

CSS - Cascading Style Sheets

Die [HTML-Datei](#) legt die Struktur der Seite sowie die Inhalte fest. Oft benötigt man aber genaue Angaben zum Layout. Kommt beispielsweise eine Auswahlbox vor, so möchte man festlegen, wie groß sie genau ist, ob sie einen Rand hat, wie dick dieser Rand sein soll, welche Farbe er hat usw. Solche Angaben kann man in einer sogenannten CSS-Datei speichern, auf die die HTML-Datei verweist. CSS steht für cascading style sheets, auf Deutsch etwa „mehrstufige Stilvorlagen“.

Seitenaufteilung mit DIV und CSS

Eine Internetseite besteht in der Regel aus mehreren Elementen und enthält z.B. ein Banner (die Haupttitelzeile), Menüs und verschiedene Inhaltsbereiche. Mit dem HTML-Tag

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

lässt sich die Seite strukturieren, also in ihre Elemente aufteilen. Anschließend kann man jedem Element einen Eintrag in einer separaten Layoutdatei zuordnen. Dadurch wird das genaue Aussehen des Elements (Rahmen, Farben, Abstände usw.) festgelegt.

Die Datei muster.htm stellt eine einfache Internetseite (siehe oben) dar, zu deren Strukturierung div-Tags benutzt wurden. In dieser Datei wird auf die zugehörige Layoutdatei layout.css verwiesen, in der jedem Element der Seite genaue Layoutangaben zugeordnet werden.

Näheres zum ID-Selektor: <https://wiki.selfhtml.org/wiki/CSS/Selektoren/ID-Selektor>

Beispiel

[muster.htm](#)

```
<html>
<head>
  <title>Muster zur Seitenaufteilung</title>
  <link rel="stylesheet" href="layout.css" type="text/css"/>
</head>

<body>
  <div id="seite">
    <div id="links">
      Inhalt links (z. B. Navigation)
    </div>
    <div id="rechts">
      Inhalt rechts (z. B. Texte, Bilder)
    </div>
  </div>
```

```
</body>
```

```
</html>
```

layout.css

```
body {
  background-color:#ffffff;
  color:#000000;
  font-family:Verdana, Arial;
  line-height:18px;
  font-size:13px;
  text-align:center;
}

#seite {
  text-align:left;
  width:950px;
  margin:0px auto;
}

#links {
  width:220px;
  float:left;
  margin-top:30px;
  border:1px;
  border-color:#27408B;
  border-style:solid;
}

#rechts {
  width:720px;
  float:right;
  margin-top:30px;
  border:1px;
  border-color:#27408B;
  border-style:solid;
}
```

<note>Lade beide Dateien (**muster.htm** und **layout.css**) herunter, speichere sie gemeinsam in einem Ordner und überzeuge dich davon, dass layout.css tatsächlich das Aussehen der Elemente bestimmt, indem du die Angaben (z.B. Schriftfarbe, Hintergrundfarbe, Textausrichtung) variiert.</note>

Selektoren

- Einache Selektoren: https://wiki.selfhtml.org/wiki/CSS/Selektoren/einfacher_Selector
- Element-/Typselektor: <https://wiki.selfhtml.org/wiki/CSS/Selektoren/Typselektor>
- Klassen-Selektor: <https://wiki.selfhtml.org/wiki/CSS/Selektoren/Klassenselektor>
- ID-Selektor: <https://wiki.selfhtml.org/wiki/CSS/Selektoren/ID-Selektor>

Regelsätze

In CSS sind Eigenschaften innerhalb von Regelsätzen (auch Regeln) organisiert. Ein Regelsatz besteht aus:

- Einem Selektor oder einer Gruppe von Selektoren, gefolgt von
- einem durch geschweifte Klammern begrenzten Bereich, in dem
- eine oder mehrere Deklarationen stehen, in denen Eigenschaften ein Wert zugewiesen wird.

https://wiki.selfhtml.org/wiki/CSS/Einstieg_in_CSS

Stylesheet-Klassen

Mit dem Konzept der Klassen ist es möglich, Formatdefinitionen sehr flexibel einzusetzen. Näheres zu Klassenselektoren: <https://wiki.selfhtml.org/wiki/CSS/Selektoren/Klassenselektor>
Beziehungsweise: <https://wiki.selfhtml.org/wiki/CSS/Selektoren/Universalselektor>

Beispiel

CSS

[style-blog.css](#)

```
body {  
  font-family: antiquea;  
  font-size: 14px;  
  background-color: white;  
  width: 800px;  
  margin-left: auto;  
  margin-right: auto;  
}  
  
a {  
  text-decoration: none;  
}
```

```
*.header {
background-color: #2E9AFE;
text-align: center;
}

*.block1 {
text-align: center;
width: 600px;
background-color: #A9E2F3;
padding: 10px;
margin: 10px;
float: right;
}

.selected {
font-weight: bold;
}

*.nav {
text-align: center;
width: 140px;
float: left;
background-color: #00BFFF;
margin: 10px;
}

*.nav ul {
list-style-type: none;
padding: 0;
}

*.footer {
clear: both;
background-color: #BFBFBF;
text-align: right;
}
```

HTML

[SimpleBlog_index.htm](#)

```
<!DOCTYPE html>
<html>
<head>
<title>Simple Blog</title>
<link rel="stylesheet" type="text/css" href="style-blog.css">
</head>

<body>
```

```
<div class="header">
<h1>Simple Blog</h1>
</div>
<h3><i>Hey, this is my blog!</i></h3>

<div class="nav">
<h3>Navigation</h3>
<ul>
<li><a class ="selected" href="SimpleBlog_index.html">Home</a></li>
<li><a href="SimpleBlog_about.html">About</a></li>
<li><a href="SimpleBlog_contact.html">Contact</a></li>
</ul>
</div>

<div class="block1">
<p>Blogpost #1</p>
<br>
</div>

<div class="block1">
<p>Blogpost #2</p>
</div>

<div class="block1">
<p>Blogpost #3</p>
</div>

<div class="footer">
Copyright &copy; 2017 MyName.
</div>
</body>

</html>
```

Das Box-Modell

- [Lernweg CSS Box-Modell](#)

Das Boxmodell beschreibt rechteckige Bereiche, die auf dem Bildschirm mit Hilfe von Blockelementen dargestellt werden. Die vorrangige Eigenschaft von Blockelementen ist, dass sie standardmäßig Zeilenumbrüche vor und nach sich erzwingen.

Ein Beispiel für ein Blockelement ist das div-Element (div = division = Bereich). Dieser Bereich besteht aus dem eigentlichen Inhalt, einem Innenabstand zu dem Rahmen des Elements, dem Rahmen und dem Abstand zu anderen Elementen, die auf einer Seite zu finden sind. Die abgebildete

Grafik soll diese veranschaulichen: 

Im CSS-Boxmodell besteht die Struktur einer Webseite aus mehreren sogenannten Div-Boxen, die entweder ineinander verschachtelt oder auf einer Webseite entsprechend angeordnet werden können.

Eine Div-Box hat dabei die Eigenschaften width, height, padding, border und margin.

Div-Elemente sind dazu gedacht, um Texte, Grafiken, Tabellen usw. in gemeinsamen Bereichen einzuschließen.

<https://wiki.selfhtml.org/wiki/CSS/Box-Modell> sowie

CSS #2

Dropdown-Menüs

HTML-Datei

[dropdown.htm](#)

```
<!DOCTYPE html>
<html>
<head>
  <title>Dropdown-Menu</title>
  <link rel="stylesheet" href="dropdown.css" type="text/css"/>
</head>

<body>
  <div class="dropdown">
    <button class="dropbtn">Dropdown</button>
    <div class="dropdown-content">
      <a href="#">Link 1</a>
      <a href="#">Link 2</a>
      <a href="#">Link 3</a>
    </div>
  </div>

</body>

</html>
```

CSS-Datei

[dropdown.css](#)

```
/* Dropdown Button */
.dropbtn {
  background-color: #4CAF50;
```

```
    color: white;
    padding: 16px;
    font-size: 16px;
    border: none;
    cursor: pointer;
}

/* The container <div> - needed to position the dropdown content */
.dropdown {
    position: relative;
    display: inline-block;
}

/* Dropdown Content (Hidden by Default) */
.dropdown-content {
    display: none;
    position: absolute;
    background-color: #f9f9f9;
    min-width: 160px;
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
    z-index: 1;
}

/* Links inside the dropdown */
.dropdown-content a {
    color: black;
    padding: 12px 16px;
    text-decoration: none;
    display: block;
}

/* Change color of dropdown links on hover */
.dropdown-content a:hover {background-color: #f1f1f1}

/* Show the dropdown menu on hover */
.dropdown:hover .dropdown-content {
    display: block;
}

/* Change the background color of the dropdown button when the dropdown
content is shown */
.dropdown:hover .dropbtn {
    background-color: #3e8e41;
}
```

Links

- https://www.w3schools.com/howto/howto_css_dropdown.asp
- https://wiki.selfhtml.org/wiki/CSS/Anwendung_und_Praxis/Dropdown-Men%C3%BCs_mit_CSS_gestalten

Farbgestaltung

- [100 Brilliant Color Combinations](#)
- [colorhexa.com - Webseite zu Farbkombinationen](#)
- [Colorcombos.com - Webseite zu Farbkombinationen](#)

Zweispaltiges Layout mit float

Angenommen unsere Seite ist 800px breit und wir möchten ein zweispaltiges Layout.

[zweispaltiges_layout.css](#)

```
body {
    /*Die Seite ist 800 Pixel breit.*/
    width: 800px;
}

/*Linke Spalte*/
.spalte1 {
    /*Die linke Spalte ist 380 Pixel breit. Plus Außenabstand von 10px;
    Links, rechts, oben und unten.
    Insgesamt ist die linke Spalte also 400 Pixel breit.*/
    width: 380px;
    margin: 10px;
    /*Mithilfe der Eigenschaft float kann ein Element an die linke oder
    rechte Innenkante
    seines Elternelements (Hier: body) verschoben werden.*/
    float: left;
}

/*Rechte Spalte*/
.spalte2 {
    /*Die rechte Spalte ist 380 Pixel breit. Plus Außenabstand von 10px;
    Links, rechts, oben und unten.
    Insgesamt ist die rechte Spalte also 400 Pixel breit.*/
    width: 380px;

    margin: 10px;
```

```
/*Mithilfe der Eigenschaft float kann ein Element an die linke oder rechte Innenkante seines Elternelements (Hier: body) verschoben werden.*/  
float: right;  
}
```

Links & Alternativen

- <https://wiki.selfhtml.org/wiki/CSS/Eigenschaften/Positionierung/float>
- <http://blog.karenmenezes.com/2014/apr/13/floats-inline-block-or-display-table-or-flexbox/> - Vor- und Nachteile von float und seiner Alternativen

Kommentare in CSS

```
/*Dies ist ein Kommentar.*/
```

Tabellendesign

- https://www.w3schools.com/w3css/w3css_tables.asp
- https://www.w3schools.com/bootstrap/bootstrap_tables.asp
- https://www.w3schools.com/css/css_table.asp

Beispiel

html

```
<table class="tdesign">
```

css

```
.tdesign {  
  font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;  
  border-collapse: collapse;  
  width: 100%;  
}  
  
.tdesign td, .tdesign th {  
  border: 1px solid #ddd;  
  padding: 8px;  
}  
  
.tdesign tr:nth-child(even){background-color: #f2f2f2;}
```

```
.tdesign tr:hover {background-color: #ddd;}

.tdesign th {
  padding-top: 12px;
  padding-bottom: 12px;
  text-align: left;
  background-color: #4CAF50;
  color: white;
}
```

Normalisierung

- <http://designerzone.de/css-basis-durch-reset-oder-normalisierung/>

Transparenz (opacity)

Hiermit lässt sich die Transparenz eines Elements steuern, z.B. den Hintergrund transparent gestalten.

- <https://wiki.selfhtml.org/wiki/CSS/Eigenschaften/Anzeige/opacity>

CSS-Validator

<http://jigsaw.w3.org/css-validator/>

LearningApps

[LearningApps zu CSS](#)

Weitere Links zu CSS

- Komplette Übersicht zu CSS - <https://wiki.selfhtml.org/wiki/CSS>
- Tutorial zu HTML und CSS - <https://internetingishard.com/html-and-css/>
- Farben - http://html-color-codes.info/webfarben_hexcodes/
- Learn CSS in 12 Minutes by Jake Wright - <https://www.youtube.com/watch?v=0afZj1G0BIE>
- [An Interactive Guide to CSS Grid](#) • Josh W. Comeau
- [Fun places to learn CSS Layout – Part 2: Grid Layout, by Stéphanie Walter](#)
- [grid layout generator](#)
- [Box Model Demo](#)

- [Test your skills: The box model - Learn web development | MDN](#)
- [Grid Garden - A game for learning CSS grid layout](#)
- [freeCodeCamp.org html css](#)
- [Schulbuch/Lernpfad zu HTML und CSS](#)
- [HTML und CSS - HTML-CSS-Kurs](#)
- [Box Shadow CSS Generator | CSSmatic](#)
- [Introduction to CSS - Lerne Webentwicklung | MDN](#)
- [CSS - Basics #1](#)
- [10 CSS HTML Form Designs | Sanwebe.com](#)
- [Creative Button Styles](#)
- [How SVG Line Animation Works | CSS-Tricks](#)
- [Floats, Inline Block or Display Table? Or Flexbox?](#)
- [AngularJS - multiple class ng-class - JSFiddle](#)
- [How To Create a Toggle Switch](#)
- [On/Off Flipswitch HTML5/CSS3 Generator - Proto.io](#)
- [18 Simple Styles for Horizontal Rules \(hr CSS Design\)](#)
- [color combo hex web color](#)
- [HTML, CSS and JavaScript demo - Liveweave](#)
- [The first commented line is your dabblet's title ☆ dabblet.com](#)
- [CodePen - Front End Developer Playground & Code Editor in the Browser](#)
- [JS Bin - Collaborative JavaScript Debugging](#)
- [Designing a perfect text frame in CSS - Martin Adamko - Medium](#)
- [100 Brilliant Color Combinations and How to Apply Them to Your Designs - Design School](#)

[informatik, thema, css, html]

From:

<https://herr-pfeiffer.de/unterrichtswiki/> - **Unterrichtswiki - Herr Pfeiffer**

Permanent link:

<https://herr-pfeiffer.de/unterrichtswiki/informatik:css:start?rev=1539685945>

Last update: **2019/12/16 21:53**

