



Netzbasierte Informationssysteme **Inhaltsdarstellung**

Prof. Dr.-Ing. Robert Tolksdorf
Freie Universität Berlin
Institut für Informatik
Netzbasierte Informationssysteme
mailto: tolk@inf.fu-berlin.de
<http://www.robert-tolksdorf.de>

Übersicht

- Trennung Inhalt und Darstellung
- Cascading Style Sheets CSS
- XML und CSS
- XSLT



Darstellung von Inhalten

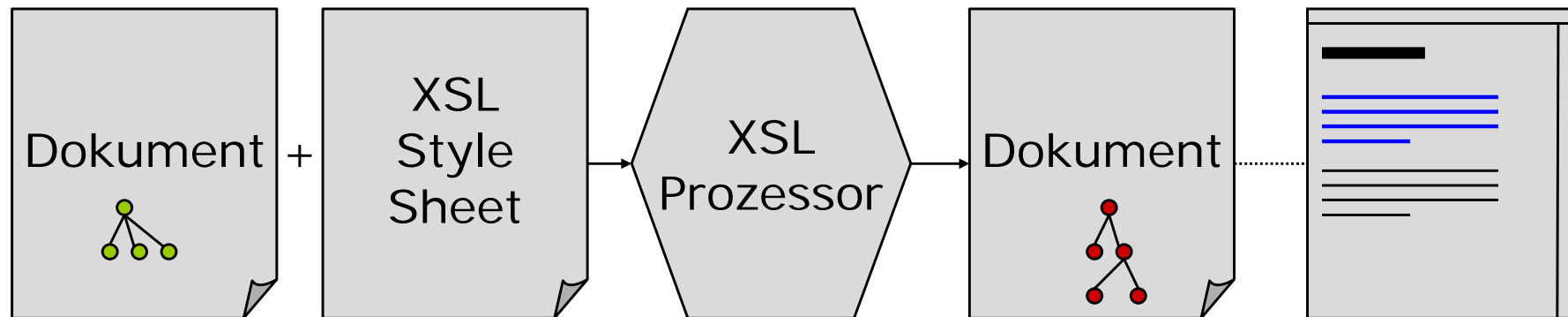
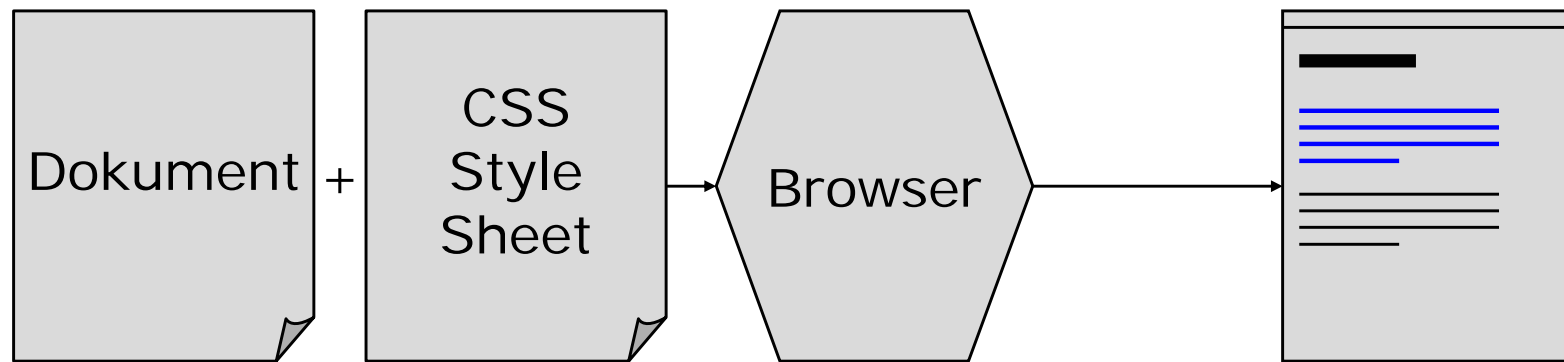
- Auszeichnungssprachen markieren
 - logische Struktur (<h1>, <p>, <table>)
 - Darstellung (, <i>, , <big>)
- Beispielhaftes Problem:
...
 - Keine Auswirkung beispielsweise mit Lynx
 - nicht lesbar bei kleinem Display
 - Suchmaschinen erkennen „Betonung“ nicht
 - nicht mit allen Seitenhintergrundfarben lesbar
 - wählt nicht vorhandenen oder nicht passenden Font aus

Inhalt und Darstellung

- Trennung von Inhalt und Darstellung ist aber wichtig für
 - Geräteunabhängigkeit von Informationen (Handy vs. PC)
 - Medienunabhängigkeit von Informationen (Grafik vs. Sprache)
 - Sprachunabhängigkeit von Informationen (" vs. „ vs. »)
 - Mehrkanal Veröffentlichungen (WAP und Web)
 - Verarbeitbarkeit von Informationen
- Notwendig:
 - Darstellungsmodelle
 - Sprachen zur Festlegung von Darstellungseigenschaften
 - Verarbeitungsmodelle für diese Sprachen

Style Sheets

- W3C Style Activity entwickelt
 - Cascading Stylesheets CSS
 - XML Stylesheet Language XSL





Cascading Style Sheets CSS

Cascading Style Sheets CSS

- ... Mechanismus zur separaten Definition von Stileigenschaften für HTML und XML Dateien (Quelle: <http://www.w3.org/Style/CSS/>)
- Cascading Style Sheets, level 1
 - Ziel: Sprache zur Definition des Darstellungsstils von HTML Dokumenten
 - Status: W3C Recommendation 17 Dec 1996, revised 11 Jan 1999
 - Quelle: <http://www.w3.org/TR/REC-CSS1>
- Cascading Style Sheets, level 2
 - Ziel: Sprache zur Definition des Darstellungsstils von HTML und XML Dokumenten für unterschiedliche Medienarten
 - Status: W3C Recommendation 12-May-1998
 - Quelle: <http://www.w3.org/TR/REC-CSS2>
- Cascading Style Sheets, level 3
 - Ziel: Modularisierte und erweiterte Sprache zur Definition des Darstellungsstils von HTML und XML Dokumenten
 - Status: „Under Construction“, unterschiedlicher Entwicklungsstand
 - Quelle: <http://www.w3.org/Style/CSS/current-work>

CSS Grundidee

- Grundidee:
Zu HTML Tags werden definierte Attribute für Darstellungseigenschaften gesetzt
- CSS-Datei getrennt von HTML-Datei gehalten
- Beispiel: Um Überschriften in großer blauer Schrift darzustellen:

```
h1 {color: blue; font-size: 22pt; }
```
- CSS definiert
 - Rahmensyntax zur Notation
 - Menge von Attributen
 - Menge von Werten
 - Bedeutung
 - Mechanismen zur Anbindung von Stilinformationen an und in HTML Seiten

CSS Anbindung an HTML

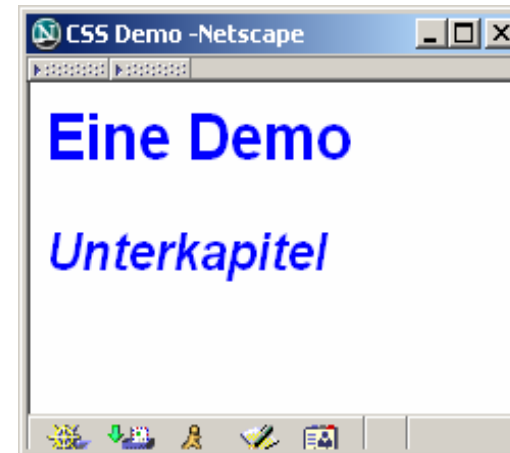
- Drei Wege der Einbindung in HTML
 - Mit externem Stylesheet über Verweis im `<link>`-Tag:
`<link rel="stylesheet" type="text/css" href="http://www.ag-nbi.de/incl ude/nbi .css">`
 - Im HTML Dokument mit dem `<style>`-Tag:
`<style>`
`h1 {color: blue; font-size: 22pt; }`
`</style>`
Kompatibel für alte Klienten:
`<style><!--`
`h1 {color: blue; font-size: 22pt; }`
`--></style>`
 - Bei den einzelnen Elementen im `style`-Attribut:
`<h1 style="color: red">Rote Überschrift</h1>`
- Einbindung innerhalb eines CSS
 - `@import url (http://www.ag-nbi.de/incl ude/col ors. css);`

CSS Gruppierungen und Vererbung

- In CSS können Angaben gruppiert werden
 - Mehrere Elemente erhalten gleiche Eigenschaften
`h1, h2, h3, h4, h5, h6 {color: blue; }`
 - Ein Element erhält mehrere Eigenschaften
`h1 {color: blue; font-size: 22pt; }`
 - Kombination
`h1, h2, h3, h4, h5, h6 {color: blue; font-style: italic}`
- Eigenschaften werden „vererbt“
 - Entlang der Element Schachtelung nach „unten“
`<h1>Das ist <u>wichtig</u></h1>`
 - Allgemeinstes Element in HTML: `<body>`
 - Eigenschaften die dort gesetzt werden, gelten für alle Elemente bei denen nichts anderes deklariert ist
 - Beispiel: Seite komplett in serifenloser Schrift
`body {font-family: arial, helvetica, sans-serif; }`

- HTML:

```
<html >
<head>
  <link rel="stylesheet" type="text/css" href=
    "cssdemo.css">
  <title>CSS Demo</title>
</head>
<body>
  <h1>Eine Demo</h1>
  <h2>Unterkapitel </h2>
</body>
```



- CSS:

```
body {font-family: arial, helvetica, sans-serif; }
h1 {color: blue; font-size: 22pt; }
h2, h3, h4, h5, h6 {color: blue; font-style: italic}
body {font-family: arial, helvetica, sans-serif; }
```

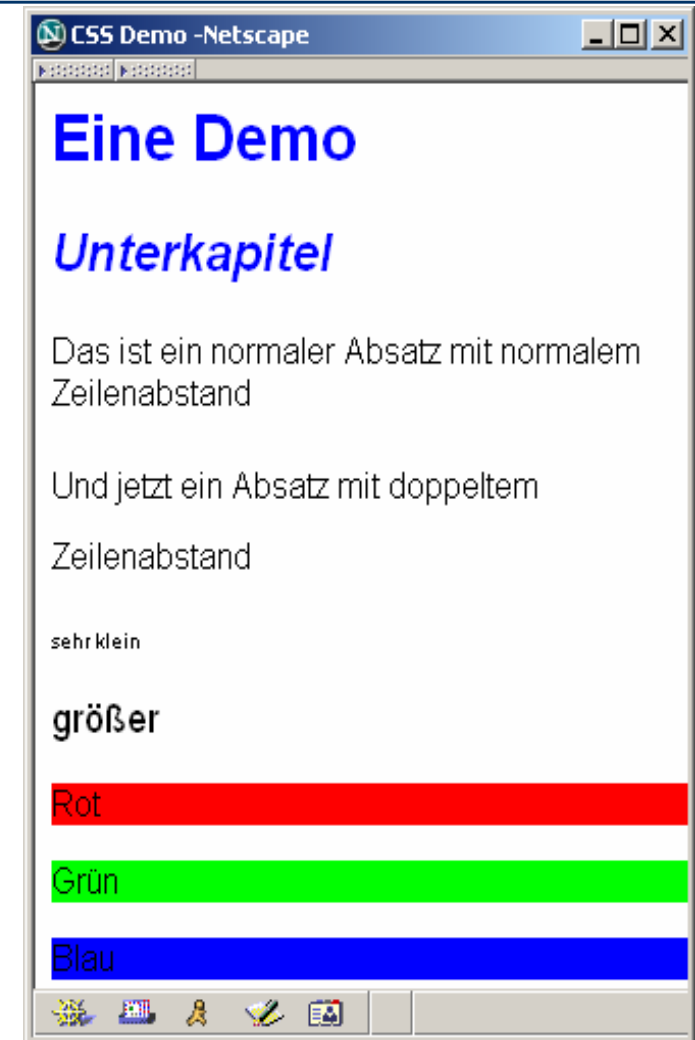
Wertetypen

- Längen ([+|-]<Zahl><Einheit>)
 - Relative Längen
 - em: Breite des M im aktuellen Font
 - ex: Höhe des x im aktuellen Font
 - px: Referenzpixel auf einem 90 dpi Gerät
 - Absolute Längen
 - in: Zoll (1in=2,54cm)
 - cm: Zentimeter
 - mm: Millimeter (10mm=1cm)
 - pt: Typographischer Punkt (1pt=1/72in)
 - pc: Pica (12pc=1pt)
- Anteilige Größen in Prozent
 - Bei Eigenschaft ist Bezug definiert
 - p { line-height: 120% }: Relativ zum aktuellen font-size
 - h1 { margin-right: 12.3% }: Relativ zum aktuellen margin-right

Wertetypen

- Schlüsselwort-Werte
 - Teilweise definiert, z.B. bei font-size
 - absolut: xx-small, x-small, small, medium, large, x-large, xx-large
 - relativ: larger, smaller
- Farbwerte
 - Farbnamen:
aqua, black, blue, fuchsia, gray, green, lime, maroon, navy,
olive, purple, red, silver, teal, white, yellow
 - Farbanteile:
 - #rgb color: #F0F
 - #rrggbb color: #FF00FF
 - rgb(val, val, val) color: rgb(255, 0, 255)
 - rgb(frac, frac, frac) color: rgb(100%, 0%, 100%)
- URLs
 - background: url (http://www.bg.com/pinkish.gif)

```
<p>Das ist ein normaler Absatz mit normaler Zeilenabstand</p>  
<p style="line-height: 200%">Und jetzt ein Absatz mit doppeltem Zeilenabstand</p>  
<p style="font-size: x-small">sehr klein</p>  
<p style="font-size: larger">größer</p>  
<p style="background-color: red">Rot</p>  
<p style="background-color: #0F0">Grün</p>  
<p style="background-color: rgb(0, 0, 100%)">Blau</p>
```



CSS Klassen

- Darstellungseigenschaften für alle Verwendungen eines bestimmten Elements gleich
- Tatsächlich aber vielleicht unterschiedliche Eigenschaften je nach Verwendung → Klassen in CSS
- Deklaration durch Punkt getrennt hinter Element:
`h1.largegreen {color: green; font-size: 22pt; }`
- Verwendung durch class-Attribut
`<h1 class="largegreen">Das ist jetzt anders</h1>`
- Deklaration von Eigenschaftsklassen ohne Elementangabe:
`.green {color: green; }`
- Verwendung bei allen Elementen möglich:
`<p class="green">Alles klar</p>`

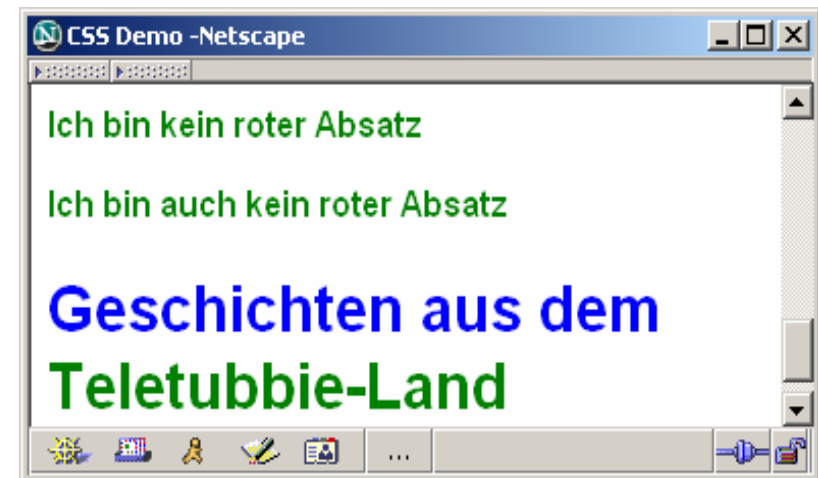
- Zwei Elemente in HTML, mit denen Stil gebunden werden kann ohne dass Darstellung erzeugt wird:
 - `<div>`: Generisches Block-Element, das andere Blöcke umschließt
 - ``: Generisches Inline-Element, das im fortlaufenden Text verwendet werden kann
- Attribute: `id`, `lang`, `dir`, `title`, `align`, `onclick`, `ondblclick`, `onmousedown`, `onmouseup`, `onmouseover`, `onmousemove`, `onmouseout`, `onkeypress`, `onkeydown`, `onkeyup`, `style`, `class`
- ```
<div class="green" >
<p>bl ah bl ah</p><p>foo foo</p>
</div>
```
- ```
<p>Das ist wirklich <span style="color: red">sehr
wichtig</span>. </p>
```

Demo

```
<p class="green"><b>Ich  
bin kein roter Absatz</b></p>
```

```
<div class="green">  
<p><b>Ich bin auch kein  
roter Absatz</b></p>  
</div>
```

```
<h1>Geschichten aus dem  
<span class="green">Teletubbie-Land</span></h1>
```



CSS Ausnahmen

- Unterschiedliche Eigenschaften je nach Kontext der Verwendung → „contextual selectors“ in CSS
- In Überschrift
`<h1>Das ist wichtig</h1>`
soll Hervorhebung farblich sein:
`h1 em { color: red }`
- Kann verknüpft werden
`ul li { font-size: small }`
`ul ul li { font-size: x-small }`
- Kann auch gemischt werden
`.red h1 {color: blue}`
→ `<h1>` in einem `<div class="red">` ist trotzdem blau

- Pseudoklassen (hier beim Anker-Element <a>)
 - a: l i n k: noch nicht besuchter Zielanker
 - a: v i s i t e d: schon besuchter Zielanker
 - a: a c t i v e: gerade ladenender Zielanker
 - a: h o v e r: Mauszeiger ist über Element
 - a: f o c u s: Element hat Eingabefocus
 - a: l i n k { t e x t - d e c o r a t i o n : n o n e ; }
 - a: h o v e r { t e x t - d e c o r a t i o n : u n d e r l i n e ; }
- Typographische Pseudoelemente
 - f i r s t - l e t t e r Erster Buchstabe
 - <p>Guten Tag</p> wird konzeptionell expandiert zu
<p><p: f i r s t l e t t e r>G</p: f i r s t l e t t e r>uten Tag</p>
 - f i r s t - l i n e Erste Zeile
 - p: f i r s t - l i n e { f o n t - f a m i l y : s e r i f }
 - The first line of an article in Newsweek.
 - f i r s t - c h i l d Erstes Unterelement
- Können mit eigenen Klassen kombiniert werden:
p: e i n l e i t u n g: f i r s t - l i n e { f o n t - f a m i l y : s e r i f }

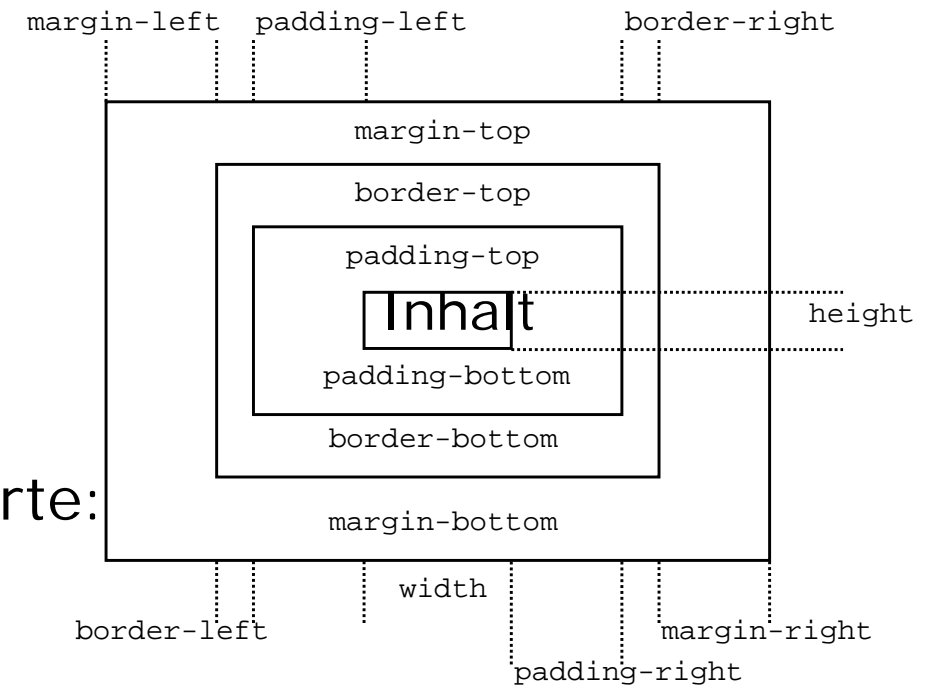
Auflösung von Konflikten

- Darstellungseigenschaften werden an verschiedenen Stellen definiert (Browser, Nutzerpräferenzen, Autor)
- Bei Konflikten gilt folgende ansteigende Gewichtung:
 - Defaults des Darstellers
 - Eventuell persönliches Stylesheet
 - Stylesheet des Dokuments
- Innerhalb von Stylesheets
 - Deklarationen mit Markierung !important wichtiger als solche ohne
(h1 { color: blue ! important ; font-size: 22pt; })
 - Deklarationen aus höher gewichteter Quelle sind wichtiger als solche aus anderen
 - Spezifischere Deklarationen gewichtiger als allgemeinere
(h1 em { color: red } vs. h1 { color: blue; })
 - Textuell spätere Deklaration gewichtiger als frühere (dabei @import am Anfang in ihrer Reihenfolge berücksichtigt)

Eigenschaften von Inhaltskästen

- Leerraum um Kasten

- margin-top: Oben
- margin-right: Rechts
- margin-bottom: Unten
- margin-left: Links
- margin: Zusammengefasste Werte:
 - margin: 1cm
Auf allen Seiten 1cm
 - margin: 1cm 2cm
Oben und unten 1cm, links und rechts 2cm
 - margin: 1cm 2cm 3cm
Oben 1cm, rechts 2cm, unten 3cm, links 2cm
 - margin: 1cm 2cm 3cm 4cm
Oben 1cm, rechts 2cm, unten 3cm, links 4cm



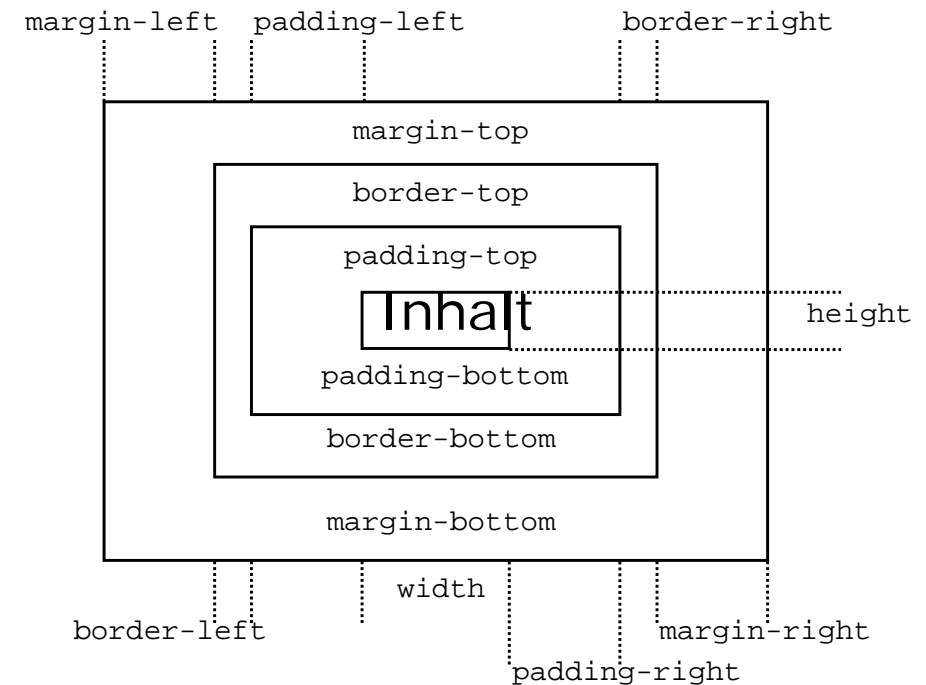
Eigenschaften von Inhaltskästen

- Leerraum um Inhalt

- padding-top: Oben
- padding-right: Rechts
- padding-bottom: Unten
- padding-left: Links
- padding:

Zusammengefasste Werte:

- padding: 1cm
Auf allen Seiten 1cm
- padding: 1cm 2cm
Oben und unten 1cm, links und rechts 2cm
- padding: 1cm 2cm 3cm
Oben 1cm, rechts 2cm, unten 3cm, links 2cm
- padding: 1cm 2cm 3cm 4cm
Oben 1cm, rechts 2cm, unten 3cm, links 4cm



Eigenschaften von Inhaltskästen

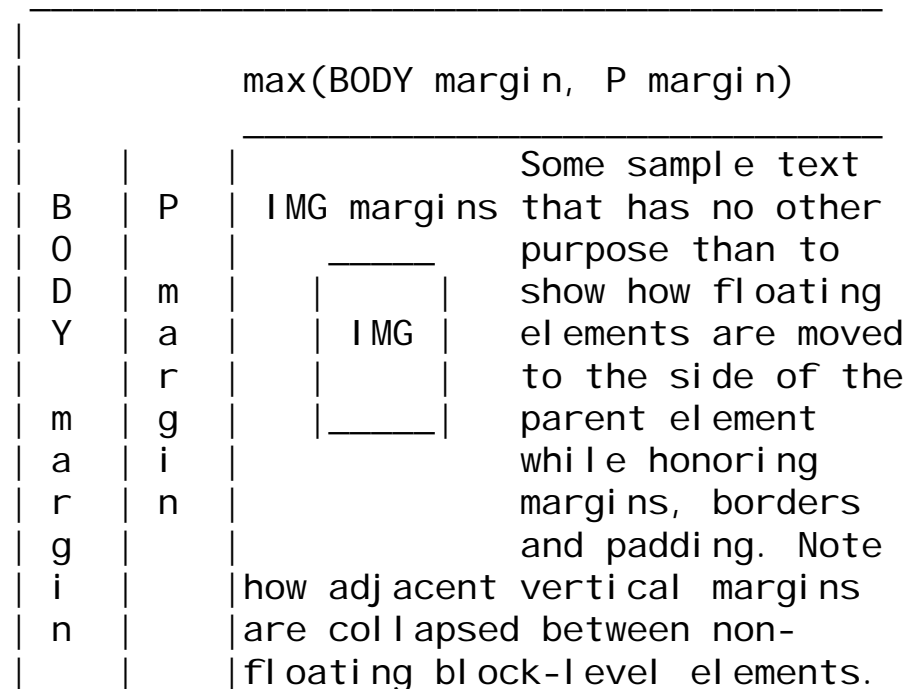
- Dicke des Rands um Inhalt + Padding
 - border-top-width: *Dicke Oben*
 - border-right-width: *Dicke Rechts*
 - border-bottom-width: *Dicke Unten*
 - border-left-width: *Dicke Links*
 - border-width: *oben rechts unten links* Zusammengefasst
 - Werte: thin, medium, thick oder Länge
- Farbe des Rands
 - border-color: *oben rechts unten links*
- Linenstil des Rands
 - border-style: *oben rechts unten links*
 - none, dotted, dashed, solid, double, groove, ridge, inset, outset
- Zusammengefasst für Seiten einzeln
 - border-top: *Dicke Linienstil Farbe*
 - Gleiches für border-right, border-bottom, border-left
 - border: *Dicke Linienstil Farbe* für alle vier Seiten gleich

Eigenschaften von Inhaltskästen

- `width`: *Länge* Breite des Inhalts
- `height`: *Länge* Höhe des Inhalts
- `float`: *Position* Inhalt kann an den Rand des enthaltenden Elements verschoben werden

```
<style type="text/css">
  img { float: left }
  body, p, img { margin: 2em }
</style>
<body>
  <p>
  Some sample text that has no
  other...</p></body>
```

- Werte: `left`, `right`, `none`
- `clear`: *Position* Ob an den Rand dieses Element Inhalt verschoben werden kann
 - Werte `none`, `left`, `right`, `both`



- CSS:

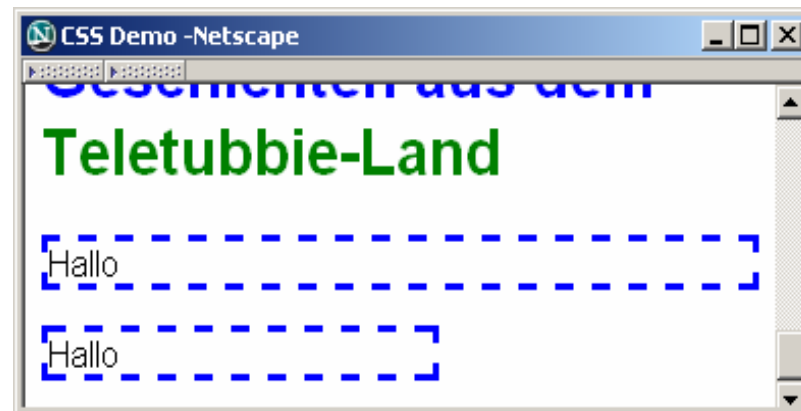
```
.aufwaendig {border: medium dashed blue; }
```

- HTML:

```
<p class="aufwaendig">Hallo</p>
```

```
<p class="aufwaendig"
```

```
style="width: 5cm; height: 2cm">Hallo</p>
```



Darstellungsklassen und Listen

- `display` Legt Darstellungsklasse eines Elements fest
 - `block`: Erzeugt Darstellungsblock (`<p>...</p>`)
 - `inline`: Wird innerhalb eines Blocks dargestellt (`<i>...</i>`)
 - `list-item`: Ist Listeneintrag (`...`)
 - `none`: Keines der obigen (``)
- Falls `display`: `list-item` gilt
 - `list-style-type` Nummerierungsart
 - Werte `disc`, `circle`, `square`, `decimal`, `lower-roman`, `upper-roman`, `lower-alpha`, `upper-alpha`, `none`
 - `list-style-image`: *URL* Bild als Markierung
 - `list-style-position` Markierungsposition relativ zum Block
 - Wert: `inside`

- Blah
 - foo blah
 - blah

outside

- Blah
 - foo
 - blah
 - `list-style` *Nummerierungsart URL Markierungsposition*
 - Kurznotation

Elementfarbe und Hintergrund

- `color`: *Farbe* Farbe des Elements
- `background-color`: *Farbe* Hintergrundfarbe
- `background-image`: *URL* Hintergrundbild
- `background-repeat` Wird das Hintergrundbild wiederholt und wie?
 - Werte: `repeat`, `repeat-x`, `repeat-y`, `no-repeat`
- `background-attachment` Ist Hintergrundbild verankert oder wird es mitgescrollt
 - Werte: `scroll`, `fixed`
- `background-position`
 - Werte: Absolute Länge oder relativ zur Umgebungsgröße
 - Schlüsselworte aus
 - `top`, `center`, `bottom`
 - `left`, `center`, `right`
 - `center right` entspricht 100% 50%

Elementfarbe und Hintergrund

- background *Farbe Wiederholung Fixierung Position*
 - Kurznotation
- Institutshomepage:

```
body {  
  color : #222222;  
  background : #ffffff  
  url (http://www.inf.fu-berlin.de/styles/institute-600x400.jpg)  
  no-repeat;  
}
```


Schrifteigenschaften

- font-family Name der Schriftfamilie
 - Generische
 - serif (Times)
 - sans-serif (Helvetica)
 - cursive (*Monotype Corsiva*) (Normale Schrift kursiv: *a*, Kursive Schrift: *a*)
 - fantasy (**Comic Sans**)
 - monospace (**Courier**, Lucida Console)
 - Konkrete Schriftnamen (Arial, Tahoma, Swiss, Garamond)
 - Auswahlversuch in Reihenfolge der Liste
`body { font-family: "new century school book", serif }`
- font-style Schriftstil
 - normal, *italic*, oblique
- font-variant Schriftvariante
 - normal, small-caps (KAPITÄLCHEN)

Schrifteigenschaften

- font-weight Stärke der Schrift (Schriftzug)
 - Absolute Werte: normal, bold, 100, 200, 300, 400, 500, 600, 700, 800, 900,
 - Relative Werte: bolder, lighter
- font-size Größe der Schrift
 - Absolute Werte: xx-small, x-small, small, medium, large, x-large, xx-large, Größenangabe
 - Relative Werte: larger, smaller
- font *Stil Variante [Stärke] Größe Zeilenabstand*
Schriftfamilie
 - Kurznotation

Texteigenschaften

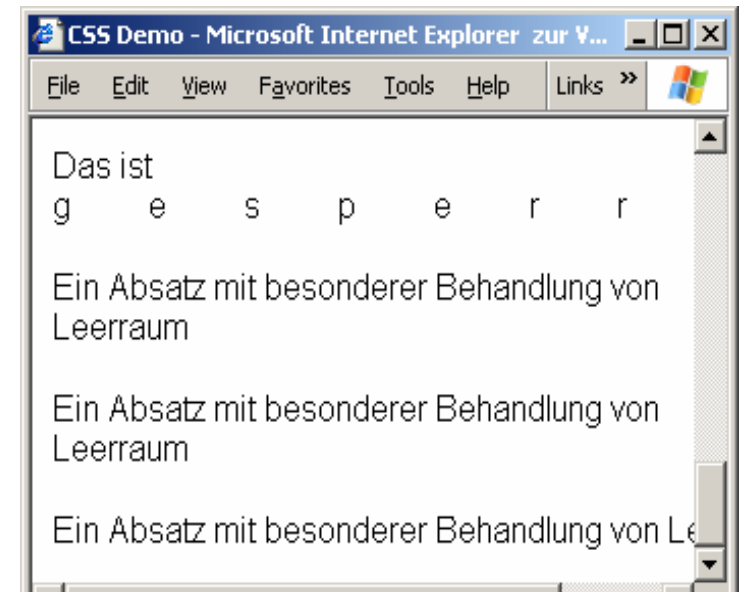
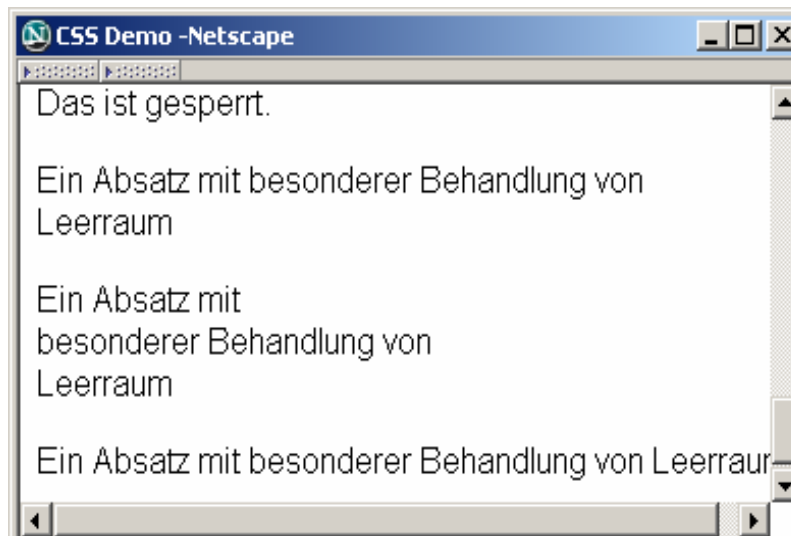
- word-spacing: *Länge*
Zusätzlicher Wortabstand
- letter-spacing: *Länge*
Zusätzlicher Buchstabenabstand (Dicke)
- white-space Legt Behandlung von Leerraum fest
 - normal: Leerzeichen fallen zusammen, automatischer Zeilenumbruch (<p>)
 - pre: Leerraum wird beachtet (<pre>...</pre>)
 - nowrap: Kein automatischer Zeilenumbruch mit


```
<p>Das ist <span style="letter-spacing:1cm">gesperrt</span>. </p>
```

```
<p style="white-space:normal">Ein Absatz mit besonderer Behandlung von Leerraum</p>
```

```
<p style="white-space:pre">Ein Absatz mit besonderer Behandlung von Leerraum</p>
```

```
<p style="white-space:nowrap">Ein Absatz mit besonderer Behandlung von Leerraum</p>
```



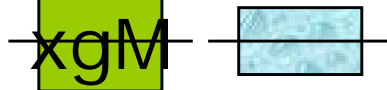
Texteigenschaften

- text-decoration Textdekoration
 - Werte: none, underline, overline, line-through, blink
- text-transform Textveränderung
 - Werte: capitalize, uppercase, lowercase, none
- vertical-align Ausrichtung eines Kastens

- baseline



- middle



- super



- sub



- text-top



- text-bottom



- top



- bottom



Texteigenschaften

- `text-align` Horizontale Ausrichtung
 - Werte: `left`, `right`, `center`, `justify`
- `text-indent`: *Länge* Einzug der ersten Zeile
- `line-height` Zeilenabstand
 - Faktor auf Schriftgröße
 - Länge
 - Prozent

- Tabellen als Darstellungsobjekt
- Eigenschaften für Tabellen
- Diverse zusätzliche Eigenschaften für Text und Schriften
- ...

CSS2: Medienarten

- Darstellungsstil ist abhängig vom Ausgabemedium
 - Bildschirm
 - Papier
 - Spache
 - Braille
 - ...
- CSS erlaubt getrennte Stildefinitionen:

...

```
a:link {  
  color: #000099;  
  text-decoration: none ;  
}
```

```
@media print {
```

```
  a:link, a:visited, a: hover, a: active, a: focus {  
    text-decoration: none;  
    color: blue  
  }
```

```
}
```


Definierte Medienarten

- Medienarten:
 - al l
 - aural Sprachausgabe
 - brai l l e Taktile Ausgabe
 - embossed Braille Drucker
 - handhel d Klein, monochrom
 - pri nt Drucker
 - proj ecti on Beamer, Foliendruck
 - screen Bildschirm
 - tty Textterminals
 - tv Geringe Auflösung eingeschränkte Interaktion

Mediengruppen

- Darstellungseigenschaften müssen jeweils von Darstellern für bestimmte Mediengruppen beachtet werden

Media Types	Media Groups			
	continuous/paged	visual/aural/tactile	grid/bitmap	interactive/static
aural	continuous	aural	N/A	both
braille	continuous	tactile	grid	both
emboss	paged	tactile	grid	both
handheld	both	visual	both	both
print	paged	visual	bitmap	static
projection	paged	visual	bitmap	static
screen	continuous	visual	bitmap	both
tty	continuous	visual	grid	both
tv	both	visual, aural	bitmap	both

- Zusätzliche Eigenschaften
 - Seite ist umgebender Darstellungskasten
 - page- Eigenschaften
 - Es treten Seitenumbrüche auf
 - Eigenschaften page-break-before, page-break-after, page-break-inside
 - Werte: auto, always, avoid, left, right
 - Eigenschaften orphans, widows Hurenkinder/Schusterjungen

Schusterjunge / widow



Hurenkind / orphan



Darstellung gesprochener Sprache

- Darstellungseigenschaften betreffen
 - Ton
 - Dauer und Reihenfolge
 - Spracheigenschaften

- Beispiel

```
H1, H2, H3, H4, H5, H6 {  
    voi ce-fami l y: paul ;  
    stress: 20;  
    ri chness: 90;  
    cue-before: url ("pi ng. au")  
}  
P. hei di { azi muth: center-l eft }  
P. peter { azi muth: ri ght }  
P. goat { vol ume: x-soft }
```

Zusammenfassung

- CSS trennt Inhalt und Darstellung
- CSS ist getrennte Sprache für Darstellungseigenschaften
- Vielfältige Einstellungsmöglichkeiten
- CSS2: Erweiterung auf andere Medienarten



XML, SVG und CSS

XML + CSS

- Cascading Style Sheets definieren Darstellung von Tags durch Belegen von CSS-Attributen
- Während ursprünglich für HTML entworfen auch für XML nutzbar
- Darstellung vom eigenen Element `<price>` weiss auf schwarz:

```
price {  
    color: white;  
    background-color: black;  
}
```
- CSS Attribute für visuelle oder auditive Ausgabe von Texten geeignet
- www.w3.org/1999/06/REC-xml-styleSheet-19990629

XML-Sprachen

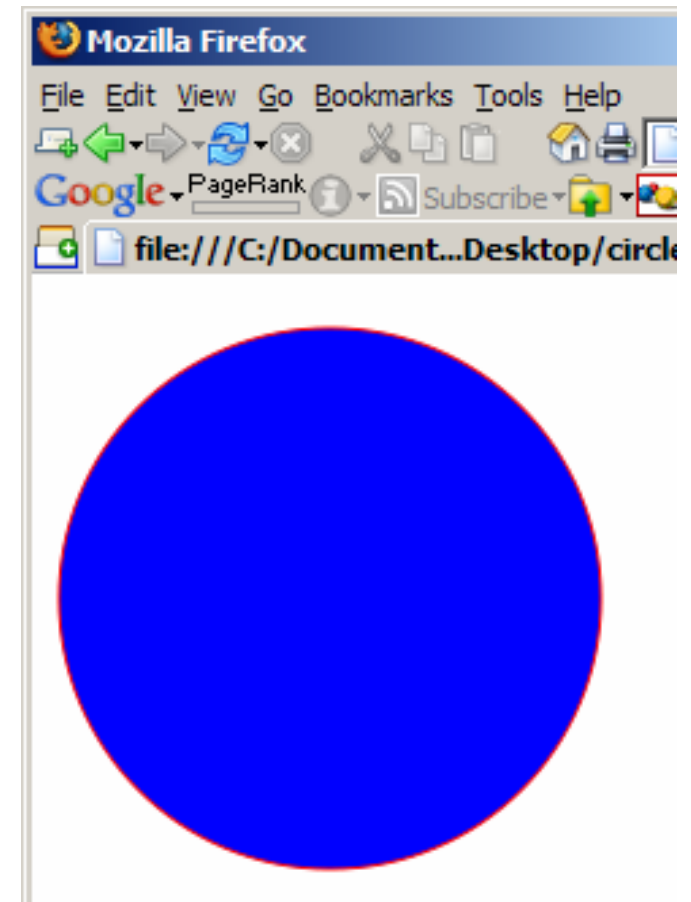
- XML-Sprachen definieren anwendungsabhängige Elemente
- Ihre Darstellung ist auch anwendungsabhängig
- CSS kann nicht alle anwendungsabhängigen Darstellungen kennen
- -> Weitere CSS Module je nach definierter Sprache

Scalable Vector Graphics (SVG)

- Scalable Vector Graphics (SVG) ist standardisiertes Zeichnungsformat (<http://www.w3.org/Graphics/SVG>)

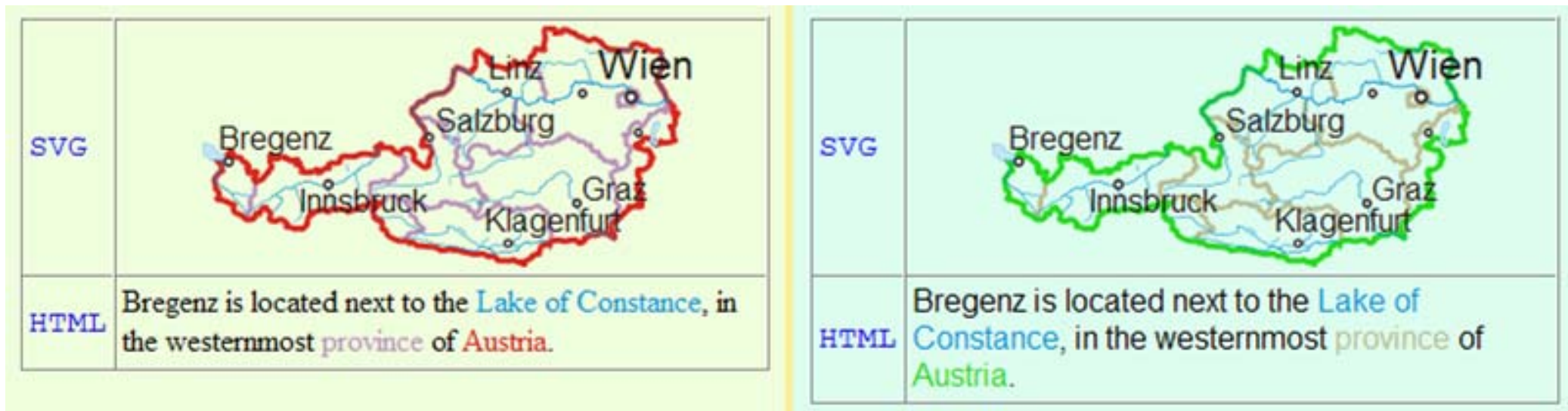
- ```
<svg width="3in" height="3in">
 <desc>A blue circle with a
 red outline</desc>
 <g><circle cx="110"
 cy="120" r="100"
 style="fill: blue;
 stroke: red" />

 </g>
</svg>
```



# Scalable Vector Graphics (SVG)

- Überführung visueller Eigenschaften in Stildefinitionen:
  - `<ellipse cx="110" cy="60" rx="100" ry="50" fill="blue" stroke="black" stroke-width="4" />`
  - `<ellipse cx="110" cy="60" rx="100" ry="50" style="fill: blue; stroke: black; stroke-width: 4" />`  
 [http://www.selfsvg.info/?section=4.8]
- Mit externen Stylesheets:



[http://www.carto.net/papers/svg/samples/styles.shtml]

# Scalable Vector Graphics (SVG)

---

- SVG übernimmt CSS2 Eigenschaften:
  - Schriften:  
'font', 'font-family', 'font-size', 'font-size-adjust', 'font-stretch', 'font-style', 'font-variant', 'font-weight'
  - Text:  
'direction', 'letter-spacing', 'text-decoration', 'unicode-bidi', 'word-spacing'
  - Etc:  
'clip', 'color', 'cursor', 'display', 'overflow', 'visibility'
- <http://www.w3.org/TR/SVG/styling.html>

# Scalable Vector Graphics (SVG)

- SVG definiert folgende neue Stileigenschaften
  - Bildkomposition:  
'clip-path', 'clip-rule', 'mask', 'opacity'
  - Filter:  
'enable-background', 'filter', 'flood-color', 'flood-opacity',  
'lighting-color'
  - Verlauf:  
'stop-color', 'stop-opacity'
  - Interaktivität:  
'pointer-events'
  - Farbe:  
'color-interpolation', 'color-interpolation-filters', 'color-profile',  
'color-rendering', 'fill', 'fill-opacity', 'fill-rule', 'image-rendering',  
'marker', 'marker-end', 'marker-mid', 'marker-start', 'shape-  
rendering', 'stroke', 'stroke-dasharray', 'stroke-dashoffset',  
'stroke-linecap', 'stroke-linejoin', 'stroke-miterlimit', 'stroke-  
opacity', 'stroke-width', 'text-rendering'
  - Text:  
'alignment-baseline', 'baseline-shift', 'dominant-baseline', 'glyph-  
orientation-horizontal', 'glyph-orientation-vertical', 'kerning',  
'text-anchor', 'writing-mode'



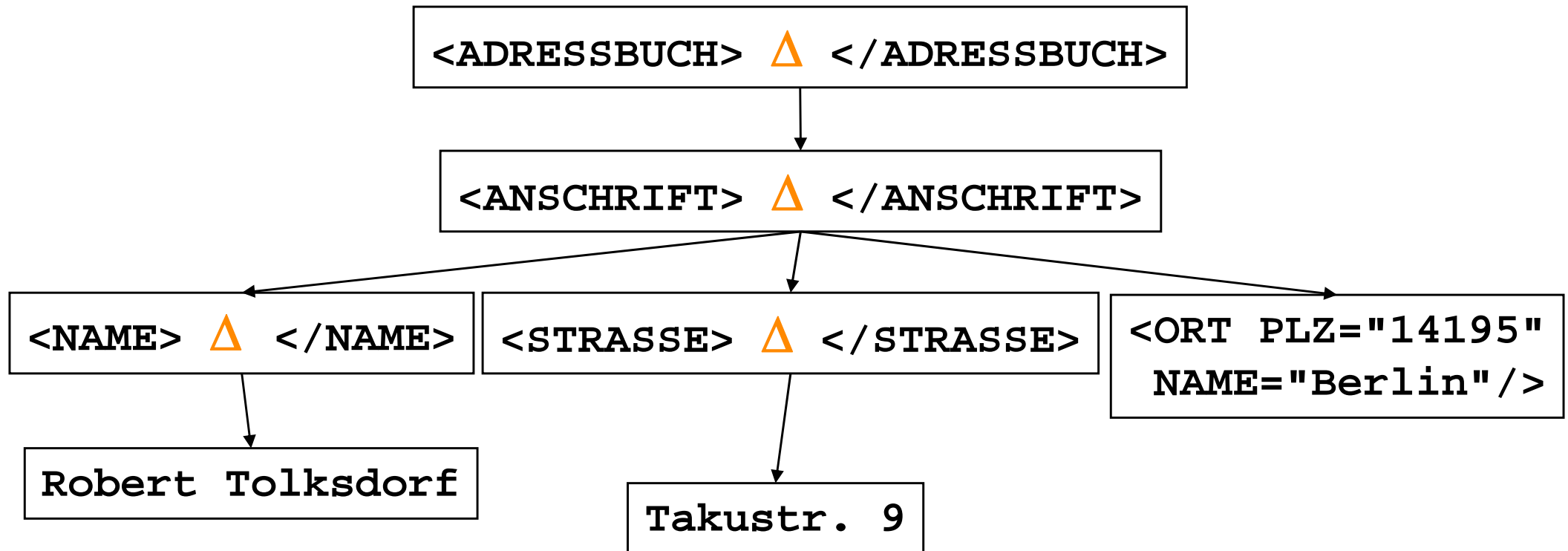
## **XSL, XSLT, XSL-FO**

- CSS Vorgehen:
  - HTML enthält Struktur in Inhalt
  - CSS definiert Darstellungseigenschaften
  - Struktur fest
- XSL Standard
  - XML enthält Inhalt
  - XSL transformiert in Darstellungsstruktur (und deren Eigenschaften)

- Extensible Stylesheet Language (XSL)
  - XSL Transformations (XSLT)
    - Zweck: Transformation von XML Dokumenten
    - Status: W3C Recommendation 16 November 1999
    - Quelle: <http://www.w3.org/TR/xslt>
  - XML Path Language (XPath)
    - Zweck: Ausdrücke mit denen Stellen im XML Dokument bezeichnet werden können
    - Status: W3C Recommendation 16 November 1999
    - Quelle: <http://www.w3.org/TR/xpath>
  - XSL Formatting Objects
    - Zweck: Pseudobäume mit Darstellungseigenschaften
    - Status: W3C Recommendation 15 October 2001
    - Quelle: <http://www.w3.org/TR/xsl/>

# XSL(T)

- XML-Dokumente sind Bäume:



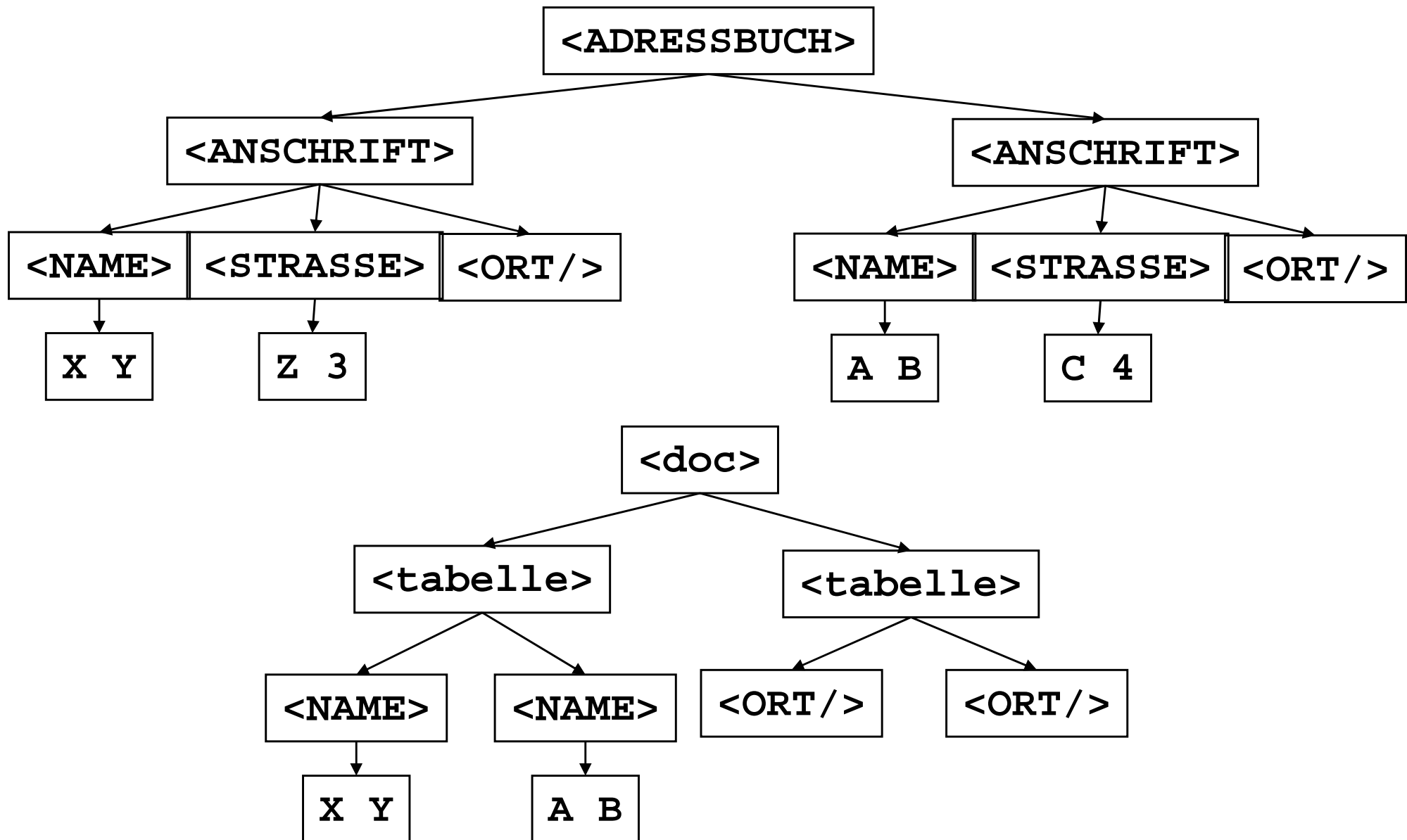
- XSLT ist eine Sprache zur Transformation eines XML-Baums in einen anderen ([www.w3.org/Style/XSL](http://www.w3.org/Style/XSL))



# XSL Regeln

- Eine XSL-Regel definiert Muster und dazugehörige Transformationen
- Sie wird auf alle passenden Knoten im Quelldokument angewandt
- Beispiel:

```
<xsl:template match="ANSCHRIFT">
 <tabelle>
 <xsl:apply-templates select="NAME" />
 </tabelle>
 <tabelle>
 <xsl:apply-templates select="ORT" />
 </tabelle>
</xsl:template>
```

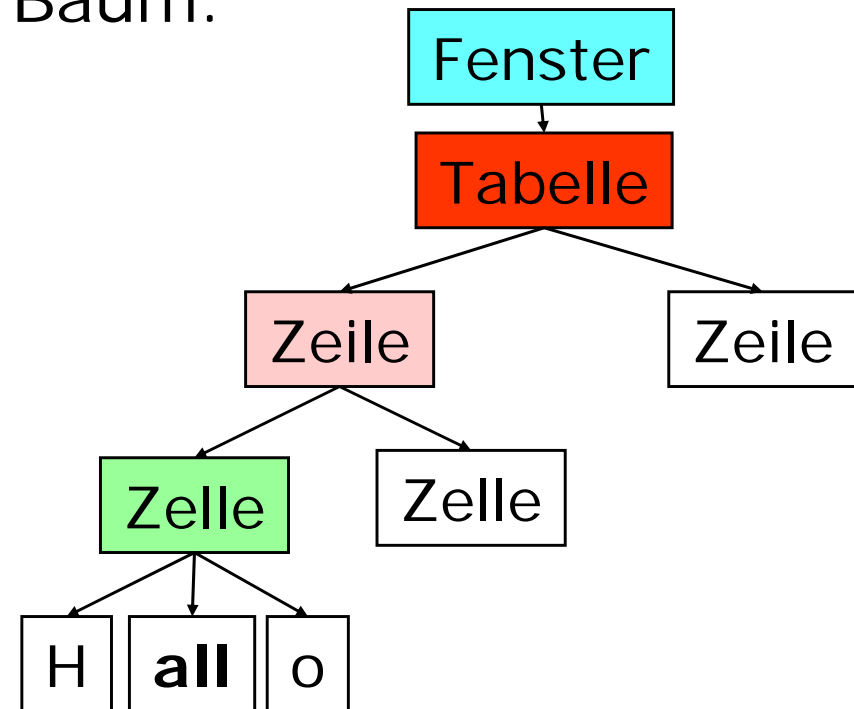
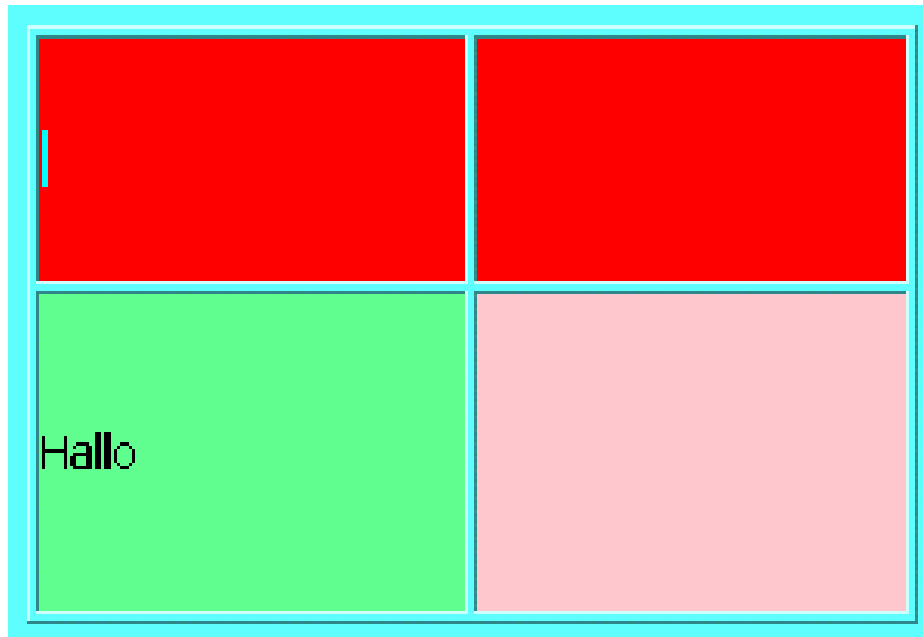


- Zählen aller Kinder eines Knotens:

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl =
 "http://www.w3.org/1999/XSL/Transform">
 <xsl:template match="/|/*">
 <xsl:value-of
 select="count(descendant::*)" />
 Elemente
 </xsl:template>
</xsl:stylesheet>
```

# Formatting Objects

- Bildschirmdarstellung ist auch Baum:



- Formatting Objects definiert Knoten und Attribute als Ziel einer Transformation
- Als "Nebeneffekt": Bildschirmdarstellung, PDF Generierung

# Zusammenfassung

---

- XSLT transformiert XML Bäume in andere XML Bäume
- Darstellung als „Nebeneffekt“ in Formatting Objects