

Grundlagen der Cascading Style Sheets (CSS)

Sommersemester 2021

Prof. Dr.-Ing. Thomas Barth

Anwendungsentwicklung und IT-Systemarchitektur

Studiengang Wirtschaftsinformatik

RFH Köln

barth@rfh-koeln.de

- ✓ Motivation und Zielsetzung von CSS kennen
- ✓ Trennung von Inhalt, Struktur (HTML) und Darstellung (CSS) kennen
- ✓ Optionen zur Verwendung von CSS mit HTML kennen
- ✓ CSS-Regeln formulieren können
- ✓ Selektoren lesen und verstehen können
- ✓ Selektoren selbst schreiben können
- ✓ Ausgewählte CSS-Eigenschaften kennen und verwenden können
- ✓ CSS-Stylesheets selbst schreiben können
- ✓ HTML-Dokumente mit CSS-Dateien erstellen können
- ✓ Layout von HTML-Dokumenten erstellen können

Kurze Wiederholung: Ziele von HTML

✓ In welche Bestandteile läßt sich eine „typische Webseite“ zerlegen?

➤ Inhalt

- Text
- Grafik
- Verknüpfung
- Interaktionselemente
 - Knöpfe
 - Texteingabe

➤ Struktur

- Seiten
- Abschnitte
- Überschriften

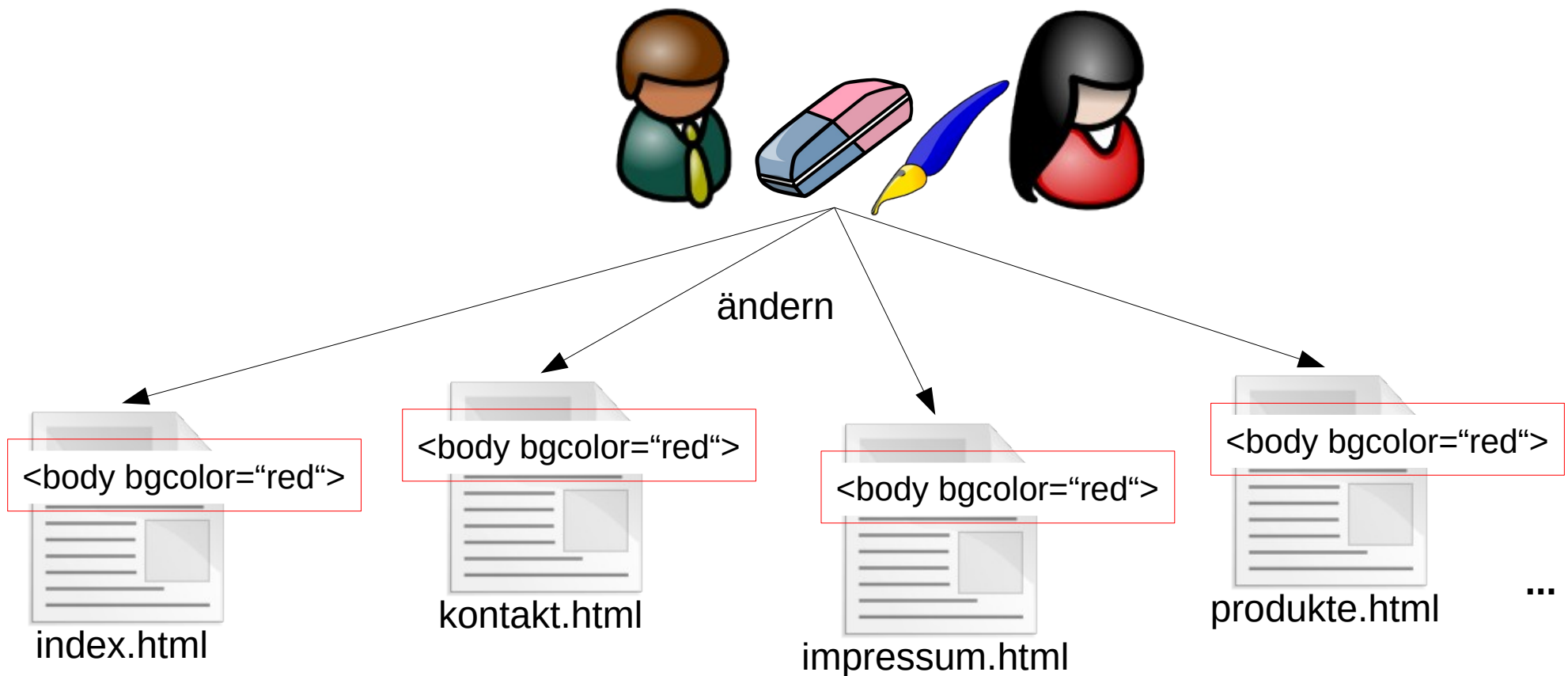
➤ Präsentation

- Vordergrund-/Hintergrundfarbe
- Schriftgröße
- Stil

- ✓ HTML bietet Möglichkeiten zur Nutzung bzw. Steuerung aller dieser Aspekte einer Webseite
- ✓ Ein HTML-Dokument enthält also ...
 - ... den eigentlichen **Inhalt**, der auf der Webseite dargestellt werden soll
 - ... Informationen zur **Struktur** der Seite, in der die Inhalte verankert sind
 - ... eingebetteten **Code** für Client-seitige Funktionalität
 - ... Informationen zur **Darstellung/Präsentation** für den Betrachter
- ✓ Die **Präsentation** des Inhalts kann zum Einen **durch HTML** bestimmt werden
 - Überschriften, Zitate, Code-Fragmente, Absätze usw.
 - Textformatierung (Fettdruck, hoch-/tiefgestellte Zeichen, Schriftfarbe, Font usw.)
- ✓ Um die Darstellung vom HTML-Dokument zu trennen (und damit die einheitliche Pflege einer größeren Menge von Seiten zu vereinfachen), kann die Bestimmung der Darstellung in **Cascading Style Sheets (CSS)** ausgelagert werden
- ✓ CSS-Dateien können zentral gepflegt werden und damit die Darstellung aller HTML-Dokumente ändern, die sich auf diese CSS-Dateien beziehen

- ✓ Die **Trennung von Inhalt und Darstellung** ist bei größeren Web-Präsenzen essentiell für Wartbarkeit und Einheitlichkeit der einzelnen Web-Seiten
- ✓ HTML berücksichtigt diese Trennung bereits, aber nur innerhalb des Dokuments
- ✓ Änderungen am Design, die für alle Seiten einer Web-Präsenz gültig sein sollen, bedeuten in HTML die Änderung ALLER Seiten, d.h.
 - Pflegeaufwand hoch
 - Fehleranfälligkeit hoch
- ✓ Prinzipiell gibt es drei Möglichkeiten, die Präsentation durch CSS zu definieren:
 - „**inline**“: Verwendung des HTML-Attributs `style` in **JEDEM HTML-Element**
 - „**intern**“: Verwendung des HTML-Elements `<style>` innerhalb von `<head>` **JEDER Seite**
 - „**extern**“: Zusammenfassung aller Darstellungseinstellungen **in einer separaten CSS-Datei (Endung .css)**
- ✓ Bzgl. Pflegeaufwand und Fehleranfälligkeit ist die Verwendung von externen CSS-Dateien zu bevorzugen

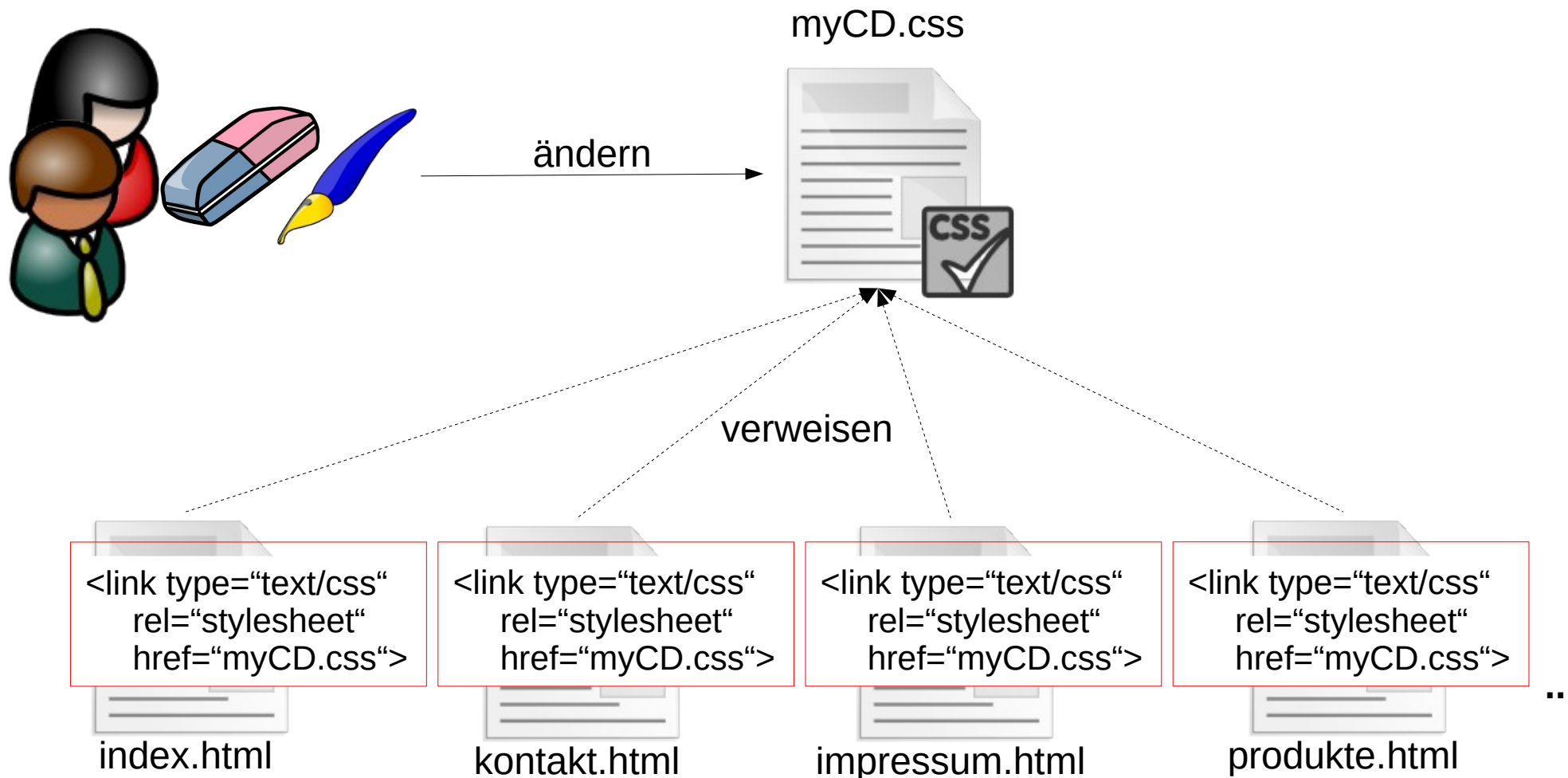
- ✓ Sowohl die Verwendung ausschließlich von HTML als auch die inline- oder interne Verwendung von CSS führt zu der folgenden Situation



Cascading Style Sheets (CSS)

✓ Lösungsansatz:

- Auslagerung aller Informationen zur Darstellung an eine zentrale Stelle/in eine oder mehrere zentrale Datei(en), die von allen HTML-Dokumenten genutzt werden



- ✓ CSS wurde 1997 durch das W3C zusammen mit HTML Version 4.0 eingeführt, um die Trennung von Inhalt und Darstellung zu verbessern
- ✓ Seitdem soll(!) die Darstellung statt in HTML über CSS-Beschreibung festgelegt werden
- ✓ HTML hat sich (bis Version 3.2) darauf beschränkt, Inhalt und dessen Struktur (z.B. Überschrift + Text + Absatz + Text + Verknüpfung + ...) zu definieren
- ✓ Mit den hinzugefügten Möglichkeiten zur Textformatierung wurde ein weiterer Aspekt hinzugefügt, der die Komplexität erhöht und die Wartbarkeit verringert hat
- ✓ Realisierung und Pflege größerer Web-Präsenzen nur auf Basis von HTML nicht mehr Stand der Technik (Status „veraltet“ seit HTML4, „unerwünscht“ seit HTML5)
- ✓ Stattdessen Nutzung von CSS, das von allen gängigen Browsern unterstützt wird
- ✓ CSS spielt auch bei der wichtiger werdenden Darstellung von Web-Dokumenten auf **mobilen Geräten** eine zentrale Rolle

Cascading Style Sheets (CSS)

- ✓ CSS besteht prinzipiell aus einer Menge von **Regeln**
- ✓ Jede dieser Regel kann übersetzt werden in:

*„für jedes HTML-Element der Art X hat die Darstellung folgende Eigenschaften:
Eigenschaft1=Wert1 und Eigenschaft2=Wert 2 und ...“*

- „**HTML-Element der Art X**“ könnte dabei z.B. für **genau eine Überschrift** <h1> oder für **ALLE Überschriften** <h1> im Geltungsbereich der Regel stehen!
- „**Eigenschaft1**“ könnte für die Schriftfarbe („color“) stehen und sie auf den Wert „red“ setzen
- ✓ Damit werden diese Eigenschaften aller angegebenen HTML-Elemente gesetzt
- ✓ Jede CSS-Regel besteht daher aus ...
 - dem **Selektor**, der die HTML-Elemente auswählt, für die die Regel gelten soll und
 - einer Aufzählung von **Deklarationen, Paaren der Form „Eigenschaft=Wert“**, wobei die Eigenschaft für das ausgewählte HTML-Element definiert sein muss

Cascading Style Sheets (CSS)

- ✓ Die Syntax jeder CSS-Regel ist wie folgt:

Selektor { Eigenschaft1:Wert1 ; Eigenschaft2 :Wert2 ; ...; }

- ✓ Beispiel für Selektoren (hier: **Typ-Selektor**):

- Die folgende Regel bewirkt, dass der Text **jedes Absatzes** (d. h. Selektor wählt alle HTML-Tags des Typs `<p>` aus) die **Schriftfarbe blau** (*erste Deklaration*) erhält und **linksbündig** (*zweite Deklaration*) ausgegeben wird

`p {color:blue; text-align:left;}`

- ✓ Regeln können auch über mehrere Zeilen hinweg notiert werden, um die Übersichtlichkeit zu erhöhen:

```
p {                               /* Selektor: „für alle HTML-Tags Typ <p> gilt:“ */
    color:blue;                  /* „Eigenschaft color auf Wert blue setzen“ */
    text-align:left;             /* „Eigenschaft text-align auf Wert left setzen“ */
}
```

- ✓ CSS-Dateien können kommentiert werden, wobei die Kommentare (der Text zwischen öffnendem „/*“ und schließendem „*/“) ignoriert werden

`p {color:blue;text-align:left;} /* Kommentar zum Absatzformat */`

Cascading Style Sheets (CSS)

- ✓ Eigenschaften der Darstellung bestimmter Klassen von HTML-Elementen können bestimmt werden, z.B. ALLE Absätze, ALLE Überschriften auf Ebene 1 usw.
- ✓ Aus Sicht des Web-Designs reicht der Typ-Selektor nicht, da evtl. nicht alle HTML-Elemente eines Typs in einer Web-Seite exakt gleich aussehen sollen
- ✓ Daher unterstützt CSS weitere Selektoren
 - Die **id eines HTML-Elements als Selektor** erlaubt es, für ein bestimmtes HTML-Element das Aussehen in einer CSS-Datei zu definieren

`<p id="kontakt"> Martina Mustermann ...</p>`

In der CSS-Datei wird die Darstellung dieses Absatzes definiert durch die Regel:

```
#kontakt
{
    text-align:center;
    color:blue;
}
```

- ✓ Für eine **Gruppe (Klasse → class) von Elementen** kann ebenfalls über den **Klassen-Selektor („class selector“)** deren Erscheinungsbild definiert werden

- Die Klasse zu der ein HTML-Element gehört, kann als Attributwert des HTML-Attributs `class` angegeben werden

```
<p class="paragraph_form1">Willkommen bei Max und Martina Mustermann!</p>  
<p class="paragraph_form1">Kontaktinformationen:</p>
```

In der CSS-Datei wird die Darstellung von Elementen dieser Klasse definiert durch die Regel:

```
.paragraph_form1 ← Klassen-Selektor  
{  
    text-align:center;  
    color:blue;  
}
```

- ✓ **Q1:**
Verwenden Sie den angegebenen Typ-Selektor (inline, intern und extern, siehe Folie 5) und testen Sie die Wirkung an einem HTML-Dokument mit mehreren Absätzen.
- ✓ Variieren Sie die Attribute (z. B. color, text-align) und Attributwerte (z. B. red, left) und verwenden Sie weitere (Liste z. B. auf <http://wiki.selfhtml.org> unter CSS/Eigenschaften).
- ✓ **Q2:**
Verwenden Sie den ID-Selektor, um drei einzelne Absätze <p>...</p> Ihres Dokuments unterschiedlich darzustellen (z. B. unterschiedliche Textfarbe).
- ✓ **Q3:**
Verwenden Sie den Klassen-Selektor, um Absätze Ihres Dokuments unterschiedlich darzustellen.

Definieren Sie dazu zwei Klassen mit jeweils unterschiedlichem Aussehen und ordnen Sie die Absätze im HTML-Dokument diesen Klassen zu.