

HTML 5



HTML 5

Nur ein **Hype** oder die **Zukunft**?

Stand: 31.03.2011

Autor: Jan Suchandt



1. Einleitung – Gliederung (1)

1. Einleitung

1. Gliederung
2. Ziele
3. Entstehung

2. Änderungen zu HTML 4.01

1. Allgemeine Änderungen
2. Obsolete Elemente/Attribute
3. Proprietäre Elemente/Attribute

3. Strukturelle Elemente

4. Webforms 2.0

...



1. Einleitung – Gliederung (2)

4) Neue Technologien

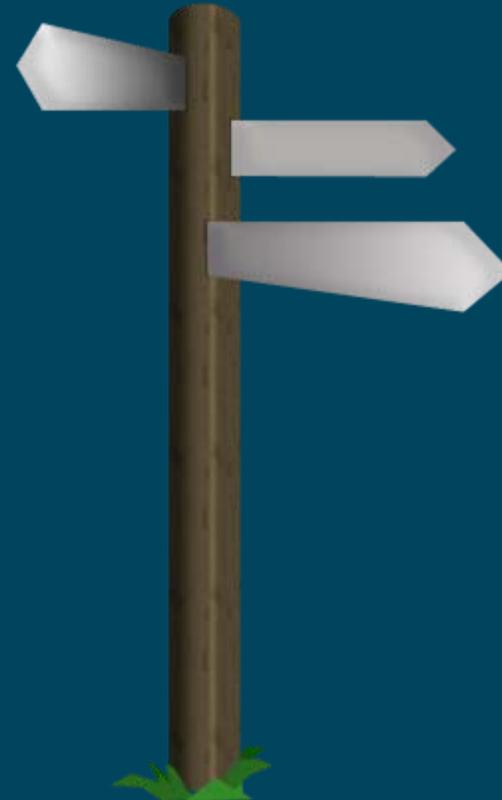
- 1) WebGL
- 2) Canvas
- 3) Audio / Video
- 4) Microdata
- 5) SVG und MathML
- 6) Cross-Document-Messaging

5) Bewertung

6) Quellen und Ressourcen

1. Ziele

- HTML 5 Überblick
 - neue Technologien
 - Browser-Kompatibilität
- Vergleich zu HTML 4.01
 - entfernte Elemente, Neuerungen existierender Elemente
- Abgrenzung
 - keine detaillierte Beschreibung von Technologien
 - Bezug zur Webentwicklung
 - Apps und spezielle Webanwendungen außen vor
 - CSS3 wird auf Grund der Komplexität vernachlässigt



1. Entstehung

- 1992 textorientierte Grundversion HTML
- 1999 HTML 4.01
- 2004 Gründung der Arbeitsgruppe WHATWG
 - Initiative der großen Browserhersteller
 - Entwurf "Web Applications 1.0"
- 2006 Gründung der W3C-Arbeitsgruppe
 - Aufgabe: Weiterentwicklung HTML
 - verwendet Fork der WHATWG
- 2007 gemeinsame Arbeit der W3C und WHATWG an HTML 5
- 2009 Scheitern von XHTML 2.0 zu Gunsten von HTML 5

2. Änderungen zu HTML 4.01 - Allgemein

- Standalone ebenso erlaubt wie HTML4 Notation
 - z.B. `
` oder `
` oder `
</br>`
- Attribute nicht mehr zwingend in Anführungszeichen
 - wird dennoch empfohlen
- Vereinfachungen von Doctype
 - `<!doctype html>` statt `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">`
 - HTML5 basiert nicht auf Rahmentechnologie, wie HTML 4 (SGML, XML)
 - ältere Browser wechseln in standardkonformen Modus
 - XHTML 5 benötigt keinen Doctype, da keine XHTML5-DTD vorhanden ist
- vereinfachte Meta-Tags
 - Meta-Tags Attribute: `name`, `http-equiv`, `content`
 - Charset stellt Ausnahme da: `<meta charset="utf-8">`
 - ältere Browser benötigen alte Angaben: `<meta http-equiv="content-type" content="text/html; charset=utf-8">`

2. Änderungen zu HTML 4.01 - Obsolete

- entfallene Elemente / Attribute auf Grund fehlender Semantik
 - oft nur optische, aber keine semantische Auszeichnung
- `<acronym>` „Aussprechbare Abkürzungen“
 - entfallen zu Gunsten von `<abbr>`
 - oft fehlende Kenntniss Unterschied zu `<abbr>`
- Elemente, die lediglich CSS-Formatierung benötigen:
 - `` (Font-family, Font-weight, ...)
 - `<u>` (Text-Decoration)
 - `<frame>` (Position:fixed)
 - weitere: `<basefont>`, `<big>`, `<applet>` (Object, embed), ...
- Attribute
 - Abbr, align, border, bgcolor, cellspacing, cellpadding, ...

2. Änderungen zu HTML 4.01 – Proprietäre

■ Proprietäre Elemente

- Elemente, die nicht zum offiziellen Stand gehören, aber z.B. browser-spezifisch Anwendung fanden
- von Verwendung wird dringend abgeraten

■ Elemente

- `<blink>` (text-decoration: blink)
- `<layer>` (position: absolute / relative)
- `<nobr>` (white-space: nowrap)
- ...

■ Attribute

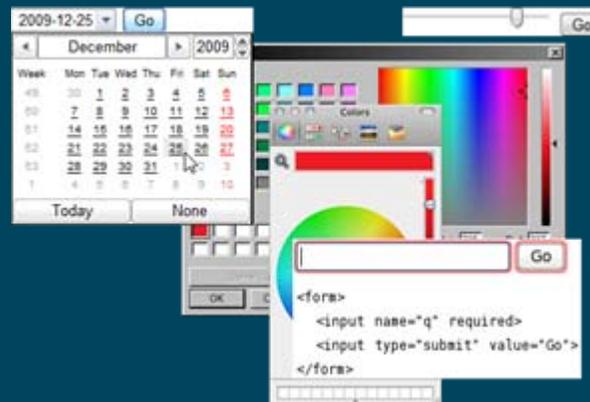
- Top / left (top /left)
- Below (z-index)
- Z-index (z-index)
- ...

3. Strukturelle Elemente

- Elemente, die der semantischen Auszeichnungen & Strukturierung dienen
 - Website für Bots (z.B. Google) leichter erfassbar
 - besserer Einstieg für Anfänger
 - Vereinheitlichung (CSS-Klassen entfallen)
- Problem:
 - ältere Browser erkennen neue Tags nicht an
 - Abhilfe: Javascript-Workaround
- Elemente
 - `<section>` Abschnitte für komplexe Markups „semantische Auszeichnung“
 - `<header>/<footer>` Kopf- und Fußbereich
 - `<nav>` Navigationsbereich
 - ...

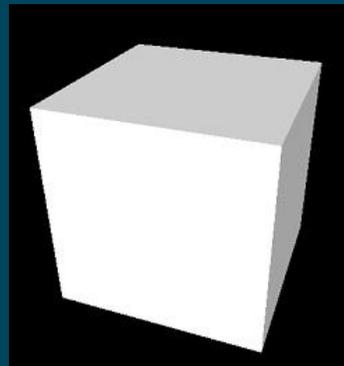
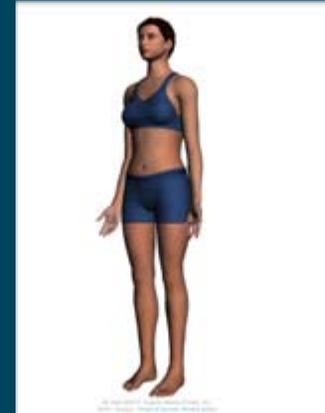
4. Webforms 2.0

- HTML5 besonderes Augenmerk auf Formulare
 - besonders viele Neuerungen
 - derzeit überwiegend nur Opera / Safari (mobile)
- neue Input-Typen:
 - Email, Url (alternative Keyboard bei mobilen OS)
 - Number / Range (min, max, step)
 - Date / Month / Week / Time / Datetime / Datetime-local
 - Search
 - Color
- neue Attribute
 - Required
 - Novalidate
 - Placeholder
 - Autofocus



5. WebGL

- Standard für 3D-Programmierung im Internet
 - Lizenzfrei Standard
- Interface zwischen Javascript und OpenGL ES
 - Open GL hat direkten Zugriff auf Hardware
 - Gerendert wird in Canvas
- Probleme:
 - fehlende Unterstützung der Hardware
 - schlechte Browserunterstützung (Chrome, Firefox4)
 - komplex, da auf OpenGL basiert



BEISPIEL

5. Canvas

- neben SVG eine Möglichkeit Vektor-Grafiken zu erstellen
- Element ist Aktionsbehälter mit Ausmaßen
 - `<canvas id="info" width="400" height="600">...</canvas>`
 - sollte als Inhalt eine Pixelgrafik enthalten = Fallback
- im Script-Bereich eigentliche Vektorerstellung
 - Linien, Rechtecke, Kreisbögen
 - Farbverläufe, Grafiken, Transparenz
 - Stacks (Stapelung) möglich
 - Animation ebenfalls möglich
- derzeit wird überwiegend 2D unterstützt
 - 3D soll in naher Zukunft folgen



5. Audio / Video

- bereits bei HTML 4.01 Multimedia-Inhalte möglich
 - Keine Konventionen über Dateiformate und Codecs
- HTML 5 führt <audio> und <video> ein
 - Nativ (ohne Plugins)
- größtes Problem sind lizenzbedingte Codecs / Formate

Audio

Browser	MP3	OGG	WAV	AU/SND	AIF
 3.5	nein	ja	ja	nein	nein
 4.0	ja	ja	nein	nein	nein
 4.0	ja	nein	ja	ja	ja
 10.0	nein	ja	ja	ja	nein

Video

Formate: MPEG, OGV (OGG)

Codecs:

H.264 (MPEG) = hohe Performance

Theora (OGV) = patentfrei

Browsersupport:

H.264: Microsoft, Apple

Theora: Opera, Mozilla, Google (mittlerweile)

5. Microdata (1)

- eine Form von RDF (System zur Beschreibung von Ressourcen)
- Microdata ermöglicht die Maschine-Maschine-Kommunikation
 - Maschine kann Website interpretieren (Google Crawler)
 - weiterer Schritt in Richtung semantisches Web
- Beispiel: `<dd itemprop="name">Mark Pilgrim</dd>`
- Attribute:
 - Itemprop: Information, dass auf Seite Microdata-Tags eingebunden sind
 - Itemtype: URL, welche das Item definiert und Kontext zu seinen Eigenschaften herstellt.
 - Itemid: Fügt globalen Identifier hinzu, so dass es mit verwandten Items in Verbindung gesetzt werden kann. So kann die Itemid des Items „book“ beispielsweise die „ISBN“ sein.
 - Itemref: Enthält Liste von IDs die mit Item in Verbindung stehen und die zusätzlich zu den Item gecrawlt werden
- Anwendungsbeispiele: vCard, vEvent, ...

5. Microdata (2)

■ Beispiel:

```
<div itemscope itemtype="http://data-vocabulary.org/Person">
  
  <span itemprop="name">Uwe Rosche</span>
  <span itemprop="title">Projektmanager</span>
  <a href="http://www.sinnerschrader.de/"
    itemprop="affiliation">SinnerSchrader</a>
  <span itemprop="address" itemscope
    itemtype="http://data-vocabulary.org/Address" >
    <span itemprop="locality">Hamburg</span>
    <span itemprop="region">Germany</span>
  </span>
</div>
```

Extracted rich snippet data from the page

Item

Type: http://data-vocabulary.org/person
photo = http://www.uerosche.de/wp-content/uploads/uwe-roschel.jpg
name = Uwe Rosche
title = Projektmanager
affiliation = http://www.sinnerschrader.de/
address = *Item(1)*

Item 1

Type: http://data-vocabulary.org/address
locality = Hamburg
region = Germany

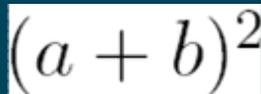
5. SVG & MathML

- SVG (Scalable Vector Graphic)

- wird bis zuletzt nicht von Microsoft unterstützt
 - wollen hauseigenes VML durchsetzen
- XML-basiertes Dateiformat
- Offener, freier Standard
- Einbindung: ``
- direkt: `<svg xmlns="http://www.w3.org/2000/svg">...</svg>`

- MathML (mathematische Formeln)

- gehört nicht direkt zu HTML5 (XML), kann aber eingebunden werden
- Beispiel:


$$(a + b)^2$$

```
<math>
  <msup>
    <mfenced>
      <mi>a</mi><mo>+</mo><mi>b</mi>
    </mfenced>
    <mn>2</mn>
  </msup>
</math>
```

5. Cross-Document-Messaging

- Website-Ursprungsrichtlinien verhindern, Daten von anderen Domänen abzurufen
 - verschiedene Webseiten können nicht miteinander kommunizieren
 - 1. Verwendung von iFrames
 - 2. Ressource und Skriptdaten aus anderen Domänen direkt hosten
 - nur unidirektionale Kommunikation (nur in eine Richtung)
 - Sicherheitsrisiko: Zugriff auf Ressourcen der Host-Website (z.B. Cookies)
- XDM stellt `postMessage`-Methode bereit
 - Kommunikation verschiedener Websites
 - Zustimmungen sind Voraussetzung
 - ist leistungsfähiger und einfacher
 - bidirektional

6. Bewertung

- HTML5 zahlreiche Vorteile:
 - SEO
 - Usability
 - mehr Möglichkeiten
 - freier Standard und Komponenten oft auch frei
 - viele Features können bereits heute verwendet werden
- Nachteile:
 - Spezifikation noch nicht final
 - starke Browserabhängigkeiten
 - benötigt viel Einarbeitungszeit
- Verwendung eher als „Gimmick“ oder bei mobilen Anwendungen
- Entwicklung sollte man im Auge behalten
- unkritische Elemente der Spezifikation sollten verwendet werden

7. Ressourcen und Quellen

■ Quellen:

- <http://Diveintohtml5.org> – Onlinefassung des Buchs „Dive into HTML 5“
- <http://meiert.com/de/publications/talks/20090924/> - Überblick HTML5
- <http://webkompetenz.wikidot.com/docs:html-handbuch> - HTML5 Handbuch
- <http://dev.w3.org/html5/html4-differences/> - W3C HTML 4.01 und 5

■ Ressourcen:

- <http://dev.w3.org/html5/spec/Overview.html> - W3C Spezifikation



MIRTSCHINK & JEGLINSKY GbR
Service Around the Web

Mirtschink & Jeglinsky SAW GbR
Rudi-Lattner-Straße 2
01187 Dresden

Fon +49 (351) 4843969
Fax +49 (351) 4848642

www.mj-sw.de
info@mj-saw.de