

XML light

# CSS – Cascading Style Sheets

Jörn Clausen

`joern@TechFak.Uni-Bielefeld.DE`

# Übersicht

- Wieso Cascading Style Sheets?
- HTML und CSS
- XML und CSS
- Ausblick

# Probleme, Probleme ...

- Probleme mit XML:

„Und wie kann ich mir das jetzt ansehen?“

- Probleme mit HTML:

- Ändere die Farbe aller h2-Überschriften in blau:

```
<h1><font color="red">HTML</font></h1>
```

```
<h2><font color="red">Einleitung</font></h2>
```

- Erstelle ein Inhaltsverzeichnis:

```
<font size="+3">HTML</font>
```

```
<font size="+2">Einleitung</font>
```

- trenne Inhalt/Struktur von Formatierung
- lege Formatierung ausserhalb des Dokuments fest

# Cascading Style Sheets

- Idee: Element ↔ Darstellung
- ursprünglich für HTML, später auch für XML
- mehrere Versionen: CSS1 (1996), CSS2 (1998), ...
- Unterstützung durch aktuelle Browser:
  - Gecko (Mozilla, Firebird, Netscape 6+)
  - khtml (Konqueror, Safari)
  - Opera
  - Internet Explorer 5.5+

# Wieso „Cascading“?

- Format-Definition an verschiedenen Stellen:
  - globale CSS-Datei, zusätzliche Definitionen in HTML-Datei
  - geschachtelte Elemente
- speziellere Definition gewinnt
- Vorgaben individuell anpassen

# Aufgaben

- Verwende im weiteren Verlauf *entweder* Mozilla oder Opera als Web-Browser.

- Lade die Datei `content.html` in den Web-Browser.

- Füge im header von `content.html` folgende Zeile ein und verwende die Reload-Funktion des Browsers:

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

- Öffne die Datei `format.css` mit einem Editor und sieh sie Dir an. Sorge dafür, daß `h2`-Überschriften blau eingefärbt werden.

# Rules

- aus format .css:

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

- Zeilenumbrüche egal:

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

- Aufbau:

```
selector { property: value; }
```

- mehrere *declarations* gleichzeitig:

```
h1 {  
  color: red;  
  background: green;  
}
```

# Farben

- zwei properties bereits gesehen: `color`, `background`
- vordefinierte Farben: `aqua`, `black`, `blue`, `fuchsia`, `gray`, `green`, `lime`, `maroon`, `navy`, `olive`, `purple`, `red`, `silver`, `teal`, `white`, `yellow`
- RGB-Farben
  - Prozent-Angaben:  

```
h1 { color: rgb(20%, 60%, 0%); }
```
  - Hexadezimal-Darstellung:  

```
h1 { color: #FF001B; }
```



# Schriften

- Font-Eigenschaften

- font-family: serif, sans-serif, ..., Times, Helvetica, ...
- font-style: italic, oblique
- font-variant: small-caps
- font-weight: bold, bolder, lighter, 100, ..., 900
- font-size: small, large, x-large, ..., 12pt, 75%

- Kurzschreibweise

```
h1 { font: bold italic 24pt serif; }
```

- alternative Schriften anbieten:

```
h1 { font-family: Times, serif; }
```

# Aufgaben

- Färbe einige der Elemente in der HTML-Datei ein. Verwende sowohl vor- als auch selbstdefinierte Farben.

- In welcher Farbe werden die unordered lists dargestellt:

```
ul { color: blue; }  
li { color: green; }
```

- Sorge dafür, daß Überschriften serifenlos gesetzt werden, für den restlichen Text aber eine Serifen-Schrift verwendet wird.

# Rules revisited

- mehrere Elemente gleichzeitig definieren

```
ul, ol, dl { color: green; }
```

- kontextabhängige Regeln

```
h1 em { color: red; }
```

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

- Element-spezifische Klassen

```
p.extra { font-size: 80%; }
```

```
<p>...</p>
```

```
<p class="extra">...</p>
```

- Element-unabhängige Klassen

```
.hilite { color: white; background: blue; }
```

# Aufgaben

- Sorge dafür, daß Absätze in Listen in blau gesetzt werden, sonstige Absätze aber in schwarz.
- Einige Hervorhebungen sind wichtiger als andere. Sorge dafür, daß das zweite `em`-Element mit einer Farbe hinterlegt wird.
- Füge diese Regeln ein:

```
p kbd { background: blue; }  
p > kbd { background: green; }
```

Was passiert, wenn Du die Reihenfolge dieser beiden Regeln vertauscht?

# XML mit CSS formatieren

- gleiche Idee: Element  $\leftrightarrow$  Darstellung
- aber: keine vordefinierten/impliziten Regeln
- keine Semantik (Hypertext, Bilder)

# Aufgaben

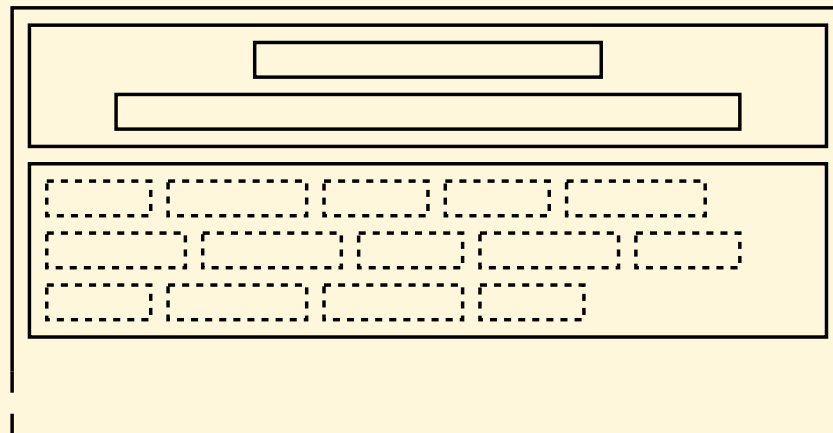
- Die Dateien `panther.xml` und `erlkoenig.xml` enthalten Gedichte. Such Dir eine der beiden Dateien aus und sieh sie Dir mit Deinem Web-Browser. Wie wird das Gedicht dargestellt?
- Füge unmittelbar nach der XML-Deklaration folgende Zeile ein:  

```
<?xml-stylesheet type="text/css" href="gedichte.css"?>
```

Wie wird das Gedicht nun im Browser dargestellt? Sieh Dir die Datei `gedichte.css` genauer an.
- Sorge dafür, daß der Titel in einer größeren Schrift und der Autor kursiv dargestellt wird.

# Box Model

- Elemente werden zu Kästen
- Anordnung: nebeneinander, untereinander, ineinander
- zwei Arten:
  - *block boxes*: vertikale Anordnung
  - *inline boxes*: horizontale Anordnung



# Box Model, cont.

- Zuordnung (block, inline) bei HTML bekannt
- explizite Zuweisung in XML: `display`
- mögliche Werte: `block`, `inline`, `none`

```
img { display: block; }
```

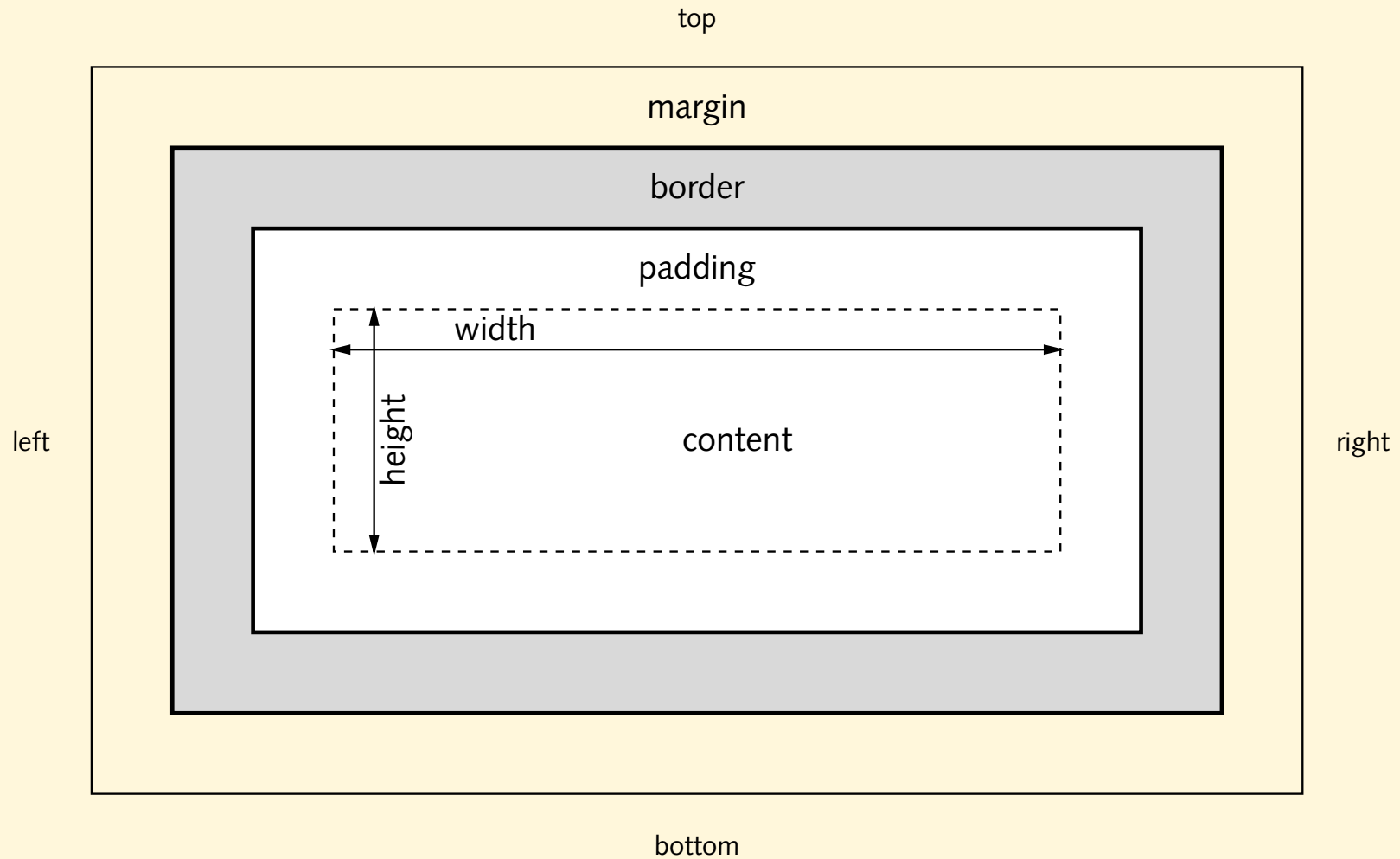
```
p.extra { display: none; }
```



# Aufgaben

- Ordne den Elementen im Gedicht sinnvolle display-Eigenschaften zu. Titel und Autor sollen in einer Zeile stehen, und die Zeilen des Gedichts sollen als solche zu erkennen sein.

# Aufbau von Kästen



# border

- Eigenschaften des Rahmens:

- `border-width:` `thin`, `medium`, `thick`, `1px`, `5em`, `2%`
- `border-style:` `solid`, `dotted`, `dashed`, `groove`, ...
- `border-color:` Farbangabe

- Kombinationen:

```
h1 { border: 2px solid blue; }
```

- einzelne Seiten definieren:

```
h1 { border-bottom-style: solid; }  
em { border-bottom: 1px dashed green; }
```

# Aufgaben

- Sorge dafür, daß der Titel unterstrichen wird.
- Füge zur Visualisierung der Kästen folgende Regel ein:

```
* { border: 1px dashed grey; }
```

Kommentiere sie später wieder aus. CSS verwendet die gleichen Kommentarzeichen wie „C“:

```
/* ... */
```

- Welche Ausdehnung haben block- und inline-Elemente?

# padding und margin

- padding: Abstand zwischen Inhalt und Rahmen
- margin: Abstand zum nächsten Kasten

```
h1 {  
  border: 2px solid red;  
  padding: 10px;  
}
```

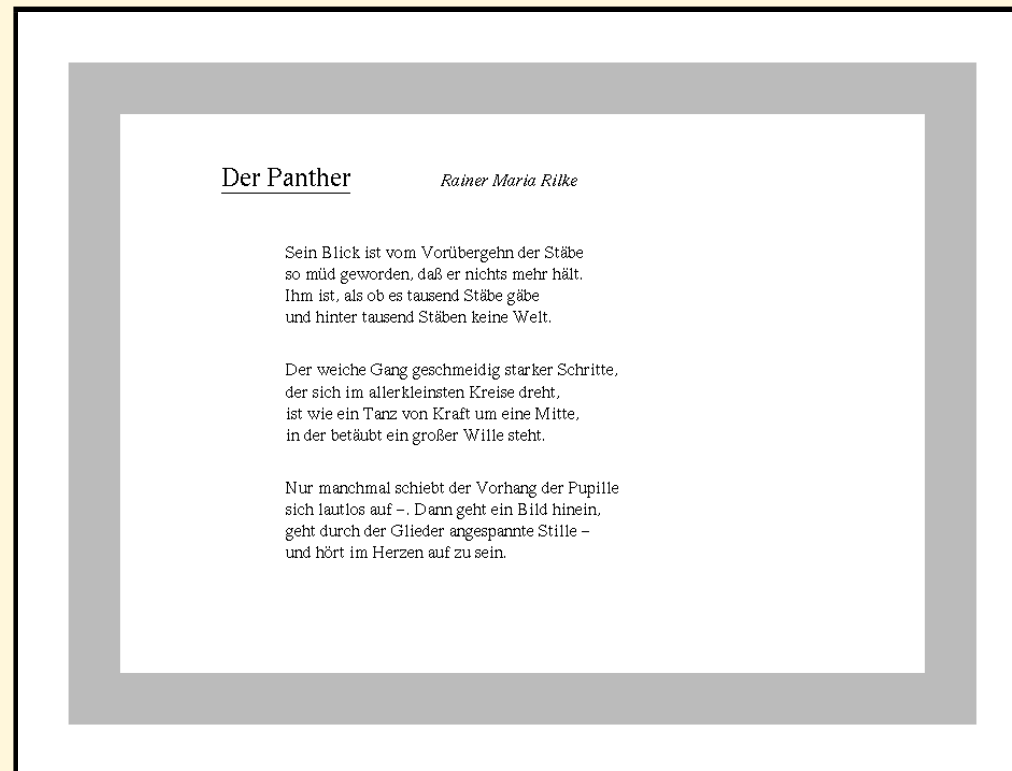
```
p {  
  margin-top: 1em;  
  margin-bottom: 1em;  
}
```

# Aufgaben

- Sorge dafür, daß man die Strophen des Gedichts erkennen kann.
- Ist es möglich, negative Werte für padding und margin anzugeben?
- Was passiert, wenn Du das padding eines gerahmten inline-Blocks auf einen relativ großen Wert (z.B. 4em) setzt?

# Aufgaben

- Versuche die CSS-Datei so zu erweitern/ändern, daß folgende Formatierung entsteht:



# Ausblick

- Grenzen von CSS:
  - XML-Struktur ändern (Sortieren, Inhaltsverzeichnis, ...)
  - Text einfügen (teilweise in CSS2)
- Extensible Stylesheet Language (XSL)
  - XSL Transformations (XSLT)
  - XSL Formatting Objects (XSL-FO)
- viele Ideen von CSS in XSL-FO aufgegriffen/erweitert
- teilweise gleiche Nomenklatur