

HTMLCSSJavaScript: Einführung

- Grundsätzliches zu HTML-Dokumenten, Whitespaces, DOM
- Grundgerüst eines HTML-Dokumentes
- Kopfdaten, Integration von CSS und JavaScript
- JavaScript: Aufruf, Programmierung
- HTML5-Elemente zur Dokumentstrukturierung
- Wichtige CSS-Formate: margin, border, padding, float, clear
- Zuweisen von CSS-Formaten zu HTML-Elementen

Literaturhinweise

SELFHTML HTML-Dateien selbst erstellen, Stefan Münz

frei verfügbar unter

<http://de.selfhtml.org>

HTML 5 Handbuch, Stefan Münz, Clemens Gull, Franzis Verlag,

ISBN 978-3645600798

frei verfügbar unter

<http://webkompetenz.wikidot.com/docs:html-handbuch>

HTML5 – Webseiten innovativ und zukunftssicher, Open Source Press,

ISBN 978-3-941841-34-5

HTML5 & CSS3 – Webentwicklung mit den Standards von morgen,

Brian P. Hogan, O'Reilly Verlag, ISBN 978-3-89721-316-6



Grundsätzliches zu HTML-Dokumenten

```
<a href="http://www.google.de">Google <i>Suchmaschine</i></a>
<!-- Dies ist ein Kommentar -->

```

- HTML-Dokument besteht aus HTML-Elementen, welche durch *Tags* beschrieben werden
- Einleitendes und schließendes Tag
- Groß-/Kleinschreibung irrelevant
- Grundlegende Tags `<p>`, ``, ``, `<h1>`, `<table>`
- *Standalone-Tags* `
`, `<hr>`, ``, `<input>`, `<meta>`
- Verschachtelung von Elementen
- *Attributwerte* in Tags mit Anführungszeichen
- *HTML-Kommentare*

HINWEIS: XHTML verlangt Kleinschreibung in Tags, es gibt keine Standalone-Tags (`
</br>` oder `
` werden verwendet)

Zeichensatz und Sonderzeichen

```
<meta charset="utf-8">
```

<	<	(less than)	>	>	(greater than)	
&	&	(Ampersand)	"	"	(Quote)	€

Whitespaces

- Mehrere aufeinanderfolgende Whitespaces werden ignoriert
- Whitespaces am Anfang eines Elementinhaltes werden ignoriert
- Absatzschaltungen und Tabulatoren in Whitespace konvertiert
- Geschütztes Whitespace durch ` `;
- Schlupfloch `<pre>`

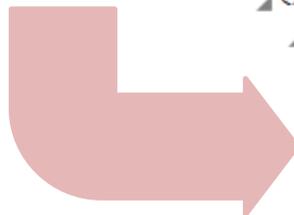
Beispiel

```

<!doctype html>
<html lang="de">
  <head>
    <meta charset="utf-8">
    <title>Grundsätzliches zu HTML-Dokumenter
  </head>
  <body>
    <a href="http://www.google.de">Google <i:
    <!-- Dies ist ein Kommentar -->
    

    <h1>&lt;&quot;HTML&quot;&gt;</h1>
    <hr>
    <p>Am Sonntag will mein Süßer mit&nbsp;&nbsp;&nbsp;&
    <pre>
      Segeln gehn
      das
      wird...
    </pre>
  </body>
</html>

```



```

<!DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="utf-8"></meta>
    <title>
      Grundsätzliches zu HTML-Dokumenten
    </title>
  </head>
  <body>
    <a href="www.google.de">
      Google
      <i>
        Suchmaschine
      </i>
    </a>
    <!-- Dies ist ein Kommentar -->
    </img>
    <h1>
      <"HTML">
    </h1>
    <hr></hr>
    <p>
      Am Sonntag will mein Süßer mit   m i r
    </p>
    <pre>
      Segeln gehn
      das
      wird...
    </pre>
  </body>
</html>

```

Document Object Model (DOM) ¹

- Baumstruktur des HTML-Dokumentes
- Jeder Eintrag wird als Knoten bezeichnet



¹ In Firefox über Tools für Web-Entwickler anzeigbar

Grundgerüst eines HTML-Dokumentes

```
<!DOCTYPE html>
```

- Legt den *Dokumenttyp* auf HTML5 fest
- Vorher keinen Kommentar positionieren, sonst wird IE6, 7, 8 die Seite im Kompatibilitäts- und nicht im Standardmodus anzeigen

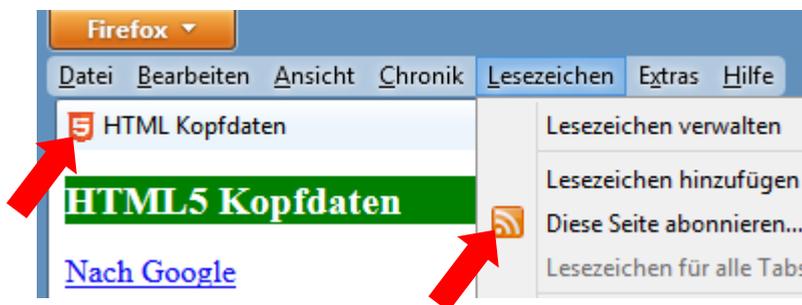
Das HTML-Dokument

```
<html lang="de"> ... </html>
```

- Inhalt des HTML-Dokumentes
 - Enthält evtl. die Spracheinstellung für Suchmaschinen
-

Kopfdaten

```
<head>
  <title>HTML Kopfdaten</title>
  <meta charset="utf-8">
  <meta name="author" content="Max Muster">
  <meta name="description" content="Profilseite Max ...">
  <meta name="robots" content="noindex, nofollow">
  <meta http-equiv="refresh" content="5; http://...">
  <link rel="alternate" type="application/rss+xml"
    href="feed.xml" title="RSS-Feed">
  <link rel="icon" type="image/x-icon" href="logo.ico"
    sizes="16x16">
  <link rel="stylesheet" type="text/css" href="layout.css">
  <style type="text/css">
    h1 { color: white; background-color: black }
  </style>
  <script type="text/javascript" src="scr.js"></script>2
  <script type="text/javascript">
    function clicked() {
      document.getElementById("ueberschrift").innerHTML =
        "HTML5 Kopfdaten und JavaScript";
    }
    function leave() {
      return confirm("Seite verlassen?");
    }
  </script>
</head>
<body>
  <h1 style="background-color: green; font-size: 20px"
    id="ueberschrift" onclick="clicked()">
    HTML5 Kopfdaten
  </h1>
  <a href="http://www.google.de"
    onclick="return leave();">Nach Google</a>
</body>
```



² <script> ist kein Standalone-Tag, ein evtl. im Tag enthaltener JavaScript-Code wird ignoriert

Positionieren von CSS und JavaScript

Datei

- `<link rel="stylesheet" type="text/css"href="...">`
- `<script type="text/javascript"src="..."></script>`

Head

- `<style type="text/css"> ... </style>`
- `<script type="text/javascript">...</script>`

Tag

- `<h1 style="..." > ... </h1>`
- `<h1 onclick="..."> ... </h1>`

3

- Titel in *Titelleiste*, *Browser-Tab*, als *Lesezeichen* und im *Verlauf*
- Titel hat in Suchmaschinen die höchste Relevanz, jede Seite sollte aussagekräftigen Titel haben
- Titel wird meist in Suchergebnissen als anklickbarer Verweis angeboten
- `charset="utf-8"`
Zeichensatz, in dem das Dokument abgelegt wird. Bei Editoren unter UNIX/Linux und MacOS X standardmäßig UTF-8, bei Windows nicht immer der Fall (siehe Notepad)
- `description` ... Seitenbeschreibung von Suchmaschinen ausgewertet
- `noindex` ... Seite wird von Suchmaschinen nicht gefunden
`nofollow` ... Links auf Seite werden von Suchmaschine nicht verfolgt
- `refresh` ... Weiterleitung nach 5 Sekunden zu Adresse
- `alternate` ... Verlinkt Seite mit alternativer Präsentationsform z.B. RSS-Feed
- `icon` ... Minigrafik die in *Titelleiste*, *Tab* und *Lesezeichen* meistens angezeigt wird

³ Ab HTML 5 kann `type`-Attribut bei CSS und JS weggelassen werden

HTML5–Elemente zur Dokumentstrukturierung

Vorteile

- *Semantische Tags* spiegeln Struktur komplexer Dokumente wieder
- Suchmaschinen und Screenreader können Dokumentinhalte besser interpretieren
- Keine Beschränkung auf maximal sechs Überschriftenebenen
- Verzicht auf verschachtelte, unübersichtliche `<div>`-Tags

<main> ... Hauptinhalt einer Seite

Durch die Auszeichnung des Hauptinhalts der Seite mit `main` können Screenreader und andere alternative Ausgabegeräte diesen erkennen und beispielsweise *Header* oder *Navigation* überspringen (siehe hinten)

- Maximal ein `main`-Element pro Seite

<section> ... Abschnitt

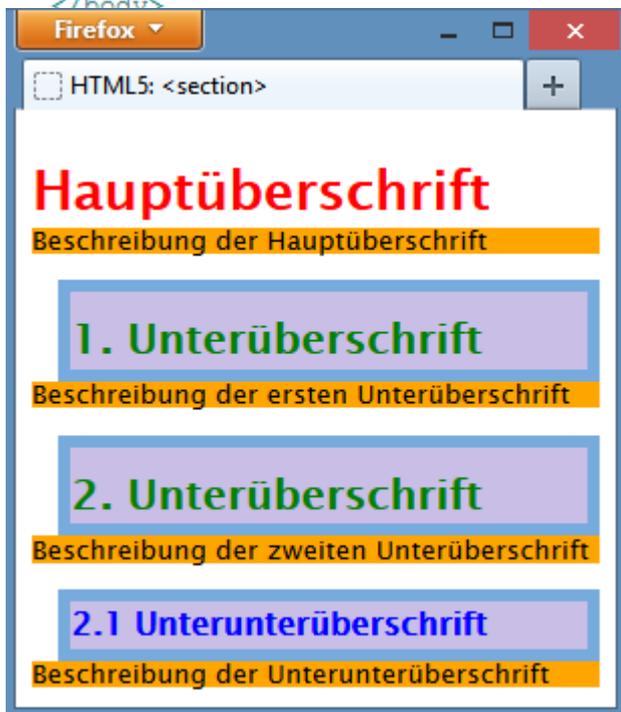
- Teilt umfangreiches Dokument auf, gliedert dieses
 - Enthält in der Regel nur Überschrift `<h1>`
 - Können beliebig verschachtelt werden
-

Beispiel: Abschnitte mit CSS⁴

```

<!doctype html>
<html>
  <head>
    <title>HTML5: &lt;section&gt;</title>
    <meta charset="utf-8">
    <link rel="stylesheet" type="text/css" href="html5reset.css">
    <link rel="stylesheet" type="text/css" href="section.css">
  </head>
  <body>
    <section>
      <h1>Hauptüberschrift</h1>
      <p>Beschreibung der Hauptüberschrift</p>
      <section>
        <h1>1. Unterüberschrift</h1>
        <p>Beschreibung der ersten Unterüberschrift</p>
      </section>
      <section>
        <h1>2. Unterüberschrift</h1>
        <p>Beschreibung der zweiten Unterüberschrift</p>
        <section>
          <h1>2.1 Unterunterüberschrift</h1>
          <p>Beschreibung der Unterunterüberschrift</p>
        </section>
      </section>
    </section>
  </body>

```



```

body {
  font-size: 10pt;
  font-family: "Lucida Sans Unicode", serif;
  margin: 6pt;
}
p {
  background-color: orange;
}
section h1 {
  font-size: 20pt; color: red;
  padding-top: 15pt; padding-bottom: 3pt
}
section section h1 {
  background-color: rgb(200,190,230);
  font-size: 16pt; color: green;
  margin-top: 10pt; margin-left: 10pt;
  border: solid 5pt #77AADD;
  padding-top: 10pt; padding-bottom: 3pt
}
section section section h1 {
  font-size: 12pt; color: blue;
  padding-top: 3pt; padding-bottom: 3pt
}

```

margin
Außenabstand

border
Rand

padding
Innenabstand

⁴ Standard-CSS-Formate des Browsers löschen siehe <http://html5doctor.com>

<article> ... Artikel

- Eigenständiger, logischer Teil innerhalb einer Webseite
- Enthält den eigentlichen Inhalt der Webseite
- Verwendbar für Blog- oder Newsartikel, Forenpostings, Benutzerkommentare, usw.
- Kann geschachtelt werden

<header> und <footer> ... Kopf- und Fußbereich

- Enthalten Überschriften, Unternehmenslogo, Suchfeld, Navigation (<nav> rechtliche Hinweise, Impressum, Copyright, Geschäftsbedingungen), usw.
- Auch in <section>, <article> verwendbar
- Dürfen keine weiteren <header> und <footer> enthalten

<nav> ... Navigation

- Beinhaltet Navigationselemente
- Kann mehrmals eingefügt werden (Hauptnavigation und Artikelnavigation)

<aside> ... Zusatzinhalte

- Enthält ergänzende Inhalte zur umgebenden Information
- Z.B. Zitate, Interviews, Diagramme, Bilder, usw.

<address> ... Autorenangaben

- Enthält Kontakt- und Autoreninformationen
 - Darf nicht <header>, <footer>, <article> oder <address> enthalten
-

Beispiel: Artikel mit float und clear

```

<article>
  <h1>Überschrift Artikel</h1>
  <address>Autorenangaben</address>
  <aside>
    Zusatzinhalte
    <ul>
      <li>Zitate</li>
      <li>Interviews</li>
      <li>Diagramme</li>
      <li>Bilder</li>
    </ul>
  </aside>
  <p>
    Text zum Artikel. Text zum Artikel. Text zum A
    Text zum Artikel. Text zum Artikel. Text zum A
  </p>
  <p style="clear: right">
    Text zum Artikel. Text zum Artikel. Text zum A
    Text zum Artikel. Text zum Artikel. Text zum A
  </p>
  <nav>
    <a href="...">Artikelnavigation</a> <a href="...
  </nav>
</article>

```

```

p {
  background-color: orange;
  margin-bottom: 5pt;
}
ul {
  margin-left: 12pt;
}
article {
  background-color: rgb(180,180,180);
  padding: 10pt;
  border-radius: 5pt;
}
article h1 {
  font-size: 20pt;
  color: red;
  padding-top: 15pt;
  padding-bottom: 3pt
}
address {
  background-color: white;
  padding: 3pt;
  border-radius: 5pt;
}
nav {
  background-color: white;
  border-radius: 5pt;
  text-align: center;
}
aside {
  background-color: yellow;
  float: right;
  width: 30%;
  padding: 3pt;
  margin: 3pt;
  border-radius: 3pt;
  border: solid 1pt;
}

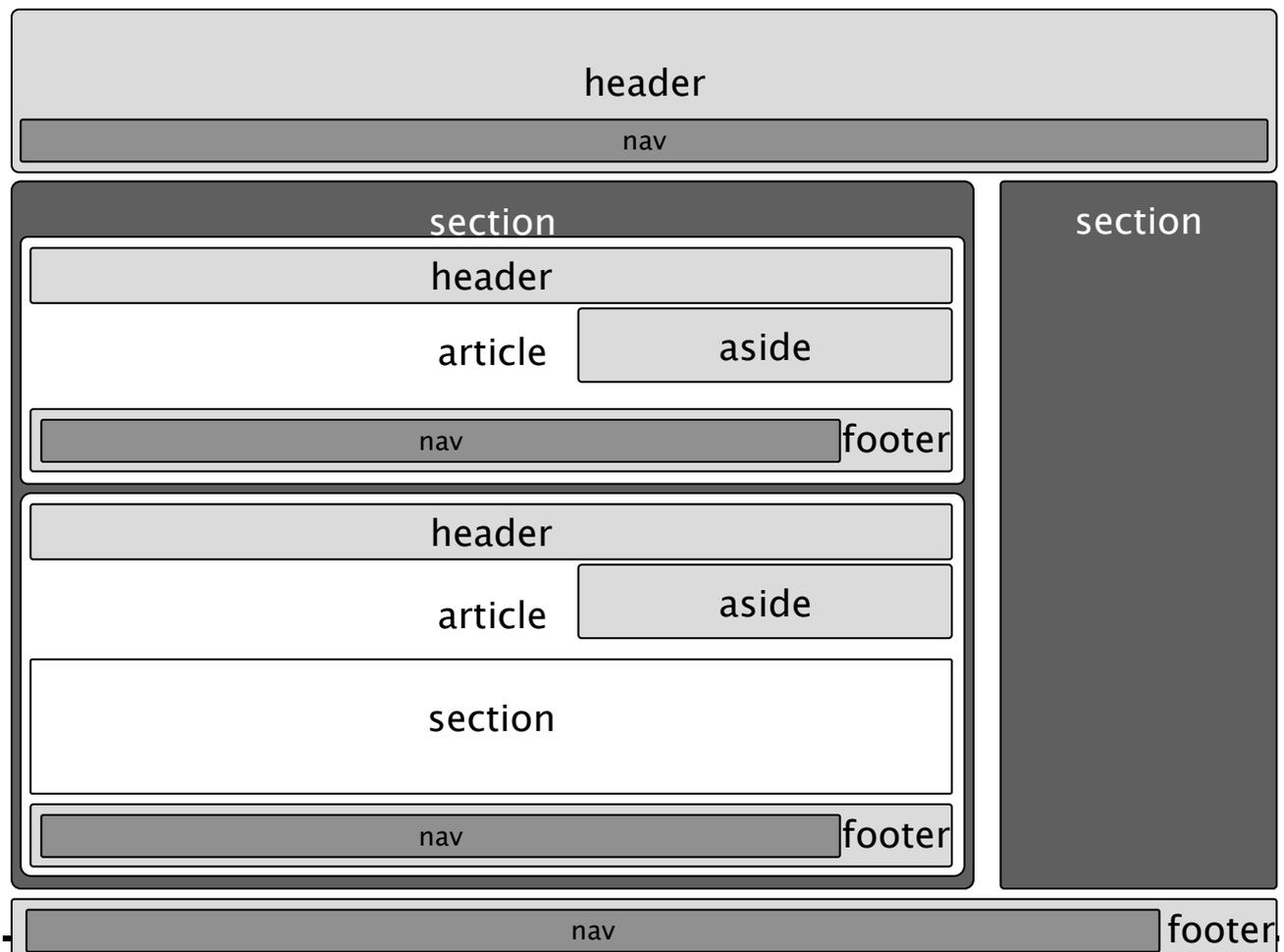
```



float: left|right

clear: left|right|both

Beispiel: Struktur einer Web-Seite



Beispiel: CSS-Formate für HTML-Elemente

```
<h1>Erste Überschrift</h1>
<p>Dies ist er Text zur ersten l
<ul>
  <li>Erste<br>Aufzählung</li>
  <li>Zweite Aufzählung</li>
  <li>Dritte Aufzählung</li>
</ul>
```



```
body {
  background-color: #FFFFCC;
  margin-left:100px;
}
* {
  color: tomato;
}
h1 {
  margin-top: 20px;
  font-size: 300%;
  color: blue;
  font-style: italic;
  border-bottom: dotted thick rgb(45,115,37);
}
p, li {
  margin: 6pt;
  font-size: 110%;
  line-height: 1.6em;
  font-family: Helvetica, Arial, sans-serif;
  letter-spacing: 0.1em;
  word-spacing: 0.3em;
}
li {
  margin-left: 100px;
  list-style-position: inside;
  list-style-image: url("tasche.png");
}
```

Farbangaben

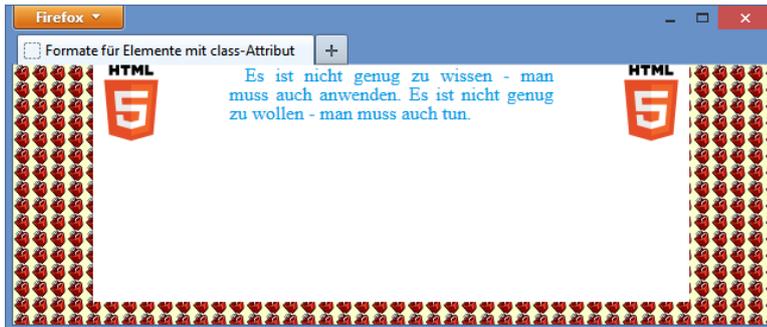
Hexadezimal (#FFFFCC), *Farbnamen* (powderblue), *rgb-Format* (rgb(60%, 90%, 75%))

Maßeinheiten

px (*Pixel*), pt (*Punkt*) und pc (*Pica*) (typografische Maßeinheiten), em (*Schriftgröße* des Elementes, z. B. 0.1em = 10% der Schriftgröße)

Beispiel: Formate für Elemente mit class-Attribut

```
<div class="wrapper">
  
  
  <p class="lichtblau blocksatz">
    Es ist nicht genug zu wissen - man
    Es ist nicht genug zu wollen - man
  </p>
</div>
```



Wrapper zentriert Inhalte im Browser
auch dann wenn Fenster in seiner Größe
geändert wird

```
body {
  background: url("tasche.png");
}
div.wrapper {
  position: absolute;
  top: 0px;
  left: 50%;
  width: 500px;
  height: 200px;
  margin: 0px auto 0px -250px;
  background-color: white;
}
img.left {
  float: left;
  margin-right: 50px;
}
img.right {
  float: right;
  margin-left: 50px;
}
.lichtblau {
  color: rgb(0, 162, 232);
}
.blocksatz {
  text-indent: 10pt;
  text-align: justify;
}
```

Auch mit id-Attribut können Formate zugeordnet werden

- id-Attribut identifiziert ein Element im HTML-Dokument und sollte in HTML-Dokument nur einmal verwendet werden
- Definition durch # anstelle des Punktes