STUDIEREN. WISSEN. MACHEN.

HOCHSCHULE DER MEDIEN

CSS

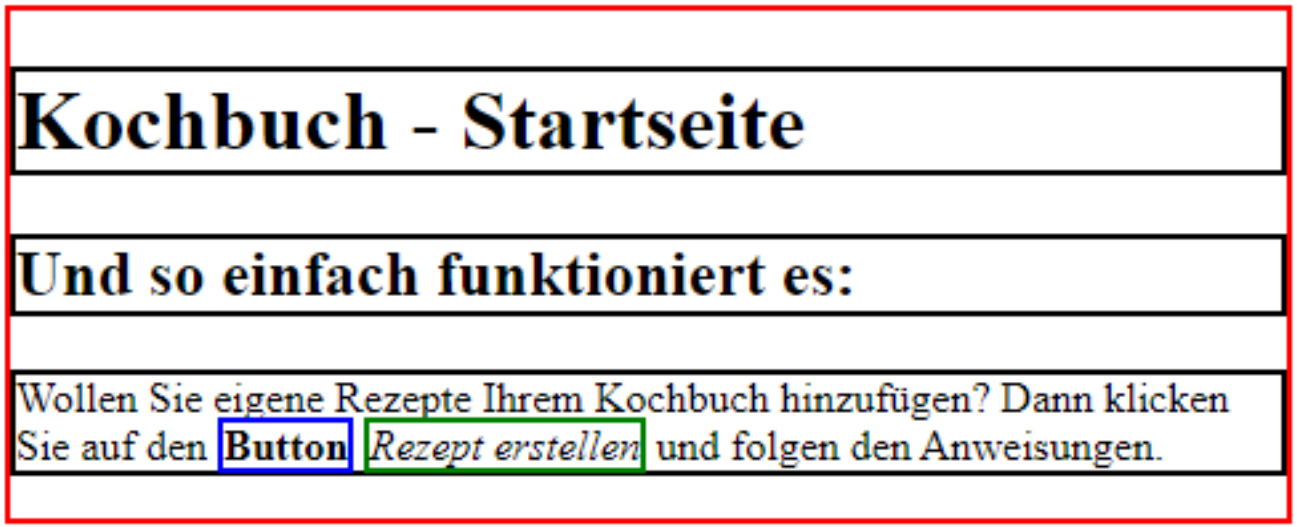
Vererbung und Konflikte

HTML - Hierarchie

› HTML-Dokumente sind hierarchisch aufgebaut.

```
<body>
    <h1>Kochbuch - Startseite</h1>
    <h2>Und so einfach funktioniert es:</h2>
    <p>Wollen Sie eigene Rezepte Ihrem Kochbuch hinzufügen?

        Dann klicken Sie auf den
       <strong>Button</strong>
       <em>Rezept erstellen</em>
      und folgen den
    </p>
</body>
```



HTML - Hierarchie

› HTML-Dokumente sind hierarchisch aufgebaut.

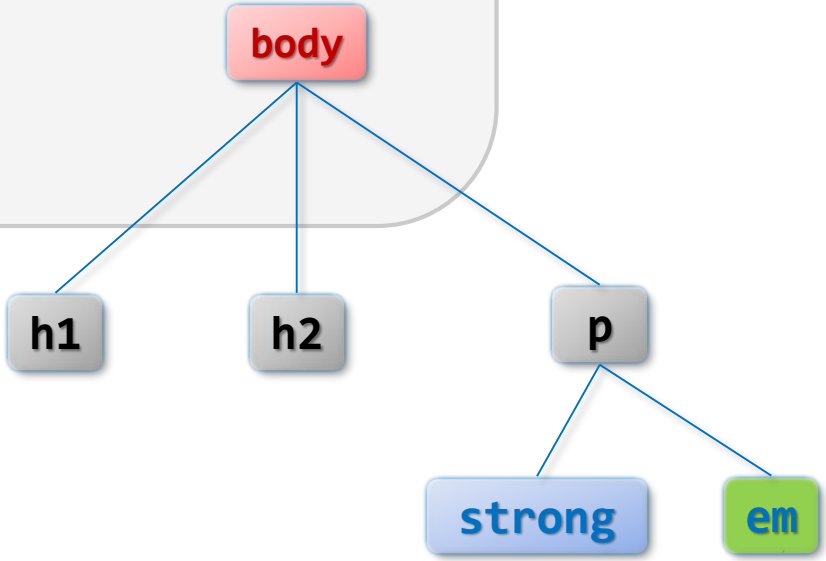
```
<body>
    <h1>Kochbuch - Startseite</h1>
    <h2>Und so einfach funktioniert es:</h2>
    <p>Wollen Sie eigene Rezepte Ihrem Kochbuch hinzufügen?

        Dann klicken Sie auf den
       <strong>Button</strong>
       <em>Rezept erstellen</em>
      und folgen den Anweisungen.
    </p>
```

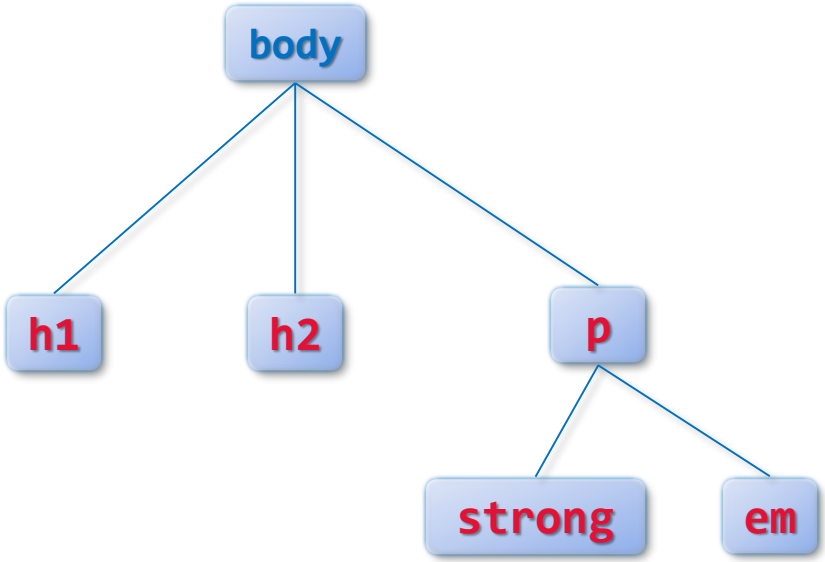
Kochbuch - Startseite

Und so einfach funktioniert es:

Wollen Sie eigene Rezepte Ihrem Kochbuch hinzufügen? Dann klicken Sie auf den **Button** *Rezept erstellen* und folgen den Anweisungen.

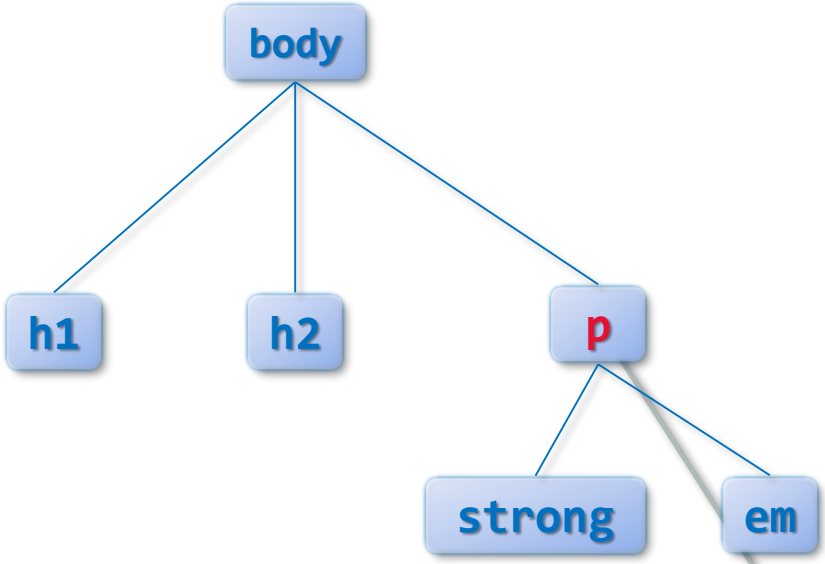


HTML – Hierarchie: Nachfahren



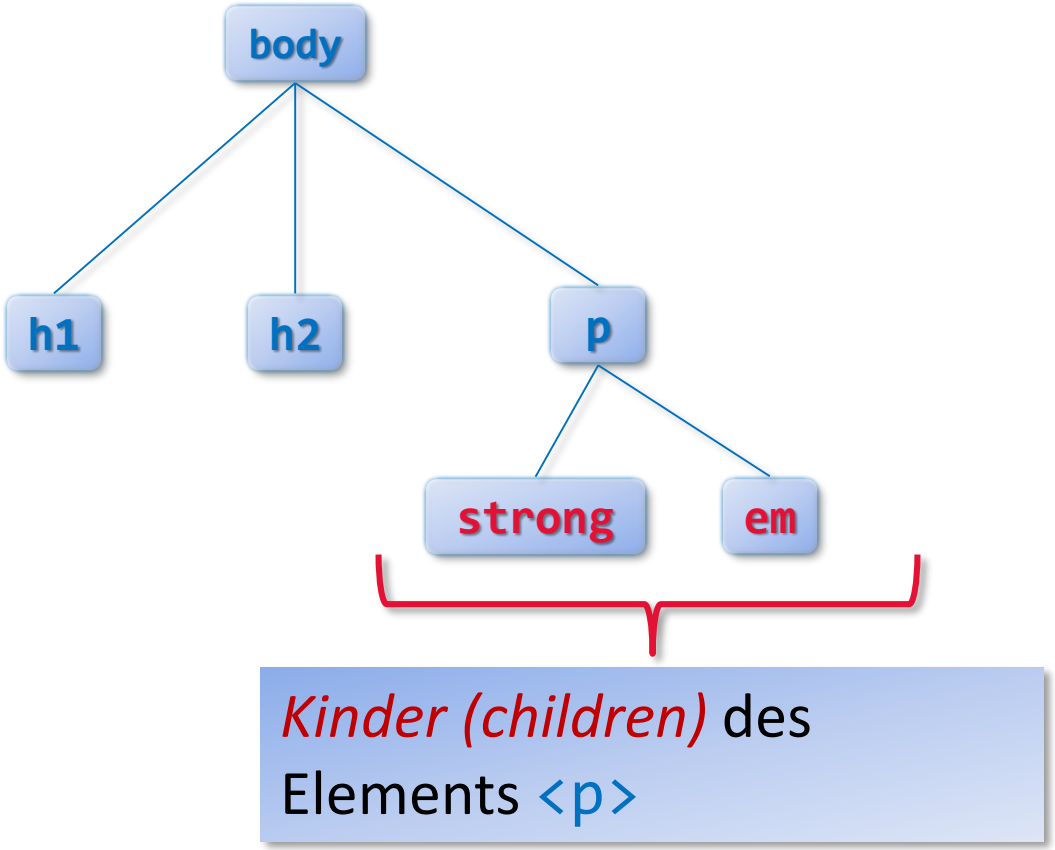
Alle markierten Elemente sind
Nachfahren (*descendants*) von <body>

HTML – Hierarchie: Eltern

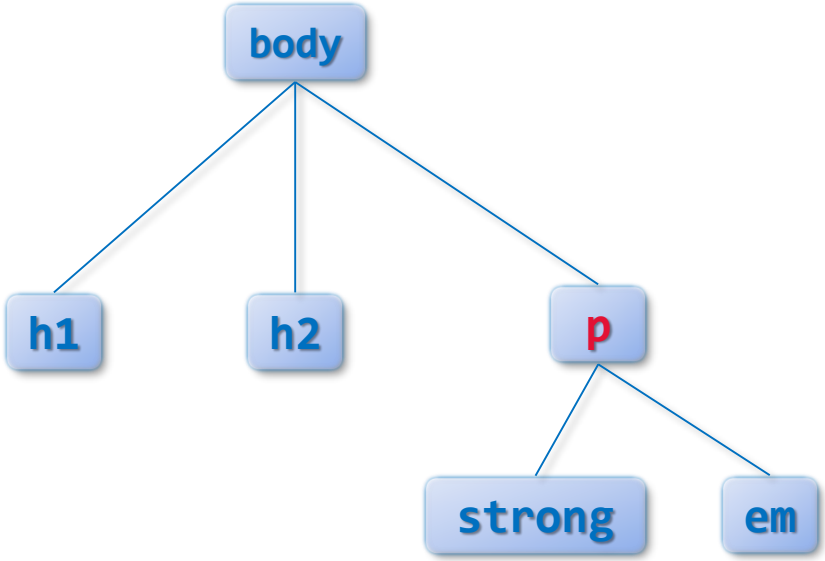


`<p>` ist *Elternteil* (*parent*) der Elemente `<strong>` und `<em>`

HTML – Hierarchie: Kinder



HTML – Hierarchie: Auswirkung von CSS-Regeln



Annahme: Eine CSS-Regel wurde für alle Absätze (p) definiert.
Wirkt sie sich auch auf weitere Elemente in dem Dokument aus?

CSS: Vererbung von Eigenschaften

HOCHSCHULE DER MEDIEN

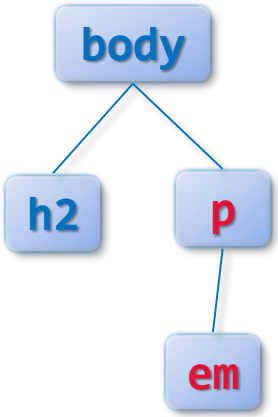
- *Eltern vererben* in CSS viele ihrer Eigenschaften automatisch auf die Kinder.

- Beispiel: alle Textformatierungseigenschaften (Schriftsatz, Textgröße, Zeilenabstand, …) werden von Eltern auf ihre Kinder vererbt.

- Nur Eigenschaften, bei denen die Vererbung keinen Sinn macht, werden nicht vererbt.

Vererbung von Eigenschaften: Beispiel mit CSS

```
...
    <p>Wollen Sie eigene Rezepte Ihrem Kochbuch hinzufügen?
       Dann klicken Sie auf den Button
       <em>Rezept erstellen</em>
      und folgen den Anweisungen.
    </p>
...
```



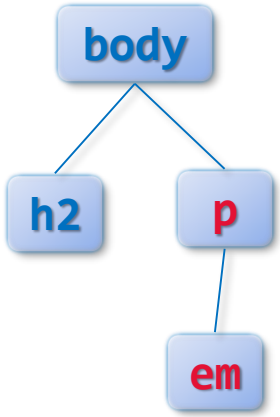
Gewollte Vererbung auf das ****-Element

```
p {color:blue}
```

[Interaktives Beispiel]

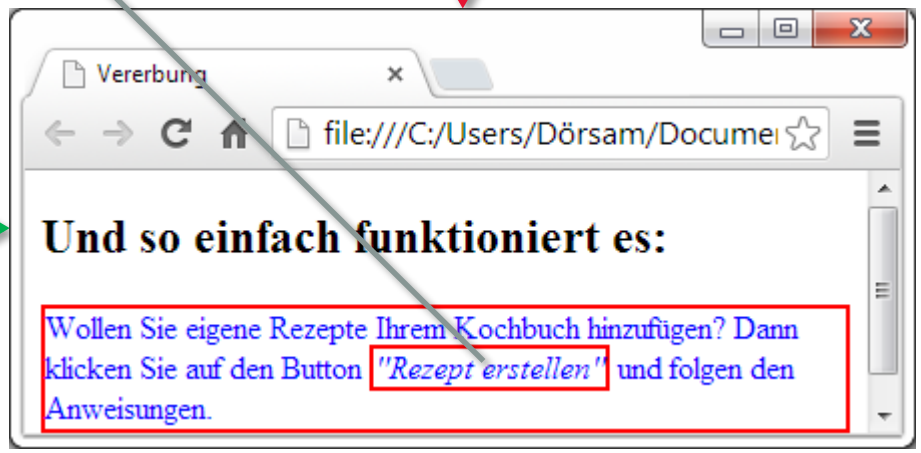
Vererbung von Eigenschaften: Beispiel mit CSS

```
...
    <p>Wollen Sie eigene Rezepte Ihrem Kochbuch hinzufügen?
       Dann klicken Sie auf den Button
       <em>Rezept erstellen</em>
      und folgen den Anweisungen.
    </p>
...
```



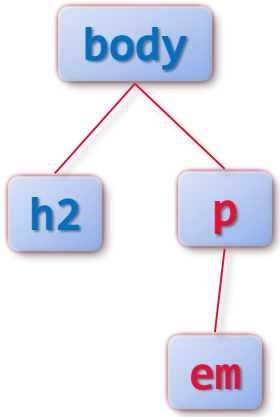
Beispiel, wie Vererbung NICHT funktionieren sollte!

```
p {color:blue;
border: 2px solid red}
```



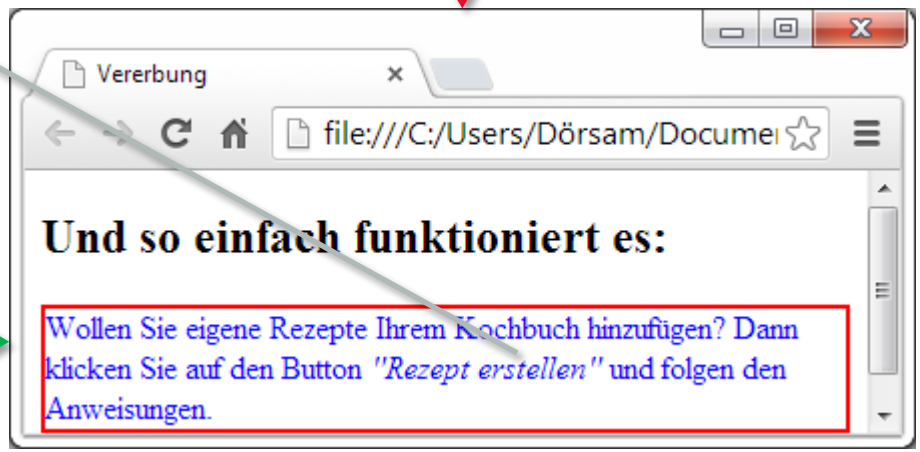
Vererbung von Eigenschaften: Beispiel mit CSS

```
...
    <p>Wollen Sie eigene Rezepte Ihrem Kochbuch hinzufügen?
       Dann klicken Sie auf den Button
       <em>Rezept erstellen</em>
      und folgen den Anweisungen.
    </p>
...
```



Das ****-Element erbt nur relevante Eigenschaften.

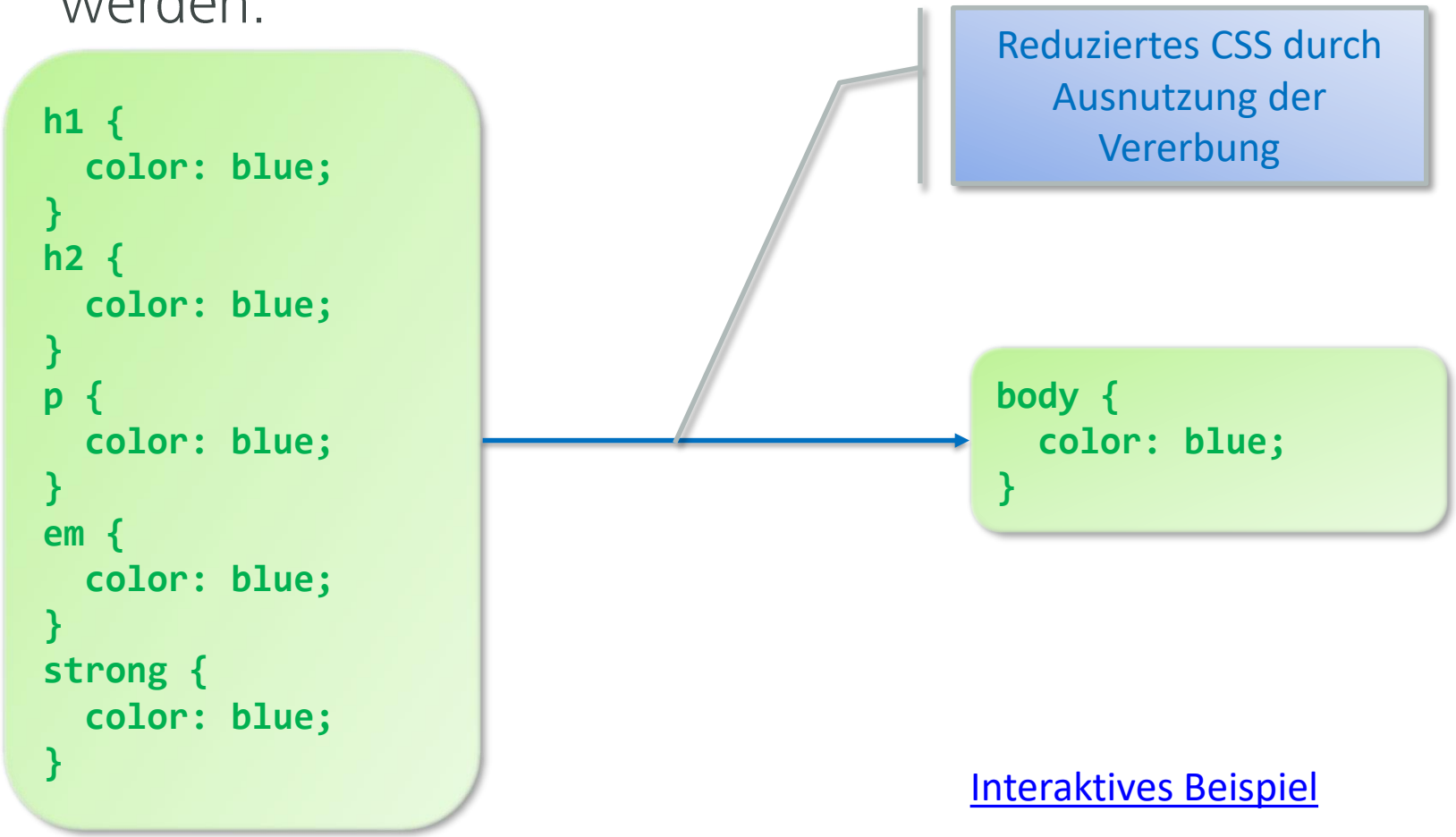
```
p {color:blue;
border: 2px solid red}
```



Interaktives Beispiel

Warum überhaupt Vererbung?

› Durch Vererbung können CSS-Dateien sehr klein gehalten werden.



Interaktives Beispiel

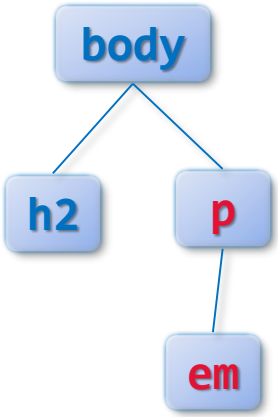
CSS: Überschreiben von Eigenschaften

HOCHSCHULE DER MEDIEN

- *Kinder* können auch Eigenschaften ihrer Eltern *überschreiben*.

Überschreiben von Eigenschaften: Beispiel mit CSS

```
...
    <p>Wollen Sie eigene Rezepte Ihrem Kochbuch hinzufügen?
       Dann klicken Sie auf den Button
       <em>Rezept erstellen</em>
      und folgen den Anweisungen.
    </p>
...
```



Überschreiben der Farbeigenschaft durch das -Element

```
p {color:blue;}
em {color:red}
```

Interaktives Beispiel

Konflikte innerhalb einer CSS-Datei

HOCHSCHULE DER MEDIEN

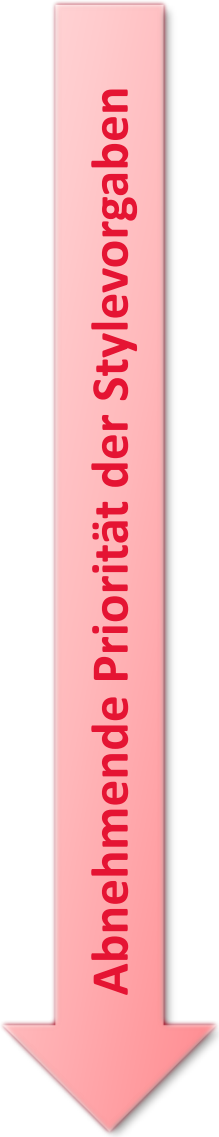
- Konflikte durch mehrfache Selektoren:
  - Durch die Benutzung mehrerer Selektoren können Konflikte entstehen, wenn in einer Datei mehrere widersprüchliche Eigenschaften für ein Element definiert werden.
  - Bei solchen Konflikten gewinnt
    - der spezifischere Selektor oder
    - bei gleichrangigen Selektoren die letzte Regel in der CSS-Datei.
- Einfache Selektoren:
  - Auch innerhalb einer Regel kann es zu Konflikten kommen.
  - Bei Konflikten innerhalb einer Regel gewinnt immer die letzte Definition.

Reihenfolge für die Spezifizität der Selektoren

HOCHSCHULE DER MEDIEN

- Was heißt genau „ der spezifischere Selektor“?

- Rangfolge der Spezifizität:
  1. `style`-Attribut in einem HTML-Element:
     ```
     <h2 style = „color: red“>
     ```
  2. Eine Regel wurde mit einem ID-Selektor definiert:
     ```
     #id01 {color:red;}
     ```
  3. Eine Regel wurde mit einem Klassen-Selektor definiert
     ```
     .blau {color:blue;}
     ```
  4. Eine Regel wurde mit einem Element-Selektor (einzeln oder als Gruppe) definiert:
     ```
     h2 {color:red;}
     h2 {color:blue; color: green}
     ```



Konflikte bei Gruppenselektoren

```
...
    <h2 id=„id01“>Und so einfach funktioniert es:</h2>
    <p>Wollen Sie eigene Rezepte Ihrem Kochbuch hinzufügen?
        Dann klicken Sie auf den Button
       <em>Rezept erstellen</em>
      und folgen den Anweisungen.
    </p>
...
```

Der *spezifischere Selektor* bestimmt die Regel.

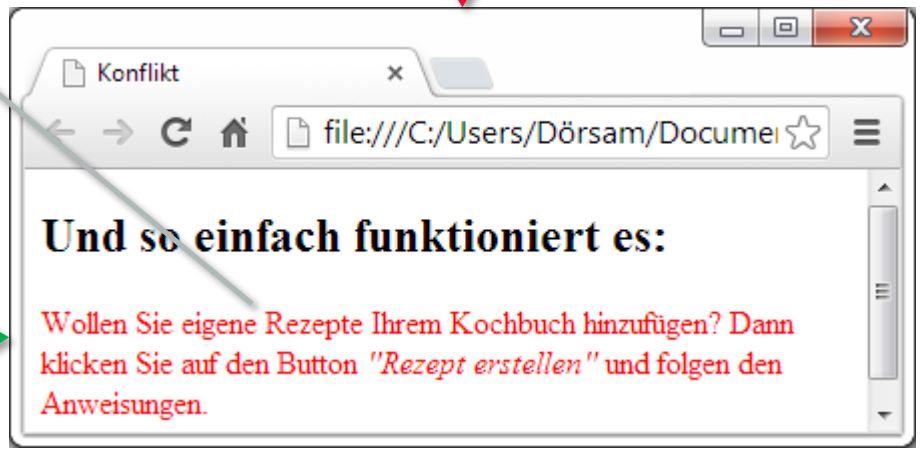
```
#id01 {color:red;}
h1, h2, p {color:blue;}
```

Konflikte innerhalb einer Regel

```
...
    <h2>Und so einfach funktioniert es:</h2>
    <p>Wollen Sie eigene Rezepte Ihrem Kochbuch hinzufügen?
        Dann klicken Sie auf den Button
       <em>Rezept erstellen</em>
      und folgen den Anweisungen.
    </p>
...
```

Die *letzte Deklaration* bestimmt die Regel.

```
p {color:blue;
   color:green;
   color:red;}
```



Konflikte zwischen Stylevorgaben

HOCHSCHULE DER MEDIEN

- CSS-Regeln können in unterschiedlichen Dateien definiert werden (inline, eingebettet, extern).
- Dadurch kann es leicht passieren, dass für ein Element mehrere, widersprüchliche Regeln definiert wurden.
- Deswegen werden Stylesheets *kaskadierend* ausgewertet: es gibt eine vordefinierte Reihenfolge, in der die Regeln ausgewertet werden.

CSS: Hierarchie der Stylesheets (Auszug)

HOCHSCHULE DER MEDIEN

1. Standardvorgaben des Browsers
2. Mit `<link>` importierte externe Stylevorgaben
3. …
4. Inline-Stylevorgabe