

HTML5-Kurs

Hilfsmittel:

<https://www.w3schools.com/default.asp> HTML & JavaScript-Seite

<http://www.css4you.de> CSS-Zusammenstellung

Abkürzungen & Ausdrücke:

API -> Application Programming Interface (Programmierschnittstelle)

Umlaute: ä -> Ää / ö -> Ä¶ / ü -> Ä¼

HTML-Code

<u>Code</u>	<u>Erklärung</u>
<html>	Beginn html-Seite
<!DOCTYPE html PUBLIC>	Html 4 (veraltet!!)
<!DOCTYPE html>	Html 5
<html lang="de">	
<head>	Kopf (wird geöffnet)
<meta charset="utf-8">	
<meta http-equiv="...">	
<meta name="Autor">	
<meta name="Generator">	
<script type="style.css" rel="stylesheet" type="text/css">	Festlegung des JS im Head
<link href="../css/main.css" rel="stylesheet" type="text/css">	Festlegung des CSS im Head
<title> zawadi_HP </title>	Seitenname Festlegung des Titels im Head
</head>	Kopf (wird geschlossen)
<body>	html-Seitenbeginn ohne Hintergrundfarbe
<body bgcolor="FFFF00">	html-Seitenbeginn mit Hintergrundfarbe gelb
<center>	zentrieren
</center>	zentrieren aufheben
<p>....</p>	neuer Absatz mit Text
<p style="text-align: center"></p>	eingemittete Text-Zeile
<div align="center"></div>	eingemitteter Abschnitt
<button id="xyz">Titel</button>	einfügen eines Knopfes
	Schriftartwahl aus Cascading Style Sheet
	Schriftart Arial
	Schriftgröße 16 (11 + 4)
	Schriftart & Schriftgröße löschen
	Schriftfarbe
	Bild bestimmter Größe einfügen
width=59 align=left>	setzt Bild links neben Text
width=59 align=center>	setzt Bild zeilenmitten ein
<p style="text-align: center;">	Bild einfügen / einmitten

<code><hr width="100%"></code>	waagrechtcr Trennstrich
<code><u> bzw. </u></code>	unterstreichen veraltet!!
<code> </code>	unterstreichen (neu)
<code> bzw. </code>	Fettschrift ein bzw. aus
<code><i> bzw. </i></code>	Kursivschrift ein bzw. aus
<code><a> ... </code>	definiert ein Hyperlink von einer Seite zur anderen
<code>href=" .. "</code>	definiert ein Link zur Zielseite mit der eingesetzten URL
<code>Hobbies</code>	Link auf Quelltext
<code>Data Becker</code>	Link auf Internetziel
<code>Mail to Walo</code>	Versand eines E-Mail
<code> </code>	Bild mit Link
<code></code>	Bild von einer anderen Website
<code></code>	Bild ab Verzeichnis images
<code></code>	Bild mit ID=xy
<code><h1>Text</h1></code>	Überschrift der Grösse 1
<code>&nbsp;</code>	Leerschlag
<code>&uuml;</code>	Umlaut ü
<code>
</code>	Zeilenumbruch ohne Absatz
<code><table style="width:700px" border="0" align="center"></code>	Tabelle eröffnen
<code><tbody></code>	
<code><tr></code>	Zelle eröffnen
<code><td style="width: 50px">Zelle</td></code>	einzelne Zelle
<code></tr></code>	Zelle schliessen
<code></tbody></code>	
<code></table></code>	Tabelle beenden
<code><marquee > Lauftext </marquee></code>	Lauftext einblenden

Erkenntnisse und Vermutungen:

- Bei gleichen CSS-Einträgen gilt immer der letzte in der Datei. Er überschreibt den vorderen im Browser-Speicher!
- Ergänzungslinks in der index.php überschreiben den vorliegenden CSS-Eintrag.
- Links ab index.php sind immer vom Root her gesehen.
- Links ab CMS css-Datei sind immer ab Ablageort dieser Datei
- Für Tipps & Tricks schaue unter <http://www.css4you.de>

HTML-Gerüst von HTML5:

Das Gerüst von HTML5 entspricht in etwa denjenigen älterer HTML-Dokumente (Rückwärtskompatibilität!).
Einzig der bis anhin verwendete DOCTYPE:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"><html
xmlns="http://www.w3.org/1999/xhtml">
```

wird stark vereinfacht und lautet **neu für HTML5 bis HTMLx:**

```
<!DOCTYPE html>
```

Gleichzeitig wird in den meisten Fällen auch gerade die **Codiersprache auf UTF-8** gesetzt. Im Folgenden wird die Strukturierung der zu erstellenden HTML5-Seite gezeigt:

Beispiel einer HTML5 Strukturierung

Die Grobstrukturierung des Body erfolgt in header-Element / Nav-Element / independent items of content (aus article-Element) / Zeilenweises Gruppieren der Items / Sections / Footer. So sieht die Feinstrukturierung der Elemente aus:

```
<!DOCTYPE html>
```

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

```
<head>
```

```
<meta charset = "UTF-8" />
```

```
<title> </title>
```

```
<script type="style.css" rel="stylesheet" type="text/css">
```

```
</head>
```

```
<body>
```

```
<header>
```

```
<div id="spinning-house"></div>
```

```
<hgroup>
```

```
<h1>Blue House Realty</h1>
```

```
<h2>Your place to find ...</h2>
```

```
</hgroup>
```

```
</header>
```

```
<nav class="gray-gradient border-radius box-shadow gray-border  
seperate">
```

```
<ul>
```

```
<li><a href = ....</a></li>
```

```
<li><a href = ....</a></li>
```

```
<li><a href = ....</a></li>
```

```
</ul>
```

```
</nav>
```

```
<div class="gray-gradient border-radius box-shadow gray-border  
seperate">
```

```
<article> xyz_1</article>
```

```
<article> xyz_3</article>
```

```
</div>
```

```
<div class="seperate space-top">
```

```
<article class="gray-gradient border-radius gray-border">
```

```
<div>xyz_4</div>
```

```
</article>
```

```
<article class="gray-gradient border-radius gray-border">
```

```
<div>xyz_5</div>
```

```
</article>
```

```
<article class="gray-gradient border-radius gray-border">
```

```

        <div>video</div>
    </article>
</div>

<section>
    <header>
        <h3>What our customers say</h1>
    </header>
    <p>text</p>
    <footer>
        Posted on: <time datetime="2010-10-09" pubdate>
    </footer>
</section>

<footer>
<p>Fusstext</p>
</footer>

</body>
</html>

```

HTML5 spezifische Elemente

Im dritten Teil werden nun HTML5-spezifische Elemente für Video, Geolocalisation, etc. hinzugefügt:

1 Video-Element:

```

<article>
    <div>
        <video autoplay loop >
            <source src="video/blue-houses.mp4" type="video/mp4"/>
            <source src="video/blue-houses.ogg" type="video/ogg"/>
            <source src="video/blue-houses.etc" type="video/etc"/>
            Your browser does not support HTML5-video.
        </video>
    </div>
</article>

```

Videos werden in allen Normen gespeichert und das erste funktionierende wird abgespielt.

2 Canvas-Element (-> javascript gesteuerte bitmap):

```

<article>
    <div>
        <canvas width="250" height="200">
            
    </div>
</article>

```

Diese Bilder werden später durch JavaScripts gesteuert.

3 Slideshow:

Variablenzusammenstellung:

Var articles = articles, a, l, slideshow;

Es bedeuten: articles -> Bildobjekte /
a -> laufender index der articles /
l -> Anzahl der articles /
slideshow -> Abspeicherort der Slideshow

Ansteuerung des JavaScripts aus dem body der html-Seite:

```
<script type="text/javascript" src="script.js"></script>
```

Script:

```
var articles = document.querySelectorAll("#overview > article"), a = 0, l = articles.length, slideshow;
```

```
slideshow = setInterval(function() {  
    for (i = 0; i < l; i++) {  
        articles[i].style.opacity = 1;  
        if (i == a) {  
            } else {  
                articles[i].style.opacity = 0;  
            }  
        }  
        a++;  
        if (a == l)  
            a = 0;  
    }, 4000);
```

Beginn Schleife zur Bildwahl
Bild i erhält Undurchsichtigkeit = 1
wenn Schleife auf 1.bzw. weiterem Bild ist
Bild i erhält Undurchsichtigkeit = 0
Bildnummer wird um 1 erhöht
Wenn Bildnummer l (length) ist
wird Bildnummer auf 0 gesetzt
Ende Schleife / Verzögerung 4 Sek

Mögliche anschließende Begrenzung der Anzeigedauer auf 5 Minuten:

```
setTimeout (function() {  
    clearInterval(slideshow);  
}, 5*60*1000);
```

Ersatz vom oben gelb markierten Teil ist durch einen ternären Operator möglich:

```
articles[i].style.opacity = i==a ? 1 : 0;
```

Dazu muss im CSS3 irgendwo die Transition (Übergang) der articles eingerichtet sein:

```
.overview article {  
    position: absolute  
    top: 0;  
    left:0;  
    width: 100%;  
    height: inherit;  
    font-size: 13px;
```

```

    -moz-transition: opacity 0.5s; // Fade-Übergang 0,5 Sekunden
    -o-transition: opacity 0.5s;
    -webkit-transition: opacity 0.5s;
    transition: opacity 0.5s;
}

```

Die grün markierten Zeilen sind speziell für ältere Browser.

4 Selektoren austauschen:

Noch zum Bearbeiten!

5 Video API:

Noch zum Bearbeiten!

6 Geolocation API & LocalStorage:

Noch zum Bearbeiten!

Beispiele von HTML5 spezifischen Elementen im CSS

Im vierten Teil werden nun HTML5-spezifische Elemente für Video, Geolocalisation, etc. hinzugefügt:

1 Runde Ecken:

funktioniert nur bei HTML5-Seiten! Normalerweise wird nur oberer Teil eine Box abgerundet.

```

.border-top-radius {
    -webkit-border-top-left-radius: 6px;
    -moz-border-radius top-left: 6px;;
    border-top-left-radius: 6px;
    -webkit-border-top-right-radius: 6px;
    -moz-border-radius top-right: 6px;;
    border-top-right-radius: 6px;
}

```

Die grün markierten Zeilen sind speziell für ältere Browser.

2 Schatten:

Erzeugt Schatten zu einer Box

```

.box-shadow {
    box-shadow: 0 1px 2px rgba(0,0,0,25);
    -webkit-box-shadow: 0 1px 2px rgba(0,0,0,25);
    -moz-box-shadow: 0 1px 2px rgba(0,0,0,25);
}

```

Die grün markierten Zeilen sind speziell für ältere Browser.

Erzeugt Schatten zu einer Schrift

```
.text-shadow {  
    1px 1px 0 #E6E6E6;  
}
```

3 variierende Graufelder:

Erzeugt Graufelder mit linearer Veränderung

```
.gray-gradient {  
    background: #d09d9d9; text-shadow: 1px 1px 0 #E6E6E6;  
    background: -webkit-gradient(linear, left top, left bottom, from(#f2f2f2, to(#bfbfbf));  
    background: -moz-linear-gradient(top, #f2f2f2, #bfbfbf); (spez für Firefox)  
}
```

Die grün markierten Zeilen sind speziell für ältere Browser.

```
body {  
    margin: auto;  
    width: 800px;  
    background-image: -webkit-gradient(linear, 0 0, 0 100, from(#eee), to(#fff));  
    background-repeat: no-repeat;  
    font-family: ‚Molengo‘;  
    font-size: 14px;  
    color: hsl(0,0%,40%);  
}
```

4 Zielfenster für Verweise festlegen

Per Voreinstellung werden alle Verweise im aktuellen Fenster geöffnet. Mit dem Attribut `target` im einleitenden `<a>`-Tag können Sie ein Zielfenster für den Verweis festlegen.

```
<a href="http://aktuell.de.selfhtml.org/" target="_blank">SELFHTML aktuell</a>
```

Für das `target`-Attribut existieren folgende Möglichkeiten:

- `_blank`, um den Verweis in einem neuen Fenster zu öffnen,
- `_self`, um den Verweis im aktuellen Fenster zu öffnen,
- `_parent`, um bei verschachtelten Framesets das aktuelle Frameset zu sprengen,
- `_top`, um bei verschachtelten Framesets alle Framesets zu sprengen.

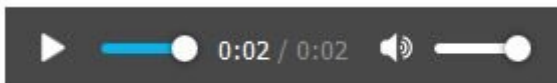
→ <http://css3please.com> Gibt Vorschläge für CSS3 Eigenschaften und einen CSS3-Generator für verschiedene Möglichkeiten (Box Radius, Box Shade, Transitions, Gradients etc.)

Noch einzufügende HTML 5 – Beispiele

- Runde Ecken
- Audio-Anschaltung automatisch
- Soundanschaltung bei Mouseover
- Video-Anschaltung
- Wechselndes Logobanner / hart
- Wechselnde Diashow weich
- Text neben Bild
- Mehrspaltigkeit
- Überlagernde Boxen
- Helle Hervorhebungsbox

audio controls per Symbol

```
<!DOCTYPE html>
<html>
<body>
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>
</body>
</html>
```



Video autoplay

Bemerkung: Geht bei Apple iPad, iPhone nicht! Da Apple in diesen Geräten keinen automatischen Download unterstützt!

```
<!DOCTYPE html>
<html>
<body>
<video width="320" height="240" autoplay>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
</body>
</html>
```


Video play 1

```
<!DOCTYPE html>
<html>
<body>
<video width="400" controls>
  <source src="mov_bbb.mp4" type="video/mp4">
  <source src="mov_bbb.ogv" type="video/ogg">
  Your browser does not support HTML5 video.
</video>
</body>
</html>
```



Video play 2

```
<!DOCTYPE html>
<html>
<body>
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogv" type="video/ogg">
  Your browser does not support the video tag.
</video>
</body>
</html>
```



Zurück-Schaltfläche

Einfügen einer „Zurück-Schaltfläche“ mit JavaScript unter body:

```
<form>
<input type="button" value="zurück" onClick="parent.history.back()">
</form>
```

Diese Methode hat den grossen Nachteil, dass sie bei CMBasic bei jeder Seitenmutation verändert wird und dann nicht mehr funktioniert. Daher besser den Retourknopf des Browser zu verwenden.

Play Audio mit Knopf (geht ohne JavaScript)

```
<!DOCTYPE html>
<html>
<body>
<button onclick="playAud()" type="button">Play Audio</button>
<button onclick="pauseAud()" type="button">Pause Audio</button><br>
<audio id="mySound" width="320" height="176">
  <source src="mov_bbb.mp3" type="audio/mp3">
  <source src="mov_bbb.ogg" type="audio/wav">
  Your browser does not support HTML5 audio.
</audio>
<script>
var aud = document.getElementById("mySound");
function playAud() { aud.play(); }
function pauseAud() { aud.pause(); }
</script>
</body>
</html>
```



Audio autoplay (geht ohne JavaScript)

Bemerkung: Geht bei Apple iPad, iPhone nicht! Da Apple in diesen Geräten keinen automatischen Download unterstützt!

```
<!DOCTYPE html>
<html>
<body>
<audio autoplay loop>
  <source src="sound.mp3" type="audio/mp3">
  <source src="sound.wav" type="audio/wav">
  Ihr Browser unterstützt das HTML5-Audio-Tag nicht!
</audio>
</body>
</html>
```

Play Audio mit Mouseover (benötigt JavaScript)

Bemerkung: Geht bei Apple iPad, iPhone nicht! Da Apple in diesen Geräten die Mouseover-Funktion nicht unterstützt!

```
<!DOCTYPE html>
<html>
<body>
<div align="center">
<audio id="mySound">
  <source src="mov_bbb.mp3" type="audio/mp3">
  <source src="mov_bbb.ogg" type="audio/wav">
  Ihr Browser unterstützt HTML5-Audio nicht!
</audio>
<script>
var aud = document.getElementById("mySound");
function playAud() { aud.play(); }
function pauseAud() { aud.pause(); }
function initso() { // Eventhandler
document.getElementById("bi").onmouseover = playAud; // startet Sound
document.getElementById("bi").onmouseout = pauseAud; // stoppt Sound
}
window.onload = initso; // startet Eventhandler nach Laden des JavaScripts
</script>
```