

## 2. Eine Layout-Grundstruktur für alle Fälle



### ***Zeit sparen - ein HTML-/CSS-Grundlayout, das mit wenigen Handgriffen angepasst werden kann***

Sie müssen das Rad nicht jedes Mal neu erfinden. Eine extrem variable HTML-Grundstruktur, die nur durch Änderungen in der *Style.css* zu modifizieren ist, können Sie am Ende dieses Kapitels erstellen und dann nach Ihrem Gusto mit Inhalt und Bildern füllen.

Man kann mit ihr nicht nur – wie gezeigt – ein dreispaltiges Layout mit Spalten links und rechts erstellen, sondern auch lediglich eine Seitenspalte nutzen oder beide auf eine beliebige Seite bringen oder unterhalb des Contents anzeigen.

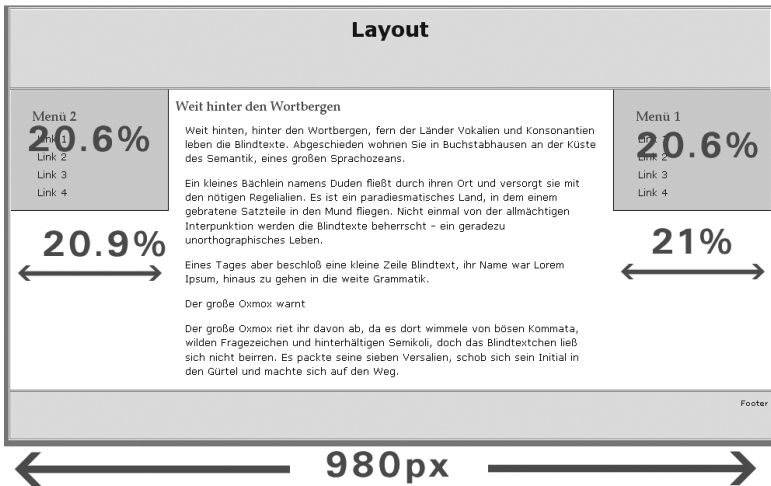
Es könnten sogar eine oder beide Spalten hoch hinauf in den sichtbaren Header reichen und so noch mehr Gestaltungsmöglichkeiten eröffnen. Und dies alles mit ganz wenigen Änderungen in der *Style.css*.

Ganz egal, wie am Schluss dieses Layouts auch aussieht: Der Inhalt wird im Quelltext immer direkt nach dem sichtbaren Header stehen, dies ist für die Suchmaschinenoptimierung einer Webseite wichtig. Die beiden Spalten stehen in HTML untereinander, und die rechte Spalte ist dabei die erste.

## 2. EINE LAYOUT-GRUNDSTRUKTUR FÜR ALLE FÄLLE

### 2.1 Ein Grundlayout mit Header, Footer und drei Spalten (fix width)

Das erste Layout, das mit der folgenden Grundstruktur erstellt wird, hat einen sichtbaren Header, zwei Spalten links und rechts, in der Mitte den Inhalt und unten den Footer.



Wie bereits im Kapiteleinleitungstext erwähnt, wird die Anordnung der Elemente jedoch eine andere sein: zuerst der sichtbare Header **#header**, dann der Inhalt **#content**; die beiden Spalten stehen untereinander – zuerst die rechte **#column01** und dann die linke **#column02**. **#footer** ist unten und kann ebenfalls mit Inhalten gefüllt werden.

#### Der (Zusammen-)Halt fehlt

Damit diese Struktur auch funktioniert, muss man diesen wenigen Boxen einen Halt geben. Sie brauchen eine weitere Box **#container**, die die einzelnen Teile zusammenhält, und auch der Inhalt **#content** muss mit einem zusätzlichen Div **#wrapper** ausgestattet werden. Das HTML – vorerst noch ohne HTML-Header – sieht dann so aus:

- `<body>`
- `<div id="container">`
- `<div id="header">`
- `</div>`

## 2.1 EIN GRUNDLAYOUT MIT HEADER, FOOTER UND DREI SPALTEN (FIX WIDTH)

```
5    <div id="wrapper">
    ■      <div id="content">
    ■        </div></div><!--content-wrapper end-->
    ■      <div id="column01">
    ■        </div>
10   <div id="column02">
    ■      </div>
    ■      <div id="footer">
    ■        </div>
    ■    </div><!--container end--></body>
```

Damit Sie den Überblick nicht verlieren, ist es ratsam, HTML-Kommentare zu verwenden, wie hier gezeigt. Die beiden Boxen **#wrapper** und **#content** müssen vor den beiden Spalten und die Box **#container** vor dem **body**-Tag geschlossen werden.

### Floats und HTML-Kommentare

Zeigt der IE einen Inhalt zweimal an, ist dies oft ein Zusammenspiel zwischen der Verwendung der Eigenschaft *float* und HTML-Kommentaren. Löschen Sie die Kommentare, und oftmals ist dann der Spuk verschwunden.

### Der Aufbau der Style.css

Zu Beginn der *Style.css* wird in kurzen Worten beschrieben, was mit ihr gestaltet wird. Dies erleichtert die Wiederverwendbarkeit sehr und spart Zeit.

- /\*
- 3-spaltiges Layout, fix width 980px, Spalten links und rechts
- /\*

### 980 px Breite wegen der Browserscrollbalken

Das Layout ist mit 980 px Breite breit genug für die Bildschirmauflösung 1.026 x 678. Macht man es breiter, besteht die Gefahr eines horizontalen Scrollbalkens, da der Browserscrollbalken Platz braucht. Nicht alle Browser berechnen diesen Platz gleich, und so zeigt die Erfahrung, dass 980 px schmal genug für die obige Bildschirmauflösung ist.

## 2. EINE LAYOUT-GRUNDSTRUKTUR FÜR ALLE FÄLLE

### Einige Standardeinstellungen werden (zurück-)gesetzt

Nach der Beschreibung des CSS beginnt das eigentliche CSS mit der Zurückstellung der *margin*- und *padding*-Werte sowie einer Schrift- und Hintergrundfarbzuweisung.

```

■ * {
■     margin: 0;
■     padding: 0;
■ }
5 html {
■     color: #fff;
■     background-color: #757575;
■ }
```

### Die erste Box bestimmt Breite und Zentrierung

Die Box mit der ID *container* gibt die Gesamtbreite vor und zentriert das Layout am Bildschirm unabhängig von der Größe der Bildschirmauflösung.

```

■ div#container {
■     width: 980px;
■     margin: 0 auto;
■     text-align: center;
5     border: 3px double #7d7e9a;
■     color: #151515;
■     background-color: #fff;
■ }
```

Um die Abwärtskompatibilität mit älteren Browsern zu gewährleisten und weil Sie vielleicht nicht immer den Doctype Strict verwenden, muss die Box auch mit *text-align:center* ausgezeichnet werden, da

nicht jeder Browser *margin auto* korrekt umsetzt. Der Rahmen von *3px* kann verändert oder weggelassen werden, er dient hier nur der Abgrenzung der Boxen in diesem Beispiel, so sie es eins zu eins übernehmen wollen.

*margin:0 auto; und text-align:center; zentrieren in allen Browsern.*

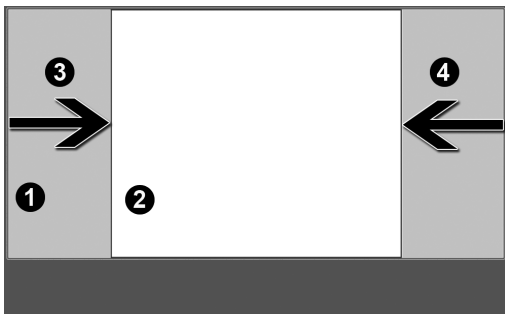
Zentrierter Text ist Schmuck und sollte genau wie dieser sehr selten eingesetzt werden, deshalb muss das *text-align:center* in der Box *#content* aufgehoben werden, da es sich sonst vererbt.

### ID wrapper stabilisiert das Layout

Die Box mit der ID *wrapper* dient der Stabilisierung und ist technisch nötig. Sie sollte **100%** breit sein, und diese Prozentangabe sollte selbst bei einem pixelgenauen Layout nicht geändert werden. Sie wird nach links gefloatet, und Erfahrungswerte zeigen, dass das Layout sehr instabil wird, wenn man das Fließen der Box nach rechts setzt. Schrift- und Hintergrundfarbe haben wiederum nur Visualisierungscharakter.

- `div#wrapper {`
- `width: 100%;`
- `float: left;`
- `background-color:#fff;`
- 5 `color:#000;`
- `}`
- 
- `div#content {`
- `margin: 0 21% 5px 20.9%;`
- 10 `text-align: left;`
- `}`

### Den Spalten Platz schaffen



*margin* gibt den Abstand von einer Box zur anderen an. Indem man die ID *content* ② links und rechts von der ID *wrapper* ① mit den *margin*-Werten (3, 4) wegdrückt, entsteht Platz für die Spalten.

Das ganze Layout ist 980 px breit, und somit werden die 21 % von den 980 px berechnet. Schlussendlich werden die Spalten genau in diesen 21%/20,9% stehen.

Verändern Sie die Breite der gesamten Seite bei der ID *container*, verändert sich automatisch die Breite der Spalten und des Inhalts.

## 2. EINE LAYOUT-GRUNDSTRUKTUR FÜR ALLE FÄLLE

### Browser berechnen Boxen mit **border** unterschiedlich

Im gezeigten Beispiel haben die beiden Spalten einen Rand auf der dem Inhalt zugewandten Seite. Mancher Browser zieht die **border**-Breite von der Breite der Box ab, andere rechnen sie dazu. Sollten Sie ein pixelgenaues Layout erstellen wollen, verzichten Sie daher am besten auf einen Seitenrand mit **border** und nutzen ein Hintergrundbild, wie auf Seite 103 erklärt, um diesen darzustellen.

- `div#column01 {`
- `float: left;`
- `width: 20.6%;`
- `margin-left: -20.8%;`
- 5 `text-align: left;`
- `border-left: 1px solid #000;`
- `border-bottom: 1px solid #000;`
- `background-color: #bdcbd0;`
- `}`
  
- `div#column02 {`
- `float: left;`
- `width: 20.6%;`
- `margin-left: -99.6%;`
- 5 `text-align: left;`
- `border-right: 1px solid #000;`
- `border-bottom: 1px solid #000;`
- `background-color: #bdcbd0;`
- `}`

Der Inhalt **#content** steht vor den beiden Spalten in HTML und innerhalb der ID **wrapper**. Damit die beiden Spalten an den Platz kommen, den die **#content** durch die *margin*-Werte bereithält, muss man den beiden Spalten die Eigenschaft **float** geben. Die negativen *margin*-Werte bringen diese dann an ihren Platz.

Der Internet Explorer 6 hat ein anderes Verständnis von Breitenangaben (siehe Seite 27), daher müssen Sie in die *ie6.css* (siehe Seite 48) für die beiden Spalten andere Werte eintragen:

- `div#column01 {`
- `float: left;`

## 2.1 EIN GRUNDLAYOUT MIT HEADER, FOOTER UND DREI SPALTEN (FIX WIDTH)

- width: 20.6%;
- **margin-left: -20%;**
- 5 text-align: left;
- border-left: 1px solid #000;
- border-bottom: 1px solid #000;
- background-color: #bdcbd0;
- }
  
- div#column02 {
- float: left;
- width: 20.6%;
- **margin-left: -99%;**
- 5 text-align: left;
- border-right: 1px solid #000;
- border-bottom: 1px solid #000;
- background-color: #bdcbd0;
- }

### Breiten ändern und die Folgen

Sie können alle diese Prozentwerte ändern, und Sie können auch pixelgenaue Werte dafür verwenden. Sollten Sie je einen Wert zu breit angeben, ist dies sofort sichtbar, weil Sie dann die Spalten nämlich nicht mehr sehen. Sie rutschen unter den Inhalt und verschwinden irgendwo in der linken Seite. Sollten Sie also wirklich Zeit sparen wollen, ändern Sie diese Werte erst dann, wenn Sie sehr sicher in der Handhabung von CSS sind.

### Header und Footer

Der sichtbare Header *#header* und der Footer *#footer* befinden sich innerhalb der ID *container*, und beide benötigen eine Breitenangabe. Die beiden Spalten haben eine *float*-Auszeichnung, und diese muss im Footer wieder aufgehoben werden, weil sonst das Layout auseinanderfließt. Alle anderen Auszeichnungen sind vom Design abhängig.

- div#header {
- **width: 100%;**
- height: 100px;
- margin: 0;