
Lernziel: Die Grundzüge des CSS kennenlernen und anwenden können.

Einzelarbeit:

60 Min.

1. Erstellen Sie mit einem Editor zwei „Textdateien“. Speichern Sie eine davon mit der Endung **.html** und die Andere mit der Endung **.css** ab.
Schauen sie, dass beide Dateien im selben Ordner abgespeichert wurden.
2. In der **.html** Datei sollte die **Grundstruktur** mit folgender Zeile im Head ergänzt werden. („**name**“ können sie selbst bestimmen, sollte jedoch mit der **.css** Datei übereinstimmen):

```
<link href="name.css" type="text/css" rel="stylesheet">
```

3. Im „**body**“ der **.html** Datei erstellen Sie zwei **div** Container, in welche sie zum Beispiel „Box 1“ in die Eine & „Box 2“ in die andere schreiben.

Im **Browser** sollte die **.html** Datei nun etwa so aussehen:



4. Damit wir nun gezielt den **div** mit dem „Box 1“ Inhalt, bzw. den anderen **div** gezielt anwählen können, müssen wir beiden eine **Identität** geben (**id**).

Diese wird wie folgt (im .html) eingefügt:

```
<div id="box1">Box 1</div>
```

Fügen sie auch beim zweiten div eine „id“ ein und nennen sie diesen zum Beispiele „**box2**“.

5. Nun öffnen Sie die .css Datei, wo wir den beiden Boxen einen Style geben werden.

Um einen **div** mit seiner **id** (Identität) ansprechen zu können, müssen wir davor ein **#** setzen. Die Eigenschaften und Werte, welche wir dem **div** zuweisen wollen befinden sich in den folgenden **{...}**. Also so:

```
#box1 {Eigenschaft: Wert;}
```

Die verschiedenen Werte werden mit einem **;** beendet, sodass eine Weitere Eigenschaft mit einer Wertangabe angefügt werden kann.

```
#box1 {Eigenschaft: Wert; Eigenschaft:Wert;}
```

6. So eine **Eigenschaft** kann zum Beispiel eine **Hintergrundfarbe** sein. Zum Beispiel würde ein roter Hintergrund wie folgt erzeugt:

```
Background: red;
```

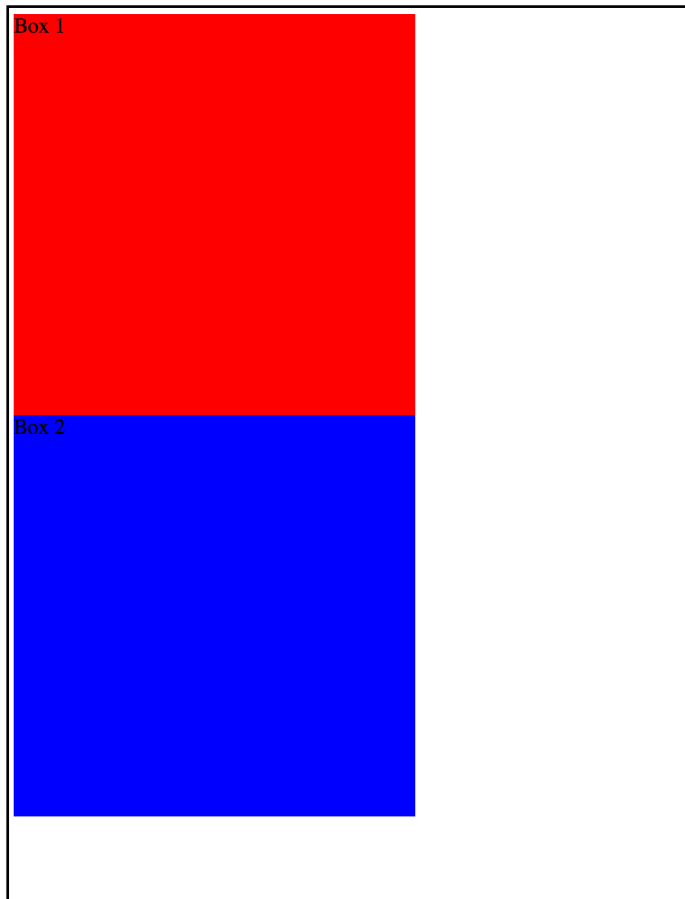
7. Schreiben Sie nun die .css Datei so, dass zum Beispiel eine der Boxen **rot** und die andere **blau** ist.

8. Fügen Sie zusätzlich noch eine **Breite** (width) und eine **Höhe** (height) jeweils den beiden Boxen zu.

Zum Beispiel so:

```
width: 300px; height: 300px;
```

9. So wenn Sie nun die Hintergrundfarbe, die Höhen und Breiten den beiden Boxen zugefügt haben, sollte ihre **.html** Datei im **Browser** etwa so aussehen:



10. Für eine bessere Übersicht schreibt man die **Eigenschaften** meist übereinander in einer **.css** Datei:

```
#box1 {  
    background: red;  
    width: 300px;  
    height: 300px;  
}
```

11. So nun stylen wir den Text.

Den **Text**, sprechen wir im css mit **font** an. Ein font (Text) kann unterschiedliche Werte haben wie **Grösse**, **Schriftart**, **Dicke** etc.. Diese werden mit einem **Leerschlag** von einander getrennt.

Zum Beispiel könnte man einen font wie folgt beschreiben:

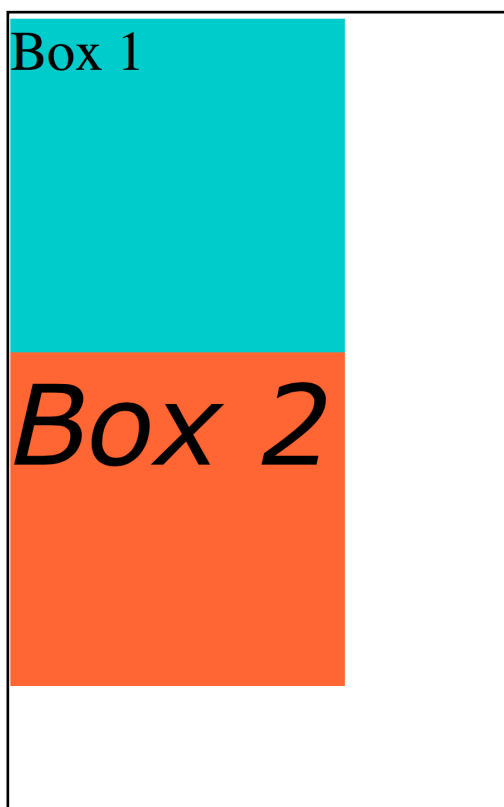
font: 14px verdana,sans-serif italic bold;

14px	steht für die Schriftgrösse.
verdana,sans-serif	ist eine Schriftart. Andere Schriftarten sind zum Beispiel: Times, Courier, etc.
italic	bedeutet kursiv, also schräg.
bold	macht die Schrift fett.

Es gibt auch noch *bolder*, *lighter* etc.

Was wir hinter **font** an **Werten** nicht schreiben, wird vom Browser im Standard Wert belassen. Man muss also nur schreiben was man benötigt.

12. Fügen sie nun die Eigenschaft **font** in ihrem **css** ein und probieren sie die **verschiedenen Werte** aus damit es im Browser etwa so aussieht:



13. Hier kommt sogleich schon die erste **Ausnahme**. Nämlich um dem **Text** (font) eine **Farbe** zu geben, bedarf es einer neuen Eigenschaft die ganz einfach nur **color** heisst.
Diese bezieht sich aber **NUR** auf den **Text** (& teilweise auf den Rahmen).

color: yellow;

oder

color: #FF0;

Probieren Sie es in ihrer css Datei aus – machen sie die Schrift bunt!

14. Eine weitere Eigenschaft sind die **Rahmen (border)**.
Ein **border** kann ebenfalls **verschieden** Werte aufweisen:

border: 4px green solid;

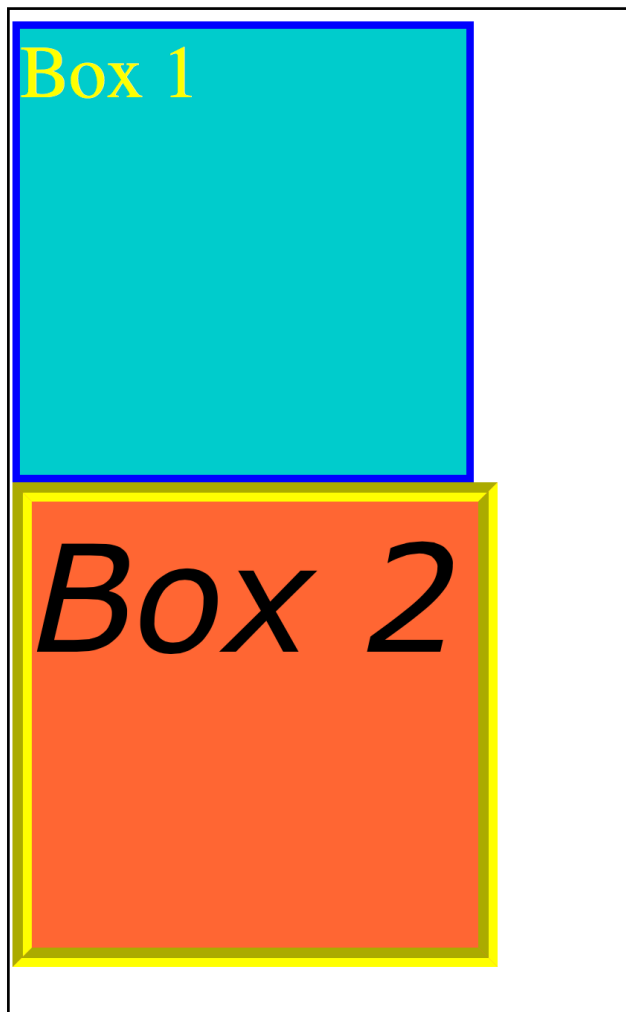
4px steht für die *Dicke* des Rahmens.

green ist die *Farbe* des Rahmen (hier nicht als Ausnahme ausserhalb der *border* Eigenschaft sondern mit dabei).
Die Farbe kann selbstverständlich auch so geschrieben werden: *#rgb*.

solid steht für die *Art* des Rahmens. *solid* bedeutet normal durchgezogen. Dann gibt es noch dotted (gepunktet), dashed, groove, double etc. probieren Sie einfach aus!

ACHTUNG: die **Art des Rahmens** MUSS angegeben werden, sonst sehen Sie nichts. Hier wird nicht wie üblich der Standardwert genommen wenn Sie nichts angeben, hier muss man mindestens **solid** (für eine normale Linie) angeben.

15. Geben Sie beiden **div** Boxen einen **Rahmen** und schauen Sie das Ergebnis im **Browser** an. Dies könnte etwa so aussehen:



16. Sehr gut! Kommen wir zu einem neuen Thema. Bis jetzt siehts so aus wie es aussieht. Aber nun wollen wir, dass sich der **Style ändert**, wenn wir mit der **Maus** drüber fahren.

Diesen Effekt nennt man **hover** (darüber schwebend).

Wenn ich nun zum Beispiel den **div Box1** verändern möchte, so muss ich eine neue **Eigenschaftengruppe** dafür schreiben, die ich dann mit **hover** beschrifte.

Ihren Code für den **ersten div** (box1) sieht momentan etwa so aus:

```
#box1 {
    background: red;
    width: 300px;
    height: 300px;
    font: 50px bold verdana;
    color: yellow;
    border: blue solid 5px;
}
```

Nun fügen Sie darunter die **hover** Gruppe ein. Dafür beginnen Sie wieder mit dem **#** und der **id**. Danach schreiben Sie einen **:** und den Befehl **hover**.

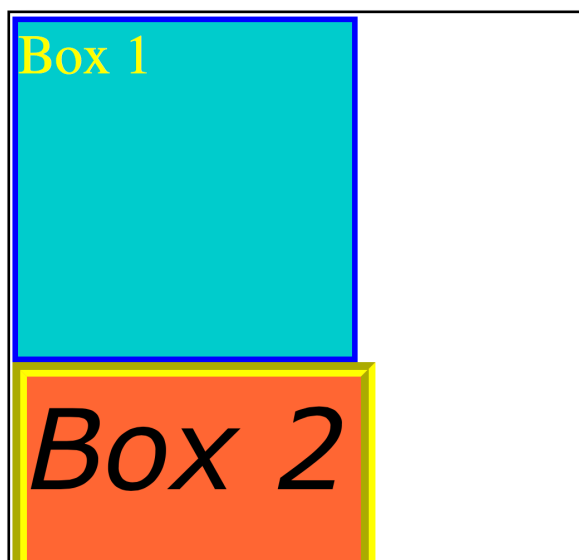
```
#box1: hover {
}
```

Alles was Sie nun in die **hover** Gruppe hinein füllen, wird, wenn die **Maus** drüber fährt, von der **normalen #box1** Gruppe **geändert**.

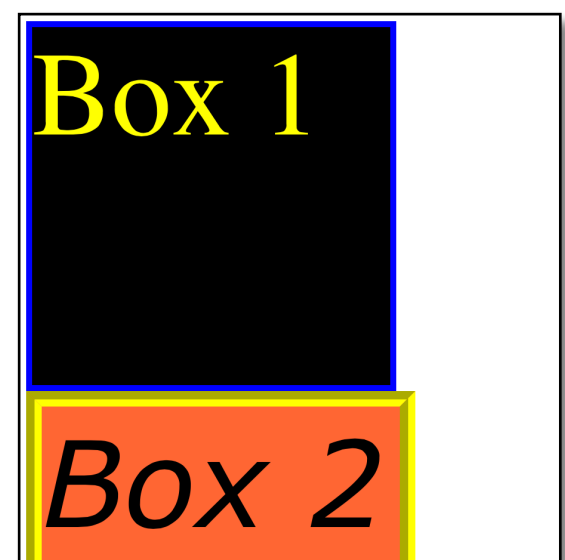
Zum Beispiel, wenn man die Schriftgrösse und die Hintergrundfarbe neu definiert (der Rest bleibt gleich, weil ja nicht verändert in der hover-Gruppe):

```
#box1: hover {
    background: black;
    font: 100px bold;
}
```

normal



hover



17. Die Wechsels sind nun ja ziemlich hart vom **normalen** Zustand zum **hover**. Um diesen Wechsel etwas **smoother** zu machen benutzen Sie als weitere Eigenschaft **transition**.

transition weist ebenfalls einige Werte auf, wir arbeiten nun aber mal mit zwei davon.

transition: *all 500ms ;*

all Hier wird definiert welche Eigenschaften (Breite, Höhe, Schrift, Rahmen etc.) es betrifft. *all* bedeutet **Alle** Eigenschaften.

500ms *MS* bedeutet *Millisekunden*. Dieser Wert ist die **Zeit** die der „Animation“ zur Verfügung steht von a nach b.

18. Die Eigenschaft **transition** benutzen sie innerhalb der **normalen Gruppe** und nicht in der **hover Gruppe**.

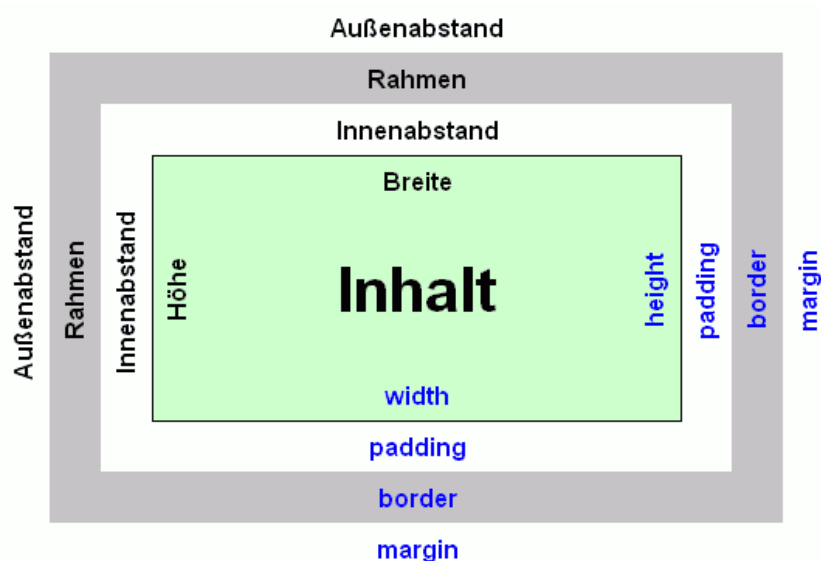
Unser Code sollte also wie folgt aussehen:

```
#box1 {  
    background: red;  
    width: 300px;  
    height: 300px;  
    font: 50px bold verdana,sans-serif;  
    color: yellow;  
    border: blue solid 5px;  
    transition: all 500ms ;  
}  
  
#box1:hover {  
    background: black;  
    font: 100px;  
}
```

Probieren Sie es **selber** aus und schauen Sie was passiert!

19. So bis anhin haben wir Farben, Grössen, Schriften etc. gestylt — nun widmen wir uns den **Abständen** von all den Dingen.
Jede div Box kann folgende zwei Arten von Abständen vorwiesen:

Aussen Also ausserhalb der Box, relativ zu anderen Boxen.
Innen In der Box selber drinnen. Also zum Beispiel wie ein Text oder andere Inhalte in der Box drin stehen.



Wie man auf der Grafik schon erkennen konnte, bedienen wir uns wiederum den englischen Begriffen, die da wären:

Aussen = **margin**
Innen = **padding**

Die beiden kann man nun als Einheitswert beschreiben, also wenn zum Beispiel der Abstand um die Box auf allen vier Seiten der selbe ist, oder unterteilt in die vier Seiten, mit vier verschiedenen Werten.

margin: 20px; Hier ist der Abstand auf allen Seiten gleich, also 20px.
margin: 4px 23px 1px 10px; Hier unterscheiden sich die vier Abstände.

Die vier Werte gelten wie folgt:

margin: linksOben rechtsOben rechtsUnten linksUnten; Also einmal im Uhrzeigersinn.

20. Benützen Sie nun **margin** und **padding** in ihrem CSS und schauen Sie was passiert.

21. Übrigens, die **vier Werte Variante** (*margin: 4px 23px 1px 10px;*), welche wir vorhin mit margin und padding kennengelernt hatten, funktioniert auch mit **border**.

So können Sie zum Beispiel jede der **Vier Seiten unterschiedlich dick** machen. Dazu aber am besten **border** mit der Zusatzeigenschaft **-size** ergänzen:

border-size: 4px 2px 10px 2px;

Oder unterschiedliche Farben oder styles. (**-color** oder **-style**).

22. Experimentieren Sie auch mal mit folgender Eigenschaft und schauen Sie was mit der div Box passiert:

border-radius: 30px;

Auch den **border-radius** können Sie mit **vier verschiedenen Radien** belegen wie es in 21. erklärt wurde.

23. Fügen Sie nun in der **.html** Datei in eine der zwei Boxen eine Liste ein. (**ol** oder **ul** was Sie wollen).

Nehmen wir mal an Sie haben eine **Liste** mit **drei Punkten** und einem **ol tag** erstellt so würde dies im **Browser** etwa wie folgt aussehen:

1. Auto
2. Baum
3. Hund

24. Nun möchten wir diese Liste etwas umstylen.

Dafür müssen wir der **Liste** aber erst eine **id** (Identität) geben wie wir es bei dem **div** schon gemacht hatten.

Im **.html** sieht das wie folgt aus:

```
<ol id="liste1">  
  </li> ....
```

Den Namen **"liste1"** können sie selber definieren nach Lust und Laune — wichtig ist nur, dass er mit der **id** im **.css** übereinstimmt.

25. Nun können wir im **.css** diese Liste mit **#liste1** ansprechen.

Innerhalb der **{...}** arbeiten wir nun mit folgender Eigenschaft:

list-style: none;

(none ist hier nur ein Beispiels Wert).

Fügen sie die letzten Schritte in ihr **.css** ein und schauen Sie was passiert.

26. Andere Werte für die Eigenschaft **list-style** sind zum Beispiel:

circle, square, armenian, decimal, decimal-leading-zero, georgian, lower-alpha, lower-greek, lower-latin, lower-roman, upper-alpha, upper-latin, upper-roman, inher etc.

Probieren Sie mal ein zwei aus und schauen Sie was passiert.

27. Wie sie im letzten Punkt (25.) sicherlich bemerkt hatten **eliminiert none** die **Aufzählungszeichen** ganz, während die anderen Werte der Liste (von 26.) das Aufzählungszeichen verändern.

mit dem **url(...);** Wert können Sie auch eigene **Bilder** als Aufzählungszeichen verwenden. Aber Achtung diese sollten **nicht zu gross** sein.

list-style: url(...);

Bei ... schreiben Sie den **Dateinamen** des Bildes hin. (Dieses sollte im selben Ordner sein).