

Google

Cascading Stylesheets

Eine vollständige Einführung in Sprache und Anwendung.

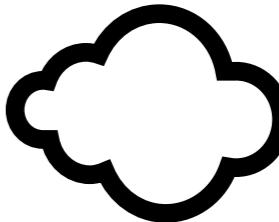
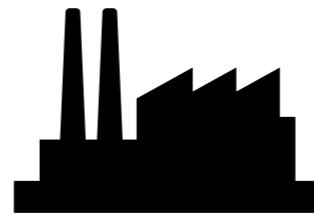
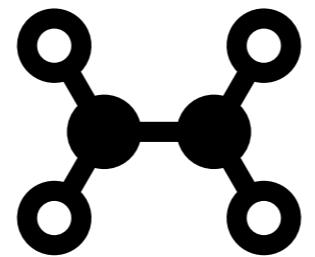
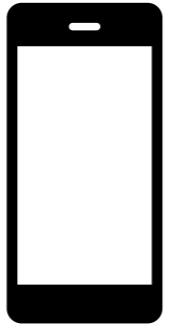
2021 Michael Reichart

CSS Grundlagen

Warum Style Sheets?

- + Stylesheets legen das Aussehen, sowie ein Teil des Verhaltens (Behaviour) eines HTML-Dokuments fest.
- + So bleibt das HTML Dokument als semantisches Struktur- und Inhaltedokument sauber und frei.
- + Völlig unabhängig vom Inhalt.

- + Aussehen und Verhalten von Information wird durch **verschiedene Stylesheets** an die jeweiligen Ausgabegerät und die jeweiligen Situationen angepasst.



CSS Syntax

Deklarationen

```
body {  
    background-color : rgb(0, 0, 200);  
}
```

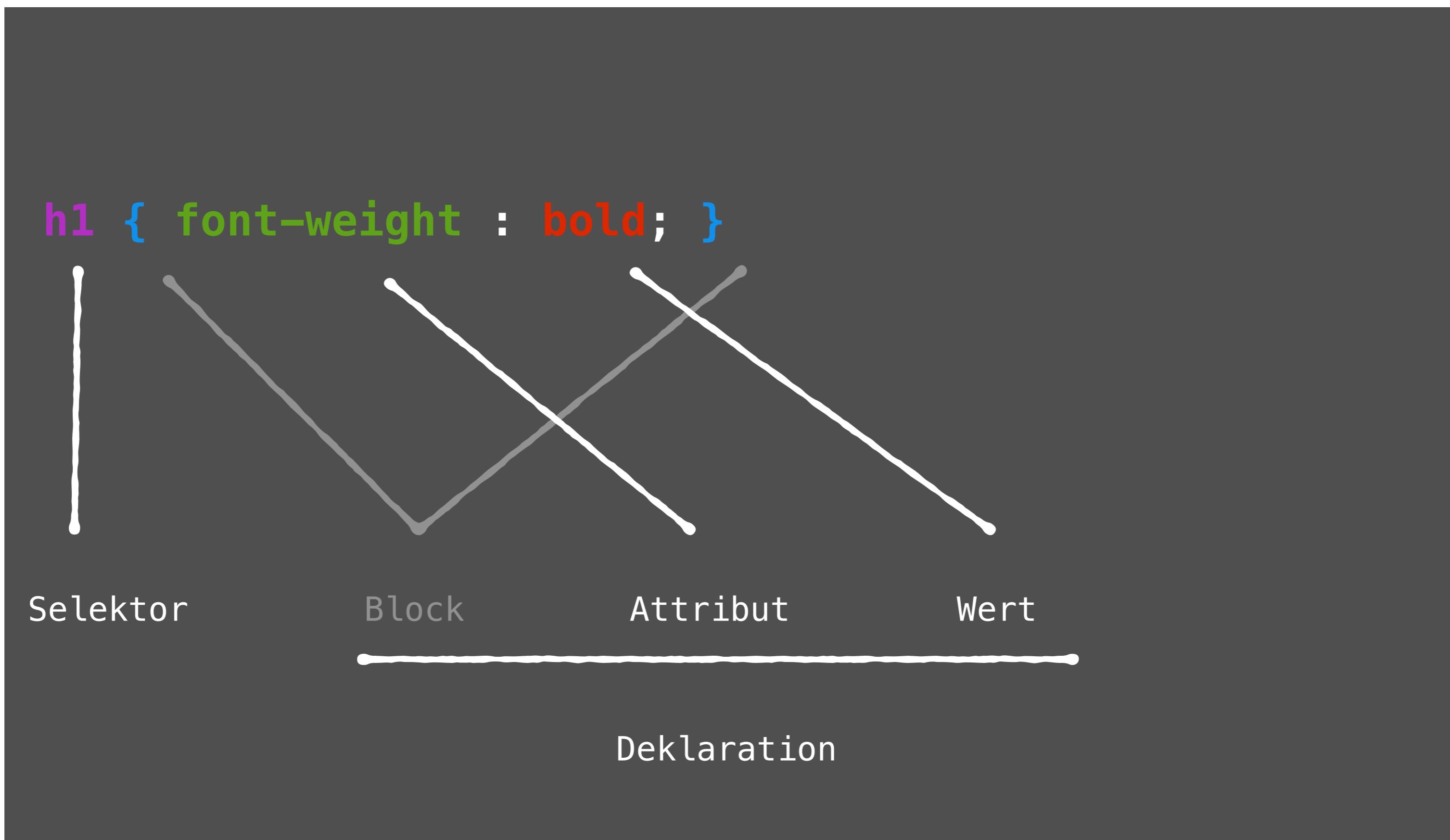
Jede beliebige Eigenschaft ist ansprechbar.

```
background-color : rgba(0,0,0,0.15);  
background-image : url(meinbild.jpg);  
background-repeat : no-repeat;  
background-attachment : fixed;  
background-position : 10px 10px;
```

Eigenschaften für Text, Schrift und Schriftverhalten (keinesfalls vollständig).

color	-> Textfarbe
direction	-> Schreibrichtung
line-height	-> Zeilenabstand
letter-spacing	-> Buchstabenabstand
unicode-bidi	-> Bidirektionalität
vertical-align	-> Ausrichtung, senkr.
white-space	-> Leerzeichenanzeige
word-spacing	-> Wortabstand
text-align	-> Textausrichtung
text-decoration	-> Textlinie
text-indent	-> Einrückung
text-shadow	-> Schatten
text-transform	-> Kapitälchen
font-family	-> Schriftart
font-size	-> Schriftgrösse
font-weight	-> Schriftgewicht

Aufbau einer Deklaration



Werteangaben, Maßsysteme & Wertebereiche

```
display: inline;  
  
font-family:  
    Arial, "Times New Roman";  
  
font-size:  
    0 | 12pt | 10pc | 13px  
    1em | 1ex | 1rem | 0.5ch  
  
width: 1in | 2.54cm | 25.4mm | 96px  
  
width: 100% | 100vh | 50 vw | 50vmin | 100vmax
```

Schriftgrößen

```
html { font-size : 16px;  
      line-height : 1.3 }  
  
h1 { font-size : 2rem; }  
h2 { font-size : 1.8 rem; }  
p { font-size : 1rem; }
```



Box-Größen

```
div.header { width: 100vw; }

div.header { width: 1024px; }

body { width: 1024px; }
div.header { width: 100%; }
```



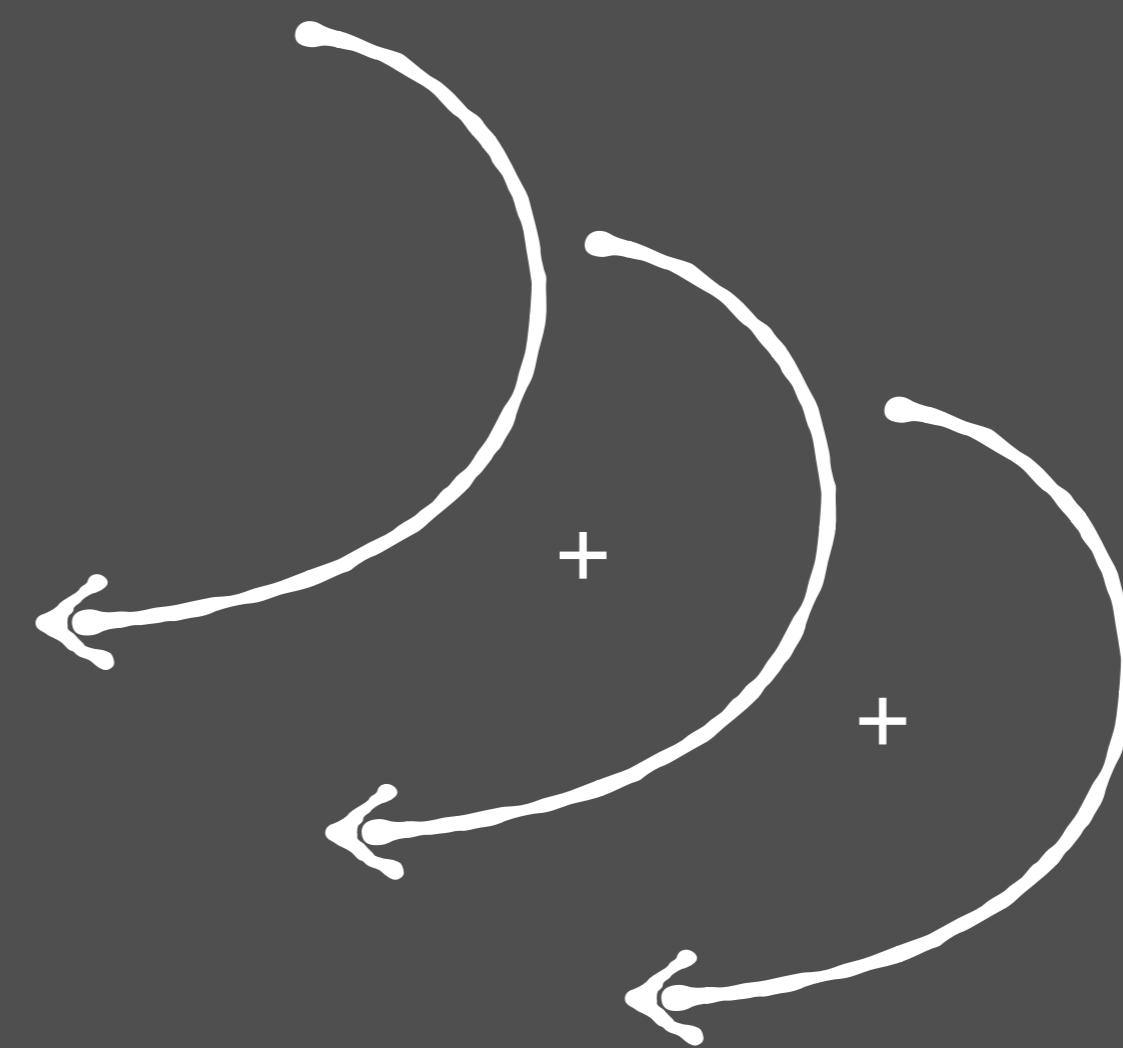
Zusammenfassen mehrerer Anweisungen

```
h1 { font-weight: bold; }          font-weight : bold;  
h1 { font-size: 12pt; }           font-size   : 12pt;  
h1 { line-height: 14pt; }         line-height : 14pt;  
h1 { font-family: Arial; }        font-family : Arial;  
h1 { font-style: normal; }        font-style  : normal;
```



Ergänzen durch Kaskadierung

```
html { font-family : Arial;  
      font-size : 16px; }  
  
h1 { font-size : 2rem; }  
  
p { font-size : 0.75rem; }  
  
<html>  
  <body>  
    <h1>...</h1>  
    <p>...</p>  
  </body>  
</html>
```



Kurzschreibweisen

```
h1 {  
    font-family: Arial, Verdana, sans-serif;  
    font-size: 12pt;  
    line-height: 14pt;  
    color: #000000;  
}  
  
h1 {  
    font: Arial 12pt/14pt #000;  
}
```

Kurzschreibweisen

```
h1 {  
    margin-top      : 1em;  
    margin-right    : 1em;  
    margin-bottom   : 1em;  
    margin-left     : 1em;  
}  
  
h1 {  
    margin : 1em; /* TRBL */  
}  
h1 {  
    margin : 1em 2em; /* TB RL */  
}  
h1 {  
    margin : 1em 0 0 2em; /* T R B L */  
}
```

Kaskadierende externe CSS-Dateien

```
@import base.css  
@import typography.css  
@import module.css  
@import skin.css
```

Einbindung in HTML

Extern: CSS-Datei via <link>

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

// mehrfach verwendbar
// Gerätespezifisch
// Möglichkeit der Modularität
// Standard!

Interner Stylecontainer

```
<style>
    h1 { font-family: Arial; }
</style>

// einfach, nicht mehrfach verwendbar
// Anwendungsspezifisch.
// Möglichkeit der punktuellen, temporären
Ergänzung.
// Einbindung durch dynamischen Nachladen!
// Kann wieder entfernt werden.
```

Inlinestyles

```
<nav style="display : block;">
    Navigation
</nav>

// Einfach und elementbezogen verwendbar
// Anwendungsspezifisch.
// Möglichkeit der punktuellen/temporären Ergänzung.
// Einbindung für dynamisches Nachladen!
// Kann wieder entfernt werden.
// Wird dynamisch und zeitlich begrenzt,
// zum Beispiel von jQuery, erzeugt.
```


Skalierbares, modulares CSS

Stylesheets strukturiert aufbauen

- + CSS können jedes Element und jede Eigenschaft einer Weboberfläche ansprechen und gestalten.
- + CSS ist eine deklarative Sprache.
- + CSS hat (noch) KEINE Struktur-Befehle.
- + Struktur muss konzeptionell erreicht werden.

kategorisieren!

- + Die Aufgaben von Stylesheets können in immer gleiche Kategorien eingeteilt werden:
 - A. Grundsätzliches, Semantisches
 - B. Layout
 - C. Zustandsveränderungen
 - D. Inhaltliche Module
 - E. Projekt oder Kundenanpassungen

Selektoren

Selektoren und deren Spezifität

* {}	> Genereller Selektor	0
body {}	> Elementeselektor	1
.my-class {}	> Klassenselektor	10
a[href]	> Attributselektor	10
:hover	> Pseudoklasse (hier state)	10
#my-id {}	> ID-Selektor	100

Selektoren und was sie einfangen ...

```
p {}           -> <p></p>  
.row          -> <div class="row"></div>  
input[type=email] {} -> <input type=email>  
.icon::after {}      -> <span class="icon"></span>  
a[href]:hover {}     -> <a href="..."></a>  
section#data {}       -> <section id="data"></section>
```

Genereller Selektor

- * Stimmt mit jedem Element überein.

Er besitzt eine Spezifität von 0 und wird daher von jeder anderen Styleanweisung überschrieben.
Eine Spezifität von 1 erreicht man mit

```
body * { ... }
```

Element Selektor

Stimmt mit jedem E-Element überein
(d. h. ein Element des Typs E).

-> `h1 {}`

Collection

`<h1>`

`<h1>`

`<h1>`

⋮

Id Selektor

Eine ID muss im Dokument unique sein!

-> section#content {}

-> #content {}

Collection

```
<section  
id="content">
```

Klassenselektor

Klassen sind mehrfach
verwendbar.

-> section.page-
content {}

-> .page-content {}

Collection

```
<section  
class="page-  
content">
```

```
<section  
class="page-  
content">
```

⋮

Attribut- selektoren

Stimmt mit jedem E-Element überein, dessen Attribut „foo“ gesetzt ist (ganz gleich, welchen Wert es hat).

→ `a[href]`

Stimmt mit jedem E-Element überein, dessen Attribut „foo“ genau den Wert von „warning“ hat.

→ `input[type=text]`

``

``

`<input type="text">`

`<input type="email">`

Attributselektor en

Liste von durch Kommas voneinander getrennten Werten, und einer dieser Werte ist gleich „bar“.

→ div[class*=column]

→ [lang|=en]

Liste mit durch Trennstriche voneinander getrennten Werten, die (von links) mit „en“ beginnen.

```
<div class="green  
module bold  
nicecolumn">
```

```
<figure  
lang="en | fr |  
de">
```

Verneinung von Attributen

```
input[type=text] {  
    background: #eee;  
}  
input[type!=text] {  
  
    a[href] { ... }  
    a[!href]
```

Collections: erst sammeln, dann filtern



```
nav ul li a[href] {}
```

1. alle href: ['href', 'href', 'href', 'href', 'href', 'href']
2. in <a>: ['href', 'href', 'href', 'href', 'href', 'href']
3. in : ['href', 'href', 'href', 'href', 'href', 'href']
4. in ['href', 'href', 'href', 'href', 'href', 'href']
5. in <nav>: ['href', 'href', 'href', 'href', 'href', 'href']

CSS Selektoren Kombinationen

div.page {	→ beschränkter Klassenselektor
.group label {} ul li {}	→ Descendantselektor → Descendantselektor
nav > ul > li {}	→ Childselektoren-Kette
h1 ~ p {} h1 + p {}	→ General Following Siblings → Immediate Following Sibling
ul > li:first-child {} ul li:nth-child(odd) {}	→ numerischer Filter → logischer Filter

Descendant Selector

```
-> nav li { }
```

Collection



```
<nav>
<ul>
  <li> </li>
  <li> </li>
  <li> </li>
</ul>
</nav>
```

Child Selector

```
-> nav>ul>li { }
```

Collection

```
↳ <nav>
    ↳ <ul>
        <li> </li>
        <li> </li>
        <li> </li>
    </ul>
</nav>
```

~

General Sibling
(All Followers)

-> h1 + p {}

Collection



<h1>
<p> </p>
<p> </p>

+

Adjactent Sibling
(Immediate Follower)

-> h1 + p {}

Collection

↳ <h1>
<p> </p>
<p> </p>
<p> </p>

CSS Selektoren Pseudoklassen

a[href]:hover
(Pseudoklasse) → Stateselector
[class*=page-]::before {} → Pseudoelement
nav li:last-child {} → Pseudoklasse

:lang(de-DE)
Selektor → Sprachattribut

Pseudo-klassen

ul li:first-child

ul li:nth-child(3)

ul li:last-child

4

Nth-Child

```
h2:first-child { ... }

h2:last-child { ... }

.row:nth-child(even) { /*Gerade Zeilen */
  background: #dde;
}
.row:nth-child(odd) { /* Ungeraden Zeilen */
  background: white;
}
.row:nth-child(5n) { /* Jede fünfte Zeile */
  background: #dde;
}
.row:nth-child(5n+1) { /* Jede fünfte Zeile */
  background: #dde;
}
```

Nth-of-type()

```
div#test p:nth-of-type(4) { ... }  
div#test p:first-of-type { ... }  
div#test p:last-of-type { ... }  
div#test p:only-of-type { ... }
```

`nth-child()` vs. `nth-of-type`

```
<style>
  footer a:nth-child(odd) { color : lightblue; }
  footer a:nth-of-type(odd) { color: green; }
</style>
```

`<footer>` **nth-child()** vs. **nth-of-type** `</footer>`

```
<footer>
  <a href="">link</a>
  <span>Span</span>
  <a href="">link</a>
  <a href="">link</a>
  <span>Span</span>
  <span>Span</span>
  <span>Span</span>
  <a href="">link</a>
  <a href="">link</a>
  <a href="">link</a>
  <a href="">link</a>
</footer>
```

```
<footer>
  <a href="">link</a>
  <span>Span</span>
  <a href="">link</a>
  <a href="">link</a>
  <span>Span</span>
  <span>Span</span>
  <span>Span</span>
  <a href="">link</a>
  <a href="">link</a>
  <a href="">link</a>
  <a href="">link</a>
</footer>
```

Zustands- selektoren

`a:link, a:visited`
Stimmt mit dem Anchor
überein, dessen Ziel noch
nicht besucht wurde
(`:link`), oder dessen Ziel
bereits besucht wurde
(`:visited`).

`a:active, a:hover,`
`input:focus`

`input:valid, input:invalid`
Reflektiert ein
Validierungsergebnis auf
feldtypenspezifische
Eingaben.

``

``

`<input type="...">`

`<input type="email">`

Sprach-Pseudo-Klasse

```
:lang(c)  
-> p:lang(fr)
```

```
<p lang="fr">
```

:: für Pseudoelemente

```
<span data-currency=„€“>10</span>

span[data-currency]::after {
    content : attr(data-currency);
}

<span data-currency=„€“>10 €</span>
```

Toolipp per Pseudoelement

```
<span data-desc="tooltip text content">?</span> Text

span[data-descr] {
    position : relative;
    color    : #00F;
    cursor   : help;
    text-decoration : underline;
}

span[data-descr]:hover::after {
    content      : attr(data-descr);
    position     : absolute;
    left         : 0;
    top          : 24px;
    min-width   : 200px;
    background-color : #ffffcc;
    padding      : 12px;
    color        : #000000;
    font-size    : 14px;
    z-index      : 1;
    border       : 1px #aaaaaa solid;
    border-radius : 10px;
}
```

Negation von Selektoren

```
:not(.box) {  
    color: #00c;  
}  
:not(span) {  
    display: block;  
}  
  
h2:not(.teaser)  
statt h2.default, h2.special  
<h2 class="teaser"></h2> -> wird nicht ausgewählt!  
<h2 class="default"></h2>  
<h2 class="special"></h2>
```

Gewichtungen

CSS Gewichtungen

*	0 0 0 0 0	0
body	0 0 0 0 1	1
.my-class	0 0 0 1 0	10
:pseudo	0 0 0 1 0	10
[href]	0 0 0 1 0	10
#id	0 0 1 0 0	100
<p style="">	0 1 0 0 0	
!important	1 0 0 0 0	

CSS Gewichtungen

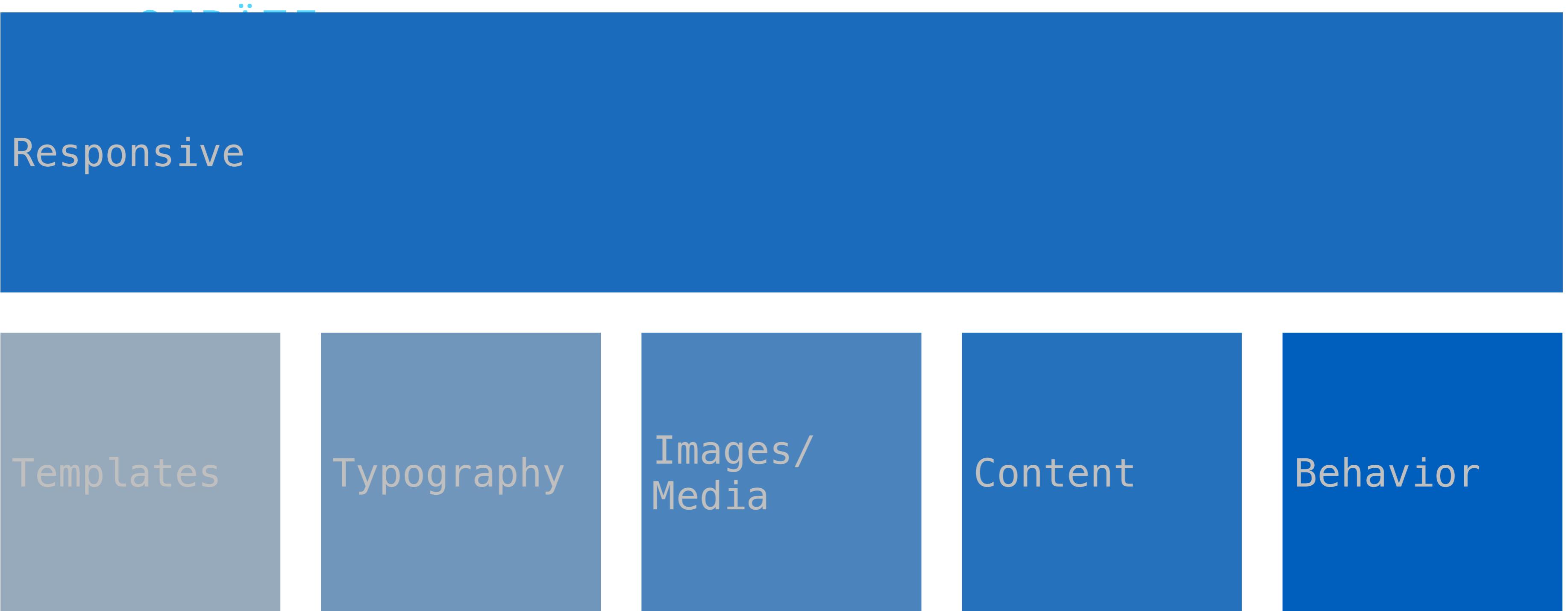
```
nav > ul > li 0 0 0 0 3  
a[href]          0 0 0 1 1  
#form-login      0 0 1 0 0  
a[href] { ... !important; } 1 0 0 1 1
```

CSS Gewichtungen

```
<body class=„red“> ... </body>

.red {
    background-color : red;
}

body {
    background-color : black !important;
}
```



Der typografische Raster

2021 Michael Reichart

$$M = 1\text{em} = M\text{-Höhe}$$

```
<html>          root  
...  
<body>          body  
<h1>...</h1>  headlines  
<p>...</p>    paragraphs  
...  
</body>  
</html>
```

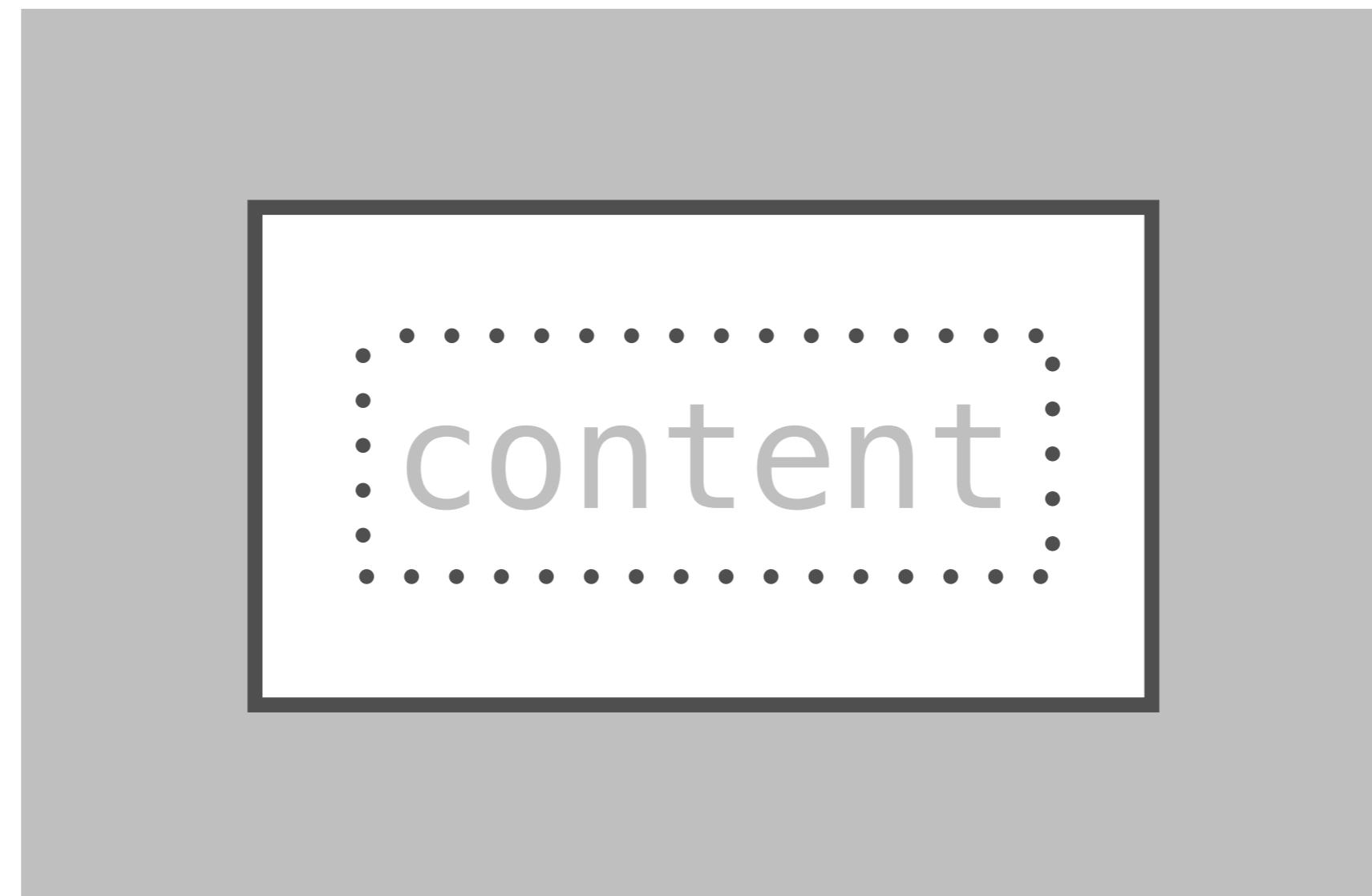
margins
paddings



font-family: Barlow;
font-size: 16px; 1rem
line-height: 1.35; 21.6px

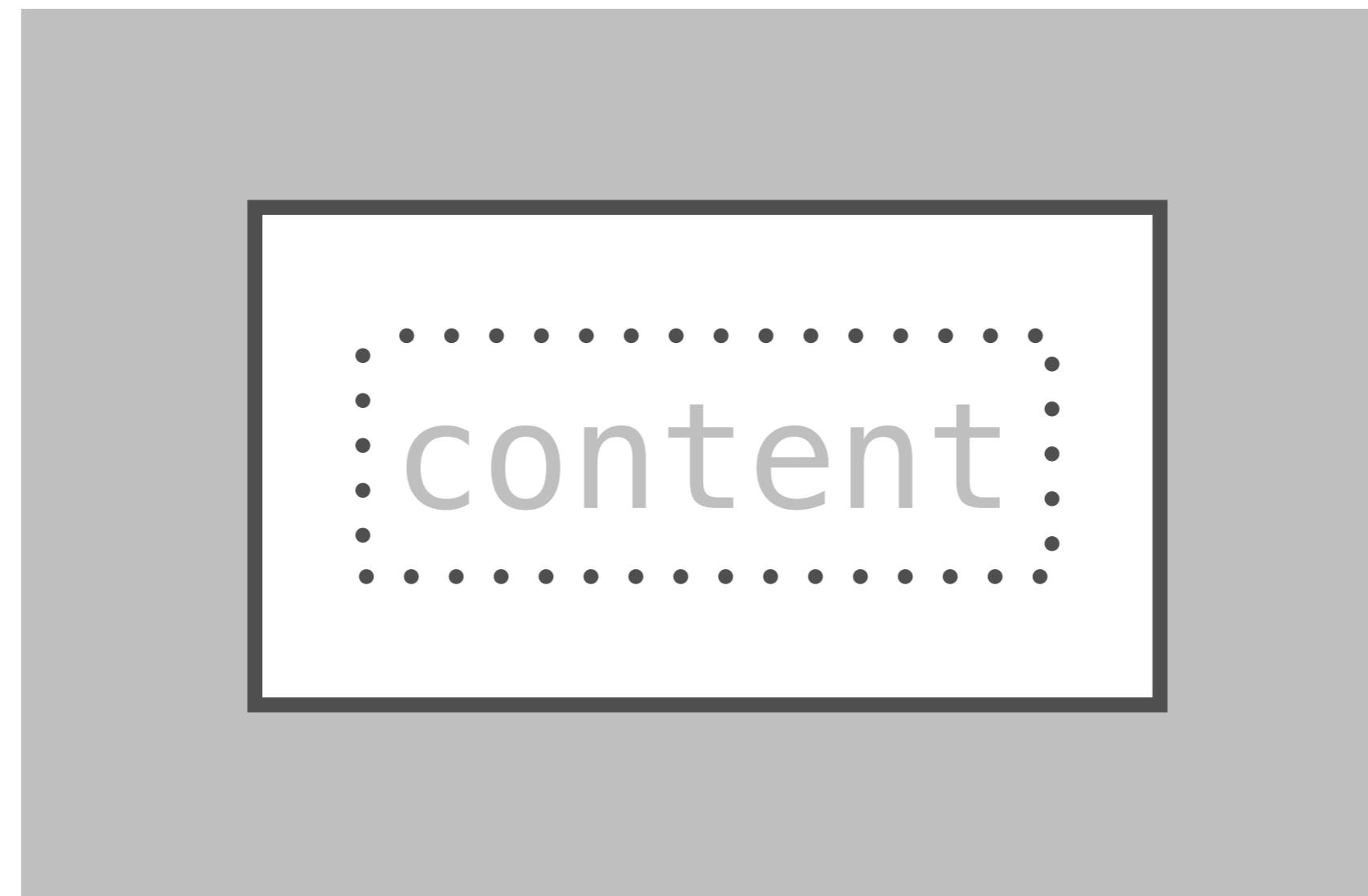
h1 { font-size: 2rem; }
h2 { font-size: 1.8rem;
 margin-bottom: 1.2rem; }

...
p { font-size: 1rem; }



height : 1rem;
padding : 1rem;
border-width : 1px;
margin-bottom : 1.2rem;

```
box-sizing: content-box;  
box-sizing: border-box;
```



```
display          : block;  
width           : 5rem; (80px)  
padding         : 1rem; (32px)  
border-width  : 1px; (2px)  
                           80px!  
  
margin-bottom   : 1.2rem;
```

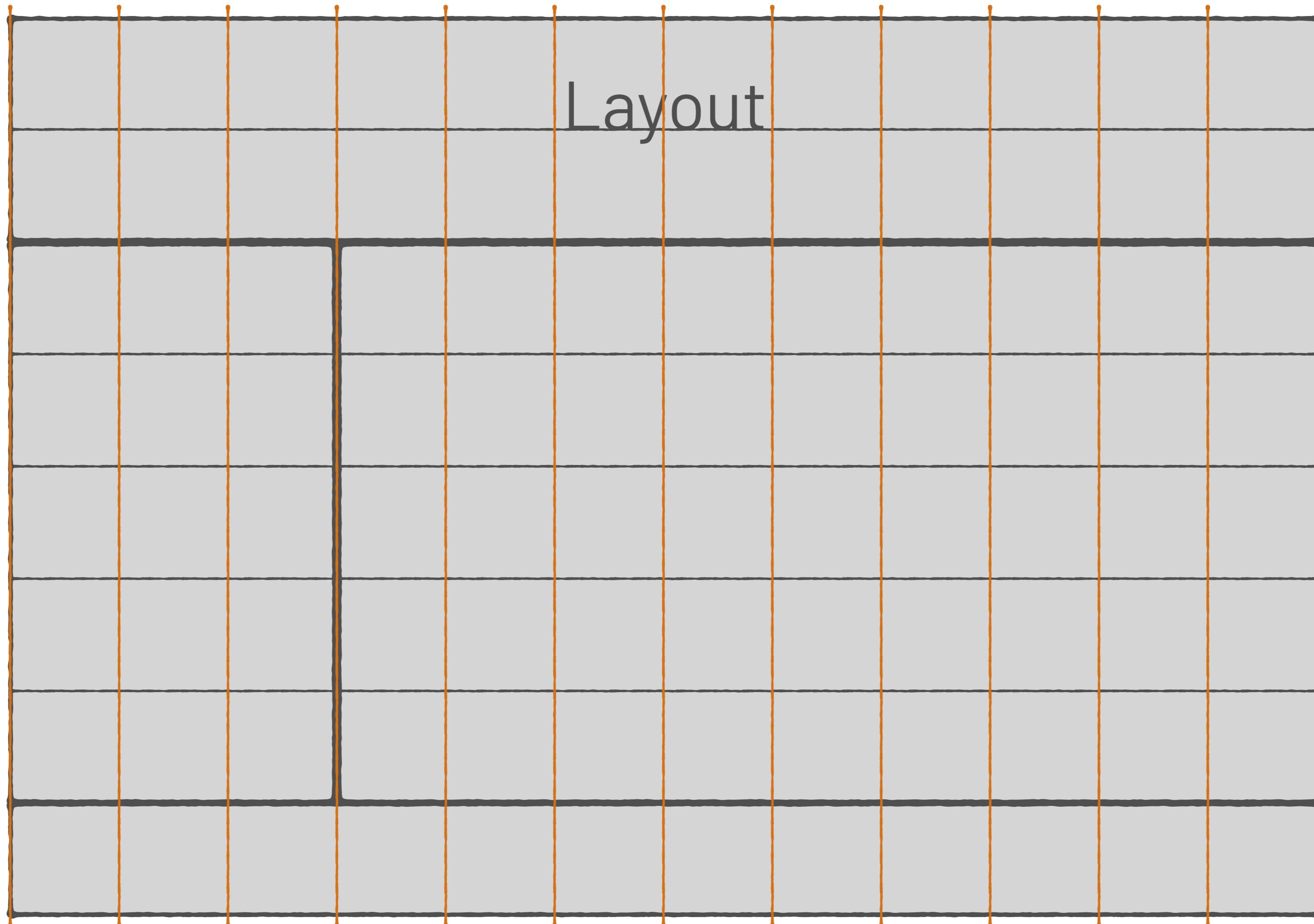
Typography

Paragraph (1rem) 1.35

Headline (2rem) 1.35
Headline (1.8rem) 1.35



Layout



Responsive Layouts

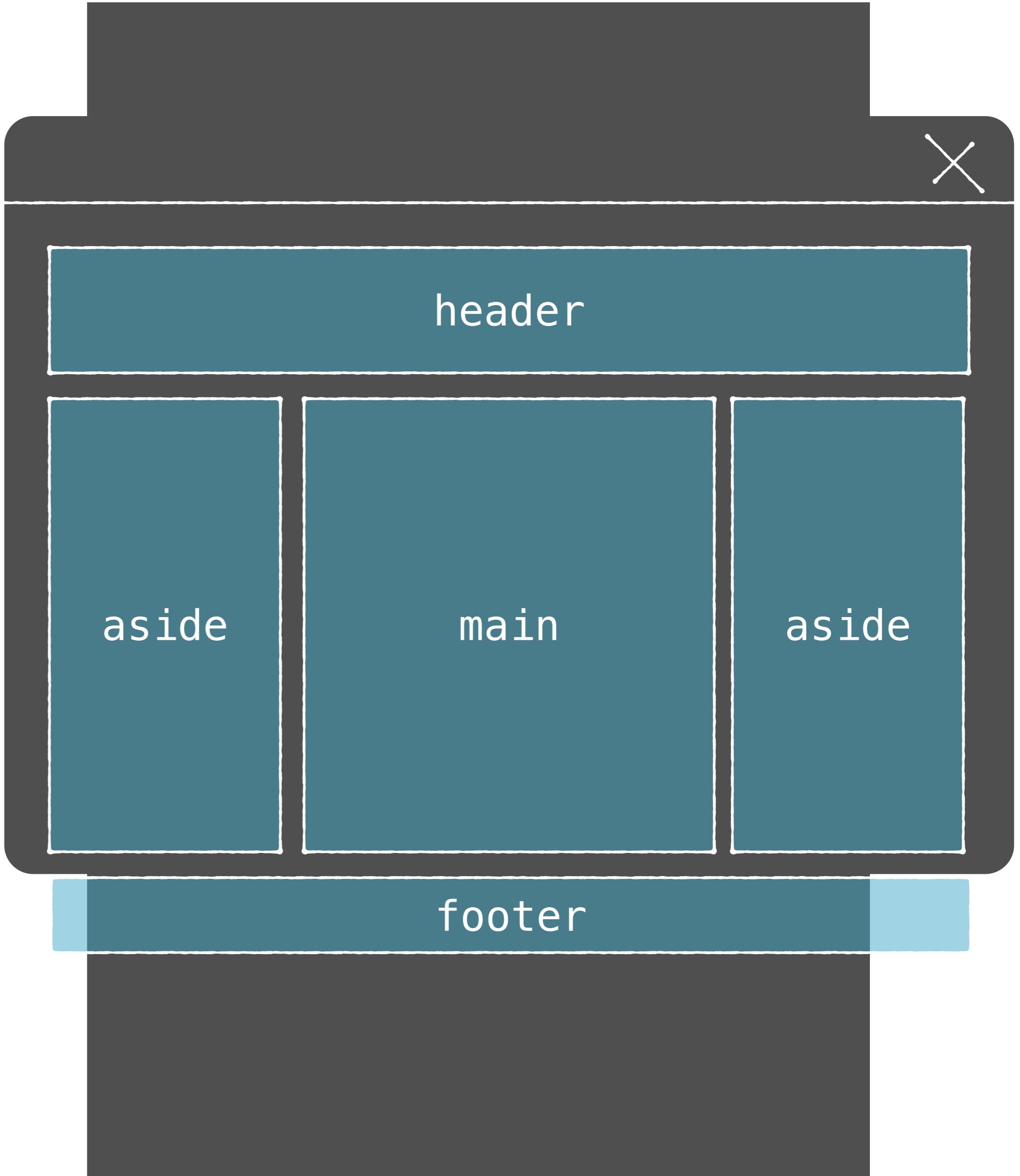
2021 Michael Reichart

Adaptives Design.

Für mobile Geräte heute Standard und ausreichend. Außerdem: Mobile First – Content First.

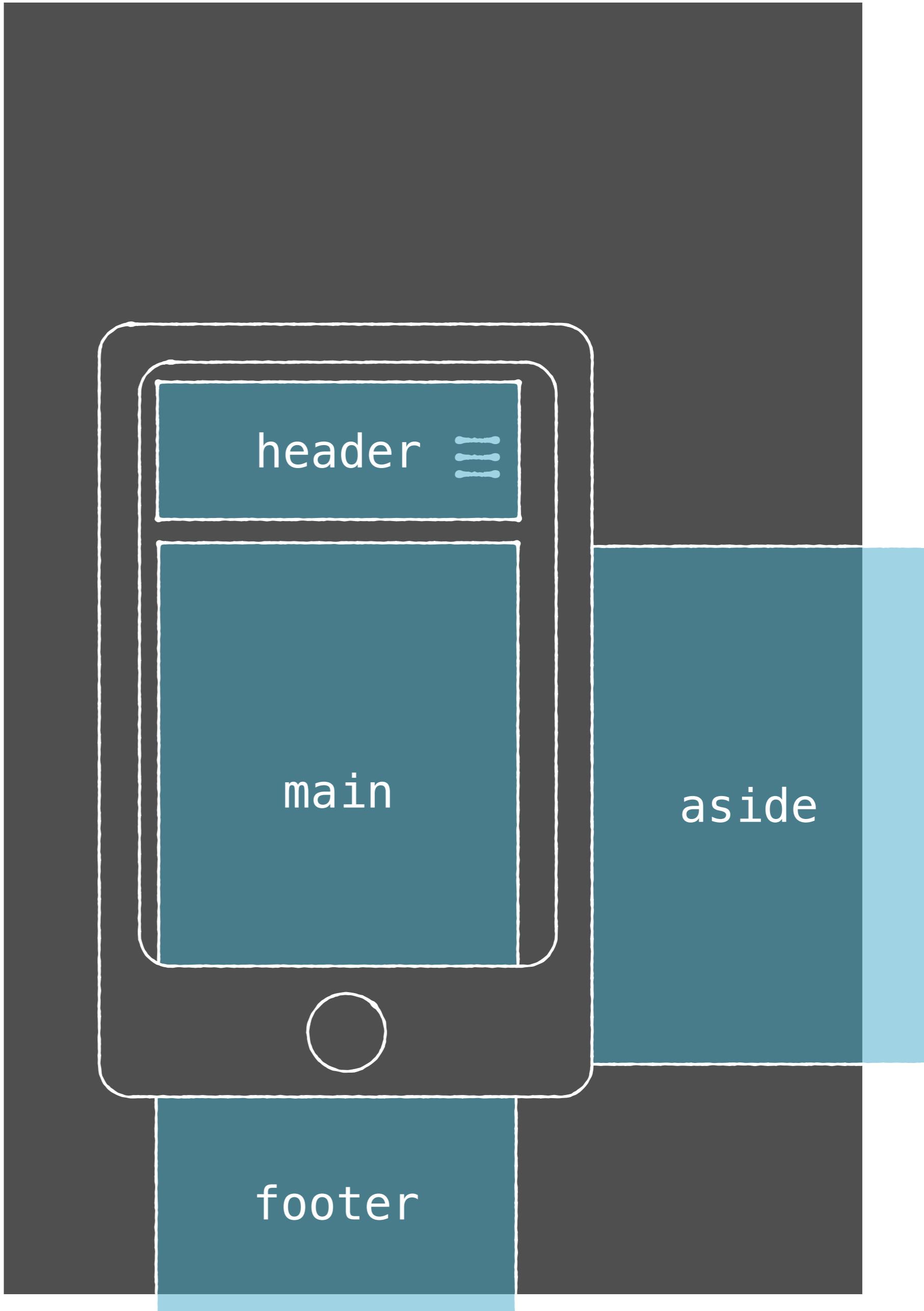
Desktop-layouts sind mehrspaltig.

- + Es gibt genügend Raum, um Navigation und Inhalte nebeneinander zu platzieren.



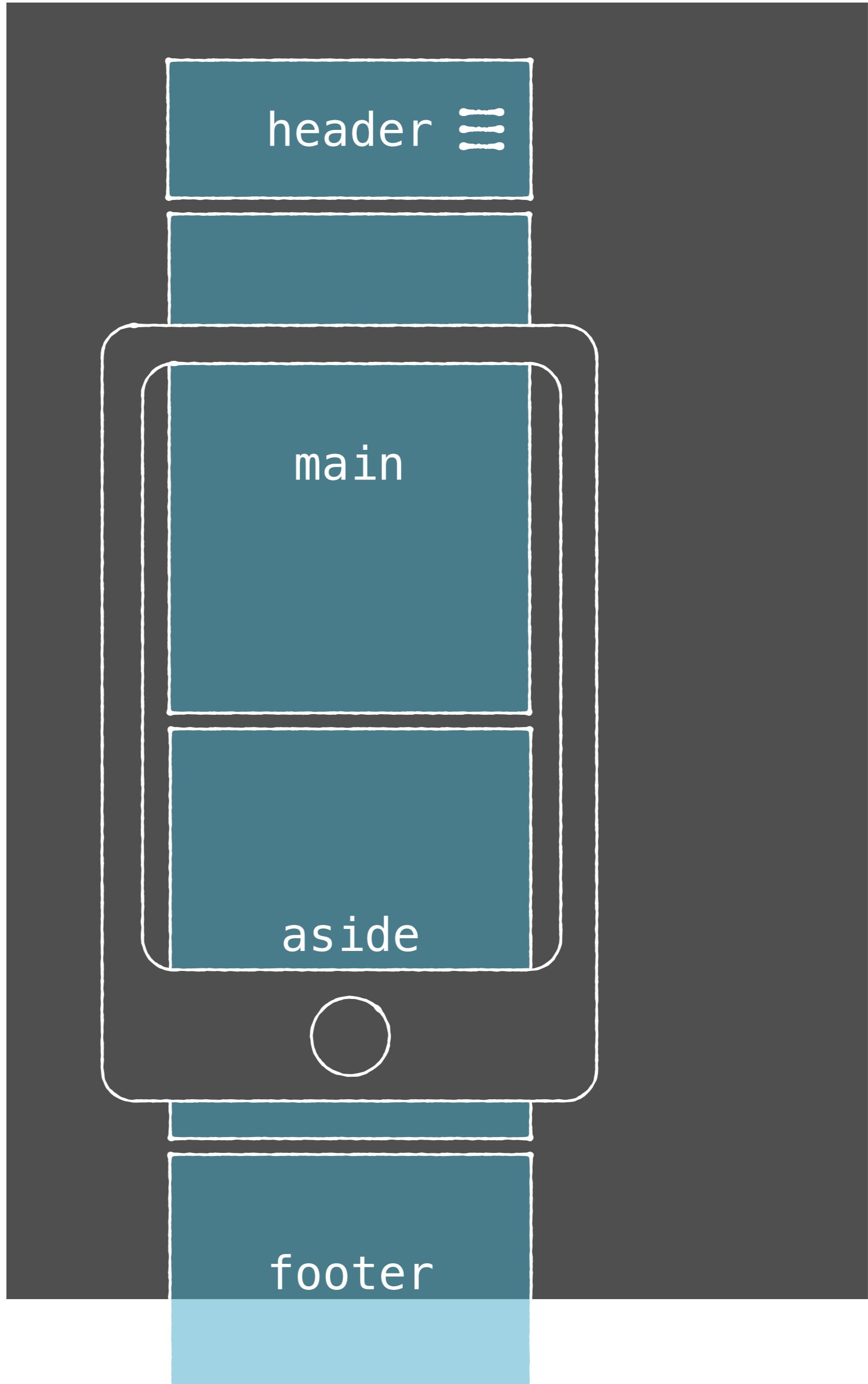
Smartphone layouts nicht.

- Besonders im Hochformat biete das Layout oft nur die Hälfte oder ein Drittel des Raumes.
- Die Anordnung der Layoutelemente muss geändert werden.

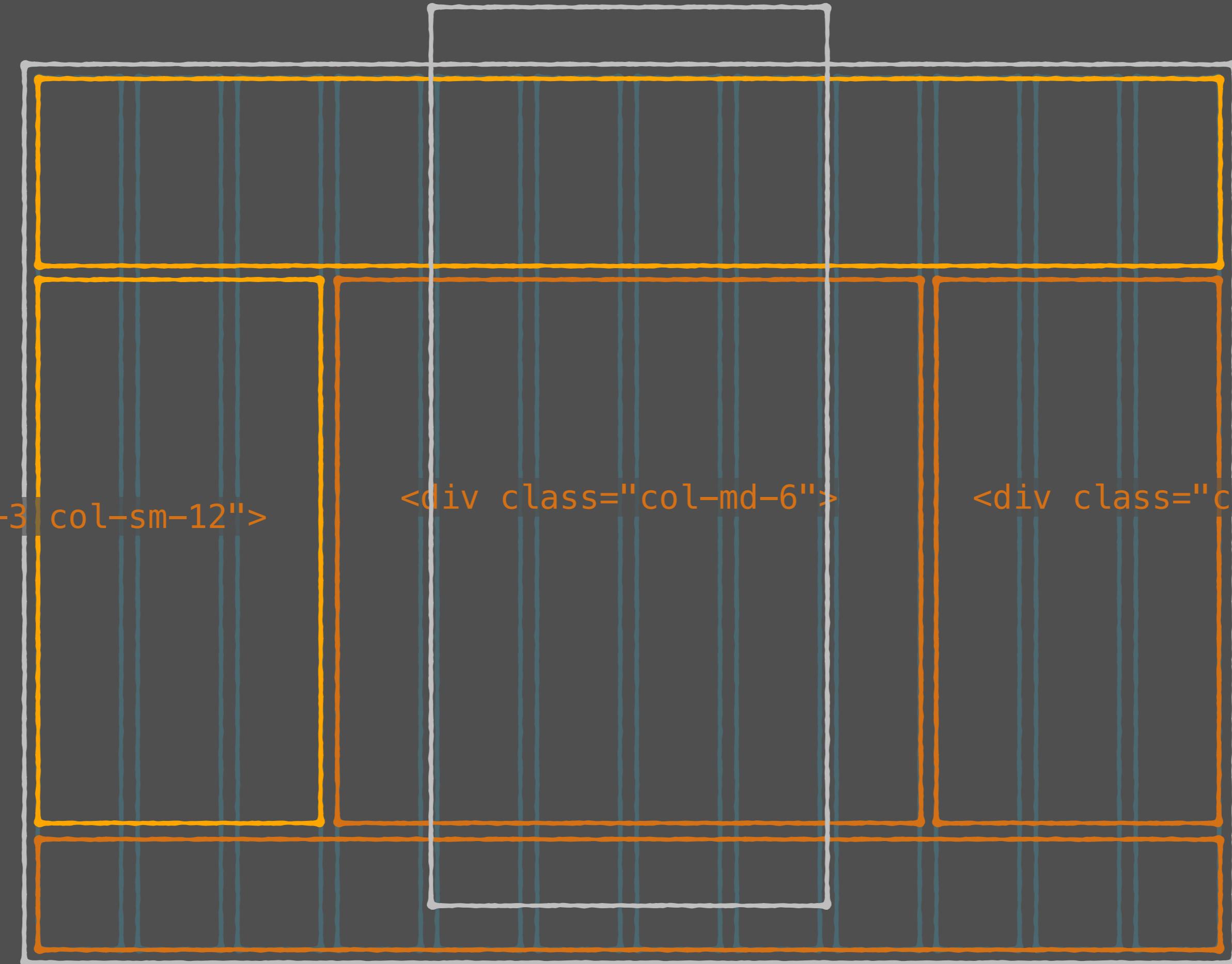


Linearisierung der Inhalte.

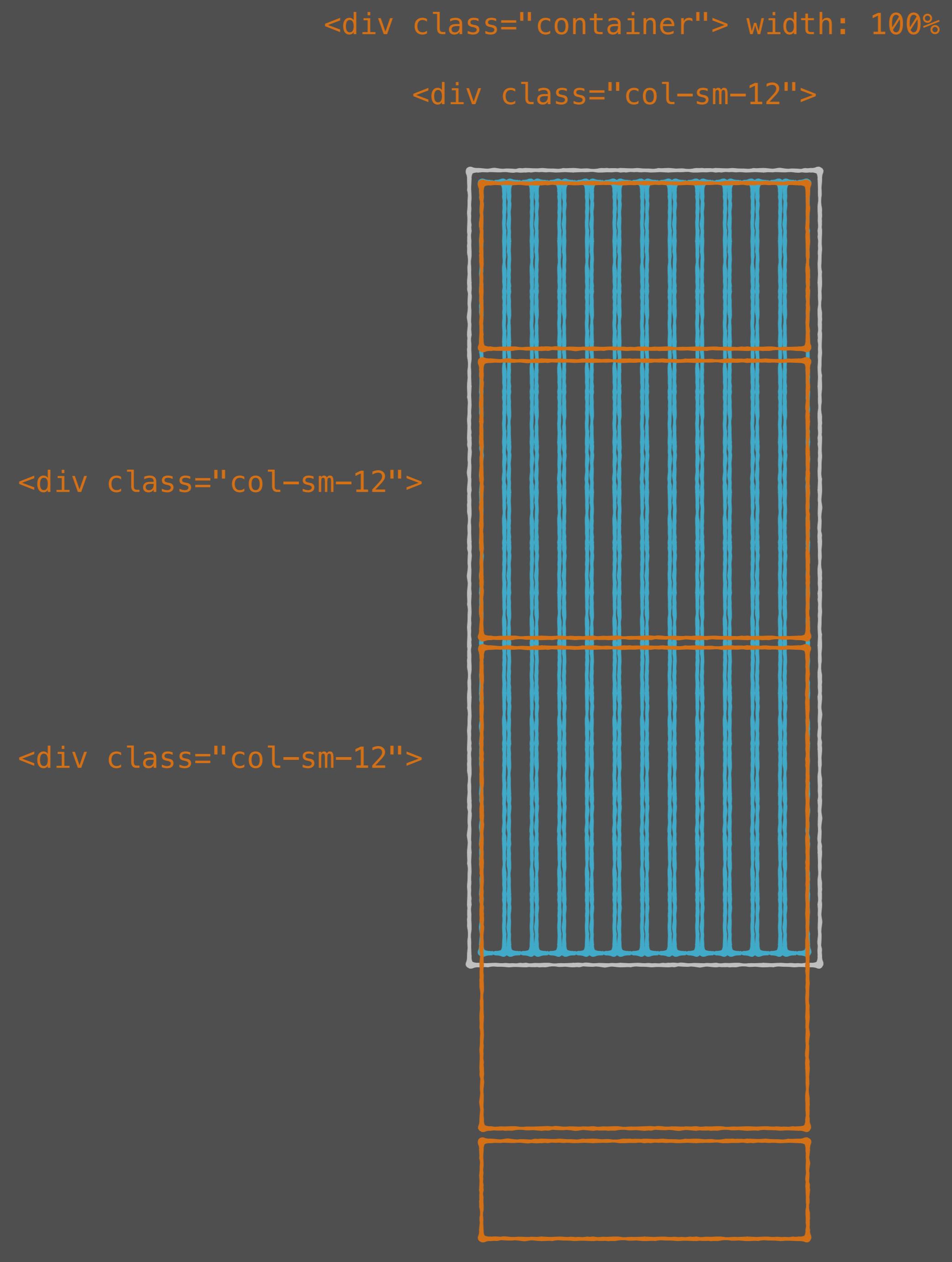
- Die Inhalte werden untereinander angeordnet, (Flow), einige Elemente werden gar nicht angezeigt.
- Mobile First = Content First



```
<div class="container"> width: 100%
  <div class="col-md-12">
```



```
<div class="col-md-12">
```



Linearisierung des Grids mit Media Queries.

Eine einfache Ergänzung: das Grid wird linearisiert,
jede Spalte liegt unter der anderen.

```
@media only screen  
  and (max-width: 767px) {  
  ...  
}
```

(Leider wird das bis IE8 nicht funktionieren.)

Überschreiben der width und margin/padding Attribute

```
.row {  
    width: auto;  
    min-width: 0;  
    max-width: 100%;  
    padding: 0 15px;  
    margin-left: 0;  
    margin-right: 0;  
}  
.row .span3 {width:25%;} muss überschrieben  
werden!  
.column, .columns {  
    width: auto !important;  
    float: none;  
}
```

Floats zurücknehmen.

```
.column:last-child, .columns:last-child {  
    float: none;  
}  
  
[class*="column"]  
+ [class*="column"]:last-child {  
    float: none;  
}  
  
.column:after, .columns:after {  
    clear: both;  
}
```

Für alle Displaybreiten eine passende Variante.

	Extra small devices Phones (<768px)	Small devices Tablets (>768px)	Medium devices Desktops (>992px)	Large devices Desktops (>1200px)
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints		
Max container width	None (auto)	750px	970px	1170px
Class prefix	.col-xs-	.col-sm-	.col-md	.col-lg
# of columns		12		
Max column width	Auto	60px	78px	95px
Gutter width	30px (15px on each side of a column)			
Nestable				
Offsets	N/A	Yes		
Column ordering	N/A	Yes		

Die Mediaqueries aus Bootstrap

```
/* Extra small devices (phones, less than 768px) */
/* No media query since this is the default in
Bootstrap */

/* Small devices (tablets, 768px and up) */
@media (min-width: @screen-sm-min) { ... }

/* Medium devices (desktops, 992px and up) */
@media (min-width: @screen-md-min) { ... }

/* Large devices (large desktops, 1200px and up) */
@media (min-width: @screen-lg-min) { ... }
```

Geräteerkennung mit CSS und Mediaqueries

Die wichtigen Änderungen und Erneuerungen

2021 Michael Reichart

Media Queries mit Attributen

- + HTML5 präzisiert Mediatypen.
- + Erweiterung durch Media Attributes wie 'width', und 'orientation'
- + Damit können medienabhängige Stylesheets angesprochen werden.

Medientypen werden bereits in HTML 4 definiert

+ screen	Bildschirme	
projection	Beamer	wird nicht benutzt
handheld	kleine Computer	wird nicht benutzt
tv	Fernsehgeräte	wird nicht benutzt
+ print	Drucker	
tty	Nadeldrucker	
+ aural	auditiv	deprecated seit CSS2
braille	Blindenschrift	
embossed	Blindenschrift	seit CSS2
speech	Sprachausgabe	seit CSS2
+ all	Alle Medientypen	seit CSS2

Externes medientyp-abhängiges Stylesheet

```
<link  
    rel="stylesheet"  
    type="text/css"  
    media="screen"  
    href="sans-serif.css"  
  >  
  
<link  
    rel="stylesheet"  
    type="text/css"  
    media="print"  
    href="serif.css"  
  >
```

Mediablock in einer CSS Datei

```
@media screen {  
    html { font-family: sans-serif  
}  
    ...  
}
```

In HTML5 gibt es abfragbare Medieneigenschaften

- width, height,
- device-width, device-height
- orientation
- aspect-ratio, device-aspect-ratio
- color, color-index, monochrome
- resolution
- scan
- grid

Extern oder interne Medientypisierung

```
<link  
    rel="stylesheet"  
    type="text/css"  
    media="screen and (color)"  
    href="example.css"  
    >  
  
@media screen and (color) { ... };
```

'all' umfasst alle Medientypen

'all' ist meist implizit
und umfasst alle Media Typen

```
@media (orientation: portrait)  
{ ... }
```

ist das gleiche, wie:

```
@media all and (orientation:  
portrait) { ... }
```

not als Negation

```
<link  
    rel="stylesheet"  
    type="text/css"  
    media="not screen and (color)"  
    href="example.css"
```

v

only screen

Das Keyword 'only' kann verwendet werden, um ein Stylesheet vor älteren User Agenten zu verbergen. Neue Browser ignorieren das Wort einfach.

```
<link  
    rel="stylesheet"  
    media="only screen and  
(color)"  
    href="example.css"  
v
```

Mehrere Media Queries als kommasseparierte Liste

```
@media  
  screen and (color),  
  projection and (color)  
{  
  ...  
}
```

Mediaqueries mit Fehlern

Wenn ein Media Feature auf ein Ausgabegerät nicht zutrifft, so liefert der User Agent ein false. Das Query wird abgelehnt.

```
<link  
  rel="stylesheet"  
  media="aural and (device-aspect-ratio: 16/9)"  
  href="example.css"  
>  
  
<link  
  rel="stylesheet"  
  media="speech and (min-device-width: 800px)"  
  href="example.css"  
>
```

Medienattribute.

Einsatz und Anwendungsbeispiele.

width, height

- + Wert: <length>, <height>
- + Anwendbar für visuelle und taktile Medientypen.
- + Kann mit min/max kombiniert werden.
- + 'width', 'height' bezeichnen die Breite/Höhe des Darstellungsbereiches.
- + Für kontinuierliche Medien sind das Breite und Höhe des Viewports inklusive der Scrollbar.
- + Für Medien, die mit Seiten arbeiten, sind das Breite und Höhe des Seitenlayouts oder der Textbox.
- + Breiten und Höhen dürfen nicht negativ sein.

Für Druckseiten mit einer Breite größer 25 cm.

```
<link  
    rel="stylesheet"  
    media="print and (min-width: 25cm)"  
    href="http://..."  
>
```

Für Geräte mit einem Viewport zwischen 400 und 700 Pixeln:

```
@media  
    screen  
    and (min-width: 400px)  
    and (max-width: 700px)  
{ ... }
```

device-width, device-height

- + Wert: <length>, <height>
- + Anwendbar für visuelle und taktile Medientypen.
- + Kann mit min/max kombiniert werden.
- + 'device-width' und 'device-height' bezeichnen die Breite/Höhe des Ausgabefläche. Bei kontinuierlichen Medien ist das die Bildschirmbreite. Bei seitenbasierten Medien ist das die Blattgröße.
- + Breiten und Höhen dürfen nicht negativ sein.

orientation

- + Value: portrait | landscape
- + Anwendbar für bitmap basierte Medien
- + Kann nicht mit min/max kombiniert werden.
- + 'orientation' ist 'portrait', wenn die Höhe größer oder gleich ist, als die Breite.
- + Sonst gilt 'landscape'.

Anwendung der Orientation

```
@media all and (orientation:portrait) { ... }  
@media all and (orientation:landscape) { ... }
```

aspect-ratio

- + Value: <ratio>
- + Anwendbar für bitmap basierte Medien
- + Kann mit min/max kombiniert werden.
- + 'aspect-ratio' definiert das Verhältnis von 'width' und 'height' eines Asugabebereiches.

device-aspect-ratio

- + Value: <ratio>
- + Anwendbar für bitmap basierte Medien
- + Kann mit min/max kombiniert werden.
- + The 'device-aspect-ratio' media feature is defined as the ratio of the value of the 'device-width' media feature to the value of the 'device-height' media feature.

Das Verhältnis 16:9

```
@media  
  screen and (device-aspect-ratio: 16/9) { ... }  
  
@media  
  screen and (device-aspect-ratio: 32/18) { ... }  
  
@media  
  screen and (device-aspect-ratio: 1280/720) { ... }
```

color

- + Wert: <integer>
- + Anwendbar für visuelle Medientypen.
- + Kann mit min/max kombiniert werden.
- + 'color' beschreibt die Anzahl von Bits pro Farbkanal des Ausgabegerätes.
- + Wenn das Gerät kein Farbgerät ist, dann ist der Wert 0.

Die beiden folgenden Queries gelten
für jedes Farbausgabegerät

```
@media all and (color) { ... }  
@media all and (min-color: 8) { ... }
```

color-index

- + Wert: <integer>
- + Anwendbar für visuelle Medientypen.
- + Kann mit min/max kombiniert werden.
- + The 'color-index' media feature describes the number of entries in the color lookup table of the output device. If the device does not use a color lookup table, the value is zero.

Zwei Wege, um Geräte mit indizierter Farbausgabe anzusprechen

```
@media all and (color-index) { ... }  
  
@media all and (min-color-index: 1) { ... }
```

monochrome

- + Wert: <integer>
- + Bei nicht monochromen Geräten wird 0 geliefert.
- + Anwendbar für visuelle Medientypen.
- + Kann mit min/max kombiniert werden.
- + Die monochrome - Angabe bezeichnet die Anzahl der bits pro Pixel eines einfarbigen Bildschirmspuffers.

Zwei Wege für ein monochromes Gerät

```
@media all and (monochrome) { ... }
```

```
@media all and (min-monochrome: 1) { ... }
```

resolution

- + Wert: <resolution>
- + Anwendbar für bitmap basierte Medien
- + Kann mit min/max kombiniert werden.
- + resolution beschreibt die Auflösung des Ausgabegerätes, zum Beispiel die Pixeldichte. Medien mit nicht-quadratischen Pixeln muss die least-dense mit dem Wert von min-resolution verglichen werden, max-resolution mit der most-dense Dimension. Ein einfache 'resolution'-Angabe funktioniert niemals mit einem Gerät mit nicht-quadratischen Pixeln.
- + Bei Druckern korrespondiert die 'resolution' mit der Druckauflösung (Anzahl der Druckpunkte) bei beliebigen Farbmodell.

Beispiele für Drucker

```
@media print and (min-resolution: 300dpi) { ... }  
@media print and (min-resolution: 118dpcm) { ... }
```

scan

- + Wert: progressive | interlace
- + Anwendbar für 'tv' Medien
- + Kann nicht mit min/max kombiniert werden.
- + 'scan' beschreibt das Zeilenaufbauverfahren eines TV- Ausgabegerätes.

Ein Beispiel für TV Geräte mit progressiven Scanning (Zeilenaufbau)

```
@media tv and (scan: progressive) { ... }
```

grid

- + Wert: <integer>
- + Anwendbar für visuelle und taktile Medien.
- + Kann nicht mit min/max kombiniert werden.
- + 'grid' wird verwendet, um Geräte mit einer festen Bildschirmmatrix (grid) zu verwenden. Das kann ein 'tty'-Terminal sein oder auch ein Handydisplay mit einer festgelegten Zeichenanzahl- und Größe. 'grid' liefert bei solchen Geräten eine '1'. Bitmapbasierte Displays erzeugen eine '0'.
- + Andere Werte als '0' oder '1' erzeugen ein falsches Mediaquery.

Zwei Beispiele:

1em entspricht dabei einer Zeichenbreite oder -höhe.

```
@media  
  handheld  
  and (grid)  
  and (max-width: 15em) { ... }
```

```
@media  
  handheld  
  and (grid)  
  and (device-max-height: 7em) { ... }
```

Mobile First Approach

Responsive Layouts

2021 Michael Reichart

Medien Attribute einsetzen

```
really very tiny devices:  
@media screen and (max-width:480px) { ... } including the device  
  
tiny devices:  
@media screen and (min-width:481px) { ... }  
  
small devices:  
@media screen and (min-width:768px) { ... }  
  
medium devices:  
@media screen and (min-width:1024px) { ... }  
  
large devices:  
@media screen and (min-width:1280px) { ... }  
  
xtra large:  
@media screen and (min-width:1600px) { ... }  
  
full hd device:  
@media screen and (min-width:1920px) { ... }
```

Mobile First Progressive Enhancement

PRINTER

```
@media print {}
@media print and (min-width:16cm) {}
```

ALL DEVICES

```
@media screen {
    declarations for all devices,
    optimized for xtra small mobile devices
}
```

XTRA SMALL

```
@media screen and (max-width:767px) {
    changes/specials for extra small
}
```

SMALL

```
@media screen and (min-width:768px) { ... }
```

MEDIUM

```
@media screen and (min-width:992px) { ... }
```

LARGE

```
@media screen and (min-width:1200px) { ... }
```

HD READY

```
@media screen and (min-width:1440px) { ... }
```

??

```
@media screen and (min-width:1600px) { ... }
```

FULL HD READY

```
@media screen and (min-width:1920px) { ... }
```

4K

```
@media screen and (min-width:3840px) { ... }
```

DESKTOP First GRACEFUL DEGRADATION

ALL DEVICES

```
@media screen {  
    declarations for all devices,  
    optimized for medium devices  
}
```

MEDIUM

```
@media screen and (min-width:1024px) and (max-  
width:1279px) {  
    changes/specials for medium  
}
```

SMALL

```
@media screen and (max-width:767px) { ... }
```

XTRA SMALL

```
@media screen and (max-width:480px) { ... }
```

LARGE

```
@media screen and (min-width:1280px) { ... }
```

XTRA LARGE

```
@media screen and (min-width:1600px) { ... }
```

HD

```
@media screen and (min-width:1920px) { ... }
```

Andere Attribute

```
@media screen and (orientation:portrait) { ... }

@media print and (resolution:300dpi) { ... }

@media screen and (min-resolution:96dpi) { ... }
@media screen and (min-resolution:192dpi) { ... } // Retina

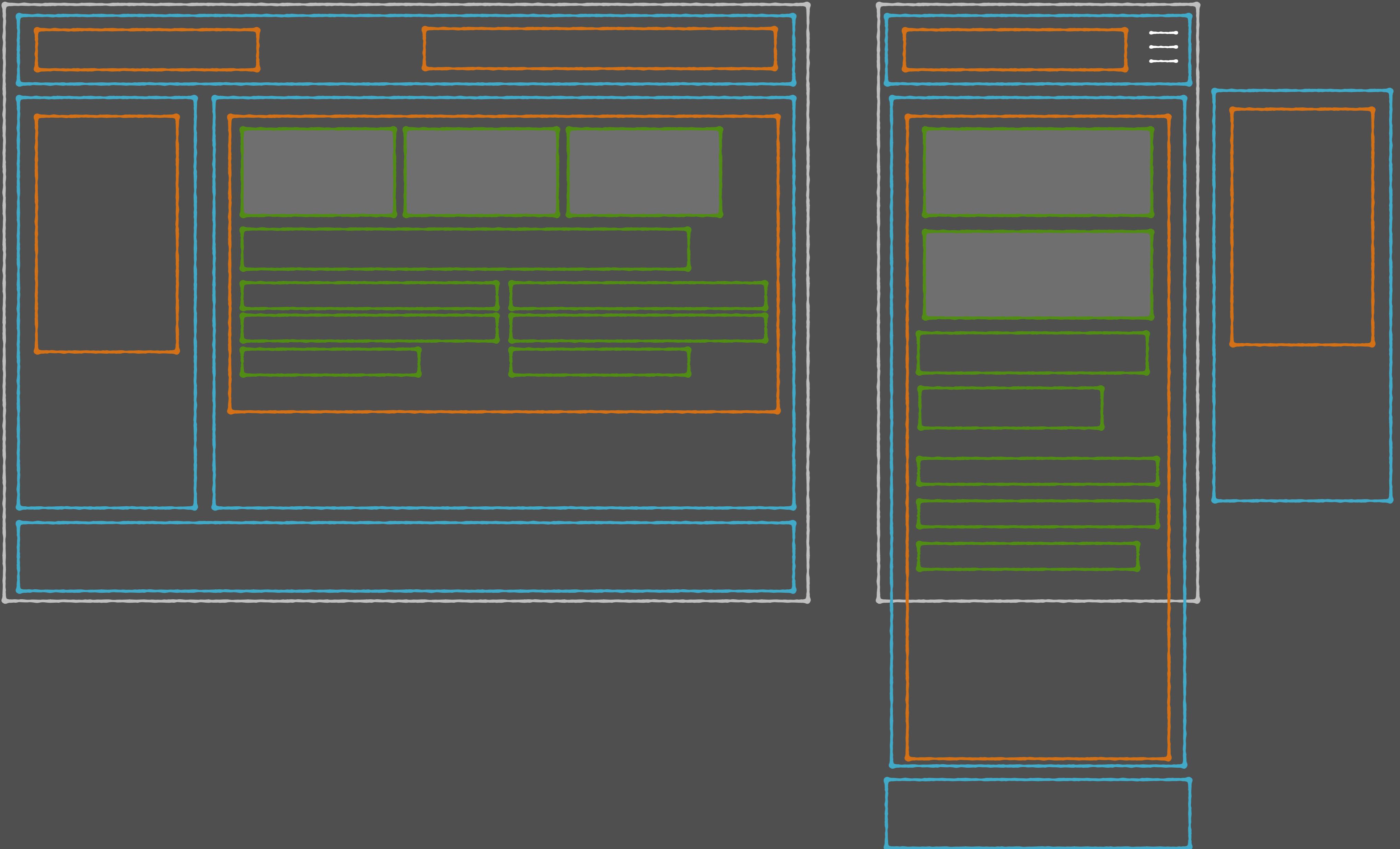
@media print and (color) { ... }

@media screen and (aspect-ratio:16/9) { ... }
```

Flex Box

Neues Layouten

2021 Michael Reichart



Was sind flex boxes?

- + Flex boxes bieten einen sehr effizienten Weg, um Layoutelemente anzurichten.
- + In einem „Container“ werden „Items“ angeordnet, ausgerichtet und verteilt, auch wenn deren Größe nicht bekannt oder dynamisch ist.
- + Responsive Layouts lassen sich mit flex boxes einfacher und stabiler umsetzen, als mit dem bisherigen float/clear Verfahren.

Flex Container

```
<div class="page" id="news">  
    <div class="page-header"> . . . </div>  
    <div class="page-content">  
        <div class="content-nav"> . . . </div>  
        <div class="content-main"> . . . </div>  
        <div class="content-aside"> . . . </div>  
    </div>  
    <div class="page-footer"> . . . </div>  
</div>
```

Flex items

```
<div class="page" id="news">  
    <div class="page-header"> . . . </div>  
  
    <div class="page-content">  
        <div class="content-nav"> . . . </div>  
        <div class="content-main"> . . . </div>  
        <div class="content-aside"> . . . </div>  
    </div>  
  
    <div class="page-footer"> . . . </div>  
</div>
```

Der Flexbox-Container

```
.page-content {  
    display        : flex;          /* Kindelemente werden Items      */  
    flex-direction : row;          /* Items zeilenweise anordnen   */  
    flex-wrap      : nowrap;        /* Items dürfen nicht umbrechen */  
    flex-flow      : /* Shorthand für flex-direction/flex-wrap */  
  
    align-items    : stretch;       /* Höhe der Items anpassen     */  
    align-content  : flex-start;    /* vertikale Ausrichtung der Items */  
  
    justify-content: space-between /* horizontale Itemausrichtung  */  
}
```

Flex-Items

```
.content-nav {  
    flex-basis : 25%; /* initiale Breite des Elements */  
    flex-grow : 1;    /* Proportion beim Vergrößern */  
    flex-shrink : 1;  /* Proportion beim Verkleinern */  
}  
.content-main {  
    flex-basis : 50%;  
    flex-grow : 2;  
    flex-shrink : 2;  
}  
.content-aside {  
    flex-basis : 25%;  
    flex-grow : 1;  
    flex-shrink : 1;  
}
```

Andere Eigenschaften für Items sind: `order` (Reihenfolge) und `align-self` (Vertikale Ausrichtung des Items)

<http://css-tricks.com/snippets/css/a-guide-to-flexbox/>

- Chris CoYier

A Fluid „holy-Grail“ Layout

```
/**  
 * LAYOUT STYLES  
 */  
  
/* BASE  
   the root container must have a 100% height */  
root,  
html, body {  
    display : flex;  
    flex-direction: column;  
    height : 100%;  
    min-height : 100%;  
}  
  
/* LAYOUT */  
.page { display : flex; width : 100%; flex-direction : column; height : 100%; min-height : 100%; }  
  
.page-header { display : block; width : 100%; flex-grow : 0; flex-shrink : 0; }  
.page-content { display : flex; width : 100%; flex-direction : row; flex-grow : 1; flex-shrink : 0; }  
.page-footer { display : block; width : 100%; flex-grow : 0; flex-shrink : 0; }  
  
.content-nav { display : block; flex-basis : 25%; flex-grow : 1; flex-shrink : 1; }  
.content-main { display : block; flex-basis : 50%; flex-grow : 2; flex-shrink : 2; }  
.content-aside { display : block; flex-basis : 25%; flex-grow : 1; flex-shrink : 1; }  
  
/* MARGINS AND PADDINGS */  
.page { display : flex; width : 100%; flex-direction : column; height : 100%; min-height : 100%; }  
  
.page-header { padding : 0.5rem; }  
.page-content { padding : 0; }  
.page-footer { padding : 0.5rem; }  
.content-nav { padding : 0.5rem; }  
.content-main { padding : 0.5rem; }  
.content-aside { padding : 0.5rem; }  
  
.content-nav > * { margin : -0.5rem; }
```

Responsive Content

Typographie, FArben, Verhalten

2021 Michael Reichart

Die Anzeige von Elementen steuern

display - appearance - text-overflow - box-sizing

block

```
div {  
    display : block;  
    width   : 100%;  
    [height  : auto;]  
    margin   : 1em;  
    padding  : 1em;  
}
```

display : block;

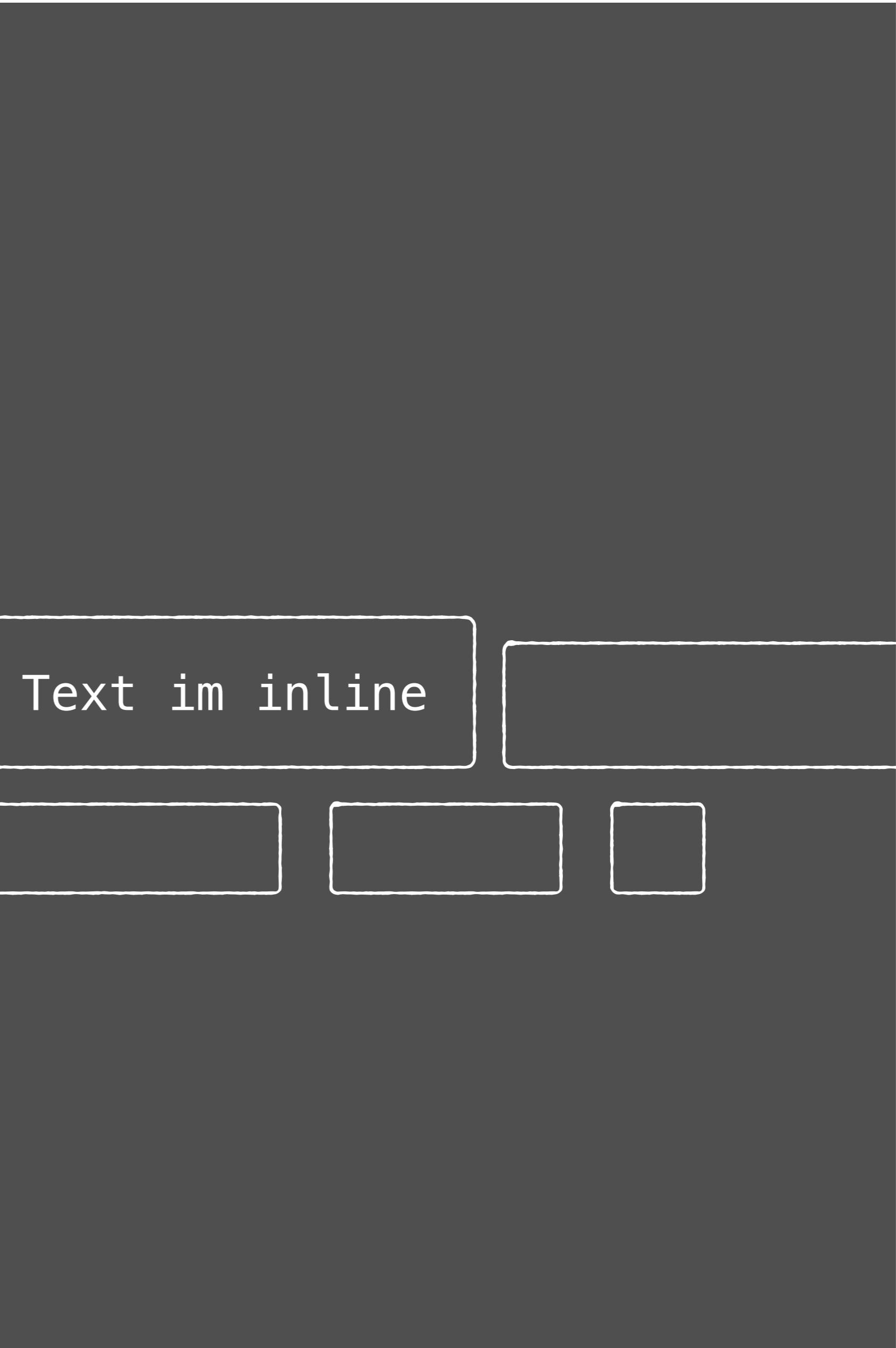
display : block;

display : block;

width : 100%;

inline

```
div {  
    display : inline;  
    width   : 100%;  
    height  : auto;  
    margin  : 1em;  
    padding : 1em;  
}
```

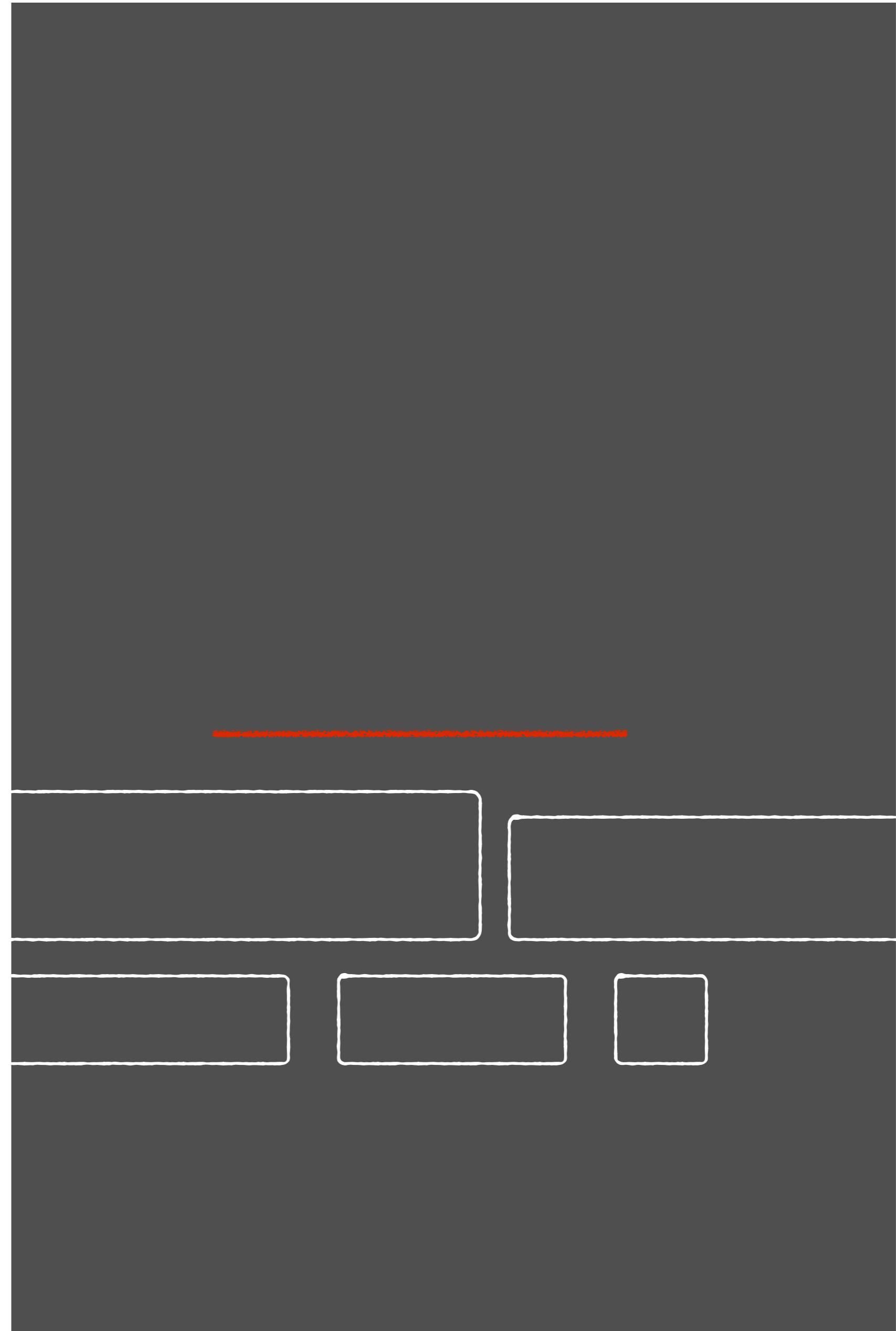


inline-block

```
div {  
    display: inline-block;  
}
```

Stattet das Inlineelement mit width/height aus, auf ein float Attribut kann hier teilweise verzichtet werden.

```
div {  
    display : inline-block;  
    width   : 12em;  
    height  : 12em;  
    margin  : 1em;  
    padding : 2em;  
}
```



Element mit Tabellenverhalten

```
div {  
  display: table;  
}  
div {  
  display: table-cell;  
}
```

Ein Blockelement erhält eine Höhe und verdrängt damit Elemente, auch wenn nichts drin steht.
Lässt eine vertikale Zentrierung zu!

Micro Clearfix von Gallagher

```
.cf:before, .cf:after {  
    content: " ";  
    display: table;  
}  
.cf:after {  
    clear: both;  
}  
.cf {  
    *zoom: 1; /* for IE 6/7 only */  
}  
  
/* seit CSS3: element::after { ... } */
```

Box Model Addition

`box-sizing: content-box`

Dimensionsverhalten wie in CSS 2 – Höhe, Breite werden definiert, padding, border, margin und outline werden dazu gerechnet.

`box-sizing: border-box`

In die Boxbreiten- und Höhenangabe werden padding und border eingerechnet.

```
div.row-fluid {  
    box-sizing : border-box;  
}
```

`box-sizing: padding-box`
Padding mit in die width Dimension

Appearance

Mit appearance kann man beliebige HTML-Elemente so aussehen lassen, dass sie wie die diversen Bedienelemente (Buttons, Textfelder) des Betriebssystems/Browsers aussehen.
Um einen Link wie eine Checkbox aussehen zu lassen, macht man etwa folgendes:

```
appearance:  
normal|icon|window|button|menu|field;  
  
section a {  
    appearance:checkbox;  
    -webkit-appearance:checkbox;  
    -moz-appearance:checkbox;  
    text-indent: 1.2em;  
}
```

WebKit values

```
checkbox
radio
push-button
square-button
button
button-bevel
listbox
listitem
menulist
menulist-button
menulist-text
menulist-textfield
scrollbarbutton-up
scrollbarbutton-down
scrollbarbutton-left
scrollbarbutton-right
scrollbartrack-horizontal
scrollbartrack-vertical
scrollbarthumb-horizontal
scrollbarthumb-vertical
scrollbargripper-horizontal
scrollbargripper-vertical
slider-horizontal
slider-vertical
sliderthumb-horizontal
sliderthumb-vertical
caret
searchfield
searchfield-decoration
searchfield-results-decoration
searchfield-results-button
searchfield-cancel-button
textfield
textarea
```

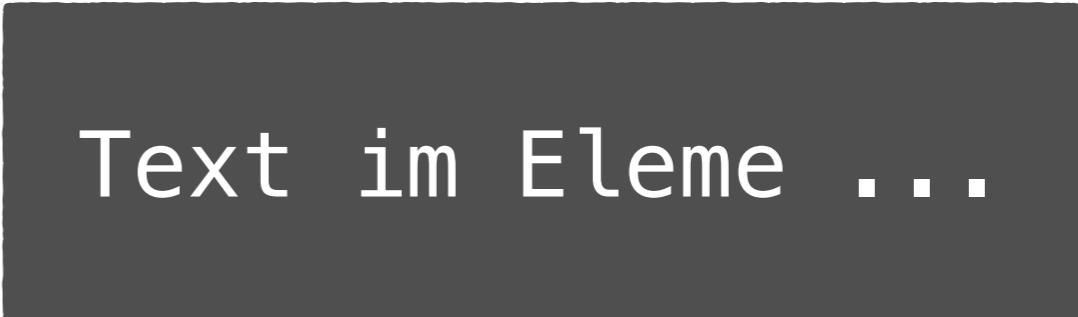
Mozilla values

```
none
button
checkbox
checkbox-container
checkbox-small
dialog
listbox
menuitem
menulist
menulist-button
menulist-textfield
menupopup
progressbar
radio
radio-container
radio-small
resizer
scrollbar
scrollbarbutton-down
scrollbarbutton-left
scrollbarbutton-right
scrollbarbutton-up
scrollbartrack-horizontal
scrollbartrack-vertical
separator
statusbar
tab
tab-left-edge Obsolete
tabpanel
textfield
textfield-multiline
toolbar
toolbarbutton
toolbox
-moz-mac-unified-toolbar
-moz-win-borderless-glass
-moz-win-browsertabbar-toolbox
```

Texte abschneiden

```
div {  
    text-overflow:  
ellipsis;  
}
```

// Nur bei
einzeligen Texten!



Text im Element ...

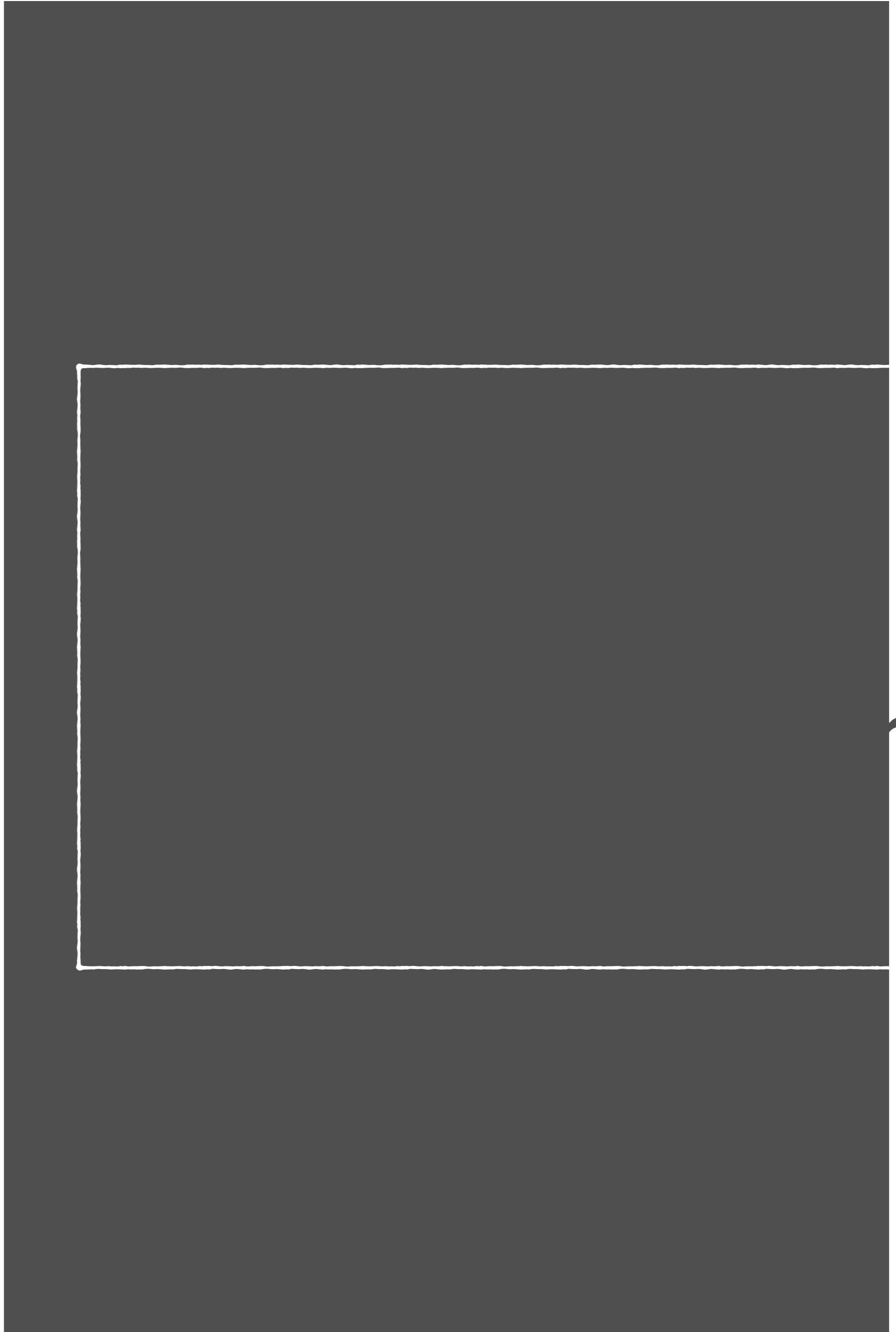
Schriften

2021 Michael Reichart

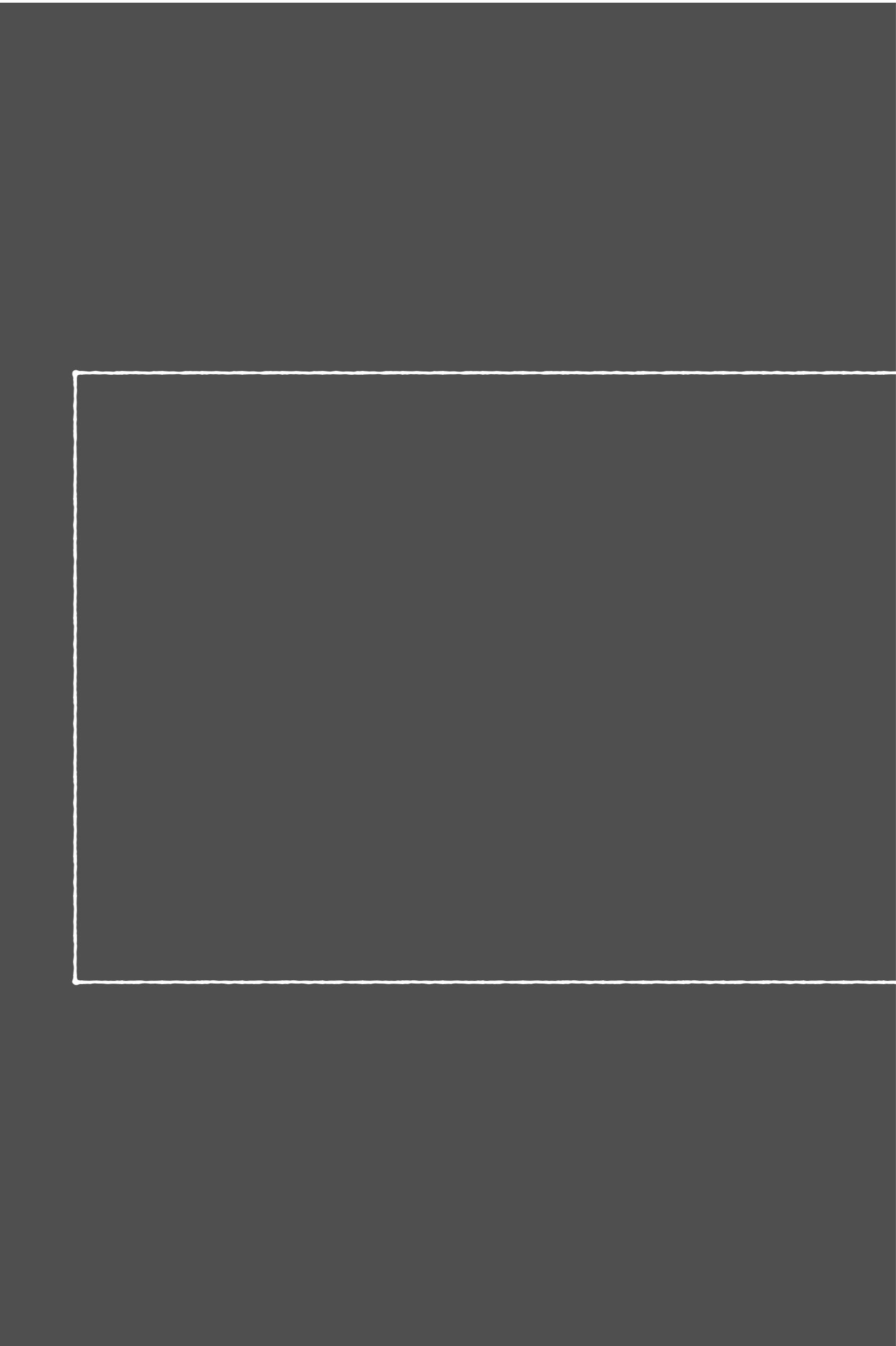
Typografie und Layout

- + Im heutigen Schriftgebrauch gibt es zwei Schriftformen:
Antiquaschriften und Groteskschriften.

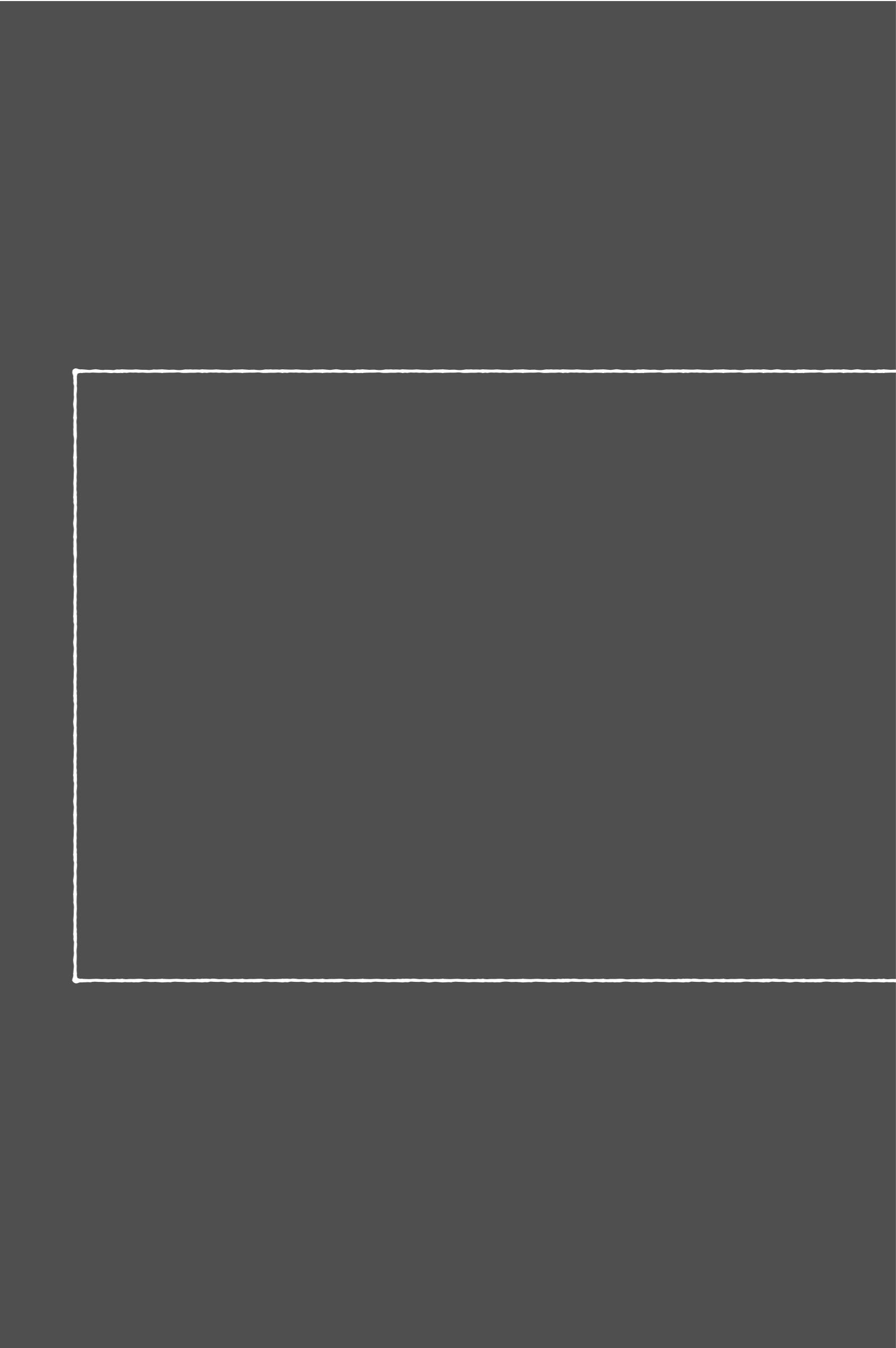
rift



- + Die Antiquaschriften sind serifen-betonte Schriften mit starken Unterscheidungen der Strichstärken.
- + Häufiger Einsatz für Texte in Büchern und Zeitungen.
- +



- + Groteskschriften sind serifelos, ohne oder mit geringen Unterschieden in der Strichstärke.
- + Häufiger Einsatz für Titel, Überschriften und in der Werbung.



Schrift-schnitte

- + dienen der Differenzierung der Schriftaussage, zum Beispiel "Überschrift", "wichtig", "Zitat"
- + Mit Schriftschnitten erreicht man ein interessantes Schriftbild.

LÄRDOCHT

Fein

normal

Halbfett

fett

kursiv

Extraschmal

schmal, fett

schmal, sehr fett

~~kleinbuchstaben~~

~~GROSSBUCHSTABEN~~

~~KAPITALCHEN~~

~~gesperrt~~

- + Ein kurzer Blick auf die Mikrotypografie:
Buchstaben, Abstände,
Zeilen



Mixgetränk

Grundlinie

Durchschuss (Zeilenabstand)

Ein kurzer Blick auf die
Mikrotypografie:

Buchstaben, Abstände,

Zeilen

Durchschuss (Zeilenabstand)

Mixgetränk

Grundlinie

Durchschuss (Zeilenabstand)

Mixgetränk

Grundlinie

Durchschuss (Zeilenabstand)

Mixgetränk

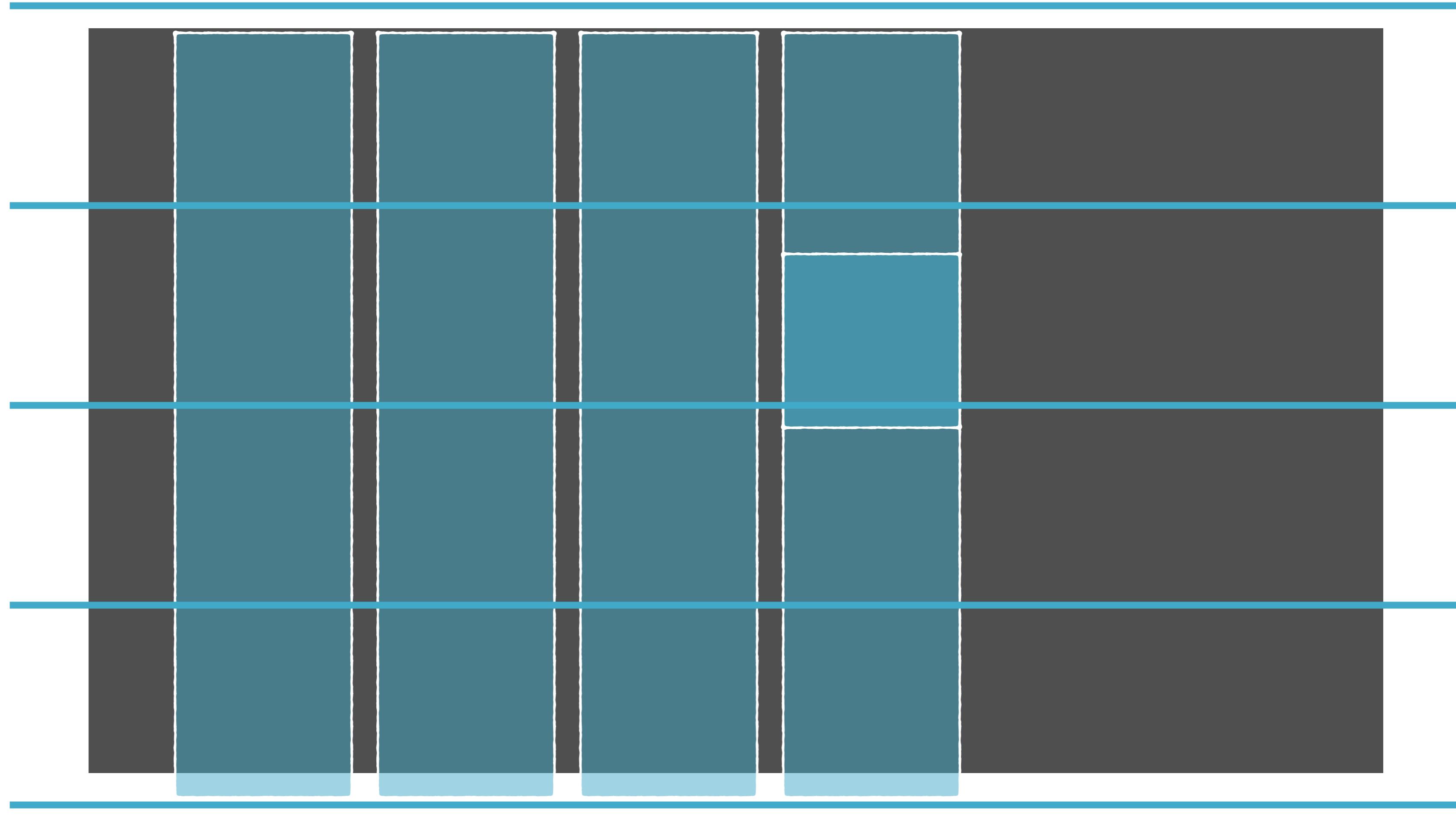
Grundlinie

Durchschuss (Zeilenabstand)

Mixgetränk

Grundlinie

Durchschuss (Zeilenabstand)



Grundlinien-

- + Ein kurzer Blick auf die Mikrotypografie:
Buchstaben, Abstände
Zeilen

Raster im

Verbund bilden

die Basis für

jedes Layout.

Schriftgrößen

Zeilenabstand

- + Ein kurzer Blick auf die Mikrotypografie:
Buchstaben, Abstände,
Zeilen

Sie werden so gewählt, dass sie

sich in ein einheitliches

Grundlinienraster einfügen.

Dies erzeugt einen ruhigen
Rhythmus und Lesbarkeit.

Rhythmus durch Grundlinien

+ Ein kurzer Blick auf die
Mikrotypografie:
Buchstaben, Abstände,
Zeilen

Nullam id dolor id nibh ultricies vehicula ut id elit. Aenean eu leo
quam. Pellentesque ornare sem lacinia quam venenatis vestibulum.

Cras mattis con-sectetur purus sit amet ferm-entum. Vivamus
sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Nulla
vitae elit libero, a pharetra augue.

Integer posuere erat a ante venenatis dapibus posuere velit aliquet.
Sed posuere consectetur est at lobortis. Curabitur blandit tempus
port-titor. Duis mollis, est non com-modo luctus, nisi erat porttitor
ligula, eget lacinia odio sem nec elit.

+

Ein kurzer Blick auf die Mikrotypografie: Buchstaben, Abstände, Der ~~Satz~~spiegel ordnet Bilder und Texte gleichermaßen.

Nullam id dolor id nibh ultricies vehicula ut id elit. Aenean eu leo quam. Pellentesque ornare sem lacinia quam venenatis vestibulum.

Cras mattis con-sectetur purus sit amet fermentum. Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Nulla vitae elit libero, a pharetra augue.

Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Sed posuere consectetur est at lobortis. Curabitur blandit tempus port-titor. Duis mollis, est non com-modoo luctus, nisi erat porttitor ligula, eget lacinia odio sem nec elit.

Lorem ipsum dolor.

Auch seitenüberspannend.

Donec ullamcorper nulla non metus auctor fringilla. Cras mattis consectetur purus sit amet fermentum. Aenean lacinia bibendum nulla sed consectetur. Nullam id dolor id nibh ultricies vehicula ut id elit. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Sed posuere consectetur est at lobortis. Donec id elit non mi porta gravida at eget metus.

Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Cras mattis consectetur purus sit amet fermentum. Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Duis mollis, est non commodo luctus, nisi erat porttitor ligula, eget lacinia odio sem nec elit.

+
Lorem ipsum dolor.

- + Ein kurzer Blick auf die Mikrotypografie:
Buchstaben, Abstände,
Der ~~Satz~~spiegel ordnet Bilder und
Texte gleichermaßen.

+
Lorem ipsum dolor.



Auch seitenüberspannend.

Nullam id dolor id nibh ultricies vehicula ut id elit. Aenean eu leo quam. Pellentesque ornare sem lacinia quam venenatis vestibulum.

Cras mattis con-sectetur purus sit amet fermentum. Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Nulla vitae elit libero, a pharetra augue.

Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Sed posuere consectetur est at lobortis. Curabitur blandit tempus port-titor. Duis mollis, est non com-modio luctus, nisi erat porttitor ligula, eget lacinia odio sem nec elit.

Donec ullamcorper nulla non metus auctor fringilla. Cras mattis consectetur purus sit amet fermentum. Aenean lacinia bibendum nulla sed consectetur. Nullam id dolor id nibh ultricies vehicula ut id elit. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Sed posuere consectetur est at lobortis. Donec id elit non mi porta gravida at eget metus.

Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Cras mattis consectetur purus sit amet fermentum. Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Duis mollis, est non commodo luctus, nisi erat porttitor ligula, eget lacinia odio sem nec elit.

Pharetra Sem Vulputate

Vulputate Dapibus

Fringilla

Nullam id dolor id nibh
ultricies vehicula ut id elit.

Aenean eu leo quam.

Pellentesque ornare sem lacinia
quam venenatis vestibulum.

Cras mattis con-sectetur purus sit
amet ferm-entum.

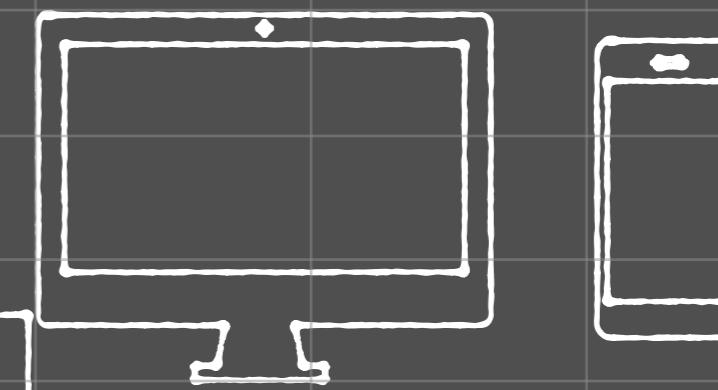
Vivamus sagittis lacus vel augue
laoreet rutrum faucibus dolor auctor.

Nulla vitae elit libero, a pharetra
augue.

Bildschirmraster verwenden sehr häufig
zwölf Spalten.

Nullam id dolor id nibh ultricies vehicula ut id elit. Aenean eu leo quam.
Pellentesque ornare sem lacinia quam venenatis vestibulum. Cras mattis
con-sectetur purus sit amet ferm-entum. Vivamus sagittis lacus vel augue
laoreet rutrum faucibus dolor auctor. Nulla vitae elit libero, a pharetra
augue.

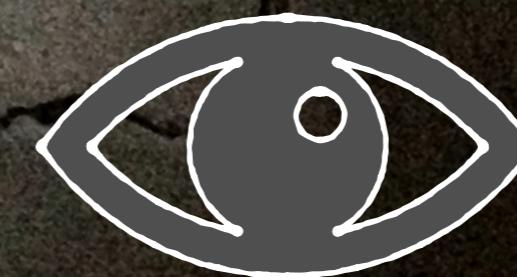
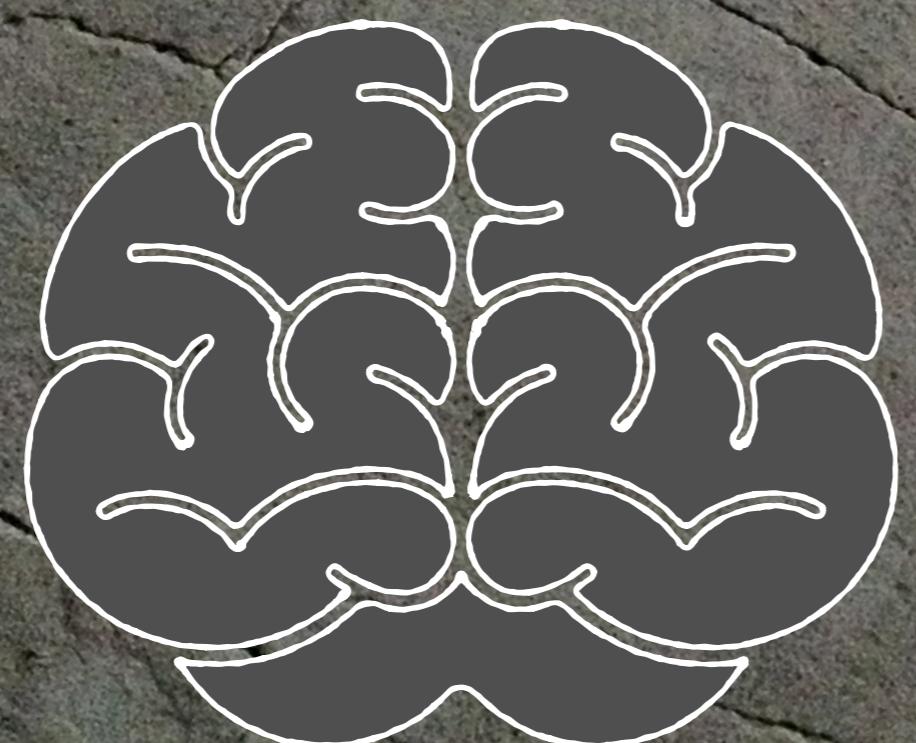
Integer posuere erat a ante venenatis dapibus posuere velit aliquet. Sed
posuere consectetur est at lobortis. Curabitur blandit tempus port-titor.
Duis mollis, est non com-modo luctus, nisi erat porttitor ligula, eget lacinia
 odio sem nec elit.



(c) 2017 Cras mattis con-sectetur purus sit amet ferm-entum.

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Nulla vitae elit libero, a pharetra augue.

iest jemand?



liest jemand?

+ Ein kurzer Blick auf die

Mikrotypografie:

Buchstaben, Abstände,
Zeilen

Ein Textabsatz wird nicht Wort

für Wort gelesen. Der Text wird

eher gescannt. Am Anfang des

Textes wird vielleicht die Zeile

noch bis zum Ende gelesen,

danach nur noch soviel, dass

sich der Inhalt erschließt.

liest jemand?

- + Ein kurzer Blick auf die Mikrotypografie:
Buchstaben, Abstände,
Zeilen

Das ghēt, wiel Wotre als
Konuetrn esrsfat wrdeen. Deis
ghet sslbet aus dem
Auegnwnikel heuras, onhe
akvties Hinhascuen. Der Txet
muss ncoh nicht eimnal rchitig
gchrieseben sein.

Richtiger Zeilenabstand ist wichtig, um den Zeilen folgen zu können

- + Ein kurzer Blick auf die Mikrotypografie:
Buchstaben, Abstände,
Zeilen

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi leo risus, porta ac consectetur ac, vestibulum at eros. Maecenas faucibus mollis interdum. Sed posuere consectetur est at lobortis. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Praesent commodo cursus magna, vel scelerisque nisl consectetur et.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi leo risus, porta ac consectetur ac, vestibulum at eros. Maecenas faucibus mollis interdum. Sed posuere consectetur est at lobortis. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Praesent commodo cursus magna, vel scelerisque nisl consectetur et.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi leo risus, porta ac consectetur ac, vestibulum at eros. Maecenas faucibus mollis interdum. Sed posuere consectetur est at lobortis. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Praesent commodo cursus magna, vel scelerisque nisl consectetur et.

enso die Länge der Zeilen. Zu lange Zeilen werden oft nicht bis zu Ende gelesen.

- + Ein kurzer Blick auf die Mikrotypografie:
Buchstaben, Abstände, Zeilen

 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi leo risus, porta ac consectetur ac, vestibulum at eros. Maecenas faucibus mollis interdum. Sed posuere consectetur est at lobortis. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Praesent commodo cursus magna, vel scelerisque nisl consectetur et. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Aenean lacinia bibendum nulla sed consectetur. Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Praesent commodo cursus magna, vel scelerisque nisl consectetur et. Cras mattis consectetur purus sit amet fermentum. Nullam quis risus eget urna mollis ornare vel eu leo. Aenean lacinia bibendum nulla sed consectetur. Cras justo odio, dapibus ac facilisis in, egestas eget quam.

 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi leo risus, porta ac consectetur ac, vestibulum at eros. Maecenas faucibus mollis interdum. Sed posuere consectetur est at lobortis. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Praesent commodo cursus magna, vel scelerisque nisl consectetur et. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Aenean lacinia bibendum nulla sed consectetur. Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Praesent commodo cursus magna, vel scelerisque nisl consectetur et. Cras mattis consectetur purus sit amet fermentum. Nullam quis risus eget urna mollis ornare vel eu leo. Aenean lacinia bibendum nulla sed consectetur. Cras justo odio, dapibus ac facilisis in, egestas eget quam.



Schrifteinbindung

2021 Michael Reichart

Einbindung von Zeichensätzen

```
@font-face {  
    font-family: LeagueGothic;  
    src: url(LeagueGothic.otf);  
}  
  
header {  
    font-family: LeagueGothic, Arial;  
}  
  
@font-face {  
    font-family: "Droid Sans";  
    src: url(Droid_Sans.ttf);  
}  
@font-face {  
    font-family: "Droid Sans";  
    src: url(Droid_Sans.woff2);  
}
```

Manuelles Einbinden verschiedener Formate

```
@font-face {  
    font-family : OpenSans;  
  
    src : url("/fonts/Open_Sans/OpenSans-Light.ttf")  
          format("ttf"),  
          url("/fonts/Open_Sans/OpenSans-Light.eot")  
          format("eot");  
    font-weight : 300;  
  
    src : url("/fonts/Open_Sans/OpenSans-Regular.ttf");  
    font-weight : 400;  
  
    src : url("/fonts/Open_Sans/OpenSans-  
              Semibold.ttf");  
    font-weight : 600;  
}
```

Schrift-schnitte

100 thin, extrafein
200
300 light (fein)
400 regular (normal)
500
600 semibold
(halbfett)
700 bold (fett)
800 black
900 heavy

Extrafein
Fein
normal
Halbfett
fett
kursiv
Extraschmal
schmal, fett
schmal, sehr fett

Google Fonts API

```
@import url(http://fonts.googleapis.com/css?  
family=Roboto:100,300,400,600,700,900);
```

* Schrift als Iconset nutzen

```
@font-face {  
    font-family : anyIconFont;  
  
    src         : url("/fonts/anyIconFont.ttf")  
                  format("ttf");  
}  
  
.icon-asterisk::before {  
    content: "\2a";          /* der UCS2 code für ein  
Zeichen */  
}  
.icon-plus::before {  
    content: "\2b";  
}  
  
<span class="icon icon-asterisk"></span>
```

Die Größe von Zeichensätzen justieren

↳ e

```
p.beispiel {  
    font-family:  
        Verdana, Arial, sans-serif;  
    font-size:12px;  
    font-size-adjust:0.58;  
}
```

Mit `font-size-adjust` kann man eine bestimmte x-Höhe erzwingen, ganz egal welche Schrift aus `font-family` eingesetzt wird. Im Beispiel wird die Schriftliste auf den Adjustwert der Verdana eingestellt.

Siehe auch

<http://www.webspaceworks.com/resources/fonts-web-typography/43/>

Textkonturen

```
h1 {  
    text-shadow: 0 0 1px black;  
}
```

Mehrspaltensatz

2021 Michael Reichart

Mehrspaltensatz - das Multicolumn Model

```
div { column-count: 3 }
div { column-width: 12rem }
div { columns: 3 12rem }

div {
  column-gap: 1rem;
  column-rule-width: 1px;
  column-rule-style:
  solid;
  column-rule-color:
  black;
}
```

In March 1936, an unusual confluence of forces occurred in Santa Clara County.

A long cold winter delayed the blossoming of the millions of cherry, apricot, peach, and prune plum trees covering hundreds of square miles of the Valley floor. Then, unlike many years, the rains that followed were light and too early to knock the blossoms from their branches.

Instead, by the billions, they all burst open at once. Seemingly

overnight, the ocean of green that was the Valley turned into a low, soft, dizzyingly perfumed cloud of pink and white. Uncounted bees and yellow jackets, newly born, raced out of their hives and holes, overwhelmed by this impossible banquet.

Then came the wind.

It roared off the Pacific Ocean, through the nearly uninhabited passes of the Santa Cruz Mountains and then, flattening out, poured down into the great alluvial plains of the Valley. A tidal bore of warm

air, it tore along the columns of trees, ripped the blossoms apart and carried them off in a fluttering flood of petals like foam rolling up a beach.

This perfumed blizzard hit Stevens Creek Boulevard, a two-lane road with a streetcar line down its center, that was the main road in the West Valley. It froze traffic, as drivers found themselves lost in a soft, muted whiteout. Only the streetcar, its path predetermined, passed on...

Umbruchsteuerung bei Mehrspaltensatz

```
h1 {  
    break-before: column;  
    break-inside: avoid-column;  
    break-after: avoid-column;  
}  
  
div {  
    width: 100px;  
    column-width: 45px;  
    column-gap: 0;  
    column-rule: none;  
}
```

Silbentrennung

```
p {  
    hyphens: none | manual | auto;  
}  
  
manual -> &shy;  
none   -> keine Trennungen  
auto   -> trennt immer dort, wo getrennt werden muss.
```

Ggf. Vendor-Präfixe, also „-moz-hyphens“, „-webkit-hyphens“

lang-Attribut verwenden, damit die Trennung nach Sprachregeln funktioniert!

Erzwungener Wortumbruch

Für Wörter mit Überlänge und ohne Trennregel:

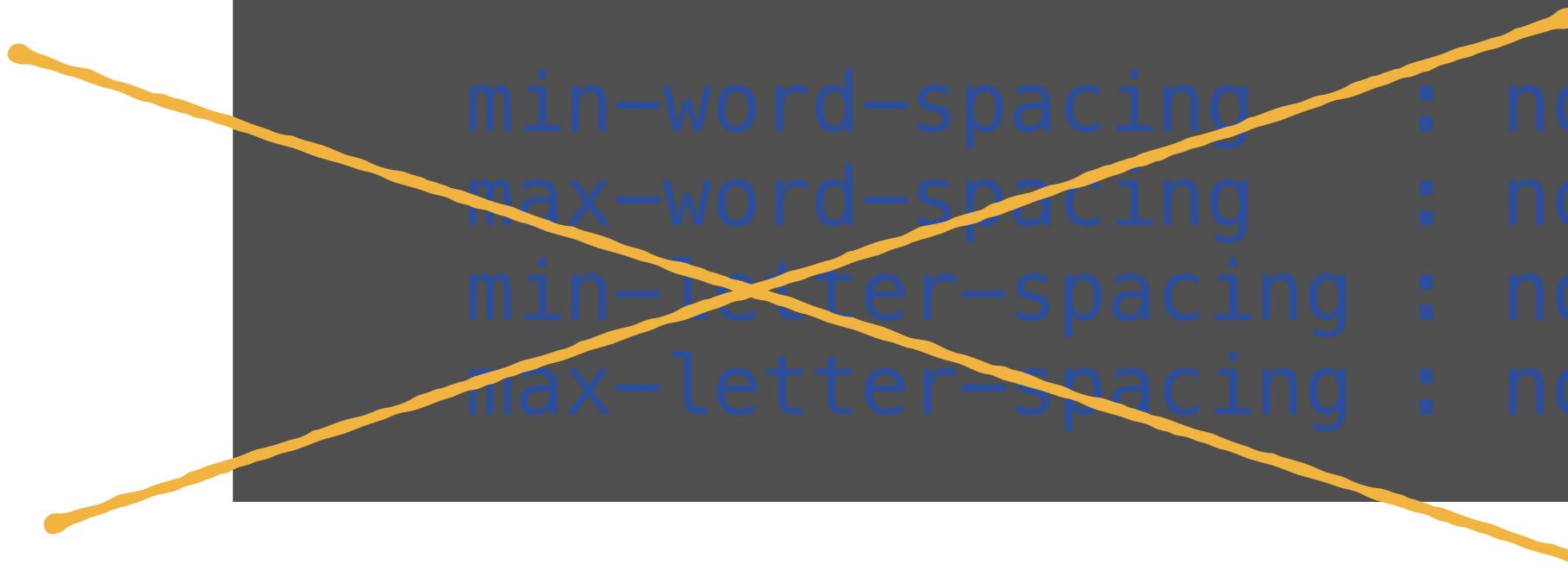
```
p {  
    word-wrap: normal | break-word;  
}
```

XML­Http<wbr>Request

Satzregeln

```
p {  
    text-align : justify;  
    hyphens   : auto;  
    widows    : 3;          /* Schusterjungen-Regel  
*/  
    orphans    : 3;          /* Hurenkinder-Regel */  
    word-spacing : normal;  
    letter-spacing : normal;  
    word-break : normal;  
}
```

```
min-word-spacing : normal;  
max-word-spacing : normal;  
min-letter-spacing : normal;  
max-letter-spacing : normal;
```

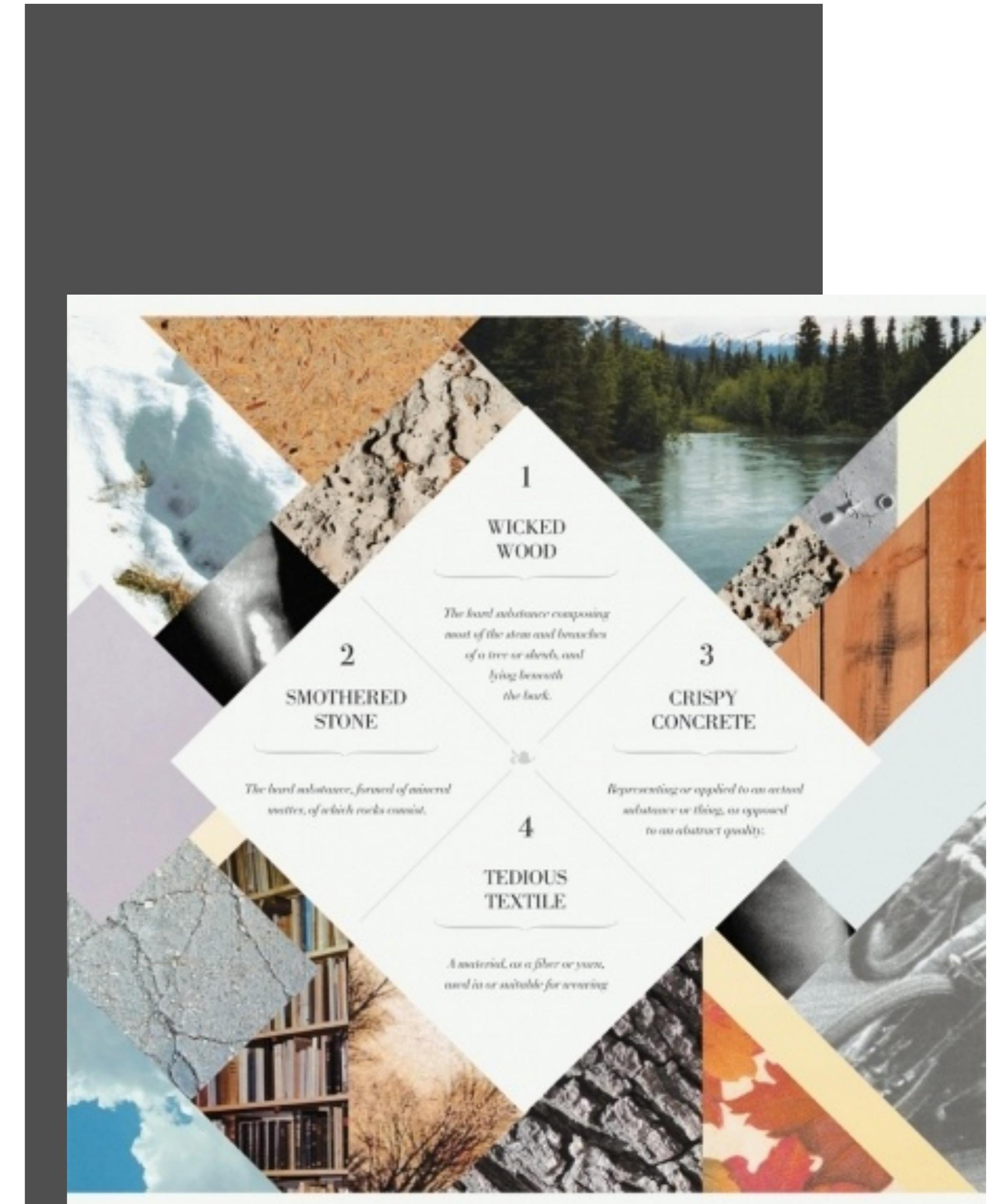


Formsatz

2021 Michael Reichart

FoRMsatz mit shape-Inside

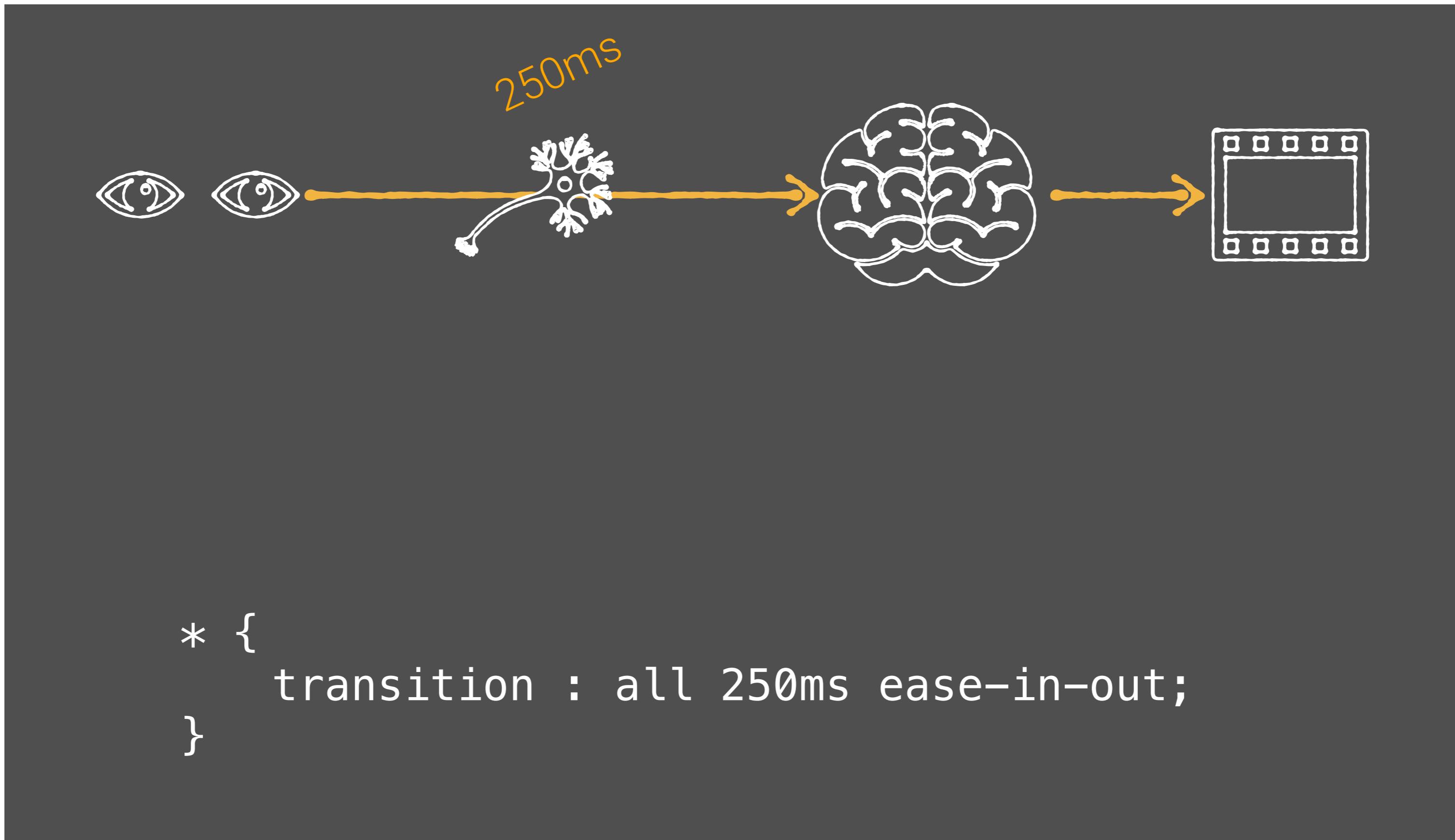
[http://
alistapart.com/
article/css-
shapes-101](http://alistapart.com/article/css-shapes-101)



Animationen und Übergänge

2021 Michael Reichart

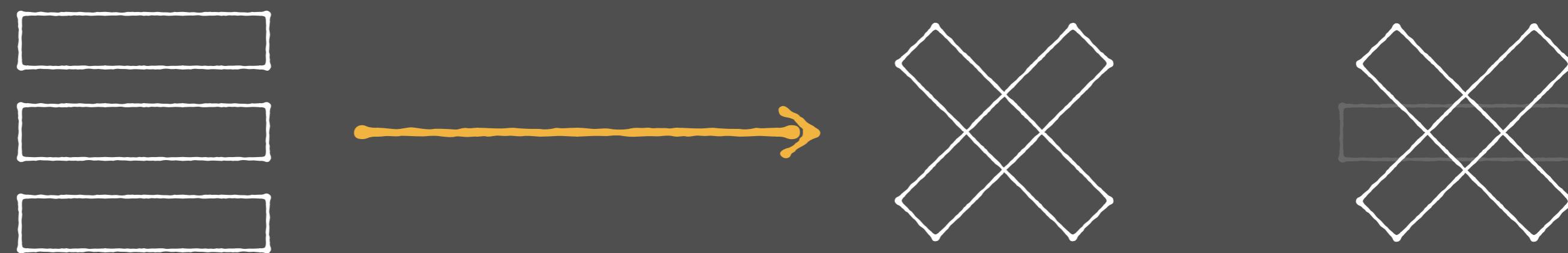
Wahrnehmung ist kein Ereignis, sondern ein Prozess!





font awesome icons einbinden und austauschen!

Transformation eines Hamburger Menüs.



10 Minuten

CSS Transition

Animationen und Effekte

- + Eine CSS Transition ändert den Wert einer CSS-Eigenschaft über eine festgelegte Zeit.
- + Dabei entstehen Animationen wie das Ein- und Ausblenden und Bewegen von HTML-Elementen ohne Javascript oder Flash.

Aufbau einer Transition

```
#box.left { margin-left: 0; }
#box.right { margin-left: 1000px; }

#box {
    transition : margin-left 1s ease-in-out;
}

document.getElementById( 'box' ).className = 'left';
document.getElementById( 'box' ).className = 'right';
```

transition property, duration, timing

transition-property

Die Eigenschaft, die während des Übergangs geändert wird.

transition-duration

Die Zeit, in der ein Übergang stattfinden soll.

transition-delay

Zeit bis zum Start des Übergangs.

transition-timing-function

Zeitlicher Verlauf des Übergangs.

Ein aufklappendes Menu

```
.content-nav {  
    display : block;  
    height  : 0;  
    overflow: hidden;  
  
    transition-property : height;  
    transition-duration : 0.25s;  
    transition-timing-function : ease-in-out;  
}  
.menu-show {  
    display : block;  
    height  : 5em;  
}  
  
$('button#menu-toggle').on('click', function () {  
    $('.content-nav').toggleClass('menu-show');  
});
```

Übergänge mit States

```
<div class="button"><a href="#">Mit Transition</a></div>  
...  
.button a {  
    background-color: #9FC0D0;  
    transition-property: background-color;  
    transition-duration: 1s; }  
  
.button a:hover { background-color: #759DB2; }
```

Die Easing Effekte

`ease`

Ohne weitere Angaben starten Transitions mit CSS langsam, beschleunigen in der Mitte und werden am Ende wieder langsamer.

`linear`

ist eine gleichförmiger Übergang,

`ease-in`

beginnt langsam und wird schneller,

`ease-out`

beginnt schnell und läuft langsamer aus

`ease-in-out`

kombiniert das langsame Anlaufen mit einem langsamen Auslaufen, aber einem deutlich schnelleren Teil in der Mitte.

`cubic-bezier(x1,y1,x2,y2)`

ist Fine-Tuning für den Verlauf der Animation

Animationen

```
@keyframes my-animation {  
    0% { background-color:black; }  
    25% { background-color:red; }  
    50% { background-color:green; }  
    75% { background-color:yellow; }  
    100% { background-color:black; }  
}  
  
.background {  
    animation: my-animation 4s infinite;  
}
```

Transformationen

```
transform : rotateY(45deg);
transform : scaleX(2.5%);
transform : translate3d(0, 0, 90px);
transform : perspective(500px)

#threed-example {
    transform : rotateZ(5deg);
    transition : transform 2s ease-in-out;
}

#threed-example:hover {
    transform : rotateZ(-5deg);
}
```

Farben und Farbverläufe, image processing

2021 Michael Reichart

Responsives Logo

```
<div class="brand"><h1 class="sr-only">Company Name</h1></div>

.brand {
    width      : 100%;
    height     : 100%;
    background : url(logo.svg) center center no-repeat;
    background-size : contain; /* cover */
}

.sr-only {
    display   : block;
    position  : absolute;
    left     : -10000px
}
```

Responsiver Vollbild Hintergrund

```
body {  
    background : url(background.jpg) center center  
no-repeat;  
    background-size : cover;  
}
```

Background Stacking

```
div {  
  background :  
    rgba(200,0,0,0.25),  
    linear-gradient(  
      to bottom,  
      #1e5799 0%,  
      #2989d8 50%,  
      #207cca 51%,  
      transparent 100%  
    );  
  repeat,  
  repeat-x;  
}url(src/overlay.png) 10px center no-  
url(src/background.png) 10px center
```

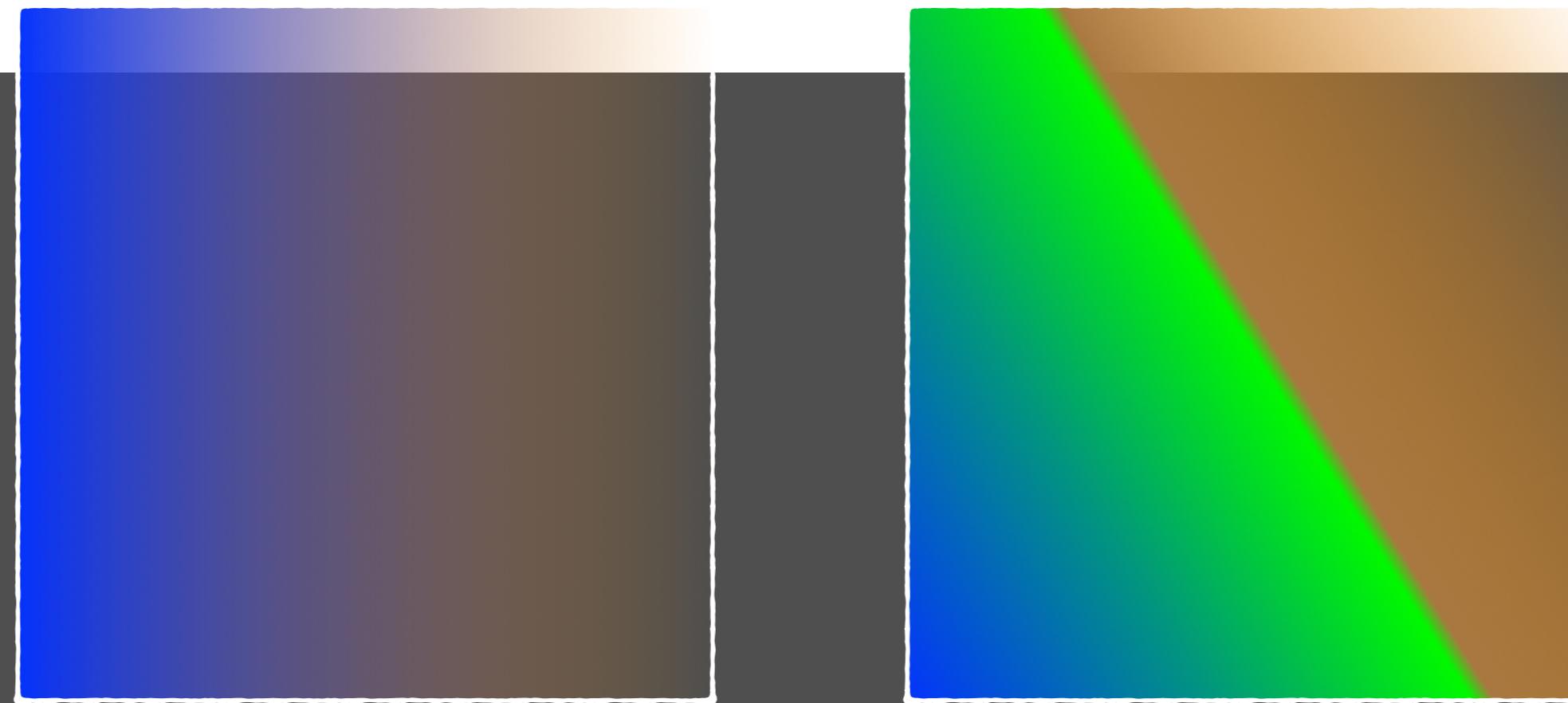
Neue Farbmodelle mit Transparenz

Für Farben kann nun eine Deckkraft eingestellt werden. Der vierte Wert steht für einen Alpha-Kanal. Im Gegensatz zu Opacity, welches jeweils für das ganze Element auswirkt, können mit den Alpha Farben Fläche, Rand oder Text einzeln eingestellt werden.

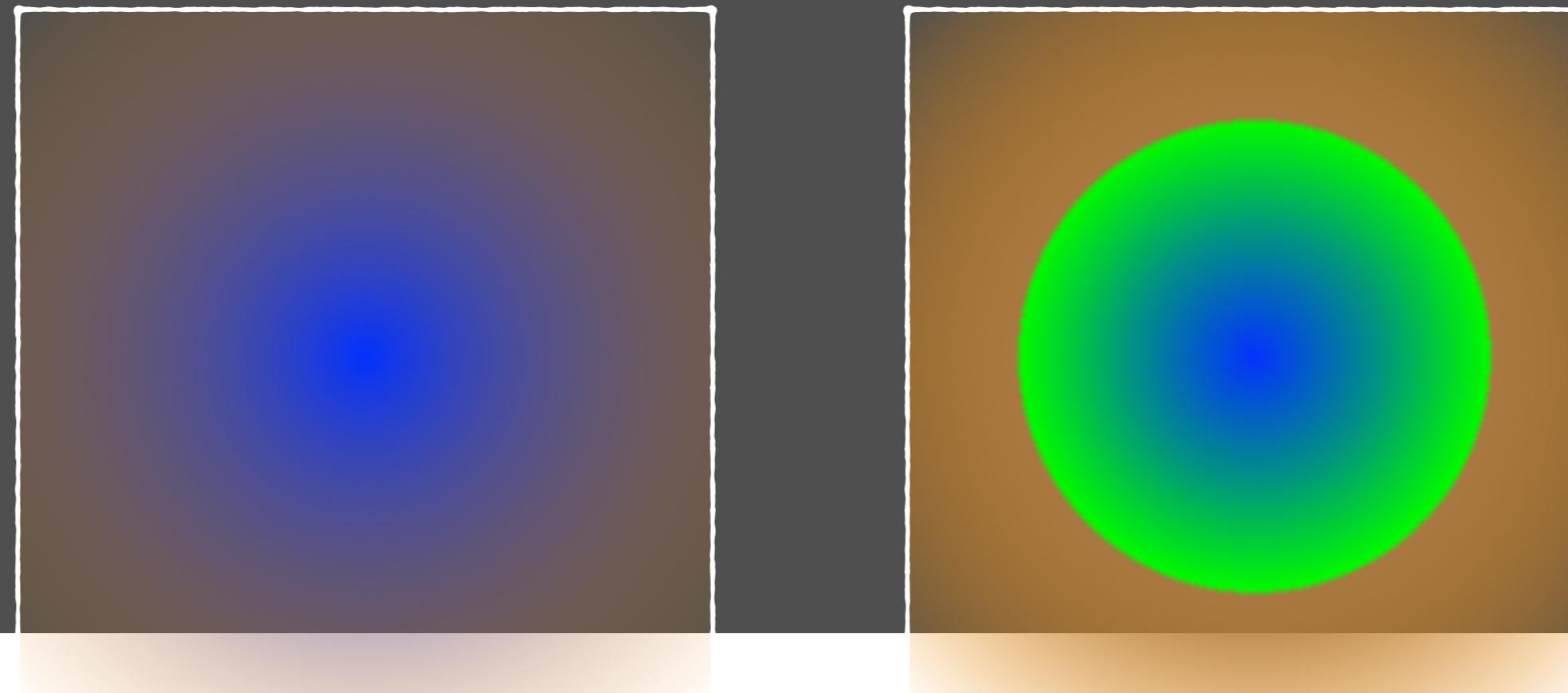
```
color: rgba(255, 0, 0, 0.75);  
background-color: hsla(360, 100%, 50%, 0.75);
```

Farbverläufe - linear und radial

```
background:  
linear-gradient(  
    32deg,  
    blue 0%,  
    green 50%,  
    brown 51%,  
    transparent 100%  
) ;
```



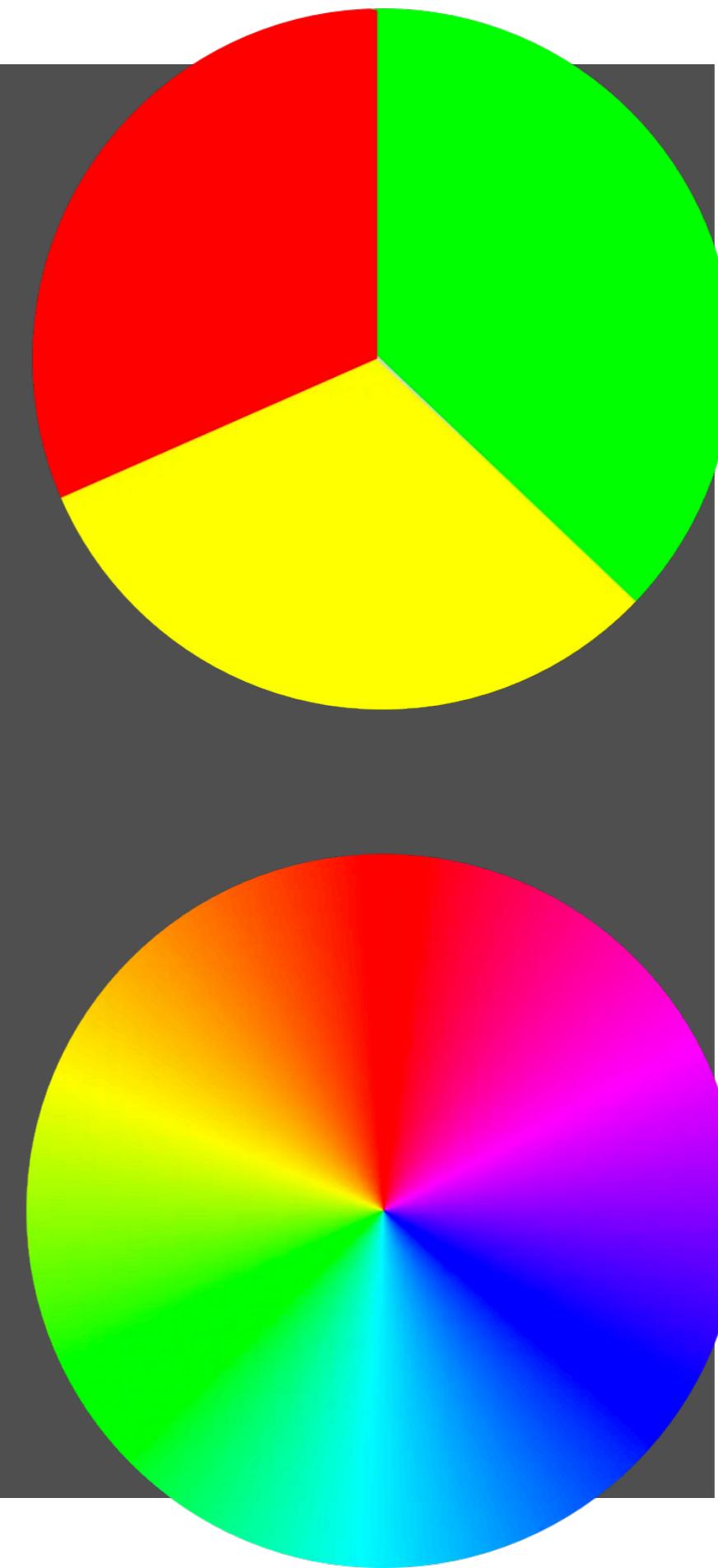
```
background:  
radial-gradient(  
    ellipse at center,  
    blue 0%,  
    green 50%,  
    brown 51%,  
    transparent 100%  
) ;
```



Konischer Farbverlauf

```
.conic-gradient {  
  background:  
    conic-gradient(  
      lime 40%,  
      yellow 0 70%,  
      red 0);  
}
```

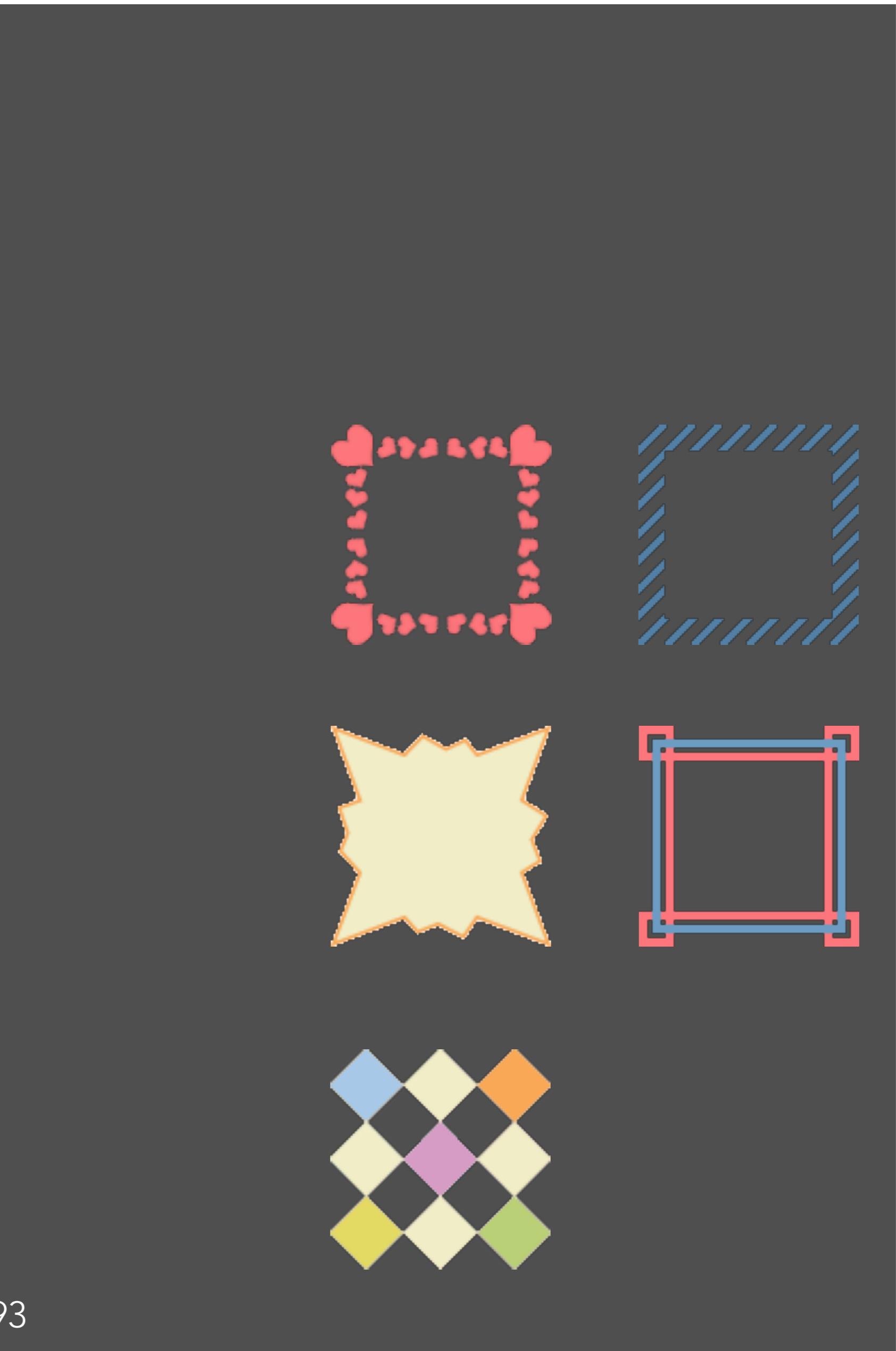
```
.conic-gradient {  
  background:  
    conic-gradient(  
      red,  
      yellow,  
      lime,  
      aqua,  
      blue,  
      magenta,  
      red);  
}
```



<https://css-tricks.com/css3-gradients/>

Unbedingt lesen!

CSS Border image



Abgerundete Ecken für alle Browser

```
border-radius: 5px;  
border-left-top-radius:10px;  
  
-moz-border-radius : 5px;  
-o-border-radius : 5px;  
-webkit-border-radius : 5px;  
-khtml-border-radius : 5px;  
-ms-border-radius : 5px;  
border-radius: 5px;
```

Für alte IE's
border-radius.htc
(Google Coderepository)

Schatten

```
text-shadow:  
    10px /* X Versatz */  
    10px /* Y Versatz */  
    10px /* Blurradius*/  
    rgba(64, 64, 64,  
0.5)  
;  
  
box-shadow:  
    10px /* X Versatz */  
    10px /* Y Versatz */  
    10px /* Blurradius*/  
    rgba(0, 0, 128,  
0.25);
```



Das ist Text mit
Schlagschatten

Reflektion

```
box-reflect:  
below | above | left | right  
5px;  
  

```



Filter Funktionen

```
filter : blur(5px)
         brightness(0.2)
         saturate(50%)
         hue-rotate(30deg)
         contrast(125%)
         invert(100%)
         greyscale(100%)
         sepia(100%)

;

// Ggf. müssen Browser-Prefixes verwendet werden:
// -moz-filter, -webkit-filter, -o-filter, -ms-
filter
```

<https://codepen.io/grayghostvisuals/pen/Cacib>



<https://css-tricks.com/almanac/properties/f/filter/>

BackgroundMischmodus

```
.blended {  
    background-image : url(face.jpg);  
    background-color : red;  
    background-blend-mode : multiply;  
}  
  
// Weitere Modus sind : normal , multiply, screen,  
// overlay,  
// darken, lighten, color-dodge, saturation, color,  
// luminosity
```

ElementMischmodus

```
// Für Elemente gilt die Eigenschaft  
mix-blend-mode : normal;
```

Die Modus sind ähnlich: multiply, screen, overlay,
darken,
lighten, color-dodge ...

<https://css-tricks.com/basics-css-blend-modes/>

Filter Funktionen

```
.element {  
  backdrop-filter: blur(5px) contrast(0.8);  
}
```

Properties:

- blur()
- brightness()
- contrast()
- drop-shadow()
- grayscale()
- hue-rotate()
- invert()
- opacity()
- saturate()
- sepia()
- url()



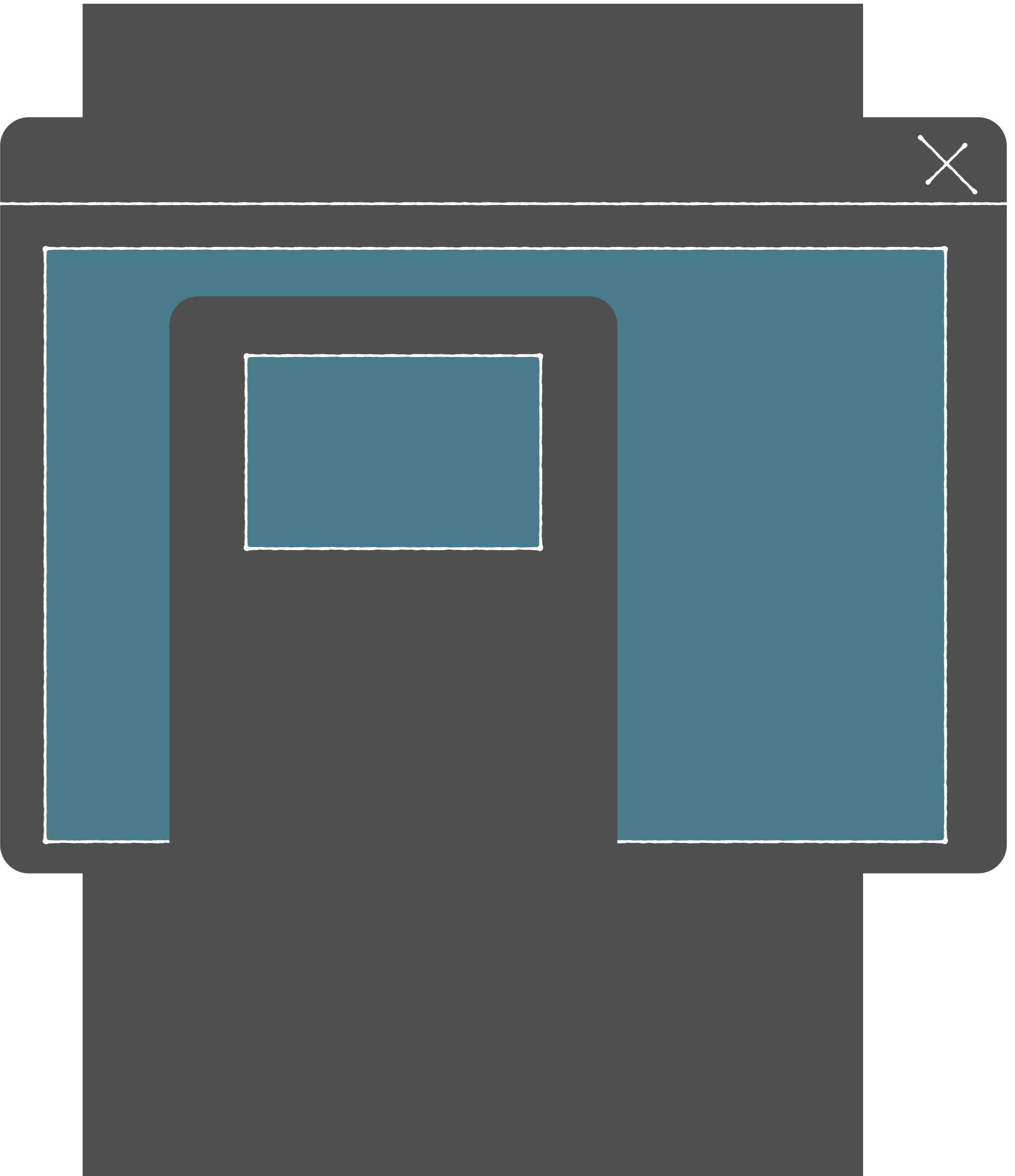
<https://css-tricks.com/the-backdrop-filter-css-property/>

Responsive Bilder

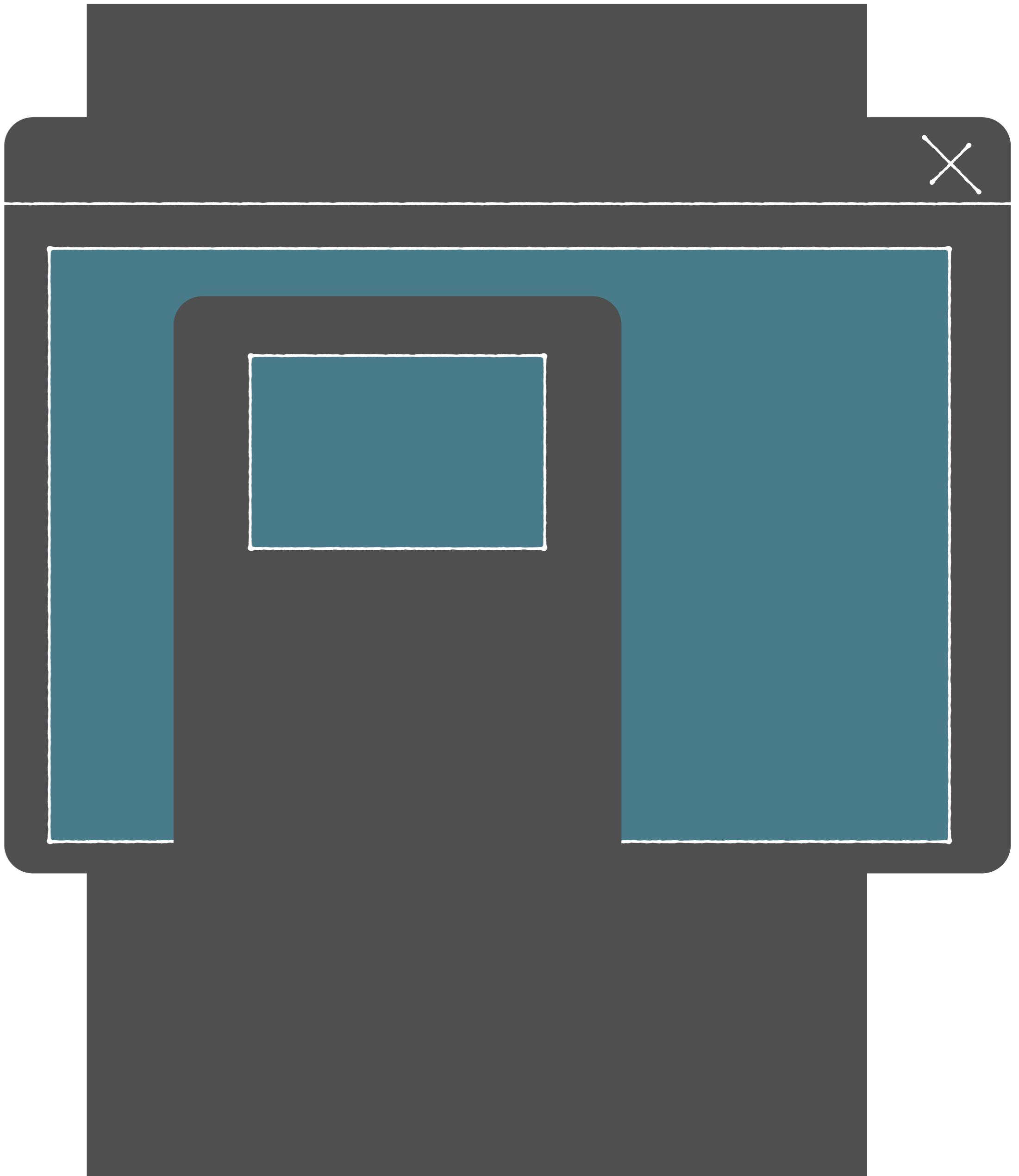
2021 Michael Reichart

Bilder responsive bemassen

```
.portrait {  
    height : auto;  
    max-width : 100%  
}
```

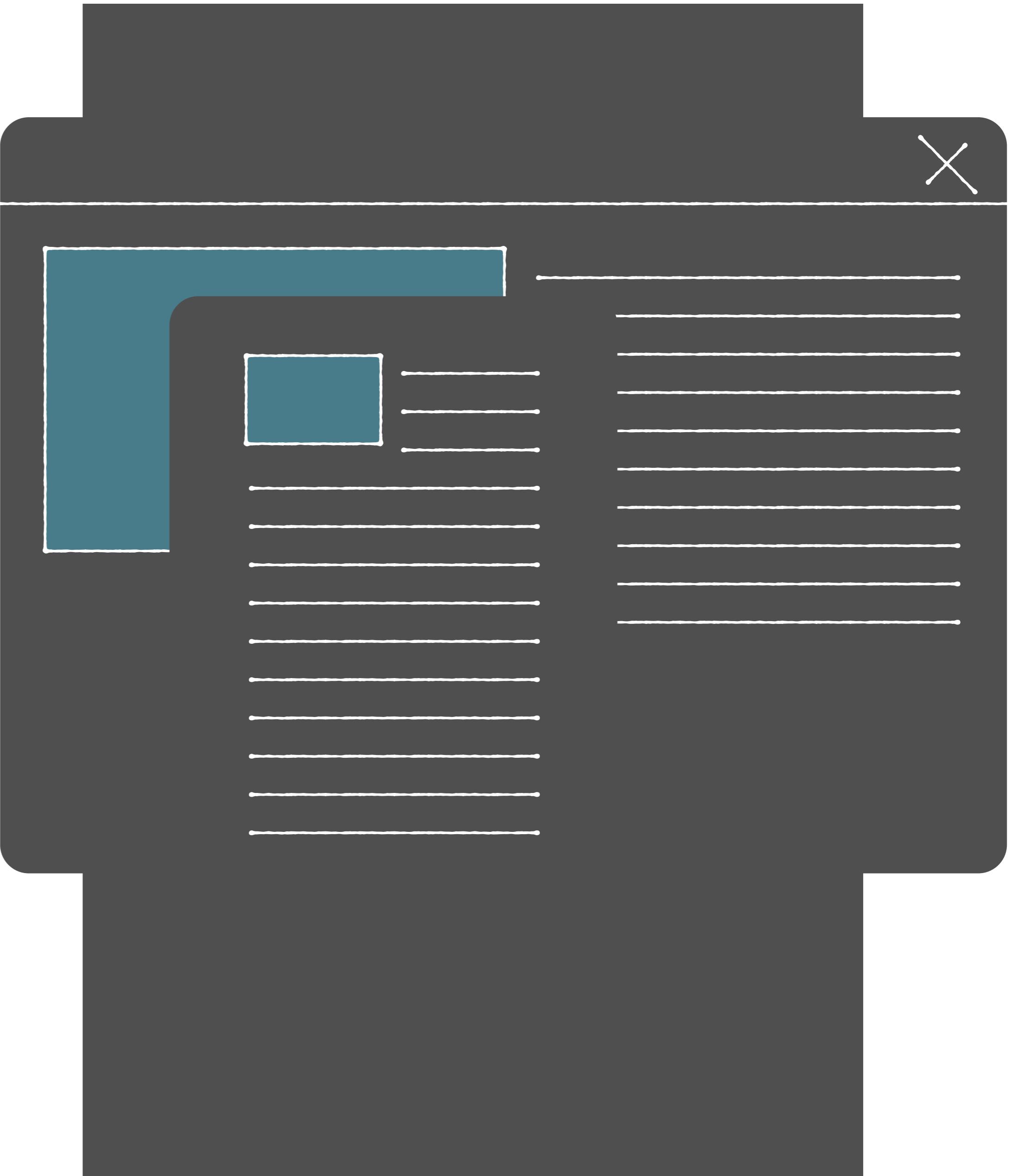


Das geht auch mit
embed, object, video,
canvas!



Bilder responsive bemassen

```
.portrait {  
    float: left;  
    height: auto;  
    max-width: 40%;  
    margin: 0 5% 2.5%  
}
```



Auflösungen beachten - Die Grenzen beim Skalieren von Bildern

- + Bilder lassen sich allerdings weder vergrößern, noch beliebig verkleinern. Ein Pixel ist ein Pixel, und wird davon nur die Hälfte gezeigt, geht Qualität verloren.

Das <picture> Element

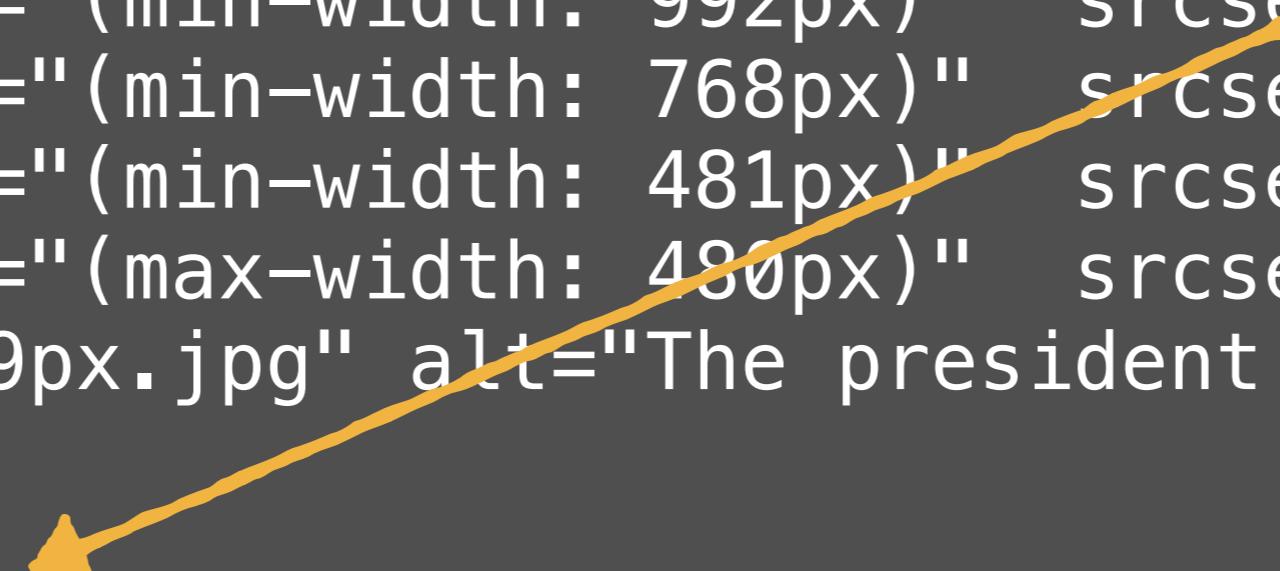
Für die Platzierung von Bildern wird ein neues <picture>-Element eingeführt. Damit können auflösungsabhängig verschiedene Bildformate bereitgestellt werden. Dabei müssen die Bilder vom Größten zum Kleinsten hin eingebunden werden.

```
<picture>
  <source media="(min-width: 1200px)" srcset="1440px.jpg">
  <source media="(min-width: 992px)" srcset="1199px.jpg">
  <source media="(min-width: 768px)" srcset="991px.jpg" >
  <source media="(min-width: 481px)" srcset="767px.jpg" >
  <source media="(max-width: 480px)" srcset="480px.jpg" >
  
</picture>
```

Das srcset Attribut

Die srcset Attribute halten verschiedenen Bildquellen bereit, von denen eine nach der Auswertung des Browsers ausgewählt wird und anstelle der src Angabe im img Element verwendet wird.

```
<picture>
  <source media="(min-width: 1200px)" srcset="1440px.jpg">
  <source media="(min-width: 992px)" srcset="1199px.jpg">
  <source media="(min-width: 768px)" srcset="991px.jpg" >
  <source media="(min-width: 481px)" srcset="767px.jpg" >
  <source media="(max-width: 480px)" srcset="480px.jpg" >
  
</picture>
```



Das Fallback Bild

Trifft keine der srcset Bilder zu oder kann ein Browser das picture/srcset nicht auswerten, dann wird das img Element als Fallback verwendet.

```
<picture>
  <source media="(min-width: 1200px)" srcset="1440px.jpg">
  <source media="(min-width: 992px)" srcset="1199px.jpg">
  <source media="(min-width: 768px)" srcset="991px.jpg" >
  <source media="(min-width: 481px)" srcset="767px.jpg" >
  <source media="(max-width: 480px)" srcset="480px.jpg" >
  
</picture>
```

<picture>

Ausser der gerätespezifischen Einstellung kann das <picture> Element auch zur Auswahl aus mehreren alternativen Bildformaten verwendet werden.

```
<picture>
  <source srcset="/uploads/100-marie-lloyd.webp"
          type="image/webp">
  <source srcset="/uploads/100-marie-lloyd.jxr"
          type="image/vnd.ms-photo">

  
</picture>
```

Retina Displays für <picture>

```
<picture>

    <source media="(min-width: 2400px),
                  (-webkit-min-device-pixel-ratio: 2),
                  (min-resolution: 192dpi)" srcset="2880px.jpg">

    <source media="(min-width: 1200px)" srcset="1440px.jpg">
    <source media="(min-width: 992px)" srcset="1199px.jpg">
    <source media="(min-width: 768px)" srcset="991px.jpg" >
    <source media="(min-width: 481px)" srcset="767px.jpg" >
    <source media="(max-width: 480px)" srcset="480px.jpg" >
    
</picture>

@media
(-webkit-min-device-pixel-ratio: 2),
(min-resolution: 192dpi) {
    /* Retina-specific stuff here */
}
```

srcset-w

Ebenfalls für auflösungsabhängig Bildformate kann das srcset-Attribut verwendet werden. Im Gegensatz zur picture Gruppe wertet das srcset-Attribut browserseitig und selbstständig aus, welche Bilddatei unter gegebenen Bedingungen die beste ist.

srcset-w:

```

```

srcset-x

Mit der x-Angabe können Bilder Displayauflösungsabhängig hinterlegt werden. So lassen sich Retina-fähige Bilder hinterlegen.

```
srcset-x:  

```

[https://css-tricks.com/
responsive-images-youre-just-changing-
resolutions-use-srcset/](https://css-tricks.com/responsive-images-youre-just-changing-resolutions-use-srcset/)

[https://www.smashingmagazine.com/
2014/05/responsive-images-done-right-
guide-picture-srcset/](https://www.smashingmagazine.com/2014/05/responsive-images-done-right-guide-picture-srcset/)

[https://css-tricks.com/snippets/css/
retina-display-media-query/](https://css-tricks.com/snippets/css/retina-display-media-query/)

Picturefill

A responsive image polyfill

- + The picture element and associated features are W3C standard HTML features that allow web developers to deliver an appropriate image to every user depending on a variety of conditions like screen size, viewport size, screen resolution, and more.
- + Picturefill is a JavaScript file (or a polyfill to be more specific) that enables support for the picture element and associated features in browsers that do not yet support them, so you can start using them today!

<http://scottjehl.github.io/picturefill/>

Serverside: adaptive-images.php

- + Adaptive Images detects your visitor's screen size and automatically creates, caches, and delivers device appropriate re-scaled versions of your web page's embedded HTML images. No mark-up changes needed. It is intended for use with Responsive Designs and to be combined with Fluid Image techniques.

<http://www.adaptive-images.com>



Focal Point: Intelligent

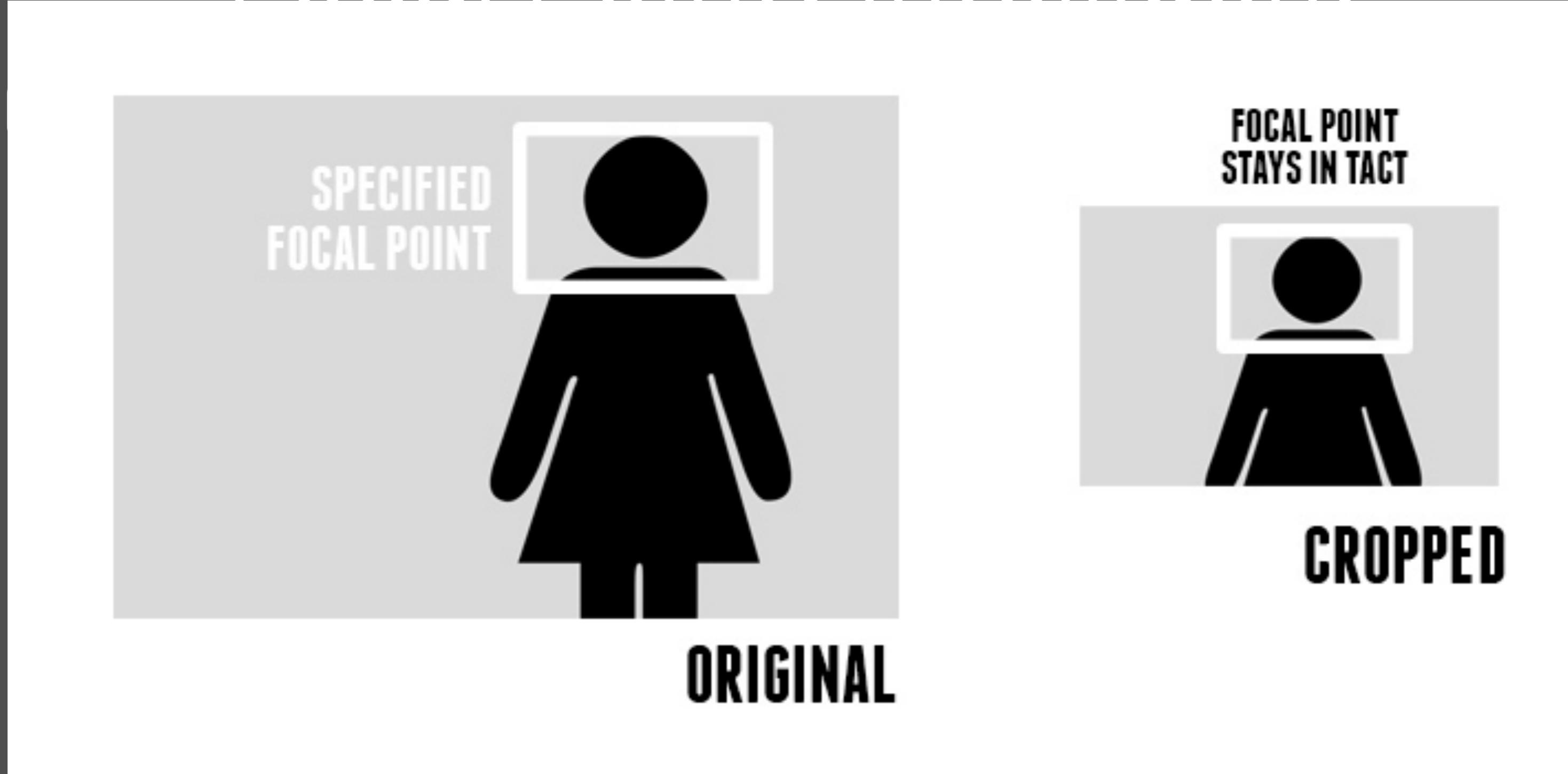


ORIGINAL



CROPPED

Focal Point: Intelligent





Focal Point: Intelligent Cropping of Responsive Images

```
<div class="focal-point right-3 up-3">
  <div></div>
</div>
```

Focal Point: Intelligent Cropping of Responsive Images

```
.focal-point {  
    width      : 100%;  
    height     : auto;  
    overflow   : hidden;  
}  
.focal-point img {  
    width      : 100%;  
    max-width  : 100%;  
    height     : auto;  
    -ms-interpolation-mode : bicubic;  
}  
.focal-point div {  
    position   : relative;  
    max-width  : none;  
    height     : auto;  
}
```

Focal Point: Intelligent Cropping of Responsive Images

```
@media all and (max-width: 767px) {  
    /* 4x3 Landscape Shape (Default) */  
    .focal-point div {  
        margin          : -3em -4em;  
    }  
    /* Landscape up (Total 6em) */  
    .up-3 div {  
        margin-top     : -1.5em;  
        margin-bottom  : -4.5em;  
    }  
    .right-3 div {  
        margin-left   : -6em;  
        margin-right  : -2em;  
    }  
}
```

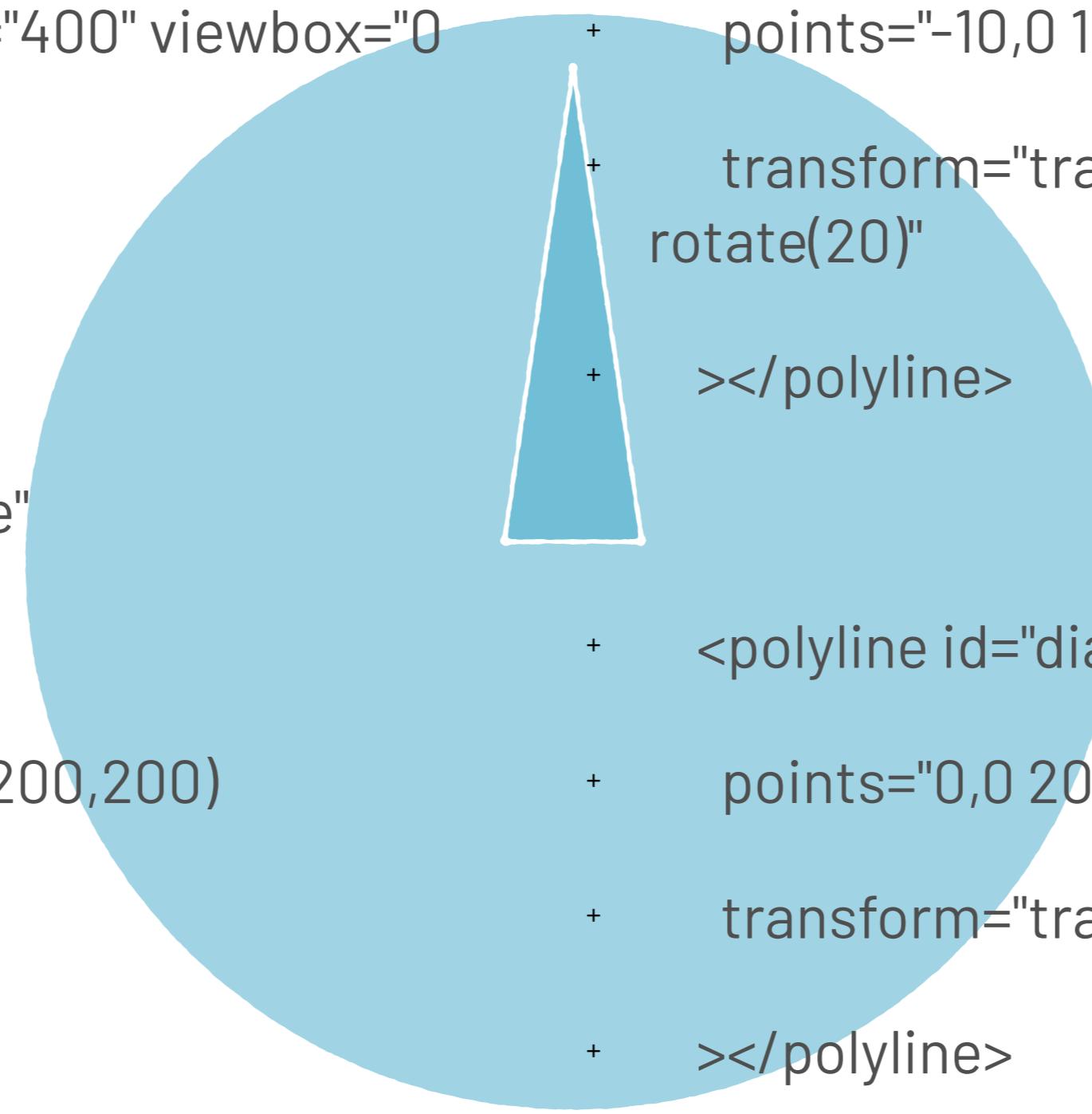
<http://designshack.net/articles/css/focal-point-intelligent-cropping-of-responsive-images/>

Scalable Vector Graphics

SVG - Scalable Vector Graphics

- + SVG Graphiken sind Vektorgraphiken auf XML-Basis.
- + SVG kann beliebig in HTML5 Dokumente eingebunden werden.
- + Sie bilden eine DOM - Struktur und können mit CSS gestaltet und mit Javascript programmiert werden.

```
+ <svg width="400" height="400" viewBox="0  
0 400 400">  
+   <g id="layer">  
+     <circle  
+       id="circle" class="circle"  
+       cx="0" cy="0" r="100"  
+       transform="translate(200,200)  
rotate(0)"  
+     ></circle>  
+     <polyline id="needle" class="needle"  
+       points="-10,0 10,0 0,80 -10,0 10,0"  
+       transform="translate(200,200)  
rotate(20)"  
+     ></polyline>  
+   </g>  
+ </svg>
```



Kreis mit Dreieck

```
<svg width="400" height="400" viewBox="0 0 400 400">
  <g id="layer" transform="translate(200,200)
rotate(30)">
    <circle
      id="circle" class="circle"
      cx="0" cy="0" r="100"
      transform="rotate(0)"
    ></circle>

    <polygon id="needle" class="needle"
      points="-10,0 0,80 10,0"
      transform="rotate(20)"
    ></polygon>
  </g>
</svg>
```

SVG CSS

```
/* MODULE */
.circle,
.needle,
.diagram {
    fill      : transparent;
    fill-rule : evenodd;

    stroke     : white;
    stroke-width : 5px;
    stroke-linecap : button;
    stroke-linejoin : round;

    transition-property : all;
    transition-duration : 2s;
    transition-timing-function : ease-in-out;

}

/* STATE */
.default { fill : rgba(0,200,0,1); }
.warning { fill : rgba(200,0,0,1); }
```

Responsives Javascript

Breakpoints mit Javascript

```
mediaTiny      = window.matchMedia('max-width:  
320px'),  
mediaXtrasmall = window.matchMedia('max-width:  
479px'),  
mediaSmall     = window.matchMedia('max-width:  
767px'),  
mediaMedium    = window.matchMedia('min-width:  
768px'),
```

Laden bei Größenänderung des Viewports

```
mediaTiny.onchange = function () {  
    console.log('tiny changed!');  
};
```

Inhalte beim Öffnen einer Seite laden

```
if ( mediaMedium.matches === true ) {  
    loadVideothek();  
    unloadButton();  
} else {  
    unloadVideothek();  
    loadButton();  
}
```

Javascript und Geräteerkennung

Browser- und Geräteinformationen abfragen

```
console.log( window.navigator.userAgent );
console.log( window.navigator.vendor );
```

Mobile-detect.js

mobile-detect.js

```
<script src="mobile-detect.js"></script>
<script>

    var md = new
MobileDetect(window.navigator.userAgent);

</script>
```

mobile-detect.js

```
var md = new MobileDetect(  
    'Mozilla/5.0 (Linux; U; Android 4.0.3; en-in;  
SonyEricssonMT11i' +  
    ' Build/4.1.A.0.562) AppleWebKit/534.30 (KHTML,  
like Gecko)' +  
    ' Version/4.0 Mobile Safari/534.30');
```

mobile-detect.js

```
console.log( md.mobile() );           // 'Sony'  
console.log( md.phone() );           // 'Sony'  
console.log( md.tablet() );          // null  
console.log( md.userAgent() );        // 'Safari'  
console.log( md.os() );              // 'AndroidOS'  
console.log( md.is('iPhone') );       // false  
console.log( md.is('bot') );          // false  
console.log( md.version('Webkit') );   //  
534.3  
console.log( md.versionStr('Build') );  //  
'4.1.A.0.562'  
console.log( md.match('playstation|xbox') ); //  
false
```

<https://github.com/hgoebl/mobile-detect.js/>

-Heinrich Goebel

Responsive Tabellen

Tabellen in einen Scrollcontainer packen

```
<div class="table table-responsive">
<table>
  <thead>
    <tr>
      <th>Bibendum Sem Mollis</th>
      <th>Bibendum Sem Mollis</th>
    </tr>
  </thead>
  <tfoot>...<tfoot>
  <tbody>
    <td>Nullam quis risus eget urna mollis leo.</td>
    <td>Nullam quis risus eget urna mollis leo.</td>
  <tbody>
</table>
</div>
```

Tabellen in einen Scrollcontainer packen

```
.table-responsive {  
    max-width: 100%;  
    overflow: auto;  
}  
.table-responsive td {  
    width : 100%;  
}
```

Tabellen als „Listen“ darstellen

```
.table-responsive table {  
    display: block; /* ersetzt table */  
}  
  
.table-responsive thead {  
    display: none;  
}  
  
.table-responsive td {  
    display: block; /* ersetzt table-cell */  
}  
  
.table-responsive td::before {  
    content: attr(data-head); /* liest ein data Attribut aus */  
    display: block;  
    font-weight: 600;  
    margin-bottom: 0.5rem;  
}
```

Skalierbares, modulares CSS

Stylesheets strukturiert aufbauen

- + CSS können jedes Element und jede Eigenschaft einer Weboberfläche ansprechen und gestalten.
- + - ist eine deklarative Sprache
- + - keine Struktur-Befehle
- + - die Struktur muss konzeptionell erreicht werden.

kategorisieren!

- + Die Aufgaben von Stylesheets können in immer gleiche Kategorien eingeteilt werden:
 - + - Projekt oder Kundenanpassungen
 - + - Grundsätzliches, Semantisches
 - + - Layout und Anordnung
 - + - Zustandsveränderungen

kategorisieren!

base.css

layout.css

module-1.css

module-2.css

theme.css

Der interne Aufbau einer Deklaration

`/* BASE */`

Alles, was grundsätzlich in einem Modul oder einem Element gelten soll.

`/* LAYOUT */`

Anordnungsvarianten, zum Beispiel vertikale oder horizontale Anordnungen.

`/* STATES */`

Zustandsveränderungen bei Mausbewegung, aktivem Cursor et.

base.css

```
html,  
html * { box-sizing : border-box; }  
  
/* FARBEN */  
/* Hue, Saturation, Lightness (Luminance) */  
html {  
    background-color : hsla(360, 0%, 80%, 1);  
    color          : hsla(180, 0%, 20%, 1);  
}  
a[href] {  
    color : hsla(210, 70%, 30%, 1);  
}  
  
/* TYPOGRAPHY */  
html {  
    font-family : sans-serif;  
    font-size   : 14px;  
    line-height : 1.428571; // entspricht 20px  
}  
p,  
ul,ol,li,  
td,th,  
code { font-size : 1.0rem; margin-top: 1rem; margin-bottom: 1rem; }  
h1 { font-size : 2.0rem; margin-top: 1rem; margin-bottom: 1rem; }  
h2 { font-size : 1.8rem; margin-top: 1rem; margin-bottom: 1rem; }  
h3 { font-size : 1.6rem; margin-top: 1rem; margin-bottom: 1rem; }  
h4 { font-size : 1.4rem; margin-top: 1rem; margin-bottom: 1rem; }  
h5 { font-size : 1.2rem; margin-top: 1rem; margin-bottom: 1rem; }  
h6 { font-size : 1.0rem; margin-top: 1rem; margin-bottom: 1rem; }
```

template.css

```
body {  
    display      : flex;  
    justify-content : space-around;  
}  
  
.page        { display : block; width : 960px; }  
.page-header  { display : block; width : 100%; }  
.page-content {  
    display      : flex;  
    width       : 100%;  
    flex-direction : row;  
    flex-wrap   : wrap;  
    justify-content : center;  
    align-items  : stretch;  
    align-content : flex-start;  
}  
.page-footer  { display : block; width : 100%; }  
  
.content-nav {  
    display      : block;  
    order       : 1;  
    flex-basis  : 25%;  
    flex-grow   : 1;  
    flex-shrink : 0;  
}  
.content-main {  
    display      : block;  
    order       : 2;  
    flex-basis  : 50%;  
    flex-grow   : 1;  
    flex-shrink : 0;  
}  
.content-aside {  
    display      : block;  
    order       : 3;  
    flex-basis  : 25%;  
    flex-grow   : 1;  
}
```

navigation.css

```
/* BASE */
.nav {
  list-style : none;
  margin     : 0;
  padding    : 0;
}

.nav h4 {
  font-size   : 1em;
  font-weight : 300;
}

.nav li a[href] {
  display : inline-block;
  width   : 100%;
  height  : 100%;
}

.nav {
  margin-left   : -1rem;
  margin-right  : -1rem;
}
.nav li {
  margin-top    : 0;
  margin-bottom : 0;
}

/* LAYOUT */
.nav-vertical li {
  display : block;
}
.nav-horizontal li {
  display : inline-block;
}
.nav-left li {
  display : block;
  float   : left;
```

<http://smacss.com>

Hier hat Jonathan Snook Regeln und Beispiele für eine skalierbare und modulare Vorgehensweise beschrieben und mit Beispielen anschaulich illustriert.



Wie kann eine CSS für die Seitennavigation aufgebaut werden?
Wie wird sie im HTML angewendet?

Entwickle in einer navigation.css :

```
/* base */  
.nav { ... }  
  
/* layout */  
.nav-horizontal li { ... }  
.nav-vertical li { ... }  
  
/* states */  
.nav a[href] { ... }  
.nav a[href:visited] { ... }  
.nav a[href:active] { ... }  
.nav a[href:hover] { ... }
```

Generisches CSS

Stylesheets generisch

- + Mit SASS (Syntactically Awesome Stylesheets) können Stylesheets "programmiert" werden.
- + Variablen halten wiederverwendbare Werte für Farben, Schriften, Breiten und mehr.
- + "Mixins" beinhalten wiederverwendbare Deklarationen und aufrufbare Funktionen.
- + Ein Compiler generiert aus den SASS-Codes CSS-Zeilen und legt diese in einem Verzeichnis ab.

_variables.scss

```
// LAYOUT
$default-padding: 0.5rem;
$default-margin: 0;

// FONTS
$font-size: 22px;
$line-height: 1.8rem;

// COLORS
$text-hue: 0;
$text-saturation: 0;
$text-lightness: 10%;
$text-alpha: 1;
$color-hue: 210;
$color-saturation: 80%;
$color-lightness: 50%;
$color-alpha: 1;
```

_mixins.scss

```
@mixin border-radius($radius) {  
  -webkit-border-radius: $radius;  
  -moz-border-radius: $radius;  
  -ms-border-radius: $radius;  
  border-radius: $radius;  
}  
  
.box { @include border-radius(10px); }
```

Imports

```
// _reset.scss
html, body, ul, ol {
  margin: 0;
  padding: 0;
}

// base.scss
@import 'reset';

body {
  font: 100% Helvetica, sans-serif;
  background-color: #efefef;
}
```

Mit Einrückungen arbeiten

```
nav {  
    ul {  
        margin: 0;  
        padding: 0;  
        list-style: none;  
    }  
  
    li {display:inline-block;}  
  
    a {  
        display: block;  
        padding: 6px 12px;  
        text-decoration: none;  
    }  
}  
nav ul {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
}  
  
nav li {  
    display: inline-block;  
}  
  
nav a {  
    display: block;  
    padding: 6px 12px;  
    text-decoration: none;  
}
```

CSS Prozess

von rechts nach links!

#my-id .my-class a[href]:active { ... }

1: [href] → [elem, elem, elem, elem, elem,
elem, ...]

2: a → [elem, elem, elem, elem, elem,
elem, ...]

3: my-class → [elem, elem, elem, elem]

4: my-id → [elem, elem]

=> [elem]

```
$ cd path/to/web/directory  
$ sass --watch assets/scss:assets/css  
  
--style=expanded | compact | compressed | nested  
--sourcemap=None
```