



Webtechnologien | 2015

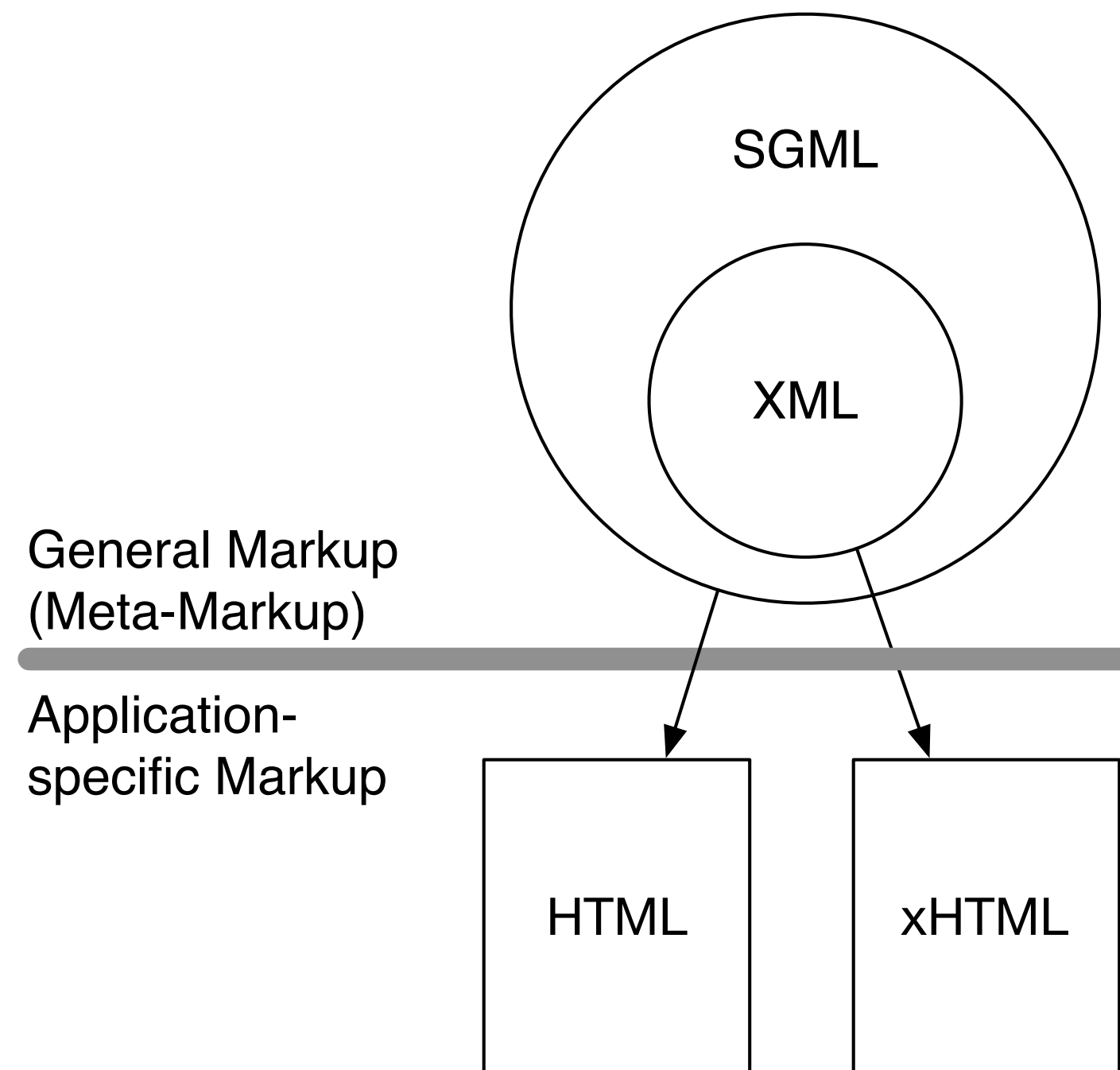
Kapitel 2:HTML & CSS

Markup-Sprachen | HTML Syntax | HTML Head | HTML Body | CSS-Grundlagen | CSS-Eigenschaften | CSS-Elementfluss | Selektoren | Ergänzung und Vererbung

Axel Küpper | Fachgebiet *Service-centric Networking* | TU Berlin & Telekom Innovation Laboratories

2.1 Markup-Sprachen

Was sind Markup-Sprachen?



Markup-Sprachen (Auszeichnungssprachen)

- Maschinenlesbare Sprachen für die Gliederung und Formatierung von Texten und anderen Daten
- Beschreibung von Eigenschaften, Zugehörigkeiten (Semantik) und Darstellungsformen eines Textes oder einer Datenmenge

Standardized General Markup Language (SGML)

- Metasprache mit deren Hilfe man verschiedene Auszeichnungssprachen für Dokumente definieren kann
- Ein SGML-Dokument basiert auf einer *Dokumenttypdefinition* (DTD), die den strukturellen Aufbau des Dokuments beschreibt

Extensible Markup Language (XML)

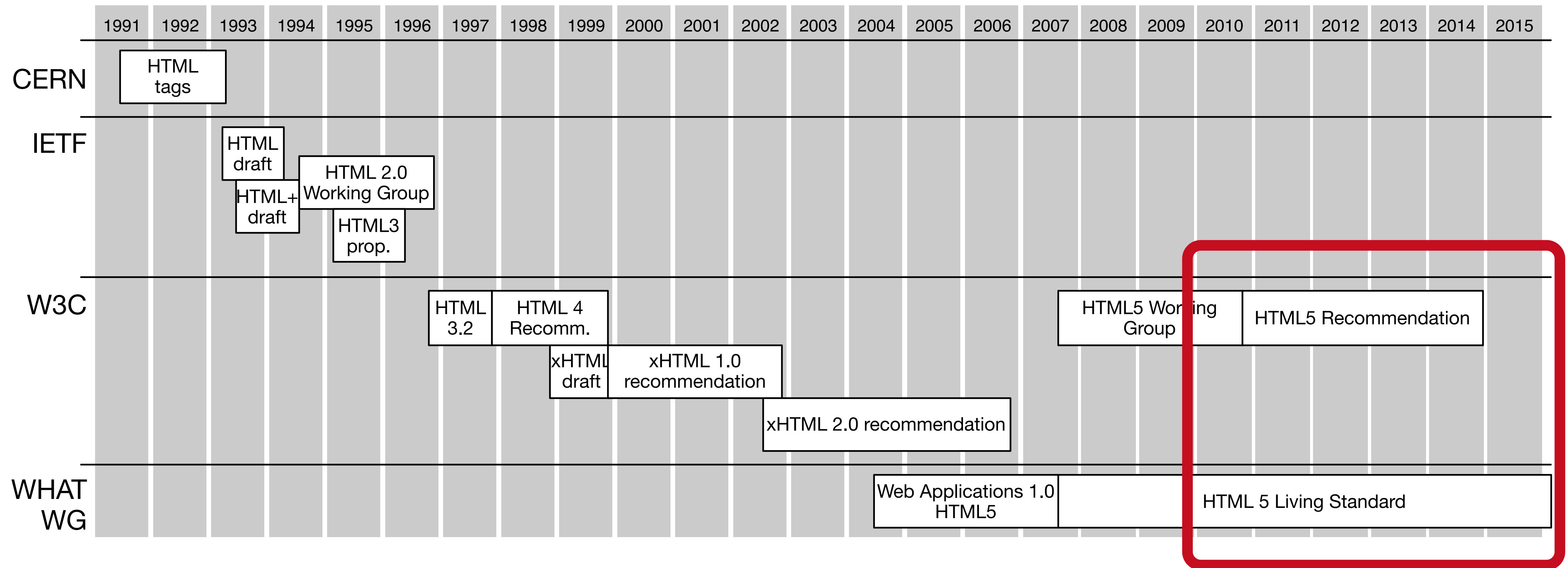
- Auszeichnungssprache die eine Menge von Regeln für die Kodierung von Dokumenten in ein Format enthält welches sowohl maschinen- als auch menschenlesbar ist
- Abgeleitet von SGML (weniger komplex, Standard umfasst 26 Seiten, anstelle von 500 bei SGML)
- Definiert durch das World Wide Web Consortium (W3C) in der XML 1.0 Specification
- Basis für viele domänenspezifische Dokumentenformate (XHTML, RSS, Open Office XML...) sowie Kommunikationsprotokolle (XMPP, SOAP)

HyperText Markup Language (HTML)

- Auszeichnungssprache für Webseiten
- Enthält Annotationen für Überschriften, Absätze, Listen, Verweise, Zitate und andere Strukturelemente

2.1 Markup-Sprachen

Entwicklung webbasierter Markup-Sprachen



Wenn nicht anders angegeben, beziehen sich alle Beispiele der Vorlesung auf HTML 5!

2.1 Markup-Sprachen

HTML versus CSS (I)



HTML5

- Aktuellste, kürzlich verabschiedete Version des HTML-Standards
- Standardisiert durch die W3C HTML WG und die Web Hypertext Application Technology Working Group (WHATWG)
- Arbeiteten an der Spezifikation starteten im Juni 2004
- Neue HTML-Elemente zu erweiterten Semantifizierung von Inhalten (section, article, footer,...)
- Neue HTML-Elemente zu Einbindung von Multimediainhalten als Ersatz für Plug-Ins (Flash)
- Umfangreiche JavaScript APIs (Zugriff auf Sensorik in Smartphones, Location-based Services, uvm.)

CSS3

- Cascading Style Sheets: Standard zur Spezifikation der visuellen Erscheinung von HTML-Elementen
- Version 1 erstmals 1996 veröffentlicht durch die W3C
- Version 2 ab 1998 mit vielen Verbesserungen zur absoluten und relativen Positionierung von Elementen, neuen Eigenschaften von Fonts uvm.
- CSS3 aktuellste Version mit mehr als 300 Eigenschaften
- Neue Merkmale sind Multi-Column-Layout, abgerundete Ecken, transparente Hintergründe uvm.

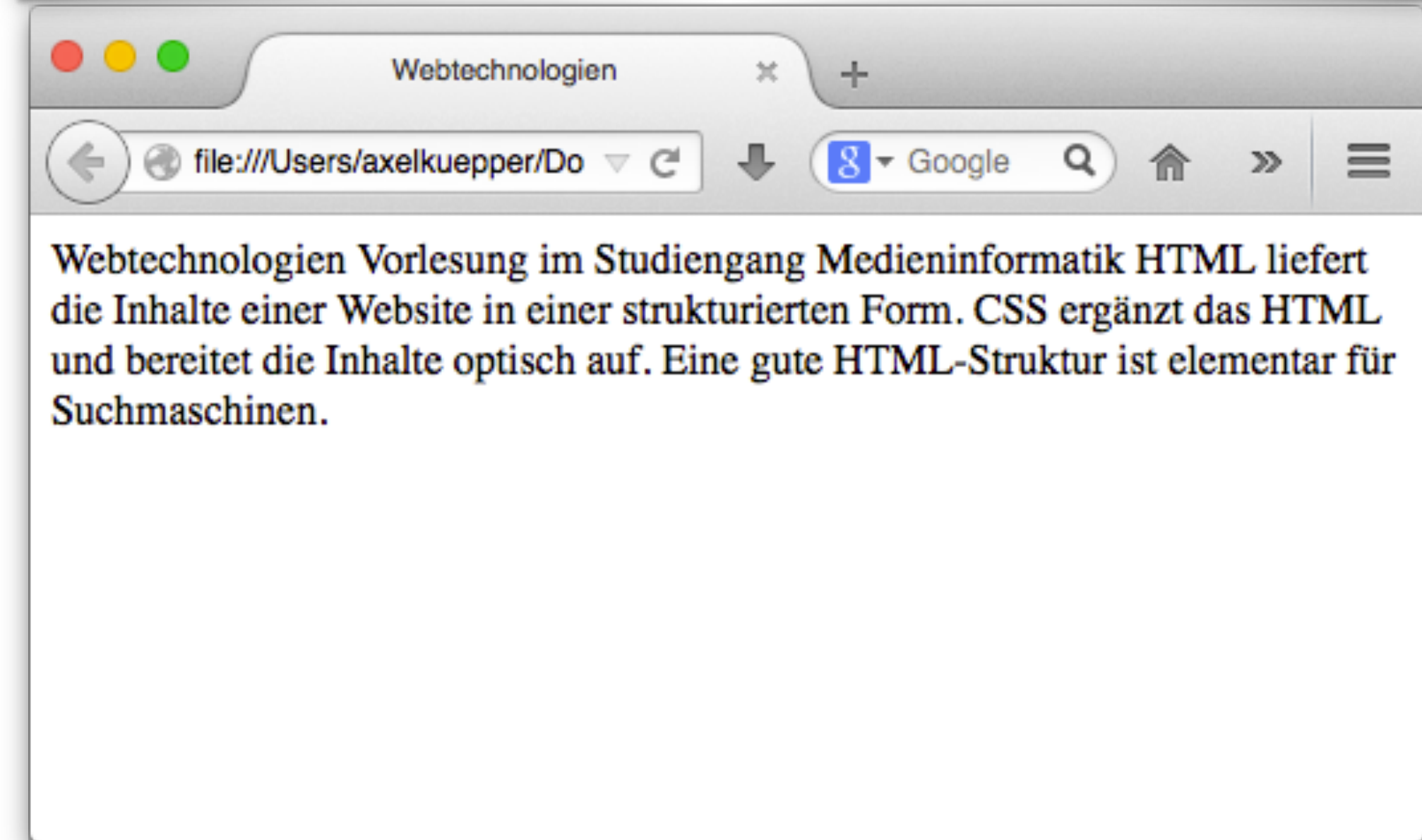
2.1 Markup-Sprachen

HTML versus CSS (II)

- HTML-Code ohne Struktur - auch Leerzeilen im Quelltext verschwinden in der Darstellung
- Mehrere Leerzeichen werden auf ein Leerzeichen reduziert
- Beispiel enthält keine Struktur, vergleichbar mit einem Buch, welches von der ersten bis zur letzten Seite den Text ohne Kapitel, Überschriften, Seitenzahlen, Absatz- und Seitenumbrüchen enthält
- Zweck von HTML ist die Strukturierung (Annotation, Auszeichnung) von Inhalten/Texten um sie in eine semantische Form zu bringen
- Strukturierung geschieht mit Hilfe von *HTML-Elementen*

```
<body>
  Webtechnologien
  Vorlesung im Studiengang Medieninformatik

  HTML liefert die Inhalte einer Website
  in einer strukturierten Form.
  CSS ergänzt das HTML und bereitet die Inhalte optisch auf.
  Eine gute HTML-Struktur ist elementar für Suchmaschinen.
</body>
```



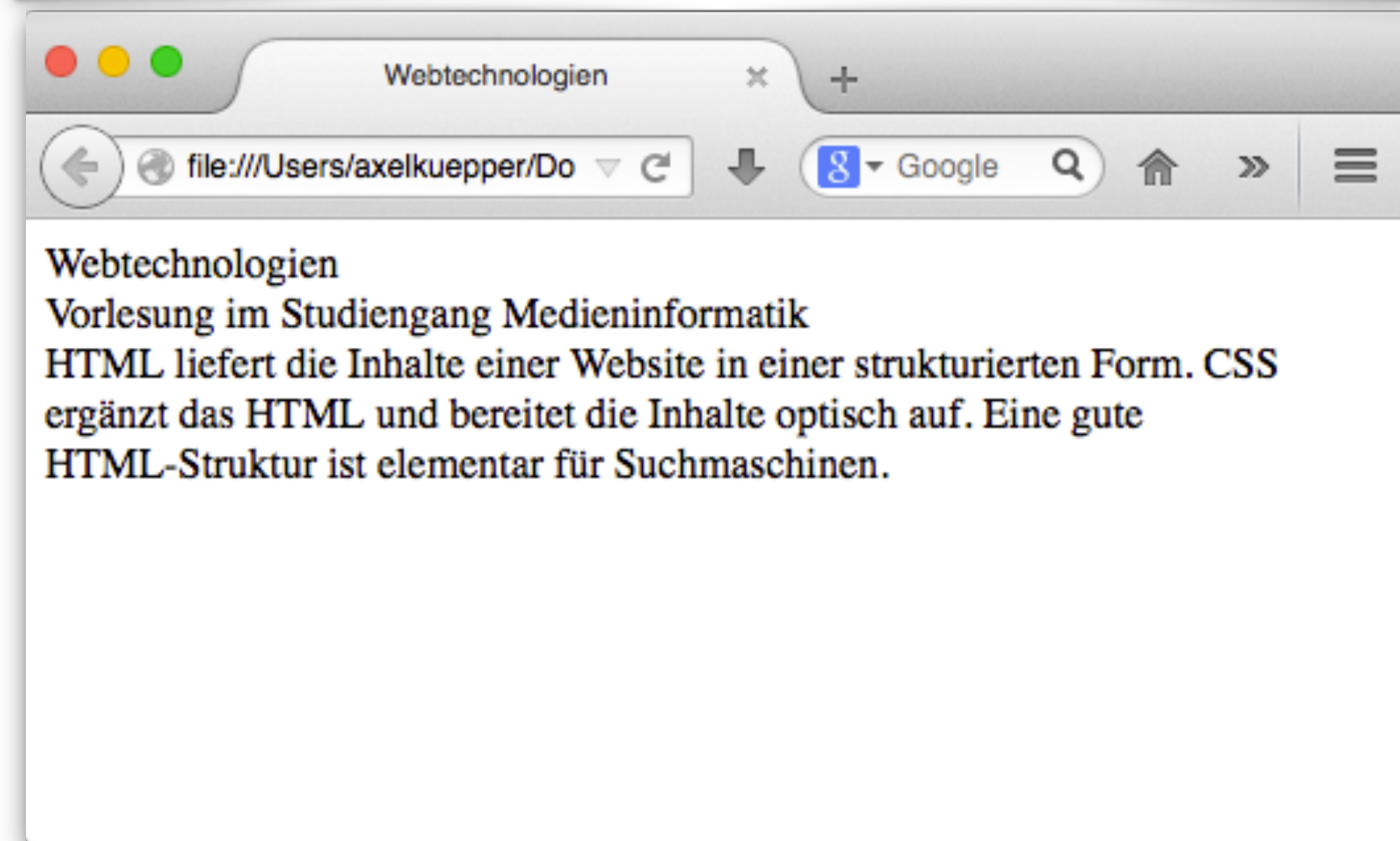
2.1 Markup-Sprachen

HTML versus CSS (III)

- Beispiel zeigt Strukturierung des Textes mit <div>-Elementen
- <div> definiert den eingeschlossenen Inhalt als einen zusammenhängenden Block, der optisch von nachfolgenden Blöcken durch eine Leerzeile getrennt wird
- <div> hat keinen Einfluss auf die logische Struktur eines Inhaltes und fügt dem Inhalt keinerlei Semantik hinzu
- <div> dient lediglich der Referenzierung von Inhalten in CSS für die Definition des Designs dieses Inhalts

```
<body>
  <div>Webtechnologien</div>
  <div>Vorlesung im Studiengang Medieninformatik</div>

  <div>HTML liefert die Inhalte einer Website
  in einer strukturierten Form.
  CSS ergänzt das HTML und bereitet die Inhalte optisch auf.
  Eine gute HTML-Struktur ist elementar für Suchmaschinen.</div>
</body>
</html>
```

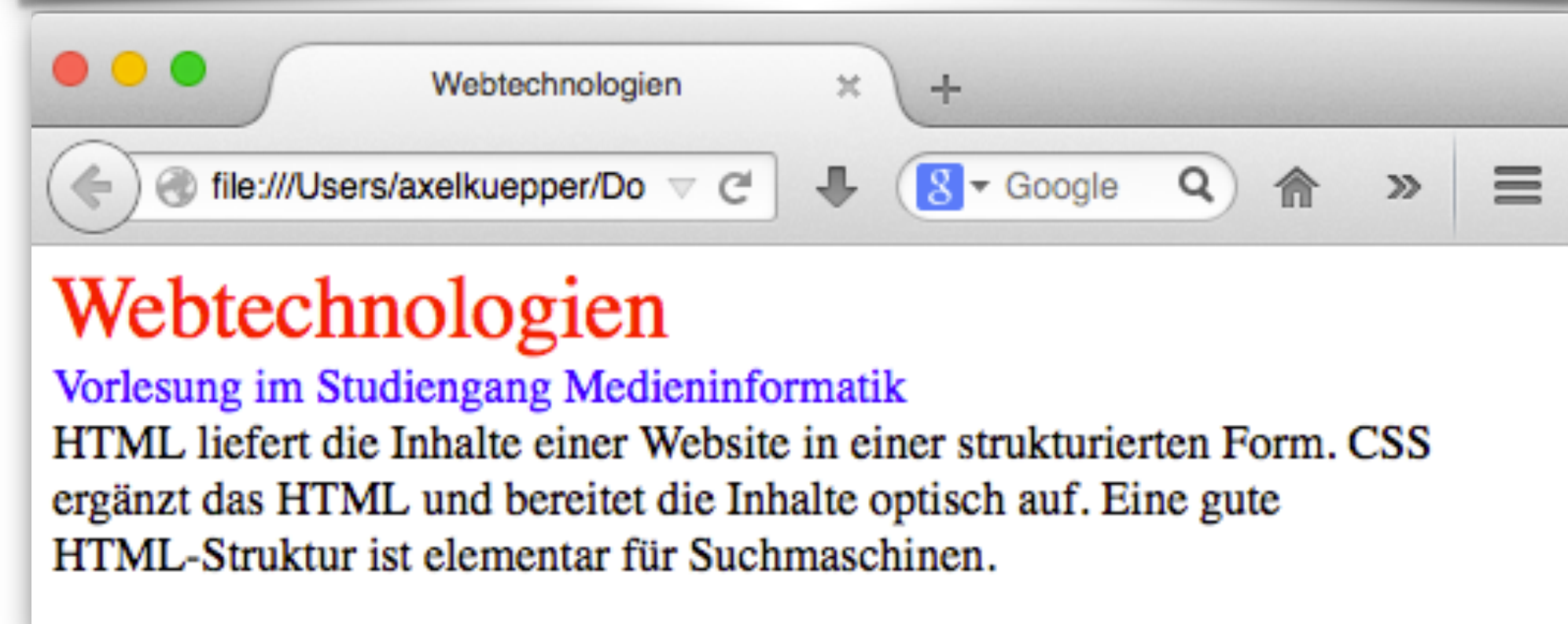


2.1 Markup-Sprachen

HTML versus CSS (IV)

- Beispiel zeigt die Einbindung von CSS-Befehlen in HTML zur optischen Anreicherung des <div>-Elemente mit Farbe und unterschiedlichen Schriftgrößen
- Optische Anreicherung ohne Struktur (wie hier im Beispiel) ist ein häufiger Fehler in der Webentwicklung - blendet man CSS aus, bleibt der Text unstrukturiert
- Keine Unterscheidung zwischen Absätzen und Überschriften möglich

```
<body>
  <div style="font-size: 30px; color: red">
    Webtechnologien
  </div>
  <div style="font-size: 24 px; color: blue">
    Vorlesung im Studiengang Medieninformatik
  </div>
  <div>
    HTML liefert die Inhalte einer Website
    in einer strukturierten Form.
    CSS ergänzt das HTML und bereitet die Inhalte optisch auf.
    Eine gute HTML-Struktur ist elementar für Suchmaschinen.
  </div>
</body>
```



2.1 Markup-Sprachen

HTML versus CSS (V)

mit Änderungen übernommen von: Heller, St. (2012) Workshop HTML 5 & CSS3, dpunkt-Verlag

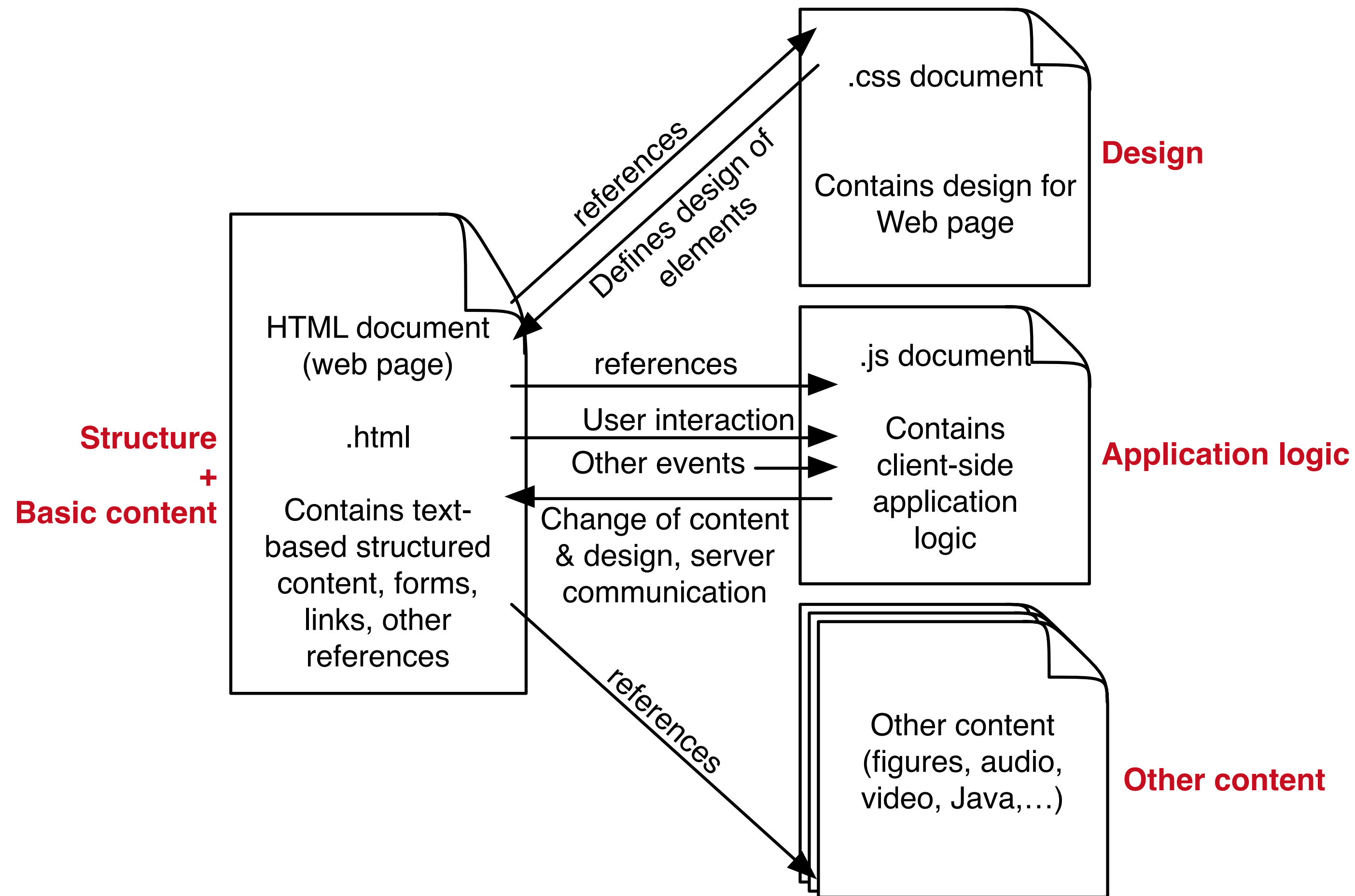
- Beispiel zeigt die Kombination von strukturiertem Text und Design mittels CSS
- Durch dedizierte Elemente kann zwischen Überschriften, Unterüberschriften und Textabsätzen unterschieden werden - der Text ist semantifiziert
- Nützlich bei der Konvertierung von Webseiten (z.B. Erstellung von Druckvorlagen) oder für die Interpretation der Inhalte durch Suchmaschinen

```
<body>
  <h1 style="font-size: 30px; color: red">
    Webtechnologien
  </h1>
  <h2 style="font-size: 24 px; color: blue">
    Vorlesung im Studiengang Medieninformatik
  </h2>
  <p>
    HTML liefert die Inhalte einer Website
    in einer strukturierten Form.
  </p>
  <p>
    CSS ergänzt das HTML und bereitet die Inhalte optisch auf.
    Eine gute HTML-Struktur ist elementar für Suchmaschinen.
  </p>
</body>
```



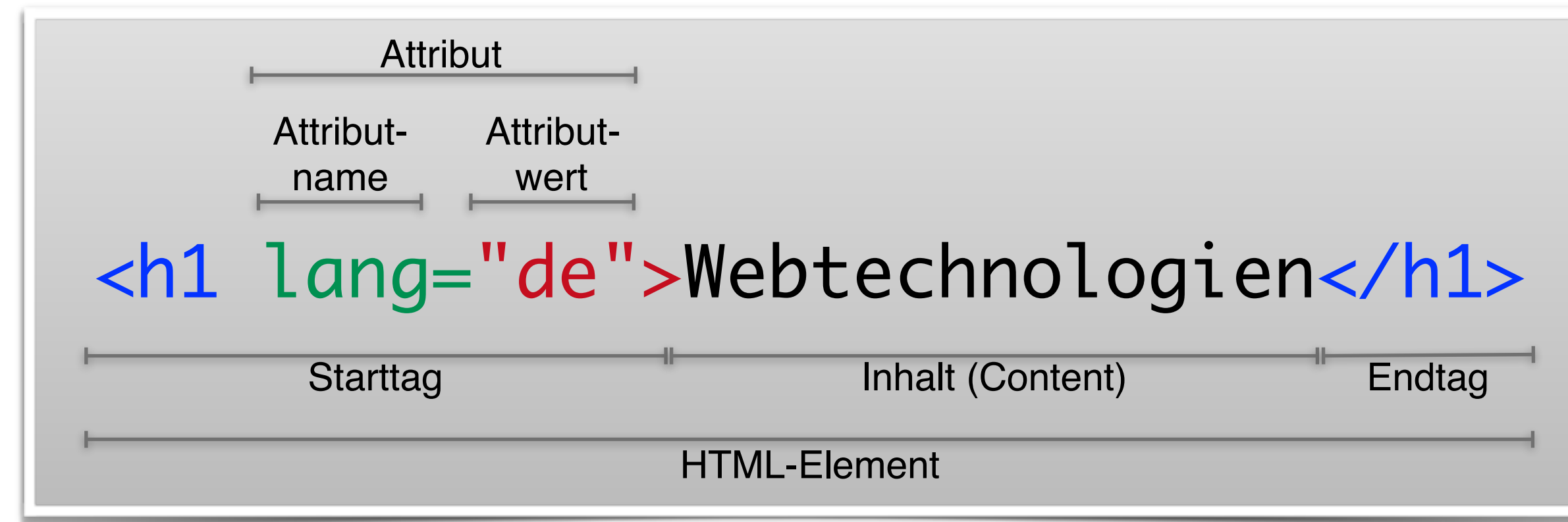
2.1 Markup-Sprachen

Verhältnis zwischen HTML, Cascading Stylesheets und JavaScript



2.2 HTML-Syntax

Basiskonzepte



HTML Element

- Die meisten *HTML-Elemente* bestehen aus
 - Starttag, einschließlich Attributen
 - Inhalt
 - Endtag
- Leere HTML-Elemente sind ebenfalls möglich und solche ohne Endtag
- Einbettung von HTML-Elementen: HTML-Element innerhalb eines anderen wird *Kindelement (child element)* genannt, das umgebende als *Elternelement (parent element)* und Kinder desselben Elternelements sind *Geschwisterelemente*
- *Case-insensitive*: es können Groß- und Kleinbuchstaben verwendet werden

Tags

- In eckigen Klammern eingefügter "Abgrenzer" der den eigentlichen Inhalt umschließt und damit semantifiziert
- *Starttag*: beginnt mit einer öffnenden Klammer, gefolgt vom HTML-Elementnamen, den Attributnamen und -werten und einer schließenden Klammer
- *Endtag*: beginnt mit einer öffnenden Klammer, einem Schrägstrich (der ihn als Endtag kennzeichnet), dem Elementnamen und einer schließenden Klammer

2.2 HTML-Syntax

Leere HTML-Elemente

Leere Elemente in
HTML

```
  
<input type="checkbox" value="1">  
<br>
```

```
</img>  
<input type="checkbox" value="1"></input>  
<br></br>
```

Schreibweise in
XHTML

```
  
<input type="checkbox" value="1" />  
<br />
```

Verkürzte
Schreibweise in
XHTML

- Es gibt leere HTML-Elemente die keinen Inhalt haben
- Verhalten von leeren Elementen wird ausschließlich über Attribute gesteuert
- Da kein Inhalt verfügbar ist, ist auch kein Endtag erforderlich
- Leere Elemente in HTML 5: area, base, br, col, command, embed, hr, img, input, keygen, link, meta, param, source, track, wbr
- XHTML benötigt zwingend Endtags, d.h. es darf keine leeren Elemente geben
- "Leere" Elemente können in XHTML abgekürzt dargestellt werden, indem am Ende des öffnenden Tags ein Schrägstrich eingefügt wird

2.2 HTML-Syntax

HTML-Elemente mit optionalem Endtag

Schreibweise in
HTML

```
<ul>
  <li>5 Eier
  <li>200g Mehl
  <li>50ml Milch
</ul>
```

```
<select name="reiseziel">
  <option>Hamburg
  <option>München
  <option>Berlin
</select>
```

Schreibweise in
XHTML

```
<ul>
  <li>5 Eier</li>
  <li>200g Mehl</li>
  <li>50ml Milch</li>
</ul>
```

```
<select name="reiseziel">
  <option>Hamburg</option>
  <option>München</option>
  <option>Berlin</option>
</select>
```

- Es gibt HTML-Elemente die keinen Endtag benötigen
- W3C: *optional schließende Tags*
- Browser erkennen das Ende des Elements entweder an einem weiteren öffnenden Tag, das damit das vorherige HTML-Element beendet, oder an einem übergeordneten Elterntag, dessen schließendes Tag darin enthaltende Kindelemente abschließt
- Elemente mit optionalem Endtag sind in XHTML ebenso wenig wie leere Elemente erlaubt - alle Tags müssen für eine korrekte Schreibweise mit einem Endtag versehen werden
- Übersichtlichkeit schaffen: auch wenn verschiedene Elemente ohne schließendes Tag auskommen, werden Quelltexte übersichtlicher, wenn alle Elemente mit einem Endtag versehen sind
- Guter Stil: HTML5 erlaubt sowohl HTML4.01 als auch XHTML1.0-Schreibweise - eine einheitlicher Schreibweise sorgt für homogene Quelltexte und gute Übersichtlichkeit

2.2 HTML-Syntax

Attribute (I)

- Attribute bestimmen die Eigenschaften eines HTML-Elements und werden ausschließlich innerhalb des Starttags angegeben
- Bestehen aus dem *Attributnamen* und einem *Attributwert* (letzterer eingegrenzt in ' oder ")
- Optionale Schreibweise ohne Anführungszeichen, dann aber nur wenn der Attribut keine Leerzeichen enthält
- Vollständiges Attribut wird mit Attributnamen eingeleitet, gefolgt von einem "=", danach folgt der Wert
- Verfügbare Attribute werden durch das HTML-Element definiert

Attribute mit festen Attributwerten

- Mögliche Werte sind vordefiniert; außer diesen können keine anderen Werte vergeben werden

```
<input type="radio">
```

Attribute mit Werten nach bestimmten Regeln

- Werte können frei vergeben werden, allerdings müssen diese bestimmten Konventionen entsprechen
- Beispiel: `tabindex` erwartet eine Zahl ohne Nachkommastellen
- Beispiel: bei `id` darf kein Leerzeichen im Wort vorkommen
- Beispiel: bei Verwendung von `class` sind keine Sonderzeichen erlaubt

```
<a href="index.html" tabindex="15">  
<p id="intro">  
<p class="beispiel-typ-3">
```

Attribute mit freien Werten

- Werte können frei vergeben werden, es müssen keine Konventionen berücksichtigt werden

```
<input src="logo.png" alt="Logo der Firma XYZ">  
<input src="logo.png" alt='Logo der Firma "XYZ"'>
```

Attribute ohne Wertzuweisung

- Boolean Attributes: Wert kann nur "wahr" oder "falsch" sein
- Alleinige Angabe des Attributnamens deklariert den Wert als "wahr"

```
<input type="checkbox" checked>
```

Schreibweise in HTML

```
<input type="checkbox" checked="checked" />
```

Schreibweise in XHTML

2.2 HTML-Syntax

Attribute (II)

Universalattribute

- Sind keinen speziellen Elementen zugeordnet und können in allen Elementen verwendet werden
- Beispiele: class and style-Attribute

```
<p class="gross">Großer Text</p>
<h1 style="color: red">Eine rote Überschrift</h1>
```

Abhängigkeiten zwischen Attributen

- Abhängigkeiten zwischen einzelnen Attributen können vorkommen

```
<input type="checkbox" checked>
```

Attribut-Standardbelegung von HTML-Elementen

- Einige Attribute verfügen über Pflichtattribute deren Werte gesetzt werden müssen
- Beispiel: <input> erfordert eine Spezifikation von type, für das mehr als 20 Attributwerte vorgesehen sind
- Weglassen von Pflichtattributen führte in alten HTML-Versionen zu Fehlern
- Default-Werte: Standardwert den HTML5 vorsieht für den Fall, dass das Pflichtattribut nicht explizit gesetzt ist
- Beispiel: Standardwert für type ist text

Zusammenfassung Attribute

- Menge der verfügbaren Attribute hängt vom jeweiligen HTML-Element ab
- Einige HTML-Elemente haben keine Attribute
- Einige Attribute erwarten vordefinierte Werte, für andere können Werte frei vergeben werden
- Universalattribute können in allen HTML-Elementen gesetzt werden
- Attribute können groß- oder kleingeschrieben werden
- Einheitliche Schreibweise verschafft Übersicht und reduziert Fehlerquellen

2.2 HTML-Syntax

Steuerzeichen, Sonderzeichen und Leerzeichen (I)

Steuerzeichen

- Syntax benötigt einige Zeichen als Steuerzeichen, zu Beispiel <, >, ", ', &
- Zur Unterscheidung zwischen Steuerzeichen und Inhalten müssen Inhaltszeichen die identisch mit Steuerzeichen sind im Inhaltsteil im HTML-Entity-Format geschrieben werden
- Zwei Schreibweisen: HTML und Unicode
- HTML: *Entity Name*
- Unicode: *Entity Number*
- Beide Schreibweisen beginnen mit einem & und enden mit einem ; (*HTML Entity*)
- Moderne Browser erkennen reservierte Zeichen im Text und können diese von HTML-Steuerzeichen unterscheiden
- Ausnahme: < muss im Textteil immer als < geschrieben (führt sonst zu Validierungsfehler), > sollte als > geschrieben werden (würde aber nicht zu Validierungsfehler führen)
- Kaufmännisches Und (&): nicht Bestandteil der HTML-Syntax, aber zur Trennung von Parametern im <a>-Element beim href-Attribut benötigt
- Attributliste wird durch Fragezeichen vom Name der Scripts getrennt

```
<input href="index.php?land=Spain&ort=Madrid">Linktext</a>
```

```
<input href="index.php?land=Spain&amp;ort=Madrid">Linktext</a>
```

Zeichen	Schreibweise in HTML	Schreibweise in Unicode	Beschreibung	Abgeleitet von
<	<	<	Öffnende spitze Klammer/Kleiner-als-Zeichen	lower than
>	>	>	Schließende spitze Klammer/Größer-als-Zeichen	greater than
"	"	"	Doppelte Anführungszeichen	quotation mark
'	'	'	Einfaches Anführungszeichen	apostrophe
&	&	&	Kaufmännisches Und	ampersand

Moderne Version (nicht durch alte Browser unterstützt)

Alte Version (durch alte und neue Browser unterstützt)

2.2 HTML-Syntax

Steuerzeichen, Sonderzeichen und Leerzeichen (II)

Sonderzeichen

- Sonderfall deutsche Sonderzeichen und Symbolzeichen
- UTF-8-Standard unterstützt Zeichen in nebenstehender Tabelle - eine Codierung als HTML Entity ist nicht notwendig
- Verwendung anderer Zeichencodierungen (beispielsweise das weit verbreitete ISO 8859-1) erfordert Verwendung des HTML-Entity-Formats damit Zeichen korrekt dargestellt werden

Geschütztes Leerzeichen

- Normales HTML passt Text an verfügbaren Platz an
- Zeilen werden nicht nach einer bestimmten Anzahl von Zeichen zeilenumgebrochen, sondern nur dort wo Leerzeichen vorhanden sind
- Einige Begriffe bestehen aber aus mehreren Wörtern - ein Zeilenumbruch innerhalb des Begriffes würde die Leserlichkeit verschlechtern
- ` ` ist ein geschütztes Leerzeichen welches den Zeilenumbruch verhindert

```
<p>Web 2.0</p>
```

Zeichen	Schreibweise in HTML	Schreibweise in Unicode	Beschreibung	Abgeleitet von
ä	ä	ä	a-Umlaut	[a] + umlaut
Ä	Ä	Ä	A-Umlaut	[A] + umlaut
ö	ö	ö	o-Umlaut	[o] + umlaut
Ö	Ö	Ö	O-Umlaut	[O] + umlaut
ü	ü	ü	u-Umlaut	[u] + umlaut
Ü	Ü	Ü	U-Umlaut	[U] + umlaut
ß	ß	ß	Scharfes S (Eszett)	[sz] + ligature
€	€	€	Euro-Zeichen	euro
©	©	©	Copyright-Zeichen	copyright
	 	 	Geschütztes Leerzeichen	non-breaking space
§	§	§	Paragrafenzeichen	section

2.2 HTML-Syntax

Relative und Absolute Pfade

- Dateien wie Grafiken, JavaScript oder Cascading Style Sheets können sowohl relativ als auch absolut eingebunden werden
- Relative Einbindung einer Grafik in den Dateien impressum.html oder kontakt.html (basierend auf nebenstehender Grafik):

```

```

- Die beiden Punkte .. vor /img führen den Browser in das übergeordnete Verzeichnis, dann in den entsprechenden Ordner, in dem die Grafik liegt
- Alternative Einbindung über einen absoluten Pfad:

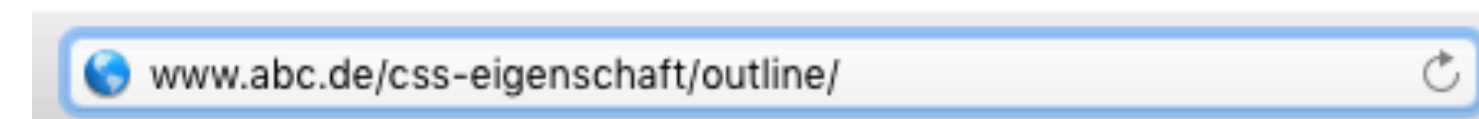
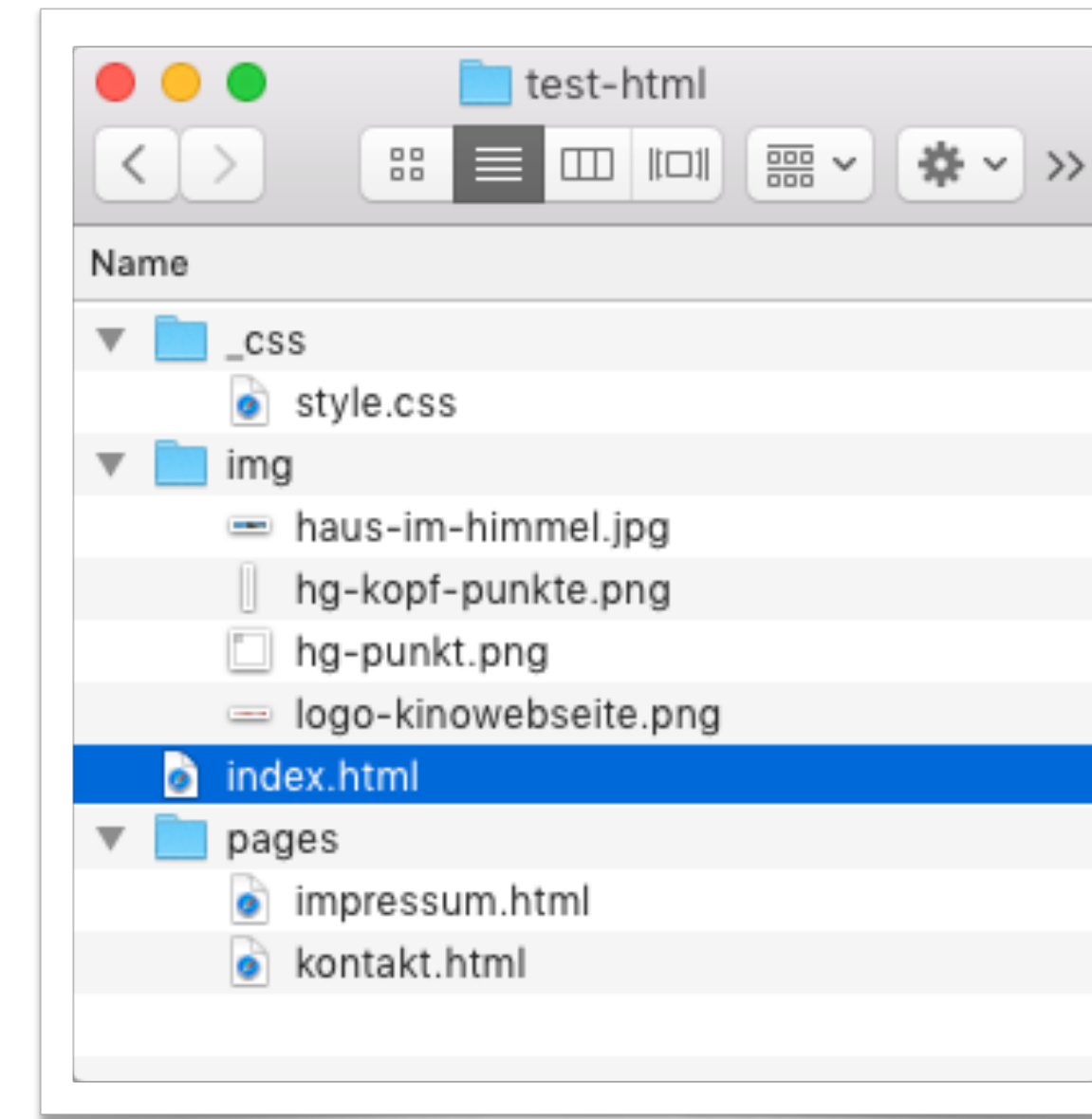
```

```

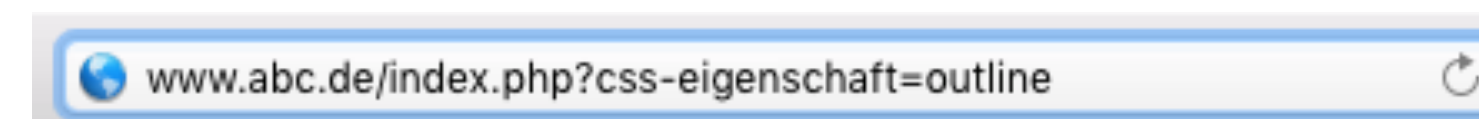
- Schrägstrich / signalisiert dem Browser die Datei ausgehend vom Root-Verzeichnis der Website aus zu suchen

"Schöne" oder "Simulierte" Pfade

- Für Suchmaschinenoptimierung werden kryptische URLs durch suchmaschinenfreundliche Adressen ersetzt
- Für den Webbrowser erscheint es, als läge eine Datei im Verzeichnis /css-eigenschaft/outline/
- Webserver (genauer eine .htaccess-Datei) bildet simulierten Pfad auf den echten Pfad ab und ruft eine index.php-Datei auf



"Schöner" Pfad



Tatsächlicher Pfad

2.2 HTML-Syntax

Aufbau einer HTML-Seite

```
<!DOCTYPE html>
<html lang="de">
  <head>
  </head>
  <body>
  </body>
</html>
```

- Grundaufbau einer HTML-Datei besteht aus den Elementen `<html>`, `<head>` und `<body>`
- Universalattribut `lang` kennzeichnet Sprache der Webseite

HTML 5

```
<!DOCTYPE html>
```

HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

XHTML 1.0

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Doctype

- Teilt dem Browser mit nach welchem Standard die HTML-Datei aufgebaut ist

Head

- Elemente im Head-Bereich sind in den meisten Fällen für den Webseitenbesucher nicht zu sehen (Ausnahme: `<title>`)
- Definition von `<meta>`-Elementen die weitere Informationen zur Webseite liefern (zum Beispiel für Auswertung durch Suchmaschinen)
- Einbindung von JavaScript- und CSS-Dateien

Body

- Liefert alle sichtbaren Elemente einer Webseite

2.3 HTML Head Elemente in <head> (I)

Schlechter Titel

```
<title>...:°l°:- Meine Homepage ...:°l°:...</title>
```

Guter Titel

```
<title>Handgestrickte Socken - Freiburg - Lieschen  
Müller Häkelstübchen</title>
```

```
<head>  
<meta charset="UTF-8">  
<meta name="description" content="Free Web tutorials">  
<meta name="keywords" content="HTML,CSS,XML,JavaScript">  
<meta name="author" content="Hege Refsnes">  
</head>
```

<title>-Element

- Inhalt von <title> wird in der Browser- bzw. Tableiste angezeigt sowie von Suchmaschinen in der Ergebnisliste
- <title> sollte aussagekräftig gewählt werden und angemessen den Inhalt der Webseite beschreiben
- Relevante Information sollte in den ersten 65 Zeichen stehen, da nur diese in den Suchergebnissen angezeigt werden

<meta>-Element

- Metadaten sind Daten über Daten
- <meta>-Element enthält Metadaten über die dargestellte Webseite
- Inhalt von <meta> wird nicht auf der Webseite dargestellt, kann aber durch Maschinen (zum Beispiel Suchroboter) gelesen werden
- Metadaten können durch Browser interpretiert werden (zum Beispiel wie Inhalte einer Seite dargestellt werden sollen), Suchmaschinen (Keywords) oder durch andere Web Services
- Attribut charset dient der Angabe des Zeichensatzes in denen eine Webseite dargestellt werden soll (bspw. zur korrekten Darstellung von Sonderzeichen)
- Verwendung von charset nicht zwingend erforderlich, aber nützlich, wenn z.B. der HTML-Editor bei der Kreierung der Webseite einen anderen Editor verwendet

2.3 HTML Head

Elemente in <head> (II)

Stylesheet-
Einbindung mit
relativem Pfad

```
<link rel="stylesheet" href="style.css">
```

Stylesheet-
Einbindung mit
absolutem Pfad

```
<link rel="stylesheet" href="lib/css/style.css">
```

Verschiedene
Stylesheets für
unterschiedliche
Ausgabegeräte

```
<link rel="stylesheet" media="screen, braille, projection"  
href="style.css">  
<link rel="stylesheet" media="print, embossed"  
href="style.css">
```

<link>-Element

- Definiert Verweis zwischen der Webseite die <link> enthält und einer externen Ressource
- Wird verwendet um CSS- und JavaScript-Dateien einzubinden, entweder über absolute oder relative Pfade
- rel-Attribut beschreibt Verhältnis zwischen der einbindenden oder der eingebundenen Webseite, z.B. stylesheet, help oder icon
- media-Attribut zusammen mit rel=stylesheet spezifiziert die Art des Ausgabegerätes für die das Stylesheet verwendet werden soll

Media-Typ	Vorgesehen für...
all	Alle Ausgabegeräte (Standard, wenn kein Media-Typ angegeben ist).
braille	Braillezahlen, d.h. Ausgabegeräte für Blinde, über die Bildschirminhalte als Braille wiedergegeben werden.
embossed	Drucker mit Blindenschrift.
handheld	"Handgeführte" Geräte, d.h. solche mit kleinem oder monochromen Bildschirmen sowie Geräten mit geringer Bandbreite.
print	Für den Ausdruck bestimmte Dokumente, aber auch solche die in Druckvorschau angezeigt werden sollen.
projection	Für die Projektion mittels Beamer.
screen	Computer-Bildschirme die Inhalte nicht seitenweise, sondern fortlaufend anzeigen.
speech	Sprachgeneratoren.
tv	Fernsehgeräte mit geringer Bildschirmauflösung und ohne Scroll-Fähigkeit.

2.3 HTML Head

Elemente in <head> (III)

Direkte CSS-Anweisung in <head>

```
<style>
  h1 {
    color: darkblue;
  }
</style>
```

Direkte JS-Anweisung in <head>

```
<script>
  alert ('Hallo Welt!');
</script>
```

Einbindung externer JS-Dateien

```
<script src="externes-js.js"></script>
```

Nicht erlaubt/nicht empfohlen

```
<script src="externes-js.js">
  alert ('Hallo Welt!');
</script>
```

besser

```
<script src="externes-js.js"></script>
<script>
  alert ('Hallo Welt!');
</script>
```

<style>-Element

- Innerhalb von <style> kann CSS direkt im Kopf der HTML-Datei definiert werden
- Standardwert für das Attribut <type> ist text/css
- <style>-Element ist ausschließlich im Head zulässig

<script>-Element

- <script> bindet JavaScript-Anweisungen im Head der HTML-Seite ein oder bindet eine externe JavaScript-Datei ein
- Einbindung einer externen Datei funktioniert nicht über <link> und href
- Keine Kombination von eingebundenen Anweisung und JavaScript-Dateien innerhalb eines <script>-Elements

2.4 HTML Body

HTML-Elemente zur Seitenstrukturierung

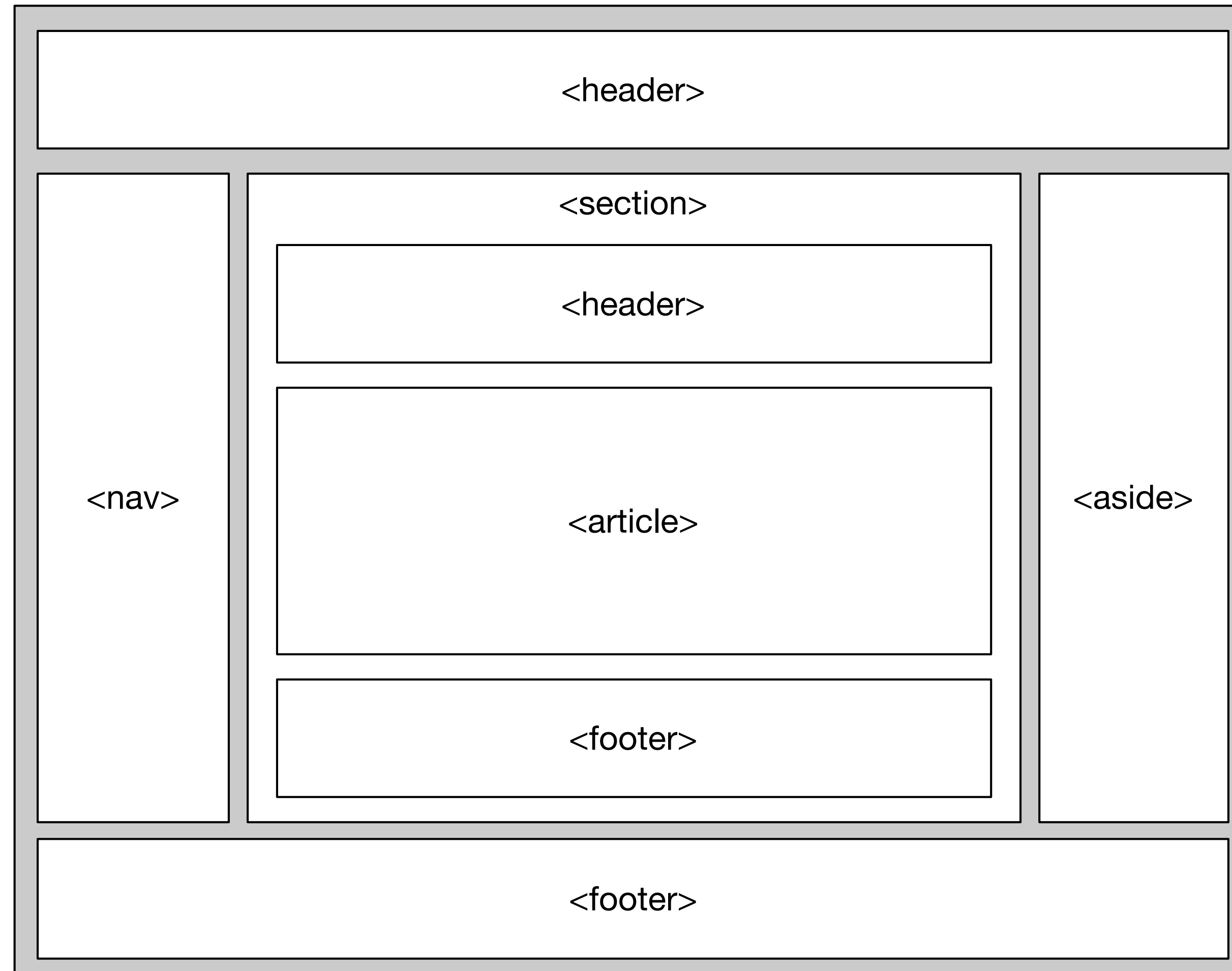
Element	Funktion	ab HTML 5
body	Umschließt den sichtbaren Inhalt einer Webseite.	
section	Definiert einen allgemeinen Bereich einer Seite, in dem zusammengehöriger Inhalt gebündelt wird.	Ja
nav	Definiert Navigationsbereiche.	Ja
article	Definiert einen Artikel mit zusammengehörigen Informationen, der grundsätzlich auch allein stehen könnte und von dem umgebenden Kontext unabhängig ist.	Ja
aside	Definiert Bereiche, die verwandte Themen zum Hauptinhalt zeigen, grundsätzlich aber auch ohne Bezug auf den Inhalt eine eigenständige Information liefern.	Ja
h1, h2, h3, h4, h5, h6	Überschriften für einen Seitenbereich.	
hgroup	Fasst mehrere Überschriften zu einem Überschriftenblock zusammen.	Ja
header	Definiert einen Bereich als Kopf eines Seitenbereichs. Kann als eigentlicher Kopf einer Website eingesetzt werden, aber auch als Kopf beispielsweise einer section.	Ja
footer	Definiert eine Fußzeile zu einem Bereich und umfasst Angaben zum Autor, Copyright, weiterführende Links usw. Kann als eigentlicher Seitenfuß einer Website eingesetzt werden, aber auch als Element in section oder article.	Ja
address	Dient zur Auszeichnung eines Abschnitts mit Kontaktmöglichkeiten, z.B. Link zur Kontaktseite oder Email-Link. Telefonnummern und postalische Adressen sind laut Standard im address-Element nicht erlaubt.	

- Auszeichnung von Seitenbestandteilen mit Semantik
- Weniger wichtig für Optik als vielmehr für die Auswertung durch Suchmaschinen
- Beispiel: durch <nav> ausgezeichnete Navigationsbereich wird anders interpretiert als <section> oder <article>

mit Änderungen übernommen von: Heller, St. (2012) Workshop HTML 5 & CSS3, dpunkt-Verlag

2.4 HTML Body

HTML-Elemente zur Seitenstrukturierung - Beispiel



2.4 HTML Body

HTML-Elemente zur Textstrukturierung

Element	Funktion	ab HTML 5
p	Definiert einen Textblock (Paragraf).	
hr	Definiert eine Trennlinie, die Inhalte in logische Abschnitte unterteilt.	
pre	Stellt Textblöcke mit Zeilenumbrüchen, Einrückungen und Leerzeichen dar, wie sie im Quelltext erfasst sind. Ist ausschließlich eine visuelle Darstellung, die beispielsweise in Sprachausgaben verloren geht.	
blockquote	Definiert einen Abschnitt als Zitat. Wird wie das q-Tag nicht mit Anführungszeichen, aber eingerückt dargestellt.	
ol	Definiert eine geordnete Liste, die automatisch nummeriert wird, entweder mit Zahlen oder Buchstaben (über CSS steuerbar). Merkmal ist, dass die Reihenfolge der einzelnen Einträge nicht frei vertauscht werden kann, ohne dass die Sinnhaftigkeit verloren geht (z.B. bei einer Bauanleitung).	
ul	Definiert eine ungeordnete Liste. Merkmal ist, dass die Reihenfolge der Listeneinträge frei vertauscht werden kann, die Liste aber trotzdem sinnvollen Inhalt liefert (z.B. die Zutatenliste eines Rezepts).	
li	Definiert einen Listeneintrag sowohl in einer ol als auch in einer ul.	
figure	Definiert eine in sich abgeschlossene Einheit, beispielsweise ein Bild, Video oder Audio mit der dazugehörigen Beschriftung oder ein Gedicht mit den dazugehörigen Angaben zu Titel, Autor und Erscheinungsdatum.	Ja
figcaption	Definiert die Beschriftung einer figure.	Ja
div	Blockelement ohne jegliche strukturelle Bedeutung. Sollte nur eingesetzt werden, wenn keine anderen Strukturelemente für die Auszeichnung eines Abschnitts sinnvoll erscheinen.	

- Auszeichnung von Text mit Semantik
- Häufige Verwendung von Paragraphen (<p>), Listen und Listenelementen (,) und Bildunterschriften (<figcaption>)
- <div> wird für "sonstige" Blöcke verwendet für die es kein dediziertes Element gibt, die jedoch optisch besonders hervor gehoben werden sollen

mit Änderungen übernommen von: Heller, St. (2012) Workshop HTML 5 & CSS3, dpunkt-Verlag

2.4 HTML Body

HTML-Elemente zur Textauszeichnung (I)

Element	Funktion	ab HTML 5
a	Definiert einen Hyperlink zu einer anderen Seite.	
em	Betont ein Wort, hebt es durch Akzentuierung hervor. Die visuelle Darstellung ist kursiv.	
strong	Verleiht einem Wort Wichtigkeit und hebt es dadurch vom restlichen Text ab. Die visuelle Darstellung ist fett.	
cite	Definiert den Titel einer Quelle, z.B. <cite>Die Bibel</cite>. Ist abzugrenzen von eigentlichen Zitaten, die über blockquote oder einen q-Tag ausgezeichnet werden sollen.	
q	Definiert ein Zitat. Wird visuell durch Anführungszeichen dargestellt.	
abbr	Definiert eine Abkürzung oder ein Akronym. Im title-Attribut sollte die Bedeutung des Akronyms beziehungsweise die Abkürzung ausgeschrieben werden.	
time	Zeichnet eine Zeitangabe aus.	Ja
code	Zeichnet einen Text als Quellcode oder Computerbefehl aus. Visuell wird ein code-Abschnitt über einen Monospace-Schrifttyp dargestellt. Um Zeilenumbrüche von Befehlen zu erhalten, ist zusätzlich der Einsatz des pre-tags notwendig.	
kbd	Zeichnet einen Text aus, der auf eine Benutzereingabe über die Tastatur hinweist. Beispielsweise: Zum Neuladen der Seite drücken Sie die Taste <kbd>F5</kbd>.	
sub	Stellt einen Text tief. Sollte nur eingesetzt werden, wenn es den Konventionen einer Sprache entspricht, zum Beispiel bei der Auszeichnung chemischer Zeichen: H₂ für H ₂ O.	
sup	Stellt einen Text hoch. Sollte ebenfalls nur eingesetzt werden, wenn es den Konventionen einer Sprache entspricht, beispielsweise bei einer Quadratmeterangabe: 45m² für 45m ²	

2.4 HTML Body

HTML-Elemente zur Textauszeichnung (II)

Element	Funktion	ab HTML 5
mark	Definiert einen markierten oder hervorgehobenen Text, ist jedoch mit keiner Betonung wie durch em oder strong verbunden. Sinnvoller Einsatz kann das Hervorheben eines Suchbegriffs in einer Suchergebnisliste sein. Die Standarddarstellung ist ein gelb hinterlegter Text.	
span	Definiert einen Bereich als Inline-Element ohne jegliche Bedeutung. Eine semantische Auszeichnung durch ein anderes Element ist immer zu bevorzugen. Sinnvoller Einsatz ist die Sprachauszeichnung eines sonst geläufigen Begriffs wie beispielsweise Meeting, damit Screenreader diesen mit der richtigen Aussprache vorlesen.	
br	Definiert einen Zeilenumbruch innerhalb eines p- oder address-Tags. Sollte auf keinen Fall eingesetzt werden, um Abstände zwischen HTML-Elementen zu erzeugen. Sind mehrere nacheinander in einem p-Tag verwendet, ist zu prüfen, ob der relevante Text in zwei p-Blöcke aufgeteilt werden kann.	
wbr	Definiert optionale Zeilenumbrüche, damit lange Wörter an der richtigen Stelle getrennt werden. Beispielsweise Do<wbr>nau<wbr>dampf<wbr>schi ff<wbr>fahrts<wbr>ge<wbr>sell<wbr>schafts<wbr>ka<wbr>pi<wbr>täns<wbr>müt<wbr>ze.	Ja

- Spezielle Auszeichnung von Text zum besseren Verständnis , zum Beispiel durch Betonung, Hervorheben, Definition, Zeitangaben etc.

2.4 HTML Body

HTML-Elemente für Formulardarstellung (I)

Element	Funktion	ab HTML 5
form	Definiert ein Formular und muss alle Formularelemente umschließen. Im <code><form></code> -Tag wird die URL angegeben an die ein Formular gesendet wird sowie der Übertragungsmodus (<code>get</code> , <code>post</code>) festgelegt. Ebenfalls wird über den <code>enctype</code> der Übertragungsmodus festgelegt, der im Falle eines Dateiuploads den Wert <code>multipart/form-data</code> haben muss.	
fieldset	Gliedert ein Formular und fasst inhaltlich zusammengehörige Formularfelder zusammen.	
legend	Definiert eine übergeordnete Beschriftung von zusammengehörigen Formularfeldern, die über <code>fieldset</code> zu einer Einheit zusammengefasst werden.	
label	Definiert eine eindeutige Beschriftung für jedes einzelne Eingabefeld. Jedem <code><label></code> kann nur ein Formularelement (<code>input</code> , <code>select</code> , <code>textarea</code>) zugeordnet werden. Die Verknüpfung wird über das <code>for</code> -Attribut des <code>label</code> und das <code>id</code> -Attribut des Eingabefeldes hergestellt.	
input	HTML-Tag für verschiedene Typen von Eingabefeldern. Einzeiliges Eingabefeld zur Eingabe von Daten durch den Benutzer, Radio- und Checkboxen zur Auswahl, Upload-Button-Feld, Absende- und Reset-Button. Mit HTML5 hat sich der Umfang der <code>input</code> -Typen und damit die Datenarten, die damit erfasst werden können, wesentlich erhöht.	Ja (viele neue Werte für <code>type</code>).
datalist	Definiert Werte die als Vorauswahl für ein einzeiliges Eingabefeld ausgewählt werden können.	Ja
select	Definiert eine Auswahlliste, auch Auswahlmenü oder Dropdown genannt, mit einer Liste fest definierter Werte.	
option	Definiert eine einzelne Auswahlmöglichkeit innerhalb einer <code>select</code> -Box.	
optgroup	Gruppiert mehrere <code><option></code> -Tags innerhalb einer <code>select</code> -Box zu einer Einheit von Auswahlmöglichkeiten.	

2.4 HTML Body

HTML-Elemente für Formulardarstellung (II)

Element	Funktion	ab HTML 5
textarea	Mehrzeiliges Eingabefeld.	
keygen	Generiert ein Schlüssel-Paar für eine sichere Datenübertragung.	Ja
output	Definiert ein Feld, in dem das Ergebnis einer Rechenoperation angezeigt wird.	Ja
progress	Definiert einen Fortschrittsbalken der den Status einer Aufgabe darstellt. Zur Dynamisierung kann kann der Wert des Balkens über JavaScript verändert werden.	Ja
meter	Definiert eine Art Pegelanzeige, welche die Leere oder Fülle darstellt, zum Beispiel die genutzte Kapazität einer Festplatte oder die Relevanz eines Suchergebnisses für einen Suchbegriff.	Ja
button	Definiert einen Button, der je nach type als Submit- oder Reset-Button eingesetzt werden kann. Ohne type wird beim Klicken des Button ein Formular nicht abgesendet. <button> kann auch außerhalb von Formularen eingesetzt werden, um beispielsweise JavaScript-Aktionen anzustoßen.	

mit Änderungen übernommen von: Heller, St. (2012) Workshop HTML 5 & CSS3, dpunkt-Verlag

2.4 HTML Body

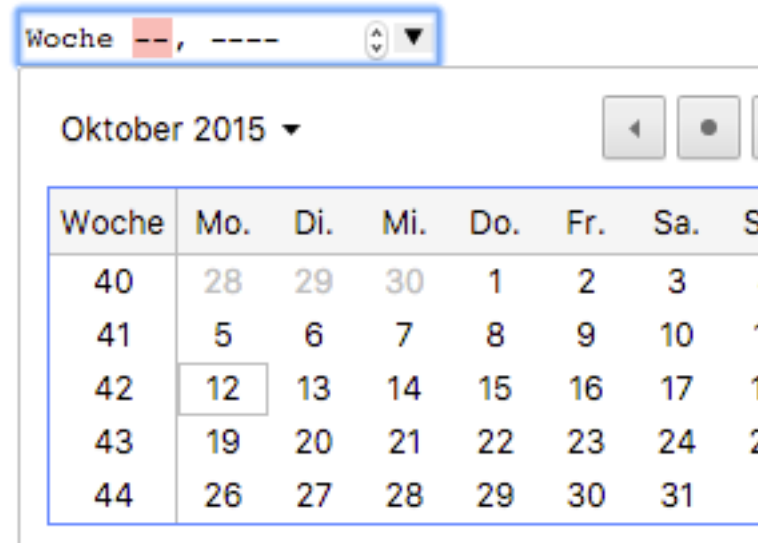
HTML-Elemente für Formulardarstellung (III)

```
<form action="zielseite.html" method="get" name="mein-formular">
  <input...
  <input...
  <input...
  ...
</form>
```

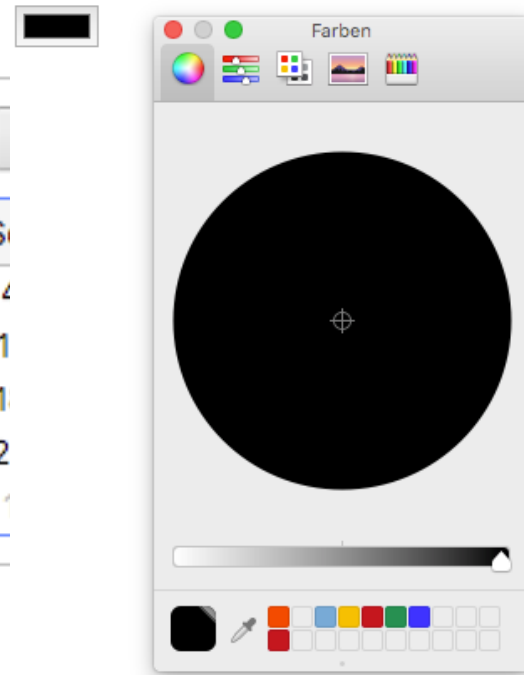
```
<input type="time">
```



```
<input type="week">
```



```
<input type="color">
```



```
<input type="range">
```



```
<meter value="80" max="100"></meter>
```



- Interaktion zwischen Nutzer und Website durch Formulare
- Formularfelder werden durch ein umschließendes `<form>`-Element definiert
- Alle Eingabefelder, Klickboxen und Auswahllisten müssen innerhalb des `<form>`-Elements stehen
- `action`-Attribut ist Pflicht und gibt an, wohin das Formular gesendet werden soll
- `method`-Attribut spezifiziert die Methode mit welcher der Formularinhalt gesendet werden soll

`method="get"`

- Formulareingabe wird mittels einer HTTP-Get-Methode gesendet, d.h. sie wird an die URL angehängt, die in `action` angegeben ist
- Nach dem Absenden des Formulars sind die Formulareingaben in der Browserzeile sichtbar
- HTTP empfiehlt eine Maximalgröße von 255 Bytes für URLs, daher nur für kleine Formulare geeignet

`method="post"`

- Übertragung der Daten im Nutzdatenteil einer HTTP-Post-Nachricht
- Übertragung der Daten ist für den Nutzer nicht sichtbar
- Je nach Konfiguration des Webserver kann die maximale Dateigröße beschränkt sein, z.B. 8 MB

2.4 HTML Body

HTML-Elemente zur Einbettung von Medien und Inhalten

Element	Funktion	ab HTML 5
img	Bindet eine Grafik oder Bild ein.	
iframe	Bindet Inhalte von einer fremden Webseite über die Angabe einer Webadresse (URL) im Attribut src ein.	
embed	Bettet eine Datei einer externen Applikation ein, die keinem W3C-Standard entspricht, z.B. ein Flash- oder QuickTime-Plugin. Setzt voraus das die entsprechende Software im Browser installiert ist.	Ja
object	Bettet eine Datei einer externen Applikation ein, die keinem W3C-Standard entspricht, z.B. ein Flash- oder QuickTime-Plugin. Setzt voraus das die entsprechende Software im Browser installiert ist.	
param	Definiert Parameter für ein Objekt, das über object eingebunden ist.	
video	Bindet Videos oder Filme ein	Ja
audio	Bindet Audiodateien oder Audiostreams ein.	Ja
source	Definiert eine oder mehrere Quelldateien innerhalb des video- oder audio-Elements. Browser suchen sich eine Quelle aus, die sie unterstützen.	Ja
track	Definiert die Quelle für eine Untertitelung innerhalb des <audio>- oder <video>-Elements.	
canvas	Definiert einen Bereich, in dem dynamische Grafiken, Diagramme oder Bilder skriptgesteuert dargestellt oder verändert werden können.	Ja

2.4 HTML Body

Beispiel Video

```
<video controls preload poster="img/universal.jpg">
  <source src="media/jurassicpark.mp4" type="video/mp4">
  <source src="media/jurassicpark.ogv" type="video/ogg">
  <p>
    Ihr Browser unterstützt die direkte Anzeige von Videos nicht.<br>Das
    Video zum Download: <a href="media/jurassicpark.mp4">Jurassic Park
  </a>
  </p>
</video>
```

- Es können verschiedene Quellen mit unterschiedlichen Kodierungen angegeben werden
- Browser wählt eine kompatible Kodierung
- Wenn Browser Kodierungen nicht unterstützt, erfolgt ein Hinweis zusammen mit einem Link zum Download



- Einbindung von Video- oder Audioinhalten vor HTML5 nur mittels <object> und <embed>-Elementen und Flash- oder anderer Plug-Ins
- HTML5 ermöglicht Einbindung ohne Plug-Ins mittels <audio>- und <video>-Elementen

Beispiel <video>

- Einbindung von Videos unterschiedlicher Kodierungen mit vielen Konfigurationsmöglichkeiten

Attribut	Funktion
controls	Ergänzt Video um eine Bedienleiste, deren Layout über CSS und JavaScript angepasst werden kann angepasst
autoplay	Startet das Video automatisch beim Laden der Seite
loop	Spielt das Video in einer Endlosschleife
poster	Referenziert eine Grafik die gezeigt wird wenn das Video nicht abgespielt wird
width und height	Breiten- und Höhenangaben für das Video
preload	Steuert ob die Videodatei beim Laden der Seite vorgeladen wird oder nicht, beziehungsweise ob nur Metadaten geladen werden (möglich Werte: auto, metadata, none)

2.4 HTML Body

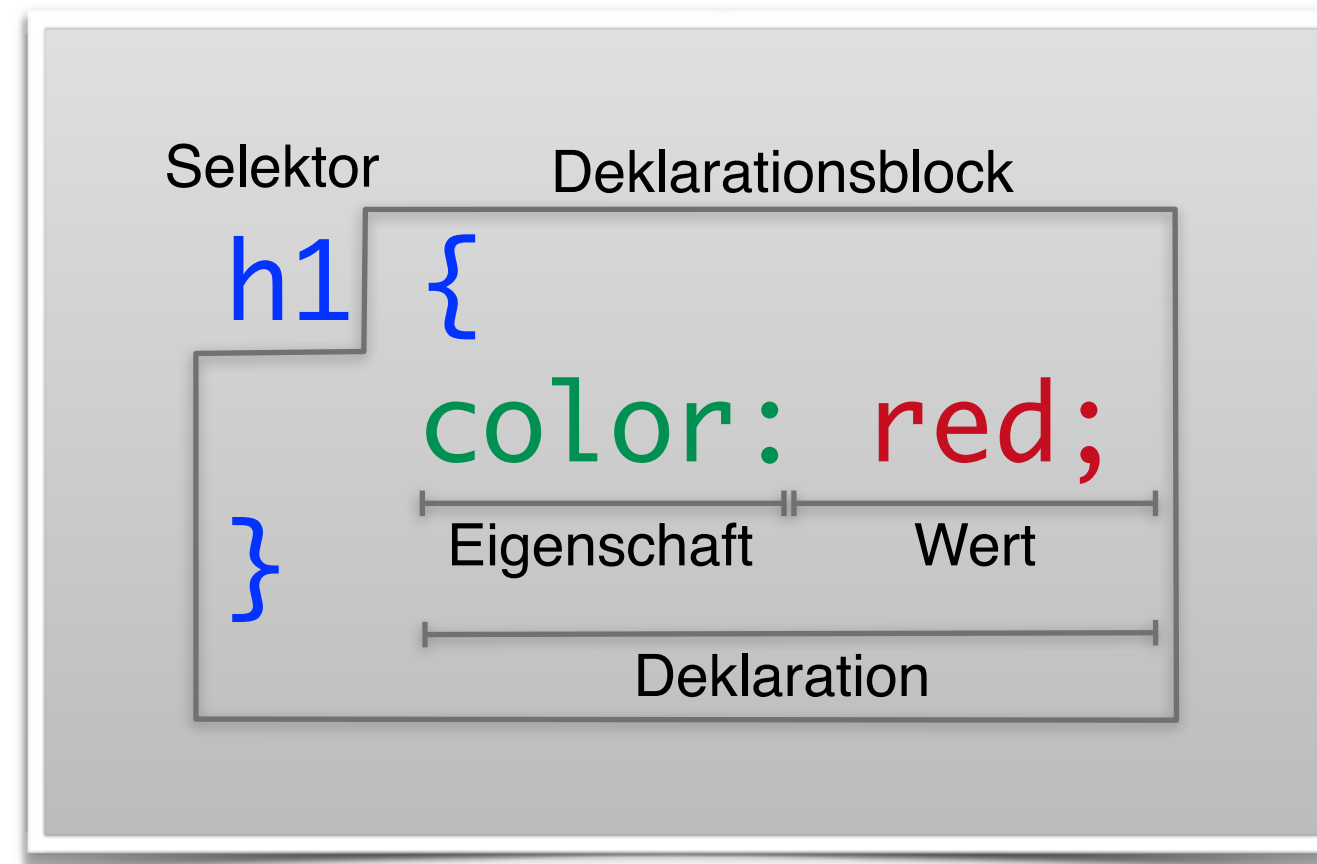
HTML-Elemente zur Einbindung von Tabellen

Element	Funktion	ab HTML 5
table	Definiert eine Tabelle zur Gliederung von tabellarischen Daten, umschließt alle benötigten Elemente (unter anderem tr, tth und td) zur Aufbereitung der Daten.	
caption	Definiert den Titel einer Tabelle.	
tbody	Definiert den eigentlichen Inhalt einer Tabelle. Ohne eine Auszeichnung durch thead, tfoot und tbody entsprechen alle Zeilen automatisch dem tbody.	
thead	Definiert eine oder mehrere Kopfzeilen einer Tabelle. Im Ausdruck wird der Tabellenkopf auf jedem weiteren Blatt erneut ausgedruckt.	
tfoot	Definiert eine oder mehrere Fußzeilen einer Tabelle und ist für Summierungen bzw. Zusammenfassungen vorgesehen. Falls eine Tabelle über <tbody>, <thead> und <tfoot> gegliedert ist, muss <tfoot> hinter dem <thead>-Tag und vor dem <tbody>-Tag angegeben werden. Bei der Druckausgabe wird der Wert von <tfoot> auf jedem einzelnen Blatt ausgegeben.	
tr	Definiert eine Zeile innerhalb einer Tabelle.	
th	Definiert eine einzelne Kopfzeile innerhalb einer Tabelle, welche eine Spalte oder Zeile betitelt.	
td	Definiert eine normale Datenzeile innerhalb einer Tabelle.	

mit Änderungen übernommen von: Heller, St. (2012) Workshop HTML 5 & CSS3, dpunkt-Verlag

2.5 CSS-Grundlagen

Selektoren und Deklarationsblöcke



- Jede CSS-Regel hat die folgende Struktur:

- Selektor
- Deklarationsblock

Selektor

- Bestimmt das oder die HTML-Elemente auf welche die CSS-Regel angewendet werden soll
- Hier: Element- oder Einfachselektor
- Ein Selektor kann aus einem einzelnen oder einer komplexen Zusammensetzung mehrerer Einfachselektoren bestehen

Deklarationsblock

- Enthält die Deklaration die auf die ausgewählten HTML-Elemente anzuwenden sind und wird durch ein Paar geschweifte Klammern umschlossen
- Eine Deklaration besteht aus folgenden Einzelteilen:
 - der CSS-Eigenschaft (Property),
 - einem Doppelpunkt,
 - einem oder mehreren Werten (Property Values) und
 - einem abschließenden Semikolon.

2.5 CSS-Grundlagen

3 Möglichkeiten der Verknüpfung von HTML mit CSS (I)

Möglichkeiten der Einbindung von CSS

- Style-Definition in separaten CSS-Dateien
- Style-Definition im Kopfbereich eines HTML-Dokuments
- Style-Definition innerhalb einzelner HTML-Elemente mittels style-Attributen

HTML-Syntax

```
<head>
  <!-- andere Definitionen im HTML-Kopfbereich --!>
  <link rel="stylesheet" type="text/css" href="stylesheet.css">
  <link rel="stylesheet" type="text/css" href="css/ie-korrekturen.css">
</head>
```

CSS-Syntax

```
<head>
  <!-- andere Definitionen im HTML-Kopfbereich --!>
  <style type="text/css">
    @import "stylesheet.css";
    @import url("css/ie-korrekturen.css");
  </style>
</head>
```

Style-Definition in separaten CSS-Dateien

- Einheitliche(s) Stylesheet(s) für alle Seiten einer Website
- Einbindung in jedes HTML-Dokument möglich
- Änderungen von CSS-Eigenschaften wirken sich auf alle einbindenden HTML-Dokumente aus
- Zwei Möglichkeiten der Einbindung: per HTML-Syntax oder per CSS-Syntax
- Einbindung mehrerer CSS-Dateien
- Vermischung mit CSS-Definitionen im Kopfbereich oder in style-Attributen möglich

2.5 CSS-Grundlagen

3 Möglichkeiten der Verknüpfung von HTML mit CSS (II)

```
<head>
  <!-- andere Definitionen im HTML-Kopfbereich --!>
  <style type="text/css">
    h1 {
      font-family: arial, helvetica, sans-serif;
      font-weight: normal;
      color: #dd9900;
      line-height: 1.3em;
      padding-bottom: 9px;
      border-bottom: dotted 3px #dd9900;}
  </style>
</head>
<body>
  <h1>Das style-Element</h1>
  <!-- weitere Inhalte --!>
</body>
```

```
<body>
  <h1>Die Seite mit dem besonderen Element</h1>
  <p>Das folgende Element soll etwas Besonderes sein:</p>
  <p style="background-color: #808040; color: #d8fd02; font-family:
    'Century Schoolbook', serif; font-size: 2em; letter-spacing: 3px;
    padding: 40px; border: double #d8fd02 4px;" title="Zitat von Francis
    Picabia">Unser Kopf ist rund, damit das Denken die Richtung wechseln
    kann!</p>
  <!-- weitere Inhalte --!>
</body>
```

Style-Definition im Kopfbereich eines HTML-Dokuments

- Styles für die verschiedenen HTML-Elemente einer Webseite werden im Header des Dokuments mittels <style>-Element eingefügt
- Innerhalb des Wertebereichs des <style>-Elements erfolgt Definition in CSS-Syntax
- Bevorzugte Methode wenn Gestaltung von HTML-Elementen per CSS nur in Ausnahmefällen erfolgt oder wenn es wenige von einer für die ganze Website gültigen Definition abweichende Elemente auf einer einzigen Webseite gibt

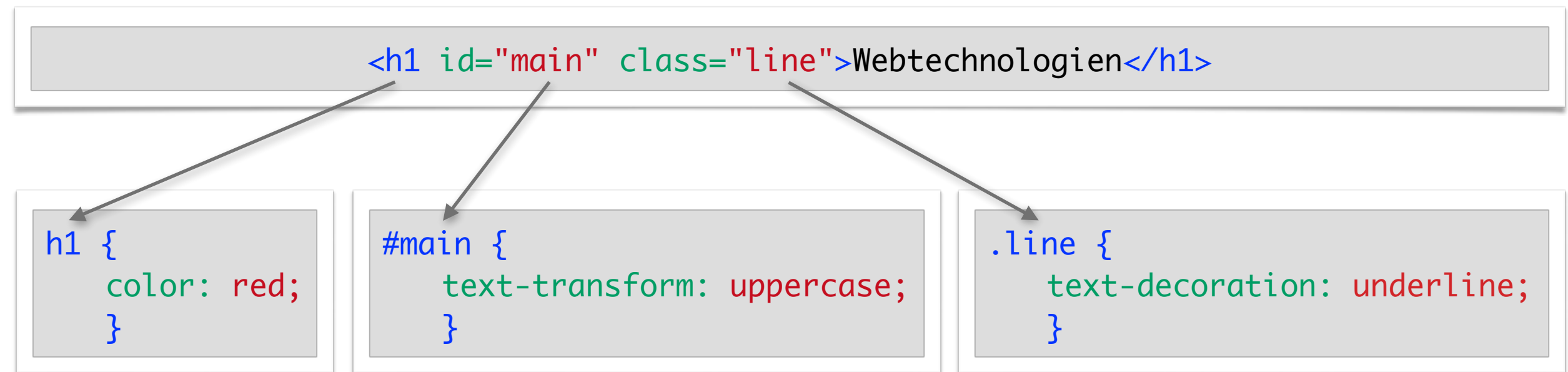
Definition mittels style-Attribut

- Formatierung einzelner Elemente innerhalb eines Dokuments mit Hilfe des style-Attributs
- Innerhalb des Wertebereichs des style-Attributs erfolgt Definition in CSS-Syntax
- Nur in Ausnahmefällen zu bevorzugen - Mischung von HTML und CSS schwer verständlich und lesbar

2.5 CSS-Grundlagen

Übersicht Selektoren

Element, ID- und Klassenselektor greifen jeweils auf entsprechende HTML-Elemente beziehungsweise Attribute zu



Verschiedenen CSS-Eigenschaften sowie Kurzschreibweisen (z.B. border) können mehrer Werte zugewiesen werden, jeweils durch Leerzeichen voneinander getrennt

```
#kopf img.logo {  
  border: 2px solid lightgrey;  
  position: relative;  
  left: 20px;  
}
```

Einigen Eigenschaften können neben verschiedenen Werten auch mehrere Wertblöcke zugewiesen werden, jeweils durch Komma voneinander getrennt

```
#content div {  
  box-shadow: blue 20px 20px 2px,  
              red -10px 30px 2px;  
}
```

2.6 CSS-Eigenschaften

Das Box-Modell

Grundlage für die Darstellung eines HTML-Elements

margin

- Bestimmt den *Außenabstand* und damit den Zwischenraum zum umliegenden Elementen
- Hintergrundfarben und -grafiken die dem Element selbst zugewiesen wurden sind dort nicht zu sehen
- Falls der Abstand mit einer Farbe oder Grafik gefüllt ist, stammt diese von einem umgebenden Element

border

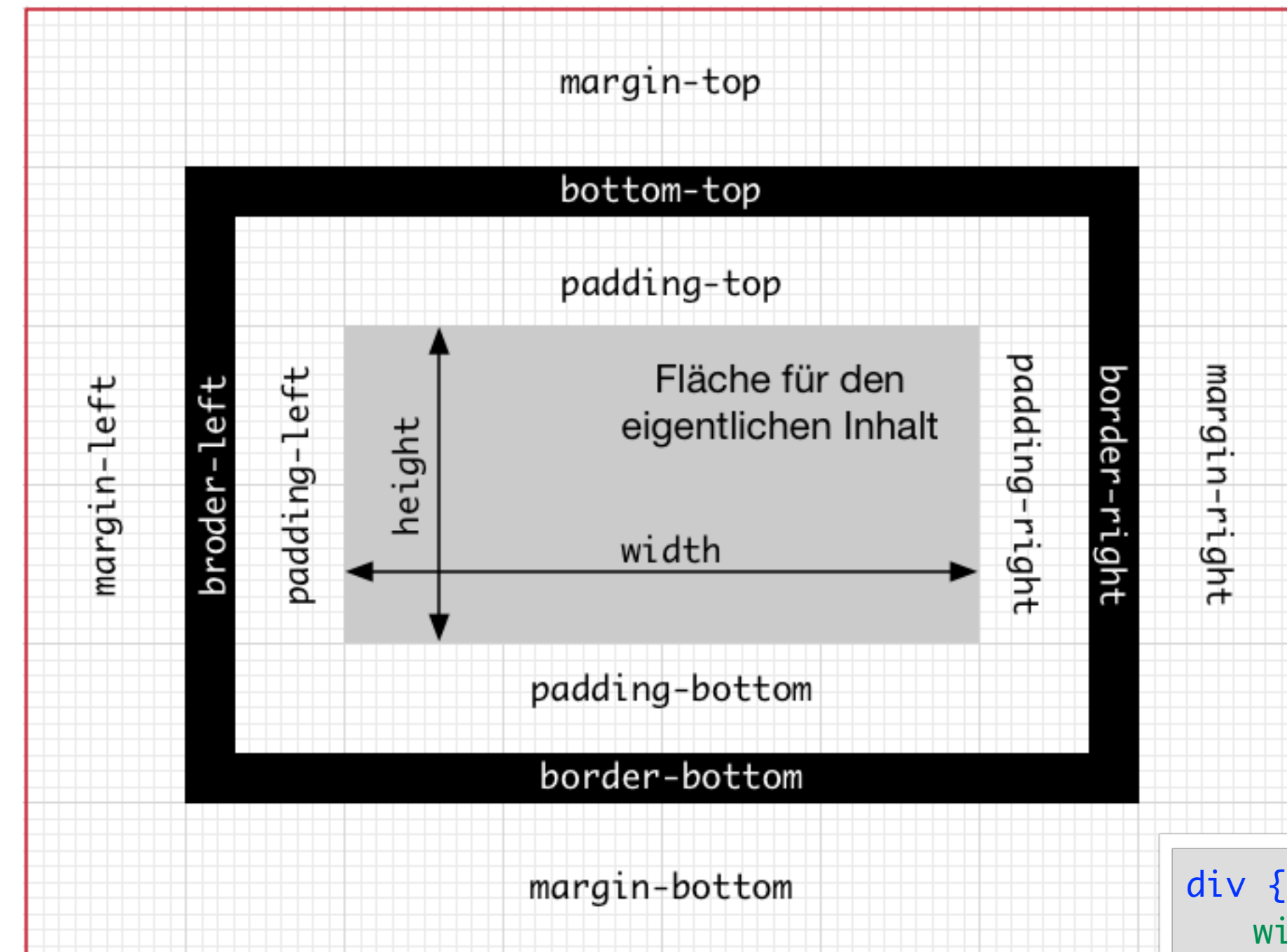
- Rahmen mit seiner Breite und Farbe

padding

- Umgibt den eigentlichen Inhabebereich mit einer eigenen Fläche (*Innenabstand*)
- Hintergrundfarben und Grafiken werden bis zum Rahmen dargestellt
- Text und innenliegende HTML-Elemente (auch Grafiken in HTML) werden standardmäßig nicht im Innenabstand angezeigt

width und height

- Dimensionen eines Elements ohne padding, border und margin



```
div {  
  width: 40px;  
  height: 20px;  
  margin: 10px;  
  border-width: 3px;  
  padding: 7px;  
}
```

2.6 CSS-Eigenschaften

Module (I)

Module

- CSS definiert den Sprachumfang (insbesondere CSS-Eigenschaften), aber auch generelle Syntax und Wertetypen in Modulen
- Module nehmen den CSS2.1-Standard auf und erweitern ihn
- Jedes Modul ist eine eigenständige W3-Spezifikation mit eigener Dynamik bei der Entwicklung und eigenem Review-Prozess

Modulname	Beschreibung	Link zum Standard
Animations	<ul style="list-style-type: none">• Ermöglicht einfache Animationen ohne JavaScript• Einführung einer eigenen DOM-Schnittstelle für erweitertes Event-Handling	http://www.w3.org/TR/css3-animations/
Backgrounds and Borders	<ul style="list-style-type: none">• Zuständig für Hintergrund und Rahmen• Skalieren von Hintergrundbildern• Definition abgerundeter Ecken• Grafische Schmuckrahmen	http://www.w3.org/TR/css3-background/

Modulname	Beschreibung	Link zum Standard
Basic User Interface	<ul style="list-style-type: none">• Gestaltung interaktiver Elemente wie Formulare, Hyperlinks oder der Cursor• Interaktive Hervorhebung von Elementen mit outline• Optische Unterscheidung gültiger und ungültiger Inhalte	http://www.w3.org/TR/css3-background/
Basic Box Model	<ul style="list-style-type: none">• Definition des grundlegenden Box-Modells zur Festlegung von Innen- und Außenabständen, Rahmenstyles und -stärken usw.	http://www.w3.org/TR/css3-box
Cascading and Inheritance	<ul style="list-style-type: none">• Prinzipien von Kaskadierung und Vererbung von CSS-Eigenschaften an Kindelemente	http://www.w3.org/TR/css3-cascade
Color	<ul style="list-style-type: none">• Festlegung von Farb- und Transparenzeigenschaften• Neue Möglichkeiten der Farbdefinition	http://www.w3.org/TR/css3-color
Flexible Box Layout	<ul style="list-style-type: none">• Neue Möglichkeiten der Anordnung von Elementen	http://www.w3.org/TR/css3-flexbox
Fonts	<ul style="list-style-type: none">• Eigenschaften für Schriftarten	http://dev.w3.org/csswg/css3-fonts
Generated Content for Paged Media	<ul style="list-style-type: none">• Gestaltung von Printlayouts und seitenorientierten Ausgaben• Laufende Kopf- und Fußzeilen mit automatisiertem Überschriftenbezug• Seiten- und Kapitelnummerierung, Fußnotenautomatik, usw.	http://www.w3.org/TR/css3-gcpm

2.6 CSS-Eigenschaften

Module (II)

mit Änderungen übernommen von: Gull, C.; Münz, St. (2014) HTML 5 Handbuch, Franzis, 10. Auflage

Modulname	Beschreibung	Link zum Standard
Generated and Replaced Content	<ul style="list-style-type: none">• Dynamische Anpassung von Inhalten• Verschiebung von Elementen innerhalb eines HTML-Dokuments	http://www.w3.org/TR/css3-content
Grid Positioning	<ul style="list-style-type: none">• Erweiterte Konzepte zur rasterbasierten Positionierung von Elementen	http://www.w3.org/TR/css3-grid
Hyperlink Presentation	<ul style="list-style-type: none">• Erweiterung des target-Attributes von HTML zur neuen Definition von Zielen für externe Verweise	http://www.w3.org/TR/css3-hyperlinks
Image Values and Replaced Content	<ul style="list-style-type: none">• Erweiterte Spezifikation von Bildformaten, zum Beispiel Auflösung, alternative Farben, etc.• Anzeige bestimmter Bildausschnitte	http://www.w3.org/TR/css3-images
Line Box	<ul style="list-style-type: none">• Neue Eigenschaften für die Kontrolle von Eigenschaften einer Zeile• Berücksichtigung typografischer Basislinien	http://www.w3.org/TR/css3-linebox
Lists and Counters	<ul style="list-style-type: none">• Gestaltung von Printlayouts und seitenorientierten Ausgaben• Laufende Kopf- und Fußzeilen mit automatisiertem Überschriftenbezug• Seiten- und Kapitelnummerierung, Fußnotenautomatik, usw.	http://www.w3.org/TR/css3-gcpm
Generated and Replaced Content	<ul style="list-style-type: none">• Listeneigenschaften• Formatierung von Aufzählungszeichen und Nummerierungszahlen	http://www.w3.org/TR/css3-lists

Modulname	Beschreibung	Link zum Standard
Marqueee	<ul style="list-style-type: none">• Definition von animiertem Lauftext	http://www.w3.org/TR/css3-marquee
Multi-column Layout	<ul style="list-style-type: none">• Mehrspaltiger Textfluss mit automatischem Zeilenumbruch• Definition von Spalten, Spaltenbreiten, Abständen, Verhaltensweisen, usw.	http://www.w3.org/TR/css3-multicol
Namespaces	<ul style="list-style-type: none">• Definition von Namensräumen analog zu XML	http://www.w3.org/TR/css3-namespace
Paged Media	<ul style="list-style-type: none">• Printlayouts und seitenorientierte Ausgabe• Definition der Seitengröße, Unterscheidung linke Seite rechte Seite• Seitenumbruchkontrolle	http://www.w3.org/TR/css3-page
Presentation Levels	<ul style="list-style-type: none">• Laufnummern oder Nummern von Hierarchieebenen für HTML-Elemente• Benötigt für Gliederungsansicht oder Diashow	http://www.w3.org/TR/css3-preslev
Ruby	<ul style="list-style-type: none">• Ruby: eine im fernöstlichen Sprachen vorkommende Notation um Schriftzeichen mit zusätzlichen Informationen auszustatten, zur Präzisierung der im Kontext gemeinten Bedeutung• Positionierung von Ruby-Annotationen mit CSS-Eigenschaften	http://www.w3.org/TR/css3-ruby
Speech Module	<ul style="list-style-type: none">• Eigenschaften zur Steuerung von Sprachsynthesizern	http://www.w3.org/TR/css3-speech

2.6 CSS-Eigenschaften

Module (III)

mit Änderungen übernommen von: Gull, C.; Münz, St. (2014) HTML 5 Handbuch, Franzis, 10. Auflage

Modulname	Beschreibung	Link zum Standard
Syntax	<ul style="list-style-type: none">• Beschreibt die allgemeine Syntax, Grammatik und das Fachvokabular von CSS	http://www.w3.org/TR/css3-syntax
Text	<ul style="list-style-type: none">• Eigenschaften zur Textkontrolle• Textumbruchkontrolle• Grafische Schrifteffekte	http://www.w3.org/TR/css3-text
Template Layout	<ul style="list-style-type: none">• Definition komplexer Webseitenlayouts• Unterteilung des Anzeigenbereichs in Regionen	http://www.w3.org/TR/css3-layout
2D Transforms, 3D Transforms	<ul style="list-style-type: none">• 2- und 3-dimensionale Drehung und Dehnung von Texten	http://www.w3.org/TR/css3-2d-transforms http://www.w3.org/TR/css3-3d-transforms
Transitions	<ul style="list-style-type: none">• Definition von Übergängen zwischen verschiedenen Werten einer visuellen CSS-Eigenschaft	http://www.w3.org/TR/css3-transitions
Values and Units	<ul style="list-style-type: none">• Beschreibung der Wertetypen für Eigenschaften und Maßeinheiten	http://www.w3.org/TR/css3-values

2.6 CSS-Eigenschaften Datentypen (I)

Schlüsselwörter

```
text-transform: uppercase;
```

Ganzzahl

```
z-index: 10;  
orphans: 4;
```

Prozentangabe

```
font-size: 75%;  
padding: 5%;
```

Fließkomma

```
width: 4.25em;  
opacity: 40.5;
```

Längenangaben

```
height: 400px;  
width: 650PX;  
margin: 0.25em;  
padding: 0.4EM;
```

Relative Längenangaben

```
font-size: 2em;  
font-size: 4ex;  
font-size: 5ch;  
font-size: 2rem;
```

Absolute Längenangaben

```
width: 10cm;  
width: 150mm;  
width: 2in;  
width: 450px;  
width: 400pt;  
width: 20pc;
```

- Jede Deklaration besteht aus einer Eigenschaft und einem oder mehreren Werten
- CSS3 definiert über 300 Eigenschaften (CSS2.1 hatte 115 Eigenschaften)
- Unterschiedliche Arten von Datentypen für Eigenschaftswerte

Datentypen

- Schlüsselwörter
- Numerische Datentypen
 - Ganzzahl
 - Fließkommazahl
 - Prozentangabe
- Längenangaben
 - Relative Längenangaben
 - Absolute Längenangaben
- Farbangaben
 - Farbnamen
 - transparent
 - Hexadezimale Farbangaben
 - rgb() und rgba() Farbangaben
 - hsl() und hsla() Farbangaben

2.6 CSS-Eigenschaften

Datentypen (II)

Abkürzung	Angabetyp	Bedeutung	Beispiel
%	relativ	Prozent. Je nach CSS-Eigenschaft relativ zur elementeigenen Größe oder zu der des Elternelements oder zu einem allgemeinen Kontext. Bei nebenstehendem Beispiel ist die Angabe zur Zeilenhöhe relativ zur Schriftgröße zu interpretieren.	line-height: 120%;
cm	absolut	Zentimeter. Allgemeines Längenmaß. 100. Teil des Urmeters.	top: 2.54cm;
em	relativ	Höhe des M der definierten Schrift. Ausnahme: Wenn die font-size-Eigenschaft mit dieser Maßangabe versehen wird, steht diese bezogen auf die Höhe des Großbuchstabens M im Elternelement.	font-size: 1.2em;
ex	relativ	Höhe des x der definierten Schrift. Ausnahme: Wenn die font-size-Eigenschaft mit dieser Maßangabe versehen wird, steht diese bezogen auf die Höhe des Kleinbuchstabens x im Elternelement.	font-size: 1.3ex;
in	absolut	Inch. Allgemeines Längenmaß im angelsächsischen Raum. 1 Inch (Zoll) entspricht 2,54 Zentimetern.	border-width: 0.1in;
mm	relativ	Millimeter. Allgemeines Längenmaß. 1000. Teil des Urmeters.	margin-bottom: 10mm;
pc	absolut	Pica. Typografische Maßeinheit. 1 Pica entspricht 12 Punkt.	line-height: 1.2pc;
pt	absolut	Punkt. Typografische Maßeinheit. 1 Punkt entspricht 1/72 Inches (Zoll).	font-size: 12pt;
px	absolut/ relativ	Pixel. Abhängig von der Pixeldichte des Ausgabegerätes, relativ also von Ausgabegerät zu Ausgabegerät, absolut dagegen auf ein und dasselbe Ausgabegerät bezogen.	border-width: 3px;

mit Änderungen übernommen von: Gull, C.; Münz, St. (2014) HTML 5 Handbuch, Franzis, 10. Auflage

2.6 CSS-Eigenschaften Farbangaben

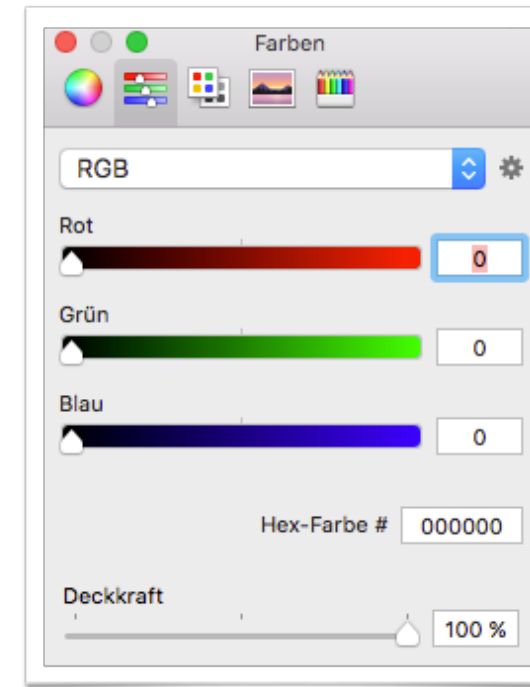
RGB

reiner rgb-Wert

`rgba(100, 25, 255, 0.8)`

rot grün blau Transparenz

```
background-color: rgb(51,0,102);  
border-left: dotted 5px rgba(60%,90%,75%,0.8);
```



Hexadezimal

`#ff5e99`

rot grün blau

```
color: #ff0000; /* rot */  
color: #008000; /* grün */  
color: #0000ff; /* blau */  
color: #ffff00; /* gelb */
```

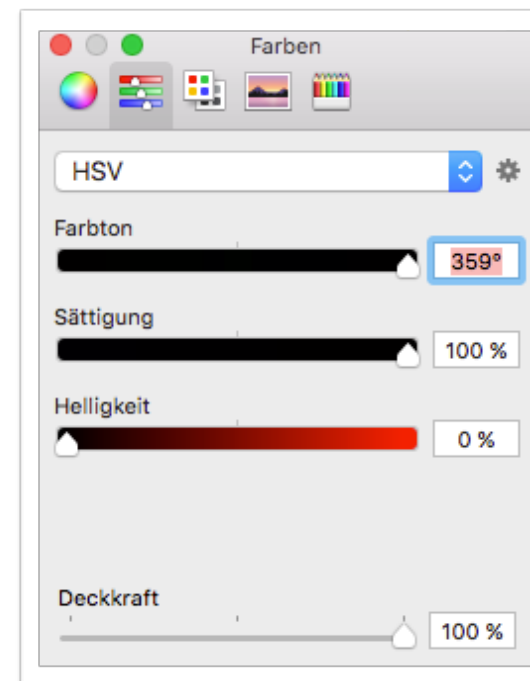
HSL

reiner hsl-Wert

`hsla(100, 100%, 50%, 0.9)`

Farbwert Sättigung Helligkeit Transparenz

```
background-color: hsl(50,30%,10%);  
border-left: dotted 5px hsla(60,90%,75%,0.8);
```



Farbnamen

aliceblue, antiquewhite, aquamarine, azure, beige, blueviolet, brown, burlywood, cadetblue, chartreuse, chocolate, coral, cornflowerblue, cornsilk, crimson, darkblue, darkcyan, darkgoldenrod, darkgray, darkgreen, darkkhaki, darkmagenta, darkolivegreen, darkorange, darkorchid, darkred, darksalmon, darkseagreen, darkslateblue, darkslategray, darkturquoise, darkviolet, deeppink, deepskyblue, dimgray, dodgerblue, firebrick, floralwhite, forestgreen, gainsboro, ghostwhite, gold, goldenrod, greenyellow, honeydew, hotpink, indianred, indigo, ivory, khaki, lavender, lavenderblush, lawngreen, lemonchiffon, lightblue, lightcoral, lightcyan, lightgoldenrodyellow, lightgreen, lightgrey, lightpink, lightsalmon, lightseagreen, lightskyblue, lightslategray, lightsteelblue, lightyellow, limegreen, linen, mediumaquamarine, mediumblue, mediumorchid, mediumpurple, mediumseagreen, mediumslateblue, mediumspringgreen, medianturquoise, mediumvioletred, midnightblue, mintcream, mistyrose, moccasin, navajowhite, oldlace, olivedrab, orange, orangered, orchid, palegoldenrod, palegreen, paleturquoise, palevioletred, papayawhip, peachpuff, peru, pink, plum, powderblue, rosybrown, royalblue, saddlebrown, salmon, sandybrown, seagreen, seashell, sienna, skyblue, slateblue, slategray, snow, springgreen, steelblue, tan, thistle, tomato, turquoise, violet, wheat, whitesmoke, yellowgreen

2.6 CSS-Eigenschaften

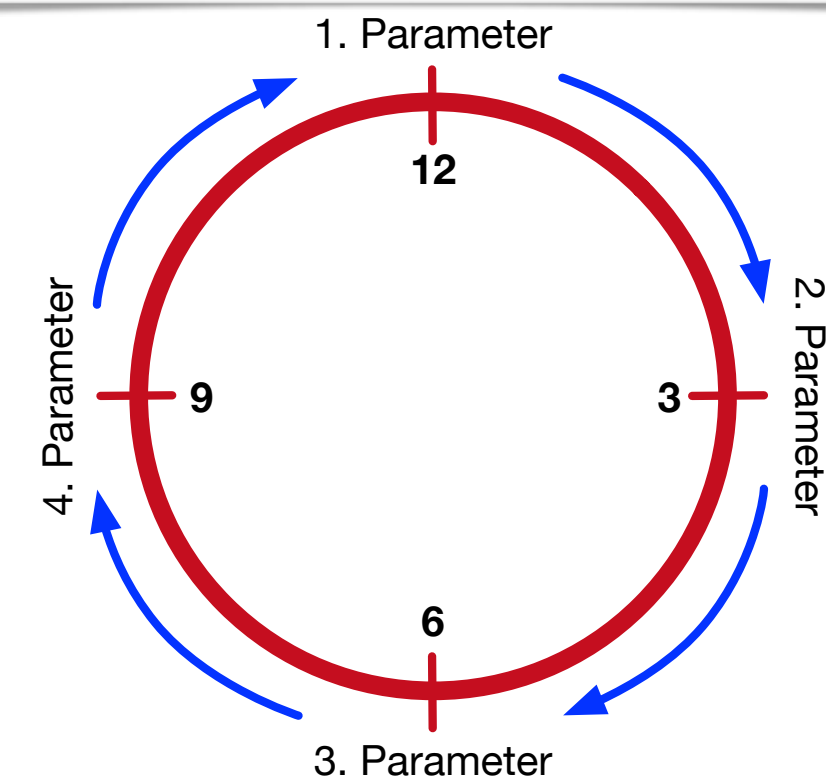
Kurzschreibweisen (I)

- Kurzschreibweise einzelner Parameter für oben, rechts, unten und links
- Anwendbar zum Beispiel auf margin und padding
- Automatische Vervollständigung beim Auslassen einzelner Werte gemäß nebenstehendem Schema

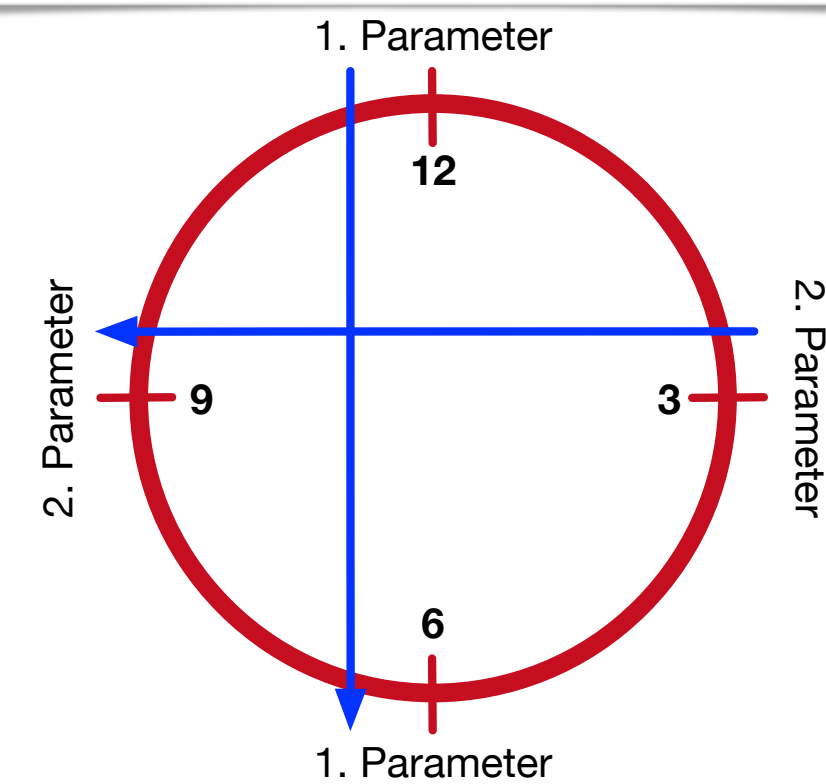
```
div {  
  padding-top: 20px;  
  padding-right: 50px;  
  padding-bottom: 20px;  
  padding-left: 10px;  
}
```

```
div {  
  padding: 20px 50px 20px 10px;  
}
```

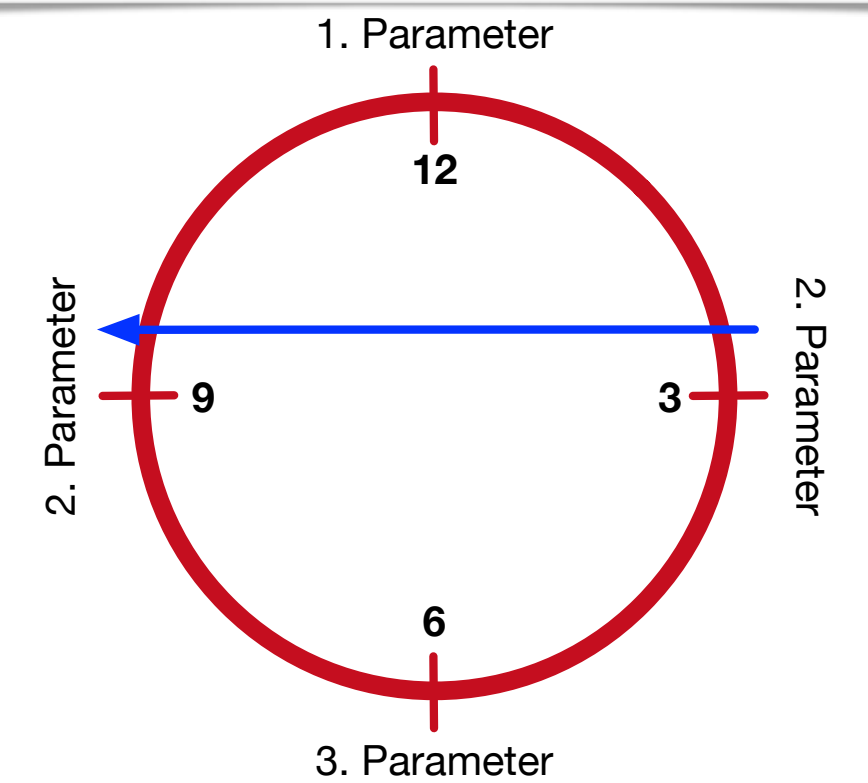
```
padding: 20px 50px 20px 10px;
```



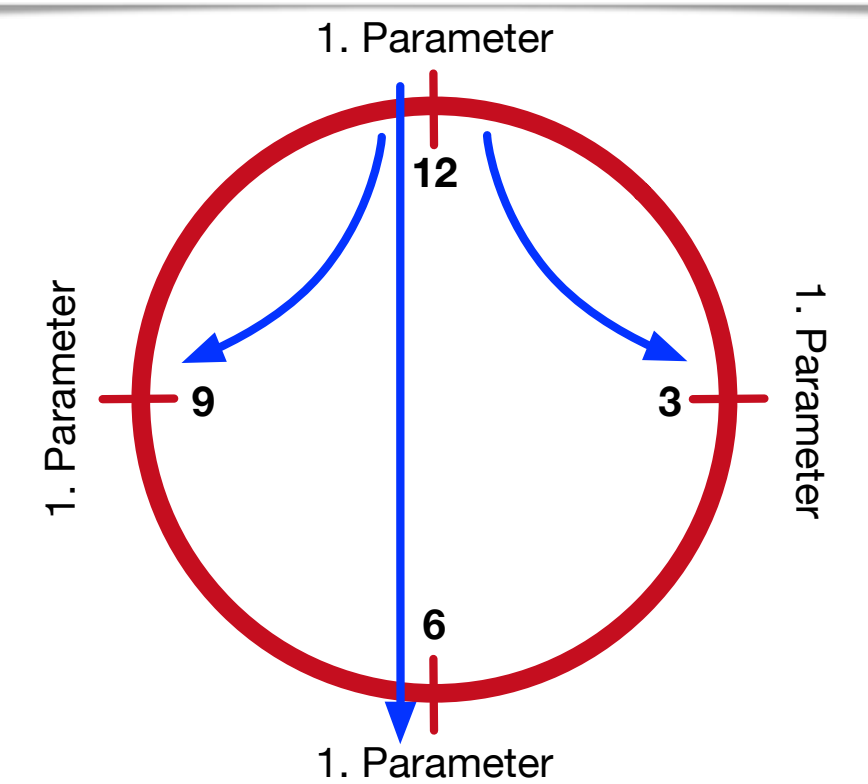
```
padding: 10px 20px;
```



```
padding: 10px 20px 40px;
```



```
padding: 10px;
```



2.6 CSS-Eigenschaften

Kurzschreibweisen (II)

Kurzschreibweisen fassen verschiedene Eigenschaften in einem Ausdruck zusammen

```
ul {  
  list-style-type: square;  
  list-style-position: outside;  
  list-style-image: url('bullet.png');  
}
```

```
ul {  
  list-style: square outside url('bullet.png');  
}
```

Die Reihenfolge der einzelnen Parameter ist in der Regel frei wählbar

```
h1 {  
  border-style: solid;  
  border-width: thin;  
  border-color: #ff0000;  
}
```

```
h1 {  
  border: thin solid #ff0000;  
}
```

Schlechtes Beispiel: Reihenfolge für font ist nicht frei wählbar und Spezifikation von font-height durch /16px ist in der Kurzschreibweise nicht sehr intuitiv

```
p {  
  font-style: italic;  
  font-variant: small-caps;  
  font-weight: 700;  
  font-size: 12px;  
  line-height: 16px;  
  font-family: Arial, sans-serif;  
}
```

```
p {  
  font: italic small-caps 700 12px/16px Arial, sans-serif;  
}
```

2.7 CSS-Elementfluss

Block-Elemente versus Inline-Elemente



```
<h1 style="background-color: silver; color: black;">
  Ein schöner Tag
</h1>
```



```
<p>
  An einem <em style="background-color: silver; color:
    black;">wirklich schönen Tag</em> im September...
</p>
```

Block-Elemente

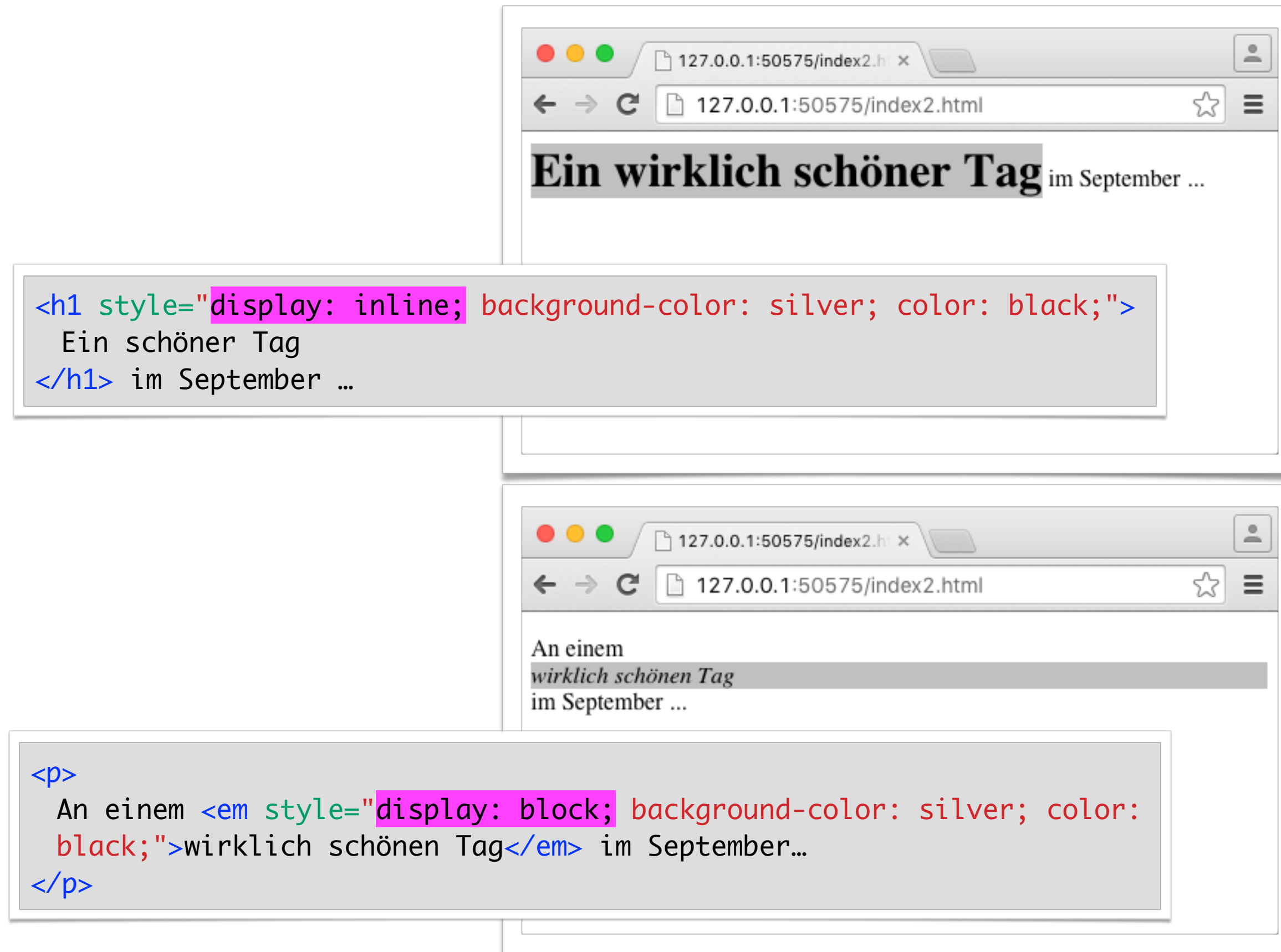
- Nehmen - wenn nicht anders erzwungen - in der Breite so viel Raum wie möglich ein und in der Höhe soviel Raum wie erforderlich
- Erzeugen eine neue Zeile im Elementfluss
- Nachfolgende Elemente beginnen ebenfalls in einer neuen Reihe
- Können mit Innen/Außenabstand und Rahmen versehen werden
- Liste aller Block-Elemente: http://html5-webdesign.de/Liste_der_HTML_5_Block-Elemente.html

Inline-Elemente

- Nehmen - wenn nicht anders erzwungen - sowohl in der Breite als auch in der Höhe nur soviel Raum wie erforderlich ein
- Mehrere Inline-Elemente werden wie Wörter hintereinander aufgereiht
- Ist die verfügbare Breite durch mehrere Inline-Elemente erreicht, brechen sie in der nächsten Zeile um
- Können mit Innen/Außenabstand und Rahmen versehen werden
- Breite einer Inline-Box wird allein durch den Inhalt bestimmt (width ist wirkungslos)
- Höhe wird in der Regel durch die Schriftgröße bestimmt (height ist wirkungslos)
- Liste aller Inline-Elemente: http://html5-webdesign.de/Liste_der_HTML_5_Inline-Elemente.html

2.7 CSS-Elementfluss

Umwandlung von Block- in Inline-Elemente und umgekehrt

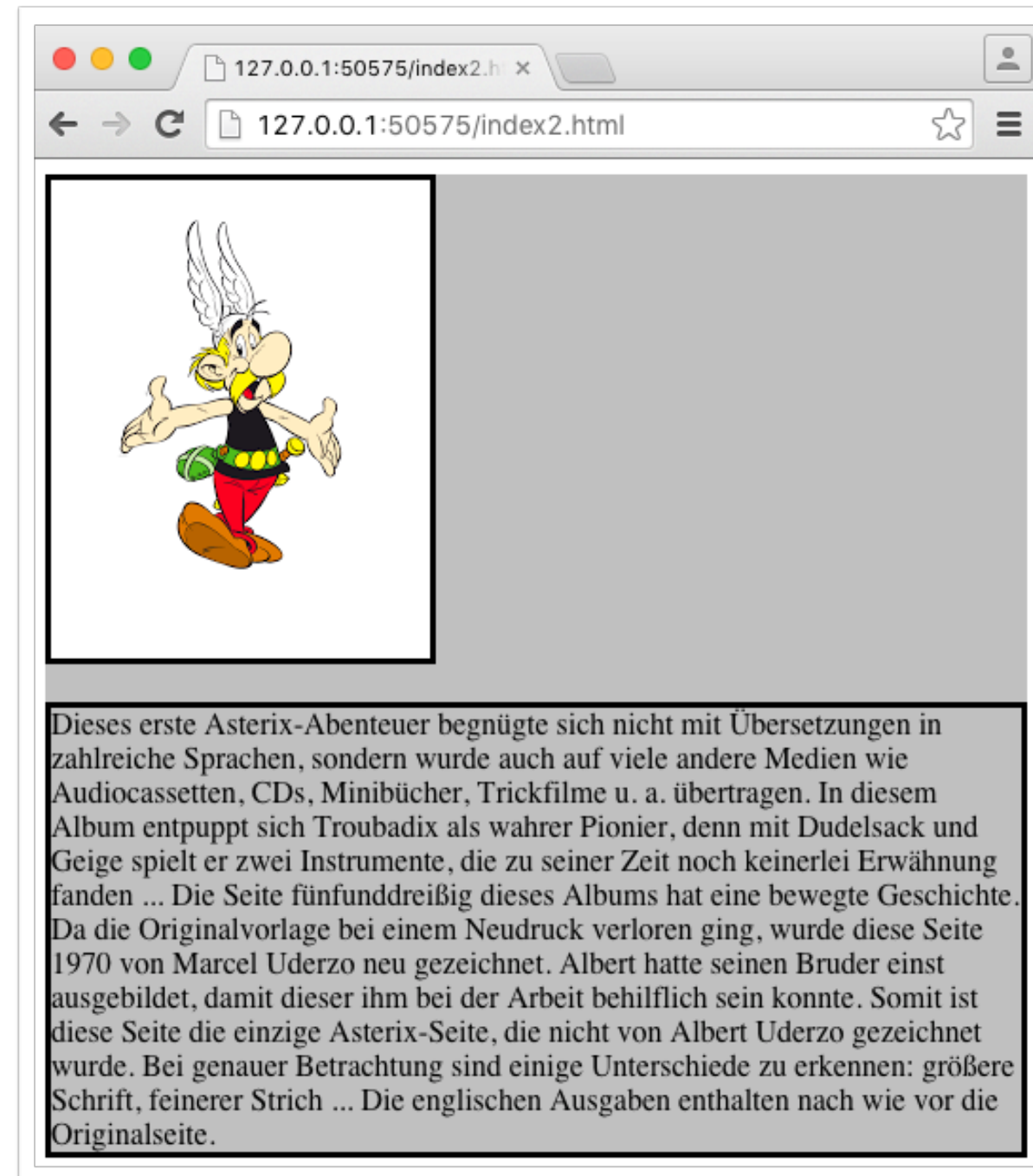


display

- Mit der `display`-Eigenschaft können aus Block-Elementen Inline-Elemente gemacht werden und umgekehrt

2.7 CSS-Elementfluss

Floats (I)



```
<div style="background-color: silver;">
  
  <p style="border: solid;">Die ersten Asterix-
    Abenteuer begnügen sich ... </p>
</div>
```

float

- Verschiebt ein Element (*Float*) an die linke oder rechte Innenkante seines Elternelements
- none: Standardwert; das Element wird nicht verschoben
- left: das Element wird nach links verschoben
- right: das Element wird nach rechts verschoben
- inherit: der Wert der float-Eigenschaft des Elternelements wird geerbt
- Außenkante eines nach link gefloateten Elements ist identisch mit der linken Innenkante des Elternelements (gilt entsprechend für rechts gefloatete Elemente)
- Elternelement eines Floats wird diesen nicht umschließen, d.h. Floats können aus ihrem Elternelement herausragen
- Blockelemente die neben einem Float dargestellt werden, nehmen ohne Berücksichtigung des Floats die volle Breite des Elternelements ein - sie werden teilweise von dem floatendem Element verdeckt
- Floatende Elemente verschieben jedoch Inline-Elemente - Texte, Bilder oder andere Inline-Elemente werden soweit verkürzt, dass sie um die Float-Box herumfließen können
- Ist nicht ausreichend Platz vorhanden, so werden der Inhalt und alle nachfolgenden Elemente sowie das floatende Element verschoben

2.7 CSS-Elementfluss

Floats (II)



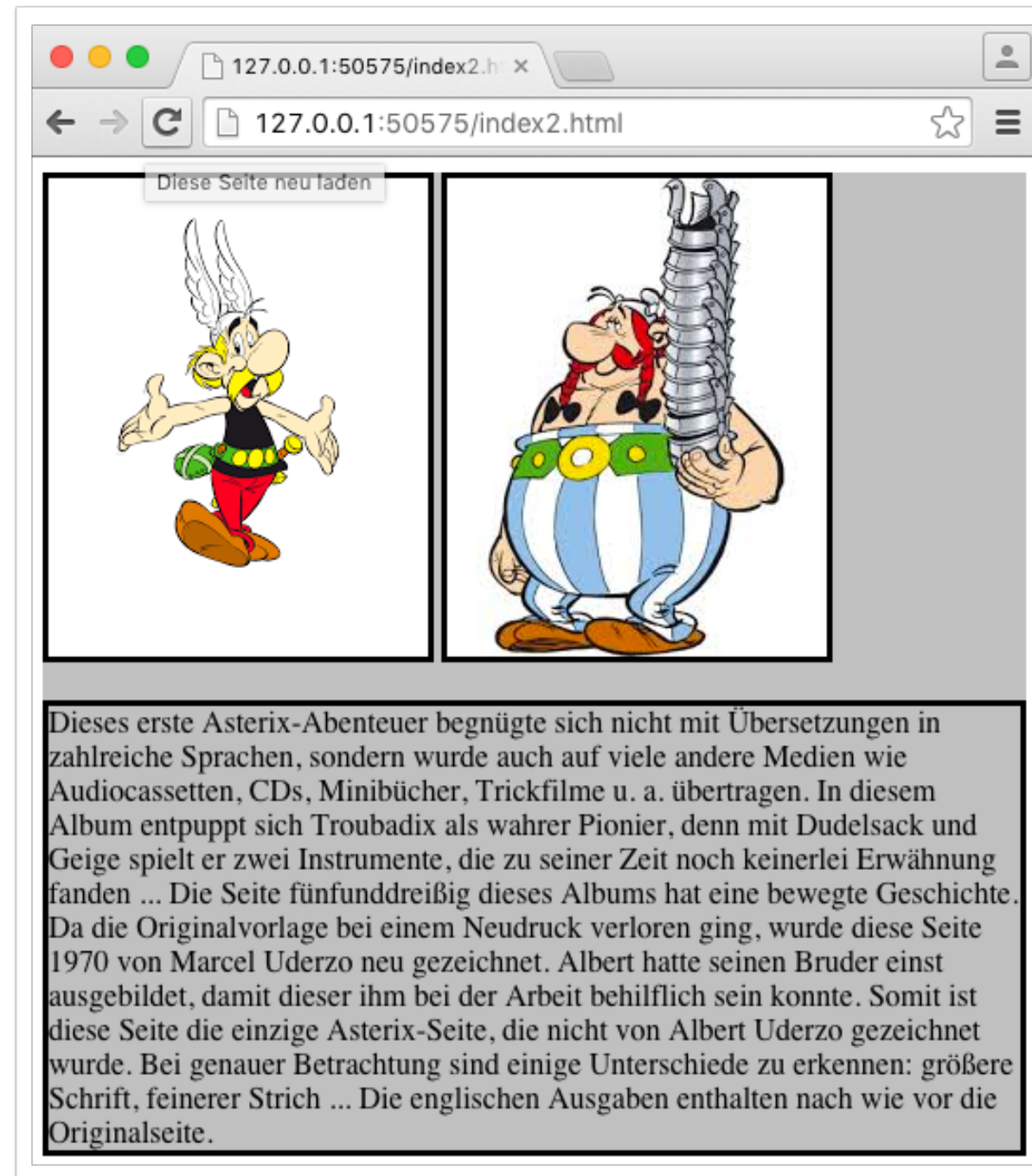
```
<div style="background-color: silver;">
  
  <p style="border: solid;">Die ersten Asterix-Abenteuer begnügen
    sich ... </p>
</div>
```



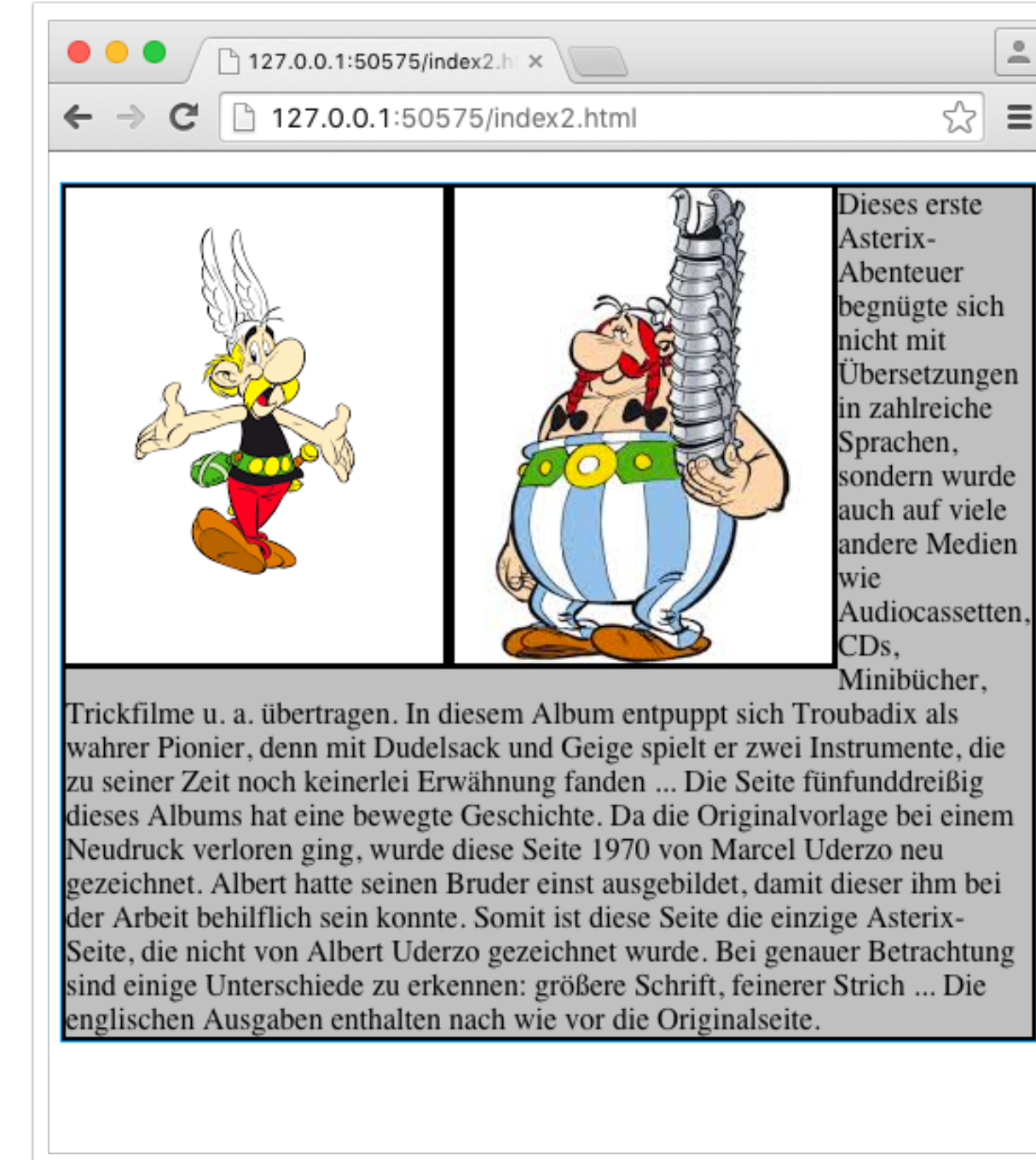
```
<div style="background-color: silver;">
  
  <p style="border: solid;">Die ersten Asterix-Abenteuer begnügen
    sich ... </p>
</div>
```


2.7 CSS-Elementfluss

Floats (III)



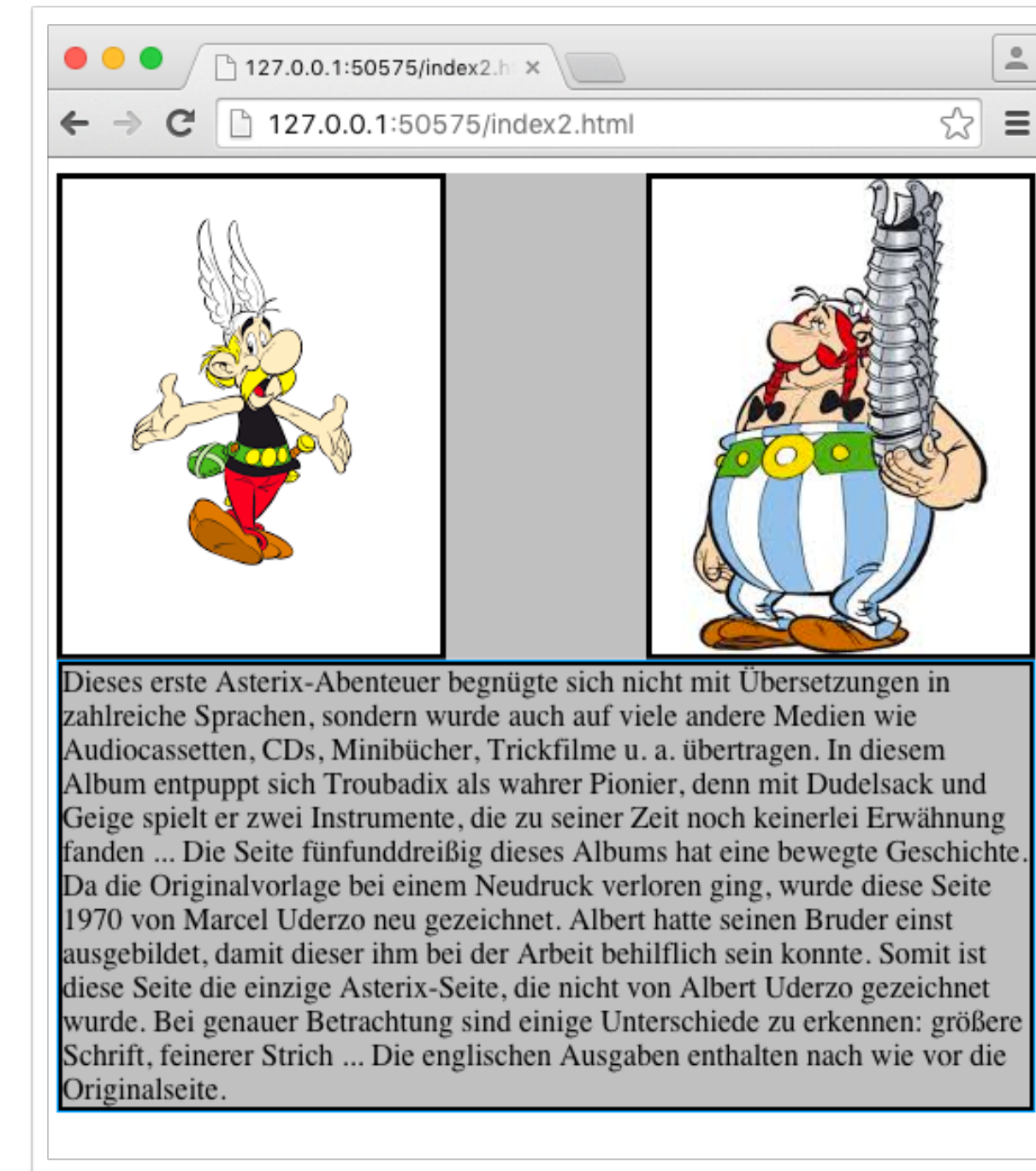
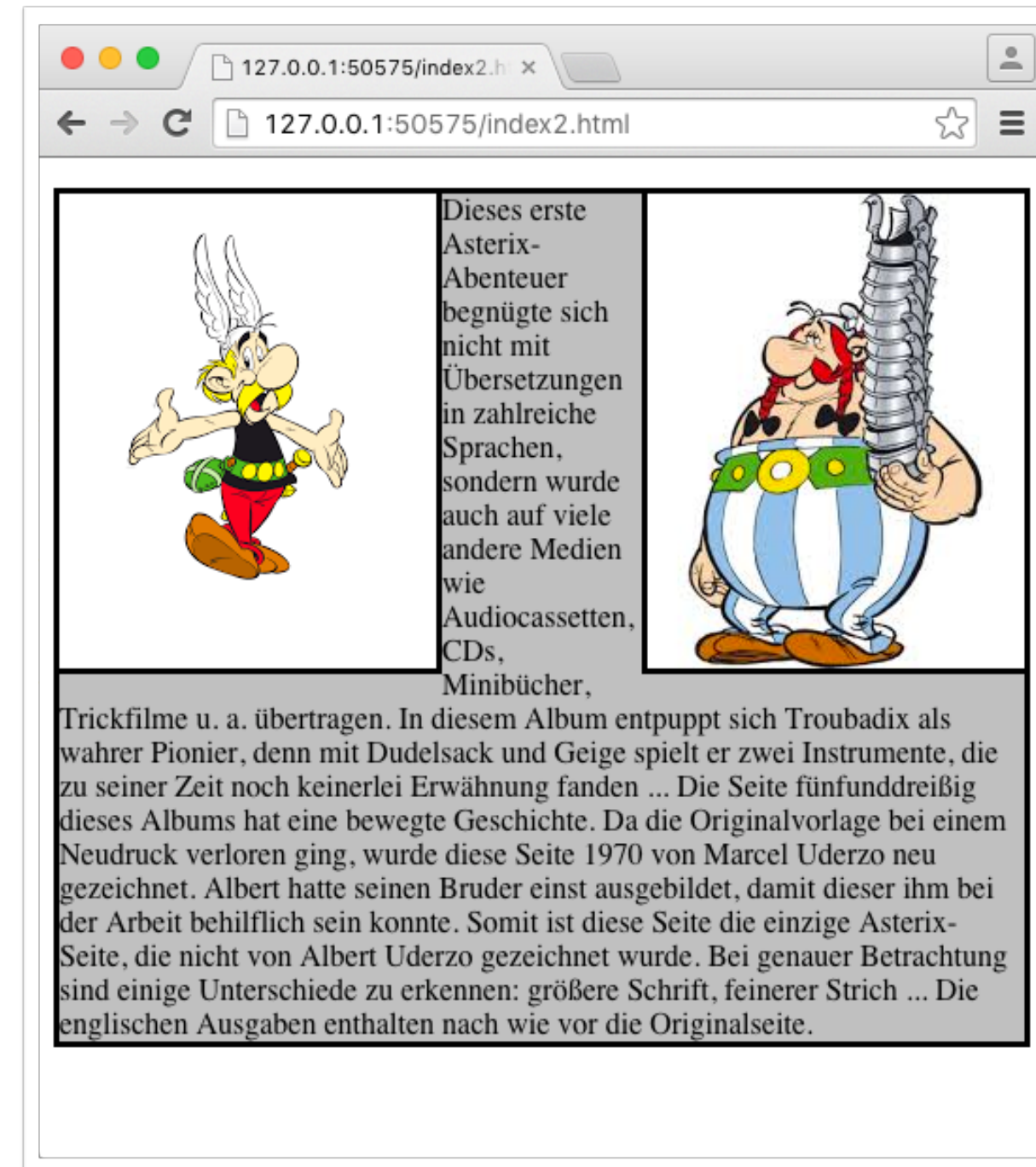
```
<div style="background-color: silver;">
  
  
  <p style="border: solid;">Die ersten Asterix-Abenteuer begnügen
    sich ... </p>
</div>
```



```
<div style="background-color: silver;">
  
  
  <p style="border: solid;">Die ersten Asterix-Abenteuer begnügen
    sich ... </p>
</div>
```


2.7 CSS-Elementfluss

Floats (IV)



clear

- Legt fest auf welcher Seite ein Blockelement nicht von einem Float verdeckt werden darf (none | left | right | both | inherit)

```
<div style="background-color: silver;">
  
  
  <p style="border: solid;">Die ersten Asterix-Abenteuer begnügen
  sich ... </p>
</div>
```

```
<div style="background-color: silver;">
  
  
  <p style="border: solid; clear: both;">Die ersten Asterix-
  Abenteuer begnügen sich ... </p>
</div>
```

2.7 CSS-Elementfluss

Positionierung (I)

- Positionierte Elemente werden komplett aus dem Elementfluss heraus gelöst und gemäß einer von drei Modi im Anzeigefenster explizit platziert
- `position` als Alternative zu `float`

Absolute Positionierung

- Element wird aus dem normalen Elementfluss herausgelöst
- Position des Elements orientiert sich am Rand des nächsthöheren Elternelements, welches ebenfalls mittels Positionierung aus dem Elementfluss herausgelöst wurde
- Wenn der Anwender im Inhalt scrollt, scrollen absolut positionierte Elemente mit

Relative Positionierung

- Element wird aus dem normalen Elementfluss herausgelöst
- Angaben zu seiner Positionierung orientieren sich an der Position, die das Element im Normalfall einnehmen würde
- Wenn der Anwender im Inhalt scrollt, scrollen relativ positionierte Elemente mit

Fixe Positionierung

- Element wird aus dem normalen Elementfluss herausgelöst
- Positionierung ohne Rücksicht auf eventuell positionierte Elternelemente direkt an einer gewünschten Stelle im Anzeigenfenster
- Wenn der Anwender im Inhalt scrollt, scrollen fix positionierte Elemente nicht mit

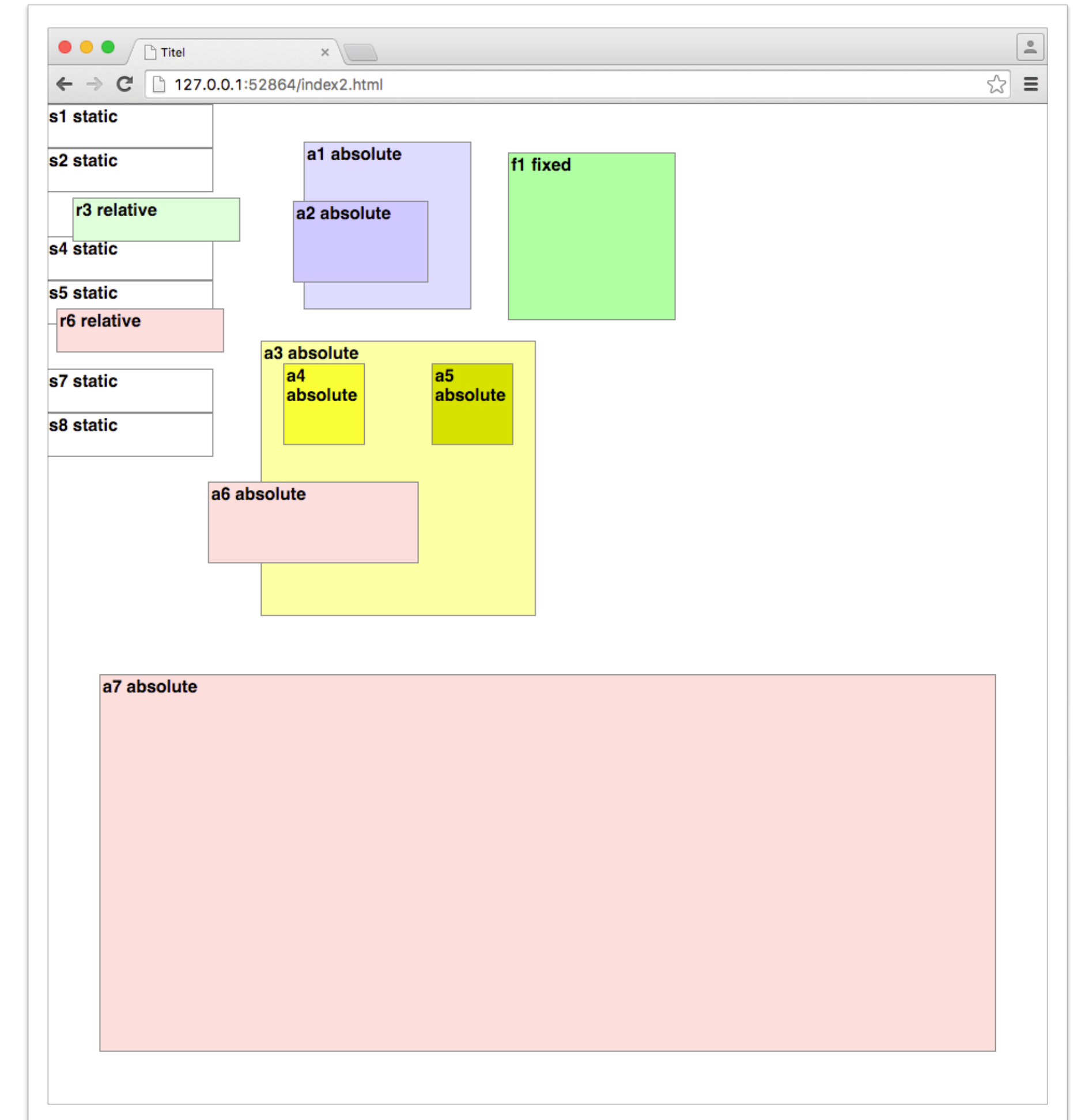
Bemerkungen

- HTML-Elemente mit `position: absolute` sitzen normalerweise relativ zum HTML-Dokument (Browserfenster)
- Wenn HTML-Element mit `position: absolute` in einem Block mit `position: relative` sitzt, wird der relativ positionierte Block zum Koordinatensystem des absolut positionierten HTML-Elements
- HTML-Blöcke mit `position: relative` verhalten sich wie Blöcke ohne Positionierung, sie fließen mit dem Inhalt und können mit `float` kombiniert werden
- Sitzt in einem relativ positionierten Block ein absolut positionierter Block wird die Position des absolut positionierten Blocks nicht mehr relativ zum HTML-Dokument, sondern relativ zum Block mit `position: relative` berechnet
- Für das Layout einer Webseite eignen sich eine statische Positionierung mittels `float` besser; `position: absolute` ist eher ungeeignet, da bei Änderung der Ausmaße für die Inhalte (z.B. Schriftgröße) immer manuell nachgearbeitet werden muss

2.7 CSS-Elementfluss Positionierung (II)

```
<head>
<style>
body {
  margin: 0;
  padding: 0;
  height: 1500px;
}
div {
  border: 1px solid #888;
  padding: 2px;
  font-family: sans-serif;
  font-weight: bold;
  color: black;
}
#s1 {width: 150px; height: 35px;}
#s2 {width: 150px; height: 35px;}
#r3 {width: 150px; height: 35px;
  position: relative; top: 5px;
  left: 25px; background-color: #dfd;}
#s4 {width: 150px; height: 35px;}
#s5 {width: 150px; height: 35px;}
#r6 {width: 150px; height: 35px;
  position: relative; top: -15px;
  left: 10px; background-color: #fdd;}
#s7 {width: 150px; height: 35px;}
#s8 {width: 150px; height: 35px;}
#a1 {position: absolute; top: 35px;
  left: 240px; width: 150px; height: 150px;
  z-index: 1; background-color: #ddf; }
#a2 {position: absolute; top: 90px; left: 230px;
  width: 120px; height: 70px;
  z-index: 2; background-color: #ccf; }
#a3 { position: absolute; top: 220px; left: 200px;
  width: 250px; height: 250px;
  z-index: 3; background-color: #ffa; }
```

```
#a4 {position: absolute; top: 20px; left: 20px;
  width: 70px; height: 70px;
  z-index: 1; background-color: #ff5;}
#a5 {position: absolute; top: 20px; right: 20px;
  width: 70px; height: 70px;
  z-index: 2; background-color: #dd2;}
#a6 {position: absolute; top: 130px; left: -50px;
  width: 190px; height: 70px;
  z-index: -1; background-color: #fdd;}
#a7 {position: absolute; top: 530px; left: 50px;
  bottom: 50px; right: 50px;
  background-color: #fdd;}
#f1 {position: fixed; top: 45px; left: 430px;
  width: 150px; height: 150px;
  background-color: #afa;}
</style></head>
<body>
<div id="s1">s1 static</div>
<div id="s2">s2 static</div>
<div id="r3">r3 relative</div>
<div id="s4">s4 static</div>
<div id="s5">s5 static</div>
<div id="r6">r6 relative</div>
<div id="s7">s7 static</div>
<div id="a1">a1 absolute</div>
<div id="a2">a2 absolute</div>
<div id="a3">a3 absolute
  <div id="a4">a4 absolute</div>
  <div id="a5">a5 absolute</div>
  <div id="a6">a6 absolute</div>
</div>
<div id="a7">a7 absolute</div>
<div id="f1">f1 fixed</div>
<div id="s8">s8 static</div>
</body></html>
```



2.8 Selektoren

Universal- und Elementselektoren

Webtechnologien

```
<h1>Webtechnologien</h1>
```

```
* {  
  font-family: arial;  
}
```

Webtechnologien

```
<h1>Webtechnologien</h1>
```

```
h1 {  
  color: red;  
}
```

Universalselektor

- Steuert alle HTML-Elemente
- Verwendung für allgemeine Angaben, wie zum Beispiel die Festlegung einer gemeinsamen Schrift

Elementselektor

- Steuert alle HTML-Elemente mit dem gleichen Namen

2.8 Selektoren

ID- und Class-Selektoren

WEBTECHNOLOGIEN

```
<h1 id="main">Webtechnologien</h1>
```

```
#main {  
  text-transform: uppercase;  
}
```

Webtechnologien

Vorlesung im Studiengang Medieninformatik

HTML liefert die Inhalte einer Website in einer strukturierten Form. CSS ergänzt das HTML und bereitet die Inhalte optisch auf. Eine gute HTML-Struktur ist elementar für Suchmaschinen.

```
<h1 class="line">Webtechnologien</h1>  
<h2>Vorlesung im Studiengang  
Medieninformatik</h2>  
<p class="line">HTML liefert die  
Inhalte einer Website in einer  
strukturierten Form. CSS ergänzt das  
HTML und bereitet die Inhalte optisch  
auf. Eine gute HTML-Struktur ist  
elementar für Suchmaschinen.</p>
```

```
.line {  
  text-decoration: underline;  
}
```

id-Selektor

- Deklaration bezieht sich auf genau ein HTML-Element welches den Selektor im id-Attribut setzt
- Eine CSS-Regel die über id vergeben wird hat eine hohe Gewichtung
- Ein id-Wert kann im HTML-Code nicht mit einem zweiten id-Wert kombiniert werden
- Über id werden eindeutige Aufgaben gesteuert die bei der Mehrfachvergabe einer id nicht funktionieren würden (zum Beispiel Steuerung von <input>-Feldern)
- CSS-Deklaration bezieht sich nur konkret auf das HTML-Element welches das zugehörige id-Attribut definiert, nicht auf alle HTML-Elemente des gleichen Namens

class-Selektor

- Deklaration bezieht sich auf alle HTML-Elemente welche den zugehörigen Selektor im class-Attribut setzen
- Wert von class kann in einem HTML-Dokument beliebig oft vorkommen
- Dient ausschließlich der Vergabe von CSS-Klassennamen
- In einem class-Attribut können mehrere Klassennamen kombiniert werden, die jeweils durch ein Leerzeichen voneinander getrennt sind
- Geringere Gewichtung als id-Selektor

2.8 Selektoren

Nachfahr- und Kind-Selektoren

Ene, mene, *Miste*,
es rappelt in der *Kiste* !

```
<p>
  Ene, mene,
  <em>Miste</em>, <br>
  <span>es rappelt in der
    <em>Kiste</em>
  </span>!
</p>
```

```
p em {
  color: red;
  font-weight: bold;
}
p span {
  background: wheat;
}
```

Ene, mene, *Miste*,
es rappelt in der *Kiste* !

```
p > em {
  color: red;
  font-weight: bold;
}
p span {
  background: wheat;
}
```

Nachfahrselektor

- Kombination zweier oder mehrerer Einfachselektoren die einen Pfad innerhalb des DOM beschreiben
- Deklaration bezieht sich auf das letzte Element des Pfades welches die zuvor genannten Elemente als direkte oder indirekte Vorgänger hat

Kind-Selektor

- Kombination zweier oder mehrerer Einfachselektoren die einen Pfad innerhalb des DOM beschreiben, wobei alle Elemente direkt miteinander verbunden sein müssen
- Deklaration bezieht sich auf das letzte Element des Pfades, welches das zuvor genannte Element als direkten Vorgänger hat

2.8 Selektoren

Direkte und Indirekte Geschwister

Schirm *Charme*
und *Melone*

```
<p>
  <strong>Schirm</strong>
  <em>Charme</em><br>
  und <em>Melone</em>
</p>
```

```
strong + em {
  color: red;
  font-weight: bold;
}
```

Schirm *Charme*
und *Melone*

```
strong ~ em {
  color: red;
  font-weight: bold;
}
```

Direkter Geschwisterselektor

- Referenziert eine Sequenz von Geschwistern, das heißt mehrere in einem Element eingebettete Kindelemente
- CSS-Deklaration bezieht sich auf das letzte "Geschwisterchen"

Indirekter Geschwisterselektor

- Wie direkter Geschwisterselektor, aber das Element auf das sich Deklaration bezieht muss nicht notwendigerweise direkt benachbart sein - es können sich beliebig viele andere Elemente zwischen dem referenzierten Element und seinen Vorgängern befinden

2.8 Selektoren

Kombination und Gruppierung von Selektoren

WEBTECHNOLOGIEN

```
<h1 id="main"  
class="line">Webtechnologien</h1>
```

```
* {  
  xfont-family: arial;  
}  
h1 {  
  color: red;  
}  
#main {  
  text-transform: uppercase;  
}  
.line {  
  text-decoration: underline;  
}
```

```
h1, h2, p {  
  color: #222222;  
}
```

```
h1 {  
  color: #222222;  
}  
h2 {  
  color: #222222;  
}  
p {  
  color: #222222;  
}
```

Kombination von CSS-Regeln mit verschiedenen Selektoren

- Nicht nur mehrere Elemente lassen sich mit einer CSS-Regel ansprechen, auch umgekehrt können sich mehrere CSS-Regeln auf ein HTML-Element beziehen

Gruppierung von Selektoren

- Elemente mit gleichen Eigenschaften können zusammengefasst werden
- Die verschiedenen Selektoren werden hintereinander geschrieben und durch ein Komma voneinander getrennt

2.8 Selektoren

Alle Selektoren im Überblick (I)

Selektor	Selektortyp	Beschreibung	ab CSS
*	Universalselektor/ Einfachselektor	Steuert all HTML-Elemente.	2
E	Elementselektor (Typselektor)/Einfachselektor	Steuert alle HTML-Elemente vom Typ E.	1
#meine-id	ID-Selektor/Einfachselektor	Steuert das HTML-Element mit der id meine-id.	1
.meine-klasse	Klassenselektor/ Einfachselektor	Steuert Elemente mit der Klasse meine-klasse.	1
E F	Nachfahrselektor/ Kombinationsselektor	Steuert alle Elemente vom Typ F, die in einem Element vom Typ E vorkommen.	1
E > F	Kindselektor/ Kombinationsselektor	Steuert Elemente vom Typ F, die direkt unterhalb eines Elements vom Typ E vorkommen (nicht solche die in der Struktur weiter unten vorkommen).	2
E + F	Direkter Geschwisterselekt/ Kombinationsselektor	Steuert Elemente vom Typ F, die direkt hinter einem Element vom Typ E stehen (nur das direkte Geschwisterelement).	2
E ~ F	Indirekter Geschwisterselektor/ Kombinationsselektor	Steuert alle Elemente vom Typ F, die hinter einem Element vom Typ E stehen (alle Geschwisterelemente).	3
E[attr]	Attributselektor	Steuert alle Elemente mit dem angegebenen Attribut. Dabei spielt es keine Rolle, ob oder welchen Wert dieses hat.	2
E[attr="value"]	Attributselektor	Steuert Elemente deren Attribute genau und ausschließlich den angegebenen Wert haben.	2
E[attr~="value"]	Attributselektor	Steuert Elemente deren Attribut den angegeben Wert besitzt, auch wenn mehrere Werte für das Attribut gesetzt sind	2

mit Änderungen übernommen von: Heller, St. (2012) Workshop HTML 5 & CSS3, dpunkt-Verlag

2.8 Selektoren

Alle Selektoren im Überblick (II)

Selektor	Selektortyp	Beschreibung	ab CSS
e[attr^="value"]	Attributselektor	Steuert Elemente deren Attributwerte mit der Zeichenkette des angegebenen Wertes anfangen,	3
E[attr\$="value"]	Attributselektor	Steuert Elemente deren Attributwerte mit der Zeichenkette des angegebenen Wertes enden.	3
E[attr*="value"]	Attributselektor	Steuert Elemente bei denen die Zeichenkette des angegebenen Wertes im Attributwert vorkommt.	3
E[attr = "value"]	Attributselektor	Steuert Elemente deren Werte des Attributs eine Reihe von mit Minuszeichen getrennten Segmenten haben, wovon das erste Segment value ist. Beispiel: Der Selektor [lang="de"] würde jedes HTML-Element finden, das lang="de" aber auch lang="de-ch" beinhaltet.	2
E:link	Pseudoklasse	Steuert Links die noch nicht angeklickt wurden.	1
E:visited	Pseudoklasse	Steuert Links die bereits angeklickt wurden und damit in der Historie des Browsers zu finden sind.	1
E:active	Pseudoklasse	Steuert den Link in dem Moment, wenn er durch den Benutzer angeklickt wird.	1
E:hover	Pseudoklasse	Steuert den Link,den der Benutzer mit der Maus überfährt.	2
E:focus	Pseudoklasse	Steuert den Link an dessen Position sich der Nutzer beim "Tabben" befindet.	2
E:target	Target-Pseudoklasse	Steuert eine Sprungmarke in dem Moment wenn sie angesprungen wird.	3
E:lang(de)	Sprach-Pseudoklasse	Steuert alle Elemente mit der Sprachauszeichnung "de". Greift auch, wenn die lang-Eigenschaft geerbt wurde.	2
E:enabled	Userinterface-Pseudoklasse	Steuert Formularfelder,in die Werte eingegeben werden können bzw. deren Bedienung möglich ist.	3
E:disabled	Userinterface-Pseudoklasse	Steuert Formularfelder die über das Attribut disabled für die Eingabe gesperrt sind bzw. deren Bedienung nicht möglich ist.	3
E:checked	Userinterface-Pseudoklasse	Steuert aktivierte Checkboxes oder Radioboxen.	3
E:root	Strukturpseudoklasse	Wurzelelement eines Dokuments, in HTML immer das html-Tag.	3
E:nth-child(n)	Strukturpseudoklasse	Steuert jedes n-te Element innerhalb eines Elternelements E.	3

mit Änderungen übernommen von: Heller, St. (2012) Workshop HTML 5 & CSS3, dpunkt-Verlag

2.8 Selektoren

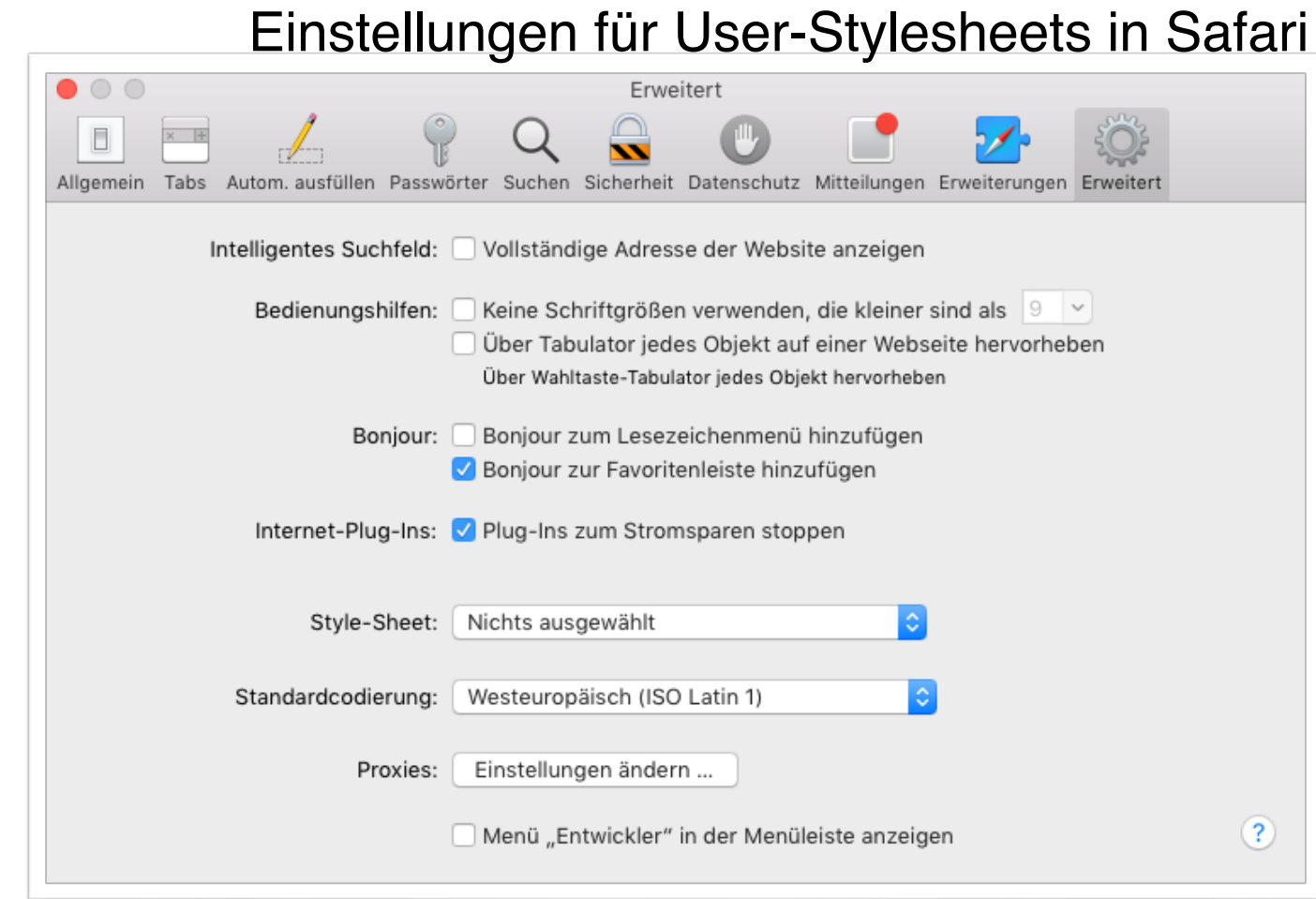
Alle Selektoren im Überblick (III)

Selektor	Selektortyp	Beschreibung	ab CSS
E:nth-last-child(n)	Strukturpseudoklasse	Steuert jedes n-te Kindelement in einem Element, dabei werden die Kindelemente von hinten durchlaufen.	3
E:nth-of-type(n)	Strukturpseudoklasse	Steuert jedes n-te Element vom gleichen HTML-Typ auf gleicher Ebene (Geschwisterlemente).	3
E:nth-last-of-Type(n)	Strukturpseudoklasse	Steuert jedes n-te Element auf gleicher Ebene (Geschwisterelemente), dabei werden die Elemente von hinten durchlaufen.	3
E:first-child	Strukturpseudoklasse	Steuert das erste Kindelement innerhalb eines Elements.	2
E:last-child	Strukturpseudoklasse	Steuert das letzte Kindelement innerhalb eines Elements.	3
E:first-of-type	Strukturpseudoklasse	Steuert das erste Element des gleichen HTML-Elementtyps innerhalb eines Elternelements.	3
E:last-of-type	Strukturpseudoklasse	Steuert das letzte Element des gleichen HTML-Elementtyps innerhalb eines Elternelements.	3
E:only-child	Strukturpseudoklasse	Steuert ein Element, das keine Geschwisterlemente hat und damit das einzige Kindelement im übergeordneten Element ist.	3
E:only-of-type	Strukturpseudoklasse	Steuert ein Element, das keine Geschwisterelemente vom gleichen HTML-Typ hat und damit das einzige Kindelement dieser Sorte im übergeordneten Element ist.	3
E:empty	Strukturpseudoklasse	Steuert leere Elemente.	3
E:not(element)	Negationspseudoklasse	Steuert alle Elemente außer dem Element, welches in der Klammer angegeben ist.	2
E::first-line	first-line Pseudoelement	Steuert die ersten Zeilen in dem Element E.	1
E::first-letter	first-letter Pseudoelement	Steuert die ersten Buchstaben in dem Element E.	1
E::before	before Pseudoelement	Steuert die generierten Inhalte vor dem Element E.	2
E::after	after Pseudoelement	Steuert die generierten Inhalte nach dem Element E.	2

mit Änderungen übernommen von: Heller, St. (2012) Workshop HTML 5 & CSS3, dpunkt-Verlag

2.9 Ergänzung und Vererbung

Kaskaden und verschiedene Stylesheets



Empfehlung: Browser-Reset durch CSS-Eigenschaften für das body-Tag überschreiben browser-spezifische Einstellungen:

```
body {  
  padding: 0px;  
  margin: 0px;  
  background-color: white;  
  color: black;  
  font-family: Arial, Helvetica, sans-serif;  
  font-size: 12px;  
  font-weight: normal;  
}
```

Kaskaden

- **Cascading** Style Sheets schalten mehrere CSS-Dateien hintereinander
- *Kaskade*: Hintereinanderschaltung oder Verkettung
- Drei Arten von Stylesheets

Browser Stylesheets

- Browser-internes CSS, welches immer verwendet wird, selbst wenn im HTML-Code keine CSS-Datei eingebunden wird
- Definition der Grundeigenschaften aller HTML-Elemente

Autoren-Stylesheets

- CSS welches über HTML eingebunden wird bzw. im HTML-Code selbst steht
- Stylesheets des Entwicklers der Website

Nutzer-Stylesheets

- Persönliches Stylesheet eines Nutzers
- Ermöglicht die Anpassung der Darstellung einer Website an persönliche Bedürfnisse
- Einbindung über die Einstellungen des Browsers

2.9 Ergänzung und Vererbung

Priorisierung von CSS-Regeln

- Mehrere Stylesheets ersetzen sich nicht, sondern ergänzen sich
- Aus den drei Arten von Stylesheets wird ein gesamtes Stylesheet errechnet
- Regeln ergänzen sich im berechneten Stylesheet
- Bei sich widersprechenden Regeln für dasselbe Element, "gewinnt" die aus dem Stylesheet mit der höchsten Priorität
- Prioritäten

1. Nutzer-Stylesheet-Definitionen mit !important
2. Autoren-Stylesheet-Definitionen mit !important
3. Autoren-Stylesheet-Definitionen
4. Benutzer-Stylesheet-Definitionen
5. Browser-Stylesheet-Definitionen

```
/* Beispiel für !important */  
p { font-size: 1em !important;}
```

```
/* Browser-Stylesheet */  
blockquote {  
  display: block;  
  margin: 0px;  
}
```

+

```
/* Autoren-Stylesheet */  
blockquote {  
  color: darkblue;  
}
```

=

```
/* berechnete CSS-Regel */  
blockquote {  
  display: block;  
  margin: 0px;  
  color: darkblue;  
}
```

Ergänzung von Regeln
im berechneten
Stylesheet

```
/* Browser-Stylesheet */  
blockquote {  
  display: block;  
  margin: 1em 40px;  
}
```

+

```
/* Autoren-Stylesheet */  
blockquote {  
  margin: 1em 0px;  
}
```

=

```
/* berechnete CSS-Regel */  
blockquote {  
  display: block;  
  margin: 1em 0px;  
  color: darkblue;  
}
```

Die margin-Regel aus
dem Autoren-
Stylesheet
überschreibt margin
aus dem Browser-
Stylesheet

2.9 Ergänzung und Vererbung

Priorisierung von CSS-Regeln basierend auf Selektoren

Selektor	Beispiel	A	B	C	D
Universalselektor	*				0
Elementselektor	h1				1
Pseudoselektor	::first-line				1
Pseudoklasse	:hover			1	
Klassenselektor	.hinweis			1	
ID-Selektor	#inhalt		1		
Inline-Style	<p style=color: red">	1			

- Prioritätsregeln nicht nur zwischen Browser-, Autoren- und Nutzer-Styles, sondern auch zwischen Selektoren
- Berechnung der Priorität anhand der *Spezifität* der Selektoren
- Eingruppierung der Selektoren in verschiedene Prioritätsklassen gemäß nebenstehender Tabelle
- Bei kombinierten und/oder gruppierten Selektoren erfolgt Aufaddierung der Punkte der zugehörigen Klassen zur Berechnung der Priorität
- Inline-Styles haben die höchste Priorität (A), der Universalselektor die niedrigste (B)

Spezifität: B:1 C:0 D:3

```
#inhalt ul li a {  
  color: red;  
}
```

Spezifität: B:1 C:0 D:2

```
#inhalt li a {  
  color: blue;  
}
```

Die erste Regel gewinnt

Spezifität: D:2

```
ul a {  
  color: red;  
}
```

Spezifität: D:2

```
li a {  
  color: red;  
}
```

Gleichwertige Selektoren,
die letzte Regel gewinnt

Faustregel

- Niemand setzt sich hin und berechnet Spezifitäten ;-)
- `id` ist immer stärker als `class`
- `class` ist immer stärker als ein einfacher Elementselektor