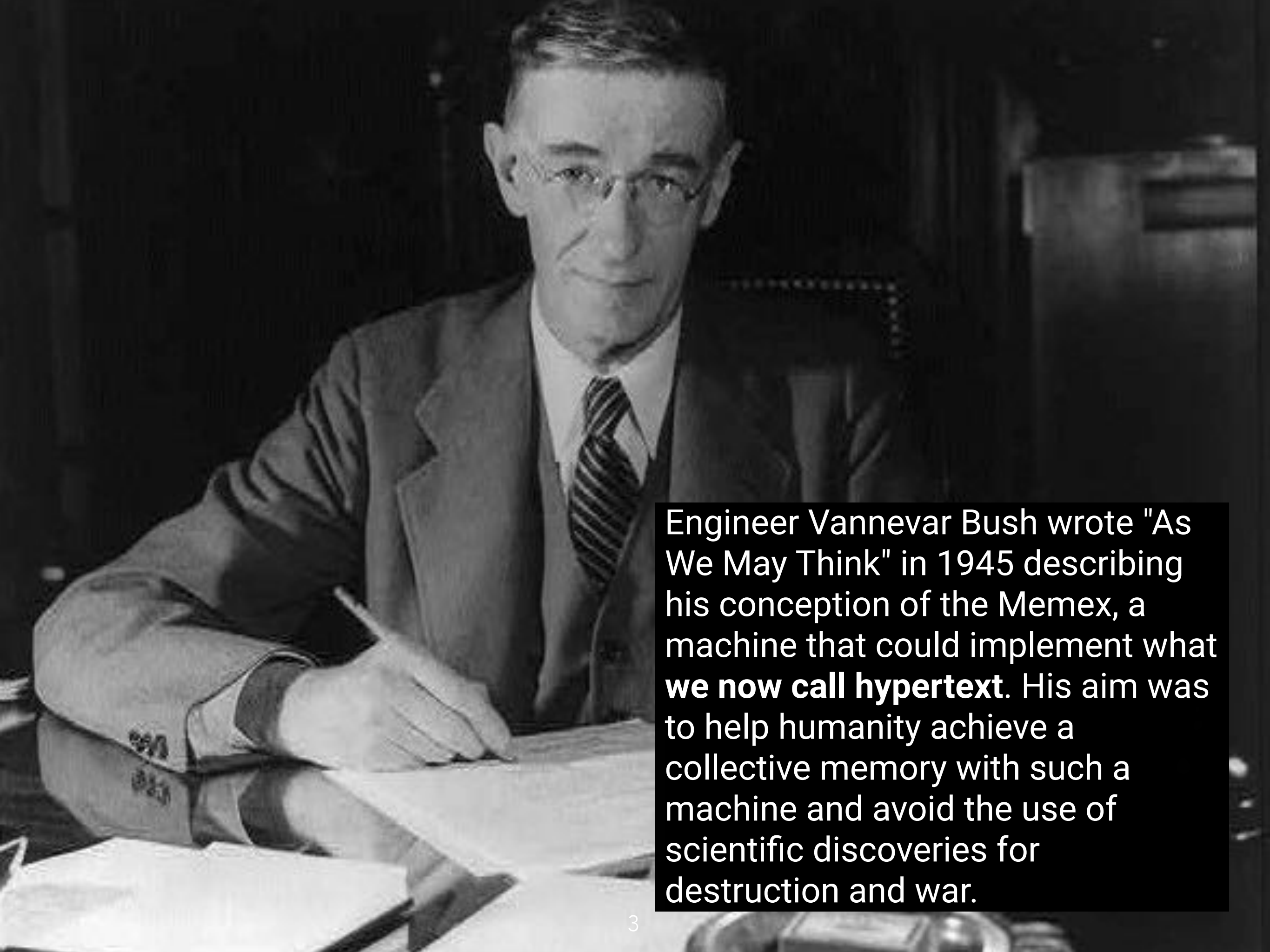


DIE AUSZEICHNUNGSSPRACHE

HTML KOMPAKT

WARUM EIGENTLICH HTML?

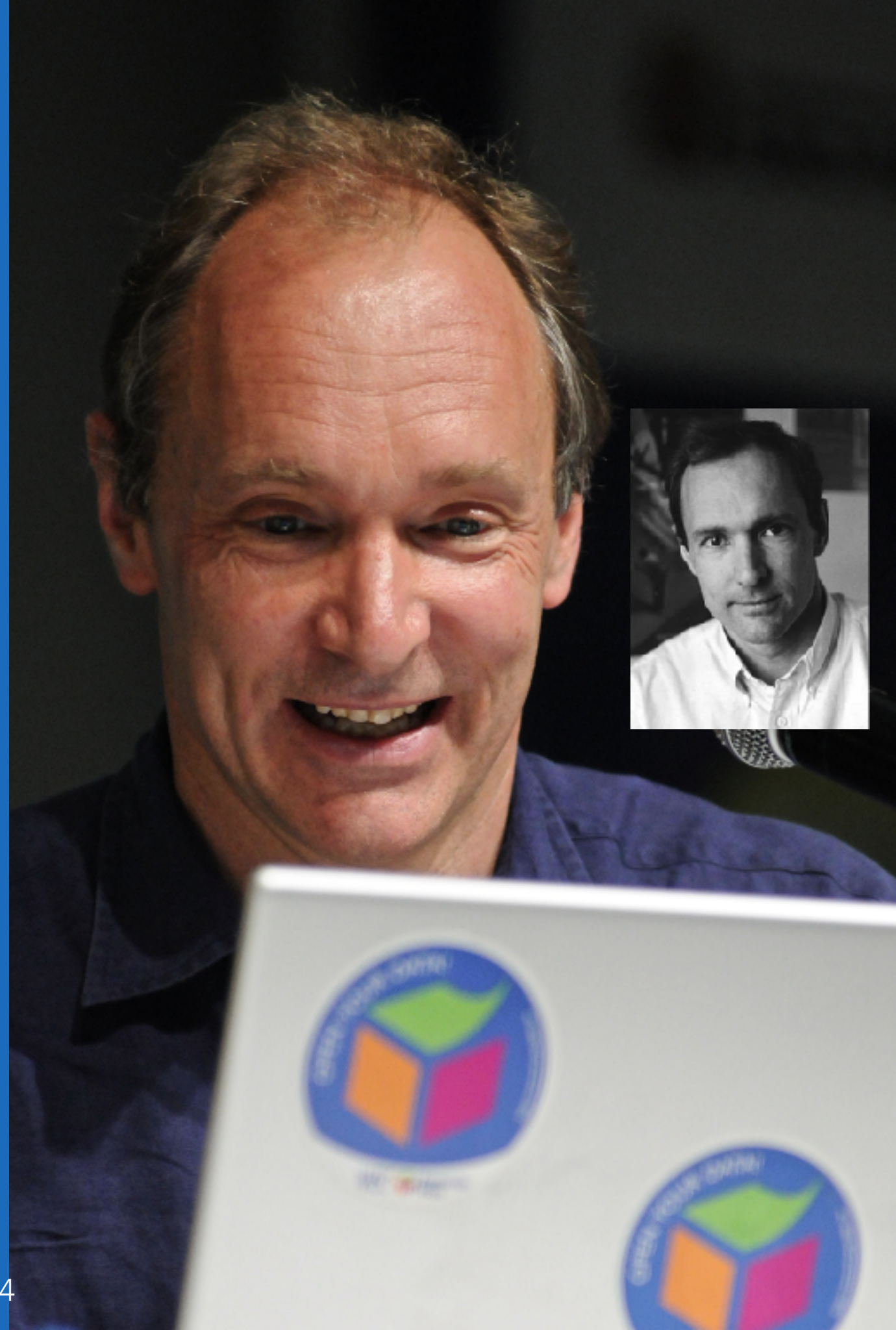
DIE ENTSTEHUNGSGESCHICHTE VON DER HYPertext MARKUP LANGUAGE



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.

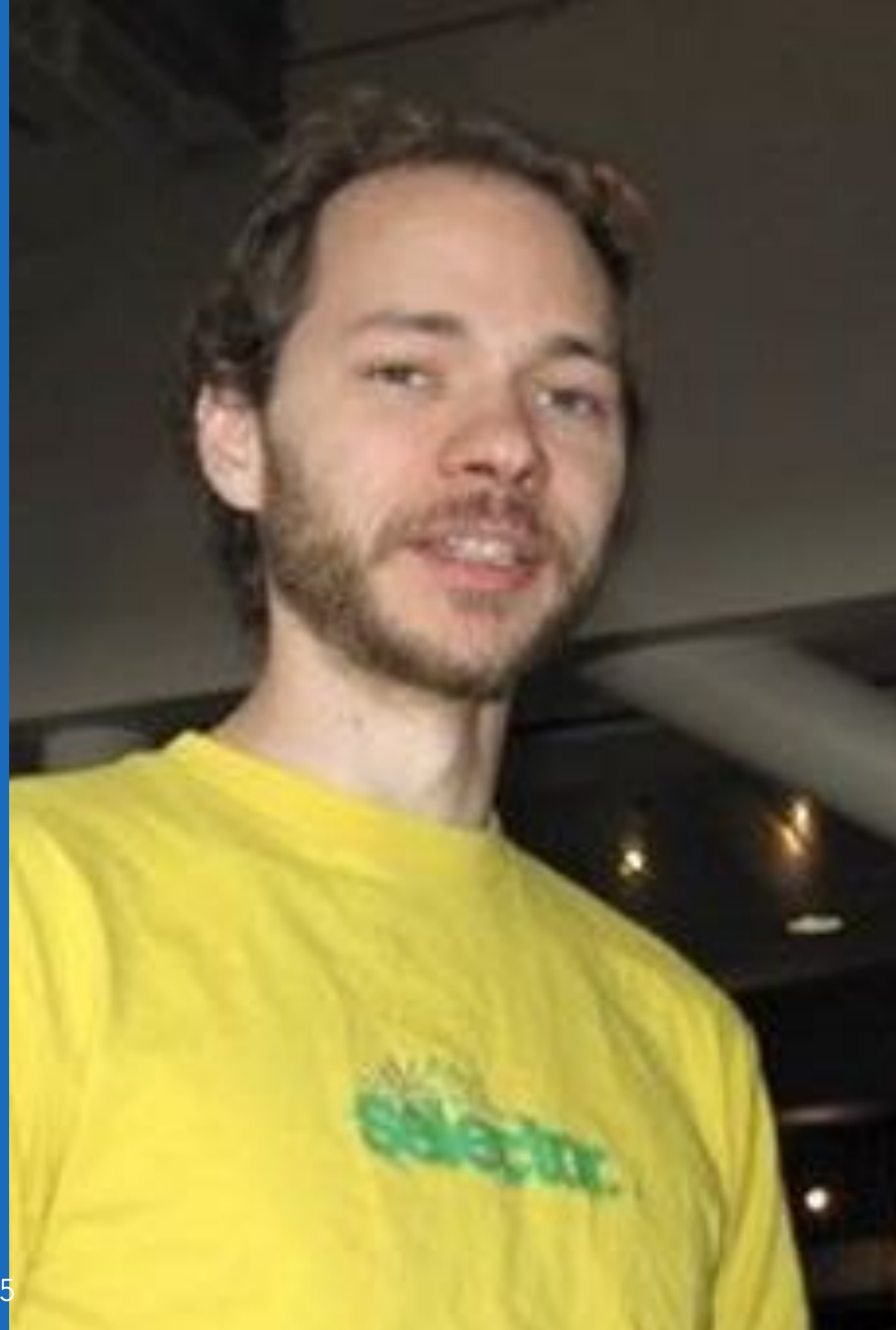
DER ERFINDER DES WORLD WIDE WEB

- 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.



- **1992HTML**

- 1994HTML 2
(Netscape)
Gründung des W3C
durch Tim Bernes-Lee
und andere

- 1995Mocha,
Livescript (Netscape,
Brendan Eich)

- 1995 Microsoft
Internet Explorer

- 1996 Opera

- Microsoft und
Netscape führen
einen Browser"krieg"

- 1996HTML 3.2, CSS
1, JavaScript
(Tim Berners-Lee
rettet HTML durch
das W3C und
Spezifikationen)

- **1997 HTML 4,
HTML 4.01**

- 1998CSS 2

- 2000XHTML 1.1

- 2002Microsoft IE 6:
Tableless Web Design
(HTML4 und CSS2)

- **2003Microsoft
entwickelt das
XMLHttpRequest
(AJAX)**

- 2003 Apple Safari
Mozilla Firefox

- 2004Gründung der
WHATWG
Web Applications 1.0

- 2008 Googles
Chrome

- **2010W3C
HTML5 Draft**

- 2014HTML5
Recommendation

- 2015 Microsoft Edge

- 2016HTML 5.1
Recommendation
HTML 5.2 Draft

HTML5 PROJEKTHEAD HEUTE

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

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

`wichtiger Text`

`betonter Text`

`<u>Unterstrichener Text</u>`

`<code>Codesnippets</code>`

`<kbd>Keyboard</kbd>`

HTML STRUKTUR-ELEMENTE

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

SEMANTISCHE INLINE ELEMENTE

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

HTML STRUKTUR-ELEMENTE

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

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; }
```

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

TABELLEN

TABELLEN - GRUNDGERÜST

<code><table></code>	Tabellencontainer
<code> <tr></code>	Tabellenzeile
<code> <td>Feld 1</td></code>	Tabellenfeld
<code> <td>Feld 2</td></code>	
<code> </tr></code>	
<code> <tr></code>	Tabellenzeile
<code> <td>Feld 1</td></code>	Tabellenfeld
<code> <td>Feld 2</td></code>	
<code> </tr></code>	
<code> <tr></code>	Tabellenzeile
<code> <td>Feld 1</td></code>	Tabellenfeld
<code> <td>Feld 2</td></code>	
<code> </tr></code>	
<code></table></code>	

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>
```

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>
```

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>

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

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

    <button>login</button>

  </fieldset>
</form>
```

The diagram illustrates the visual structure of the HTML form using nested boxes. The outermost box represents the form. Inside it, a box labeled 'Login' represents the first fieldset. Within the 'Login' box, there are two stacked boxes: the top one is labeled 'Email' and contains an input field; the bottom one is labeled 'Password' and contains an input field. Below these two boxes, still within the 'Login' box, is a 'Login' button. This visual nesting corresponds to the nested fieldset and legend tags in the code on the left.

DAS INPUT ELEMENT KENNT VIELE TYPEN

<code>type="checkbox"</code>	Checkbox
<code>type="file"</code>	Dateiupload
<code>type="hidden"</code>	Verstecktes Formularfeld
<code>type="password"</code>	Passwordfeld
<code>type="radio"</code>	Optionsfeld
<code>type="text"</code>	Einzeiliges Textfeld
<code>// deprecated</code>	
<code>type="button"</code>	beliebige Schaltfläche
<code>type="image"</code>	Eigener Button aus Grafik
<code>type="reset"</code>	Zurücksetzenbutton
<code>type="submit"</code>	Absendenbutton

WEITERE DEFINIERTE INPUT TYPEN AUS HTML5

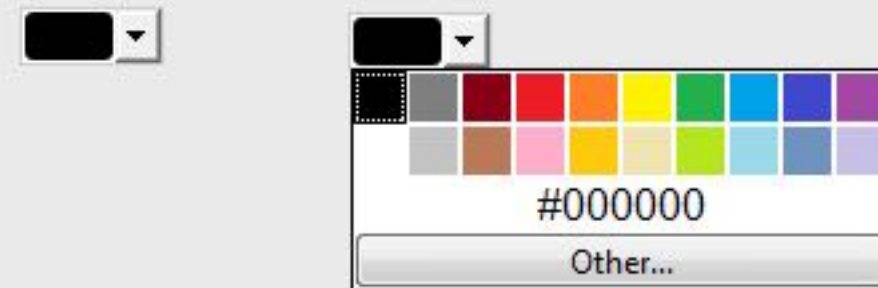
```
<input type="email">  
<input type="url">  
<input type="text">  
<input type="search">
```

`<input type="date">` und
Subtypen

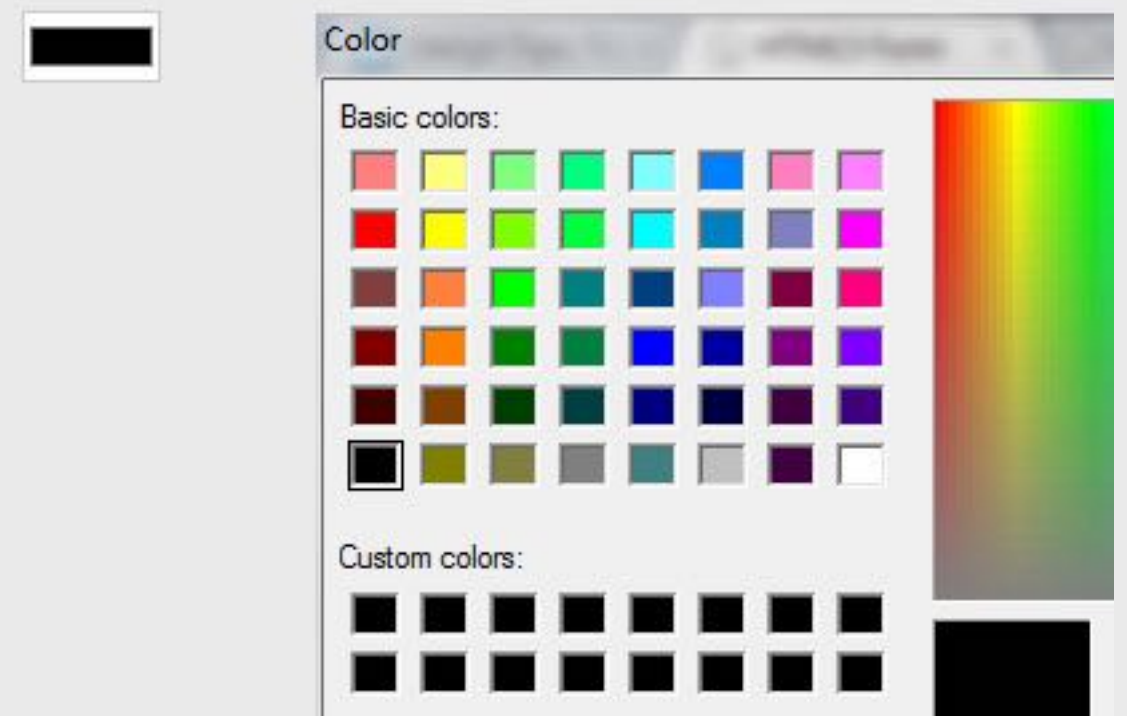
```
<input type="color">
```

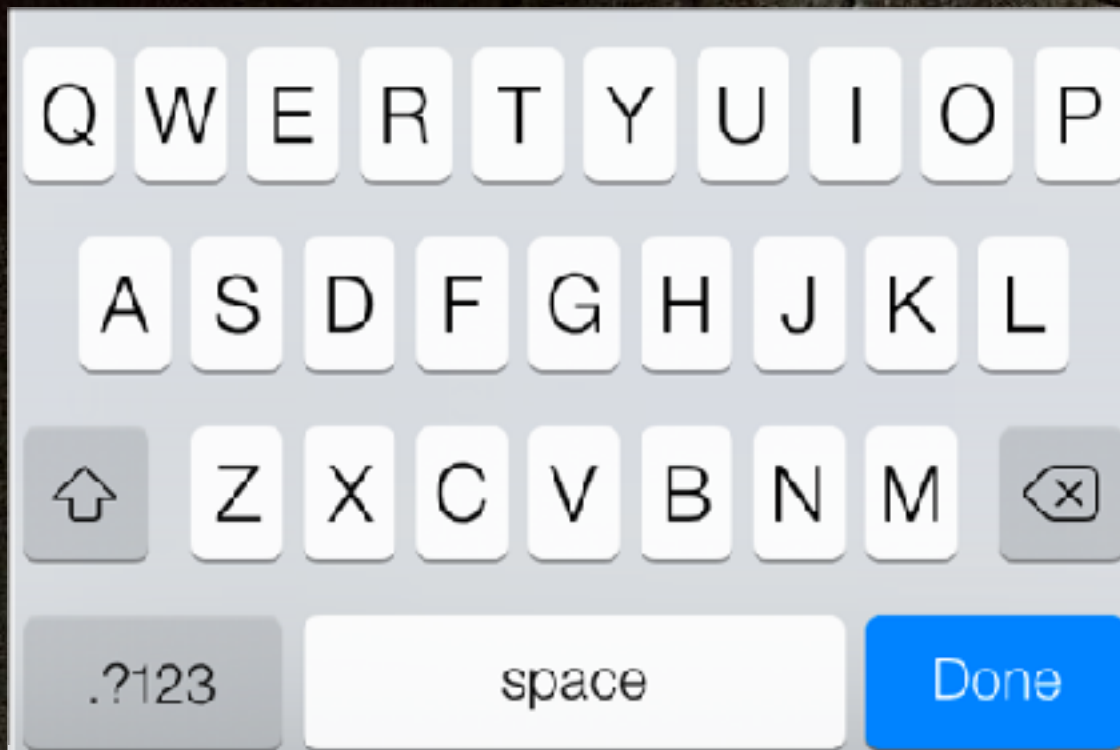
```
<input type="range">  
<input type="tel">  
<input type="number">
```

NATIVE COLOR PICKER IN OPERA

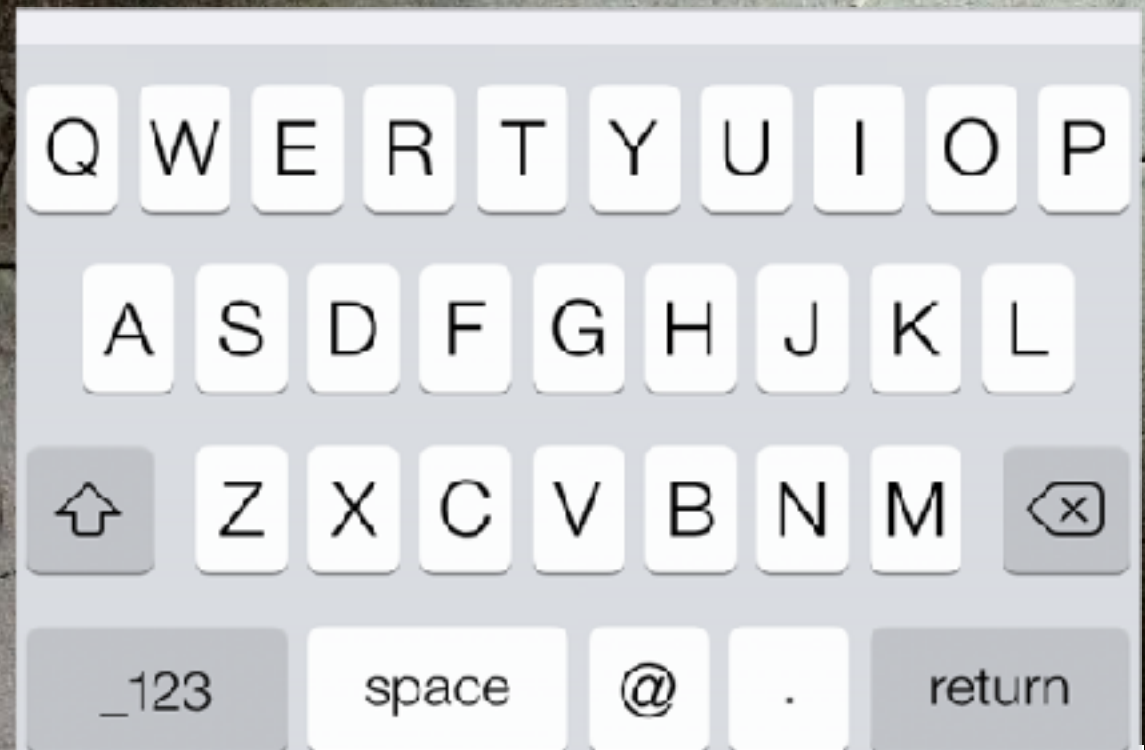


NATIVE COLOR PICKER IN CHROME

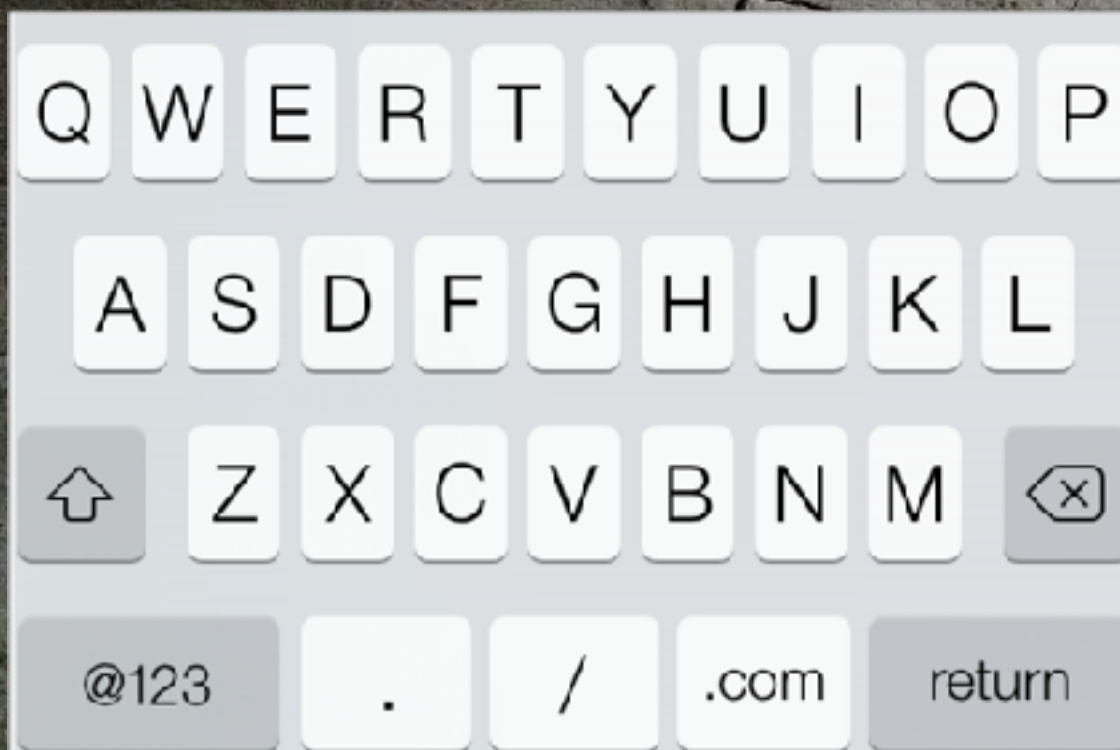




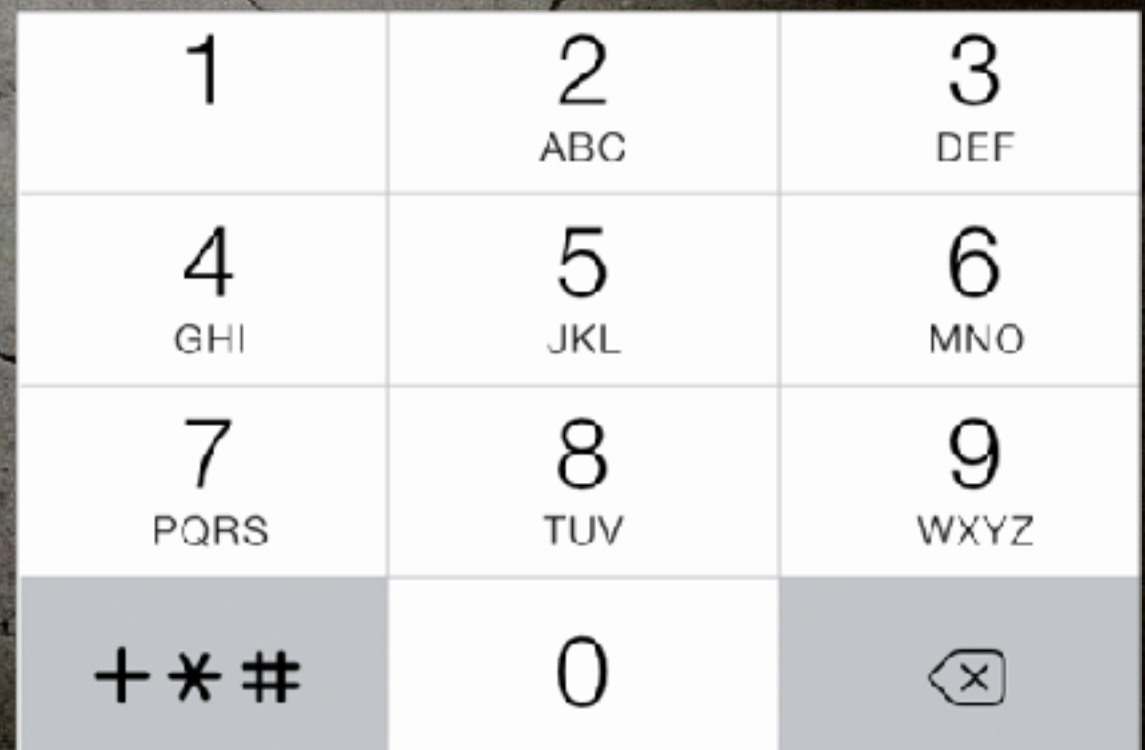
Default



Email

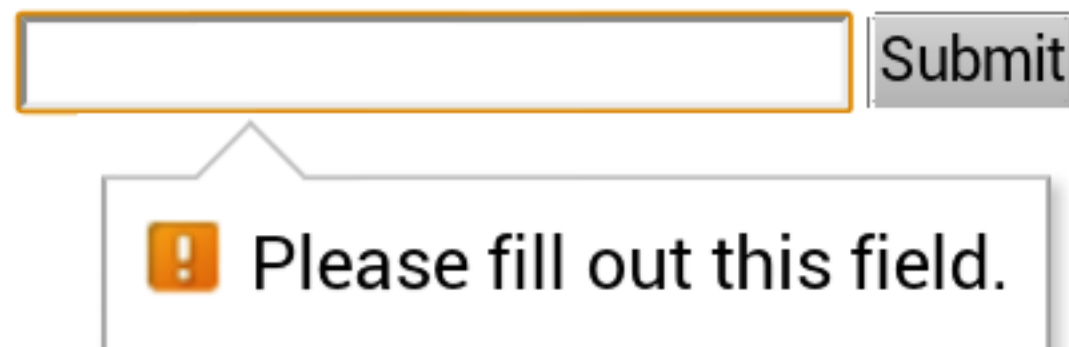


URL



Phone

VALIDIERUNG UND STATEKLASSEN FÜR EINGABELELEMENTE



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

<code>id="any unique value"</code>	Uniquewert als Ident
<code>name="a_name"</code> <code>value="any value ..."</code>	Name für den Versand Voreingestellter Wert
<code>placeholder="value ..."</code> <code>required="required"</code>	Platzhaltertext (HTML5) Pflichtfeld (HTML5)
<code>checked="checked"</code> <code>selected="selected"</code> <code>multiple="multiple"</code>	Ausgewählte Checkbox Ausgewählter Listeneintrag Mehrfachauswahl bei Listen
<code>accesskey="e"</code> <code>tabindex="1"</code>	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

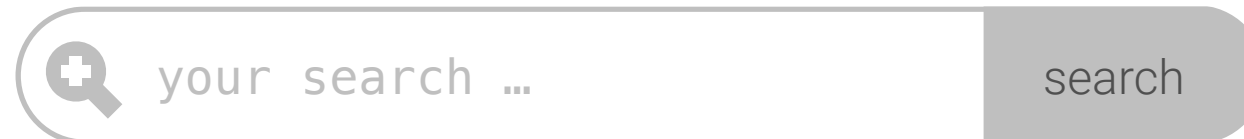
EIN MEHRZEILIGES EINGABEELEMENT

```
<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>
```


DROPDOWN 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.

BEKANNTE HTML ELEMENTE MIT NEUER BEDEUTUNG

BEKANNTE HTML ELEMENTE MIT NEUER BEDEUTUNG

`<i>`, ``, `<u>` – nicht-semantische grafische Auszeichnungen

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

TAGS, DIE NUR DER FORMATIERUNG DIENEN, FALLEN WEG.

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

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

ATTRIBUTE, 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

<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.

<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> hàn </rt><rp>)</rp>
```

```
  字 <rp>(</rp><rt> zì </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">
```

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

SEMANTISCHE HTML5 ELEMENTE

<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>
```

Nebenbeinhalt, 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.

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));
```


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, sondern hat sich für RDF entschieden.

RICH SNIPPETS MIT MICRODATEN

```
<div itemscope itemtype="http://schema.org/Event">
  <a itemprop="url" href="nba-miami-philidelphia-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

fo = Pizza My Heart

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

"DATA-" ATTRIBUTE

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"  
>
```

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

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

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.

DEN VIEWPORT FÜR FEATUREPHONES UND WINDOWSPHONES EINRICHTEN

```
<meta name="HandheldFriendly" content="true">  
<meta name="MobileOptimized" content="320">
```

Um bei Featurephones anzuzeigen, dass die Seite für 320px optimiert wurde, wird "HandheldFriendly", für Windowsphones das das "MobileOptimized" verwendet.

ICONS FÜR DEN BROWSER UND MOBILE GERÄTE

```
<link rel="apple-touch-icon-precomposed" sizes="144x144"
      href="src/ico/apple-touch-icon-144-precomposed.png" />
<link rel="apple-touch-icon-precomposed" sizes="114x114"
      href="src/ico/apple-touch-icon-114-precomposed.png" />
<link rel="apple-touch-icon-precomposed" sizes="72x72"
      href="src/ico/apple-touch-icon-72-precomposed.png" />
<link rel="apple-touch-icon-precomposed"
      href="src/ico/apple-touch-icon-57-precomposed.png" />

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

apple-touch-icons werden von iOS verwendet, um ein Icon auf der iPhoneoberfläche als Bookmark zur Seite/App anzulegen. Die verschiedenen Auflösungen berücksichtigen die Displayvarianten.

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 WEBCRAWLER II

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

Die Suchmaschine soll nicht die Beschreibungstexte des Open Directory Projects verwenden – sofern die Seite dort eingetragen ist –, sondern einen Ausschnitt der Seite selbst. Diese Anweisung wird von Google und MSN Search unterstützt.

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

Die Suchmaschine soll anstelle der Beschreibungstexte aus dem Yahoo-Verzeichnis einen Ausschnitt der Seite selbst. Diese Anweisung wird nur von Yahoo (Robot: Slurp) unterstützt.

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" />
```

Autor: Name des Autors und inhaltlich Verantwortlichen.

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

Verlinken externer Stylesheets

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

Externe Stylesheets bieten den Vorteil der Wiederverwendbarkeit und Modularität.

Sie werden im Kopfteil der HTML eingebunden, damit sie möglichst frühzeitig zur Verfügung stehen.

BROWSER- KOMPATIBILITÄT

GERÄTESPEZIFISCHE EINSTELLUNGEN

```
<meta name="apple-mobile-web-app-capable"  
      content="yes">
```

Adressleiste und Toolbar in Safari verschwinden.

```
<meta name="apple-mobile-web-status-bar-style"  
      content="black">
```

Farbe der Statuszeile wird auf Schwarz gesetzt.

```
<link rel="apple-touch-start-image"  
      href="img/splash.png">
```

Ein Bild als Startbild laden und anzeigen.

```
<meta http-equiv="cleartype"  
      content="on">
```

Sorgt für eine bessere Schriftdarstellung
im mobilen Internet Explorer.

JAVASCRIPT ZUM FIXEN VON IE FEHLERN VON VERSION 5.5 - 8

```
<!--[if lte IE 8]>  
  <script type="text/javascript" charset="utf-8"  
    src="http://ie7-js.googlecode.com/svn/  
      version/2.0(beta3)/IE8.js"></script>  
<![endif]-->
```

Conditional Comments!

Das bedingte HTML ist eine Internet Explorer eigene Angelegenheit. Sie wird von alle anderen Browsern ignoriert.

IE SPEZIFISCHE ANPASSUNGEN FÜR VERSIONEN < 9

```
<!-- HTML5 shim, for IE6-8 support of HTML5 elements -->  
<!--[if lt IE 9]>  
    <script src="lib/js/html5shiv.js"></script>  
<![endif]-->
```

vorher: <section></section> -> section {display:inline-block}
-> passiert nix!
-> createElement('section'); // Jetzt geht es!

Hier kommt zunächst das eigene CSS ...

```
<!-- CSS for IE6-8 -->  
<!--[if lt IE 9]>  
    <link rel="stylesheet" href="ie-color.css" />  
<![endif]-->
```

RESET.CSS

<http://meyerweb.com/eric/tools/css/reset/>

Setzt alle browser-eigenen Eigenschaften der HTML Elemente auf null. Damit ist es möglich, ein für alle Browser kompatibles CSS-basiertes Grundlayout zu gestalten.

```
<link rel="stylesheet" type="text/css" media="all"  
href="https://cdnjs.cloudflare.com/ajax/libs/meyer-reset/2.0/reset.min.css" />
```

NORMALIZE.CSS

`https://necolas.github.io/normalize.css/`

Erzeugt ein einheitliches Erscheinungsbild für alle Browser. Normalize wurde von Nicolas Gallagher geschrieben und beständig weiter entwickelt.

```
<link rel="stylesheet" type="text/css" media="all"  
href="https://cdnjs.cloudflare.com/ajax/libs/normalize/5.0.0/normalize.css" />
```

DIE BODY SEKTION

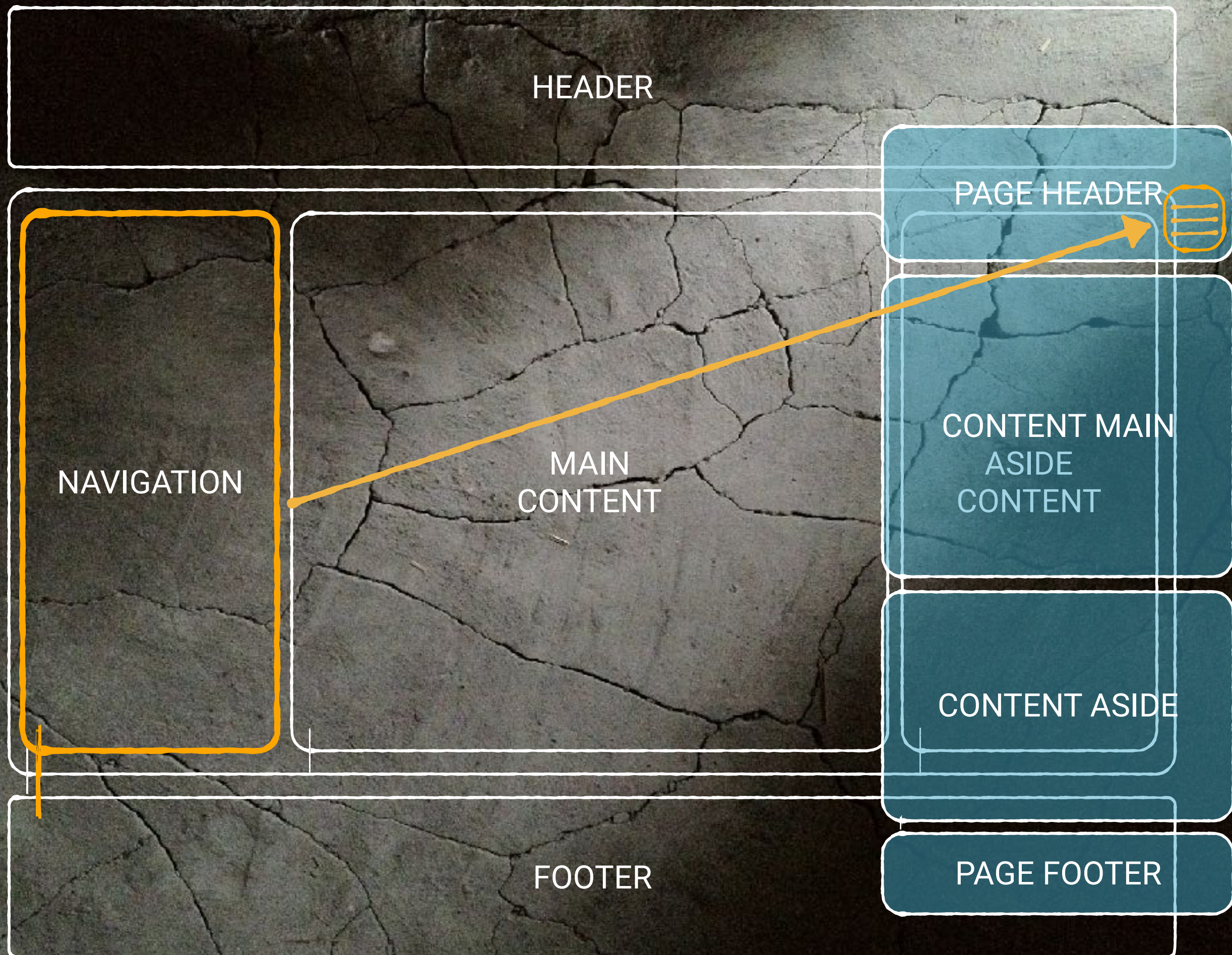
HEADER

NAVIGATION

MAIN
CONTENT

ASIDE
CONTENT

FOOTER



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>
```

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"> ... </div>
```

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-SEMANTISCHEN 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 NAVIGATIONSKOMPONENTE

```
<!-- 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 NAVIGATIONSKOMPONENTE

```
<!-- 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 NAVIGATIONSKOMPONENTE MIT CSS KLASSEN ZUR GESTALTUNG UND ANORDNUNG

```
<!-- 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>
```