CSS Merkblatt

# Zweck

Cascading Style Sheets (CSS) ist eine Stilsprache, die das Aussehen von HTML-Dokumenten definiert. CSS kann man z.B. zum Festlegen von Schriftarten, Farben, Rändern, Linien, Höhen, Breiten, Hintergrundbildern, Positionierung und viele andere Sachen benutzen.

# Links

<http://de.selfhtml.org/>

# Aufbau der CSS-Befehle

## Syntax von CSS-Befehlen

Um den Aufbau der CSS-Befehle besser verstehen zu können, wird dieser hier als Text aufgezeigt.

**CSS-Anweisung in Deutsch beschrieben:**

**BEI WEM** soll **welche EIGENSCHAFT** einen **bestimmten WERT** annehmen.

**Als Beispiel:**

Alle **<h2> Überschriften** sollen bei ihrer **Eigenschaft Schriftfarbe** den **Wert rot** annehmen.

**Als Beispiel mit mehreren Eigenschaften und Werten:**

Alles im <body> **Seitenkörper** soll die Eigenschaft Text Dekoration mit dem Wert unterstrichen annehmen und die Eigenschaft Schriftart soll mit dem Wert Arial angezeigt werden.

Die oben genutzten Farben werden nun im Quellcode wiederverwendet. Die Schreibweise im Quellcode mit Eigenschaft und Wert sieht dann wie folgt aus.

**CSS-Anweisung als Code:**

|  |
| --- |
| h2 {color: red ; }  body{  text-decoration: underline;  font-style: arial;  } |

Wichtig ist, dass die geschweiften Klammern und das Semikolon (;) nicht vergessen werden.

# CSS in HTML-Seiten einbinden

Es gibt drei Möglichkeiten die CSS-Anweisungen in eine HTML-Seite einzubinden.

1. direkt im Quellcode
2. am Anfang der HTML-Datei
3. ausgelagert in extra CSS-Datei

Wir schauen uns nur die dritte Variante an, da es für die Entwicklung die empfehlenswerteste Variante ist. Dort werden die CSS-Anweisungen in einer externen Datei hinterlegt und diese Datei wird dann in jede HTML-Seite eingebunden.

## Auslagern von CSS-Befehlen

Zum Auslagern werden 2 Dinge benötigt. Als erstes eine neue Datei mit der Endung ".css"   
und als zweites einen Verweis darauf in der HTML-Datei, die diese ausgelagerte Datei nutzen soll.

### Verweis in der HTML-Datei

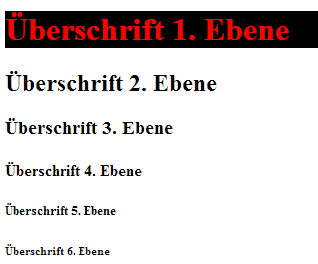
Der folgende Verweis muss im Kopfelement der HTML-Datei eingegeben werden.

|  |
| --- |
| <**link****href**="**design.css**"**type**="**text/css**"**rel**="**stylesheet**"> |

Die HTML-Datei kann dann wie folgt aussehen:

|  |
| --- |
| <!DOCTYPE HTML >  <html>  <head>  <meta charset="utf-8">  <title>Testwebseite</title>  <**link****href**="**design.css**"**type**="**text/css**"**rel**="**stylesheet**">  </head>  <body>  <h1>Überschrift 1. Ebene</h1>  <h2>Überschrift 2. Ebene</h2>  <h3>Überschrift 3. Ebene</h3>  <h4>Überschrift 4. Ebene</h4>  <h5>Überschrift 5. Ebene</h5>  <h6>Überschrift 6. Ebene</h6>  </body  </html> |

### Neue Datei erstellen

Die neue Datei kann zum Beispiel den Namen design.css haben und **muss** im selben Verzeichnis liegen wie die HTML-Datei.

Sie soll z.B. den folgenden Inhalt haben:

|  |
| --- |
| **h1** {  **color**: **red**;  **background**-**color**: **black**;  } |

Nun sollte die Überschrift 1 in roter Schrift und auf schwarzem Grund sein.

# Box-Modell bei CSS

Für jedes Element wird eine rechteckige Fläche in CSS reserviert. Der englische Begriff "Box" steht für Schachtel (in diesem Fall, da html-Seiten ja in 2 Dimensionen leben, also eine Fläche). Über das Box-Modell wird das Design einer Seite erstellt, daher ist ein Verständnis dieses Box-Modells wichtig.

## Aufbau der Box

Von Innen nach aussen:

* **Inhalt**:

für den Inhalt kann eine Breite (width) und eine Höhe (height) definiert werden

* **Innenabstand / padding**:

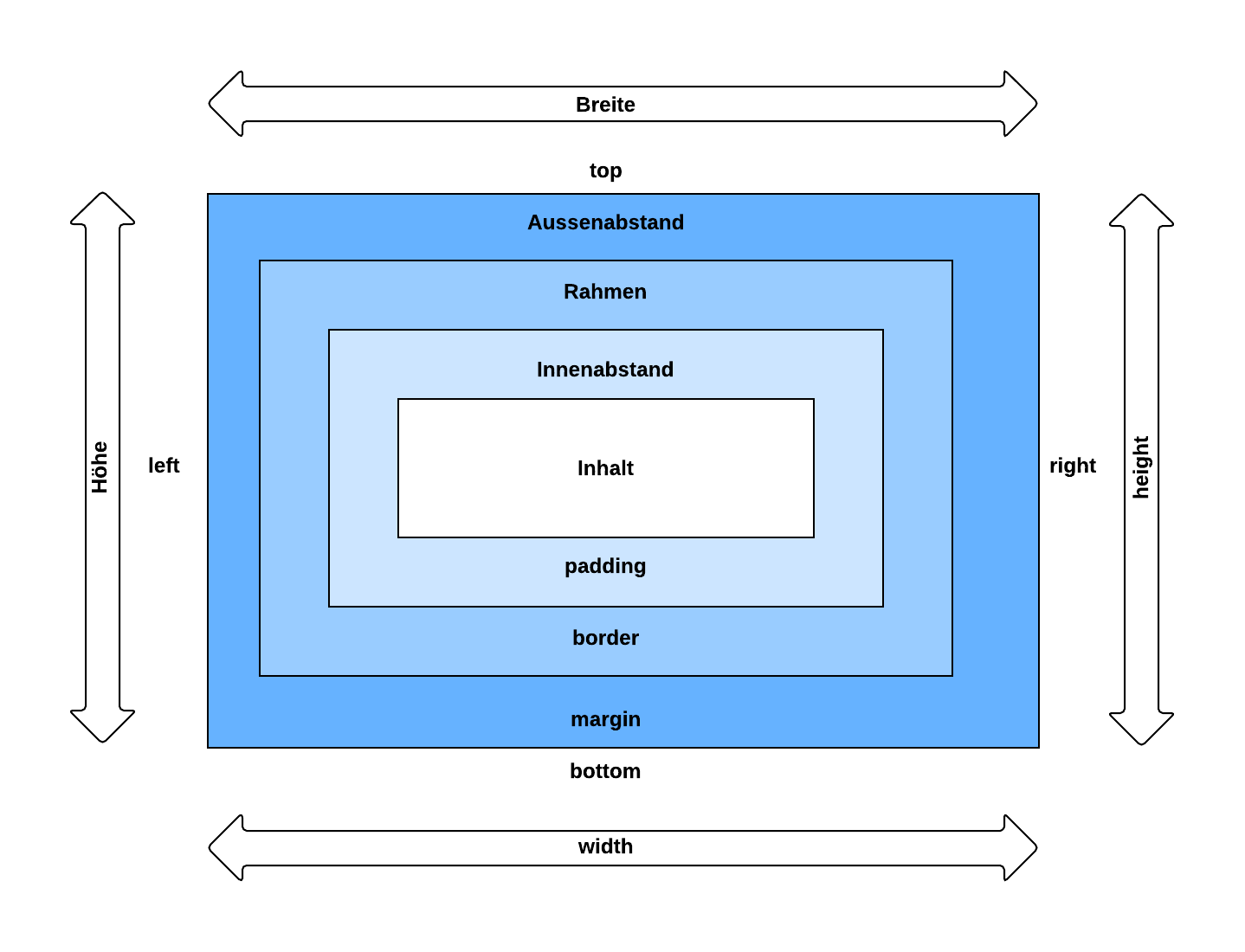
definiert den Platz zwischen Inhalt und Rahmen

* **Rahmen / border**:

dem Rahmen kann eine Stärke mitgegeben werden, die Strichart und eine Farbe   
(der Rahmen kann auch unsichtbar sein)

* **Aussenabstand / margin**:

Abstand zu anderen Elementen



## Box-Model in CSS uns HTML

Das wichtigste Element, um das Box-Modell zu realisieren, ist der Div-Container. DIV steht für das englische "division" in der Bedeutung "Bereich" und ist ein allgemeines Block-Element. Der Befehl für den Div-Container lautet **<div></div>.**

### HTML-Code

Das Div wird im Seitenkörper gesetzt.

|  |
| --- |
| <!DOCTYPE HTML >  <html>  <head>  <meta charset="utf-8">  <title>Testwebseite</title>  <link href="design.css" type="text/css" rel="stylesheet">  </head>  <body>  **<div>**  Inhalt  **</div>**  </body  </html> |

### Div sichtbar machen

Nach dem setzen der Hintergrundfarbe und der Höhe und der Breite, sieht man den Bereich des Divs und wo sich der Inhalt im Div befindet.



|  |
| --- |
| div {  **background-color: #eab2b2;**  **width: 400px;**  **height: 100px;**  } |

### Innenabstand hinzufügen

Nun wird dem Div ein padding hinzugefügt. Danach erkennt man, dass sich der Inhalt 10 Pixel vom Rand entfernt befindet.



|  |
| --- |
| div {  background-color: #eab2b2;  width: 400px;  height: 100px;  **padding: 10px;**  } |

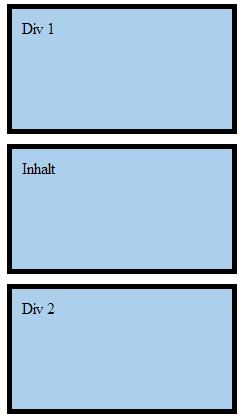
### Rahmen hinzufügen

Wenn der Rahmen dem Div hinzugefügt wurde, sieht man, dass der Rahmen dem Div angehängt wurde, somit wurde das Div grösser.

|  |
| --- |
| div {  background-color: #acd0eb;  width: 400px;  height: 100px;  padding: 10px;  **border: 5px solid;**  } |

### Aussenabstand hinzufügen

Um den Aussenabstand sichtbar zu machen, werden mehrere Divs benötigt.

**HTML:**

|  |
| --- |
| <div>Div 1 </div>  <div>Inhalt</div>  <div>Div 2 </div> |

**CSS:**

|  |
| --- |
| div {  background-color: #acd0eb;  width: 400px;  height: 100px;  padding: 10px;  border: 5px solid;  **margin: 10px;**  } |

Die linke Abbildung wurde ohne Aussenabstand generiert. Die Rahmen der Divs grenzen direkt aneinander. Die rechte Abbildung wurde mit einem Aussenabstand generiert. Zwischen den Divs kann man einen Abstand von 10 Pixel erkennen.

### Anordnung der Divs ändern

Mit dem Befehl "float" kann man die Anordnung der Divs ändern.

|  |
| --- |
| div {  background-color: #acd0eb;  width: 400px;  height: 100px;  padding: 10px;  border: 5px solid;  margin: 10px;  **float: left;**  } |

# Klassen und ID

CSS-Elemente können über diese beiden "Bezeichner" angesprochen werden. Dazu werden die HTML-TAGs um das Attribut **class="klassenname"** bzw. **id="idname"** erweitert.

Das besondere an CLASS ist, dass damit mehrere Elemente ausgezeichnet werden können, dagegen die ID nur für ein einziges Element pro Seite verwendet werden darf. ID ist also zur eindeutigen Bestimmung eines Elementes.

## Definierung in HTML

Im folgenden Beispiel wird die Verwendung von class und id anhand von Divs (<div>) gezeigt.

Wichtig ist, dass die Namen für die Klassen oder Ids nicht mit Zahlen beginnnen.

|  |
| --- |
| <div> Inhalt </div>  <div **class="divEins"**>  Div 1 - class  </div>  <div **id="divZwei"**>  Div 2 - id  </div> |

## Definierung in CSS

Klassen werden mit einem Punkt (.) vor dem Namen gekennzeichnet und IDs mit einem   
Gartenhag (#). Divs welche weder einer Klasse noch einer ID zugewiesen wurden, können über "div" angesprochen werden.

|  |
| --- |
| div {  background-color: #acd0eb;  width: 200px;  height: 100px;  }  .divEins {  background-color: #7bc471;  width: 200px;  height: 100px;  }  #divZwei {  background-color: #e6c686;  width: 200px;  height: 100px;  } |

# Hintergrundbilder

Es gibt 3 Grafikformate, die in Betracht kommen. Diese 3 Formate werden von allen Browsern unterstützt. JPEG, PNG und GIF sind die Grafikformate, die heute im Internet verwendet werden.

img = engl. **im**a**g**e = Bild  
src= engl. **s**ou**rc**e = Quelle

|  |
| --- |
| <img src="Pfad zum Bild" /> |

Wird kein Pfad zum Bild angegeben, sondern nur das Bild, so muss dieses im gleichen Verzeichnis liegen.

|  |
| --- |
| <img src="sfs\_logo.png" alt="SFS Logo" /> |

Dabei gibt es für Bilder weitere Attribute, die aus verschiedenen Gründen sehr wichtig sind.

| **Attribut** | **Bedeutung** |
| --- | --- |
| width="X" | Die Breite **X** des Bildes - sollte angegeben werden, damit der Browser den Platzhalter in entsprechender Größe vorsehen kann, solange das Bild noch nicht komplett übertragen ist! |
| height="Y" | Die Höhe des Bildes - sollte angegeben werden, damit der Browser den Platzhalter in entsprechender Größe vorsehen kann, solange das Bild noch nicht komplett übertragen ist! |
| border="0" | Ob ein Rahmen gezeichnet werden soll (wenn nicht, dann 0) und wenn er gezeichnet wird, mit welcher Stärke |
| alt="ERKLÄRUNG ZUM BILD" | Der **ALT**ernativtext - dieser ist aus verschiedenen Gründen sehr wichtig:   * für Sehbehinderte (die ihn von speziellen Browsern vorgelesen bekommen) * für Suchmaschinen, da diese auch diesen Text indizieren * zur Information für den Surfer |