



Frank L. Schad

Das CSS3 Lernbuch

Ein Webmasters Press Lernbuch

Version 4.2.2 vom 15.11.2016

Autorisiertes Curriculum für das Webmasters Europe Ausbildungs- und Zertifizierungsprogramm.

www.webmasters-europe.org

Inhaltsverzeichnis

Vorwort	11
1 Einführung	12
2 CSS verwenden	14
2.1 Einbinden von CSS-Vorgaben in den Dateikopf	14
2.2 Allgemeine Syntax	16
2.3 Maßeinheiten	18
2.4 Kommentare	19
2.5 Browserspezifische Eigenschaften mit Herstellerpräfix (Vendor Specific Prefix)	19
2.6 Fragen zur Selbstkontrolle	21
3 Einbindung von CSS und Testen von Websites	22
3.1 Auslagerung der CSS-Vorgaben in eine externe Datei	22
3.2 Inline-Formatierungen	23
3.3 Webseiten testen	25
3.3.1 Der Web Inspector	25
3.3.2 Mobilgeräte testen	26
3.4 Fragen zur Selbstkontrolle	27
4 Die Verwendung von Farben	28
4.1 Das RGB-Farbmodell	28
4.2 Hexadezimale Farbzuzuweisung	28
4.3 Dezimale Farbzuzuweisung	31
4.4 Fragen zur Selbstkontrolle	32
5 Allgemeine Vorgaben für Ihr Dokument	33
5.1 Vererbung	33
5.2 Hintergrundfarben definieren	34
5.3 Die Maßeinheiten em und rem	34
5.4 Überschriften formatieren	39
5.5 Die Darstellung von strong, b, em und i definieren	40
5.6 Weitere Eigenschaften zur Textformatierung	40
5.7 Hyperlinks formatieren	41
5.8 Fragen zur Selbstkontrolle	43
6 Pseudoklassen und Pseudoelemente	45
6.1 Pseudoklassen	45
6.1.1 Verschiedene Zustände für Hyperlinks definieren	45
6.1.2 Elemente automatisch durchzählen	47
6.1.3 Überblick Pseudoklassen	50
6.2 Pseudoelemente	51
6.2.1 Initialen und andere typographische Effekte	52

6.2.2	Pseudoelemente für automatisch erzeugten Inhalt	52
6.3	Fragen zur Selbstkontrolle	55
7	Hintergründe, Konturen und Abstände	56
7.1	Hintergrundfarben	56
7.2	Innenabstände	57
7.3	Einbinden von Hintergrundbildern	57
7.3.1	Kachelmuster steuern	58
7.3.2	Hintergrundbild positionieren	58
7.3.3	Wasserzeicheneffekt	58
7.3.4	Zusammenfassen der Eigenschaften	59
7.3.5	Größenangaben für Hintergrundbilder	59
7.4	Here Be Dragons: Farbverläufe in CSS	61
7.4.1	Lineare Farbverläufe	62
7.4.2	Radiale Farbverläufe	64
7.5	Festlegen von Konturen	66
7.5.1	Konturstärke	66
7.5.2	Randstil	67
7.5.3	Konturfarbe	67
7.5.4	Alle Kontureigenschaften auf einmal definieren	68
7.6	Außenabstände	68
7.7	Fragen zur Selbstkontrolle	69
8	Klassen	71
9	Textumfluss	74
10	Größenangaben	76
10.1	Angaben für Breite und Höhe	76
10.2	Übersatz	77
10.3	Mindest- und Maximalgrößen	79
10.4	Die Maßeinheiten vw und vh	79
10.5	Fragen zur Selbstkontrolle	80
11	IDs und einfache Positionierung	81
11.1	Der Unterschied zwischen Klassen und IDs	81
11.2	HTML-Tags für das Seitenlayout	82
11.3	Fragen zur Selbstkontrolle	86
12	Positionierung von Elementen	87
12.1	Statische Positionierung	87
12.2	Relative Positionierung	88
12.3	Absolute Positionierung	88
12.4	Fixierte Positionierung	91
12.5	Fragen zur Selbstkontrolle	92

13	Ändern der Anzeigart und bedingte Auswahlen	93
13.1	Ändern der Anzeigart von Elementen	93
13.2	Positionsbedingte Auswahlen	94
13.2.1	Descendant Selector (Nachfahren-Selektor)	94
13.2.2	Child Selector (Kind-Selektor)	96
13.2.3	Adjacent Sibling Selector (Selektor für »benachbarte Geschwister«)	97
13.2.4	General Sibling Selector (Selektor für alle nachfolgenden Geschwister)	98
13.3	Attribut-Selektoren	98
13.3.1	Einfacher Attribut-Selektor	98
13.3.2	Tag mit Attribut-Selektor	99
13.3.3	Wertabhängige Attribut-Selektoren	99
13.4	Der Stern-Selektor	100
13.5	Fragen zur Selbstkontrolle	100
14	Flexible Layouts mit Flexbox	102
14.1	Grundlagen des Flexbox-Modells	102
14.2	Ein Flex-Layout gestalten	103
14.3	Alle Flexbox-Eigenschaften im Überblick	105
14.3.1	Eigenschaften für den Container	105
14.3.2	Eigenschaften für die enthaltenen Elemente	108
14.4	Fragen zur Selbstkontrolle	109
15	Web Fonts	111
15.1	Goodbye, Arial!	111
15.1.1	Unterstützte Formate	111
15.1.2	Konvertierung	112
15.1.3	Einbettung mit CSS	114
15.2	Awesome Icons — Font Awesome	117
15.3	Fragen zur Selbstkontrolle	118
16	Internet-Explorer-Optimierung	119
16.1	Alternative CSS-Vorgaben für den IE mit Conditional Comments	119
16.2	Seitenanzeige im Standardmodus	120
16.3	HTML5	121
17	Praxisbeispiel: Buttons mit Rollover-Effekt	122
17.1	Der Klassiker: Grafiken als Buttons	122
17.1.1	Allgemeine Formatierung	123
17.1.2	Hintergrundbild und Größe definieren	123
17.1.3	Ändern der Darstellungsart	124
17.1.4	Linktext formatieren	124
17.1.5	Rollover definieren	125
17.1.6	Textschatten erzeugen	125
17.1.7	Alternative: Sprites	126
17.2	CSS-basierte Buttons ohne Grafik	127
17.2.1	Kontur und Hintergrundfarbe hinzufügen	128
17.2.2	Ecken abrunden	128
17.2.3	Schatten hinzufügen	128

17.2.4	Farbverlauf hinzufügen	129
17.2.5	Rollover gestalten	129
18	Praxisbeispiel: Wiederkehrende Layoutelemente und Textformatierung	131
18.1	Projektvorgabe	131
18.2	Allgemeine Formatierung und Layout	132
18.3	Zentrierung des Inhalts	133
18.4	Währungszeichen anhängen und Kodierungsmethode festlegen	134
18.5	Pfeile	134
18.6	Buttons gestalten	135
18.7	Textfluss um Bilder und Clearfix	135
18.8	Buttons positionieren	139
19	Praxisbeispiel: Ein komplexes Layout mit HTML5- und CSS3-Elementen	140
19.1	Projektvorgaben	140
19.2	Allgemeine Formatierung	142
19.3	Grundlayout	142
19.4	Formatierung der Hauptnavigation	144
19.4.1	Entfernen der Listenelemente	144
19.4.2	Einrückung abschalten	145
19.4.3	Horizontale Anordnung der Listenelemente	145
19.4.4	Breiten- und Höhenangaben	146
19.4.5	Konturen hinzufügen	147
19.4.6	Farbverläufe einfügen und Links formatieren	147
19.5	Formatierung der Subnavigation	148
19.6	Gestaltung des Kopfbereichs	149
19.6.1	Formatierung der Box	149
19.6.2	Formatierung der Überschrift	149
19.6.3	Logo	150
19.7	Gestaltung des Inhaltsbereichs	150
19.7.1	Textformatierung	150
19.7.2	Spaltensatz	150
19.7.3	Automatische Silbentrennung	151
19.7.4	Download-Links gestalten	153
19.7.5	Weiches Ein- und Ausblenden mit CSS-Transitions	156
19.8	Gestaltung der rechten Spalte	157
19.9	Schatten hinzufügen	158
19.10	Ecken abrunden	158
20	CSS-Farben	161
	Lösungen der Wissensfragen	167
	Index	177

19 *Praxisbeispiel: Ein komplexes Layout mit HTML5- und CSS3-Elementen*

In dieser Lektion lernen Sie

- ein anspruchsvolles Layout mit neuen CSS-Techniken aufzubauen.
- Listen zu formatieren.
- Spaltensatz zu erzeugen.
- eine automatische Silbentrennung einzustellen.
- wie Sie verhindern können, dass das *padding* der Gesamtbreite hinzugezählt wird.
- Animierte Übergänge zwischen zwei Zuständen zu erzeugen.

19.1 Projektvorgaben

In dieser Lektion werden wir ein etwas aufwendigeres 3-spaltiges Layout mit Hilfe von Listen, HTML5-Elementen und fortgeschrittenen CSS-3-Techniken erstellen.

Übung 30: Recycling Man

Unser Layout besteht aus fünf Hauptelementen: Einem großzügigen, horizontalen Kopfbereich mit Firmenname und »Logo« (Tüte), einer ebenfalls horizontalen Hauptnavigation, einer linken Spalte mit der Subnavigation, einer rechten Spalte mit Zusatzinfos sowie dem eigentlichen Inhaltsbereich in der Mitte.

Die Hauptelemente haben abgerundete Ecken. Die Schriftart für den Titel im Kopfbereich wird als Web Font eingebunden.

Die hier gestaltete Seite stellt nicht die Startseite, sondern den Downloadbereich der Website dar, deshalb ist der entsprechende Button in der Hauptnavigation hervorgehoben. Im Inhaltsbereich befinden sich drei großflächige Links zu PDF-Dateien mit Icon und beschreibendem Text. Beim Überfahren der Links mit der Maus wird zusätzlich die Dateigröße eingeblendet.

Die fertig vorbereitete HTML-Datei sowie die benötigten Grafiken finden Sie im Ordner *recycling_man*.



Abb. 19.1 Das fertige Layout

1. Öffnen Sie die Datei *index.html* und verknüpfen Sie sie mit einer neuen, leeren CSS-Datei.

Wenn Sie den HTML-Quelltext näher betrachten, sehen Sie, dass sowohl die Haupt- als auch die Subnavigation aus *ul*-Listen bestehen. Dies ist ein gewisser Standard im modernen Webdesign und entspricht auch der Logik von Navigationen: Sie sind *Auflistungen* von Links. Zusätzlich stehen diese Listen noch innerhalb eines *nav*-Elements, damit ihnen auch die Semantik einer Navigation zukommt.

Diese Navigationen verfügen über eindeutige IDs: *hauptnavi* und *linke_spalte*, um die beiden gleichartigen Elemente `<nav>` voneinander zu unterscheiden. Die anderen Bereiche — *header* (Kopfbereich), *aside* (rechte Spalte) und *main* (Haupt-Inhaltsbereich) — benötigen hier nicht zwingend eine ID. Der letzte *article* im Inhaltsbereich trägt zusätzlich die Klasse *zweispaltig*, da wir diesen später im Spaltensatz formatieren wollen.

Der Kopfbereich *header* enthält lediglich eine Überschrift und einen Hyperlink mit der ID *logo* (der in der Praxis zurück zur Startseite führen würde). Der Link ist leer, da die Logo-Grafik später mit CSS hinzugefügt werden soll.

Die Infoboxen in der rechten Spalte sind *articles* mit *h1*-Überschriften.

19.2 Allgemeine Formatierung

2. Definieren Sie wie immer zunächst die allgemeinen Vorgaben für Ihr Dokument:

Schriftart: Lucida Grande und serifenlose Alternativschriften

Schriftgröße: Standardschriftgröße

Textfarbe: R = 51, G = 51, B = 51

Vergessen Sie auch nicht die Eigenschaften für *strong*, *b*, *em* und *i*.

3. Stellen Sie sicher, dass die neuen HTML5-Elemente auch von älteren Browsern wie gewünscht als Blockelemente dargestellt werden:

```
header, main, nav, aside, article {  
    display: block;  
}
```

4. Als Hintergrundbild für den *body* wird die Grafik *bg.gif* im Ordner *media* definiert. Diese Grafik ist lediglich 2 x 4 Pixel groß und nutzt den automatischen Wiederholungseffekt von Hintergrundbildern: Sie wird zu einem Streifenmuster gekachelt.
5. Der gesamte *body*-Bereich soll 80% breit sein und im Browserfenster zentriert werden.

19.3 Grundlayout

6. Vergeben Sie zunächst allen fünf Hauptelementen eine sichtbare Kontur von 1px in der Farbe des Textes. Dieses Vorgehen ist in vielen Fällen sinnvoll und nützlich, auch wenn Ihr Design am Ende keine Konturen vorsieht: Dadurch werden die einzelnen Elemente sichtbar und können leichter positioniert werden.

Das Vergeben der Konturen kann in einem Aufwasch erfolgen:

```
header, #hauptnavi, #linke_spalte, aside, main {  
    border: 1px solid rgb(51, 51, 51);  
}
```

Der Kopfbereich und die Hauptnavigation stehen grundsätzlich bereits an der richtigen Stelle und nehmen, da sie Block-Elemente sind, auch wie gewünscht die gesamte Breite des *body* ein. Lediglich die drei Spalten für die Subnavigation, den Info- und den Inhaltsbereich müssen noch positioniert werden. Dies wollen wir diesmal mit einer *absoluten Positionierung* erreichen.

7. Vergeben Sie zunächst jeder der drei Spalten eine feste Breite. Die linke und die rechte Spalte sollen dabei gleich breit sein:


```
#linke_spalte, aside {  
    width: 20%;  
}  
  
main {  
    width: 56%;  
}
```

Die Breite des Inhaltsbereichs sollte etwas schmaler als die verbleibenden 60% sein, um einen Abstand zu den beiden äußeren Spalten zu ermöglichen.

8. Definieren Sie dann eine absolute Positionierung für die drei Spalten:

```
#linke_spalte, aside, main {  
    position: absolute;  
}
```

9. Notieren Sie nun die Positionsangaben. Alle drei Spalten sollen gleich weit von der oberen Kante entfernt sein:

```
#linke_spalte, aside, main {  
    position: absolute;  
    top: 22rem;  
}
```

Verwenden Sie als Maßeinheit *rem* und nicht *em*, um sicherzustellen, dass als Faktor auf jeden Fall die Grundschriftgröße (16px) verwendet wird, die im *html*-Selektor definiert ist.

10. Horizontal werden die Spalten über den *body* verteilt:

```
#linke_spalte {  
    left: 0;  
}  
  
aside {  
    right: 0;  
}  
  
main {  
    width: 56%;  
    left: 22%;  
}
```

Bei einer absoluten Positionierung werden die Elemente standardmäßig am Browserfenster ausgerichtet. Um zu erreichen, dass stattdessen der 80% breite *body* als Bezugspunkt benutzt wird, benötigt dieser wieder eine (im Prinzip beliebige) andere Positionierungsmethode als *static*. In unserem Fall ist hierfür die *relative Positionierung* am besten geeignet, da diese keine optischen Veränderungen bewirkt:

11. Fügen Sie dem *body*-Selektor folgende Eigenschaft hinzu:

```
body {  
    position: relative;  
    /* weitere Eigenschaften */  
}
```

Nun benutzen die drei Spalten den *body* als Bezugspunkt.

12. Fügen Sie nun noch dem Inhaltsbereich ein *padding* hinzu, damit der Text nicht direkt am Rand klebt.

```
main {  
    width: 56%;  
    left: 22%;  
    padding: 0 1rem 1rem 1rem; /* oben, rechts, unten, links */  
}
```

Die Werte für *padding* werden jedoch der Gesamtbreite hinzugezählt, wodurch der Inhaltsbereich nun in die rechte Spalte hineinragt. Dieses Verhalten können wir ändern:

Verhindern, dass das padding zur Gesamtbreite addiert wird

Mit dem neuen Attribut **box-sizing** können Sie das Verhältnis zwischen *padding*, *border* und *width* bzw. *height* steuern. Der Standardwert ist *content-box* (*border* und *padding* werden der Breite und Höhe hinzugezählt).

13. Ändern Sie diesen Wert in *border-box* (zurzeit wieder mit Herstellerpräfix):

```
main {  
    width: 56%;  
    left: 22%;  
    padding: 0 1rem 1rem 1rem;  
  
    -webkit-box-sizing: border-box;  
    -moz-box-sizing: border-box;  
    box-sizing: border-box;  
}
```

Nun hat der Inhaltsbereich wieder eine Gesamtbreite von 56%.

19.4 Formatierung der Hauptnavigation

19.4.1 Entfernen der Listenelemente

Zunächst wollen wir die Listenelemente der `` entfernen. Dazu gibt es in CSS die Eigenschaft *list-style-type*. Diese Eigenschaft kann vier Werte annehmen:

`list-style-type:` `disc` (gefüllter Kreis als Listenpunkt, Standardeinstellung)
 `circle` (Kreis mit Outline als Listenpunkt)
 `square` (rechteckiger Listenpunkt)
 `none` (kein Listenpunkt)

Daneben existiert noch die Eigenschaft `list-style-image`, mit der Sie eine beliebige (kleine) Grafik als Listenpunkt definieren können:

```
list-style-image: url(Dateipfad);
```

14. Notieren Sie für die Liste in der Hauptnavigation:

```
#hauptnavi ul {  
    list-style-type: none;  
}
```

19.4.2 Einrückung abschalten

Daneben müssen wir auch noch die standardmäßige Einrückung der Liste deaktivieren. Diese wird jedoch von den unterschiedlichen Browsern unterschiedlich interpretiert. Die einen interpretieren sie als *margin*, die anderen als *padding*.

15. Setzen Sie daher beide auf 0:

```
#hauptnavi ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
}
```

19.4.3 Horizontale Anordnung der Listenelemente

Die Hauptaufgabe bei der Navigation ist die horizontale Darstellung der Listenelemente, die ja standardmäßig untereinander stehen. Dafür gibt es mindestens vier Möglichkeiten:

1. die Eigenschaft `display:inline` für die Listenelemente
2. die Eigenschaft `float:left` für die Listenelemente
3. eine Flexbox
4. die Eigenschaften `display:table` für die `` und `display:table-cell` für die einzelnen Listenelemente

`display:inline` wäre hier die einfachste Lösung. Wie wir aber in [Lektion 17](#) gesehen haben, hat dies den Nachteil, dass die Elemente Breiten- und Höhenangaben nicht mehr akzeptieren, die wir hier jedoch benötigen.

Die 2. Variante mit `float` wird in der Praxis durchaus häufig verwendet, hat aber ebenfalls einen Nachteil: Jedem einzelnen Listenelement muss manuell eine Breite zugewiesen werden,

um auf die Gesamtbreite zu kommen. (Und zudem erzeugt *float* den bekannten Bug, dass die betreffenden Elemente aus ihrem Container »heraushängen«. Dies müssten wir durch ein *Clearfix* beheben, vgl. [Abschnitt 18.7](#).)

Die Flexbox ist die moderne, elegante Variante, wird jedoch nur von neueren Browsern unterstützt und ist zudem vergleichsweise aufwendig für eine simple Navigation.

Wir entscheiden uns deshalb für die Variante 4 mit *display:table*. Dadurch verhält sich die Liste wie eine HTML-Tabelle, und die einzelnen *li*-Elemente werden wie die Zellen dieser Tabelle nebeneinander dargestellt. Das hat den zusätzlichen Vorteil, dass sich die einzelnen Listenelemente dynamisch an die Gesamtbreite der »Tabelle« anpassen (sich also ganz ähnlich verhalten wie in einer Flexbox).

Sie benötigen dazu drei zusätzliche Eigenschaften:

- 16.** Weisen Sie der `` in der Hauptnavigation die Eigenschaft *display:table* zu. Da sie dadurch jedoch nicht mehr automatisch die Gesamtbreite einnimmt (sie ist ja nun kein Blockelement mehr), benötigt sie noch eine Breitenangabe von 100%:

```
#hauptnavi ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
    width: 100%;  
    display: table;  
}
```

- 17.** Die einzelnen Listenelemente `` sollen sich nun verhalten wie *Tabellenzellen*:

```
#hauptnavi ul li {  
    display: table-cell;  
}
```

Nun sollten die Listenelemente nebeneinander stehen und sich dynamisch über die Gesamtbreite der Navigationsleiste verteilen.

- 18.** Zentrieren Sie nun noch den Text in den einzelnen Listenelementen.

19.4.4 Breiten- und Höhenangaben

Als nächstes wollen wir die Navigation noch etwas höher machen. Da sowohl die gesamte Navigation als auch die einzelnen *li*-Elemente später Hintergrundbilder bekommen sollen, muss dies für die Elemente *nav* und *ul* gleichermaßen erfolgen:

- 19.** Notieren Sie in einem gemeinsamen Selektor:

```
#hauptnavi,  
#hauptnavi ul {  
    height: 2rem;  
}
```

19.4.5 Konturen hinzufügen

20. Als nächstes wollen wir die einzelnen Listenelemente optisch voneinander abgrenzen. Mit hellen und dunklen Konturen erzielen wir einen leichten »Prägeefferkt«:

```
#hauptnavi ul li {  
    text-align: center;  
    display: table-cell;  
    border-left: 1px solid #fff;  
    border-right: 1px solid #333;  
}
```

Die doppelten Konturen wirken zwischen den einzelnen Menüpunkten recht schick, am linken und rechten Rand der Navigation wirken sie jedoch eher störend. Wir benötigen also eine Möglichkeit, nur **die erste Kontur des ersten Listenelements** und **die letzte Kontur des letzten Listenelements** abzuschalten.

Eine Möglichkeit wären Klassen. Diese haben jedoch den Nachteil, dass sie bei einer späteren Änderung der Menüeinträge jedesmal neu angepasst werden müssten. Wesentlich eleganter sind hier die Pseudoklassen `:first-child` und `:last-child` (vgl. [Abschnitt 6.1.2](#)).

21. Notieren Sie zwei neue Selektoren für das erste und das letzte Listenelement in unserer `ul` und deaktivieren Sie die entsprechenden Konturen:

```
#hauptnavi ul li:first-child {  
    border-left-style: none;  
}  
  
#hauptnavi ul li:last-child {  
    border-right-style: none;  
}
```

19.4.6 Farbverläufe einfügen und Links formatieren

22. Fügen Sie der gesamten `nav#hauptnavi` einen linearen Farbverlauf von oben nach unten hinzu, bestehend aus den Farben `#7793a2`, `#fff`, `#7793a2` und `#9abcce`.
23. Erstellen Sie einen Rollover-Effekt für die einzelnen `li`-Elemente, der beim Überfahren mit der Maus einen Verlauf in den Farben `#e5c134`, `#fff`, `#e5c134` und `#ffe069` anzeigt.

24. Wie können Sie erreichen, dass der Menüpunkt *Downloads* mit demselben gelben Farbverlauf dauerhaft hervorgehoben wird?
25. Formatieren Sie nun noch die Links in den einzelnen Menüpunkten: Die Textfarbe entspricht der dunkelgrauen Farbe des Fließtextes, keine Unterstreichung, fett.



Wenn die Linkfarbe der Farbe des Fließtextes entsprechen soll, müssen Sie die Farbe nicht nochmals notieren. Schreiben Sie stattdessen

```
color: inherit;
```

Dadurch vererbt der Fließtext seine Farbe an die Links.

26. Wenn Sie genau hinsehen, gibt es noch einen kleinen Bug: Die Links sind nur anklickbar, wenn man genau auf den Text klickt (erkennbar am Link-Cursor). Wie lösen Sie dieses Problem?
27. Wie können Sie den Linktext vertikal in der Menüleiste zentrieren?

19.5 Formatierung der Subnavigation

28. Gestalten Sie analog zur Hauptnavigation die Subnavigation:

Navigation:

Kontur: 1px #333 durchgezogen

Liste:

Innen- und Außenabstand: 0 (um die Einrückung abzuschalten)

Listenpunkte: keine

Listenelemente:

Hintergrundfarbe: #adbdc5

Kontur oben: 1px Weiß durchgezogen

Kontur unten: 1px #333 durchgezogen

Letztes Listenelement: keine Kontur

Rollover-Effekt für die Listenelemente:

Hintergrundfarbe: #87a8b9

Links:

Innenabstand: 0.5rem oben und unten, 1rem links und rechts

Weiß, fett, keine Unterstreichung, Darstellung als Block-Elemente

19.6 Gestaltung des Kopfbereichs

19.6.1 Formatierung der Box

29. Vergeben Sie für den *header* folgende Eigenschaften:

Innenabstand: 1rem

Außenabstand: 1rem oben, 5rem unten, 0 links und rechts

Höhe: 10rem

Hintergrund: Linearer Farbverlauf von oben (weiß) nach unten (#c0cbd3)

19.6.2 Formatierung der Überschrift

30. Erstellen Sie nun einen neuen Selektor für die Überschrift **innerhalb** des Kopfbereichs:

```
header h1
```

31. Weisen Sie diesem Selektor folgende Eigenschaften zu:

Innen- und Außenabstand: 0

Schriftart: Scriptina

Schriftschnitt: nicht fett (Deaktivierung der Standardeinstellung für *h1*-Überschriften)

Textfarbe: #a00000

Schriftgröße: 6rem

Zeilenabstand: 1.5em

Textschatten: 2px von links, 2px von oben, 3px Weichzeichnung, Schwarz mit 40% Deckkraft

Schriftart einbinden

Da die verwendete Schriftart *Scriptina* nicht zu den Schriften gehört, die auf den Systemen der Anwender installiert sind, müssen wir sie als *Web Font* einbinden (vgl. [Lektion 15](#)).

32. Laden Sie sich — falls noch nicht geschehen — die Schriftart *Scriptina Pro* von der *Font-Squirrel*-Website herunter:

www.fontsquirrel.com/fonts/scriptina-pro

33. Konvertieren Sie die Schrift mit Hilfe des *@font-face Generators* in die benötigten Formate.

34. Kopieren Sie alle Schriftdateien ins *Recycling-Man*-Projekt in einen Unterordner *fonts*.

35. Binden Sie die Schriftart mit einer *@font-face*-Regel in Ihre CSS-Datei ein.

19.6.3 Logo

36. Formatieren Sie nun das Logo im Kopfbereich. Vergeben Sie für den Selektor

`a#logo`

folgende Eigenschaften:

Breite: 13rem

Höhe: 13rem

Hintergrundbild: Datei *bag.png* ohne Wiederholungseffekt, Breite des Hintergrundbilds = Breite des Elements

Absolute Positionierung, sodass das Logo rechts unten aus dem Kopfbereich herausragt, keine Unterstreichungen

37. Erzeugen Sie einen Rollover-Effekt mit Hilfe des Hintergrund-Sprites.

19.7 Gestaltung des Inhaltsbereichs

19.7.1 Textformatierung

38. Erweitern Sie den Selektor für den Inhaltsbereich um folgende Eigenschaft:

Hintergrundfarbe: Weiß mit 60% Deckkraft

39. Erstellen Sie einen neuen Selektor für alle *h1*-Überschriften **innerhalb des Inhaltsbereichs** und vergeben Sie folgende Eigenschaft:

Schriftgröße: 2rem fett

40. Erstellen Sie einen weiteren Selektor für die Absätze *p* des Inhaltsbereichs und vergeben Sie einen 1,6-fachen **Zeilenabstand**.

41. Formatieren Sie nun noch die Links im Inhaltsbereich:

Textfarbe: #a00000, Unterstreichungen nur beim Rollover

19.7.2 Spaltensatz

CSS3 ermöglicht echten Spaltensatz, d.h. Sie können ein Element — z.B. einen *article* oder einen *div*-Container — in mehrere Spalten aufteilen. Wenn der Text das Ende der ersten Spalte erreicht, fließt er automatisch in die nächste Spalte.

Dafür stehen Ihnen mehrere Eigenschaften zur Verfügung:

`column-count`: Anzahl der Spalten

`column-gap`: Breite des Stegs (Abstand zwischen den Spalten)

`column-rule`: optionale Trennlinie zwischen den Spalten, erwartet Werte für Farbe, Stärke, Stil (wie *border*)