CSS: Referenz – Grundlagen

(\*) – Fortgeschrittene / Winf

# Einbinden von CSS-Definitionen

|  |  |
| --- | --- |
| <style type="text/css">  (DEFINITIONEN)  </style> | Stildefinition im head-Bereich einer HTML-Seite |
| <link rel="stylesheet" type="text/css" href="stylesheet.css"> | externes Stylesheet einbinden |
| <body style="background-color: blue;">  <p style="font-size: 20px; color: grey;"> | Stilangaben innerhalb von html-Befehlen |

# Kommentare

|  |  |
| --- | --- |
| **/\* Ich bin ein Kommentar \*/** | Kommentar, wird nicht berücksichtigt |

# Größenangaben

Es gibt zahlreiche Größenangaben, wir verwenden die folgenden:

|  |  |
| --- | --- |
| **px** | Pixel = Bildpunkte (Desktop-Bildschirm hat z. B. 1280 Pixel Auflösung). Problematisch für Responsivität (auf einem Bildschirm mit 3840 Pixel Auflösung ist 10px sehr klein, auf einem Handy-Bildschirm hochkant mit 320 Pixel Auflösung sind 10px recht groß). |
| **%** | Prozent des Elternelements. Häufig zur Größenangabe von Blockelementen verwendet ("Diese Box soll halb so breit sein wie die sie umgebende.") |
| **em**  (\*) | "M" (gesprochen: "em") 1em entspricht der Größe des Buchstabens "M" des Elternelements. Das wird bspw. durch den Browser/Betriebssystem definiert. Unabhängig von der Bildschirmgröße ist 1em immer so groß wie ein durchschnittliches "M" – also immer lesbar. |

# Anwendung von Klassen und ids in HTML

|  |  |
| --- | --- |
| **<p class="bla">test</p>** | Dem p-Element wird die Klasse .test zugewiesen. |
| **<div id="autor">…</div>** | Dem div-Element wird die id #autor zugewiesen. |
| **<h1 class="rot sonderklasse">…</h1>**  (\*) | Der h1 werden die beiden Klassen .rot und .sonderklasse zugewiesen |

# Stildefinitionen: HTML-Tags, Klassen, IDs; Kaskadierung

|  |  |
| --- | --- |
| **h1 {**  **color: red;**  **font-size: 2em;**  **}** | Ein **HTML-Tag** (Beispiel: **h1**) bekommt bestimmte Stilzuweisungen.  HTML-Tags sind bspw. h1, body, p, ul, li … |
| **h1, h2, h3 {**  **font-family: Arial, sans-serif;**  **}** | **Durch Komma getrennt** können wir **mehrere Tags** (Beispiel: **h1, h2, h3**), Klassen oder IDs gleichzeitig mit Stildefinitionen versehen. |
| **h1, h2, h3 {**  **font-family: Arial, sans-serif;**  **color: red;**  **}**  **h1 {**  **font-size: 18px;**  **}**  **h2 {**  **font-size: 16px;**  **}**  **h3 {**  **font-size: 13px;**  **color: black;**  **}** | **Kaskadierung**:  "spätere" Definitionen überschreiben "frühere". Im Beispiel: h1, h2, h3 erhalten die Farbe Rot, später wird das für die h3 mit der Farbe Schwarz überschrieben. (Trotzdem behält h3 die oben definierte Schriftart.) |
| **.bewertungstext {**  **border: 1px solid red;**  **padding: 2px 2px 2px 5px;**  **}** | **Klasse** (darf mehrmals auf jeder HTML-Seite vorkommen) |
| **#autor {**  **background-color: grey;**  **}** | **id** (darf nur EINMAL pro HTML-Seite vorkommen)  Interessant für Javascript-Ansteuerung und für Anchor-Links (Sprungmarken innerhalb der Seite) |
| **p.kommentar {**  **font-size: 12px;**  **}** (\*) | Ziel: alle p-Tags, denen die Klasse kommentar zugewiesen ist  <p class="kommentar">blaha</p> |
| **.kommentar p {**  **font-size: 10px;**  **}** (\*) | Ziel: alle p-Tags innerhalb eines mit Klasse kommentar ausgezeichneten Elements  <div class="kommentar"><p>bla1</p><p>bla2</p></div> |
| **#autor ul {**  **border: 1px solid blue;**  **}** (\*) | Ziel: alle ul-Tags innerhalb eines mit id autor ausgezeichneten Elements (ähnlich wie voriger Punkt) |

# Formatierung von Links

|  |  |
| --- | --- |
| a {  color: red;  text-decoration: underline;  } | Links sind rot und unterstrichen. |
| **a:hover {**  **text-decoration: none;**  **}** | Hover-Effekt: ohne Unterstreichung |

# Farben, Schriften, Ausrichtung

Die meisten Editoren verfügen über automatische Vervollständigung, wenn Sie bei der Bearbeitung einer als .css abgespeicherten Datei **Strg + Leertaste** drücken. Die folgenden Screenshots aus Notepad++:

Ein Bild, das Text enthält.

Automatisch generierte BeschreibungEin Bild, das Text enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Tastatur, Elektronik enthält.

Automatisch generierte Beschreibung

Deshalb hier nur einige Beispiele.

|  |  |
| --- | --- |
| **border; font-family;** | Kommentar, wird nicht berücksichtigt |
| **border: 1px solid blue;** | Rahmen: 1px Strichstärke, durchgezogen, blau. Alle drei Eigenschaften müssen angegeben werden, sonst klappt es nicht! |
| **background-color: white;** | Hintergrundfarbe |
| **color: #000000;** | Schriftfarbe: schwarz – Wir verwenden Namen (aqua, green, red, blue, gray, maroon, fuchsia, lime …) oder Hexadezimalwerte im Format #RRGGBB (R = rot, G = grün, B = blau). #0000FF = blau, #FF0000 = rot usw. Schwarz = #000000; weiß = #FFFFFF. |
| **font-family: Georgia, serif;** | Schriftart: Georgia. Falls nicht auf dem System verfügbar, wird eine Serifenschrift genommen.  Serifen sind die kleinen Strichlein am Ende der Buchstabenlinien, manche Schriften haben das nicht. |
| **font-weight: bold;** | fett |
| **font-size: 30px;** | Schriftgröße |
| **text-align: center;** | Textausrichtung; möglich sind left, center, right, justify (Blocksatz) |
| **text-decoration: underline;** | unterstrichen, nicht unterstrichen (none)… etc. |
| **line-height: 115%;** | Zeilenhöhe. Angabe bspw. in Prozent (der Größe des Textes), in Pixel (px) oder in em. |
| **width: 300px;** | Breite eines (Block-)Elements. |
| **list-style-type: square;**  **list-style-type: none;** | Definiert Bullet-Zeichen (Aufzählungszeichen) bei unnummerierten Listen (ul) bzw. Nummerierung bei nummerierten Listen (ol). Verfügbar sind **für ul-Listen** z.B. disc (gefüllter Kreis), circle (leerer Kreis) etc.; **für ol-Listen** z.B. decimal (1.,2.,3. ...), upper-roman (I., II., III., ...) |
| **list-style-image: url(bild.png);** | Bild als Aufzählungszeichen |
| **display: inline**  (\*) | (angewendet auf li) – Liste wird horizontal ausgerichtet |

# Box-Modell; margin, padding

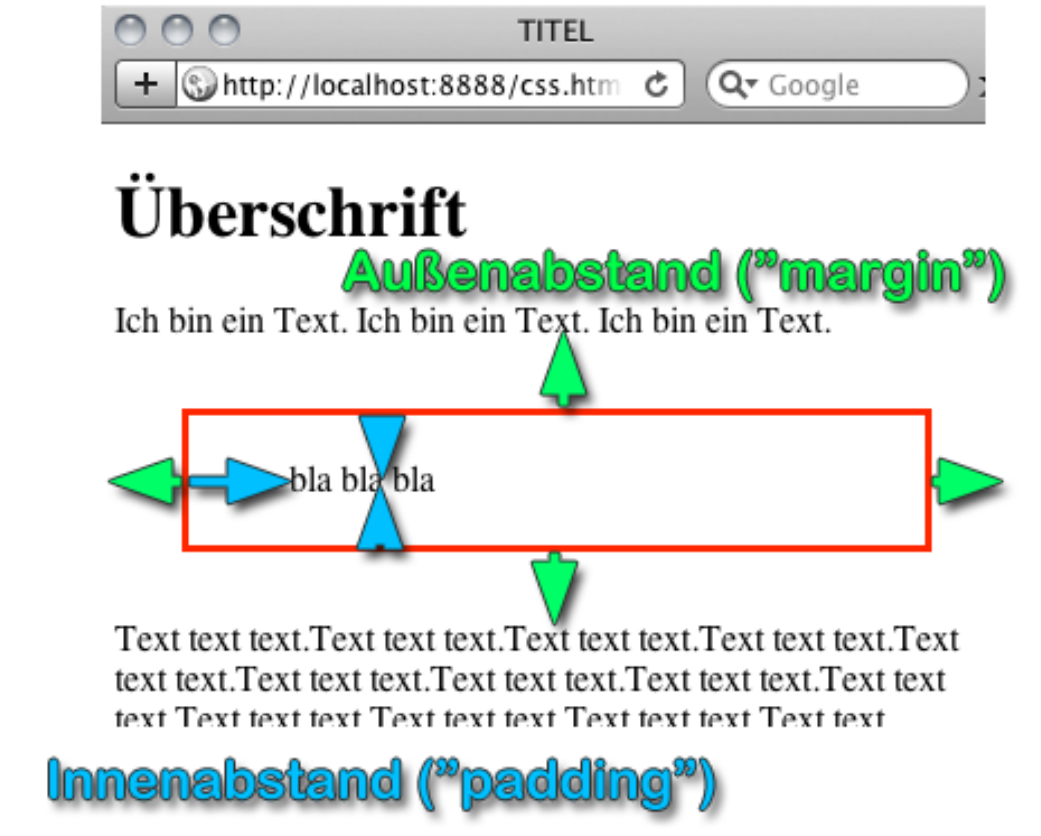
Sog. Block-Elemente wie p, h2, div, ul … haben einen Außenabstand (margin) und einen Innenabstand (padding).

**margin** (deutsch: Rand): äußerer Abstand eines Elements zu anderen Elementen.

*Tipp: Zentrieren von Blockelementen mit margin: auto (s.u.)*

**padding** (deutsch: Polsterung): innerer Abstand vom Rand der Box.

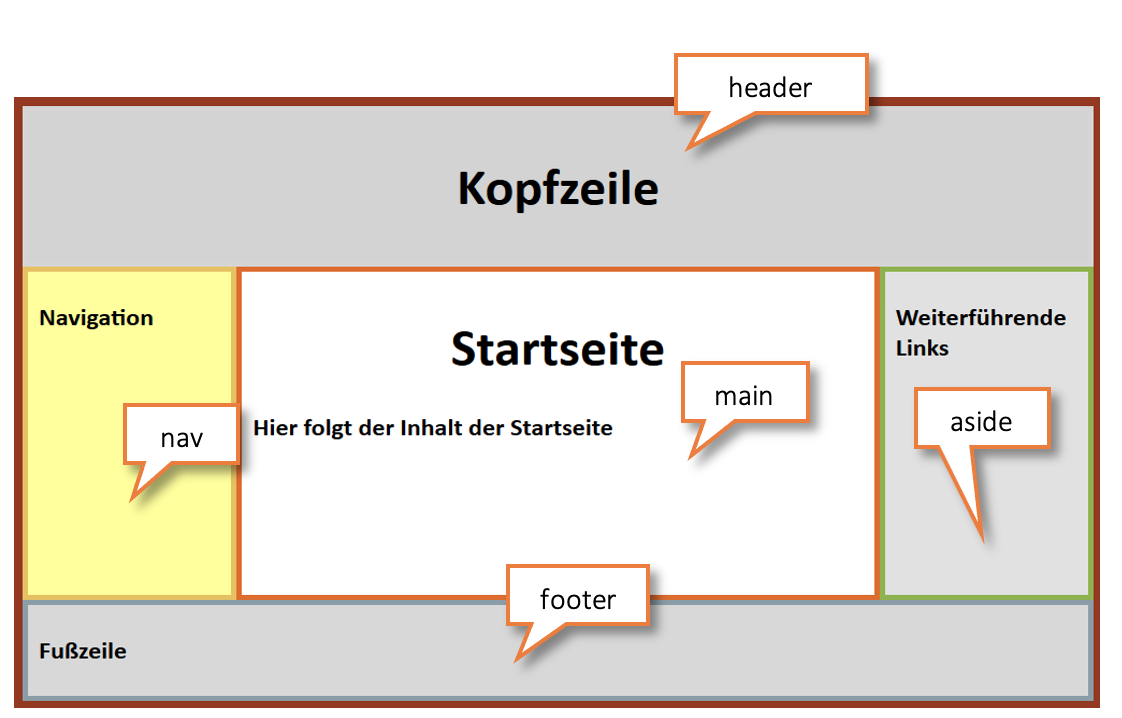
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

|  |  |
| --- | --- |
| **margin: 30px;** | Alle Außenabstände sind 30px. |
| **margin-top: 5px;** | Außenabstand nach oben ist 5px. |
| **margin: 5px 3px 5px 8px;** | Außenabstand: oben 5px, rechts 3px, … etc. (Erinnerungshilfe: Wie bei den Himmelsrichtungen: Norden, Osten …) |
| **margin: 10%;** | prozentuale Breite des umschließenden Blocks:  <div style="width: 600px">  <div style="margin: 30%">  <p>links und rechts 200px Außenabstand</p>  </div>  </div> |
| **margin: auto; /\* zum Beispiel: \*/**  **.zentriertesElement {**  **width: 50%;**  **margin: 0 auto;**  **}**  (\*) | <body>  <div class="zentriertesElement">  …  </div>  </body>  Der div-Block hat 50% Breite des umgebenden Elements (body); nach oben 0 Außenabstand, links/rechts automatisch (nämlich zentriert) |

# Grid (\*)

Wir stellen uns diese HTML-Struktur vor:



|  |  |
| --- | --- |
| **display: grid** | Wir wollen ein Grid-Layout verwenden. Falls die ganze Seite ein Grid sein soll, wenden wir diese Stildefinition auf body an. |
| **grid-template-columns: repeat(5, 1fr);** | Wir wollen 5 Spalten. Alle Spalten sind gleich grooß, nämlich einen Anteil der Breite des Grids (Anteil = **fr**action) |
| **header {**  **grid-area: h;**  **}**  **footer {**  **grid-area: f;**  **}** | Wir vergeben Namen für die einzelnen Blöcke – am besten nur einen eindeutigen Buchstaben (f = footer, n = nav etc.). |
| **grid-template-areas:**  **"h h h h h"**  **"n m m m a"**  **"n m m m a"**  **"f f f f f";** | Wir teilen das Grid auf. Wenn wir den main-Bereich schmaler und die nav breiter wollen, schreiben wir bspw. "n n m m a". |

# Responsives Verhalten / media queries (\*)

Ziel: Abhängig von der Bildschirmgröße des Geräts soll die gleiche Webseite verschieden dargestellt werden!

Mobile First: Zuerst die Darstellung für die kleinste Bildschirmgröße festlegen! Beispiel:

|  |  |
| --- | --- |
| **body**{  **display**: **grid**;  **grid-template-columns: 1fr**;  **grid-template-areas**:  **"h"  "n"  "m"  "a"  "f"; }**  **h1 {  font-size: 30px; }** |  |

Weitere Bildschirmgrößen: Hier ein Beispiel für Bildschirmgrößen ab 40 em!

|  |  |
| --- | --- |
| **@media (min-width: 40em)** **{  body{  grid-template-columns: repeat(2, 1fr);  grid-template-areas:  "h h"  "n a"  "m m"  "f f";  }  h1 {  font-size: 40px;  } }** |  |

# Schrift einbinden von Google Fonts (\*)

fonts.google.com – Fonts auswählen, dann den ausgegebenen Code einbinden, z. B.

<link rel="preconnect" href="https://fonts.gstatic.com">

<link href="https://fonts.googleapis.com/css2?family=Lobster&display=swap" rel="stylesheet">

in den head der Seite. Anwendung im CSS wie auf fonts.google.com angegeben, z. B.

font-family: 'Lobster', cursive;

**Achtung! Bei Verwendung von Google Fonts muss ein entsprechender Hinweis in die Datenschutzerklärung!**