

# XML

## Teil 4: Cascading Style Sheets (CSS)

Abteilung Informatik – WS 02/03

# W3C-Empfehlungen

---

- Nicht XML-Syntax zur Beschreibung der Formatierung von XML-Dokumenten
- 3 Stufen:
  - ▶ Stufe 1 (engl.: CSS level 1): W3C-Empfehlung von 1996 für HTML
  - ▶ Stufe 2 (engl.: CSS level 2): aktuelle W3C-Empfehlung
  - ▶ Stufe 3 (engl.: CSS level 3): in Bearbeitung

# CSS-Syntax

---

- CSS-Dokument: Liste von Elementen mit der entsprechenden Formatierungsinformation

- Beispiel:

```
recipe {font-family: Times New Roman; font-size: 12pt }
```

```
dish   {display: block; font-family: Times New Roman }
```

- Für ein Element in einem Namensraum muss der qualifizierte Name verwendet werden
- Kommentare beginnen mit `/*` und enden mit `*/`

# Beispiel

---

```
/* Defaults for the entire document */
recipe {font-family: "New York", "Times New Roman", serif;
        font-size: 12pt }

/* Make the dish look like a headline */
dish {
  display: block;
  font-family: Helvetica, Arial, sans-serif; font-size: 20pt;
  font-weight: bold; text-align: center}

/* A bulleted list */
ingredient {display: list-item}

/* Format these two items as paragraphs */
directions, story {
  display: block; margin-top: 12pt; margin-left: 4pt}
```

# CSS-Stylesheet Deklaration

---

- `xml-stylesheet`-Prozess-Anweisung im Prolog des XML-Dokuments
- Zwei zwingende und vier optionale Pseudoattribute:
  - ▶ **type**, zwingend, Wert: `text/css` (MIME-Type der Stylesheet)
  - ▶ **href**, zwingend, Wert: URL der Stylesheet
  - ▶ **charset**, optional, Wert: Charaktermenge der Stylesheet
  - ▶ **name**, optional, Wert: Name der Stylesheet
  - ▶ **alternate**, optional, Wert: `yes`, falls es eine alternative Stylesheet ist, `no` sonst. Default: `no`
  - ▶ **media**, optional, Wert: `all`, `screen`, `tv`, `print`, `braille`, ...

# Auswahlregel

---

- Frage: Beschreibung der Auswahl der Elemente, welche einer gegebenen Formatierungsregel zugeordnet werden muss
- 1. Möglichkeit: Liste der Elementennamen, durch ein Komma getrennt. Beispiel:  
`directions, story {display: block}`
- 2. Möglichkeit: Verwendung eines Selektors

# Universeller Selektor

---

- Syntax:

  - \* *{Regel}*

- Beispiel: \* {font-size: large}

- Bedeutung: Die Regel gilt für alle Elemente

# Descendants, Children, Siblings (1/2)

---

## ■ Descendants

- ▶ Syntax: `element1 element2 {style}`
- ▶ Bedeutung: die Regel *style* gilt für die Elemente `element2`, welche Nachfolger eines Elements `element1` sind
- ▶ Beispiel: `ingredients quantity {font-size: medium}`

## ■ Children

- ▶ Syntax: `element1 > element2 {style}`
- ▶ Bedeutung: die Regel *style* gilt für die Elemente `element2`, welche Child-Elemente eines Elements `element1` sind
- ▶ Beispiel: `ingredients > quantity {font-size: inherit}`



# Descendants, Children, Siblings (2/2)

---

## ■ Siblings

- ▶ Syntax: `element1 + element2 {style}`
- ▶ Bedeutung: die Regel *style* gilt für ein Element `element2`, welches das Geschwister-Element (engl. Sibling) eines Elements `element1` ist
- ▶ Beispiel: `directions + story {border-top-style: solid}`

# Attribut-Selektor (1/2)

---

- Bedeutung: Auswahl eines Attributs oder eines Attributswerts
- Beispiele
  - ▶ `step[optional] {display: italic}`  
Auswahlbedingung: `step`-Element mit `optional`-Attribut
  - ▶ `step[optional= "yes"] {display: italic}`  
Auswahlbedingung : `step`-Element, wenn das `optional`-Attribut den Wert `yes` hat
  - ▶ `*[optional] {display: italic}`  
Auswahlbedingung : Irgendein Element mit `optional`-Attribut

# Attribut-Selektor (2/2)

---

## ■ Beispiele (Fortsetzung):

▶ `recipe[source~= "Anderson"] {display: italic}`

Auswahlbedingung : `recipe` -Element, wenn das `source`-Attribut das Wort `Anderson` enthält

▶ `recipe[source|= "Anderson"] {display: italic}`

Auswahlbedingung : `recipe`-Element, wenn das `source` -Attribut wie folgt aussieht: `Anderson-Wort1-Wort2-...`  
(z.B. `Anderson-Harold`)

# Pseudoklasse-Selektor

---

- Auswahlregel für Elemente, ohne Verweis auf ihren Namen
- Syntax: *element:selektor {style}*
- Beispiele:

## **first-child**-Pseudoklasse-Selektor

- ▶ Auswahlbedingung: Erstes Child-Element des betroffenen Elements
- ▶ Beispiel: `step:first-child {font-style: italic}`

## **link**-Pseudoklasse-Selektor

- ▶ Auswahlbedingung: Das Element ist ein noch nicht besuchtes Link
- ▶ Beispiel: `*:link {color: blue; text-decoration: underline}`

# Pseudoelement-Selektor

---

- Auswahlregel für Nicht-Elemente:
  - ▶ `first-letter`: wählt den ersten Buchstaben eines Elements aus
  - ▶ `first-line`: wählt die erste Zeile eines Blocks aus
  - ▶ `before`: wählt den „zero-width“-Punkt aus, der gerade vorher kommt (in Zusammenhang mit der `content`-Eigenschaft)
  - ▶ `after`: wählt den „zero-width“-Punkt aus, der gerade nachher kommt
- Syntax: `element:selektor {style}`
- Beispiel:
  - ▶ `story:first-letter {font-size: 200%}`
  - ▶ `ingredients: before {content: "Ingredients!"}`

# CSS-Eigenschaften

---

- CSS-Eigenschaften haben Namen und Werte
- Beispiele:
  - ▶ Name: `font-size`, Wert: `12pt`
  - ▶ Name: `color`, Wert: `#cc003`

# display-Eigenschaft

---

- **display-Eigenschaft:** Darstellung des Elements
- **Syntax:**  
`element{display: value}`
- **Beispiel:**  
`step{  
 display: list-item;  
 list-style-type: decimal;  
 list-style-position: inside  
}`

# display-Eigenschaftswerte

---

## ■ Mögliche Werte

- ▶ inline
- ▶ block
- ▶ list-item
- ▶ hidden
- ▶ table, inline-table, table-row-group, table-header-group, table-footer-group, table-row, table-column-group, table-column, table-cell, table-caption

## ■ Beispiele

- ▶ 

```
step{
  display list-item; list-style-type: decimal; list-style-
  position: inside}
```
- ▶ 

```
story {display: none}
```
- ▶ 

```
ingredients {display: table}
ingredient {display: table-row}
quantity, components {display: table-cell}
```



# Längeneinheiten

---

- CSS-Eigenschaften, welche Längen darstellen:  
`border-width, font-size, line-height, left, top, height, width, etc.`

- Beispiele:

```
dish{font-height: 0.5cm}
```

```
dish{font-height: 150%}
```

# Font-Eigenschaften

---

- CSS definiert sieben Eigenschaften, um die Schriftart (engl. Font) zu beschreiben: `font-family`, `font-style`, `font-size`, `font-variant`, `font-weight`, `font-stretch`
- Beispiel:

```
dish{
    font-family: Helvetica, "Times New Roman", serif;
    font-size: x-large;
    font-style: italic;
    font-variant: small-cap;
    font-weight: 900;
    font-stretch: semi-expanded
}
```

# Text-Eigenschaften

---

- CSS definiert fünf Eigenschaften, um die Textformatierung zu beschreiben: `text-indent`, `text-align`, `text-decoration`, `text-transform`, `white-space`

- **Beispiel:**

```
title{
    text-indent: 0.5cm;
    text-align: left;
    text-decoration: underline;
    text-transform: capitalize;
    white-space: normal
}
```

# Farben

---

- CSS definiert verschiedene Eigenschaften, um die Farbe zu beschreiben: `color` (für den Text), `background-color`, `border-color`.
- CSS unterstützt einen 24-Bit-Farbenraum, jede Farbe ist durch ihre grüne, blaue und rote Komponente definiert (jede Komponente ist eine Zahl zwischen 0 und 255). Zum Beispiel: Grün ist gegeben durch `#FF0000` oder `rgb(255, 255, 255)`

# Referenz

---

- E. Harold, W. Means: *XML in a Nutshell*, Kapitel 12