



HTML5

[1]

Referat: HTML5

Referenten: Wioletta Cemerika, Alexander Kunze, Tobias Fielitz

Seminar: Moderne Webtechnologien

AG NBI, Institut Informatik, FU Berlin

28.10.2009

Beispiel HTML5

```
<div id="header">
```

```
<div id="nav">
```

```
<div class="article">
```

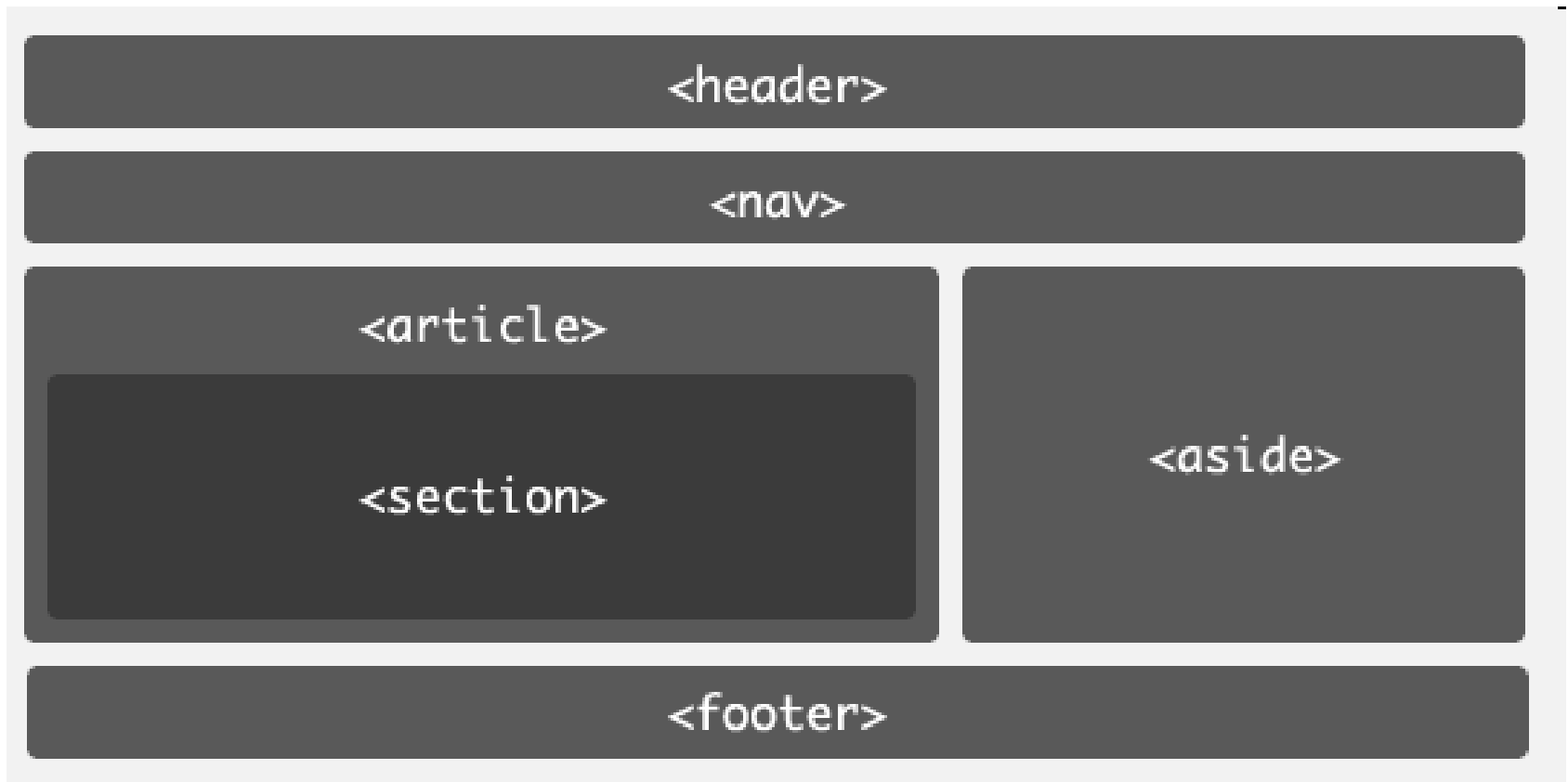
```
<div class="section">
```

```
<div id="sidebar">
```

```
<div id="footer">
```

Quelle: [2]

Beispiel HTML5



Quelle: [2]

Agenda

1. Einführung und Übersicht
2. Ziele von HTML5
3. Neue Features und Veränderungen zu HTML4
4. Formulare
5. Was geht schon und wann geht's los?
6. Literatur und Quellen

1. Einführung und Übersicht

1.1 Allgemein über HTML. [3]

1.2 HTML und seine Geschichte. [4][5]

1.3 HTML und die große SGML-Familie. [6]

1.4 DOM, seine Funktionen und Regeln.[6][7]

1.5 Vor- und Nachteile.

1.1 Allgemein über HTML.

- eine Auszeichnungssprache
- *HyperText Markup Language*
- HTML Dokumente stellen Grundlage im WWW
- Webbrowser: lesen, interpretieren und machen HTML Dokumente sichtbar.



Quellen: [8][9][10][11][12][13]

1.2 HTML und seine Geschichte - 1.

- 1989 Tim Berners-Lee startet Entwicklung von HTML
- 1993 der erste Browser – Mosaic
- 1994 HTML 2 als Spezifikation übernommen
- 1994 W3C wird ins Leben gerufen
- 1995 Erweiterung um neue Tags und Attribute
Veröffentlichung des HTML 2- Standards, enthält
Befehle zur Internationalisierung des Internets

1.2 HTML und seine Geschichte - 2.

- 1996 W3C präsentiert neuen Entwurf, in dem Java Skript zum HTML - Standard wird;
und CSS v 1.0 erscheint, dadurch können einzelne HTML- Befehle definiert werden
- 1997 erscheint HTML 3.2 Standard
- 1997 wird HTML 4 standardisiert
- 1998 überarbeitet Version des HTML 4 Standards

1.2 HTML und seine Geschichte - 3.

1999 aktuelle HTML 4.01 liegt vor, Spezifikation ist im Internet :

www.W3.Org/TR/html40

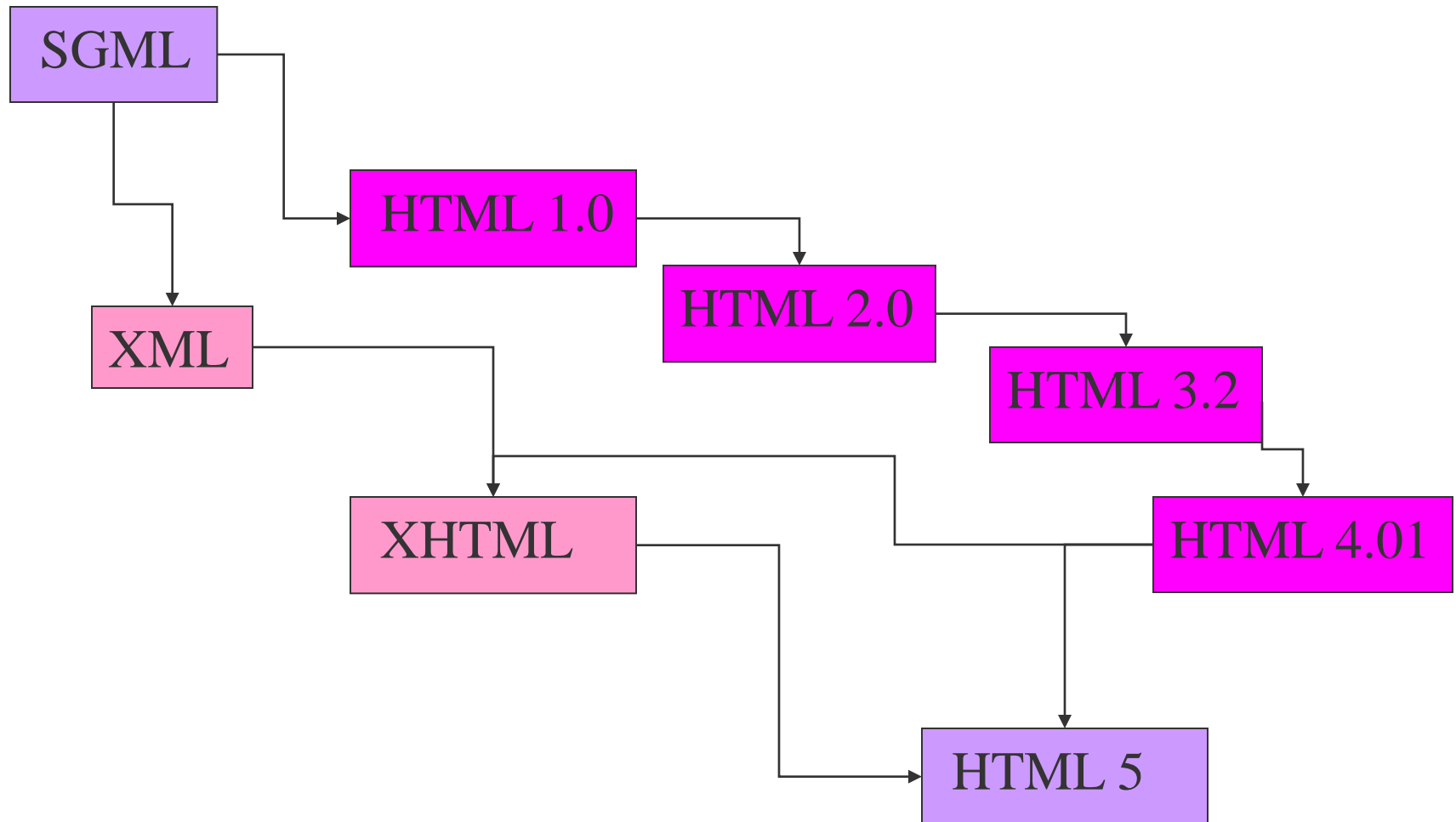
2004 WHATWG *Web Hypertext Applications Technology Working Group* veröffentlicht Web Applications 1

2006 Tim Berners-Lee und seine Arbeitsgruppe arbeitet an XHTML 2

1.2 HTML und seine Geschichte - 4.

- 2007 WHATWG setzt sich durch, und der Ansatz von Web Applications 1 soll die Grundlage für HTML 5 Weiterentwicklung dienen
- 2008 W3C legt die erste Working Draft HTML 5 Spezifikation vor

1.3 HTML und die große SGML-Familie



1.4 DOM, seine Funktionen und Regeln

DOM *Document Objekt Model* :

- W3C Standard
- unabhängig von Programmiersprachen
- abstrakte Schnittstelle zum Zugreifen auf ein Modell
- logisches Modell zur Datendarstellung
- Einschnitt - Pull - Parser

1.4 DOM, seine Funktionen und Regeln

- unterschiedliche DOM Versionen (Levels) mit verschiedenen Modulen
- am Anfang schwache Unterstützung für HTML ergänzende Skriptsprachen
- durch Verbreitung von HTML - Weiterentwicklung von „DOMCore“ und „DOMHtml“

1.4 DOM, seine Funktionen und Regeln

DOMCore:

- definiert Schnittstellen
- abstrakte Klassen, legt Klassenstrukturen und Methoden fest
- Wichtig: Knoten nur in einem Dokumentenkontext gültig

1.4 DOMCore

- Node ist eine Basisklasse für einen Knoten

- Methoden von Node :

 - `getNodeTypes();`

 - `getChildNodes();`

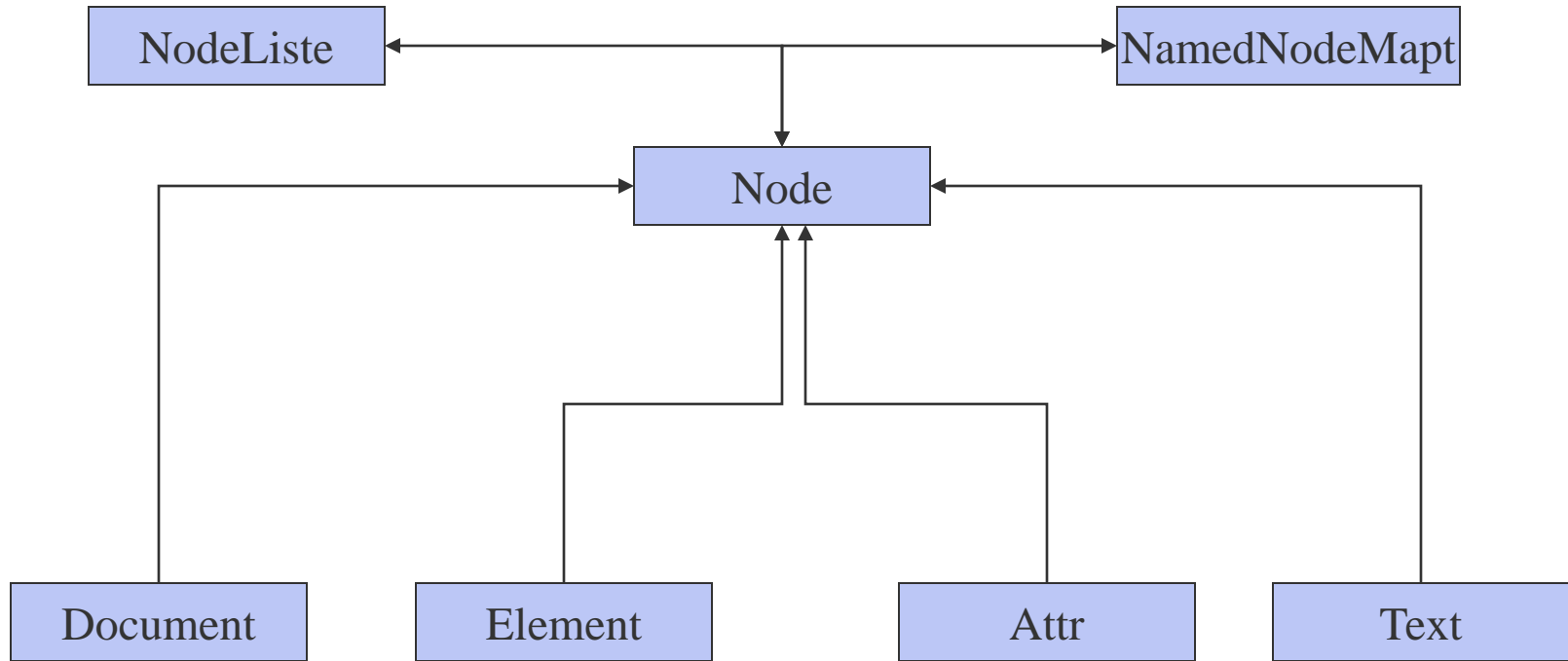
 - `getAttributes();`

- Document enthält Methoden zum Erstellen von Knoten:

 - `createElement();`

 - `createAttribute();`

1.4 DOMCore

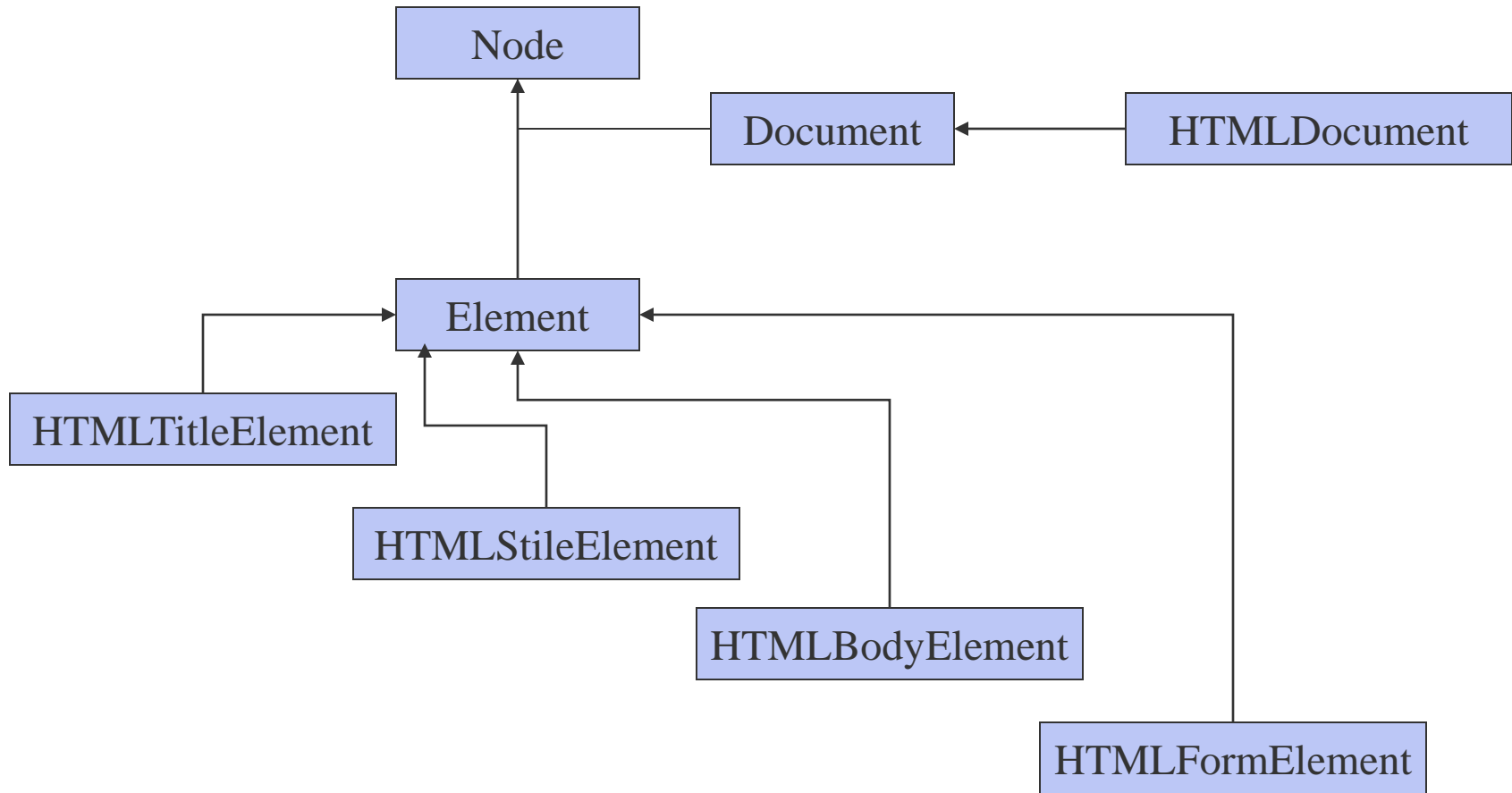


1.4 DOM, seine Funktionen und Regeln

DOMHtml:

- basiert auf der DOMCore - Entwicklung
- HTMLDocument beinhaltet zusätzliche Informationen über HTML Dokument:
 - String titel,
 - String URL, ...
- Zugriff auf Bilder, andere Formulare und Elemente

1.4 DOMHtml



1.5 Vor- und Nachteile

Vorteile:

- + ein einfacher Texteditor reicht aus
- + das Erlernen der Programmiersprache ist einfach
- + der Quellcode ist zugänglich und somit lernt man von anderen HTML -Designern
- + flexible Bearbeitung einer Seite

1.5 Vor- und Nachteile

Nachteile

- die Darstellung von Seiten durch Browser unterschiedlich
- Erstellung von z.B. Gleichungen sehr umständlich
- kein Schutz von Seiten

2. Ziele von HTML5

Update der HTML-Recommendation von 1997

- Seitdem eher Browser- statt Spracherweiterungen

Mehr Semantik

- Dadurch geeigneter für Mobile Web

Mehr Accessibility

- (z.B. Sektionen vergrößern/vorlesen)

Neue APIs

- (z.B. 2D Zeichnen auf Canvas, API Video- und Audiowiedergabe,
- API für Offline Applications, API für Editing, API für Draggable)

HTML5-Zielgruppen:

- UserAgents vs. WebAuthors

3. Neue Features und Veränderungen zu HTML4

3.1 MathML, SVG

3.2 Canvas, Drag&Drop

3.3 Offline Applications

3.4 Neue Tags

3.5 Neue Attribute

3.6 Veränderte Tags und Attribute

3.7 Obsolete Tags und Attribute

3.1 MathML & SVG

MathML & SVG nun inline im Dokument möglich

-schachtelbar,

-z.B. ((HTML & MathML) in SVG) in HTML

Bsp:

```
<!doctype html>
```

```
<title>SVG in text/html</title>
```

```
<p> A green circle:
```

```
  <svg>
```

```
    <circle r="50" cx="50" cy="50" fill="green"/>
```

```
  </svg>
```

```
</p>
```

3.2 Canvas, Drag&Drop, contentEditable

Canvas Element *definiert Zeichenbereich*

- als statische bitmap oder dynamisch veränderbar
- Viele grafische Operationen dank neuer 2D-Drawing API

Universalattribut **draggable** erlaubt *Drag&Drop*

- Transfer von Dateien und Daten ist möglich, z.B. via **href**
- Diverse Events für Beginn, Bewegung, Ende und Aktionen

Universalattribut **contenteditable** erlaubt *Editieren von Content*

3.3 Offline Web Applications (neue API)

Browser bildet „Application Cache“ aus benötigten Dateien
-Manifest-Deklaration cached Dateien für Offline-Nutzung

Bsp: Datei: clock.html

```
<!DOCTYPE HTML>
<html manifest="clock.manifest">
<head>
  <title>Clock</title>
  <script src="clock.js"></script>
  <link rel="stylesheet" href="clock.css">
</head>
<body>
  <p>The time is: <output id="clock"></output></p>
</body>
</html>
```

Datei: clock.manifest

```
CACHE MANIFEST
clock.html
clock.css
clock.js
NETWORK:
comm.cgi
```

3.4 Neue Tags: Semantische Sektionen

Semantische Struktur des Dokuments wird deutlicher:
Sinn: Semantik, Vorlesen, mobile Geräte, einfache Struktur

- **nav**, **header**, **footer** deklarieren zweckgebundene Bereiche
- **section** strukturiert Dokument. Darin Header mit **hgroup**
- **article** ist unabhängiges Element (z.B. Blogeintrag)
- **figure** ist beschriftetes, einbettbares Objekt/Abbildung
- **aside** definiert Nebenbemerkung, die Textfluss verlässt

3.4 Neue Tags: Menu, Command, Button

Menu und **command**, hier via **button** (alt: **a**, **input** , **option**)

- Menüs schachtelbar und kontextabhängig
- Darstellung abhängig von UserAgent (alt: bulleted List)
- > kein browserabhängiges CSS/JavaScript mehr nötig

Bsp:

```
<menu type="toolbar">
```

```
<li>
```

```
<menu label="File">
```

```
<button type="button" onclick="fnew()">New...</button>
```

```
<button type="button" onclick="fopen()">Open...</button>
```

```
<button type="button" onclick="fsave()" id="save">Save</button>
```

```
<button type="button" onclick="fsaveas()">Save as...</button>
```

```
</menu>
```

```
</li>
```

```
...
```

```
</menu>
```



3.4 Neue Tags: Video, Audio

Embedded Content mit `video`, `audio`, (`figure`)

- Keine umständlichen `object/embed` Schachtelungen mehr
- Keine proprietären Lösungen mehr auf Webseiten?
 - (Flash, Quicktime, Windows Media)
- Welche Formate sich durchsetzen, wird die Zeit zeigen
 - Streit zugange zwischen Ogg Theora und H.264 (Lizenzen) [14]

Bsp:

```
<video src="http://bla.com/movie.mov">
```

```
<audio src="http://uebellaut.de/droehn.wav" autoplay="true" loop="200">
```

3.4 Neue Tags/Attribute: Diverses

meter zeigt messbaren Wert mit Kontext an

```
<p>Your score <meter value="88.7" min="0" max="100" low="32" high="96",  
optimum="100">B+</meter>. </p>
```

progress zeigt den Fortschritt einer Operation

```
<p>DL: <progress value="120430" max="8949859">7%</progress> </p>
```

mark markiert Text ohne Bedeutung, z.B. Suchergebnisse

ping URLs beim Anklicken von **a**-Links

- Methode des user Trackings, aber abschaltbar (wer es kann)

details interaktiv anforderbare Informationen/Controls

3.5 Neue Attribute

Insgesamt 54 Eventattribute der Art **onXYZ**

- Nun an nahezu allen Stellen verwendbar

Andere wichtig scheinende neue Attribute:

meta: **charset**-Attribut zum Content-Type setzen

style: **scoped**-Attribut für Gültigkeitsbereiche des CSS

ol: **reversed**-Attribut für Umkehrung der Zählung

iframe: **sandbox**-Attribut für Einschränkung seines Inhalts

- z.B. Verbot der Skriptausführung, Pop-Ups, Forms und Plugins

3.6 Veränderte Tags und Attribute

Veränderte Tags/Attribute, *meist hin zu mehr Semantik:*

b: nun „stilistisch verschoben, ohne erhöhte Bedeutung“

- (Bsp: Schlüsselwort, Produktname, „was oft fettgedruckt ist“)

- **i**: nun Worte „in anderer Stimme/Stimmung“

- (Bsp: technischer Term, Schiffsname, „was oft kursivgedruckt ist“)

- **small**: nun Kleingedrucktes

- (Bsp: rechtliche Bestimmungen, Nebenbemerkungen)

- **strong**: nun „Bedeutung, nicht mehr starke Emphase“

Diverse Umdefinitionen (meist bereit besprochen) :

- **menu**, **label**, **address** (Sektion), **a**

3.7 Obsolete Tags und Attribute

Obsolet *aus inhaltlichen Gründen*, eine Auswahl:

-**frame**, **frameset**, **applet**, **acronym**, **dir**, **isindex**

-für **html**: Attribut **version**, für **meta**: **schema**

-für **a**: **name** (**id** benutzen)

-für **object**: Attribut **classid**, **codebase** (video/audio benutzen)

-für **param**: Attribut **type**

3.7 Obsolete Tags und Attribute

Obsolet *aus gestalterischen Gründen*. **Mantra:** „nimm CSS“

-font und center (!), basefont, big, s, strike, tt, u

-keine align-, bgcolor-, background-Attribute

-für body: kein alink, vlink, link, text

-für table, td, th: kein valign, cellpadding, cellspacing

-für li, ol, ul: kein type, compact

-keine Tricks mit br clear=„all“ mehr

4. Formulare

Client – Server Interaktion

4.1 Neue input types

4.2 Neue Attribute für Forms

4.3 Pflichtfelddefinition

4.4 Formzugehörigkeit

4.5 Autofocus, Autocompletion



4.1 Neue input types

Vorteil: Syntaxüberprüfung auf Clientseite ohne Javascript.

Achtung!

Syntaxüberprüfung auf Serverseite muss nach wie vor geschehen.

„Böse Nutzer“

Alte Browser

Wie bisher - bei deaktiviertem Javascript

- tel
- url
- email
- datetime
- date
- month
- week
- time
- datetime-local
- number
- range
- color

4.1 Vergleich Javascript vs. HTML5 forms

Bisherige Lösung mit Javascript:

```
<script language="javascript">[15]
function checkEmail() {
    var email = document.getElementById('emailaddress');
    var filter = /^[a-zA-Z0-9_.-]+@(([a-zA-Z0-9-]+.)
+([a-zA-Z0-9]{2,4})+)$/;
    if (!filter.test(email.value)) {
        alert('Please provide a valid email address');
        email.focus;
        return false;
    }
}
</script>
<input type="text" id="emailaddress" name="email" />
<input type="submit" onclick="return checkEmail();" />
```

4.1 Vergleich Javascript vs. HTML5 forms

Lösung in HTML5:

```
<input type="email" name="email"/>  
<input type="submit" />
```

4.2 Neue Attribute: min, max, pattern

- min/max

```
<input type="number" min="5" max="23" step="2"
name="size" />
```

- pattern

```
<label> US ZIP Code or Canadian Postal Code:
```

```
<input pattern="([0-9]{5}(-[0-9]{4})?)|([A-Z][0-9][A-
Z]\s+[0-9][A-Z][0-9])" name="postCode"
```

```
title="US: 99999-1234; Canadian: A1B 2C3"/>
```

```
</label> [16]
```

4.2 Neue Attribute: list

- list

```
<input type="text" name="favCharacter"  
list="characters" required="required" />  
  <datalist id="characters">  
    <option value="Homer Simpson">  
    <option value="Bart">  
    <option value="Fred Flinstone">  
  </datalist>[17]
```

Der Browser filtert Elemente aus `datalist` heraus, die nicht auf den `type` des `input` Elements passen.

4.3 Pflichtfelddefinition: required, multiple

- multiple

```
<select multiple="multiple" size="5">  
  <option value = "berlin">Berlin</option>  
  <option value = "münchen">München</option>  
  <option value = "stuttgart" selected>Stuttgart</option>  
  <option value = "bonn">Bonn</option>  
  <option value = "hamburg">Hamburg</option>  
  <option value = "bremen">Bremen</option>  
  <option value = "hannover">Hannover</option>  
</select>
```

- required

```
<input type="text" name="creditcard"  
required="required" />
```

4.4 Formzugehörigkeit

- form

```
<form id="foo"></form>
```

```
<form id="bar"></form>
```

```
<input type="text" name="name" form="foo" />
```

```
<input type="text" name="name" form="bar" />
```

```
<input type="number" name="age" form="foo" />
```

```
<input type="submit" form="foo" value="submit foo" />
```

```
<input type="submit" form="bar" value="submit bar" />
```

-> verschachtelte Formulare möglich.

Anwendbar auf `input`, `select`, `fieldset`, `label`, `button`, `textarea`, `keygen`, `output`, `object`.

4.5 Attribute: autofocus

- autofocus

```
<form name="f">  
  <input name="q" autofocus="" />  
</form>
```

statt:

```
<body onload="document.f.q.focus();>...</body>
```

-> autofocus somit gezielt abschaltbar.
Nur auf sichtbare Elemente anwendbar.

4.5 Attribute: autocomplete

- autocomplete

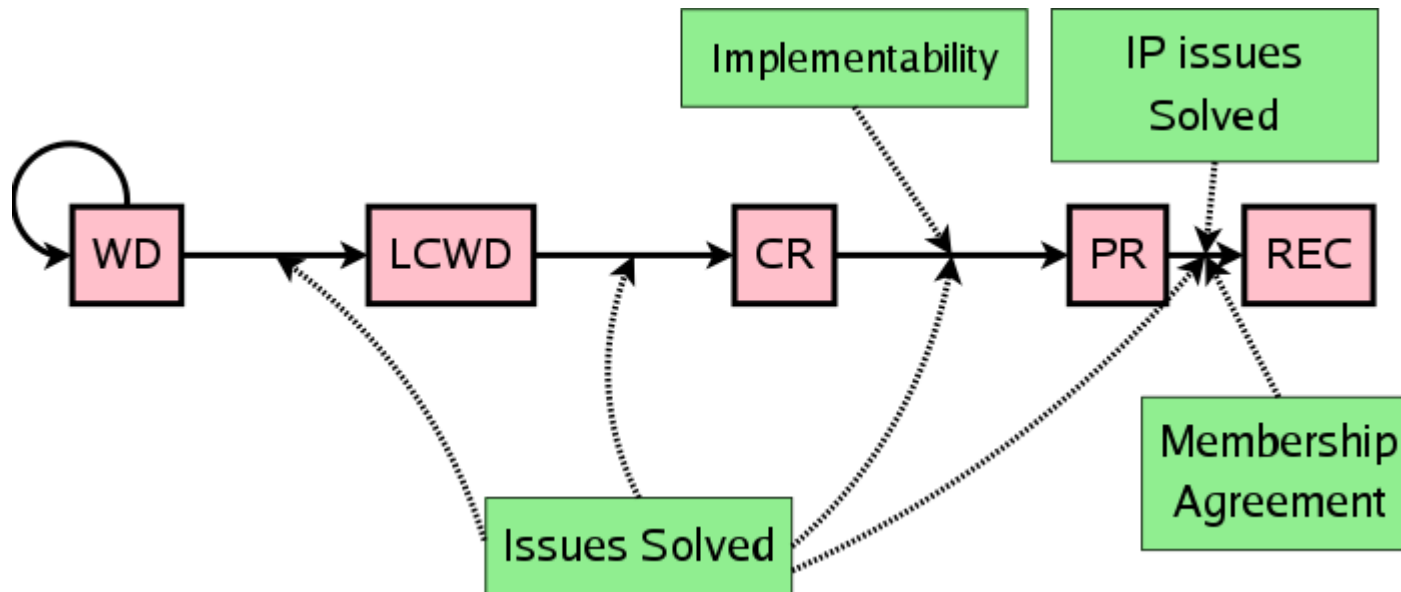
```
<input type="text" name="top_secret_username"  
      autocomplete="off" />
```

Anwendbar auf: `form`, `input`.

„The off state indicates either that the control's input data is particularly sensitive (for example the activation code for a nuclear weapon);“^[18]

5. Was geht schon und wann geht's los?

Im *W3C Recommendation Track* ist HTML5 im Working Draft:



Bildquelle: <http://www.w3.org/2005/Talks/11-maxf-w3c/>

- WD Working Draft
- LCWD Last Call Working Draft
- CR Candidate Recommendation
- PR Proposed Recommendation
- REC Recommendation

5. Was geht schon und wann geht's los?

Was können die Browser schon?

<http://ishtml5readyyet.com>

Prominente Beispiele

<http://www.youtube.com/html5>

Wie kann ich mitdiskutieren?

<http://www.w3.org/participate/discussion.html>

6. Zusammenfassung

HTML Dokumente als Grundlage im Web

HTML5 vs HTML4:

- Behält *Do-It-Yourself-Charakter* von HTML4 bei,
- Weg der formalen Strenge (XHTML) nur optional,
- Spezifikation liegt in der Hand der Browser-Hersteller,
- Befreit Web vermutlich nicht aus „Plugin-Prison“? [14]

Forms:

Erleichterung für den Programmierer

Mehr Möglichkeiten

Vielen Dank

Vielen Dank

7. Literatur und Quellen

W3C HTML5 Working Draft

<http://dev.w3.org/html5/spec/Overview.html>, eingesehen:
20.10.2009

[1] W3C keys

<http://www.w3.org/2009/Talks/0916Heraklion2-KB/scripts/keys.jpg>, eingesehen: 27.10.2009

[2] <http://www.alistapart.com/articles/previewofhtml5>,
eingesehen: 27.10.2009

[3] www.w3.org/TR/html40, eingesehen: 16.10.2009

[4] www.dejavu.de, eingesehen: 16.10.2009

[5] www.whatwg.org, eingesehen: 16.10.2009

7. Literatur und Quellen

[6] http://www.ag-nbi.de/lehre/07/V_XML/Folien/03_DTDsUndSchemata.pdf, eingesehen: 16.10.2009

[7] http://www.dpunkt.de/java/Referenz/Das_Paket_org.w3.dom/15.html, eingesehen: 16.10.2009

[8] <http://www.devx.com/assets/articlefigs/6939.gif>, eingesehen: 27.10.2009

[9] http://www.gewclan.com/webspell/design/1280/bilder//big_firefox.png, eingesehen: 27.10.2009

[10] http://www.phodana.de/wp-content/uploads/2008/07/safari_compass.jpg, eingesehen: 27.10.2009

[11] <http://thepcreport.net/wp-content/uploads/2009/08/opera-logo.jpg>, eingesehen: 27.10.2009

7. Literatur und Quellen

[12] [http://p3.focus.de/img/gen/D/0/](http://p3.focus.de/img/gen/D/0/HBD0cHFy_Pxgen_r_467xA.jpg)

HBD0cHFy_Pxgen_r_467xA.jpg, eingesehen: 27.10.2009

[13] [http://www.os-informer.de/screenshots/original/](http://www.os-informer.de/screenshots/original/2009/01/Microsoft_Internet_Explorer.jpg)

[2009/01/Microsoft_Internet_Explorer.jpg](http://www.os-informer.de/screenshots/original/2009/01/Microsoft_Internet_Explorer.jpg), eingesehen:
27.10.2009

[14] [http://lists.whatwg.org/htdig.cgi/whatwg-](http://lists.whatwg.org/htdig.cgi/whatwg-whatwg.org/2009-June/020620.html)

[whatwg.org/2009-June/020620.html](http://lists.whatwg.org/htdig.cgi/whatwg-whatwg.org/2009-June/020620.html) [whatwg] Codecs for
<audio> and <video>, eingesehen: 25.10.2009

7. Literatur und Quellen

- [15] <http://marketingtechblog.com/programming/javascript-regex-emailaddress/>, eingesehen: 20.10.2009
- [16] <http://www.alistapart.com/articles/get-ready-for-html-5/>, eingesehen: 20.10.2009
- [17] http://www.quackit.com/html_5/tags/html_datalist_tag.cfm, eingesehen: 20.10.2009
- [18] <http://dev.w3.org/html5/spec/Overview.html#autofocusing-a-form-control>, eingesehen: 20.10.2009