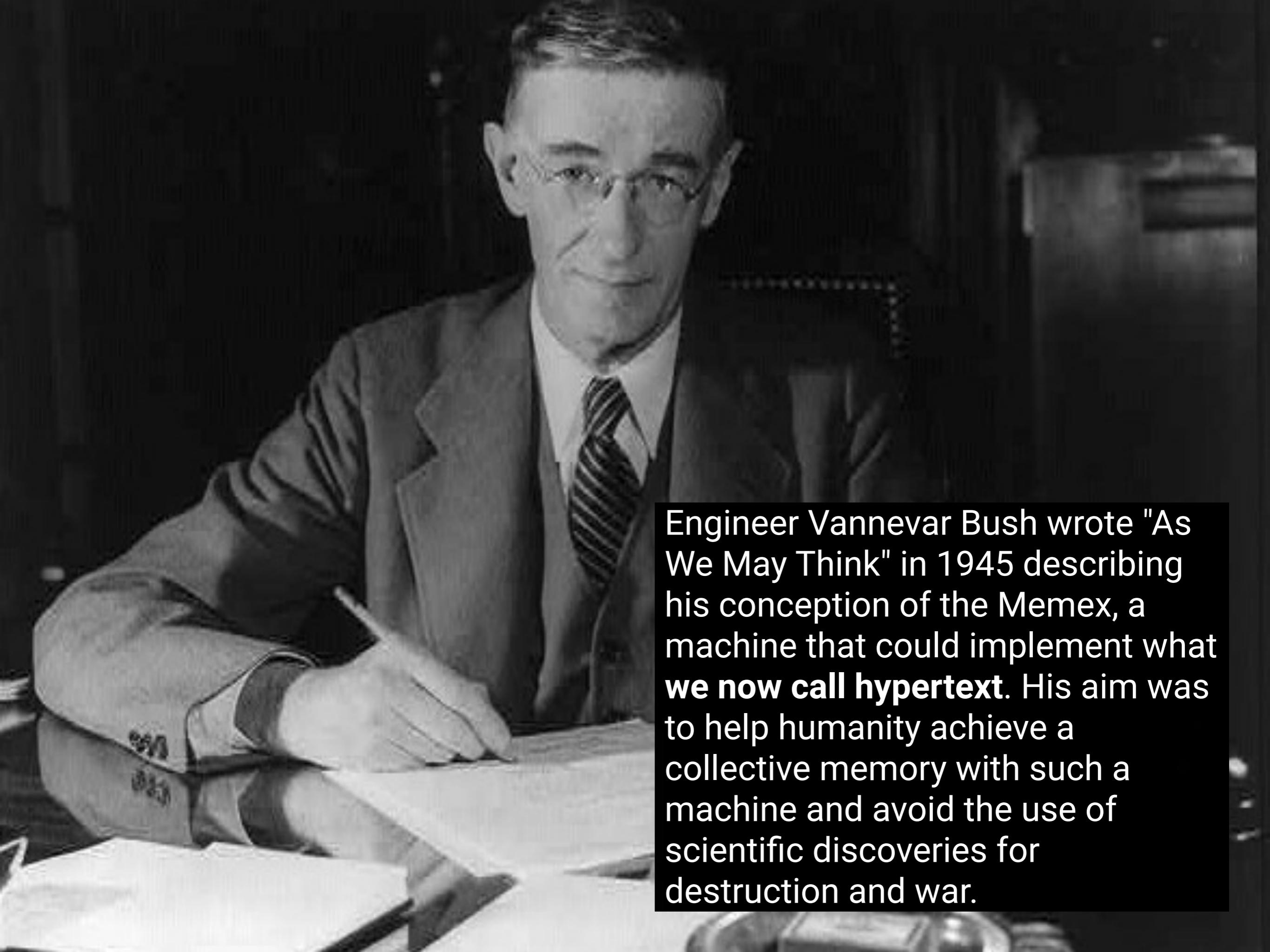


# DIE AUSZEICHNUNGSSPRACHE HTML KOMPAKT

WARUM EIGENTLICH HTML?

DIE ENTSTEHUNGSGESCHICHTE  
VON DER HYPERTEXT MARKUP  
LANGUAGE



Engineer Vannevar Bush wrote "As We May Think" in 1945 describing his conception of the Memex, a machine that could implement what **we now call hypertext**. His aim was to help humanity achieve a collective memory with such a machine and avoid the use of scientific discoveries for destruction and war.

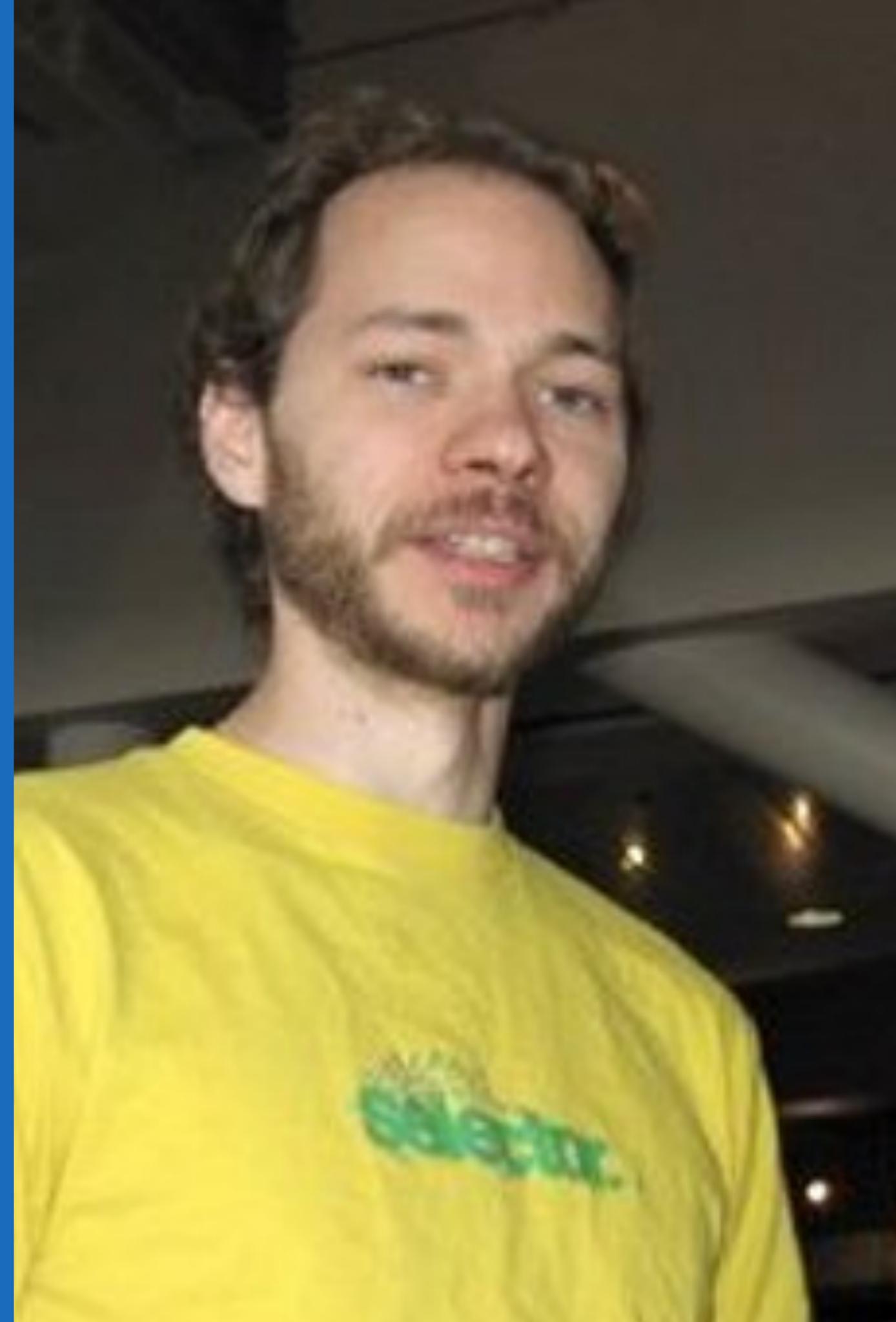
# DER ERFINDER DES WORLD WIDE WEB

- Am 13. März 1992 hat Tim Berners-Lee HTML festgelegt.
- HTML ist eine textbasierte Auszeichnungssprache zur Strukturierung von Inhalten wie Texten, Medien und Hyperlinks in Dokumenten.
- Standardisiert wird HTML vom W3C - dem World Wide Web Consortium.
- <http://www.w3.org/>



# IAN HICKSON

- Ian "Hixie" Hickson war bis 2012 Autor von HTML5 und (u.a.) von Web Application 1.0.
- Hickson ist Coautor von CSS 2.1.
- Hickson arbeitete bei Netscape, Opera und ist heute für Google tätig.



- **1992 HTML**
- 1994 HTML 2  
(Netscape)  
Gründung des W3C  
durch Tim Bernes-Lee  
und andere
- 1995 Mocha, Livescript  
(Netscape, Brendan  
Eich)
- 1995 Microsoft Internet  
Explorer
- 1996 Opera
- Microsoft und Netscape  
führen einen  
Browser "krieg"
- 1996 HTML 3.2, CSS 1,  
JavaScript  
(Tim Berners-Lee rettet  
HTML durch das W3C  
und Spezifikationen)
- **1997 HTML 4,  
HTML 4.01**
- 1998 CSS 2
- 2000 XHTML 1.1
- 2002 Microsoft IE 6:  
Tableless Web Design  
(HTML4 und CSS2)
- **2003 Microsoft  
entwickelt das  
XMLHttpRequest  
(AJAX)**
- 2003 Apple Safari  
Mozilla Firefox
- 2004 Gründung der  
WHATWG  
**Web Applications 1.0**
- 2008 Googles Chrome
- **2010 W3C  
HTML5 Draft**
- 2014 HTML5  
Recommendation
- 2015 Microsoft Edge
- 2016 HTML 5.1  
Recommendation
- 2017 HTML 5.2  
Recommendation
- 2018 HTML 5.3  
Working Draft

# HTML5 PROJEKTLEAD HEUTE

- Steve Faulkner (The Paciello Group)
- Arron Eicholz (Microsoft)
- Travis Leithead (Microsoft)
- Alex Danilo (Google)
- Sangwhan Moon (Invited Expert)

# HTML5 SPEZIFIKATIONEN UND TUTORIALS

- WHATWG -  
Web Hypertext Application Technology Working Group
- **<http://whatwg.org/>**
- W3C - World Wide Web Consortium
- **<http://www.w3.org/TR/>**
- Tutorials:  
<http://w3schools.com>, <http://de.selfhtml.org>

# SEITENSTRUKTUR UND INHALTE DAS HTML DOKUMENT

# STRUKTUR, SYNTAX UND SEMANTIK

- Wie man mit HTML Dokumente baut.

# SEMANTISCHE STRUKTURIERUNG

- HTML dient als Auszeichnungssprache dazu, einen Text semantisch zu strukturieren.
- HTML beschreibt die Struktur des Inhalts - was gehört wozu und besitzt dort welche Bedeutung - Überschrift, Textabsatz, Seitenkopf etc.

# CONTAINER - KNOTEN - ELEMENTE

```
<html> ... </html>
```

# HTML ELEMENT BESITZEN ATTRIBUTE

```
<html  
  xmlns="http://www.w3.org/1999/xhtml"  
  xml:lang="de" lang="de"  
> ... </html>  
  
<a href="link_zur_datei.html">Linktext</a>
```

# GERÜST EINER HTML SEITE

```
<html>
  <head> ...
  <body> ...
</html>
```

# TEXT UND BILD

```
<h1>Titel 1</h1>

<p>
  Ein Textabsatz, der ein
  <em>betontes</em> Wort enthält. Eine Textzeile,<br>
  die hier fortgesetzt wird.
</p>


```

# TEXTAUSZEICHNUNGEN

<h1>Titel 1</h1> ... <h6>Titel 6</h6>

<strong>wichtiger Text</strong>

<em>betonter Text</em>

<u>Unterstrichener Text</u>

<code>Codesnippets</code>

<kbd>Keyboard</kbd>

<abbr title="zum Beispiel">z.B.</abbr>

# HTML STRUKTUR-ELEMENTE

display-type	Bedeutung	Beispiele
<b>text</b>		Unicode Text, HTML Entities
<b>block</b>	Strukturgebende Blockelemente	ol, ul, dl, table, tr, thead, tfoot, tbody, colgroup, col
	Multifunktionale Blockelemente	div, li, dd, td, th, form, noscript
	Begrenzende Blockelemente	h1, p, dt, caption, address, blockquote
<b>inline</b>	inline-semantik	Textauszeichnungen, wie <b>, <strong> ...
	inline-flow (Textfluss)	BR, BDO
	inline-block	zum Beispiel Form Controls

# SEMANTISCHE INLINE ELEMENTE

Elementtyp	Inhaltliche Bedeutung	Elemente
<b>inline-semantic</b>	<i>Hervorhebung der Wichtigkeit</i>	<b>span, em, strong</b>
	Satzauszeichnungen	<b>a, cite, code, kbd, samp, var</b>
	Wortauszeichnungen	<b>abbr, dfn, cite</b>
	Kennzeichnung einzelner Zeichen	<b>sup, sub</b>
<b>inline-flow</b>	<i>Textfluss (Umbruch, Bidirektionalität)</i>	<b>br, bdo</b>
<b>inline-block</b>	<i>Inhalt mit externer Referenz</i>	<b>img, obj, embed, iframe, audio, video, canvas, svg</b>
	Kontrollelemente	<b>input, textarea, select, button, label, video (mit controls)</b>

# HTML STRUKTUR-ELEMENTE

Element	Beinhaltet	Beinhaltet
html		head, body
head		title, meta, link, object, script, style, base
body		noscript, div
article, section, main ...		alle inline   block
nav, header, footer, aside ...		alle inline   block
div, li, dl, address ...		alle inline   block
h1-h6, p, time ...		alle inline

WICHTIGE SEMANTISCHE HTML-AUSZEICHNUNGEN  
UND WANN MAN SIE EINSETZT.

INHALTE -  
TEXTAUSZEICHNUNGEN, LISTEN,  
TABELLEN UND FORMULARE

LISTEN SIND ANSAMMLUNGEN VON UNVOLLSTÄNDIGEN  
SÄTZEN.

LISTEN

# LISTENDARSTELLUNG

```
<ul>
  <li>Zeile 1 einer ungeordneten Liste</li>
  <li>Zeile 2 einer ungeordneten Liste
    <ul>
      <li>Zeile 1 einer ungeordneten Liste</li>
      <li>Zeile 2 einer ungeordneten Liste</li>
    </ul>
  </li>
</ul>
```

```
<ol type="A">
  <li>Zeile 1 einer geordneten Liste</li>
  <li>Zeile 2 einer geordneten Liste</li>
</ol>
```

```
<dl>
  <dt>Listenitem 1 einer Definitionsliste</dt>
  <dd>Beschreibung des Items 2</dd>
  <dt>Listenitem 1</dt>
  <dd>Beschreibung des Items 2</dd>
</dl>
```

# EINE NAVIGATIONSLISTE

```
<nav>
<ul>
  <li><a href="#">Linktext</a></li>
  <li><a href="#">Linktext</a></li>
  <li><a href="#">Linktext</a></li>
</ul>
</nav>
```

Falls sie keine Bulletpoints und Einrückungen an ihren Links möchten, dann entfernen Sie sie mit Hilfe der CSS Anweisungen:

```
ul { margin-top: 0; padding-left: 0; list-style-type: none; }
```

# VERLINKEN VON ANDEREN SEITEN UND MEHR. DAS ANCHOR ELEMENT

# DAS ANCHOR TAG - A

```
<a href="link_zur_datei.html">Linktext</a>

<a href="link_zur_datei.html" target="_blank">
    Link in ein neues Fenster
</a>

<a href="#">Linkdummy ohne Funktion</a>

<a href="mailto:mail@domain.de">
    Link ins Mailprogramm
</a>

<a href="javascript:void();"
    onclick="doFunction();">Linktext</a>
```

# DER BENANNTE ANCHOR

```
<a href="#top">nach oben</a>
<a name="top">Erster Absatz</a>
```

# A RELATIONS

```
<a rel="archives" href="http://myblog.com/archives">old  
posts</a>  
  
<a rel="external" href="http://notmysite.com">tutorial</a>  
  
<a rel="license" href="http://www.apache.org/licenses/  
LICENSE-2.0">license</a>  
  
<a rel="tag" href="http://myblog.com/category/games">games  
posts</a>  
  
<a rel="nofollow" href="http://notmysite.com/sample">wannabe</  
a>  
  
<a rel="tel" href="tel:+49711767676123">  
+49 (0) 711 8 76 76 76-123 </a>
```



Schreibe eine HTML Seite mit folgendem Inhalt. Der Inhalt sollte semantisch richtig aufgebaut sein.

# **Lorem headline 1 ipsum dolor sit**

Maecenas Paragraph sed diam eget risus varius blandit sit amet non magna. Donec id elit non mi porta gravida at eget metus. **Integer bold posuere erat** a ante venenatis dapibus posuere velit aliquet. *Nullam italic id dolor id nibh ultricies vehicula ut id elit.* Donec sed odio dui.

- Morbi unordered list leo risus, porta ac consectetur ac, vestibulum at eros.
- Donec ullamcorper nulla non metus auctor fringilla.
- Maecenas sed diam eget risus varius blandit sit amet non magna.



Füge ein Bild über der Headline ein und einen Link unterhalb des Textabsatzes.

Bilder gibt es zum Beispiel unter  
<https://picsum.photos/200/300>

Die Zahlen geben die Breite und die Höhe des Bildes an.  
Mehr dazu unter <https://picsum.photos/>.

Der Link soll zu Google führen und in einem neuen Fenster geöffnet werden.

5 MINUTEN

TABELLEN SIND EINE SORTIERTE DARSTELLUNG VIELER WERTE  
IN ZEILEN UND SPALTEN.

# TABELLEN

# TABELLEN - GRUNDGERÜST

```
<table>
  <tr>
    <td>Feld 1</td>
    <td>Feld 2</td>
  </tr>
  <tr>
    <td>Feld 1</td>
    <td>Feld 2</td>
  </tr>
  <tr>
    <td>Feld 1</td>
    <td>Feld 2</td>
  </tr>
</table>
```

Tabellencontainer

Tabellenzeile

Tabellenfeld

Tabellenzeile

Tabellenfeld

Tabellenzeile

Tabellenfeld

# TABELLEN- UND SPALTENÜBERSCHRIFTEN - CAPTION, TH

```
<table>
  <caption>Überschrift</caption>  Überschrift
  <tr>
    <th>Feld 1</th>                Tabellenkopfeld
    <th>Feld 2</th>
  </tr>
  <tr>
    <td>Feld 1</td>
    <td>Feld 2</td>
  </tr>
</table>
```

# TABELLENTEILE KORREKT KENNZEICHNEN - THEAD, TFOOT, TBODY

```
<table>
  <thead>
    <tr> ... </tr>
  </thead>
  <tfoot>
    <tr> ... </tr>
  </tfoot>
  <tbody>
    <tr> ... </tr>
    <tr> ... </tr>
  </tbody>
</table>
```

# COL - ANWEISUNGEN FÜR SPALTEN

```
<table>
  <col align="left" />
  <col align="left" />
  <col align="right" />
  <tr>
    <th>ISBN</th>
    <th>Title</th>
    <th>Price</th>
  </tr>
  <tr>
    <td>3476896</td>
    <td>My first HTML</td>
    <td>$53</td>
  </tr>
</table>
```



Füge eine Tabelle mit 3 Zeilen und 3 Spalten am Ende deines Dokuments ein.  
Die erste Zeile soll Spaltenbeschriftungen enthalten.

**Nullam id dolor**

**Id nibh ultricies**

**vehicula ut elit.**

---

Maecenas Sed Diam

Eget Risus Varius  
Blandit

Sit Amet Non Magna

---

Maecenas Faucibus

Mollis Interdum

Ligula Tristique

10 MINUTEN

# FORMULARE

# AUFBAU VON FORMULAREN

```
<form id="form-login" action="form_action.php" method="get">  
    <label for="login-email">Email</label>  
    <input id="login-email" type="email" name="loginEmail" />  
  
    <label for="login-password">Password</label>  
    <input id="login-password" type="password" name="loginPassword" />  
  
    <button type="submit">login</button>  
</form>
```

# FIELDSET, LEGEND BESSERE STRUKTUR MIT GRUPPIERUNGEN

```
<form>
  <fieldset>
    <legend>Login</legend>

    <div>
      <label ...>Email</label>
      <input type="email" ... />
    </div>

    <div>
      <label ...>Password</label>
      <input type="password" ... />
      <p> notes . . .</p>
    </div>

    <button>login</button>

  </fieldset>
</form>
```

The diagram illustrates the structure defined by the code. It shows a main container with a title 'Login'. Inside, there are two separate input groups, each with a label ('Email' and 'Password') and a corresponding input field. A large, rounded rectangular button labeled 'Login' is located at the bottom right.

# DAS INPUT ELEMENT KENNT VIELE TYPEN

`type="checkbox"`  
`type="file"`  
`type="hidden"`  
`type="password"`  
`type="radio"`  
`type="text"`

// deprecated  
`type="button"`  
`type="image"`  
`type="reset"`  
`type="submit"`

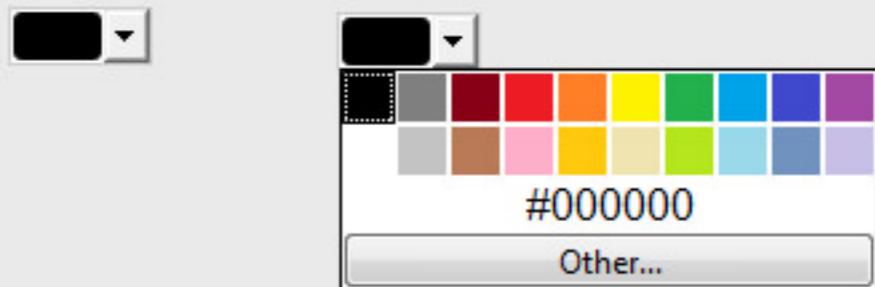
**Checkbox**  
**Dateiupload**  
**Verstecktes Formularfeld**  
**Passwordfeld**  
**Optionsfeld**  
**Einzeiliges Textfeld**

**beliebige Schaltfläche**  
**Eigener Button aus Grafik**  
**Zurücksetzenbutton**  
**Absendenbutton**

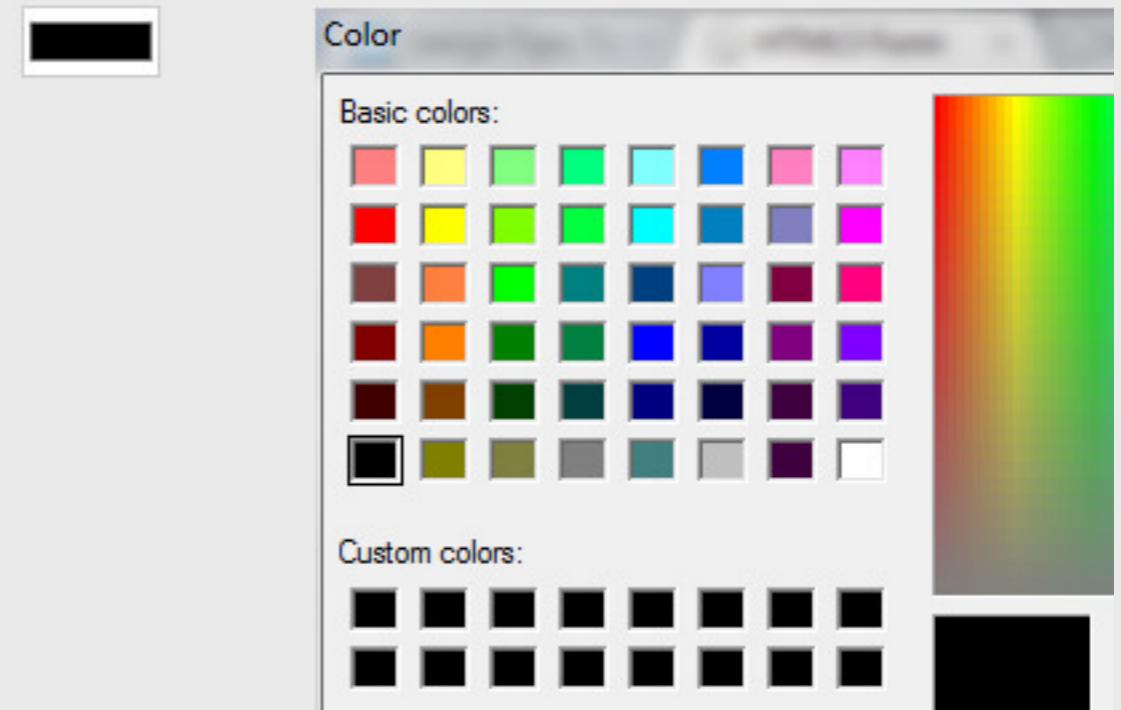
# WEITERE DEFINIERTE INPUT TYPEN AUS HTML5

```
<input type="email">  
<input type="url">  
<input type="text">  
<input type="search">  
  
<input type="date"> und  
Subtypen  
  
<b><input type="color"></b>  
  
<input type="range">  
<input type="tel">  
<input type="number">
```

NATIVE COLOR PICKER IN OPERA



NATIVE COLOR PICKER IN CHROME





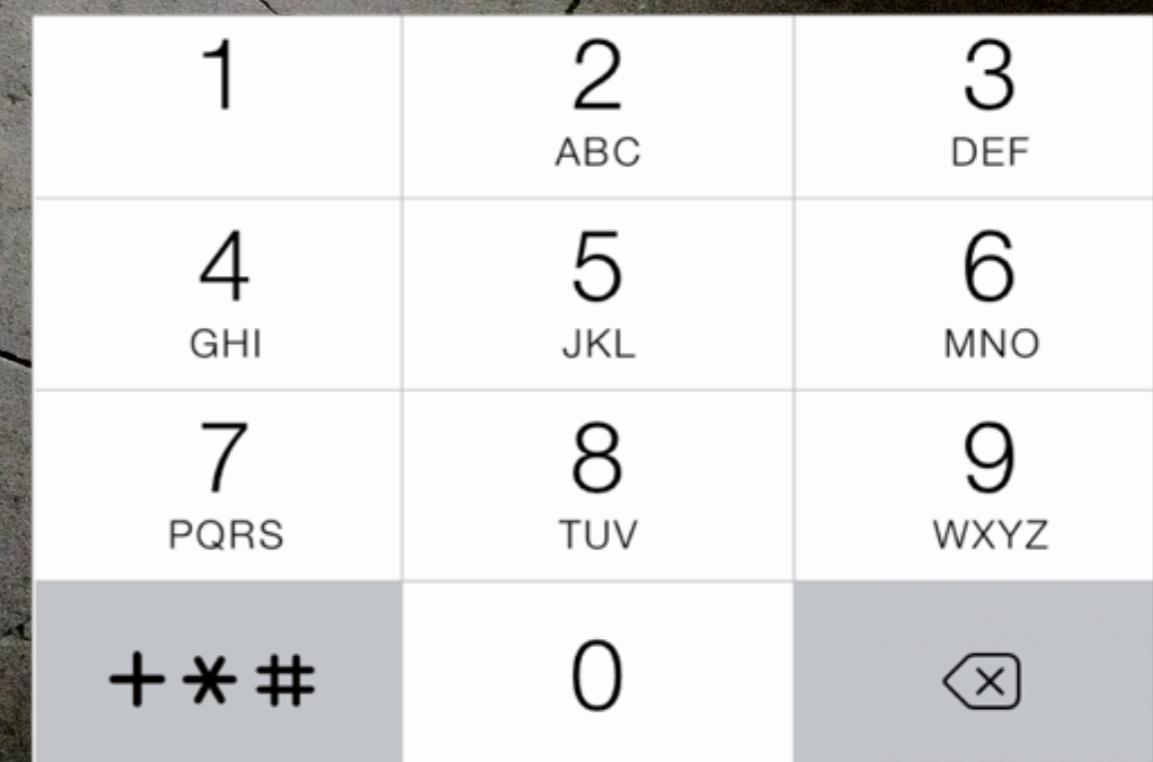
Default



Email

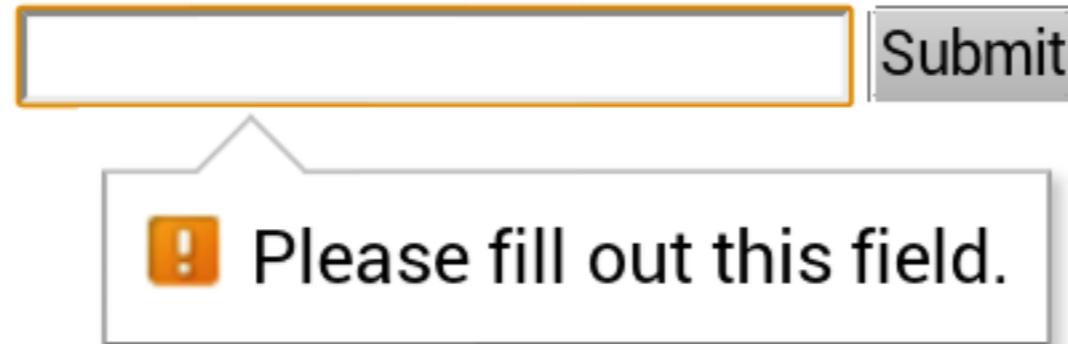


URL



Phone

# VALIDIERUNG UND STATEKLASSEN FÜR EINGABELEMENTE



Definierte Inputtypen besitzen eine automatische Validierung. Es steht ein Javascript Objekt zur Steuerung der Validierung zu Verfügung. Mit einer **Message API** können die Validierungsergebnisse ausgegeben werden.

CSS-Properties zur Kennzeichnung

```
<style>
  :invalid { box-shadow: 0px 0px 5px red; }
  :valid { box-shadow: 0px 0px 5px green; }
  :focus { box-shadow: 0px 0px 5px blue; }
</style>
```

# DATENTYP FÜR DATUMS- UND ZEITANGABEN

```
<input type="day">  
<input type="month">  
<input type="year">  
<input type="datetime">  
<input type="date">  
<input type="week">  
<input type="time">
```

Pick a date

10/dd/2013

October 2013

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

# UND VIELE ATTRIBUTE

`id="any unique value"`

Uniquewert als Ident

`name="a_name"  
value="any value ..."`

Name für den Versand  
Voreingestellter Wert

`placeholder="value ..."  
required="required"`

Platzhaltertext (HTML5)  
Pflichtfeld (HTML5)

`checked="checked"  
selected="selected"  
multiple="multiple"`

Ausgewählte Checkbox  
Ausgewählter Listeneintrag  
Mehrfachauswahl bei Listen

`accesskey="e"  
tabindex="1"`

Zugriff per Taste  
Tabulatorreihenfolge

# ATTRIBUTE, DIE SIE BEI BEDARF EINSETZEN SOLLTEN

disabled="disabled"  
dir="rtl | ltr"  
lang="language"

Element deaktivieren  
Schreibrichtung in einem Feld  
Sprachspezifikation für ein Feld

autofocus  
autocomplete= "on | off"

Automatische Cursorplatzierung  
An-/Abschalten des Wertespeichers

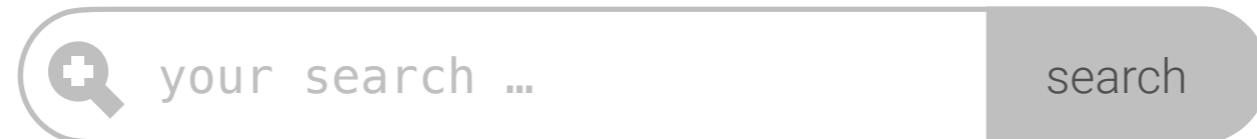
# EIN MEHRZEILIGES EINGABEELEMENT

```
<textarea>  
  Einiger u.U. mehrzeiliger Text im Textfeld.  
</textarea>
```

# AUSWAHLLISTEN UND MENÜS

```
<select name="auto" multiple size="3">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="mercedes">Mercedes</option>
  <option value="audi">Audi</option>
</select>
```

# EIN EINGABEFELD FÜR SUCHANFRAGEN GESTALTEN



```
<form action="http://www.google.com/search">
  <label>Google:</label>
  <input type="search" name="q" placeholder="your search...">
  <button type="submit">search</button>
</form>
```

```
input[type=search] {
  border-left-radius : 0.75rem;
  background         : url(src/ico/lupe.png);
  ...
}

button {
  border-right-radius : 0.75rem;
  ...
}
```

# RANGE



```
<input type="range" list="scala"
      min="0" max="100" step="1" value="17"
      onchange="showValue(this.value)">
<output>17</output>

<datalist id="scala"> ... </datalist>

<script>
function showValue(val) {
  document.querySelector('output').innerHTML = val;
}
</script>
```

# DROPODOWN LISTE FÜR EINE KOMBOBOX

```
<input type="email" id="contacts" list="contactList">  
  
<datalist id="contactList">  
  <option value="michael@zenbox.de" label="Michael">  
  <option value="peter@zenbox.de" label="Peter">  
</datalist>
```

# <PROGRESS>, <METER>



```
<progress min="0" max="100" value="60"></progress>
```

```
<meter min="0" max="5" value="3" step="1"></meter>
```



# <OUTPUT>

```
<output id="calculated-value"> ... </output>
```

Markiert Inhalte, die dynamisch berechnet werden, beispielsweise die Summe eines Warenkorbes, oder die Ausgabe eines Slider-Elementes.



Baue ein Loginformular. Es soll eine Emailadresse, ein Password und eine Checkbox haben, die danach fragt, ob das Passwort gespeichert werden soll. Über einen Button "Login" soll das Formular an login.php abgeschickt werden.

10 MINUTEN

# FÜR ALTE HASEN BEKANNTES HTML ELEMENTE MIT NEUER BEDEUTUNG

# BEKANNTES HTML ELEMENT MIT NEUER BEDEUTUNG

<em>, <b>, <u> – nicht-semantische grafische Auszeichnungen

- <em> – semantische Betonung
- <strong> – semantische Hervorhebung (als wichtig)
- <hr> – Sinnabschnitt in Texten (hr { border : 0; })
- <small> – semantisch Kleingedrucktes

# TAGS, DIE NUR DER FORMATIERUNG DIENEN, FALLEN WEG.

~~<applet>~~ ~~<acronym>~~ ~~<bgsound>~~ ~~<basefont>~~ ~~<big>~~ ~~<blink>~~ ~~<center>~~  
~~<dir>~~ ~~<font>~~ ~~<frame>~~ ~~<frameset>~~ ~~<isindex>~~ ~~<listing>~~ ~~<noframes>~~  
~~<nobr>~~ ~~<s>~~ ~~<spacer>~~ ~~<strike>~~ ~~<tt>~~

Anstelle von ~~<frameset>~~, ~~<frame>~~ wird CSS oder ~~<iframe>~~ verwendet.

# ATTRIBUITE, DIE NUR DER FORMATIERUNG DIENEN, FALLEN WEG.

align, color, size, width, height, hspace, vspace und margin  
in Verbindung mit zum Beispiel <body>, <table>, <iframe>.

Aber nicht unbedingt width und height des img-Tags. Denn hier  
sind das Eigenschaften des Bildes.

```
  
<canvas [width="100"] [height="100"]></canvas>
```

# NEUE SEMANTISCHE HTML ELEMENTE AUS HTML5

# <MARK>

Das mark-Element stellt hervorgehobenen Text dar. Dabei geht es hier etwa um Text, der von einer Suchmaschine gefunden wurde und sichtbar markiert wird, sodass er schnell von Benutzern gefunden werden kann.

```
<mark> Fundstelle </mark>
```

```
mark {  
    background-color : #ffff00;  
    color : black;  
}
```

# <RUBY>, <RP>, <RT>

```
<ruby>
  汉 <rp>(</rp><rt> han </rt><rp>)</rp>
  字 <rp>(</rp><rt> zi </rt><rp>)</rp>
</ruby>
<ruby>
  <rp> Ruby Notiz für Nicht–Rubybrowser.
  <rt> Erklärung für eine Ruby Notiz.
</ruby>
```

Beispiel in Ruby–Browsern



Abbildung in Nicht–Ruby–Browsern  
汉(hàn) 字(zì)

# <FIGURE> UND <FIGCAPTION>

Definiert eine Mediencontent Gruppe und deren Beschriftung.

Das **figure**-Element erlaubt es, ein bestimmtes Medium (z. B. Bilder, Videos etc.) mit einer Bildunterschrift zu versehen.

```
<figure>
  <video src="video.webm"></video>
  
  
  
  <figcaption>Bildunterschrift</figcaption>
</figure>
```

# <ADDRESS>

Kontaktinformationen einem aktuellen Artikel. Die Kontaktinformation bezieht sich auf den Besitzer oder Autor des Dokumentes.

```
<address>
  <p>Magda Mustermann</p>
  <p>Bei Spielweg 12</p>
  <p>54321 Ort im Satz</p>
  <p><a href="mailto:mail@beispiel.de">
    mail@beispiel.de</a></p>
</address>
```

# <TIME>

Das `time`-Element wird verwendet, um Zeitangaben auszuzeichnen und Zeitangaben mit Metadaten auszustatten, sodass ein Browser damit z. B. einen interaktiven Kalender erzeugen kann.

Um `<time>10:00 h</time>` ist geöffnet.

`<time datetime="2012-11-19">Today</time>`

# LINK RELATIONS

Das rel - Attribut weist die Beziehung zu einer Dateiverlinkung aus. Dies kann eine semantische oder auch eine funktionale Beziehung sein. Oft muss auch der MIME-Type angegeben werden.

```
<link rel="alternate" type="application/rss+xml" href="http://  
myblog.com/feed" />  
<link rel="icon" href="/favicon.ico" />  
<link rel="pingback" href="http://myblog.com/xmlrpc.php">  
<link rel="stylesheet" type="text/css" href="styles.css">
```

# DAS TEMPLATE ELEMENT

Es stellt ein Vorlagenelement zur Wiederverwendung innerhalb des HTML DOMs zur Verfügung. Das template Element ist im User Agent Stylesheet mit `display:none` versehen. Es muss vor seiner Verwendung aktiviert werden. Dies geschieht zweckmäßigerweise mit einer tiefen Kopie via `cloneNode()`.

```
<table id="master">
<tr>
  <template id="cells-to-repeat">
    <td></td>
  </template>
</tr>
</table>
```

```
var t = document.querySelector('#cells-to-repeat');

t.content.querySelector('img').src = 'logo.png';

document.querySelector('#master tr').appendChild(t.content.cloneNode(true));
```

<http://microformats.org/wiki/existing-rel-values>

# SEMANTISCHE HTML5 ELEMENTE ZUR STRUKTURIERUNG DES DOKUMENTS UND VON CONTENT

# <HEADER>, <FOOTER>

`<header>, <footer>`

Kopfteil einer Sektion oder einer Seite  
oder einer anderen Gruppe.

Das header-Element definiert den Kopfbereich für den aktuellen Abschnitt (z. B. in einem section-Element). Neben einer Überschrift kann es auch weitere Informationen (z. B. Versionsdaten) enthalten.

Das footer-Element definiert eine Fußzeile für den aktuellen Abschnitt (z. B. in einem article-Element) und gibt z. B. Auskunft über den Autor oder die Erstellungszeit des Textes.

# <NAV>

```
<nav>
  <header>
    <h4>Überschrift</h4>
    <time>2013</time>
  </header>
  <ul>
    <li><a href="">Linktext</a></li>
    <li><a href="">Linktext</a></li>
  </ul>
</nav>
```

Ein Navigationsbereich.

Dieses Element umschließt einen Navigationsbereich. Damit sind nicht nur Verweise zu anderen Seiten eines Webauftritts gemeint, sondern auch Verweise, die auf Abschnitte innerhalb eines Dokuments zeigen.

# <ARTICLE>

`<article> ... </article>`

Kennzeichnet einen inhaltlichen Artikel.

Das article-Element umfasst einen unabhängigen Abschnitt, der jedoch im Kontext zu den Vorfahren-Elementen steht. Damit sind unter anderem Nachrichtenartikel, Blog- oder Foreneinträge gemeint.

# <ASIDE>

`<aside> ... </aside>`

Nebenbeiinhalt, Marginalspalteninhalt.

Bezeichnet Inhalt, der thematisch zum umschließenden Inhalt passt, aber nicht direkt dazugehört. Ein Nachrichtentext könnte in einem aside-Element Querverweise zu ähnlichen Nachrichten beinhalten.

# <MAIN>

<main> ... </main>

Der wesentliche Inhalt einer Seite. Muss gelesen werden, wenn der Inhalt der Seite erfasst werden soll. <main> soll nur einmal pro Seite, ggf. pro <section> vorkommen.

# EINE ARTIKELGRUPPE MIT VERSCHIEDENEN INHALTEN

```
<article>
  <header>
    <h4>Überschrift der Nav</h4>
    <address>Autorenangaben</address>
    <time>November 2012</time>
  </header>
  <p>Das ist der Text für den Artikel.</p>
  <figure>
    
    <figcaption>Bildunterschrift</figcaption>
  </figure>
  <aside>
    <h2>Weitere Links</h2>
    <nav>
      <ul>
        <li><a href="">Linktext</a></li>
        <li><a href="">Linktext</a></li>
      </ul>
    </nav>
  </aside>
</article>
```

# <SECTION>

```
<section id="news"> ... </section>
```

Definiert eine inhaltliche Sektion.

Ein section-Element gruppiert thematisch zusammenhängenden Inhalt und besitzt meist einen Kopf- und Fußbereich. Eine typische Webseite wird section-Elemente ein Produkt-, ein Nachrichten- und ein Kontaktbereich verwenden.



Baue eine HTML Struktur für dieses Layout und füge die bisher geschriebenen Inhalte in die *Dokumentensektionen* ein.

10 MINUTEN

EINBINDEN MASCHINENLESBARER INFORMATIONEN IN HTML-DOKUMENTE.

# MICRODATEN

# MICRODATEN

- Einbinden maschinenlesbarer Informationen in HTML-Dokumente.
- Es wird eine Kompatibilität zu anderen Formaten wie dem Resource Description Framework (RDF) und der JavaScript Object Notation (JSON) angestrebt.
- Der Standard "Microdata" ist nicht mehr in der Entwicklung, sonder hat sich für RDF entschieden.

# RICH SNIPPETS MIT MICRODATEN

```
<div itemscope itemtype="http://schema.org/Event">
  <a itemprop="url" href="nba-miami-philadelphia-game3.html">
    NBA Eastern Conference First Round Playoff Tickets:
    <span itemprop="name"> Miami Heat at Philadelphia 76ers – Game 3
    (Home Game 1) </span>
  </a>
  <meta itemprop="startDate" content="2016-04-21T20:00">
    Thu, 04/21/16
    8:00 p.m.
  <div itemprop="location" itemscope itemtype="http://schema.org/Place">
    <a itemprop="url" href="wells-fargo-center.html">
      Wells Fargo Center
    </a>
    <div itemprop="address" itemscope itemtype="http://schema.org/PostalAddress">
      <span itemprop="addressLocality">Philadelphia</span>,
      <span itemprop="addressRegion">PA</span>
    </div>
  </div>
</div>
```

Rich Snippets Testing Tool at  
<https://search.google.com/structured-data/testing-tool?url=>

Rich Snippets

Help with:

[Documentation](#)

[Tips & Tricks](#)

## Rich Snippets Testing Tool Beta

Rich Snippets allows you to enhance your Google search results by marking up web pages with Microformats, RDFa or Microdata.

[Test your website](#)

Enter a web page URL to see how it may appear in search results:

[Preview](#)

Examples: [Urbanspoon](#), [LinkedIn](#)

[Google search preview](#)

[Pizza My Heart - Santa Cruz | Urbanspoon](#)

★★★★★ 10 reviews - Price range: Under \$10 per entree

Excerpt from the page will show up here. Excerpt from the page will show up here.

Excerpt from the page will show up here. Excerpt from the page will show up here.

[www.urbanspoon.com/r/6/765421/restaurant/Pizza-My-Heart-Santa-Cruz](#) - [Cached](#) - [Similar pages](#)

Note that there is no guarantee that a Rich Snippet will be shown for this page on actual search results. For more details, see the [FAQ](#).

[Extracted Rich Snippet data from the page](#)

hreview-aggregate

item hcard

fn = Pizza My Heart

MIT DEM "DATA DASH" BASIS ATTRIBUT KÖNNEN SIE MACHEN,  
WAS SIE MÖCHTEN.

"DATA-" ATTRIBU<sup>E</sup>T

# DATA-MEIN-EIGENES-ATTRIBUT

Oft ist es notwendig, in einem HTML Element Attribute zweckentfremdet zu verwenden. So wird das title-Attribut oft als eine zusätzliche Informationsebene in Tags oder als CSS-Hook angewandt.

Mit dem Prefix "data-" können eigene Attribute an beliebiger Stelle hinzugefügt werden:

```
<a href="" title="" data-tooltip="Mein Tooltiptext">...</a>
<div id="header" data-section="page-header">...</div>
```

Diese lassen sich mit Selektoren ansprechen:

```
[data-section=page-header] { display : block; }
```

```
document.querySelectorAll("[data-
tooltip]").addEventListener( ... );
```

```
domElement.dataset('tooltip') ... -> data-tooltip=""
```

BAU EINES LAYOUTS MIT HTML UND CSS.

# TEMPLATES - BAUANLEITUNG FÜR EIN SEITENLAYOUT

# EINRICHTEN DER HEAD SEKTION

# DOCTYPE, NAMESPACE UND SPRACHANGABE

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      lang="de-DE">
  <head>
    <meta charset="utf-8">
```

Das lang-Attribut stellt die Sprache für Inhalte und Attribute-Werte des Dokumentes ein. Die Sprache kann für einzelne Elemente überschrieben werden.

<p>erster Absatz</p>	→ deutsch
<p lang="fr">Deuxième</p>	→ französisch
<p>dritter Absatz</p>	→ wieder deutsch

# VERALTETE DOCTYPES

//deprecated

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

v

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

v

<!DOCTYPE HTML PUBLIC  
"-//W3C//DTD XHTML 1.1//EN"  
"http://www.w3.org/TR/xhtml11/DTD/**xhtml11.dtd**"

v

# DIE ZEICHENSATZANGABE

```
<!DOCTYPE html>
<html lang="de-DE">
  <head>
    <meta charset="utf-8">
```

UTF – UCS Transformation Format  
UCS – Universal Character Set

# <TITLE> - DER DOKUMENTENTITEL

```
<!DOCTYPE html>
<html lang="de-DE">
  <head>
    <meta charset="utf-8">
    <title>Titel des Dokumentes – Herkunft</title>
```

Kürzeste Identifikation eines Dokuments für Suchmaschinenlistings, Bookmarks, Fenstertitel; SEO-relevant!

# DEN VIEWPORT FÜR MOBILE GERÄTE EINRICHTEN

```
<head>
  <meta charset="utf-8">
  <title>Titel des Dokumentes</title>
  <meta name="viewport"
        content="width=device-width,
                  initial-scale=1.0">
```

Die obige Zeile stellt die Darstellung auf Bildschirmbreite ein, dabei wird die Darstellungsgröße um einen Gerätefaktor vergrößert (iPhone = 2,0), der für eine bessere Darstellungsgröße und Lesbarkeit sorgt. Weitere Werte: min-scale, max-scale, user-scalability.

Dies ist derzeit ausschließlich nur für mobile Geräten wirksam und wird von Desktopbrowsern ignoriert.

# ICONS FÜR DEN BROWSER

```
<link rel="shortcut icon" href="src/ico/favicon.png" />
```

# METAELEMENTE ALS HTTP-KOPFDATEN-ÄQUIVALENT

```
<meta http-equiv="Content-Type"  
      content="text/html; charset=utf-8" />
```

Sie enthalten Informationen, die in Kopfzeilen der HTTP-Kommunikation vom Server zum Client übertragen werden. Die Daten können bereits vor dem Erzeugen der HTTP-Antwort vom Webserver gelesen und in die Kopfzeilen eingebaut werden. Eine Information im HTTP-Header wird einem Meta-Element vorgezogen. Angabe des MIME-Type (im Beispiel text/html) und der verwendeten Zeichenkodierung (im Beispiel UTF-8) des Dokuments.

# METAELEMENTE ALS HTTP-KOPFDATEN-ÄQUIVALENT

```
<meta http-equiv="refresh"  
      content="5;  
      URL=http://www.example.org/" />
```

## Weiterleitung durch „refresh“

Leitet beim Aufruf einer Seite zu einer anderen URL um. Das content-Attribut enthält eine Zeitangabe für die Weiterleitung, z. B. um den Anwender noch auf den Grund für die Weiterleitung hinzuweisen.

# METAELEMENTE FÜR WEBCRAWLER I

<meta name="robots" content="index, follow" />  
oder <meta name="robots" content="all" />

Der Webcrawler (Suchroboter) soll die Webseite ausdrücklich in den Suchmaschinen-Index aufnehmen, damit sie auffindbar ist, und den Hyperlinks auf der Seite folgen. Diese Angabe ist im Allgemeinen unnötig, da sie dem normalen Verhalten der meisten Webcrawler entspricht.

# METAELEMENTE FÜR WEBCRAWLER I

`<meta name="robots" content="noindex" />`

Der Suchroboter soll die Webseite nicht in den Index aufnehmen. Die Webseite soll also nicht über Suchmaschinen auffindbar sein. Siehe auch: Robots Exclusion Standard.

`<meta name="robots" content="nofollow" />`

Der Suchroboter darf die Webseite zwar aufnehmen, aber soll den Hyperlinks auf der Seite nicht folgen. Die Wirkung ist so, als wären die Links auf der betreffenden Seite gar nicht vorhanden. Derselbe Effekt lässt sich auch für einzelne Links erreichen, indem diese um das Attribut `rel="nofollow"` ergänzt werden. (siehe `nofollow`)

# METAELEMENTE FÜR SUCHMASCHINEN (ODER AUCH NICHT MEHR)

Kurze **Beschreibung** des Inhalts der Webseite

```
<meta name="description"  
      content="Lorem ipsum dolor sit amet,  
              consetetur sadipscing elitr." />
```

## **Stichwörter**

Nennt Stichwörter oder Themen, die auf der Webseite behandelt werden. Komma-getrennte Liste.

```
<meta name="keywords"  
      content="Stichwort, zwei Stichworte, Stichwort3" />
```

**Name des Autors** und inhaltlich Verantwortlichen.

```
<meta name="author"  
      content="Autorennname" />
```

# Verlinken externer Stylesheets

```
<link href="src/css/default.css"  
      rel="stylesheet" type="text/css" />  
</head>
```

Externe Stylesheets bieten der Vorteil der Wiederverwendbarkeit und Modularität.  
Sie werden im Kopfteil der HTML eingebunden, damit sie möglichst frühzeitig zur Verfügung stehen.

# DIE BODY SEKTION

Die Body Sektion ist der  
sichtbare Teil des Dokuments

Layout (nicht semantisch)

+

User Interface (semantisch)

+

Content (semantisch)

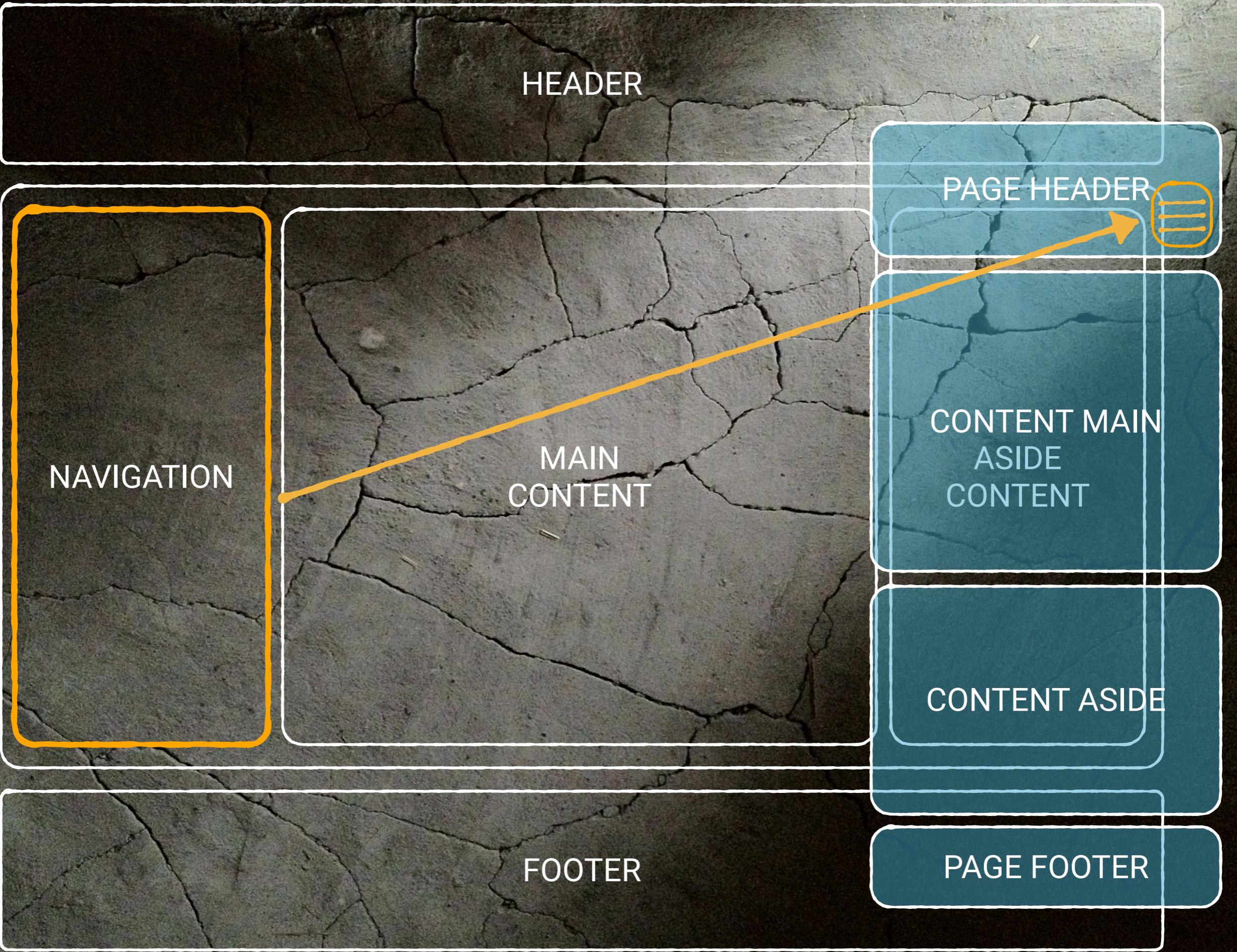
HEADER

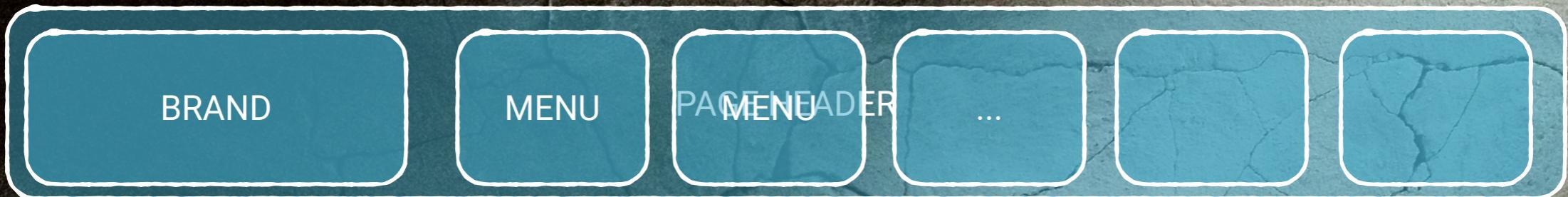
NAVIGATION

MAIN  
CONTENT

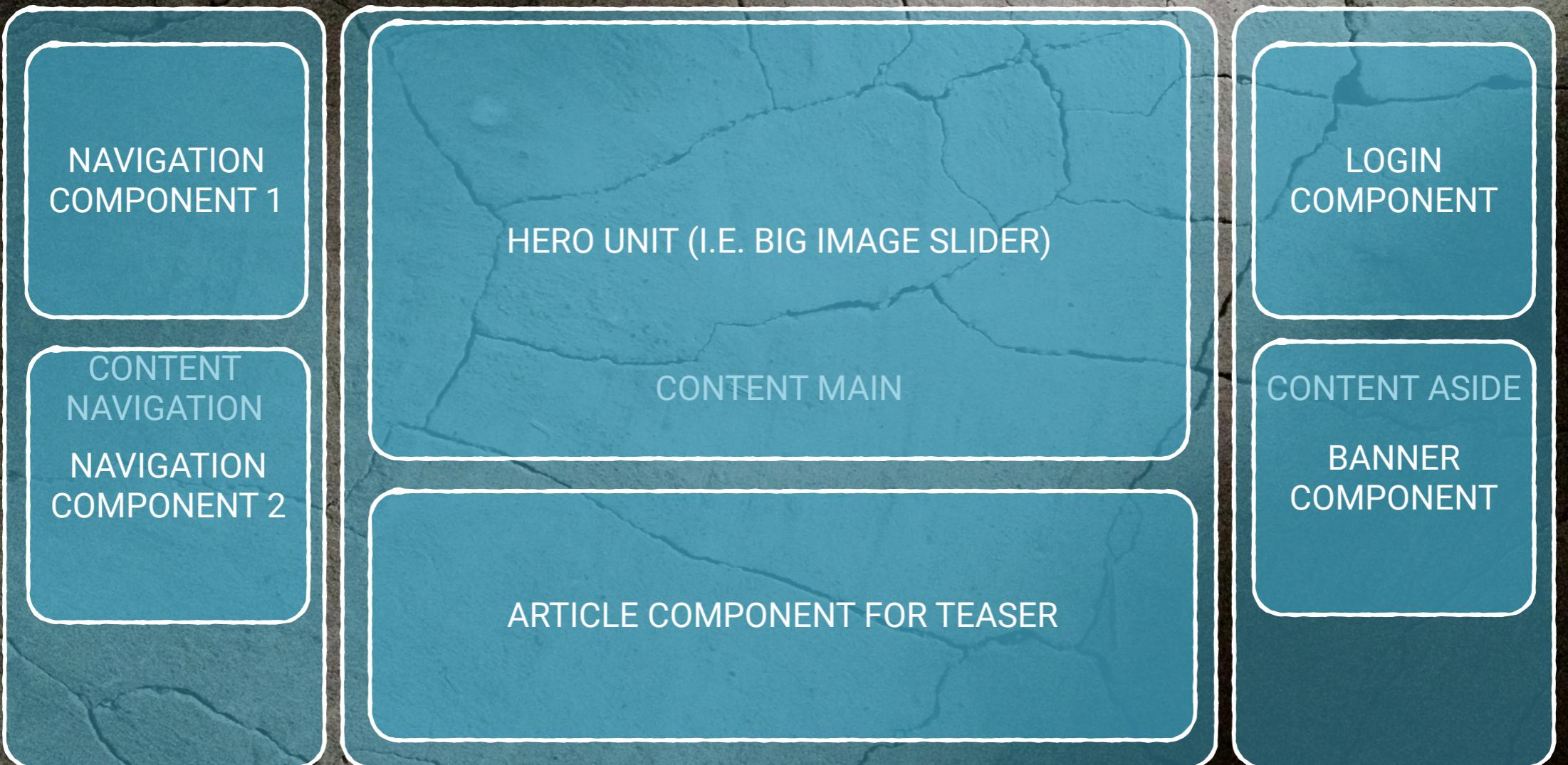
ASIDE  
CONTENT

FOOTER





PAGE HEADER



# AUFBAU VON LAYOUTELEMENTEN

```
<body>  
    <div class="row"> ... </div>  
  
    <div class="row">  
        <div class="two columns">...</div>  
        <div class="three columns">...</div>  
    </div>  
  
    <div class="row">...</div>  
</body>  
</html>
```

# SEMANTISCHE AUSZEICHNUNG VON LAYOUT ELEMENTEN

```
<body>

    <div class="row"><header> ... </header></div>

    <div class="row">
        <div class="two columns"><nav> ... </nav></div>
        <div class="three columns"><main> ... </main></div>
    </div>

    <div class="row"><footer> ... </footer></div>
</body>
</html>
```

# SEMANTISCHE AUSZEICHNUNG VON LAYOUT ELEMENTEN

```
<html>
<head>
    <title> ... </title>
</head>
<body>

    <header> ... </header>
    <nav> ... </nav>
    <main> ... </main>
    <aside> ... </aside>
    <footer> ... </footer>

</body>
</html>
```

# SEMANTISCHE AUSZEICHNUNG VON USER INTERFACE ELEMENTEN

```
<nav>
  <ul>
    <li> ... </li>
    <li> ... </li>
    <li> ... </li>
  </ul>
</nav>

<form>
  <fieldset>
    <legend> ... </fieldset>
    <label> ... </label>
    <input />
    <button></button>
  </fieldset>
</form>
```

# SEMANTISCHE AUSZEICHNUNG VON CONTENT ELEMENTEN

```
<main>
  <h1> ... </h1>
  <article>
    <h2> ... </h2>
    <p> ... </p>
    <p> ... </p>
    <table> ... </table>
  </article>
</main>
```

# ID'S EINSETZEN

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

Ein Id-Attribut gibt einem HTML Element eine Identität. Ein Id-Wert muss innerhalb einer Seite unique bleiben.

Id's funktionieren als 'Hooks' in das Dokument und werden von Selektoren in Javascript und gelegentlich auch von CSS verwendet.

```
Javascript: document.getElementById('wrapper');  
CSS: div#wrapper { some styles ... }
```

# KLASSENATTRIBUTE VERWENDEN

```
<div class="page-header"
```

Ein Klassen-Attribut verbindet ein HTML Element mit einer Style-Anweisung in einem Stylesheet. In Gegensatz zur ID ist eine Klasse mehrfach verwendbar.

Klassen werden in der Stylesheet-Datei mit Klassenselektoren geschrieben.

```
Javascript: document.getElementsByClassName('page-header');  
CSS: .page-header { some styles ... }
```

# EIN TEMPLATE MIT NICHT-SEMATISCHEN POSITIONSRAHMEN

```
<div class="page">  
  <div class="page-header"> ... </div>  
  
  <div class="page-content">  
    <div class="content-nav"> ... </div>  
    <div class="content-main"> ... </div>  
    <div class="content-aside"> ... </div>  
  </div>  
  
  <div class="page-footer"> ... </div>  
</div>
```

# AUFFÜLLEN MIT SEMANTISCHER DOKUMENTENSTRUKTUR

```
<div class="page">  
  
  <div class="page-header">  
    <header> ... </header>  
  </div>  
  
  <div class="page-content">  
    <div class="content-nav">  
      <nav> ... </nav>  
    </div>  
    <div class="content-main">  
      <main> ... </main>  
    </div>  
    <div class="content-aside">  
      <aside> ... </aside>  
    </div>  
  </div>  
  
  <div class="page-footer">  
    <footer> ... </footer>  
  </div>  
  
</div>
```

# EINE NAVIGATIONS KOMPONENTE

```
<!-- navigation component -->
<nav>
  <ul>
    <li><a href="page-1.html">Page 1</a></li>
    <li><a href="page-2.html">Page 2</a></li>
    <li><a href="page-3.html">Page 3</a></li>
  </ul>
</nav>
<!-- /navigation component -->
```

Komponentencodes sind wieder verwendbar. Sie bilden die Grundlage für User Interface Libraries.

```
<div class="content-nav"> navigation component</div>
<div class="page-footer"> navigation component</div>
```

# EINE LOGINFORM

```
<!-- login form component -->
<form>
  <fieldset>
    <legend>login</legend>

    <fieldset>
      <label><a href="page-1.html">Page 1</a></li>
      <input><a href="page-2.html">Page 2</a></li>
    </fieldset>

    ...
    <button type="submit">log in</form>
<!-- /login form component -->

<div class="page-aside">login form component</div>
```

# EINE LOGINFORM MIT CSS

```
<!-- login form component -->
<form class="form form-vertical">
  <fieldset class="fieldset">
    <legend>login</legend>

    <fieldset class="input-group input-group-vertical">
      <label><a href="page-1.html">Page 1</a></li>
      <input><a href="page-2.html">Page 2</a></li>
    </fieldset>

    ...
    <button type="submit"><a href="page-3.html">Page 3</a></li>
  </fieldset>
</form>
<!-- /login form component -->

<div class="page-aside">login form component</div>
```