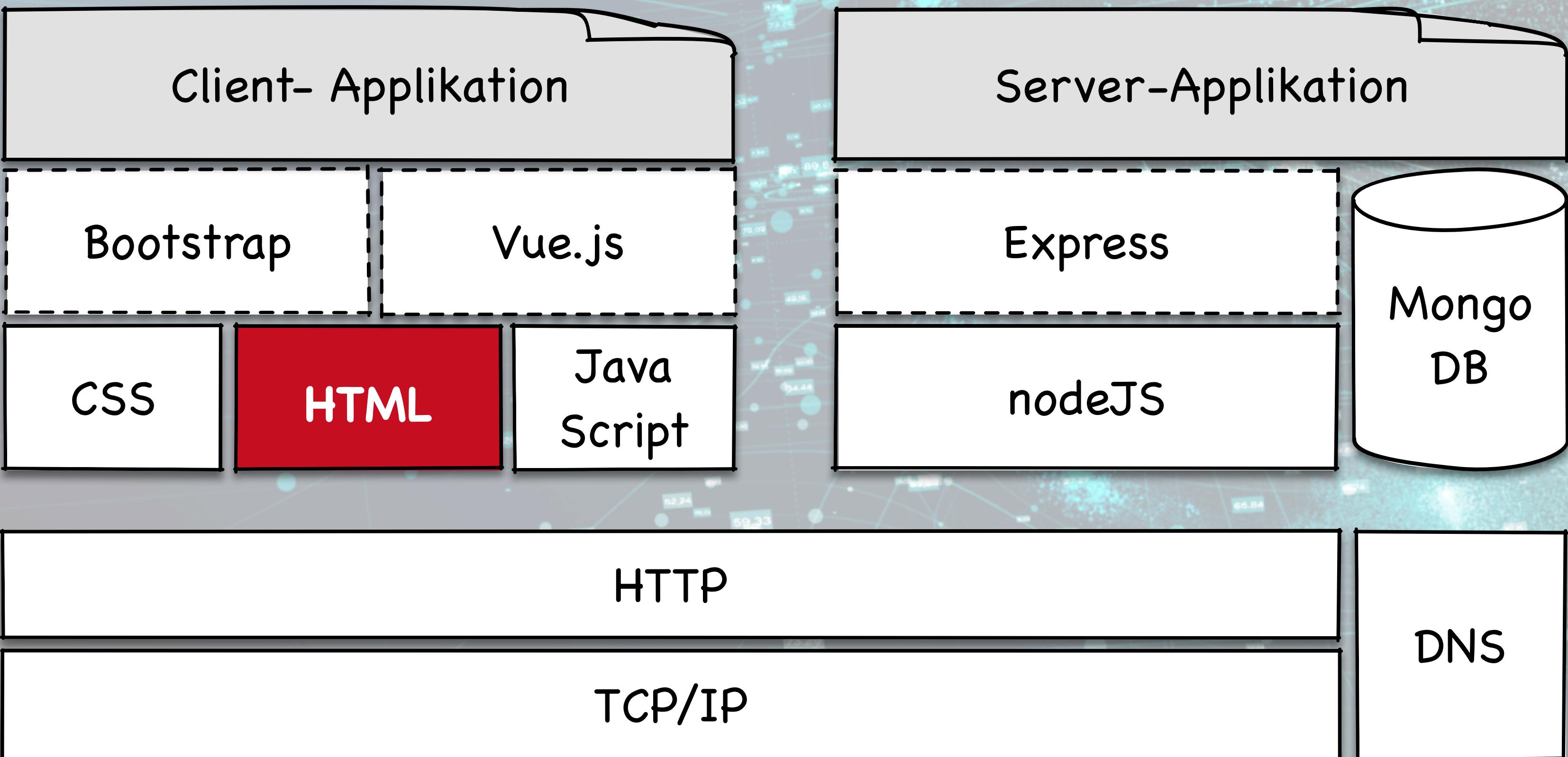


# WEB TECHNOLOGIEN 2022

KAPITEL 2: HTML

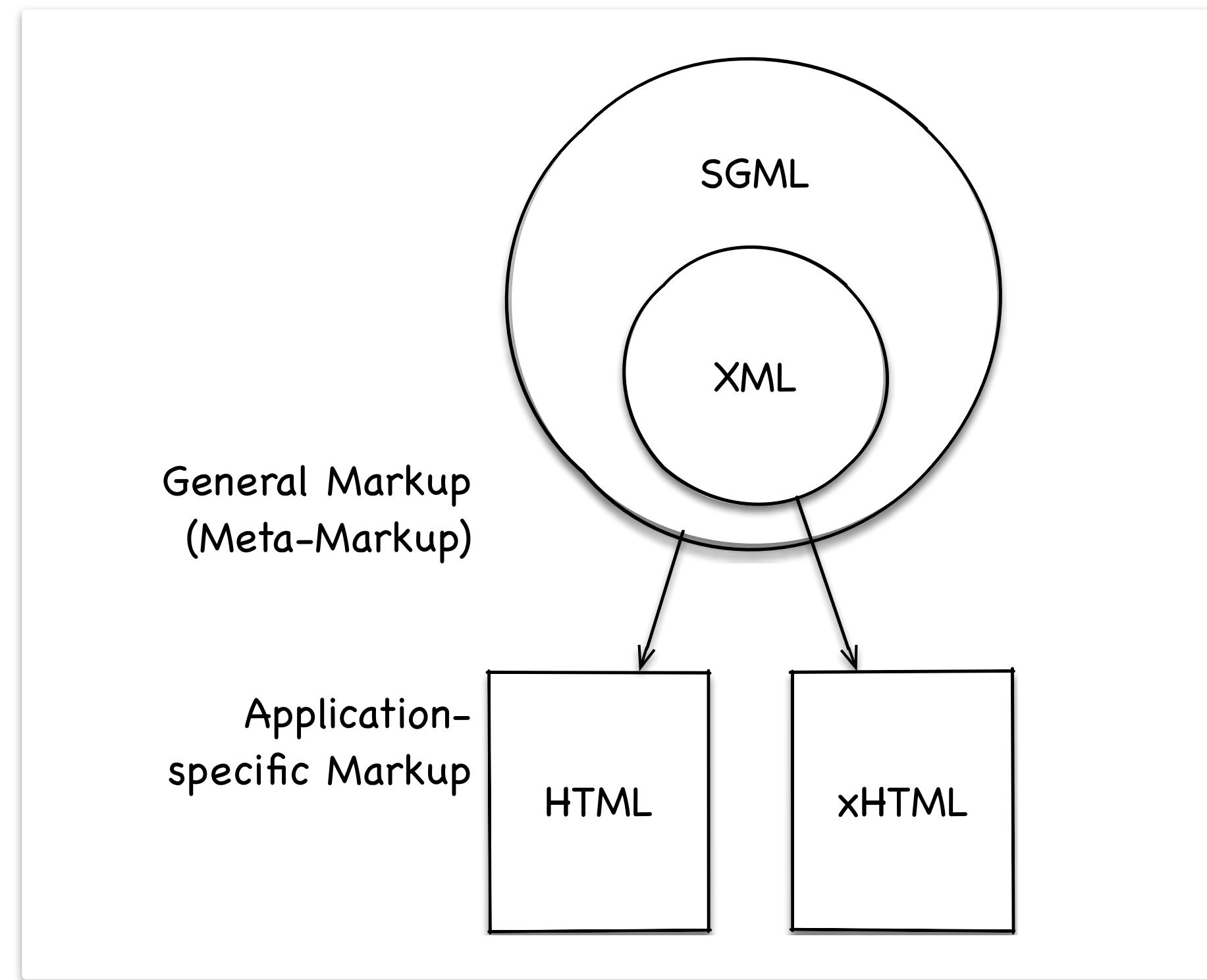
PROF. DR. AXEL KÜPPER  
FACHGEBIET SERVICE-CENTRIC NETWORKING | TU BERLIN & T-LABS

# WEBTECHNOLOGIEN ÜBERBLICK



# 2.1 MARKUP-SPRACHEN

## WAS SIND MARKUP-SPRACHEN?



### MARKUP-SPRACHEN (AUSZEICHNUNGSSPRACHEN)

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

### STANDARDIZED GENERALIZED MARKUP LANGUAGE (SGML)

- Metasprache zur Definition von Auszeichnungssprachen für Dokumente
- Basiert auf einer Dokumenttypdefinition (DTD), die den strukturellen Aufbau des Dokuments beschreibt

### EXTENSIBLE MARKUP LANGUAGE (XML)

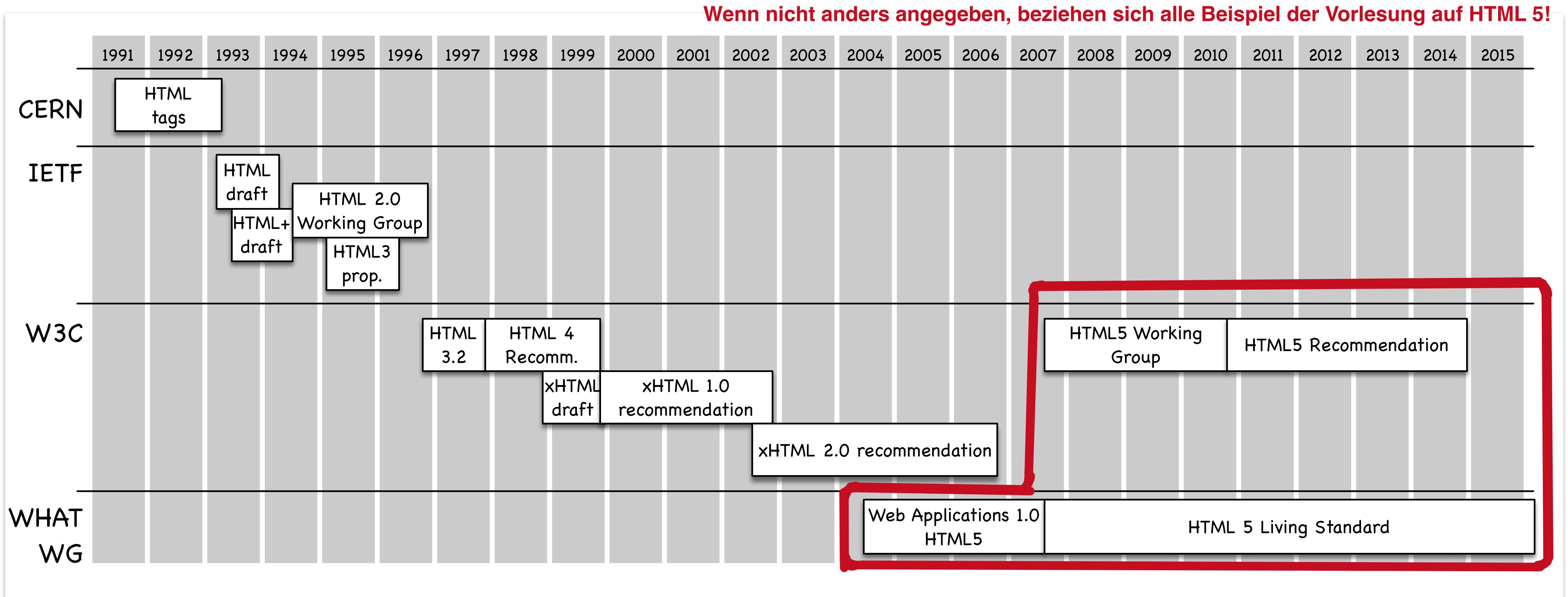
- Auszeichnungssprache für die maschinen- und menschenlesbare Kodierung
- Abgeleitet von SGML (weniger komplex)
- Basis für viele domänen spezifische Dokumentenformate

### HYPERTEXT MARKUP LANGUAGE (HTML)

- Auszeichnungssprache für Webseiten
- Annotationen für Überschriften, Absätze und andere Strukturelemente

# 2.1 MARKUP-SPRACHEN

## ENTWICKLUNG WEBBASIERTER MARKUP-SPRACHEN



## 2.1 MARKUP-SPRACHEN

### HTML VERSUS CSS (I)



#### HTML5

- Aktuellste verabschiedete Version des HTML-Standards
- Standardisiert durch die W3C HTML WG und die Web Hypertext Application Technology Working Group (WHATWG)
- 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

↳ visuelle Erscheinung

- 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 uvm.
- CSS3 aktuellste Version mit mehr als 300 Eigenschaften
- Neue Merkmale sind Multi-Column-Layout, abgerundete Ecken, transparente Hintergründe uvm.

# 2.1 MARKUP-SPRACHEN

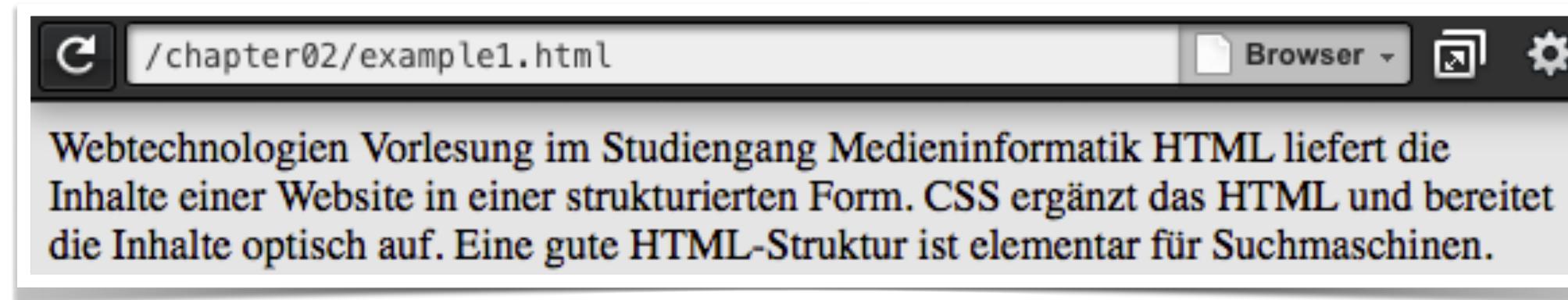
## HTML VERSUS CSS (II)

– innerhalb von body gibt es keine weitere Semantik

↳ da keine weiteren HTML Elemente, wie bspw. div

```
1 <body>
2   Webtechnologien
3   Vorlesung im Studiengang Medieninformatik
4
5   HTML liefert die Inhalte einer Website
6   in einer strukturierten Form.
7   CSS ergänzt das HTML und bereitet die Inhalte optisch auf.
8   Eine gute HTML-Struktur ist elementar für Suchmaschinen.
9 </body>
```

/chapter02/example01.html



### HTML 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 und Texten, um sie in eine semantische Form zu bringen
- Strukturierung geschieht mit Hilfe von HTML-Elementen

# 2.1 MARKUP-SPRACHEN

## HTML VERSUS CSS (III)

The screenshot shows a code editor on the left and a browser window on the right. The code editor displays the following HTML code:

```
1 <body>
2   <div>Webtechnologien</div>
3   <div>Vorlesung im Studiengang Medieninformatik</div>
4
5   <div>HTML liefert die Inhalte einer Website
6   in einer strukturierten Form.
7   CSS ergänzt das HTML und bereitet die Inhalte optisch auf.
8   Eine gute HTML-Struktur ist elementar für Suchmaschinen.</div>
9 </body>
```

The browser window shows the rendered HTML with the following content:

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.

### HTML MIT STRUKTUR

- Strukturierung des Textes mit `<div>`-Elementen
- `<div>` definiert den eingeschlossenen Inhalt als 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 (und JavaScript) für die Definition des Designs dieses Inhalts

## 2.1 MARKUP-SPRACHEN

### HTML VERSUS CSS (IV)

Design Information

The screenshot shows a code editor window at the top and a browser window below it. The code editor displays the following HTML code:

```
1 <body>
2   <div style="font-size: 30px; color: red">
3     Webtechnologien
4   </div>
5   <div style="font-size: 24 px; color: blue">
6     Vorlesung im Studiengang Medieninformatik
7   </div>
8   <div>
9     HTML liefert die Inhalte einer Website
10    in einer strukturierten Form.
11    CSS ergänzt das HTML und bereitet die Inhalte optisch auf.
12    Eine gute HTML-Struktur ist elementar für Suchmaschinen.
13  </div>
14 </body>
```

The browser window shows the rendered HTML with the following content:

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.

### HTML MIT EINBINDUNG VON CSS

- Optische Anreicherung der `<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

## 2.1 MARKUP-SPRACHEN

### HTML VERSUS CSS (V)

Semantik von HTML (dabspur. Als aufdrin ist)  
CSS/Design Information

```
1 <body>
2   <h1 style="font-size: 30px; color: red">
3     Webtechnologien
4   </h1>
5   <h2 style="font-size: 24px; color: blue">
6     Vorlesung im Studiengang Medieninformatik
7   </h2>
8   <p>
9     HTML liefert die Inhalte einer Website
10    in einer strukturierten Form.
11  </p>
12  <p>
13    CSS ergänzt das HTML und bereitet die Inhalte optisch auf.
14    Eine gute HTML-Struktur ist elementar für Suchmaschinen.
15  </p>
16 </body>
```

**Webtechnologien**

**Vorlesung im Studiengang Medieninformatik**

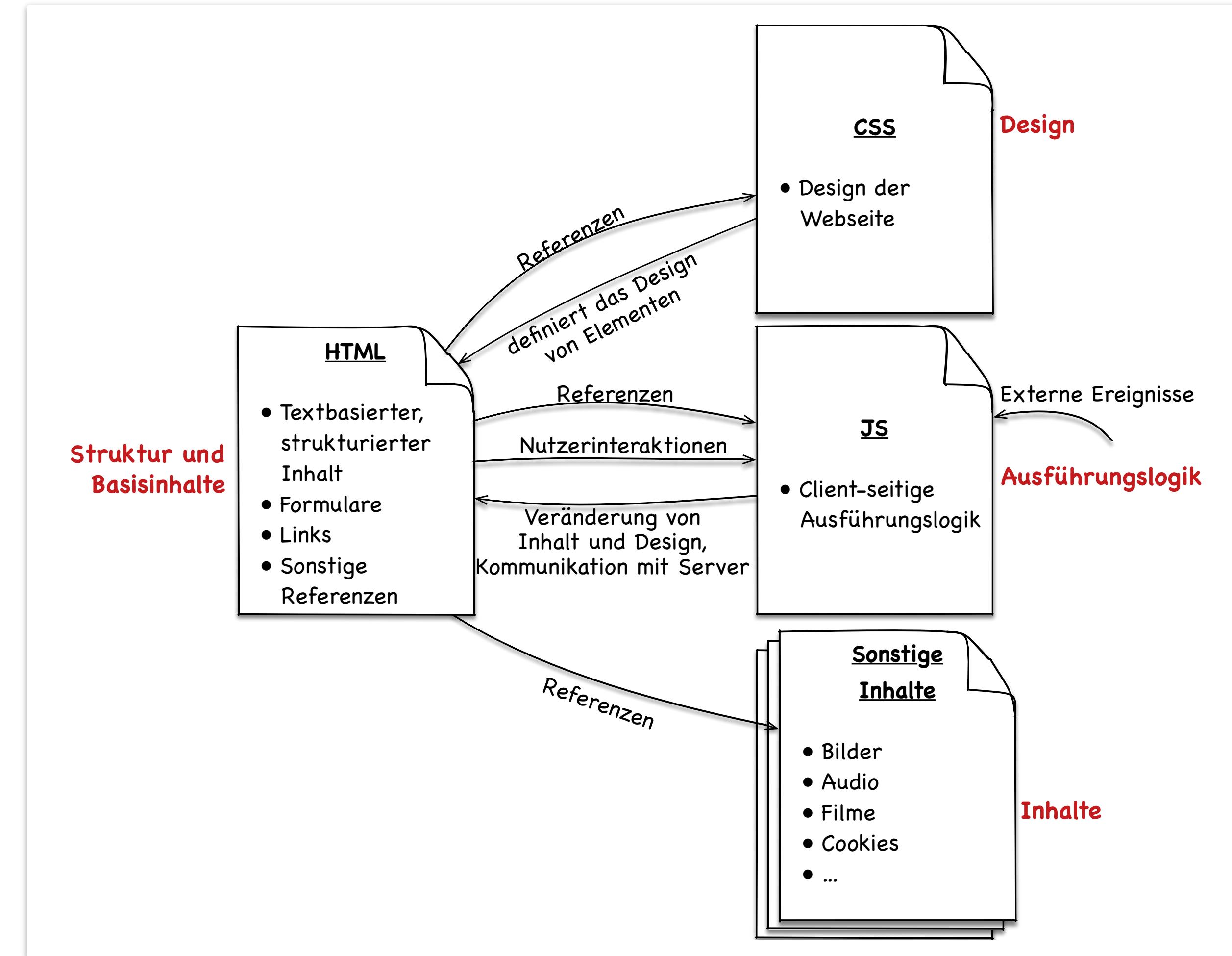
HTML liefert die Inhalte einer Website in einer strukturierten Form.

CSS ergänzt das HTML und bereitet die Inhalte optisch auf. Eine gute HTML-Struktur ist elementar für Suchmaschinen.

### HTML MIT SEMANTIK UND EINBINDUNG VON CSS

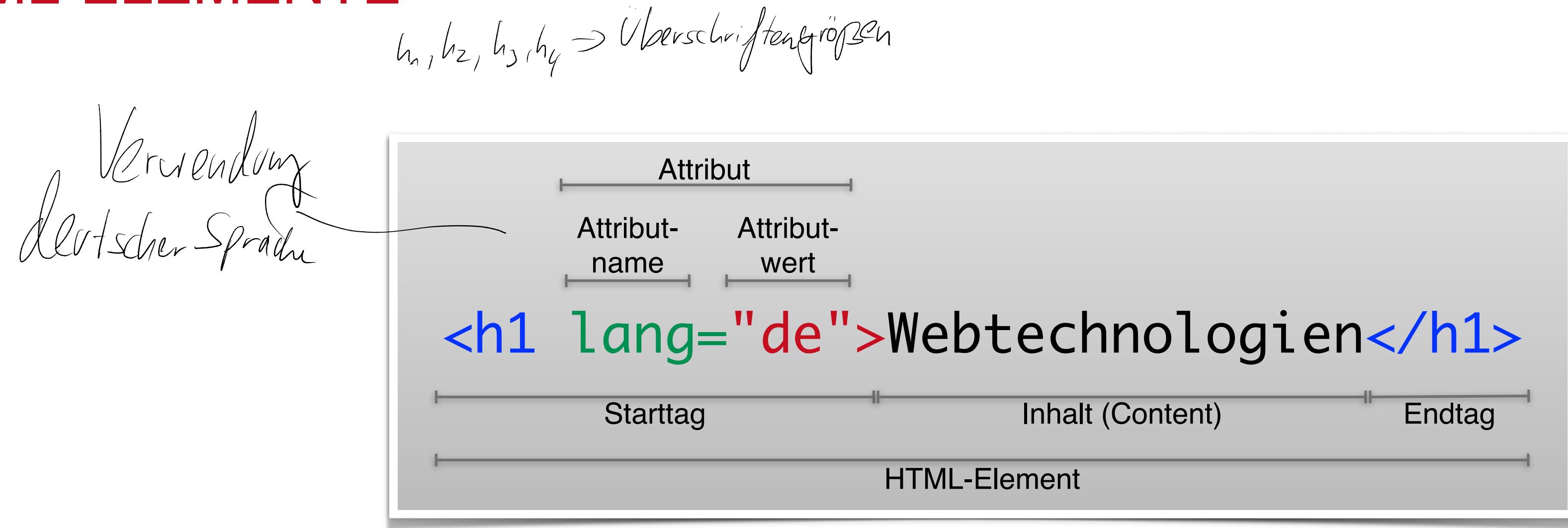
- Kombination von strukturiertem Text und Design mittels CSS
- Durch dedizierte Elemente kann zwischen Überschriften, Unterüberschriften und Textabsätzen unterschieden werden - der Text ist semantifiziert
- Maschinen können eingeschränkt die Semantik von Inhalten auf Webseiten erfassen
- Nützlich bei der/für die
  - Erstellung von Druckvorlagen,
  - Interpretation der Inhalte durch Suchmaschinen
  - automatischen Konvertierung von Webseiten
  - ...

# 2.1 MARKUP-SPRACHEN VERKNÜPFUNGEN ZWISCHEN HTML, CSS UND JAVASCRIPT



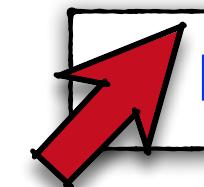
## 2.2 HTML-SYNTAX

### HTML-ELEMENTE



- Die meisten *HTML-Elemente* bestehen aus
  - Starttag (optional mit Attributen)
  - Inhalt
  - Endtag
- Leere HTML-Elemente sind ebenfalls möglich und solche ohne Endtag
- Einbettung: HTML-Element innerhalb eines anderen wird *Kindelement (child element)*, das umgebende *Elternelement (parent element)* und Kinder desselben Elternelements werden *Geschwisterelemente* genannt

- *Case-insensitive*: Für Elementnamen, Attributnamen und Attributwerte können Groß- und Kleinbuchstaben verwendet werden (*case insensitive*)
- *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



<https://www.w3.org/TR/html/introduction.html>

## 2.2 HTML-SYNTAX

### LEERE HTML-ELEMENTE

Vollständige Schreibweise

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

Schreibweise ohne End-Tags  
*(fehleranfällig - nicht empfohlen)*

```

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

Verkürzte Schreibweise in xHTML  
*(nicht empfohlen)*

```

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

- 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
- "Leere" Elemente können in xHTML abgekürzt dargestellt werden, indem am Ende des



## 2.2 HTML-SYNTAX

### HTML-ELEMENTE MIT OPTIONALEM ENDTAG

```
<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
- Übersichtlichkeit schaffen: auch wenn verschiedene Elemente ohne schließendes Tag auskommen, werden Quelltexte übersichtlicher, wenn alle Elemente mit einem Endtag versehen sind

## 2.2 HTML-SYNTAX ATTRIBUITE (I)

- *Attribute* bestimmen die Eigenschaften eines HTML-Elements und werden ausschließlich innerhalb des Starttags angegeben
- Bestehen aus *Attributnamen* und *Attributwert* (letzterer eingegrenzt in ' oder ")
- Optionale Schreibweise ohne Anführungszeichen, dann aber nur wenn das 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

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

### ATTRIBUTE MIT FESTEN ATTRIBUTWERTEN

- Mögliche Werte sind vordefiniert; anderen Werte sind unzulässig

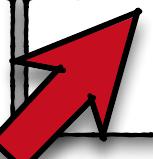
```
<a href="index.html" tabindex="15">  
  <p id="intro"></p>  
  <p class="beispiel-typ-3"></p>  
</a>
```

### 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



<https://www.w3.org/TR/html/introduction.html>



<https://www.w3.org/TR/html/infrastructure.html>

## 2.2 HTML-SYNTAX ATTRIBUTÉ (II)

### 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>
```

### ATTRIBUTE MIT FREIEN WERTEN

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

```
  

```

### 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>
```



<https://www.w3.org/TR/html/introduction.html>



<https://www.w3.org/TR/html/infrastructure.html>

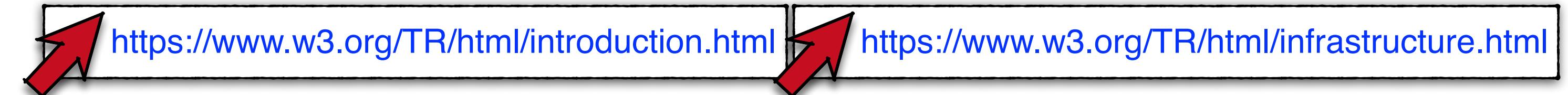
## 2.2 HTML-SYNTAX ATTRIBUTES (III)

### 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: Standardwerte, die HTML5 vorsieht für den Fall, dass das jeweilige 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
- Zur Unterscheidung zwischen Steuerzeichen und Inhalten müssen Inhaltszeichen, die identisch mit Steuerzeichen sind, im Inhaltsteil als HTML Entity Name oder Unicode Entity Number geschrieben werden
- Beide Schreibweisen beginnen mit & und enden mit einem ;
- Moderne Browser erkennen reservierte Zeichen im Text und können diese von HTML-Steuerzeichen unterscheiden
- Ausnahme: < muss im Textteil immer als &lt; geschrieben (führt sonst zu Validierungsfehler), > sollte als &gt; 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

mit Änderungen übernommen von: Heller, St. (2012) Workshop HTML5 & CSS3, dpunkt-Verlag

Zeichen	Schreibweise in HTML	Schreibweise in Unicode	Beschreibung	Abgeleitet von
<	&lt;	&#60;	Öffnende spitze Klammer/Kleiner-als-Zeichen	lower than
>	&gt;	&#62;	Schließende spitze Klammer/Größer-als-Zeichen	greater than
"	&quot;	&#34;	Doppelte Anführungszeichen	quotation mark
'	&apos;	&#39;	Einfaches Anführungszeichen	apostrophe
&	&amp;	&#38;	Kaufmännisches Und	ampersand

### MODERNE VERSION (NICHT BEI ALTEN BROWSERN)

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

### ALTE VERSION (ALTE UND NEUE BROWSER)

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

## 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 ISO 8859-1) erfordert Verwendung des HTML-Entity-Formats

### 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
- &nbsp; ist ein geschütztes Leerzeichen welches den Zeilenumbruch verhindert

<p>Web&nbsp;2.0</p>

Zeichen	Schreibweise in HTML	Schreibweise in Unicode	Beschreibung	Abgeleitet von
ä	&auml;	&#228;	a-Umlaut	[a] + umlaut
Ä	&Auml;	&#196;	A-Umlaut	[A] + umlaut
ö	&ouml;	&#246;	o-Umlaut	[o] + umlaut
Ö	&Ouml;	&#214;	O-Umlaut	[O] + umlaut
ü	&uuml;	&#252;	u-Umlaut	[u] + umlaut
Ü	&Uuml;	&#220;	U-Umlaut	[U] + umlaut
ß	&szlig;	&#223;	Scharfes S (Eszett)	[sz] + ligature
€	&euro;	&#8364;	Euro-Zeichen	euro
©	&copy;	&#169;	Copyright-Zeichen	copyright
	&nbsp;	&#160;	Geschütztes Leerzeichen	non-breaking space
§	&sect;	&#167;	Paragrafenzeichen	section

## 2.2 HTML-SYNTAX

### RELATIVE UND ABSOLUTE PFÄDE

- Dateien wie Grafiken, JavaScript oder Cascading Style Sheets können sowohl relativ als auch absolut eingebunden werden
- 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
- Für Suchmaschinenoptimierung werden kryptische URLs durch suchmaschinenfreundliche Adressen ersetzt
- Für den Webbrowser erscheint es, als läge eine Datei im Verzeichnis /webtechnologien/outline/
- Webserver (genauer eine .htaccess-Datei) bildet simulierten Pfad auf den echten Pfad ab und ruft eine index.php-Datei auf



## 2.2 HTML-SYNTAX

### AUFBAU EINER HTML-SEITE

```
1  <!DOCTYPE html>
2  <html lang="de">
3      <head>
4          ...
5      </head>
6
7      <body>
8          ...
9      </body>
10 </html>
```

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">
```

- Grundaufbau einer HTML-Datei besteht aus den Elementen `<html>`, `<head>` und `<body>`

- Universalattribut `lang` kennzeichnet Sprache der Webseite

#### 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



<https://www.w3.org/TR/html/document-metadata.html>

## 2.3 HTML HEAD

### ELEMENTE IN <HEAD> (I)

#### SCHLECHTER TITEL

```
<title>::: Meine Homepage :::</title>
```

#### GUTER TITEL

```
<title>TU Berlin | Vorlesung Webtechnologien | WS 2020/21</title>
```

```
1 <head>
2   <meta charset="UTF-8">
3   <meta name="description" content="Vorlesung Webtechnologien">
4   <meta name="keywords" content="HTML,CSS,JavaScript">
5   <meta name="author" content="Axel Küpper">
6 </head>
```

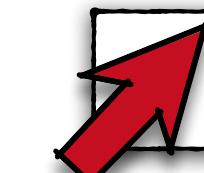
↳ key      ↳ value

#### <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

#### <meta>-ELEMENT

- Metadaten sind "Daten über Daten" → Bspw. UTF-8
- <meta>-Element enthält Metadaten über die Webseite
- Inhalt von <meta> wird nicht auf der Webseite dargestellt, kann aber durch Maschinen (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)



<https://www.w3.org/TR/html/document-metadata.html>

## 2.3 HTML HEAD ELEMENTE IN <HEAD> (III)

DIREKTE CSS-  
ANWEISUNG IN  
<head>

```
h1 soll dunkelblau sein  
<style type="text/css">  
  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

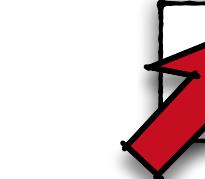
```
<script src="externes-js.js"></script>  
<script>  
  alert ('Hallo Welt!');  
</script>
```

### <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 eine externe JavaScript-Datei
- Einbindung einer externen Datei funktioniert nicht über <link> und href
- Keine Kombination von eingebundenen Anweisung und JavaScript-Dateien innerhalb eines <script>-Elements



<https://www.w3.org/TR/html/document-metadata.html>

## 2.3 HTML HEAD ELEMENTE IN <HEAD> (II)

URL des CSS



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

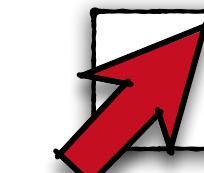
```
<link rel="stylesheet" href="lib/css/style.css">
```

```
<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
- rel-Attribut beschreibt Verhältnis zwischen der einbindenden oder der eingebundenen Ressource
- media-Attribut zusammen mit stylesheet spezifiziert Ausgabegerät für das 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.



<https://www.w3.org/TR/html/document-metadata.html>

## 2.3 HTML HEAD VERLINKUNGEN (I)

nicht links im Text/Body, sondern  
Stylesheets etc.

Beziehung zwischen Dokumenten (im Head)

```
<link rel="stylesheet" href="selfhtml.css" type="text/css"></link>
```

Klickbarer Link (im Body) → darauf kann der User anklicken

```
<a href="https://snet.tu-berlin.de">Willkommen zu Webtechnologien</a>
```

- HTML stellt zwei Möglichkeiten bereit, Beziehungen zwischen Dokumenten zu beschreiben:
  - <a>-Element (Anchor) im Body eines HTML-Dokuments führt einen klickbaren Verweis im Dokument ein
  - <link>-Element im Head beschreibt eine Beziehung zwischen dem Dokument, welches den Link enthält, und einem anderen Dokument sowie die Art dieser Beziehung
- Unterschied: <a> liefert einen für den Nutzer sichtbaren Verweis, <link> ist für den Nutzer unsichtbar und dient der Steuerung des Browsers
- Attribute in <a> und <link>
  - href: URI des Zieldokuments
  - rel: Art der Beziehung
  - media: Ausgabemedium für welches das Zieldokument bestimmt ist
  - type: MIME-Type des Zieldokuments

## 2.3 HTML HEAD VERLINKUNGEN (II)

```
<link rel="icon" href="favicon.ico" type="image/vnd.microsoft.icon" />
```

Beispiel: Favicons einbinden

```
<link rel="canonical" title="weiterleitung" href="../neue-seite.html">
<link rel="contents" title="Inhaltsverzeichnis" href="....navigation/inhalt.htm">
<link rel="index" title="Stichwortverzeichnis" href="....navigation/stichwort.htm">
<link rel="help" title="Hilfe" href=".../editorial/index.htm">
<link rel="search" title="Suche" href="....navigation/suche/index.htm">
<link rel="top" title="SELFHTML" href="....index.htm">
<link rel="up" title="HTML-Kopfdaten" href="index.htm">
<link rel="next" title="Durchsuchbarkeit mit Server-Kommunikation" href="durchsuchbarkeit.htm">
<link rel="prev" title="Adressbasis und Zielfensterbasis" href="basis.htm">
<link rel="first" title="Titel einer HTML-Datei" href="titel.htm">
<link rel="last" title="Durchsuchbarkeit mit Server-Kommunikation" href="durchsuchbarkeit.htm">
```

Beispiel: Logische Beziehungen festlegen

```
<link rel="canonical" href="https://example.com/content.html">
```

Beispiel: Kanonische Links

- Oftmals sind Webseiten unter mehreren Links erreichbar, aber nur eine bestimmte Seite soll von einer Suchmaschine indiziert werden
- Um Indizierung einer falschen Seite zu vermeiden, setzt man mit dem `<link>`-Element eine kanonische Beziehung zu der zu indizierenden Seite

# 2.4 HTML BODY

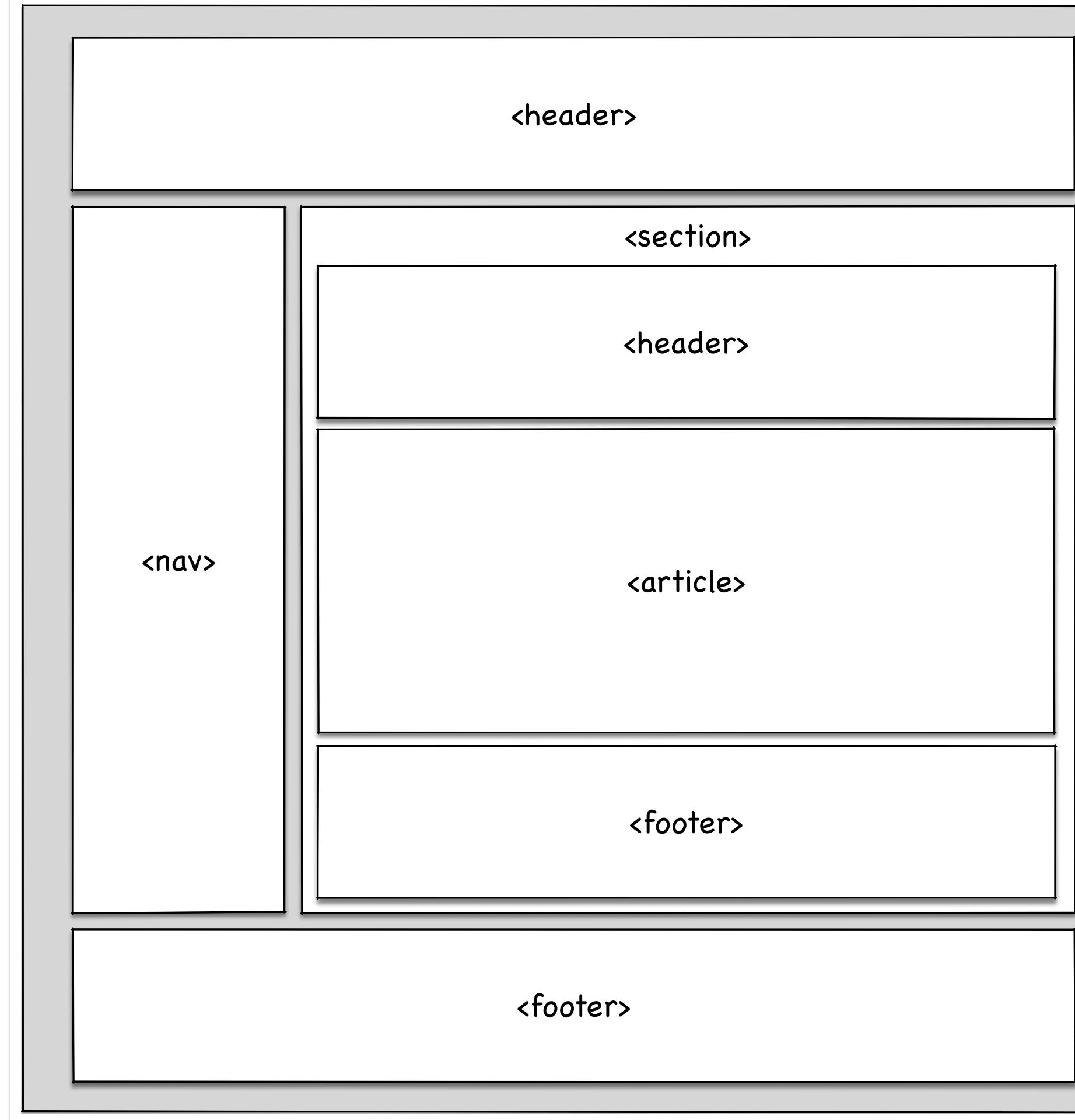
## HTML-ELEMENTE ZUR SEITENSTRUKTURIERUNG

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

Element	Funktion	ab HTML 5
<code>body</code>	Umschließt den sichtbaren Inhalt einer Webseite.	
<code>section</code>	Definiert einen allgemeinen Bereich einer Seite, in dem zusammengehöriger Inhalt gebündelt wird.	Ja
<code>nav</code>	Definiert Navigationsbereiche.	Ja
<code>article</code>	Definiert einen Artikel mit zusammengehörigen Informationen, der grundsätzlich auch allein stehen könnte und von dem umgebenden Kontext unabhängig ist.	Ja
<code>aside</code>	Definiert Bereiche, die verwandte Themen zum Hauptinhalt zeigen, grundsätzlich aber auch ohne Bezug auf den Inhalt eine eigenständige Information liefern.	Ja
<code>h1</code> , <code>h2</code> , <code>h3</code> , <code>h4</code> , <code>h5</code> , <code>h6</code>	Überschriften für einen Seitenbereich.	
<code>main</code>	Repräsentiert den Hauptinhalt eines <code>&lt;body&gt;</code> -Elements und darf nicht in <code>&lt;section&gt;</code> oder <code>&lt;article&gt;</code> enthalten sein.	
<code>header</code>	Definiert einen Bereich als Kopf eines Seitenbereichs. Kann als eigentlicher Kopf einer Website eingesetzt werden, aber auch als Kopf beispielsweise einer <code>&lt;section&gt;</code> .	Ja
<code>footer</code>	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 <code>&lt;section&gt;</code> oder <code>&lt;article&gt;</code> .	Ja
<code>address</code>	Dient zur Auszeichnung eines Abschnitts mit Kontaktmöglichkeiten, z.B. Link zur Kontaktseite oder Email-Link. Telefonnummern und postalische Adressen sind laut Standard im <code>&lt;address&gt;</code> -Element nicht erlaubt.	

## 2.4 HTML BODY

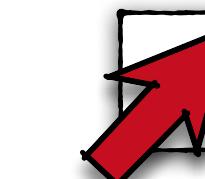
# HTML-ELEMENTE ZUR SEITENSTRUKTURIERUNG - BEISPIEL



- Allgemeine Inhaltsbereiche eines Dokuments werden durch `<section>`-Elemente dargestellt.
- Ziel: thematische Gruppierung von Inhalten typischerweise mit einer Überschrift
- `<article>`-Elemente stellen in sich geschlossene Abschnitte eines Dokuments dar, vergleichbar mit einem Zeitungsartikel
- `<article>` können in `<section>` eingebettet werden, beispielsweise verschiedene Kommentare zu einem Zeitungsartikel

```
<article>
  <header>
    <address>Autor: <a href="...">Max Mustermann</a></address>
    <h1>Warum soll ich bloggen?</h1>
  </header>
  <p>Früher, als die Weblogs noch Internet-Tagebücher hießen, ...</p>
</article>

<section>
  <article>
    <h1>Kommentar von Mario</h1>
    <p>Gutes Argument. Es muss nicht jeder alles wissen.</p>
  </article>
</section>
```



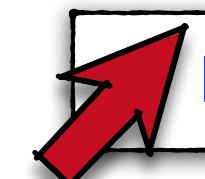
<https://www.w3.org/TR/html/sections.html#sections>

# 2.4 HTML BODY

## HTML-ELEMENTE ZUR TEXTSTRUKTURIERUNG

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

Element	Funktion	ab HTML 5
<code>p</code>	Definiert einen Textblock (Paragraf).	
<code>hr</code>	Definiert eine Trennlinie, die Inhalte in logische Abschnitte unterteilt.	
<code>pre</code>	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.	
<code>blockquote</code>	Definiert einen Abschnitt als Zitat. Wird wie das <code>q</code> -Tag nicht mit Anführungszeichen, aber eingerückt dargestellt.	
<code>ol</code>	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).	
<code>ul</code>	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).	
<code>li</code>	Definiert einen Listeneintrag sowohl in einer <code>ol</code> als auch in einer <code>ul</code> .	
<code>figure</code>	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
<code>figcaption</code>	Definiert die Beschriftung einer <code>figure</code> .	Ja
<code>div</code>	Blockelement ohne jegliche strukturelle Bedeutung. Sollte nur eingesetzt werden, wenn keine anderen Strukturelemente für die Auszeichnung eines Abschnitts sinnvoll erscheinen.	

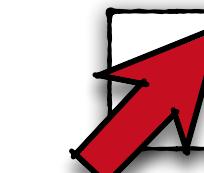


<https://www.w3.org/TR/html/grouping-content.html>

## 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: H<sub>2</sub> für H <sub>2</sub> O.	
sup	Stellt einen Text hoch. Sollte ebenfalls nur eingesetzt werden, wenn es den Konventionen einer Sprache entspricht, beispielsweise bei einer Quadratmeterangabe: 45m<sup>2</sup> für 45m <sup>2</sup>	



<https://www.w3.org/TR/html/textlevel-semantics.html>

## 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 <code>&lt;em&gt;</code> oder <code>&lt;strong&gt;</code> 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 <code>&lt;span lang="en"&gt;Meeting&lt;/span&gt;</code> , damit Screenreader diesen mit der richtigen Aussprache vorlesen.	
br	Definiert einen Zeilenumbruch innerhalb eines <code>&lt;p&gt;</code> - oder <code>&lt;address&gt;</code> -Elements. Sollte auf keinen Fall eingesetzt werden, um Abstände zwischen HTML-Elementen zu erzeugen. Sind mehrere <code>&lt;br&gt;</code> nacheinander in einem <code>&lt;p&gt;</code> -Element verwendet, ist zu prüfen, ob der relevante Text in zwei <code>&lt;p&gt;</code> -Blöcke aufgeteilt werden kann.	
wbr	Definiert optionale Zeilenumbrüche, damit lange Wörter an der richtigen Stelle getrennt werden. Beispielsweise <code>Do&lt;wbr&gt;nau&lt;wbr&gt;dampf&lt;wbr&gt;schiff&lt;wbr&gt;fahrts&lt;wbr&gt;ge&lt;wbr&gt;sell&lt;wbr&gt;schafts&lt;wbr&gt;ka&lt;wbr&gt;pi&lt;wbr&gt;täns&lt;wbr&gt;müt&lt;wbr&gt;ze.</code>	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 <form>-Tag wird die URL angegeben, an die ein Formular gesendet wird, sowie der Übertragungsmodus (get , post) festgelegt. Ebenfalls wird über den enctype der Übertragungsmodus festgelegt, der im Falle eines Dateiuploads den Wert multipart/form-data haben muss.	
fieldset	Gliedert ein Formular und fasst inhaltlich zusammengehörige Formularfelder zusammen.	
legend	Definiert eine übergeordnete Beschriftung von zusammengehörigen Formularfeldern, die über <fieldset> zu einer Einheit zusammengefasst werden.	
label	Definiert eine eindeutige Beschriftung für jedes einzelne Eingabefeld. Jedem <label> kann nur ein Formularelement (<input>, <select>, <textarea>) zugeordnet werden. Die Verknüpfung wird über das for-Attribut des <label> und das id-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 <input>-Typen und damit die Datenarten, die damit erfasst werden können, wesentlich erhöht.	Ja (viele neue Werte für type).
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 <select>-Box.	
optgroup	Gruppiert mehrere <option>-Tags innerhalb einer <select>-Box zu einer Einheit von Auswahlmöglichkeiten.	

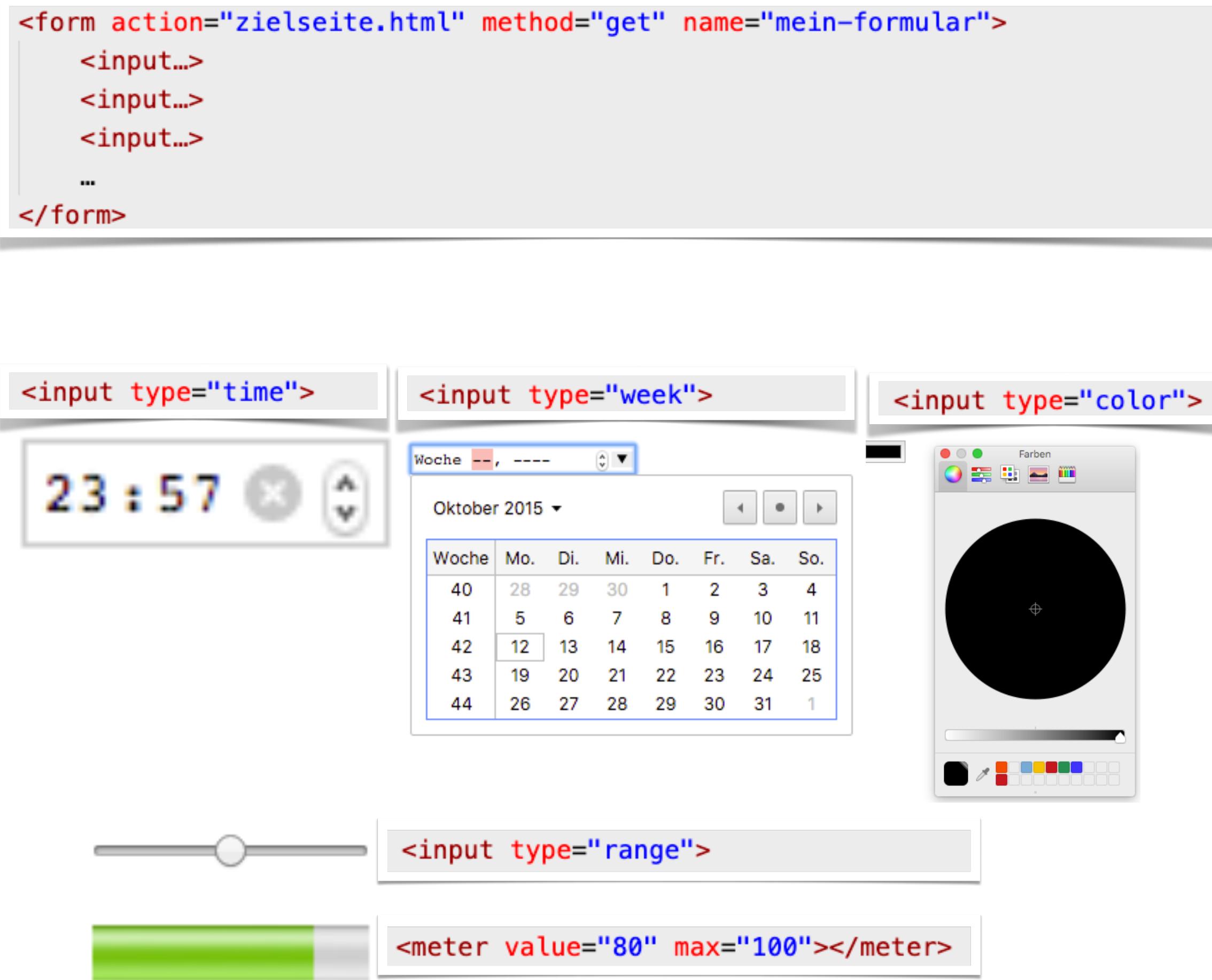
## 2.4 HTML BODY

# 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 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.	

## 2.4 HTML BODY

# HTML-ELEMENTE FÜR FORMULARDARSTELLUNG (III)



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



<https://www.w3.org/TR/html/sec-forms.html>

## 2.4 HTML BODY

# FORMULARDATEN VERSENDEN - GET VERSUS POST

### method="get"

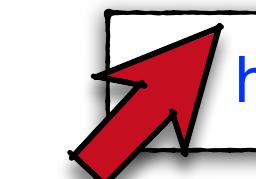
- Formulareingabe wird mittels einer HTTP-GET-Methode gesendet, d.h. die Daten werden an die URL angehängt, die in action angegeben ist
- Nach dem Absenden des Formulars sind die Formulareingaben in der Browserzeile sichtbar
- HTTP empfiehlt eine Maximalgröße von 255 Bytes für URLs, daher nur für kleine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

```
GET /test/demo_form.asp?name1=value1&name2=value2
```

### method="post"

- Übertragung der Daten im Nutzdatenteil einer Nachricht, die mit der HTTP-POST-Methode versendet wird
- Übertragung der Daten ist für den Nutzer nicht sichtbar
- Je nach Konfiguration des Webservers kann die maximale Dateigröße beschränkt sein, z.B. 8 MB

```
POST /test/demo_form.asp HTTP/1.1
Host: www.snet.tu-berlin.de
name1=value1&name2=value2
```



<https://www.w3.org/TR/html/sec-forms.html>

# 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

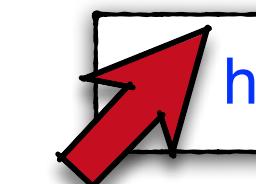
Attribut	Funktion
controls	Ergänzt Video um eine Bedienleiste, deren Layout über CSS und JavaScript angepasst werden kann angepasst
autoplay	Startet das Video automatisch beim Laden der Seite
loop	Spielt das Video in einer Endlosschleife
poster	Referenziert eine Grafik die gezeigt wird wenn das Video 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 (mögliche Werte: auto, metadata, none)

```
<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>
```

- HTML5 ermöglicht Einbindung von Audio- und Videoinhalten ohne Plug-Ins mittels `<audio>`- und `<video>`-Elementen

### BEISPIEL `<video>`

- Einbindung von Videos unterschiedlicher Kodierungen mit vielen Konfigurationsmöglichkeiten
- Browser wählt eine kompatible Kodierung
- Wenn Browser Kodierungen nicht unterstützt, erfolgt ein Hinweis zusammen mit einem Link zum Download



<https://www.w3.org/TR/html/semantics-embedded-content.html>

## 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>, <th> 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 Datenzelle innerhalb einer Tabelle.	

## 2.5 HTML-BEISPIEL ÜBERBLICK

The screenshot shows a web browser window with the URL `/chapter02/example5.html` in the address bar. The page title is "YASN" and the sub-headline is "Yet Another Social Network". The form consists of four main sections:

- 1. Registriere Dich bei YASN**:  
- Benutzername:  (highlighted with a blue border)  
- Mobiltelefon:   
- Mailadresse:   
- Wohnort:
- 2. Lade Deine Freunde ein (Mailadressen angeben):**  
Mehrere Mailadressen durch Komma trennen
- 3. Vervollständige Dein Profil:**  
- Homepage:   
- Geburtsdatum:  (format: tt.mm.jjjj)  
- Geburtsort:  (placeholder: Gib eine Stadt ein)
- 4. Persönliche Einstellungen:**  
Welche Hintergrundfarbe soll Dein Profil haben?

At the bottom left is a **Absenden** button.

- Beispiel eines Webformulars mit verschiedenen Arten von Formularfeldern
- Achtung: nicht alle Formularfelder werden von allen Browsern unterstützt, Beispiel basiert auf Chrome

## 2.5 HTML-BEISPIEL FORMULARE UND LISTEN

```
<form action="zielseite.html" method="post" name="mein-formular">
  <ol>
    <li>Registriere Dich bei YASN
    ...
    <li>Lade Deine Freunde ein (Mailadressen angeben):
    ...
    <li>Vervollständige Dein Profil:
    ...
    <li>Persönliche Einstellungen:
  </ol>
  <input type="submit" value="Absenden"></input>
</form>
```

- Jedes Formularfelder wird durch ein `<input>`-Element zusammen mit spezifizierenden Attributen repräsentiert
- Alle Formularfelder sind in einem `<form>` eingeschlossen
- Der Absenden-Button bildet meistens das letzte Formularfeld eines Formulars
- Im Beispiel ist das Formular mit Hilfe einer Liste strukturiert
- `<ol>` eröffnet eine geordnete Liste (*Order List*), d.h. die Liste wird nummeriert
- `<ul>` eröffnet eine ungeordnete Liste (*Unordered List*) mit Aufzählungspunkten (Bullet Points)
- `<li>` eröffnet ein neues Element der Liste (*List Element*)

## 2.5 HTML-BEISPIEL

### TABELLEN UND TEXTEINGABEFELDER

```
<table>
  <tr><td>Benutzername:<td><input ...>
  <tr><td>Mobiltelefon:<td><input ...>
  <tr><td>Mailadresse:<td><input ...>
  <tr><td>Wohnort:<td><input ...>
</table>
```

```
<tr><td>Benutzername:
  <td><input name="username"
    type="text"
    size="20"
    maxlength="20"
    required
    autofocus></input>
```

- Für eine bündige Anordnung können (vorläufig) Tabellen genutzt werden
- Eine Tabelle wird mit dem `<table>`-Element eröffnet
- `<tr>` eröffnet eine neue Reihe (*Table Row*)
- `<td>` eröffnet eine neue Zelle für Daten (*Table Data*) innerhalb einer Reihe
- Felder für die Eingabe von Text (`type="text"`) sind die am häufigsten verwendeten Formularfelder
- Übermittlung durch Key/Value-Pairs: `name` spezifiziert die Bezeichnung des Keys für die Übertragung des Wertes
- `size` ist die Größe des Eingabefeldes
- `maxlength` ist die maximale Länge der Eingabe
- `required` erzwingt eine Eingabe durch den Nutzer
- Formularfeld mit `autofocus`-Attribut erhält den Fokus nach dem Laden der Seite

## 2.5 HTML-BEISPIEL

### VERSCHIEDENE FELDTYPEN (I)

```
<input type="tel"  
      name="telefon"  
      required="required"></input>  
  
<input type="email"  
      name="mailadresse"  
      required="required"></input>
```

```
<input type="text" name="wohnort" list="cities">  
  <datalist id="cities">  
    <select>  
      <option value="Hamburg">Hamburg</option>  
      <option value="Hannover">Hannover</option>  
      <option value="Bochum">Bochum</option>  
      <option value="Berlin">Berlin</option>  
      <option value="München">München</option>  
    </select>  
  </datalist>  
</input>
```

- Formularfelder mit Syntaxcheck: tel, email oder url erzwingen die Eingabe gemäß eines bestimmten Musters
- Beispiel: Wert für ein Feld vom Typ email mit @-Zeichen
- Alternativen: Syntaxcheck...
  - ... durch JavaScript
  - ... durch jQuery
  - ... im Webserver
  - ... eine Kombination aus Frontend und Webserver
- Spezifikation von Auswahllisten (Auto Complete) für Textfelder mittels list-Attribut
- Auswahl wird mit <datalist> und <select>-Elementen in <input> eingebettet
- Alternativen: Einblendung von Auswahllisten...
  - ... durch JavaScript
  - ... durch jQuery
  - ... dynamisches Laden vom Webserver während der Eingabe

## 2.5 HTML-BEISPIEL VERSCHIEDENE FELDTYPEN (II)

The block contains four screenshots demonstrating various HTML input types:

- Email Input:** Shows an input field with type="email", placeholder="Mehrere Mailadressen durch Komma trennen", size="35", name="mailadressen", and multiple="multiple".
- Date Input:** Shows an input field with type="date", name="geburstag", max="1998-10-01", and min="1900-01-01". Below it is a calendar picker showing October 1998.
- Color Input:** Shows an input field with type="color", name="hintergrundfarbe". It includes a color palette and a preview box showing "Limone".
- File Input:** Shows an input field with type="file". A file dialog is open, showing a list of files including "Screenshot 2022-09-27 11.45.40.png".

- Durch `placeholder` können Texte in Eingabefelder geschrieben werden, zum Beispiel Anweisungen oder Beispiele
- Platzhaltertexte werden bei Eingabe überschrieben
- Platzhaltertexte werden nicht an den Webserver gesendet, wenn keine Eingabe des Feldes erfolgt
- `multiple` ermöglicht die Angabe mehrerer Werte durch Trennung mit Kommas

- Felder vom Typ `date` oder `color` unterstützen (häufig) die Eingabe durch grafische Elemente
- Anpassung der Elemente durch zusätzliche Attribute, zum Beispiel `min` oder `max` zur Einschränkung eines Datums

## 2.A XML IN A NUTSHELL

### XML-DEKLARATION UND -ELEMENTE

The diagram illustrates the structure of an XML document. It shows a code block with red annotations pointing to specific parts:

- A red bracket on the left labeled "Deklaration" points to the first two lines of the XML code: `<?xml version="1.0" encoding="UTF-8"?>` and `<!- File name: BillingInformation.xml -->`.
- A red bracket on the left labeled "Wurzel-element" points to the root element `<BillingInformation ...>`.
- A red line connects the attribute `customer-type="manufacturer"` to the label "Attribut".
- A red bracket on the right labeled "Einge-bettete Elemente" groups the child elements of the root element: `<Name> Right Plastic Products </Name>`, `<BillingDate> 2002-09-15 </BillingDate>`, `<Address>` (with its children `<Street> 158 Edward st. </Street>`, `<City> Brisbane </City>`, `<State> QLD </State>`, and `<PostalCode> 4000 </PostalCode>`), and `</Address>`.

```
<?xml version="1.0" encoding="UTF-8"?>
<!- File name: BillingInformation.xml -->

<BillingInformation customer-type="manufacturer">
    <Name> Right Plastic Products </Name>
    <BillingDate> 2002-09-15 </BillingDate>
    <Address>
        <Street> 158 Edward st. </Street>
        <City> Brisbane </City>
        <State> QLD </State>
        <PostalCode> 4000 </PostalCode>
    </Address>
</BillingInformation>
```

### XML-DEKLARATION

- Erste Zeile eines XML-Dokuments
- Wird von der verarbeitenden Software benötigt, um zu bestimmten, wie nachfolgende XML-Inhalte verarbeitet werden
- Definiert Versionsnummer, Zeichencodierung usw.

### ELEMENTE

- Ein XML-Dokument besteht aus einem oder mehreren Elementen
- Elemente enthalten andere Elemente, Text, Attribute oder eine Kombination von alldem
- Elemente werden durch einen Start- und End-Tag mit spitzen Klammern gekennzeichnet
- Elemente (und auch Attribute) sollten beschreibende Namen haben, damit die Inhalte eines Dokumentes intuitiv verständlich sind

# 2.A XML IN A NUTSHELL

## BENENNUNGSREGLEN UND AUFWÄRTSKOMPATIBILITÄT

### INITIALE VERSION DES XML-ELEMENTS <NOTE>

```
<note>
  <to>Tove</to>
  <from>Jani</from>
  <body>Don't forget me this weekend!</body>
</note>
```

### ERWEITERTE VERSION DES XML-ELEMENT <NOTE>

```
<note>
  <date>2008-01-10</date>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

### XML BENENNUNGSREGELN

- Namen können Buchstaben, Zahlen oder andere Zeichen enthalten
- Namen dürfen nicht mit einer Zahl oder einem Interpunktionszeichen starten
- Namen dürfen nicht mit dem Präfix XML, Xml, xml, usw. beginnen
- Namen dürfen keine Leerzeichen enthalten
- Jeder Name kann verwendet werden, keine Schlüsselwörter

### XML-ELEMENTE SIND ERWEITERBAR

- Eine auf XML-basierende Sprache kann um neue Elemente erweitert werden, ohne dass die Software, die auf der vorherigen Version basiert, unbrauchbar wird
- Unbekannte Elemente werden ignoriert
- Eine Anwendung, die für XML-Dokument (a) geschrieben wurde, ist bezüglich der Syntax aus (a) aufwärtskompatibel, wird also auch auf (b) ausgeführt, ohne abzustürzen

## 2.A XML IN A NUTSHELL

### ATTRIBUTES

The diagram shows an XML document with red annotations:

- A red bracket labeled "Deklaration" points to the XML declaration: <?xml version="1.0" encoding="UTF-8"?>
- A red bracket labeled "Wurzel-element" points to the root element: <BillingInformation>
- A red bracket labeled "Attribut" points to the attribute: customer-type="manufacturer" on the root element.
- A red bracket labeled "Einge-bettete Elemente" points to the child elements of the root element: <Name>, <BillingDate>, <Address>, and </BillingInformation>.

```
<?xml version="1.0" encoding="UTF-8"?>
<!- File name: BillingInformation.xml -->

<BillingInformation customer-type="manufacturer">
    <Name> Right Plastic Products </Name>
    <BillingDate> 2002-09-15 </BillingDate>
    <Address>
        <Street> 158 Edward st. </Street>
        <City> Brisbane </City>
        <State> QLD </State>
        <PostalCode> 4000 </PostalCode>
    </Address>
</BillingInformation>
```

- Attribute werden verwendet, um Elemente mit zusätzlichen Steuerinformationen zu annotieren, die von der parsenden Software interpretiert werden
- Werden dem Start-Tag als Schlüssel/Wert-Paar (Key/Value-Pair) hinzugefügt
- Der Wert muss in einfachen oder doppelten Anführungszeichen angegeben werden
- Attribute können nicht mehrere Werte enthalten oder geschachtelt sein
- Attribute sind nicht im Sinne von Elementen erweiterbar (siehe letzte Folie)
- Keine vorgegebenen Regeln wann Elemente oder Attribute zu verwenden sind
- Empfehlung: Attribute sind schwer lesbar - verwende Elemente für die Annotation von Daten, verwende Attribute für zusätzliche Steuerinformationen



## VORLESUNG

Prof. Dr. Axel Küpper

TU Berlin | T-Labs | Fachgebiet *Service-centric Networking*  
Ernst-Reuter-Platz 7 | 10587 Berlin | Germany

 [axel.kuepper@tu-berlin.de](mailto:axel.kuepper@tu-berlin.de)

 <https://twitter.com/kuepp>

 <https://www.linkedin.com/in/axelkuepper/>

 <http://www.snet.tu-berlin.de/kuepper>

## ÜBUNGSLEITER

- Thomas Cory
- Sanjeet Raj Pandey
- Christian René Sechting

## TUTOREN

- Nastassia Lukyanovich
- Maximilian Oliver Fisch
- Leonhardt Frederik Hollatz
- Adrian Siebing