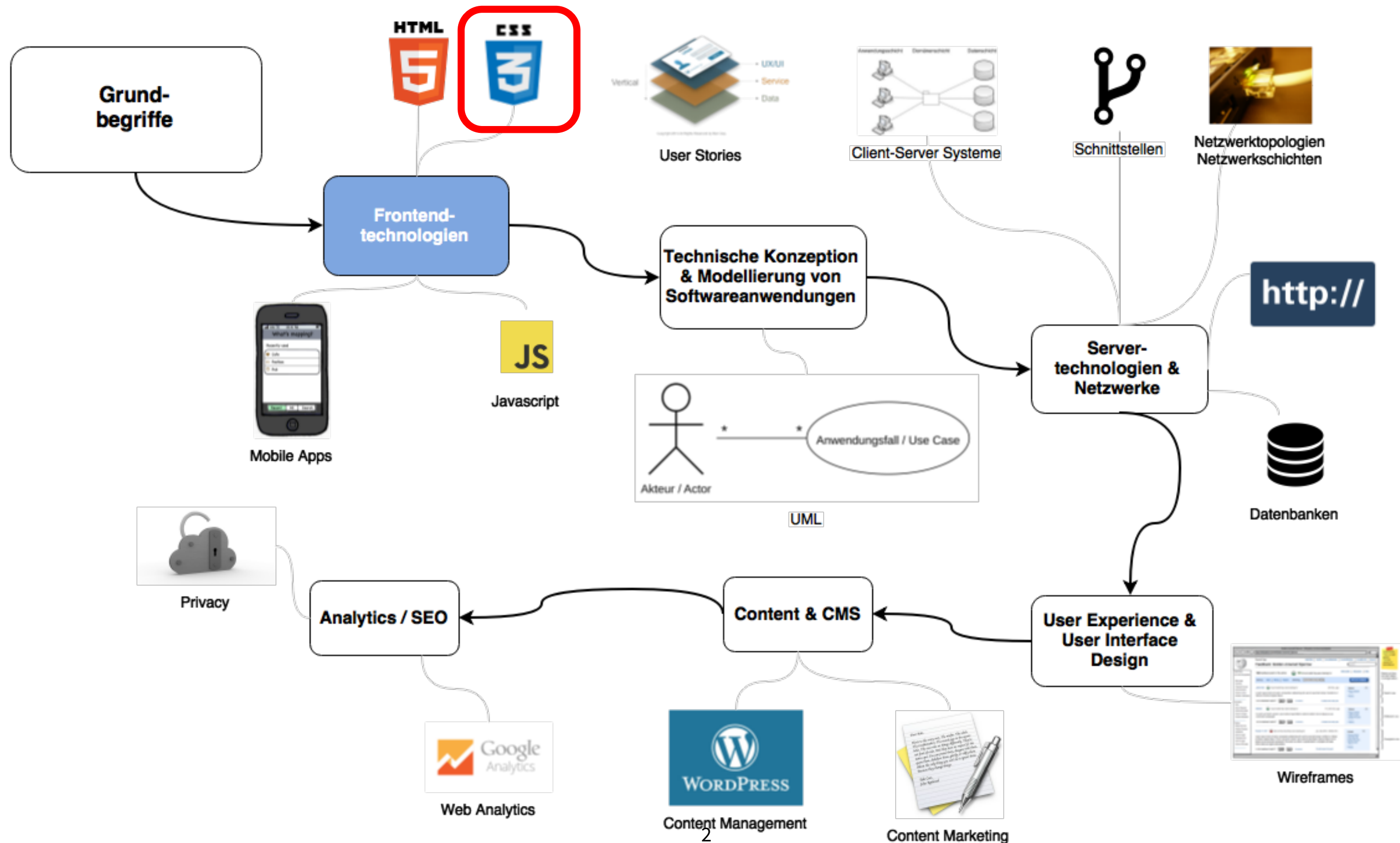


Frontendtechnologien

CSS 3

Einführung in Softwaretechnologien

Überblick



Titel dieses HTML-Dokuments

×

+

←

example.org

🔍

Suchen

⬇

⚙

⌵

⚙

🐞

⌵

📧

⌵

ℹ

Se

k


☰

Thema des Dokuments

Lorem ipsum dolor sit amet, consectetur adipiscing elit. In eget interdum neque. Donec lobortis, mauris volutpat *rutrum mattis, nisi sem gravida eros*, ut sagittis ligula urna a dui.

Ein erwähnenswerter Abschnitt

Ein Artikel



This is a really beautiful butterfly
Courtesy of [photos.com](#)

Quisque rutrum pulvinar orci, sed fermentum arcu semper ut. Donec placerat pretium ipsum porta sollicitudin. Sed eu ex enim. Ut sed feugiat metus. Proin posuere **justo nec mauris** elementum.

Ebenfalls wichtig:

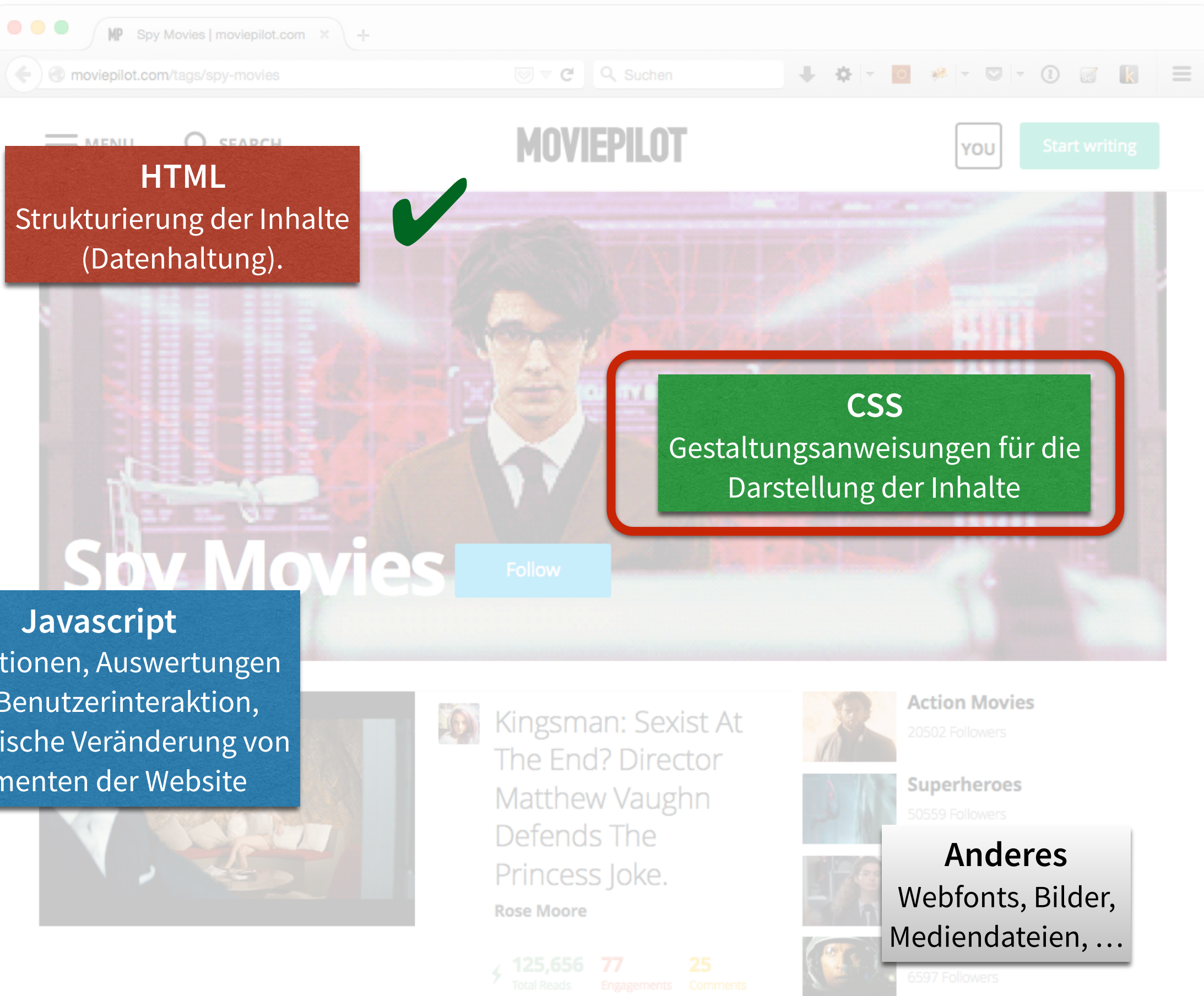
Phasellus blandit pulvinar turpis, non blandit justo suscipit eget. Suspendisse enim turpis, feugiat eget placerat eget, ultricies nec tortor. Duis venenatis elit eu vulputate molestie.

Das Wetter von Berlin und Hamburg.

| Stadt | Wetter |
|------------------------------------|------------------------------------|
| Berlin | Sonne, 22°C |
| Hamburg | Regen, 16°C |
| Durchschnittstemperatur: 11,5°C | Durchschnittstemperatur: 10,8°C |

Und immer schön Listen benutzen

- Donec
- [id quam quis](#)
- **tortor**
 - [molestie](#)
 - [volutpat](#)
 - [quis](#)



HTML

Strukturierung der Inhalte
(Datenhaltung).



CSS

Gestaltungsanweisungen für die
Darstellung der Inhalte

Javascript

Animationen, Auswertungen
von Benutzerinteraktion,
dynamische Veränderung von
Elementen der Website

Anderes
Webfonts, Bilder,
Mediendateien, ...

CSS

Definition

- „**Cascading Style Sheets (CSS)** ist ein einfacher Weg, um **Gestaltungsanweisungen** (z.B. Schriftart, Farben oder Abstände) **für Webseiten** zu definieren.“

(<http://www.w3.org/Style/CSS/>)

- **Gestaltungsanweisungen** (Regeln) bestehen aus einem **Selektor** und einem **Deklarationsblock**

```
h1 {  
    color: red;  
    background-color: #FDE8D2;  
}
```

CSS

Grundaufbau

- Mit dem **Selektor** werden ein oder mehrere HTML-Elemente ausgewählt
- In **Deklarationsblöcken** werden für diese Elemente dann Gestaltungsanweisungen in Form von Deklarationen hinterlegt
- Eine **Deklaration** besteht immer aus einer **CSS-Eigenschaft** (Property), einem **Doppelpunkt**, dem **Wert** der Eigenschaft und einem **Semikolon**

```
...  
background-color: #FDE8D2;  
...
```


CSS

Selektoren

- Das **Ziel eines Selektors** ist es, ein oder mehrere Elemente eines HTML-Dokuments **auszuwählen**, damit ihm **Gestaltungsregeln** zugewiesen werden können
- Es gibt **4 verschiedene Arten** ein Element mit CSS zu selektieren
 - Tag-Selektoren
 - ID-Selektoren
 - Class-Selektoren
 - Pseudo-Class Selektoren

CSS

Tag Selektoren

- Ein **Tag Selektor** adressiert Elemente an Hand von **HTML-Tags** eines bestimmten Typs
- Im folgenden Beispiel werden *alle Überschriften 1. Grades* selektiert:

```
h1 {  
    ...  
}
```


CSS

ID-Selektoren

- Ein ID-Selektor bezieht sich auf das **Attribut `id`**, dass jedem Element eines **HTML-Dokuments** zugewiesen werden kann
- Da jeder **Identifikator** in einem Dokument **eindeutig** sein muss, lassen sich so sehr gut spezifische Element beschreiben

```
<div id="books">  
  ...  
</div>
```

```
#books {  
  ...  
}
```

CSS

Class-Selektoren

- Ein Class-Selektor bezieht sich auf das **Attribut class**, dass jedem Element eines **HTML-Dokuments** zugewiesen werden kann
- In einem HTML-Dokument können beliebig **viele Elemente** die **gleiche Klasse** erhalten. Class Attribute eignen sich daher vor allem zur **Selektion gleichartiger Elemente**

```
<div id="books">  
  <div class="book">...</div>  
  <div class="book">...</div>  
  ...  
</div>
```

```
.book {  
  ...  
}
```

CSS

Pseudo-Class Selektoren

- Pseudoklassen sind **Schlüsselwörter**, die an **Selektoren** angehängt werden, um den **Zustand** eines Elements auszuwählen

```
#selektor:pseudo-class {  
    ...  
}
```

- Beispiele für **auswählbare Zustände**:
 - **a:visited**: ein besuchter Link
 - **input:checked**: eine angeklickte Checkbox in einem Formular
 - **li:first**: das erste, letzte oder n-te Element einer Liste!

CSS

Kaskadierung von Selektoren

```
<div id="good-books">
  <div class="book">...</div>
  <div class="book">...</div>
  ...
</div>
```

```
<div id="bad-books">
  <div class="book">...</div>
  <div class="book">...</div>
  ...
</div>
```

```
.book {
  font-size: 12px;
}
```

```
#good-books .book {
  font-size: 25px;
}
```

Aufgabe:

Gute Bücher sollen einen größeren Titel haben!

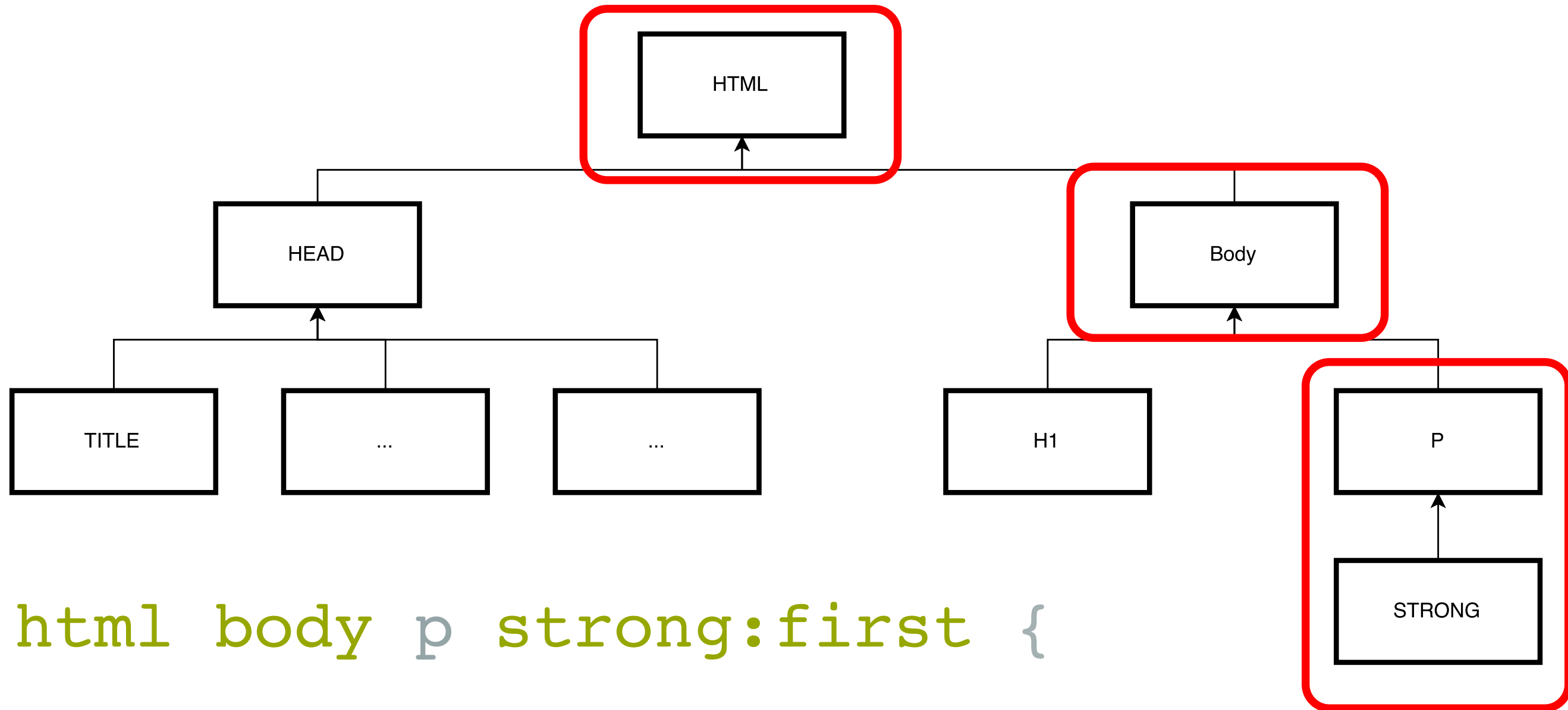
CSS – Selektoren

Vererbung und Gewichtung

- CSS Selektoren können **kombiniert** und **verschachtelt** und **verkettet** werden! Auf diese Weise können gezielt bestimmte Elemente ausgewählt werden
- Man beschreibt den **Pfad durch das HTML-Dokument**, um zum gewünschten Element zu gelangen.
- Je **spezifischer ein Pfad**, desto **höher die Priorität** der Regel

CSS

Elemente selektieren



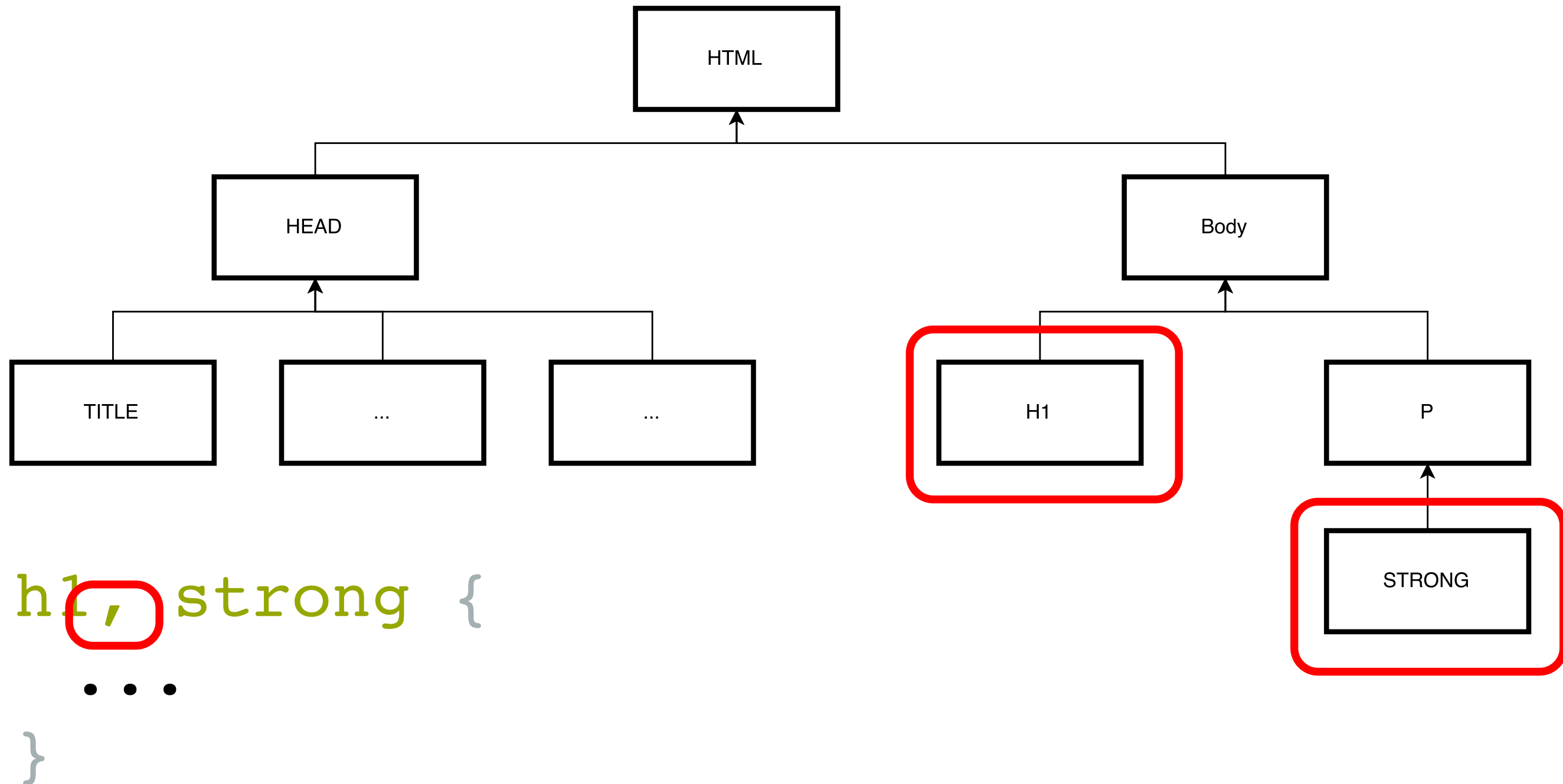
```
html body p strong:first {
```

```
...
```

```
}
```

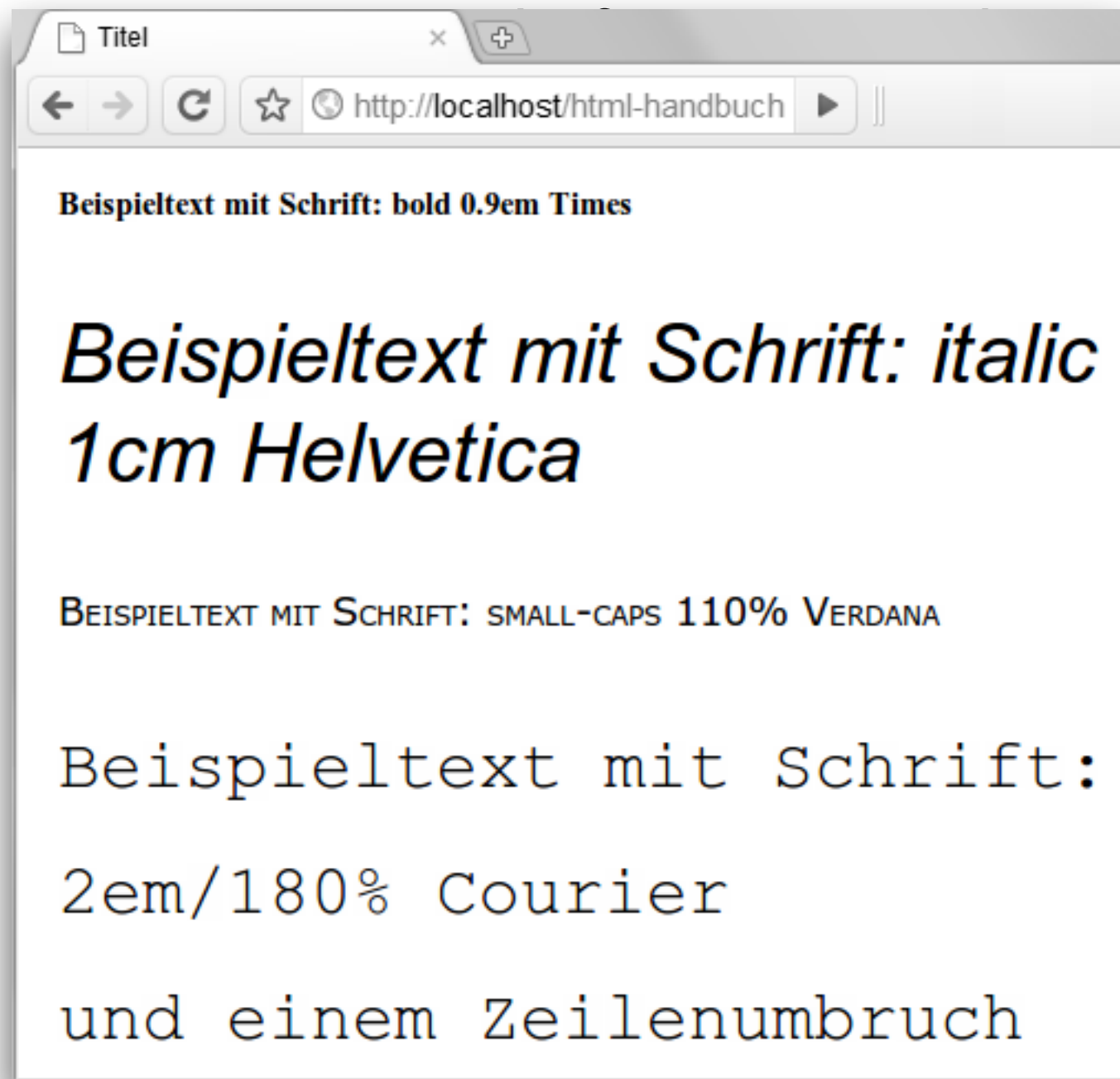
CSS

Elemente selektieren



CSS

Schriftformatierung



```
#Text01 {  
    font-weight: bold;  
    font-size: .9em;  
    font-family: Times;  
}  
  
#Text02 {  
    font-style: italic;  
    font-family: Helvetica;  
}  
  
#Text03 {  
    font: small-caps 110%  
    Verdana;  
}  
  
#Text04 {  
    font: 2em/180% Courier;  
}
```

CSS

Maßangaben

- Es gibt **absolute und relative** Maßangaben
 - **Absolut:** 12px (Pixel), 8in (Inch), 3cm,...
 - **Relativ:** 80%, 0.8em (Verhältnis zur Schriftgröße)
- Heute sind **Angaben in Pixel** zur Regel geworden, da moderne Browser eine Anpassung der Darstellung erlauben („Browser-Zoom“)

CSS

Farbangaben

- CSS erlaubt die Spezifikation
 - der **Hintergrundfarbe**,
 - der **Rahmenfarbe** und
 - der **Schriftfarbe** jedes Elements
- **Farbwerte** werden entweder
 - mit einem **vordefinierten Namen** (red, green, grey)
 - mit mit einem **RGB-Wertetripel**

```
div.article {  
    background-color: red;  
    border-color: rgb(0,0,0);  
    color: #20A020;  
}
```

beschrieben

CSS

Farbwerte

- Die Bezeichnung **RGB** steht für **Rot, Grün, Blau**
- RGB ist ein **additives Farbmodell**: aus diesen drei Farben lassen sich alle Farben mischen (alle Farben ergeben Weiss)

CSS

Farbangaben

- Farbangabe mit dem **RGB-Format**:

`rgb(12, 0, 200)`

- **Hexadezimale Farbangabe**:

`#0C00C8`

- Jede hexadezimale Ziffer kann **16 Zustände** haben (0-9, A-F)
- **Pro Farbwert (R,G,B)** gibt es **2 Ziffern**, also insgesamt **256³ unterschiedliche Farben**, was das volle RGB-Farbschema abdeckt

CSS

Einbindung in HTML

- Es gibt drei Arten, HTML-Inhalte mit CSS zu versehen:
 - Style-Definitionen in separaten CSS-Dateien
 - im Kopfbereich eines HTML-Dokuments, oder
 - direkt in einzelnen HTML-Elementen (inline)
- Diese Varianten können kombiniert werden

CSS

Einbindung in separaten Dateien

index.html:

```
<!DOCTYPE html>
<html>
<head>
  <title>Titel dieses HTML-Dokuments</title>
  <link rel="stylesheet" href="css/styles.css" />
  ...
  <link rel="stylesheet" href="css/styles2.css" />
```

css/styles.css:

```
h1 {
  font-weight: bold;
}

article p {
  color: black;
}
```


CSS

Einbindung im Kopfbereich

index.html:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8" />
  <title>Titel dieses HTML-Dokuments</title>

  <style type="text/css">
    body {
      color:red;
    }
  </style>

  ...
</head>
```

CSS

Einbindung in einzelnen Elementen

```
<div id="good-books" style="font-size:24px">
  <div class="book">...</div>
  <div class="book" style="font-size:10px;color:red;">.
div>
  ...
</div>
```

Über das HTML-Attribut **style** können CSS-Eigenschaften
direkt an einem HTML-Element notiert werden

CSS

Gewichtung

index.html:

```
...  
<head>  
1. <link rel="stylesheet"  
    href="css/styles.css" />  
  
2. <style>  
    #good-book {  
        color:red;  
    }  
    </style>  
</head>  
<body>  
3. <div id="good-book"  
    style="color:purple">  
    Welche Farbe hat das Buch?  
</div>
```

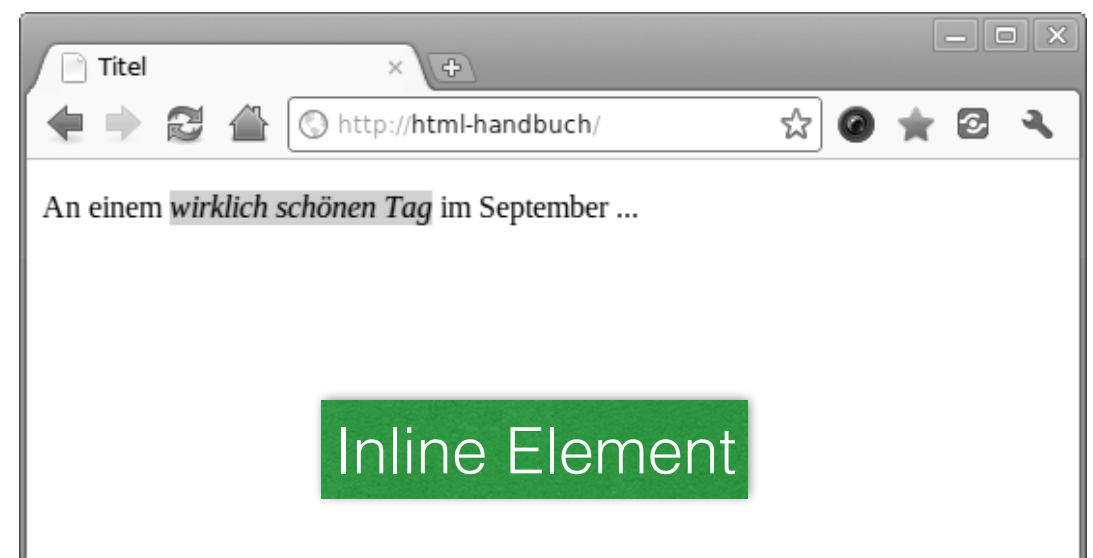
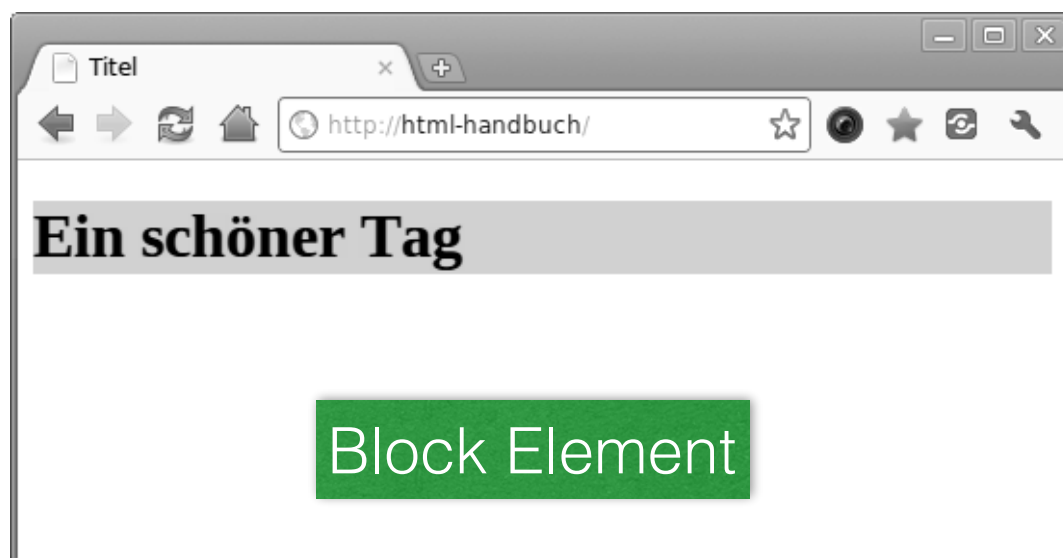
css/styles.css:

```
#good-book {  
    color: black;  
}
```

CSS

Boxmodell

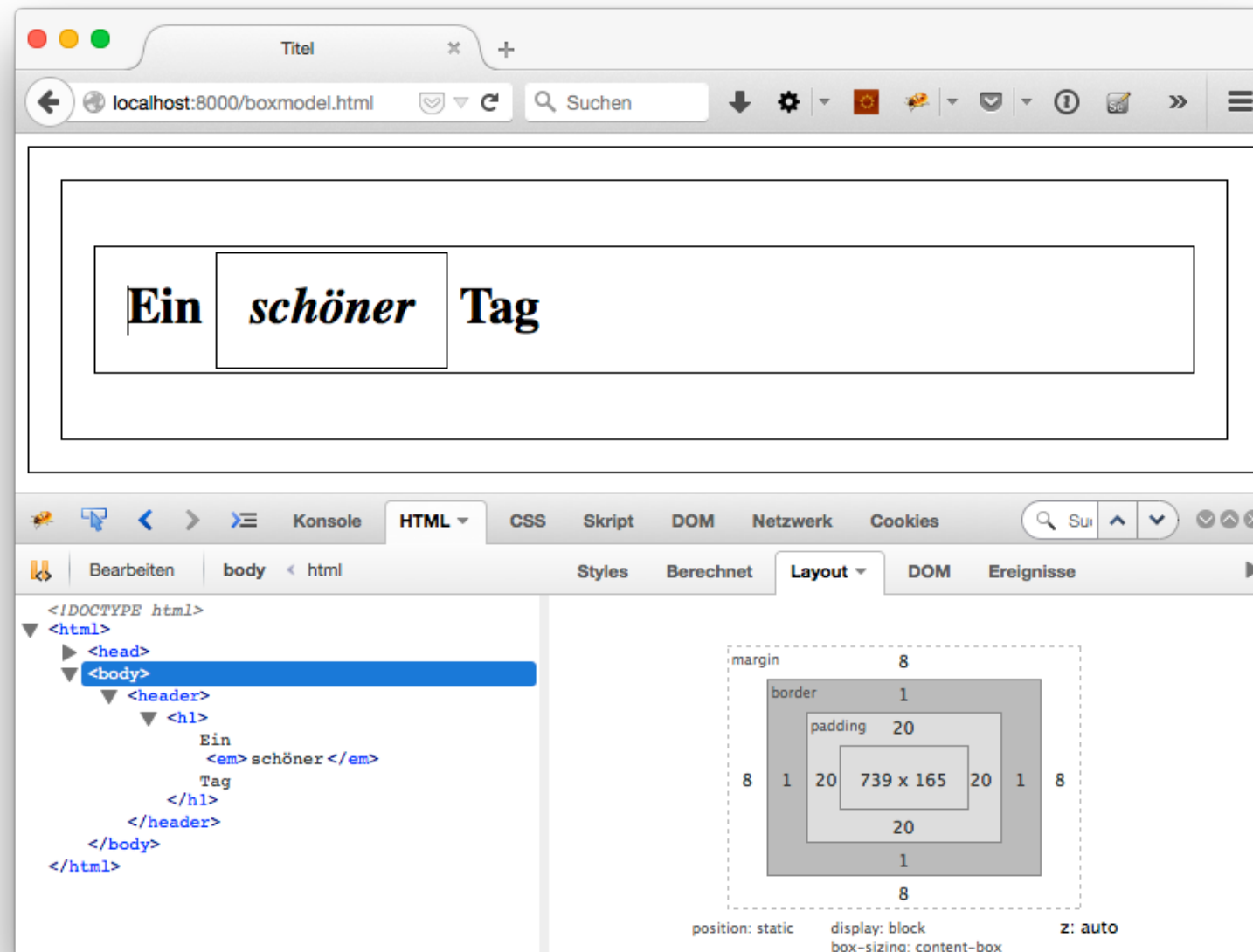
- CSS unterscheidet zwischen **Block-Elementen** und **Inline-Elementen**
 - Block-Elemente: nehmen **in der Breite so viel Raum ein wie möglich** und in der Höhe so viel wie erforderlich
 - Inline-Elemente nehmen in der Breite und in der Höhe **nur so viel Raum ein wie erforderlich**



CSS

Boxmodell

- Das Elternelement wird als **Container** bezeichnet und bestimmt den zur Verfügung stehenden Raum eines Elements



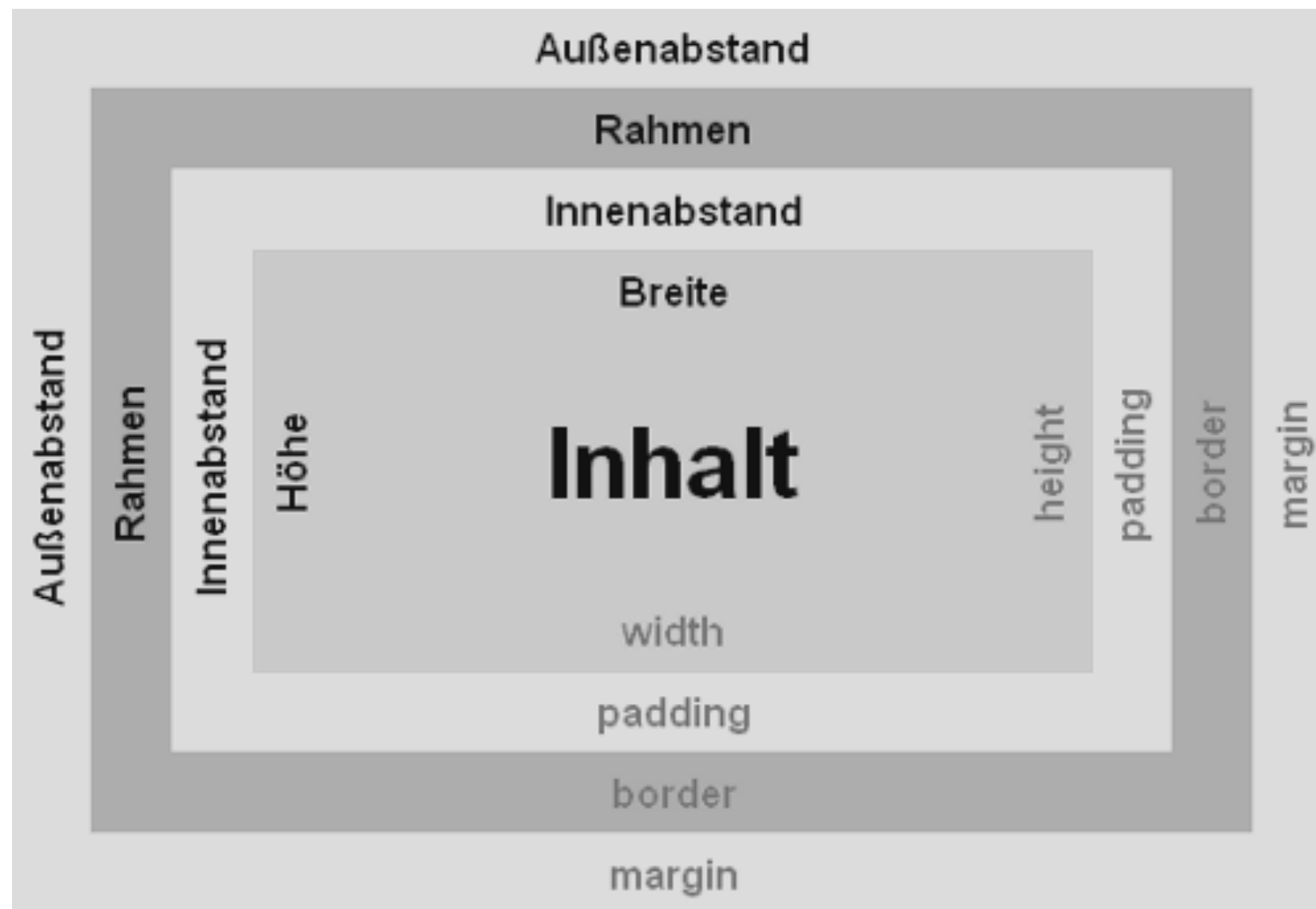
CSS

Ausmaß von Elementen

- Die Maße eines HTML-Elements bestehen immer aus:
 - der **Breite/Höhe** des Elementinhalts (`width/height`),
 - des **Innenabstands** (`padding`),
 - der **Rahmenstärke** (`border-width`) und
 - des **Außenabstands** (`margin`).

CSS

Ausmaß von Elementen



<http://webkompetenz.wikidot.com/html-handbuch:boxmodell>

CSS

Limitierungen

- Große Projekte haben üblicherweise mehrere tausend Zeilen CSS-Code.
- Bei der Verwaltung solcher Mengen stößt man mit CSS an Grenzen:
 - **Redundanz** bei ähnlichen (aber nicht gleichen) Regeln
 - Keine **Berechnungen** möglich
 - Keine Möglichkeit **Werte zentral zu pflegen** (z.B. die Hausfarbe)



Scalable and Modular Architecture for CSS

A flexible guide to developing sites small and large.



"SMACSS is becoming one of the most useful contributions to front-end discussions in years" *

I've been analyzing my process (and the process of those around me) and figuring out how best to structure code for projects on a larger scale. What I've found is a process that works equally well for sites small and large.

Learn how to structure your CSS to allow for flexibility and maintainability as your project and your team grows.

Get the book!

Get the workshop!

CSS

Preprozessoren

- CSS Preprozessoren sind Werkzeuge, mit denen der **Funktionsumfang** und die **Syntax von CSS** erweitert werden
- Der Code, der für einen solchen Preprozessor geschrieben wurde, **muss nach CSS übersetzt** (kompiliert) werden
- Bekannte Vertreter sind: LESS, SASS, Stylus

CSS

Stylus Preprozessor

main.styl

```
fonts = helvetica, arial, sans-serif
```

```
body
```

```
  padding 50px
```

```
  font 14px/1.4 fonts
```

main.css

```
body {
```

```
  padding 50px;
```

```
  font 10px helvetica, arial, sans-serif;
```

```
}
```

CSS

Stylus Preprozessor

main.styl

```
border-radius ( )  
  -webkit-border-radius arguments  
  -moz-border-radius arguments  
  border-radius arguments  
  
a.button  
  border-radius 5px
```

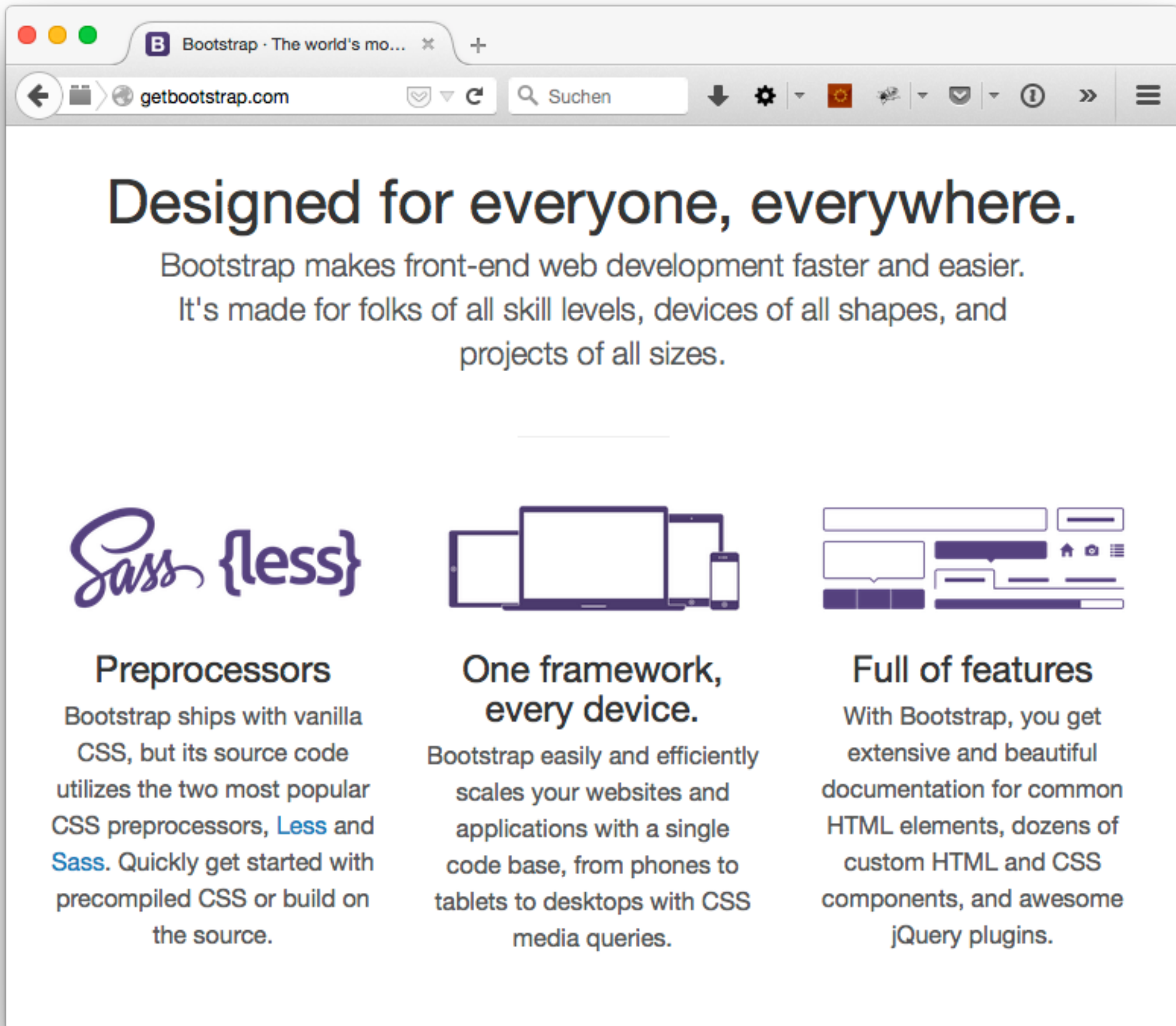
main.css

```
a.button {  
  -webkit-border-radius 5px;  
  -moz-border-radius 5px;  
  border-radius 5px;  
}
```

CSS

Frameworks

- Bei der Bestrebung die Entwicklung von Websites zu vereinfachen, haben sich Webentwickler zusammengeschlossen und **abstrahierte bzw. vereinheitlichte Vorgehensmodelle** entwickelt
- Diese Modelle stehen in Form so genannter **Frameworks** zur Verfügung
- Ein bekanntes und verbreitetes Framework für Websites ist **Bootstrap**



<http://getbootstrap.com>

CSS

Bootstrap

- Bootstrap stellt u.a. ein **Rastersystem** für die **Platzierung und Ausrichtung** von HTML-Elementen bereit
- Dafür reicht es, den Elementen vordefinierte **Klassenbezeichnungen** zu geben

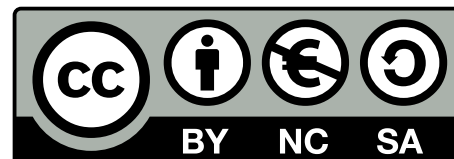
```
<!-- Columns are always 50% wide -->  
<div class="row">  
  <div class="col-xs-6">.col-xs-6</div>  
  <div class="col-xs-6">.col-xs-6</div>  
</div>
```

CSS

Bootstrap

- Es gibt eine ganze Reihe ähnlicher Frameworks, die mit verschiedenen Ansätzen für unterschiedliche Einsatzfälle optimiert wurden
- Es ist immer empfehlenswert, zu prüfen, ob das Framework zum Einsatzzweck passt und ob es von einer Community aktiv weiterentwickelt wird

Vielen Dank.



Dieses Werk ist lizenziert unter einer Creative Commons 4.0 International Lizenz mit folgenden Eigenschaften:

- Namensnennung
- Nicht-kommerzielle Nutzung
- Weitergabe unter gleichen Bedingungen.

Ressourcen

- <http://blog.millermedeiros.com/solid-css/>
- <http://smacss.com/>
- <https://developers.google.com/web/tools/starter-kit/>
- <http://www.sitepoint.com/5-most-popular-frontend-frameworks-compared/>