



HTML5

github.com/INT-WAW



META-DATEN

META-DATEN

- Gehoeren in den Header der Datei (<head>)
- Anweisungen fuer den Browser
- Anweisungen fuer Spiders oder Scraper
- Anweisungen fuer Crawler und Bots

BEISPIEL: META-DATEN

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="author" content="Christoph Martens">
  <meta name="robots" content="noindex, follow">
  <meta name="viewport" content="width=device-width">
  <meta name="apple-mobile-web-app-capable" content="yes">
</head>
```

BEISPIEL: META-DATEN (OPEN GRAPH)

- Meta-Tags auch fuer Social Web
- Open Graph API (Facebook, Twitter)

```
<head>  
  <meta name="og:image" content="/design/preview.png">  
  <meta name="og:image:width" content="256">  
  <meta name="og:image:height" content="256">  
  <meta name="og:description" content="Awesome website with some content">  
</head>
```

LINKS

- Links fuer Dokumente
- Aus Zeiten der digitalen Buecher
- Anweisungen fuer den Browser
- rel Attribut (next , prev , stylesheet)

BEISPIEL: LINKS

```
<head>
  <link rel="stylesheet" href="./design/index.css"/>
  <link rel="stylesheet" media="print" href="./design/print.css"/>

  <link rel="prev" href="page-01.html">
  <link rel="next" href="page-03.html">
</head>
```

LIVE CODING

AUFGABE

- Stylesheet `./design/index.css` hinzufuegen
- Print Stylesheet `./design/print.css` hinzufuegen
- Beschreibung und Autor der Website hinzufuegen
- Encoding auf `UTF-8` setzen
- Optional: Setze korrektes `viewport` meta-Tag (Mobile-ready)

INHALTE

BILDER

- `` Tag
- `src` Attribut mit URL Wert.
- `alt` Alternativer Text (nicht ladbares Bild)
- `width height` jeweils mit Pixelwert

```

```

```

```

HYPERLINKS

- Verknuepfung von Dokumenten und/oder Elementen
- Verweis via URL und Identifier (#blub)
- Absolute URL (/blub.html)
- Relative URL (../blub.html)
- Download einer Datei (Anhang oder Text)
- `protocol://server:port/path/file.ext#element`

BEISPIEL: HYPERLINKS

```
<a href="https://github.com">Click here to visit GitHub</a>  
<a href="https://google.com">Click here to visit Google</a>  
<a href="./index.html#seite2">Go to page 2</a>  
<a href="//google.com">Visit google (use current protocol)</a>
```

BEISPIEL: PROTOKOLLE

```
<a href="mailto:support@github.com">E-Mail to GitHub</a>  
<a href="ftp://ftp.hs-mannheim.de/setup.zip">Download via FTP</a>  
<a href="./setup.zip">Download via HTTP or HTTPS</a>  
<a href="file:///home/cookiengineer/index.html">Local File</a>
```

BEISPIEL: ELEMENTE

```
<a href="#paragraph1">First paragraph</a>  
<a href="#paragraph2">Second paragraph</a>  
  
<p id="paragraph1">A long, long time ago ...</p>  
<p id="paragraph2">there was a little script kiddo.</p>
```

HYPERLINK BEZIEHUNGEN

- Unterschiedliche Typen von Links
- Angabe via `rel` Attribut
- `author`, `help`, `license`, `next`, `prev`
- `prefetch`, `nofollow`, `noreferrer`

HYPERLINK BEZIEHUNGEN

```
<a rel="download" href="setup.zip">Download</a>  
<a rel="nofollow" href="./login.html">Login</a>  
<a rel="license" href="/LICENSE.txt">License</a>  
<a rel="noreferrer" href="https://google.com">Google (external Site)</a>
```

LIVE CODING

AUFGABE

- Ein menu , header , main und zwei section
- Eine section mit id status
- Eine section mit id tasks
- Eine table mit id status-overview
- Verlinke das Menu mit den IDs der Sektionen

DOM

DOM

- Document Object Model
- Trennung Inhalt, Design und Verhalten
- HTML5 CSS3, JavaScript / ES
- Bereitstellung DOM APIs fuer JS
- Bereitstellung DOM APIs fuer CSS

DISPLAY MODEL

- **Alle** Elemente haben ein Display Model
- Häufigst verwendete Models sind:
- Block, Inline-Block, Inline
- Neuerdings auch Flex bzw. Flexbox

BEISPIEL: DISPLAY MODEL

```
<div>Dieses Element hat einen folgenden Absatz</div>  
<p>Dieses Element ebenfalls</p>
```

```
<span>Diese</span>  
<span>Elemente</span>  
<span>nicht!</span>
```

DISPLAY MODEL

- Beeinflussung des Display Models via CSS
- CSS Angabe **ehemals** via `style` Attribut
- CSS Angabe **sauberer** via Style Sheet
- CSS Syntax: `display: block`

```
<head>
  <style>
    span {
      display: block;
    }
    span.old {
      display: inline;
      text-decoration: line-through;
    }
  </style>
</head>
<body>
<div>Dieses Element hat einen folgenden Absatz</div>
<p>Dieses Element ebenfalls</p>

<span style="background:red;color:blue;">Diese</span>
<span>Elemente</span>
<span><span class="old">nicht!</span> jetzt auch!</span>
</body>
```


CSS

CSS

- Cascading Style Sheet
- Aktueller Standard: CSS3 / CSS4
- Selektoren fuer Elemente
- Mehrere Elemente: Klassen (`.class`)
- Einzelne Elemente: IDs (`#id`)
- Direkte Strukturen: Child (`html > body`)
- Verhalten: Pseudoklassen (`a:hover` , `a:visited`)

CSS3

- Optische Gestaltung im Browser
- Statisches Design und Animiertes Design
- Textuelle Inhalte (Schriftart, -groesse, -stil)
- Zeichenabstand Zeilenabstand und Einrueckung
- Textfarbe Hintergrundfarbe, Rahmen
- Transforms (Positionierung in 2D und 3D Raum)
- Transitions (Uebergaenge von - bis)
- Animationen (wiederholte Uebergaenge von - bis)

CSS3

- CSS enthaelt CSS Regeln (Rules)
- Eine Regel enthaelt mehrere Statements
- CSS 2.1+ Lexer mergen nach gleichwertigen Selektoren

AUFBAU CSS REGEL

- Selektor legt Geltungsbereich fest
- Deklarationsblock bestimmt Darstellung des Geltungsbereichs
- Deklarationsblock besteht aus (n) Eigenschaft : Wert
- Statements werden durch Semikolon (;) getrennt

AUFBAU CSS REGEL

```
Selektor {  
  Eigenschaft: Wert;  
  Eigenschaft: Wert;  
}
```

BEISPIEL: CSS REGEL

```
h1 { font-size: 12px; }  
h1 { background: red; }
```

```
h1 {  
  font-size: 12px;  
  background: red;  
}
```

SELEKTOREN

- Unterschiedliche Selektorarten
- Unterschiedliche Wertigkeit
- Vererbung der Wertigkeit in Hierarchie

```
#foo .bar > div { color: red; }  
#foo .bar div {   color: blue; }
```


SELEKTOREN

- Inline-Styles (`style="color:red"`)
- Wertigkeit 1000

```
<div style="color:red">Red Text</div>
```

SELEKTOREN

- Identifier (`id="foo"`)
- Wertigkeit 100

```
<style>  
#foo { color: red; }  
</style>
```

```
<div id="foo">Red Text</div>
```

SELEKTOREN

- Klassen und Pseudo-Klassen
- Klassen sind (n) Arten der Elemente
- Pseudo-Klassen sind Zustände der Elemente
- Wertigkeit 10

```
<style>  
.foo    { color: red; }  
a:hover { color: purple; }  
</style>
```

```
<div class="foo">Red Text</div>  
<a href="#">Blue Text on Mouse Hover</a>
```

SELEKTOREN

- Attribute
- Wertigkeit 10

```
<style>
div      { color: blue; }
div[id]  { color: red;  }
</style>

<div>Blue Text</div>
<div id="foo">Red Text</div>
```

SELEKTOREN

- Elemente und Pseudo-Elemente
- Pseudo-Elemente sind innerhalb der Elemente
- Wertigkeit 1

```
<style>
div { color: red; }
div:before, div:after {
  display: inline-block;
  content: "woop";
  color: blue;
}
</style>

<div>Text</div>
```

SELEKTOREN

- Universalselektor
- Wertigkeit 1

```
<style>  
* { background: red; }  
</style>
```

```
<div>Foo</div>  
<p>Bar</p>
```

SELEKTOREN

- $a \sim b$ Jedes b, was ein vorhergehendes a hat
- $a + b$ Jedes b, was direkt nach einem a folgt
- $a > b$ Jedes b, was ein direktes Kindelement von a ist

Weitere Selektoren auf [CSS/Reference](#) im MDN.

BEISPIEL: SELEKTOREN

```
html body {  
  background: #222222;  
  color: #ffffff;  
  font-size: 32px;  
}  
  
h1 {  
  font-size: 2em;  
}  
  
#headline {  
  background: red;  
}  
  
.stuff {  
  border: 1px solid red;  
}
```


BOX MODEL

WIDTH, HEIGHT UND Z-INDEX

- **Alle** Elemente haben `width` und `height`
- Default-Wert ist `inherit` bzw. `auto`
- Einheiten fuer Dimensionen sind `em` , `px` , `pt`

BEISPIEL: WIDTH, HEIGHT

```
<style>
div {
  width: 30%;
  height: 200px;
  background: red;
}
</style>

<div>Foo</div>
```

POSITION

- **Alle** Elemente sind statisch by default
- Absolute Inhalte mit `position: absolute`
- Fixierte Inhalte mit `position: fixed`
- Statische Inhalte mit `position: static`
- Relative Inhalte mit `position: relative`

ABSOLUTE POSITION

Wenn position `absolute` or `fixed` dann haben Elemente eine Position absolut zur linken oberen Ecke des Bildschirms und eine absolute Breite und Hoehe.

BEISPIEL: ABSOLUTE POSITION

```
<style>
div {
  position: absolute;
  width: 100px;
  height: 100px;
  top: 100px;
  left: 100px;
  right: auto;
  bottom: auto;
  background: red;
}
</style>
<div>Example</div>
```

RELATIVE POSITION

Wenn position `static` or `relative` dann haben Elemente eine Position `relative` zu ihrem umgebenden Element. Der Uebergang von `static` zu `absolute` or `fixed` ist **NUR** mit `relative` moeglich.

BEISPIEL: STATIC UND RELATIVE

```
<style>
div {
  margin: 50px;
  padding: 50px;
  border: 1px solid black;
}
div > div {
  position: relative;
  background: rgba(255,0,0,0.5);
}
div > div > p {
  position: absolute;
  top: 0px;
  left: 0px;
  background: red;
}
</style>
<div>
  I'm static
  <div>
    I'm relative
    <p>I'm absolute</p>
  </div>
</div>
```


BOX MODEL: ABSOLUTE UND FIXED

- Angabe `width` und `height` mit `px` , `%` möglich
- Absolut zur linken oberen Ecke des Bildschirms
- Absolut zum eigenen Element ohne Vererbung
- Position **NUR** noch mit `top` , `right` , `bottom` , `left`
- `z-index` zum Einfluss der Zeichenreihenfolge möglich

BEISPIEL: ABSOLUTE UND FIXED

```
<style>
  div {
    position: absolute;
    width: 100px;
    height: 100px;
  }
</style>
```

```
<div style="background:#ff0000;top:20px;left:20px;z-index:1"></div>
<div style="background:#ffff00;top:50px;left:50px;z-index:10"></div>
<div style="background:#0000ff;top:80px;left:80px;z-index:1"></div>
```

BOX MODEL: STATIC UND RELATIVE

- Angabe `width` und `height` mit `px` , `%` möglich
- Relativ zum umgebenden Element
- `z-index` hat keine Wirkung bei `static`
- `z-index` hat Einfluss bei `relative`

BEISPIEL: ABSOLUTE UND FIXED

```
<style>
  div {
    position: static;
    width: 100px;
    height: 100px;
  }
</style>
```

```
<div style="background:#ff0000;margin-top:20px;margin-left:20px;z-index:1"></div>
<div style="background:#ffff00;margin-top:-50px;margin-left:50px;z-index:10"></div>
<div style="background:#0000ff;margin-top:-30px;margin-left:80px;z-index:1"></div>
```

BOX MODEL: MARGIN UND PADDING

- **Alle** Elemente haben `margin` und `padding`
- **Alle** Elemente koennen eine `border` haben
- `margin` wirkt nur auf Position **RELATIV** zum umgebenden Element
- `margin` wirkt nicht bei `absolute` oder `fixed` Positionierung

DEMO: MARGIN UND PADDING

Live-Demo mit Inspect Element. Aufzeigen von Static, Relative, Absolute, Fixed.

LIVE CODING

AUFGABE

- Baue das `menu` mit `position:fixed` linksbündig
- Baue die `main` (Inhalt) mit `position:absolute` rechtsbündig
- Setze `overflow:auto` auf die `main` und erkläre den Unterschied
- Optional: Passe die Farben im Menu an (auch fuer `a:hover`)

TRANSITIONS

TRANSITIONS

- Keyframe Interpolation
- Uebergang (Transition) zwischen zwei Zustaenden

TRANSITION PROPERTIES

- `transition` ist Shorthand Notation fuer:
- `transition-delay` in ms oder s
- `transition-duration` in ms oder s
- `transition-property` mit `all` oder einzelne Properties
- `transition-timing-function` fuer eigene Interpolation

BEISPIEL: TRANSITION

```
<style>
div {
  width: 50px;
  height: 50px;
  background: red;
  margin-left: 0px;
  transition: 1000ms margin ease-out;
}
div.active {
  margin-left: 200px;
  transition: 2000ms margin ease-out;
  background: blue;
}
</style>
<div onclick="this.className = this.className === 'active' ? '' : 'active';">Foo</div>
```

VENDOR PREFIXES

VENDOR PREFIXES

- Nicht-Standardisiert (in Entwicklung)
- Prototypischer CSS Support
- Alter Browser ist scheisse und alt
- Alter Browser braucht Vendor Prefixes

BEISPIEL: VENDOR PREFIXES

- -moz-transform wird zu transform
- -ms-transform wird zu transform
- -o-transform wird zu transform
- -webkit-transform wird zu transform

ANIMATIONS

ANIMATIONS

- Wiederholungen von Transitions
- Keyframe Interpolation
- Angabe wie Transitions
- Keyframes mit Vendor Prefix

BEISPIEL: ANIMATIONS

```
<style>
@-webkit-keyframes move {
  from { background:red; margin-left: 0px; }
  to   { background:blue; margin-left: 200px; }
}

@keyframes move {
  from { background:red; margin-left: 0px; }
  to   { background:blue; margin-left: 200px; }
}

div {
  width: 50px;
  height: 50px;
  background: red;
  margin-left: 0px;
  animation: move 1000ms ease-out 0s infinite;
}
</style>
<div>Foo</div>
```

BEISPIEL: ANIMATIONS

```
<style>
@-webkit-keyframes move {
    0%    { background:red;   margin-left: 0px;   }
    50%   { background:blue; margin-left: 200px; }
    100%  { background:red;   margin-left: 0px;   }
}

@keyframes move {
    0%    { background:red;   margin-left: 0px;   }
    50%   { background:blue; margin-left: 200px; }
    100%  { background:red;   margin-left: 0px;   }
}

div {
    width: 50px;
    height: 50px;
    background: red;
    margin-left: 0px;
    animation: move 1000ms ease-out 0s infinite;
}
</style>
<div>Foo</div>
```