id="main">
</div>
```

- Dem Attribut id wird der Name einer Formatvorlage übergeben.
- Mit Hilfe der ID kann ein Element eindeutig identifiziert werden.
- Die ID ist ein Platzhalter für das Element.
- Jede ID kommt exakt einmal in einem HTML-Dokument vor.

ID definieren

- Jede ID in CSS beginnt mit einem Hash-Zeichen (#).
- Im Anschluss an das Hash-Zeichen folgt ein benutzerdefinierter Name.
- Eine ID kann in einem Kontext definiert werden (element#ID).

```
#head{  
    border: 2px solid #0FF;  
    padding: 10px;  
    margin: 30px;  
}  
#main {  
    border: 2px solid #000;  
    padding: 10px;  
    margin: 30px;  
}
```

Deklarationen

```
border: 2px solid #0FF;  
padding: 10px;  
margin: 30px;
```

- legen das Layout eines Elements fest.
- ... sind Formatanweisungen.
- ... beschreiben Regeln für das Layout eines bestimmten Elements.
- ... werden durch ein Semikolon beendet.
- ... bestehen aus
 - ... Attributen, die das Element beschreiben und
 - ... einem oder mehreren Attributwerten.
- Attribute und deren Attributwerte werden durch ein Doppelpunkt getrennt.

Deklarationsblöcke

- ... fassen Regeln für die Formatierung von bestimmten Elementen oder Kategorien von Elementen zusammen.
- ... beginnen und enden mit den geschweiften Klammern.
- ... bestehen aus einer beliebigen Anzahl von Deklarationen.
- Pro Zeile eines Blocks steht eine Deklaration.
- Die Reihenfolge der Regeln ist innerhalb eines Blockes egal.

```
#head {  
    border: 2px solid #0FF;  
    padding: 10px;  
    margin: 30px;  
}  
#main {  
    border: 2px solid #000;  
    padding: 10px;  
    margin: 30px;  
}
```

Attributwerte

- ... können aus Text bestehen. Zum Beispiel *solid* für einen Rahmen, *red* für eine Farbangabe.
- ... können numerische Werte sein.
 - Dezimalzahlen werden häufig mit einer Einheit versehen und beschreiben eine Größe. Zum Beispiel *12px* für die Schriftgröße.
 - Hexadezimalzahlen werden für Farbangaben genutzt.

Hexadezimalzahlen

- ... beginnen immer mit dem Hash-Zeichen (#).
- ... nutzen die Ziffern 0 bis 9 für die Dezimalwerte 0 bis 9 und die Buchstaben A bis F für die Dezimalwerte 10 bis 15.
- Zum Beispiel wird die Hexadezimalzahl 9d in eine Dezimalzahl umgerechnet:

$$d * 16^0 = 13 * 1 = 13$$

$$9 * 16^1 = 9 * 16 = \underline{144}$$

$$157$$

- Zum Beispiel wird die Dezimalzahl 255 in eine Hexadezimal-Zahl umgerechnet:

$$255 / 16 = 15 \text{ Rest } 15 \text{ (F)}$$

$$15 / 16 = 0 \text{ Rest } 15 \text{ (F)}$$

Beispiele für Attribute

```
font-size: 12pt;  
font-size: 1em;  
font-size: 100%;
```

```
color: #000000;
```

```
border: none;
```

```
font-family: Arial, Trebuchet, sans-serif;
```

```
border: 2px solid #0FF;
```

Hintergrundfarbe und Textfarbe

Hintergrundfarbe für Block-Elemente etc.

```
background-color: #000000;
```

```
background-color: rgb(255,0,255);
```

```
background-color: white;
```

Textfarbe oder Vordergrundfarbe

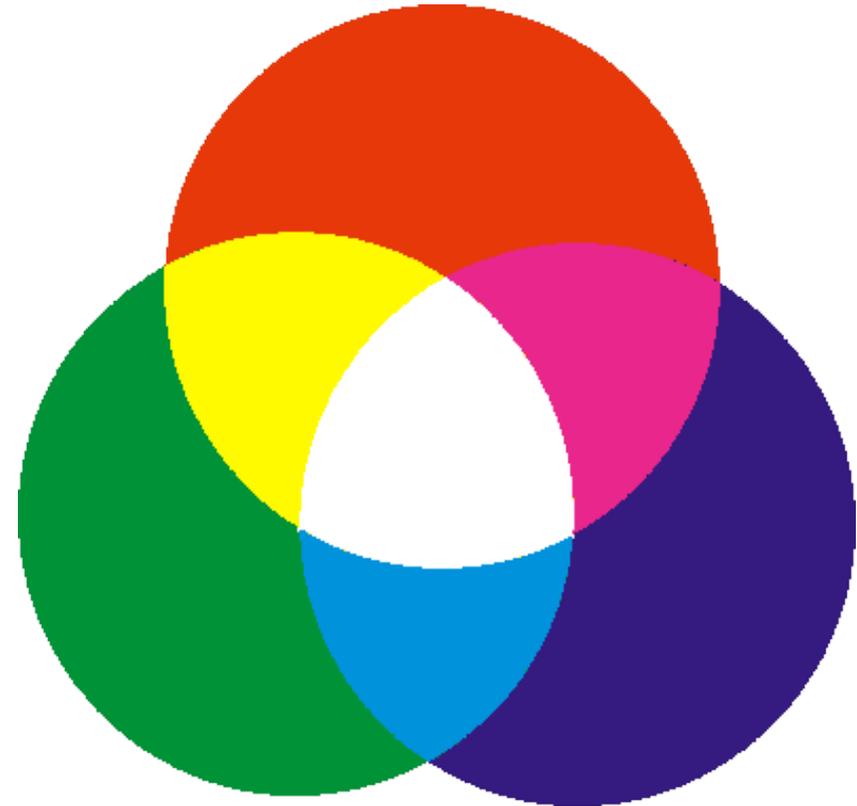
```
color: #000000;
```

```
color: #000;
```

```
color: black;
```

... nutzen RGB-Farben

- RGB-Farben werden für die Darstellung von Farben am Bildschirm genutzt.
- Das RGB-Farbsystem addiert (mischt) Licht in verschiedenen Farben.
- Jede Farbe (Rot, Grün, Blau) wird in 256 Helligkeitsstufen unterteilt. Zum Beispiel:
 - Weiß: 255, 255, 255
 - Schwarz: 0, 0, 0
 - Gelb: 255, 255, 0
- Um so mehr sich eine Farbe Weiß annähert, um so heller wird sie.



Informationen zur Farbwahl und -gestaltung

- <http://www.metacolor.de/>
- <http://daswebdesignblog.de/farbwirkungen-und-farbverwendung/393.html>
- <http://daswebdesignblog.de/farben-kombinieren-mit-dem-farbkreis/892.html>
- <http://www.newmediamen.de/download/lassfarbensprechen.pdf>
- <http://www.drweb.de/magazin/uber-die-wirkung-der-farben/> und <http://www.drweb.de/magazin/uber-die-wirkung-der-farben-teil-2/>
- <http://de.selfhtml.org/helferlein/farben.htm>

RGB-Farben

- ... können mit Hilfe eines Hexadezimalwertes gesetzt werden (http://www.w3schools.com/CSS/css_colors.asp).
- ... können mit Hilfe eines Namens angegeben werden (http://www.w3schools.com/css/css_colornames.asp).
- ... können mit Hilfe der Funktion *rgb(wert, wert, wert)* berechnet werden.
- Die Darstellung der Farben ist abhängig von der Interpretation des Browsers.

16 Grundfarben

black	#000000	0, 0, 0
white	#FFFFFF	255, 255, 255
red	#FF0000	255, 0, 0
maroon	#800000	128, 0, 0
green	#008000	0, 128, 0
lime	#00FF00	0, 255, 0
olive	#808000	128, 128, 0
blue	#0000FF	0, 0, 255
aqua	#00FFFF	0, 255, 255
navy	#000080	0, 0, 128
teal	#008080	0, 128, 128
purple	#800080	128, 0, 128
fuchsia	#FF00FF	255, 0, 255
yellow	#FFFF00	255, 255, 0
gray	#808080	128, 128, 128
silver	#c0c0c0	192, 192, 192

Nutzung eines Hexadezimalwertes

- Für die Farbanteile Rot, Grün, Blau wird jeweils ein zweistelliger Wert gesetzt.
- Syntax: #RRGGBB.
 - Jeder Farbanteil besteht aus zwei Zeichen für die Codierung des Wertes.
- Syntax: #RGB
 - Falls beide Zeichen für alle Farbanteile gleich sind, können diese zusammengefasst werden.
 - Zum Beispiel kann die Farbe #FF00FF durch die drei Zeichen #F0F dargestellt werden.

Beispiele

- #17519d beschreibt das LUH-Blau.
- #FFFFFF stellt die Farbe Weiß dar.
- #FF0000 steht für einen Rotton.
- #00FF00 stellt einen Grünton dar.
- #0000FF steht für die Farbe Blau.
- #000000 stellt die Farbe Schwarz dar.

Funktion rgb()

```
color: rgb(255,0,255);
```

```
color : rgb(50%, 0%; 50%);
```

- Die Funktion wird mit Hilfe der Bezeichnung rgb aufgerufen.
- In runden Klammern folgt eine Parameterliste. Jeder Parameter symbolisiert einen Farbanteil. Die Parameter werden durch Kommata getrennt.
- Die Funktion gibt eine RGB-Farbe zurück.
- Als Parameter
 - ... können Werte von 0 bis 255 genutzt werden.
 - ... können Prozentzahlen zwischen 0 und 100% übergeben werden. Es sind Dezimalwerte sowohl als auch Ganzzahlen erlaubt. Als Dezimaltrennzeichen wird der Punkt genutzt.

Textausrichtung

```
text-align: left;      /* linksbündig */  
text-align: right;    /* rechtsbündig */  
text-align: center;   /* zentriert */  
text-align: justify;  /* Blocksatz */
```

- ... in horizontaler Richtung.
- Standardmäßig wird der Text linksbündig ausgerichtet.
- ... bezieht sich auf Textabsätze oder inline-Elemente.
- Der Internet Explorer wendet die Einstellung auch auf Block-Elemente an.

Schriften

- Schriftfamilie: font-family
- Schriftgröße: font-size
- Schriftstil: font-style
- Schriftgewicht: font-weight

Schriftfamilien

font-family: arial, helvetica, "trebuchet ms", sans-serif;

- Die Auflistung der Schriftfamilien werden durch Kommata getrennt.
- ... sind vererbbar.
- Beachte:
 - Jedes Betriebssystem nutzt andere Schriftarten. Zum Beispiel nutzt das Betriebssystem Linux Helvetica, aber nicht Arial.
 - Die angegebenen Schriften müssen auf dem Computer installiert sein.
- Erläuterung des Beispiels:
 - Falls Arial nicht vorhanden ist, nutze Helvetica.
 - Falls Helvetica nicht vorhanden ist, nutze Trebuchet MS.
 - Falls Trebuchet MS nicht vorhanden ist, nutze irgendeine vorhandene serifenlose Schrift.

Bezeichnungen der Schriften

font-family: arial, helvetica, "trebuchet ms", sans-serif;

- Es werden nur die Familiennamen als Bezeichnung genutzt.
- Schriftfamilien mit Leerzeichen in der Bezeichnung
 - ... werden in einer CSS-Datei in Anführungszeichen geschrieben.
 - ... werden durch Hochkommata begrenzt, wenn der Stil in eine HTML-Datei eingebettet ist.

Schriftarten

- Serifenschriften
 - Am Ende des Buchstabenstrichs befinden sich kleine Häkchen.
 - ... wurden für die Druckindustrie entwickelt.
 - Times New Roman, Palatino.
- Serifenlose Schriften
 - Schrift ohne Schnörkel.
 - Arial, Verdana, Trebuchet MS.
- Nichtproportionale Schriften (Monospace)
 - Jeder Buchstabe benötigt gleich viel Platz.
 - Courier, Monotype.
- Kursivschriften
 - Handschriften in elektronischer Form.
- Informationen zu Schriften finden Sie unter <http://www.webmaster-crashkurs.de/schriften-mit-css-gestalten.htm>.

Beispiele

```
/* Beliebige Serifen-lose Schrift */  
font-family: sans-serif;
```

```
/* Arial, Helvetica, Beliebige Serifen-lose Schrift */  
font-family: arial, helvetica, sans-serif;
```

```
/* Beliebige Serifen-Schrift */  
font-family: serif;
```

```
/* Courier, Courier New, Beliebige nicht-proportionale Schrift */  
font-family: courier, 'courier-new', monospace;
```

Schriftgröße

```
font-size: 12pt;  
font-size: 1em;  
font-size: 100%;
```

- Standardmäßig werden Schriftgrößen in Pixel angegeben.
- Schriftgrößen können
 - ... absolut oder
 - ... relativ in Abhängigkeit ihrer Vorfahren angegeben werden.

Absolute Werte

- ... sind nicht veränderbar und damit nicht skalierbar.
- Absolute Schriftgrößen können in folgenden Einheiten gemessen werden:

Bezeichnung	Einheit	Umrechnung
Pixel	px	... relativ zur Bildschirmauflösung. Standardeinstellung.
Punkt	pt	Anglo-amerikanischer Raum: 0.351 mm oder 1/72 Zoll Didot-System: 0.376 mm
Pica	pc	12pt oder 1/6 Zoll

Relative Werte

- ... passen Werte proportional an.
- ... beziehen sich immer auf die Schriftgröße der Vorfahren.
- Verschachtelte Text können durch Vererbung eine sehr kleine Schrift bekommen.

Einheit	Beschreibung
em	Schriftgröße in Abhängigkeit des Elements oder der Eltern. Standardmäßig sind 1em = 16px.
ex	Schriftgröße in Abhängigkeit der Höhe von x des Elements oder deren Eltern.
%	Prozentualen Anteil. Eltern: Schriftgröße 20px Kind: Schriftgröße 50% = 10px

Schriftgröße relativ zu den Eltern

```
div.absatz {  
    font-family: Arial, Trebuchet, sans-serif;  
    font-size: 18px;  
    color: #000000;  
}  
  
div.absatz p {  
    font-size: 90%;           /* ca. 16 Pixel */  
    font-style:italic;  
    text-indent: 2em         /* Texteinrückung am Anfang des Absatzes*/  
}
```

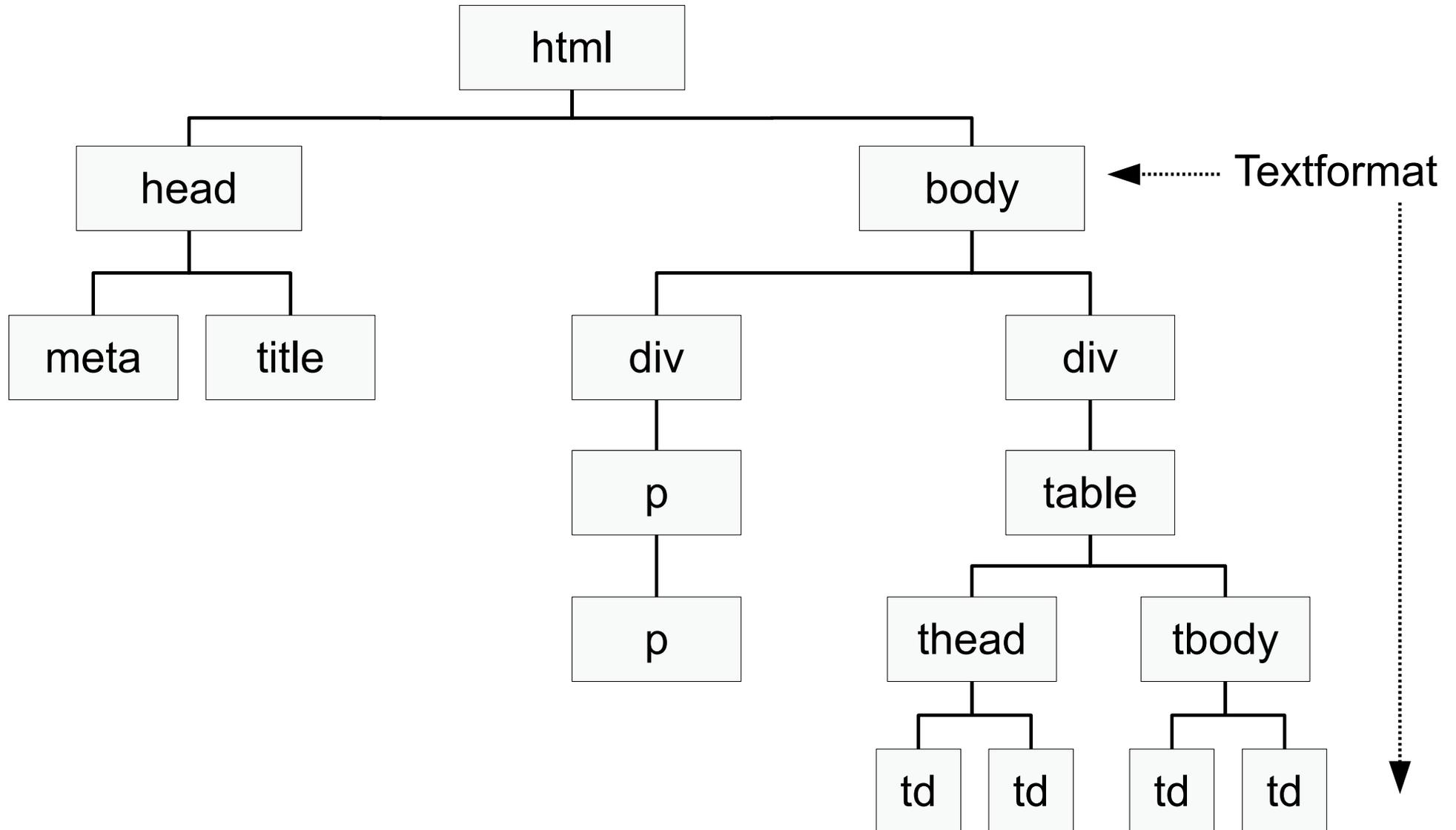
Schriftstil

- `font-style : italic` für kursive Schrift.
- `font-style : oblique` für schräggestellte Schrift.
- `font-style : normal`.

Schriftgewicht

- font-weight : bold oder font-weight : 700 für Fettschrift.
- font-weight : bolder für eine extra Fettschrift.
- font-weight : lighter für eine dünne Schrift.
- font-weight : 500 für eine mittlere Schrift.
- font-weight : normal.
- Es können die Werte 100, 200, 300, 400, 500, 600, 700, 800 und 900 genutzt werden.

Vererbung von Formatierungen



Hinweise

- Ein Nachkomme
 - ... erbt Textformatierungen von seinen Vorfahren automatisch.
 - ... erbt keine Layout-Vorlagen für Rahmen, Abstände, Ränder oder Hintergrundfarben von seinen Vorfahren.
 - ... kann mit Hilfe des Wertes `inherit` alle Eigenschaften eines Vorfahrens erben.
- `<body>`
 - ... besitzt allgemein gültige Formatierung.
 - ... hat eine Standardformatierung.
- Um so tiefer man in der Dokumentstruktur ist, um so spezieller werden die Formatierungen.

Inherit nutzen

```
div {  
    border: 2px solid #000;  
    padding: 10px;  
}  
  
#main {  
  
    font-family: Arial, Trebuchet, sans-serif;  
    margin: 30px;  
}  
  
.absatz {  
    font-size: 18px;  
    color: #000000;  
    margin: inherit;           /* Vererbung von den Eltern */  
}
```

Kaskade

- Stile werden ähnlich wie bei einem Wasserfall von oben nach unten weitergereicht.
- In Abhängigkeit der Kaskade werden die CSS-Regeln durchsucht und in einer bestimmten Reihenfolge abgearbeitet.
- Konflikte von verschiedenen Regeln, die auf ein Element angewendet werden können, werden vermieden.

Stufe 1: Regeln Elementen zuordnen

- Finde alle Regeln die auf ein Element zutreffen.
- Wenn nur eine Regel zutrifft, wird diese auf das Element angewandt.
- Wenn nicht, gehe zu Stufe 2.

Stufe 2: Woher kommen die Regeln?

- Sortiere die Regel nach ihren Ursprung

Quelle	Element	Eigenschaft	Wert
Browser-Stylesheet	<p>	font-family	Times
Benutzer-Stylesheet	<p>		
Verknüpftes Stylesheet	<p>	font-family	Verdana
Eingebettetes Stylesheet	<p>		
Inline-Stylesheet	<p>	font-family	Arial

- Wenn Regeln gleichen Ursprungs vorhanden sind, spielt die Spezifität der Regel eine Rolle.

Stylesheet-Arten

- Browser-Stylesheet
 - ... ist das Standard-Stylesheet des Web-Browsers.
 - `html.css` für Firefox.
- Benutzer-Stylesheet
 - Jeder Benutzer kann sein eigenes Stylesheet anlegen.
 - Standardgröße für Text etc.
 - `Firefox/Profiles/.../chrome/userChromeexamples.css`
- Autoren-Stylesheet
 - Stylesheets, die der Entwickler der Webseite erstellt hat.
 - Deklarationen, die die Webseite betreffen.
 - Verknüpfte, eingebettete oder inline-Stylesheets.

Stufe 3: Spezifität

- Um so spezifischer eine Regel ist, um so mehr Gewicht hat sie.
- Beispiel:

Regel	Style	ID	Klasse	Element
<code>p.classParagraph span { font-style: italic;}</code>	0	0	1	2
<code>p.classParagraph { color: #FF0000;}</code>	0	0	1	1
<code>p { font-family : Arial, sans-serif; }</code>	0	0	0	1
<code>p#idParagraph { color: #FF0000; }</code>	0	1	0	1

- Die Regel `p#idParagraph { color: #FF0000; }` setzt alle anderen Regeln außer Kraft. Die Regel hat die höchste Spezifität.
- Beachte: Definierte Regeln haben Vorrang vor ererbten Regeln.

Stufe 4: Vorrangstellung

- Voraussetzung:
 - Mehr als eine Regel kann auf ein Element angewandt werden.
 - Alle Regeln haben den gleichen Ursprung.
 - Alle Regeln haben die gleiche Spezifität.
- Die zuletzt notierte Regel hat Vorrang.

Box-Modell

- CSS legt um jedes Element in einer HTML-Datei ein Rechteck.
- Das Rechteck besteht aus dem Inhalt und einem Innenabstand zum Rahmen.
- Der Rahmen
 - ... ist standardmäßig ausgeblendet (`border-style:none`).
 - ... ist standardmäßig so breit wie das Browserfenster.
- Jedes Rechteck kann einen Außenabstand zu einem anderen Rechteck haben.
- Ein Rechteck kann eine Hintergrundfarbe haben.

file:///D:/Kursmat...beispiel/box.html

Goslar

Kaiserpfalz



Die Kaiserpfalz Goslar umfasst ein Areal von etwa 340 mal 180 Metern, gelegen am Fuße des Rammelsberg im Süden der Stadt Goslar, auf dem sich im wesentlichen das Kaiserhaus, das ehemalige Kollegiatstift "St. Simon und Judas", die Pfalzkapelle "St. Ulrich" und die Liebfrauenkirche befinden, bzw. befanden. Das Kaiserhaus ist der größte, älteste und zugleich besterhaltene Profanbau des 11. Jahrhunderts in Deutschland. Er diente insbesondere den Salierkaisern als bevorzugte Aufenthaltsstätte. Das Gebäudeensemble der Kaiserpfalz hat bereits im 11. Jahrhundert derart beeindruckt, dass der Chronist Lampert von Hersfeld vom "berühmtesten Wohnsitz des Reiches" spricht. Der Pfalzbezirk gehört seit 1992 gemeinsam mit der Goslarer Altstadt und dem Rammelsberg zum Weltkulturerbe der UNESCO.

Rahmen in CSS formatieren

```
<style type="text/css">  
  * {  
    border-width: 2px;  
    border-style: solid;  
    border-color: #000;  
  }  
</style>
```

```
<style type="text/css">  
  * {  
    border: 2px solid #0FF;  
  }  
</style>
```

Rahmenbreite und -farbe

`border-width: 1px;`

- Die Rahmenbreite wird in Pixel angegeben.
- Die Werte *thin*, *medium* und *thick* können von Browser zu Browser variieren.

`border-color: #000;`

- Farben für Rahmen werden genauso wie die Schriftfarbe definiert.

Rahmenstil

`border-style: solid`

`border-style: double`

`border-style: dotted`

`border-style: dashed`

`border-style: groove`

`border-style: ridge`

`border-style: inset`

`border-style: outset`

Kurzschreibweise

border: width style color;

- ... formatiert alle vier Ränder gleich.
- Die Reihenfolge der Werte ist egal.
- Die Anzahl der Werte ist egal.

```
<style type="text/css">  
* {  
    border: 2px solid #0FF;  
}  
</style>
```

Einzelne Rahmen formatieren

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

.rahmenLinks {
    border-left-color: #FF0000;
}

.rahmenRechts {
    border-right-color: #00FF00;
}

.rahmenOben {
    border-top-color: #0000FF;
}

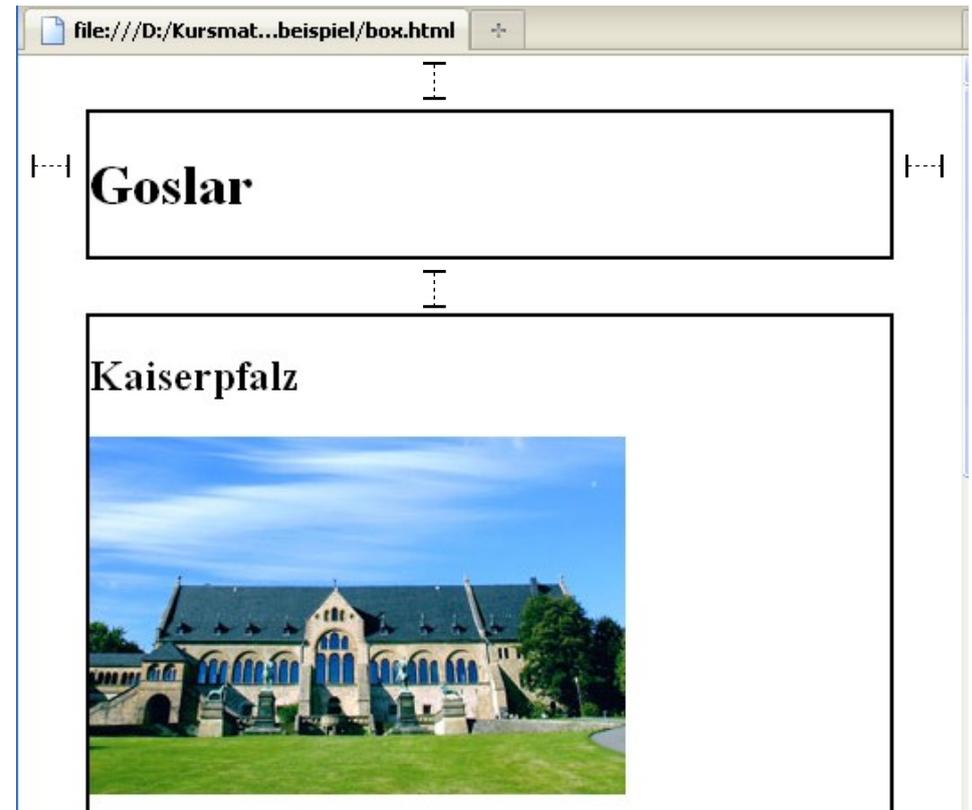
.rahmenUnten {
    border-bottom-color: #FF00FF;
}

</style>
```

Außenabstand

```
<style type="text/css">
  div {
    border-width: 2px;
    border-style: solid;
    Border-color: #000;

    margin: 30px;
  }
</style>
```

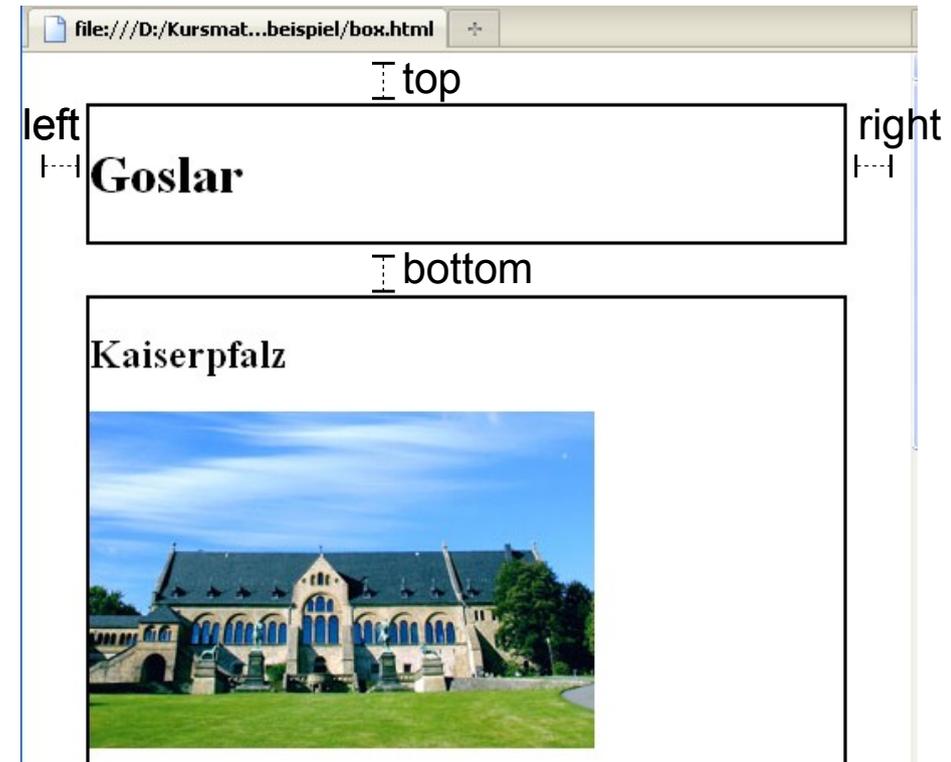


Außenabstände für jeden Rand

```
<style type="text/css">
  div {
    border-width: 2px;
    border-style: solid;
    border-color: #000;

    margin-left: 30px;
    margin-right: 0px;
    margin-top: 15px;
    margin-bottom: 15px

    /* margin: top right bottom left */
    margin: 15px 0 15px 30px
  }
</style>
```



Außenabstände relativ setzen

```
<style type="text/css">
  div {
    border-width: 2px;
    border-style: solid;
    Border-color: #000;

    margin: 0.75em 0 0.75em 30px;
  }
</style>
```

Hinweise

- `margin wert` betrifft alle Seiten.
- `margin wert01 wert02` nutzt den ersten Wert für den oberen und den unteren Abstand. Der zweite Wert wird für den Abstand links und rechts genutzt.
- `margin wert01 wert02 wert03` nutzt den ersten Wert für den oberen Abstand. Der zweite Wert wird für den Abstand links und rechts genutzt. Der dritte Wert wird für den unteren Abstand genutzt.

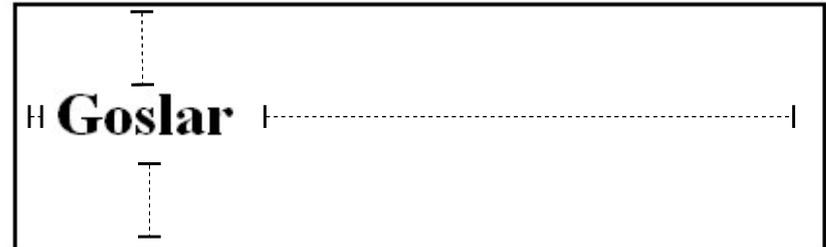
Vertikale Abstände

- ... kollabieren, wenn die Elemente in der Reihenfolge der Dokumentstruktur abgebildet werden.
- Der größere Wert der oberen oder unteren Außenabstände legt den Gesamtabstand zwischen zwei Block-Elementen fest.
- Ein negativer Außenabstand wird vom positiven Wert abgezogen.
- Zum Beispiel:
 - Die erste Box hat einen unteren Abstand von 20 Pixel.
 - Die nachfolgende Box hat einen oberen Abstand von fünf Pixel.
 - Die Außenabstände der Boxen werden nicht addiert. Der Außenabstand zwischen beiden Boxen beträgt 20 Pixel.

Innenrand

```
<style type="text/css">
  div {
    border-width: 2px;
    border-style: solid;
    border-color: #000;
    margin: 15px 0 15px 30px;

    padding: 15px 20px 30px 20px;
  }
</style>
```



Kaiserpfalz



Die Kaiserpfalz Goslar umfasst ein Areal von etwa 340 mal 180 Metern, gelegen am Fuße des Rammelsberg im Süden der Stadt Goslar, auf dem sich im wesentlichen das

Textbreite

```
<style type="text/css">
  div {
    border-width: 2px;
    border-style: solid;
    border-color: #000;
    margin: 15px 0 15px 30px;
    padding: 15px 20px 30px 20px;
  }
  p {
    border-width: 2px;
    border-style: solid;
    border-color: #000;
    padding: 15px 20px 30px 20px;
    width: 50%;
  }
</style>
```

Einheiten für die Werte

Bezeichnung	Einheit	Umrechnung
Zoll	in	1in = 24.4mm = 72.27pt
Zentimeter	cm	
Millimeter	mm	
Prozent	%	

Hinweise

- Die Höhe sollte nicht gesetzt werden.
- Inline-Elemente können nicht in der Größe verändert werden.
- Die Breite der Box wird folgendermaßen berechnet
Textbreite + Innenabstand + Rahmenbreite + Außenabstand

Elemente ausblenden

- Elemente können mit Hilfe des Attributs `display: none` ausgeblendet werden.
- Elemente werden mit Hilfe von `display: block` zu einem Block-Element konvertiert.

```
<style type="text/css">  
  img {  
      display: None;  
  }  
</style>
```

Elemente im normalen Elementfluss

- ... werden im Web in Abhängigkeit der Dokumenthierarchie angezeigt.
- ... werden nicht durch das Attribut `position` beeinflusst.
- ... werden als statische Elemente bezeichnet.
- Block-Elemente
 - ... werden vertikal von oben nach unten angeordnet.
 - ... sind eine Zeilenlänge breit.
 - ... werden automatisch mit einem Zeilenumbruch abgeschlossen.
- Inline-Elemente
 - ... fließen innerhalb ihrer Block-Elemente von links nach rechts.

Elemente absolut positionieren

- Mit Hilfe von top und bottom sowie left und right kann ein Element absolut auf der Seite positioniert werden
- Das Element wird aus dem Elementfluss herausgenommen. Alle anderen Elemente rutschen nach.
- Das Element wird in Abhängigkeit der linken, oberen Ecke des nicht-statischen Eltern-Elements verschoben. Häufig wird es in Abhängigkeit des Browserfensters verschoben.

```
.block {  
    position: absolute;  
    top:35px;  
    left:240px;  
    z-index: 1;  
  
    margin: 15px 0 15px 30px;  
    padding: 15px 20px 30px 20px;  
}
```

Elemente relativ positionieren

- Das Element wird immer zu seiner Ausgangsposition verschoben.
- Es kann zu Überlagerung von Elementen kommen. Das relativ verschobene Element wird wie ein Element im Elementfluss behandelt. Sein Platz im Elementfluss wird nicht von anderen Elementen genutzt.
- Mit Hilfe des `z-index` kann die Reihenfolge der gestapelten Elemente bestimmt werden. Um so höher der Attributwert, um so weiter vorn liegt das Element.

```
.toolTip {  
    border-width: 1px;  
    border-style: solid;  
    margin: 0;  
    padding: 5px 20px 5px 20px;  
    width: 100px;  
  
    position: relative;  
    top: -120px;  
    left: -20px;  
    z-index: 2;  
  
    background-color: yellow;  
}
```

Fließendes Element

- Ein Element kann ein anderes Element rechts oder links umfließen.
- Elemente werden rechts- oder linksbündig ausgerichtet.
- In vertikaler Ausrichtung wird das Element nicht neu positioniert.

```
<style type="text/css">  
  img{  
    border-style: solid;  
    padding: 15px 20px 30px 20px;  
    width: 50%;  
  
    float: right;  
  }  
</style>
```

Hinweise

- Ein fließendes Element wird immer als Block-Element dargestellt.
- Ein fließendes Element kann ein Block-Element im normalen Elementfluss überlagern.
- Es sollte eine Breite angegeben werden, damit andere fließende Elemente Platz haben.
- Fließende Elemente werden aus dem Elementfluss herausgenommen. Nachfolgende Elemente nehmen ihren Platz ein. Ausnahme: Text wird von fließenden Elementen verdrängt. Der Text umfließt das Element.

... löschen

- Mit Hilfe von `clear:left` darf kein vorhergehendes Element weiter links liegen.
- Mit Hilfe von `clear:right` darf kein vorhergehendes Element weiter rechts liegen.
- Mit Hilfe von `clear:both` darf kein vorhergehendes Element weiter links oder rechts liegen.
- Das Element wird unterhalb eines fließenden Elements positioniert.

```
.blockNoFloat {  
  border-width: 2px;  
  border-style: solid;  
  border-color: #000;  
  
  margin: 15px 0 0px 30px;  
  padding: 15px 20px 30px 20px;  
  
  width: 100%;  
  float:left;  
  clear: both;  
}
```

Pseudo-Klassen

- ... funktionieren nur mit einem HTML-Element zusammen.
- In diesem Beispiel wird das Element `<a>` für Links genutzt.
- Als Trennzeichen zwischen der Bezeichnung der Pseudoklasse und dem Elementnamen wird ein Doppelpunkt genutzt.

```
a {  
    color: black;  
    text-decoration: none;  
}
```

```
a:visited {  
    color: gray;  
    text-decoration: none;  
}
```

```
a:hover {  
    color: blue;  
    text-decoration: underline;  
}
```

Reihenfolge der Pseudoklassen

- `a:link` formatiert den noch unbesuchten Hyperlink.
- `a:visited` formatiert den besuchten Hyperlink.
- `a:hover`. Die Maus fährt über das Element.
- `a:active` formatiert den aktiven, gerade angeklickten Link.
- `a:focus`. Das Element wird mit Hilfe der `<Tab>`-Taste angesprochen. Das Element bekommt den Fokus.
- Um Fehler zu vermeiden sollte die Reihenfolge eingehalten werden.
- Es müssen nicht alle Klassen genutzt werden.