

O'REILLY®

5. Auflage
Behandelt CSS3

CSS

kurz & gut

O'REILLYS TASCHENBIBLIOTHEK



Eric A. Meyer

Übersetzung von Jørgen W. Lang

Inhalt

Cover

Titel

Impressum

Inhalt

Vorwort

1 Grundkonzepte

HTML mit Stildefinitionen versehen

Struktur einer CSS-Regel

@-Regeln

Kommentare

Rangfolge der Stildefinitionen (Präzedenz)

Klassifizierung der Elemente

Darstellungsrollen

Grundlagen der visuellen Darstellung

Floating

Positionierung

Flexbox

Grid-Layout (CSS-Layoutraster)

Tabellenlayout

2 Werte

Schlüsselwörter

Farbwerte

Numerische Werte

Prozentwerte

Länge

Teilwerteinheiten (fraction values)

URIs

Winkel

Zeitangaben

Frequenzen

Position

Strings

Identifizier

Attributwerte

Berechnungswerte

CSS-Variablen

3 Selektoren und Queries

Selektoren

Strukturelle Pseudoklassen

Die Negations-Pseudoklasse

Interaktions-Pseudoklassen

Pseudoelemente

Media-Queries (Medienabfragen)

Feature-Queries

4 Eigenschaften-Referenz

Vererbung und Animation

Typografische Konventionen für Werte

Universelle Werte

Eigenschaften

Index

Über den Autor

Eigenschaften-Referenz

Vererbung und Animation

Für alle in diesem Kapitel aufgeführten Eigenschaften wird angegeben, ob sie vererbbar (»Vererbt«) und ob sie animierbar (»Anim.«) sind. Die Werte »N« (für nein) und »J« (für ja) geben an, ob eine Eigenschaft an Nachfahrenelemente *vererbt* wird und ob die Eigenschaft *animierbar* ist bzw. von den verschiedenen Eigenschaften für Animationen und Übergänge (Transitions) beeinflusst wird. Können nur einige Werte der Eigenschaft animiert werden, lautet der Wert »T« (für teilweise). Weitere Details finden Sie gegebenenfalls in der Definition der Eigenschaft.

Typografische Konventionen für Werte

Alle Wörter, die in `Festbreitenschrift` stehen, sind Schlüsselwörter, die literal ohne Anführungszeichen verwendet werden müssen. Der Schrägstrich (/) und das Komma (,) müssen ebenfalls literal (»wörtlich«) verwendet werden.

Kursiv geschriebene Wörter zwischen »<<« und »>>« bezeichnen einen Wertetypen oder eine Referenz auf den Wert einer anderen Eigenschaft. So akzeptiert die Eigenschaft `font` beispielsweise Werte, die ursprünglich zur Eigenschaft `font-family` gehören. Das wird durch den Text `<font-family>` ausgedrückt. Ist ein Wertetyp z. B. eine Farbe, wird dies durch `<Farbe>` angegeben.

Es gibt verschiedene Möglichkeiten, die Teile einer Wertedefinition miteinander zu kombinieren:

- Zwei Schlüsselwörter, die nur durch ein Leerzeichen getrennt sind, müssen in exakt der angegebenen Reihenfolge verwendet werden.

- Ein vertikaler Balken (X | Y) trennt Alternativen voneinander. Eine der Alternativen *muss* verwendet werden. Es darf nur eine der Alternativen verwendet werden.
- Ein doppelter vertikaler Balken (X || Y) bedeutet, dass entweder X *oder* Y *oder* beide Werte verwendet werden *müssen*. Die Reihenfolge ist beliebig.
- Ein doppeltes Ampersand-Zeichen (X && Y) bedeutet, dass X *und* Y verwendet werden *müssen*, aber in beliebiger Reihenfolge erscheinen dürfen.
- Eckige Klammern ([...]) werden zum Gruppieren von Dingen verwendet. »[b i t t e || h i l f || m i r] mach das« bedeutet demnach, dass mindestens eines der Wörter b i t t e, h i l f und m i r (in beliebiger Reihenfolge und höchstens einmal) verwendet werden muss. mach das muss immer verwendet werden mit exakt diesen Wörtern in exakt dieser Reihenfolge.

Auf jeden Bestandteil oder jede Gruppe in eckigen Klammern kann (muss aber nicht) einer der folgenden Modifier folgen:

- Ein Asterisk (*) besagt, dass der vorangehende Wert/die vorangehende Gruppe in eckigen Klammern null oder mehrmals wiederholt wird.
- Ein Pluszeichen (+) besagt, dass der vorangehende Wert/die vorangehende Gruppe in eckigen Klammern einmal oder mehrmals wiederholt wird.
- Eine Raute (#) besagt, dass der vorangehende Wert/die vorangehende Gruppe in eckigen Klammern einmal oder mehrmals wiederholt wird, gegebenenfalls durch Kommata getrennt.
- Ein Fragezeichen (?) besagt, dass der vorangehende Wert/die vorangehende Gruppe in eckigen Klammern optional ist.
- Ein Ausrufungszeichen (!) besagt, dass der vorangehende Wert/die vorangehende Gruppe in eckigen Klammern zwingend erforderlich ist. Das Ergebnis muss also mindestens ein Wert sein, auch wenn die Syntax etwas anderes vorzugeben scheint.
- Ein Zahlenpaar in geschweiften Klammern ({M,N}) bedeutet, dass der vorangehende Wert/die vorangehende Gruppe in eckigen Klammern mindestens M und maximal N Mal wiederholt wird.

Universelle Werte

Jedes Benutzerprogramm (User Agent, Browser), das das Modul »Cascading and Inheritance« vollständig implementiert hat, akzeptiert die folgenden Werte für alle Eigenschaften. Die Wertesyntax einer Eigenschaft könnte in etwa so aussehen:

```
[ (Notierte Wertesyntax) ] | inherit | initial | unset
```

Die drei oben genannten Schlüsselwörter werden in den folgenden Eigenschaften-Definitionen aus Gründen der Klarheit nicht explizit aufgeführt. Die Ausnahme bildet die Eigenschaft `all`, die *nur* diese drei Schlüsselwörter als Werte akzeptiert (siehe Kapitel 2).

Eigenschaften

align-content

Vererbt: N Anim.: N

Werte

`flex-start` | `flex-end` | `center` | `space-between` | `space-around` | `space-evenly` | `stretch`

Startwert

`stretch`

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Flex-Container.

Beschreibung

Definiert die Verteilung von Flex-Zeilen entlang der Querachse (Cross Axis) eines Flex-Containers. Hierbei darf die Länge der Querachse nicht gleich der Summe der Größen der Flex-Zeilen auf der gleichen Achse sein.

Beispiele

```
aside {display: flex; align-content: center;}
```

```
section {display: flex; height: 90vh;
```

```
    align-content: flex-end;}
```

Hinweis

Anfang 2018 war geplant, dass die Eigenschaft nicht nur auf Flex-Container, sondern auch auf viele andere (oder alle) Elemente anwendbar sein sollte. Die Werte `start` und `end` sollen es gegebenenfalls ermöglichen, das Verhalten von `flex-start` und `flex-end` in Nicht-Flex-Umgebungen nachzubilden. Der Wert `center` würde diese Änderung des vertikalen Zentrierens von Inhalten in fast allen Fällen stark vereinfachen.

align-items

Vererbt: N Anim.: N

Werte

`flex-start` | `flex-end` | `center` | `baseline` | `stretch`

Startwert

`stretch`

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Flex-Container, Grid-Container und mehrspaltige (multicolumn) Container.

Beschreibung

Definiert einen containerweiten Standard für die Ausrichtung von Flex-Elementen (Flex Items), bezogen auf die Querachse der Flex-Zeile, auf der die Flex-Elemente stehen. Der Wert `baseline` bedeutet, dass die Elemente einer Flex-Zeile so ausgerichtet werden, dass die Grundlinien der ersten enthaltenen Textzeilen zur Deckung kommen.

Beispiele

```
div.flexy {align-items: flex-start;}
```

```
section.gallery {align-items: baseline;}
```

Hinweis

Anfang 2018 war geplant, diese Eigenschaft für viele (oder alle) Elemente anwendbar zu machen. Anhand der Werte `start` und `end` sollte das Verhalten von `flex-start` und `flex-end` für Nicht-Flex-Umgebungen nachgebildet werden.

align-self

Vererbt: N Anim.: N

Werte

`flex-start` | `flex-end` | `center` | `baseline` | `stretch`

Startwert

`stretch`

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Flex- und Grid-Elemente.

Beschreibung

Definiert die Ausrichtung für ein einzelnes (Flex- bzw. Grid-)Element, bezogen auf die Querachse (Cross Axis) der Flex-Zeile, auf der sich das Element befindet. Der Wert `baseline` bedeutet, dass die Grundlinie der ersten Textzeile des Elements an der am tiefsten liegenden Grundlinie auf der betreffenden Flex-Zeile ausgerichtet wird.

Beispiele

```
div.flexy .midpointed {align-self: center;}
```

```
section.gallery h1 {align-self: stretch;}
```

Hinweis

Anfang 2018 war geplant, diese Eigenschaft für viele (oder alle) Elemente anwendbar zu machen. Anhand der Werte `start` und `end` sollte das Verhalten von `flex-start` und `flex-end` für Nicht-Flex-Umgebungen nachgebildet werden.

all

Vererbt: N Anim.: N

Werte

`inherit` | `initial` | `unset`

Startwert

Siehe die einzelnen Eigenschaften.

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente.

Beschreibung

Wendet den deklarierten Wert auf alle Eigenschaften *außer* `direction` und `unicode-bidi` an. Diese sind aus Gründen der Zugänglichkeit und aus historischen Gründen ausgeschlossen. Mit dieser Eigenschaft können Sie beispielsweise erzwingen, dass ein Element sämtliche Stileigenschaften auf ihre Standardwerte zurücksetzt, wodurch die Vererbung von Werten für alle unterstützten Eigenschaften blockiert wird.

Beispiele

```
*.blandin {all: inherit;}
```

```
*.embedded {all: unset;}
```

animation

Vererbt: N Anim.: N

Werte

```
[ <animation-name> || <animation-duration> || <animation-timing-function>
|| <animation-delay> || <animation-iteration-count> || <animation-direction>
|| <animation-fill-mode> || <animation-play-state> ]#
```

Startwert

0s ease 0s 1 normal none running none

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente, die Pseudoelemente `::before` und `::after`.

Beschreibung

Eine Kurzschrifteigenschaft, die alle Aspekte einer oder mehrerer durch Kommata getrennten CSS-Animationen umfasst. Die Bestandteile des Werts können in beliebiger Reihenfolge auftreten. Beachten Sie daher die Verwechslungsgefahr zwischen den Werten für die Verzögerung (`animation-delay`) und die Animationsdauer (`animation-duration`). Beim Schreiben dieses Buchs ist es am wahrscheinlichsten, dass der erste Wert die Animationsdauer und der zweite Wert die Verzögerung definiert. Dies kann allerdings nicht garantiert werden.

Beispiele

```
div#slide {animation: 'slide' 2.5s linear
```

```
0 1 normal;}
```

```
h1 {animation: 'bounce' 0.5s 0.33s ease-in-out
```

```
infinite alternate;}
```

animation-delay

Vererbt: N Anim.: N

Werte

<Zeit>#

Startwert

0s

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente, die Pseudoelemente `::before` und `::after`.

Beschreibung

Definiert, wie lange das Benutzerprogramm wartet, bevor die CSS-Animation gestartet wird. Der Timer startet, sobald das Benutzerprogramm den CSS-Code für die Animation anwendet. Bei nicht interaktiven Elementen ist dies sehr wahrscheinlich (aber nicht garantiert) der Zeitpunkt, zu dem die Seite vollständig geladen ist.

Beispiele

```
body {animation-delay: 1s, 2000ms, 4s;}
```

```
a:hover {animation-delay: 400ms;}
```

animation-direction

Vererbt: N Anim.: N

Werte

[normal|reverse|alternate|alternate-reverse]#

Startwert

normal

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente, die Pseudoelemente `::before` und `::after`.

Beschreibung

Gibt an, ob eine mehrmals abgespielte CSS-Animation (siehe `animation-iteration-count`) bei der folgenden Wiederholung in gleicher Richtung oder umgekehrt wiedergegeben werden soll. Wird beispielsweise bei einer Animation ein Element um 300 Pixel nach rechts verschoben, wird es bei der Verwendung von `animation-direction: alternate` bei jedem zweiten Ablauf der Animation um 300 Pixel nach links an seine ursprüngliche Startposition verschoben. Verwenden Sie in der gleichen Animation `animation-direction: normal`, würde das Element zuerst um 300 Pixel nach rechts verschoben, dann an seine Startposition zurückspringen, wieder um 300 Pixel nach rechts rücken und so weiter, bis die Animation anhält (sofern das jemals passiert).

Beispiele

```
body {animation-direction:
    alternate, normal, normal;}

#scanner {animation-direction: normal;}
```

animation-duration

Vererbt: N Anim.: N

Werte

`<Zeit>#`

Startwert

`0s`

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente, die Pseudoelemente `::before` und `::after`.

Beschreibung

Definiert, wie lange ein Abspielzyklus einer CSS-Animation von Anfang bis Ende dauern soll. Bei einer Animation mit nur einem Durchlauf definiert `animation-duration` demnach, wie lange die Animation insgesamt dauern soll. Der Standardwert `0s` bedeutet, dass das Element direkt von seinem Startzustand zu seinem Endzustand springt, ohne dass eine Animation stattfindet. Negative Werte werden in `0s` umgewandelt.

Beispiele

```
h1 {animation-duration: 10s, 5s, 2.5s, 1250ms;}
```

```
.zip {animation-duration: 90ms;}
```

animation-fill-mode

Vererbt: N Anim.: N

Werte

[none | forwards | backwards | both]

Startwert

???

Berechneter Wert

Wie angegeben.

Angewendet auf

Alle Elemente, die Pseudoelemente `::before` und `::after`.

Beschreibung

Definiert, welche Eigenschaftswerte von der Animation verwendet werden, wenn diese nicht ausgeführt wird.

Beispiele

```
h1 {animation-fill-mode: forwards;}
```

animation-iteration-count

Vererbt: N Anim.: N

Werte

[<Zahl> | infinite]#

Startwert

1

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente, die Pseudoelemente `::before` und `::after`.

Beschreibung

Definiert die Anzahl der Wiederholungen für eine oder mehrere Animationen. Der Standardwert `1` bedeutet, dass die Animation genau einmal vom Start- bis zum Endzustand durchlaufen wird. Ein Dezimalbruch (z. B. `2.75`) bedeutet, dass die Animation während ihres letzten Durchlaufs angehalten wird. Der Wert `0` bedeutet, dass keine Animation stattfindet; negative Werte werden zu `0` konvertiert. Wie der Name vermuten lässt, bedeutet `infinite`, dass die Animation niemals endet. Verwenden Sie diesen Wert mit Vorsicht.

Beispiele

```
body {animation-iteration-count: 2, 1, 7.5875;}
```

```
ol.dance {animation-iteration-count: infinite;}
```

animation-name

Vererbt: N Anim.: N

Werte

[<Name einer einzelnen Animation> | none]#

Startwert

none

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente, die Pseudoelemente `::before` und `::after`.

Beschreibung

Definiert die deklarierten Namen für eine oder mehrere CSS-Animationen. Jeder bezieht sich auf die @-Regel eines CSS-Animations-Keyframes. Wurde weder ein Animationsname deklariert noch das Schlüsselwort `none` angegeben, wird die Animation nicht ausgeführt, unabhängig von den Werten anderer Animationseigenschaften. Wurde bei der Deklaration `animation-name: bounce, none, jumper` der Animationsname `jumper` nicht definiert, wird die erste Animation ausgeführt, die zweite und dritte jedoch nicht.

Beispiele

```
html {animation-name: turn, slide, none;}
```

```
h2 {animation-name: flip;}
```

animation-play-state

Vererbt: N Anim.: N

Werte

[running | paused]#

Startwert

running

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente, die Pseudoelemente `::before` und `::after`.

Beschreibung

Definiert den Abspielzustand einer oder mehrerer CSS-Animationen. Der Standardzustand `running` ist vor allem in statischen CSS-Umgebungen hilfreich. Die Eigenschaft kann aber auch benutzt werden, um Animationen via DOM-Scripting oder interaktives CSS (z. B. `:hover`) zu starten und anzuhalten.

Beispiele

```
pre {animation-play-state:
```

```
    running, paused, running;}
```

```
table {animation-play-state: running;}
```

animation-timing-function

Vererbt: N Anim.: N

Werte

[`ease` | `linear` | `ease-in` | `ease-out` | `ease-in-out` | `step-start` | `step-end` | `steps(<Integerwert>, start)` | `steps(<Integerwert>, end)` | `cubic-bezier(<Zahl>,<Zahl>,<Zahl>,<Zahl>)`]#

Startwert

`ease`

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente, die Pseudoelemente `::before` und `::after`.

Beschreibung

Definiert, wie Animationen über den Gesamtverlauf einer Animation oder eines bestimmte Keyframes ausgeführt wird, je nachdem, wo die Eigenschaft verwendet wird. Alle Schlüsselwörter sind als `cubic-bezier()`-Äquivalent definiert. So ist z. B. `linear` gleichbedeutend mit `cubic-bezier(0,0,1,1)`. Daher sollten auch die Schlüsselwörter browserübergreifend konsistente Ergebnisse liefern. Allerdings sollten Sie sich, wie immer, nicht darauf verlassen.

Beispiele

```
h1 {animation-timing-function: ease, ease-in,  
  
      cubic-bezier(0.13,0.42,0.67,0.75)}  
  
p {animation-timing-function: linear;}
```

backface-visibility

Vererbt: N Anim.: N

Werte

`visible|hidden`

Startwert

`visible`

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Jedes transformierbare Element.

Beschreibung

Definiert, ob die Rückseite eines Elements sichtbar ist, nachdem es in einem simulierten 3-D-Raum rotiert wurde und vom Betrachter »abgewandt« ist. Lautet der Wert `hidden`, ist das Element quasi »unsichtbar«, bis es so rotiert wird, dass seine Vorderseite dem Betrachter »zugewandt« ist.

Beispiele

```
div.card {backface-visibility: hidden;}
```

```
span.cubeseide {backface-visibility: visible;}
```

background

Vererbt: N Anim.: T

Werte

[*<Hintergrund-Eigenschaften>* ,]* *<abschließende Hintergrund-Eigenschaften>*

Startwert

Siehe die einzelnen Eigenschaften.

Berechneter Wert

Siehe die einzelnen Eigenschaften.

Angewendet auf

Alle Elemente.

Animierbar

Siehe die einzelnen background-Eigenschaften.

Beschreibung

Eine Kurzschrifteigenschaft, mit der die verschiedenen Hintergrund-Eigenschaften für ein oder mehrere Elemente in einer gemeinsamen Deklaration angegeben werden können. Wie alle Kurzschrifteigenschaften setzt diese Eigenschaft sämtliche erlaubten Eigenschaften (z. B. `background-repeat`, `background-position` etc.) auf ihre Standardwerte zurück, sofern keine expliziten Werte angegeben wurden. Die folgenden Regeln führen demnach zum gleichen Erscheinungsbild:

```
background: yellow;
```

```
background: yellow none top left repeat;
```

Außerdem können diese Standardwerte vorherige Deklarationen überschreiben, auch wenn diese für spezifischere Hintergrund-Eigenschaften vorgenommen wurden, zum Beispiel:

```
h1 {background-repeat: repeat-x;}
```

```
h1, h2 {background: yellow url(headback.gif);}
```

Für h1- und h2-Elemente wird der Wiederholungswert (**-repeat*) auf das standardmäßige *repeat* zurückgesetzt. Dadurch wird der zuvor deklarierte Wert *repeat-x* überschrieben.

Werden mehrere Hintergründe deklariert, darf nur der letzte eine Hintergrundfarbe besitzen. Überschneiden sich mehrere Hintergrundbilder, werden diese »gestapelt«. Dabei liegt das zuerst definierte Bild am weitesten »oben« (dem Betrachter am nächsten) und das zuletzt definierte am weitesten »unten« (vom Betrachter entfernt). Das scheint nicht besonders intuitiv, zumal bei der CSS-Positionierung genau umgekehrt vorgegangen wird.

Beispiele

```
body {background: white url(bg41.gif)
```

```
    fixed center repeat-x;}
```

```
p {background:
```

```
    url(/pix/water.png) center repeat-x,
```

```
    top left url(/pix/stone.png) #555;}
```

```
pre {background: yellow;}
```

background-attachment

Vererbt: N Anim.: N

Werte

`[scroll|fixed|local]#`

Startwert

`scroll`

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente.

Beschreibung

Definiert, ob Hintergrundbilder beim Scrollen des Dokuments zusammen mit dem Element gescrollt werden. Diese Eigenschaft kann verwendet werden, um festzulegen, ob das Ursprungsbild für einen Hintergrund z. B. mit der Seite scrollen soll oder ob es an einer bestimmten Stelle fixiert ist. Weitere Details finden Sie in Kapitel 9 von *CSS: The Definitive Guide*, 4th Edition (O'Reilly).

Beispiele

```
body {background-attachment:
```

```
    scroll, scroll, fixed;}
```

```
div.fixbg {background-attachment: fixed;}
```

background-blend-mode

Vererbt: N Anim.: N

Werte

`[normal|multiply|screen|overlay|darken|lighten|color-dodge|color-burn|hard-light|soft-light|`

difference | exclusion | hue | saturation | color |
luminosity]#

Startwert

normal

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente.

Beschreibung

Legt den Mischmodus für sich überschneidende Hintergrundbilder vor einem »Backdrop« fest. Der »Backdrop« ist in diesem Fall eine transparente Ebene hinter der Hintergrundfarbe (am weitesten vom Betrachter entfernt). Der Standardwert `normal` sorgt für eine einfache Alphaüberblendung, die unterstützt wird, seit es CSS gibt. Andere Mischmodi sorgen dafür, dass das Hintergrundbild und sein »Backdrop« auf verschiedene Weise miteinander kombiniert werden. So bedeutet `lighten`, dass im Resultat für jedes Pixel das jeweils hellere dargestellt wird, je nachdem, ob dies nun dem Hintergrundbild oder dem »Backdrop« entstammt. `darken` funktioniert entsprechend, nur dass hier das jeweils dunklere Pixel verwendet wird. Die Ergebnisse werden Benutzern von Photoshop oder ähnlichen Grafikprogrammen bekannt vorkommen. Das Mischen mehrerer Hintergrundebenen findet von hinten nach vorne statt.

Beispiele

```
li.shadowed {background-blend-mode: darken;}
```

```
aside {background-blend-mode:
```

```
color-burn, luminosity, darken;}
```

background-clip

Vererbt: N Anim.: N

Werte

[border-box | padding-box | content-box | text]#

Startwert

border-box

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente.

Beschreibung

Definiert, wo (bezogen auf die Elementbox) der Hintergrund beschnitten, das heißt nicht weiter dargestellt wird. Früher entsprach dieser Wert grundsätzlich dem Standardwert `border-box`, bei dem der Hintergrund bis zur Außenkante des Rahmens reicht. Diese Eigenschaft ermöglicht es, den Inhalt auf die Außenkante des Innenabstands oder die Außenkante des Inhaltsbereichs selbst zu beschränken.

Beispiele

```
body {background-clip: content-box;}
```

```
.callout {background-clip:
```

```
content-box, border-box, padding-box;}
```

background-color

Vererbt: N Anim.: J

Werte

<Farbe>

Startwert

transparent

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente.

Beschreibung

Definiert eine einfache Farbe für den Hintergrund eines Elements. Diese Farbe füllt die durch den Wert von `background-clip` definierte Box aus. Standardmäßig gehören dazu Inhaltsbereich, Innenabstand und Rahmen bis zu seiner Außenkante. Bei Rahmen mit transparenten Bereichen (z. B. gestrichelten Rahmen) ist in den Lücken die Hintergrundfarbe zu sehen – sofern sich die Hintergrundfarbe in den Bereich der Rahmen erstreckt.

Beispiele

```
h4 {background-color: white;}
```

```
p {background-color: rgba(50%,50%,50%,0.33);}
```

```
pre {background-color: #FF9;}
```

background-image

Vererbt: N Anim.: N

Werte

[<Bild> | none]#

Startwert

none

Berechneter Wert

Entspricht dem deklarierten Wert, die URIs werden gegebenenfalls in ihre absolute Entsprechung umgewandelt.

Angewendet auf

Alle Elemente.

Beschreibung

Platziert ein oder mehrere Bilder im Hintergrund eines Elements. Abhängig vom Wert von `background-repeat` kann das Bild entlang der x- und/oder der y-Achse unendlich oft gekachelt werden oder auch überhaupt nicht. Das Ursprungsbild wird entsprechend dem Wert von `background-position` platziert.

Beispiele

```
body {background-image:
```

```
    url(bg41.gif), url(bg43.png), url(bg51.jpg);}
```

```
h2 {background-image:
```

```
    url(http://www.pix.org/dots.png);}
```

background-origin

Vererbt: N Anim.: N

Werte

[`border-box` | `padding-box` | `content-box`]#

Startwert

`padding-box`

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente.

Beschreibung

Definiert die Grenze innerhalb des Elements, anhand deren der Ursprung für die Positionierung des Hintergrundbilds berechnet wird. Früher entsprach dies dem Wert von `padding-box`. Diese Eigenschaft ermöglicht die Definition verschiedener Positionierungskontexte. Beachten Sie, dass ein Teil des Hintergrundbilds abgeschnitten werden kann, falls der Wert von `background-origin` »weiter außen« liegt als der Wert von `background-clip` und das Bild an einer Kante positioniert wird. Hier ein Beispiel:

```
div#example {background-origin: border-box;

background-clip: content-box;

background-position: 100% 100%;}
```

In diesem Fall wird das Bild so platziert, dass seine rechte untere Ecke an der rechten unteren Ecke der äußeren Rahmenkante ausgerichtet wird. Sichtbar sind allerdings nur die Teile, die innerhalb des Inhaltsbereichs liegen.

Beispiele

```
html, body {background-origin: border-box;}

h1 {background-origin: content-box, padding-box;}
```

background-position

Vererbt: N Anim.: J

Werte

<Position>#

Startwert

0% 0%

Berechneter Wert

Die absoluten Abstände, sofern die Werte für *<Position> als <Länge>* deklariert sind; ansonsten Prozentwerte.

Prozentwerte

Beziehen sich auf den entsprechenden Punkt sowohl im Element als auch im Ursprungsbild.

Angewendet auf

Blockelemente und ersetzte Elemente.

Beschreibung

Definiert die Position(en) der Ursprungsbilder eines oder mehrerer Hintergründe (gemäß der Definition durch `background-image`). Dies ist der Punkt, von dem eine eventuelle Hintergrundwiederholung oder -kachelung ausgeht. Prozentwerte definieren nicht nur einen Punkt innerhalb des Elements, sondern auch denselben Punkt im Ursprungsbild selbst, sodass es möglich ist, ein Bild zu zentrieren, indem seine Position mit `50% 50%` angegeben wird. Werden Prozent- oder Längewerte verwendet, bezeichnet der erste Wert stets die horizontale Position und der zweite die vertikale. Wird nur ein Wert angegeben, bestimmt er die horizontale Position, während für den fehlenden Wert entweder `center` oder `50%` angenommen wird. Negative Werte sind erlaubt und können den Ursprung des Bilds außerhalb des Inhaltsbereichs des Bilds platzieren, ohne dass es tatsächlich dargestellt wird. Der Kontext, in dem ein Ursprungsbild platziert wird, kann durch den Wert von `background-origin` beeinflusst werden.

Beispiele

```
body {background-position: top center;}
```

```
div#navbar {background-position:
```

```
    right, 50% 75%, 0 40px;}
```

```
pre {background-position: 10px 50%;}
```

background-repeat

Vererbt: N Anim.: N

Werte

<Art der Wiederholung>#

Definition

<Art der Wiederholung>

repeat-x | repeat-y | [repeat | space | round | no-repeat]
{1,2}

Startwert

repeat

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente.

Beschreibung

Definiert, nach welchem Muster eines oder mehrere Hintergrundbilder gekachelt werden. Die Kachelung beginnt mit dem Ursprungsbild, das als Wert von `background-image` definiert ist und gemäß dem Wert von `background-position` (und möglicherweise `background-origin`) platziert wird. Bei den Schlüsselwörtern `space` und `round` wird das Bild so oft gekachelt, wie es ohne Beschnitt in den Hintergrundbereich passt. Anschließend werden das erste und das letzte Bild an ihren jeweiligen Hintergrundkanten platziert. Der Unterschied besteht darin, dass bei `space` die Abstände zwischen den dazwischenliegenden Bildern gleichmäßig verteilt werden, während die Bilder bei `round` so gedehnt werden, dass sie einander berühren. Beachten Sie, dass `repeat-x` und `repeat no-repeat` sowie `repeat-y` und `no-repeat repeat` jeweils gleichbedeutend sind.

Beispiele

```
body {background-repeat: no-repeat;}
```

```
h2 {background-repeat: repeat-x, repeat-y;}
```

```
ul {background-repeat:  
  
    repeat-y, round space, repeat;}
```

background-size

Vererbt: N Anim.: J

Werte

[[*<Länge>* | *<Prozentwert>* | auto]{1,2} | cover | contain]#

Startwert

auto

Berechneter Wert

Entspricht dem deklarierten Wert, Längenangaben werden gegebenenfalls in absolute Werte umgewandelt, und möglicherweise fehlende auto-Schlüsselwörter werden hinzugefügt.

Angewendet auf

Alle Elemente.

Beschreibung

Definiert die Größe eines oder mehrerer Ursprungsbilder. Werden zwei Werte verwendet (z. B. 50px 25%), definiert der erste die horizontale Ausdehnung des Bilds, der zweite die vertikale. Das Ursprungsbild kann durch die Angabe von 100% 100% so skaliert werden, dass es den Hintergrund genau abdeckt. Im Gegensatz dazu skaliert cover das Bild so, dass es den gesamten Hintergrund bedeckt, auch wenn dadurch ein Teil davon über den Hintergrund hinausragt und dadurch Beschnitt erzeugt. Dagegen skaliert der Wert contain das Ursprungsbild so, dass mindestens eine seiner Dimensionen die entsprechende Achse des Hintergrundbereichs vollständig ausfüllt.

Beispiele

```
body {background-size: 100% 90%;}
```

```
div.photo {background-size: cover;}
```

border

Vererbt: N Anim.: T

Werte

[*<border-width>* || *<border-style>* || *<border-color>*]

Startwert

Siehe die einzelnen Eigenschaften.

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente.

Animierbar

Rahmenbreite und -farbe, nicht Rahmenstil.

Beschreibung

Eine Kurzschrifteigenschaft, die Breite, Farbe und Stil für den Rahmen eines Elements definiert. Auch wenn keiner der Werte tatsächlich erforderlich ist, führt das Weglassen des Rahmenstils dazu, dass kein Rahmen dargestellt wird, da der Standardwert für Rahmenstile `none` ist.

Beispiele

```
h1 {border: 2px dashed olive;}
```

```
a:link {border: blue solid 1px;}
```

```
p.warning {border: double 5px red;}
```

border-bottom

Vererbt: N Anim.: T

Werte

[*<border-width>* || *<border-style>* || *<border-color>*]

Startwert

Siehe die einzelnen Eigenschaften.

Berechneter Wert

Siehe die einzelnen Eigenschaften.

Angewendet auf

Alle Elemente.

Animierbar

Rahmenbreite und -farbe, nicht Rahmenstil.

Beschreibung

Eine Kurzschrifteigenschaft, die Breite, Farbe und Stil für den unteren Rahmen eines Elements definiert. Wie bei `border` kann das Fehlen eines Rahmenstils dazu führen, dass der Rahmen nicht dargestellt wird.

Beispiele

```
ul {border-bottom: 0.5in groove green;}
```

```
a:active {border-bottom: purple 2px dashed;}
```

border-bottom-color

Vererbt: N Anim.: J

Werte

<Farbe>

Startwert

`currentColor`

Berechneter Wert

Eine Farbe.

Angewendet auf

Alle Elemente.

Beschreibung

Definiert die Farbe für die sichtbaren Teile des unteren Rahmens für ein Element. Damit der Rahmen tatsächlich dargestellt wird, muss der Rahmenstil einen anderen Wert haben als none oder hidden.

Beispiele

```
ul {border-bottom-color: green;}
```

```
a:active {border-bottom-color: purple;}
```

border-bottom-left-radius

Vererbt: N Anim.: J

Werte

[<Länge> | <Prozentwert>]{1,2}

Startwert

0

Berechneter Wert

Zwei Werte, entweder beide ein <Prozentwert> oder eine absolute <Länge>.

Prozentwerte

Beziehen sich auf die relevanten Ausmaße der Rahmenbox.

Angewendet auf

Alle Elemente außer interne Tabellenelemente.

Beschreibung

Definiert den Radius für das Abrunden der linken unteren Rahmenecke eines Elements. Wurden zwei Werte angegeben, steht der erste für den horizontalen und der zweite für den vertikalen Radius. Siehe border-

radius für eine Beschreibung davon, wie die Werte die Form der Abrundung bestimmen.

Beispiele

```
h1 {border-bottom-left-radius: 10%;}
```

```
h2 {border-bottom-left-radius: 1em 10px;}
```

border-bottom-right-radius

Vererbt: N Anim.: J

Werte

[<Länge> | <Prozentwert>]{1,2}

Startwert

0

Berechneter Wert

Zwei Werte, entweder beide ein <Prozentwert> oder eine absolute <Länge>.

Prozentwerte

Beziehen sich auf die relevanten Ausmaße der Rahmenbox.

Angewendet auf

Alle Elemente außer interne Tabellenelemente.

Beschreibung

Definiert den Radius für das Abrunden der rechten unteren Rahmenecke eines Elements. Wurden zwei Werte angegeben, steht der erste für den horizontalen und der zweite für den vertikalen Radius. Siehe border-radius für eine Beschreibung davon, wie die Werte die Form der Abrundung bestimmen.

Beispiele

```
h1 {border-bottom-right-radius: 10%;}
```

```
h2 {border-bottom-right-radius: 1em 10px;}
```

border-bottom-style

Vererbt: N Anim.: N

Werte

none | hidden | dotted | dashed | solid | double | groove |
ridge | inset | outset

Startwert

none

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente.

Beschreibung

Definiert den Rahmenstil für den unteren Rahmen eines Elements. Damit der Rahmen tatsächlich dargestellt wird, muss der Rahmenstil einen anderen Wert haben als none oder hidden.

Beispiele

```
ul {border-bottom-style: groove;}
```

```
a:active {border-bottom-style: dashed;}
```

border-bottom-width

Vererbt: N Anim.: J

Werte

[thin | medium | thick | <Länge>]

Startwert

Siehe die einzelnen Eigenschaften.

Berechneter Wert

Siehe die einzelnen Eigenschaften (`border-top-style` etc.).

Angewendet auf

Alle Elemente.

Beschreibung

Definiert die Breite des unteren Rahmens für ein Element. Das funktioniert nur, wenn der Rahmenstil einen anderen Wert hat als `none` oder `hidden`. Hat der Rahmenstil den Wert `none`, wird die Rahmenbreite effektiv auf `0` zurückgesetzt. Negative Längenangaben sind nicht erlaubt.

Beispiele

```
ul {border-bottom-width: 0.5in;}
```

```
a:active {border-bottom-width: 2px;}
```

border-collapse

Vererbt: J Anim.: J

Werte

`collapse` | `separate` | `inherit`

Startwert

`separate`

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Elemente mit dem `display`-Wert `table` oder `table-inline`.

Beschreibung

Legt fest, welches Layoutmodell bei der Darstellung von Rahmen in einer Tabelle verwendet wird. Hierzu gehören z. B. die Rahmen für Zellen, Zeilen

etc. Obwohl die Eigenschaft nur auf Tabellen angewendet wird, wird sie von allen Elementen innerhalb der Tabelle geerbt und auch verwendet.

Beispiel

```
table {border-collapse: separate;  
  
border-spacing: 3px 5px;}
```

Hinweis

In CSS2 lautete der Standardwert `collapse`.

border-color

Vererbt: N Anim.: J

Werte

<Farbe>{1,4}

Startwert

Siehe die einzelnen Eigenschaften.

Berechneter Wert

Siehe die einzelnen Eigenschaften (`border-top-color` etc.).

Angewendet auf

Alle Elemente.

Beschreibung

Eine Kurzschrifteigenschaft, die die Farbe für die sichtbaren Teile des gesamten Rahmens eines Elements oder unterschiedliche Farben für die vier Rahmenseiten festlegt. Denken Sie daran, dass der Stil eines Rahmens etwas anderes als `none` oder `hidden` sein muss, damit der Rahmen tatsächlich dargestellt wird.

Beispiele

```
h1 {border-color: purple;}
```

```
a:visited {border-color: maroon;}
```

border-image

Vererbt: N Anim.: T

Werte

<border-image-source> || <border-image-slice> [/ <border-image-width> | / <border-image-width>? / <border-image-outset>]? || <border-image-repeat>

Startwert

Siehe die einzelnen Eigenschaften.

Berechneter Wert

Siehe die einzelnen Eigenschaften.

Angewendet auf

Siehe die einzelnen Eigenschaften.

Animierbar

Siehe die einzelnen `border-image`-Eigenschaften.

Beschreibung

Eine Kurzschrifteigenschaft, die Quelle, Slicing-Muster, Rahmenbreite, Richtungswinkel und Wiederholung eines bildbasierten Rahmens definiert. Die Syntax ist im Vergleich zum restlichen CSS etwas ungewöhnlich. Daher sollten Sie besonders sorgfältig vorgehen. So werden beispielsweise drei der fünf möglichen Werte durch Schrägstriche (/) voneinander getrennt und müssen in einer bestimmten Reihenfolge angegeben werden.

Beachten Sie, dass es so gut wie unmöglich ist, ein einfaches Bild (beispielsweise einen Stern) entlang der Kanten eines Elements zu wiederholen. Um diesen Effekt zu erzeugen, müssen Sie ein Bild erzeugen, das neun Kopien des gewünschten Rahmenbilds enthält, das in einem 3×3-Raster wiederholt werden muss. Möglicherweise müssen Sie einen ausreichend großen Wert für `border-width` (*nicht* `border-image-width`) wählen, damit das Bild angezeigt wird, je nachdem, welchen Wert `border-image-outset` hat.

Beispiele

```
div.starry {border-image:
```

```
  url(stargrid.png) 5px repeat;}
```

```
aside {border-image: url(asides.png)
```

```
  100 50 150 / 8 3 13 / 2 stretch round;}
```

border-image-outset

Vererbt: N Anim.: J

Werte

[<Länge> | <Zahl>]{1,4}

Startwert

0

Berechneter Wert

Vier Werte, entweder eine Zahl oder eine absolute <Länge>.

Angewendet auf

Alle Elemente außer interne Tabellenelemente, wenn border-collapse den Wert collapse hat.

Beschreibung

Legt fest, wie weit ein Rahmenbild über den Rahmenbereich des Elements hinausragen darf. Die Werte definieren Abstände von der oberen, rechten, unteren und linken Kante des Rahmenbilds in genau dieser Reihenfolge. Zahlen werden unter Berücksichtigung des intrinsischen Koordinatensystems des Bilds berechnet. So steht die Zahl 7 bei einem Rasterbild (also pixelbasiert) für sieben Pixel. Bilder in Formaten wie SVG können andere Koordinatensysteme besitzen. Negative Werte sind nicht erlaubt.

Beispiele

```
aside {border-image-outset: 2;}
```

```
div#pow {border-image-outset: 10 17 13 5;}
```

border-image-repeat

Vererbt: N Anim.: N

Werte

[stretch | repeat | round | space]{1,2}

Startwert

stretch

Berechneter Wert

Zwei Schlüsselwörter, eines für jede Achse.

Angewendet auf

Alle Elemente außer interne Tabellenelemente, wenn `border-collapse` den Wert `collapse` hat.

Beschreibung

Definiert Wiederholungsmuster für die Seiten eines bildbasierten Rahmens. `stretch` sorgt dafür, dass eine einzelne Kopie des Bilds so gestreckt wird, dass sie in das Rahmensegment (oben, rechts, unten oder links) passt. `repeat` »kachelte« das Bild auf die von Hintergrundbildern bekannte Weise, wobei Rahmenbilder immer nur entlang einer Achse gekachelt werden. `round` kachelt das Rahmenbild gerade so oft, dass es ohne Beschnitt passt, und skaliert dann (falls nötig) den gesamten Satz gekachelter Bilder so, dass er genau in das Rahmensegment passt.

Beispiele

```
div.starry {border-image-repeat: repeat;}
```

```
aside {border-image-repeat: stretch round;}
```

border-image-slice

Vererbt: N Anim.: T

Werte

[<Zahl> | <Prozentwert>]{1,4} && fill?

Startwert

100%

Berechneter Wert

Vier Werte, jeder eine Zahl oder ein Prozentwert, optional zusätzlich das Schlüsselwort `fill`.

Prozentwerte

Beziehen sich auf die Größe des Rahmenbilds.

Angewendet auf

Alle Elemente außer interne Tabellenelemente, wenn `border-collapse` den Wert `collapse` hat.

Animierbar

Nur Werte vom Typ `<Zahl>` und `<Prozentwert>`.

Beschreibung

Definiert die Abstände der einzelnen Slices von der oberen, rechten, unteren und linken Kante des bildbasierten Rahmens. Zusammengenommen, teilen sie das Bild in neun Regionen, die den acht Segmenten des Elementrahmens (vier Ecken und vier Seiten) sowie dem Hintergrundbereich des Elements entsprechen.

Werden zwei einander gegenüberliegende Regionen so kombiniert, dass sie ihre gemeinsamen Gesamtausmaße überschreiten, werden beide vollständig transparent. Beträgt beispielsweise der Wert des oberen Abschnittsabstands 10 und der des unteren 20, während das Quellbild nur 25 Pixel hoch ist, überschreiten beide Werte die Bildhöhe. Deshalb werden sowohl das obere als auch das untere Segment des Rahmens vollkommen transparent. Dasselbe gilt für den rechten und linken Abschnitt und die Breite. Ecken werden niemals absichtlich transparent dargestellt, auch dann nicht, wenn sich ihre Abschnitte mit dem Quellbild überschneiden.

Beispiele

```
div.starry {border-image-slice: 5px;}
```

```
aside {border-image-slice: 100 50 150;}
```

border-image-source

Vererbt: N Anim.: N

Werte

none | *<Bild>*

Startwert

none

Berechneter Wert

none oder das Bild als absoluter URI.

Angewendet auf

Alle Elemente außer interne Tabellenelemente, wenn border-collapse den Wert collapse hat.

Beschreibung

Gibt an, wo das Bild zu finden ist, das als Rahmenbild verwendet werden soll.

Beispiele

```
div.starry {border-image-source:
```

```
url(stargrid.png);}
```

```
aside {border-image-source: url(asides.png);}
```

border-image-width

Vererbt: N Anim.: J

Werte

[<Länge> | <Prozentwert> | <Zahl> | auto]{1,4}

Startwert

1

Berechneter Wert

Vier Werte, jeder ein <Prozentwert>, eine <Zahl>, eine absolute <Länge> oder das Schlüsselwort auto.

Prozentwerte

Relativ zur Breite/Höhe des gesamten Rahmenbildbereichs, das heißt die Außenkanten der Border-Box.

Angewendet auf

Alle Elemente außer Tabellenelemente, wenn border-collapse den Wert collapse hat.

Beschreibung

Definiert eine Bildbreite für jede der vier Seiten eines bildbasierten Rahmens. Rahmenbild-Slices, deren Breite sich von der des Rahmenbilds unterscheidet, werden so skaliert, dass die Breite übereinstimmt, was möglicherweise die Kachelung beeinflusst. Ist beispielsweise die rechte Kante eines bildbasierten Rahmens 10 Pixel breit, während border-image-width den Wert 3px hat, werden die Rahmenbilder für die rechte Seite auf drei Pixel Breite skaliert.

Dabei ist border-image-width nicht das Gleiche wie border-width: Die Breite eines Rahmenbilds kann sich von der Breite des Rahmenbereichs unterscheiden. Ist das Bild breiter oder höher als der Rahmenbereich, wird das Bild standardmäßig beschnitten (was sich allerdings per border-image-outset verhindern lässt). Ist das Rahmenbild schmaler oder kürzer als der Rahmenbereich, wird es *nicht* hochskaliert. Negative Werte sind nicht erlaubt.

Beispiele

```
aside {border-image-width: 8 3 13;}
```

```
div#pow{border-image-width: 25px 35;}
```

border-left

Vererbt: N Anim.: T

Werte

[*<border-width>* || *<border-style>* || *<border-color>*]

Startwert

Siehe die einzelnen Eigenschaften.

Berechneter Wert

Siehe die einzelnen Eigenschaften (border-width etc.).

Angewendet auf

Alle Elemente.

Animierbar

Rahmenbreite und -farbe, nicht Rahmenstil.

Beschreibung

Eine Kurzschrifteigenschaft, die Breite, Farbe und Stil für den linken Rahmen eines Elements definiert. Wie bei border kann das Fehlen eines Rahmenstils dazu führen, dass der Rahmen nicht dargestellt wird.

Beispiele

```
p {border-left: 3em solid gray;}
```

```
pre {border-left: double black 4px;}
```

border-left-color

Vererbt: N Anim.: J

Werte

<Farbe>

Startwert

currentColor

Berechneter Wert

Eine Farbe.

Angewendet auf

Alle Elemente.

Beschreibung

Definiert die Farbe für die sichtbaren Teile des linken Rahmens für ein Element. Damit der Rahmen tatsächlich dargestellt wird, muss der Rahmenstil einen anderen Wert haben als none oder hidden.

Beispiele

```
p {border-left-color: gray;}
```

```
pre {border-left-color: black;}
```

border-left-style

Vererbt: N Anim.: N

Werte

none | hidden | dotted | dashed | solid | double | groove |
ridge | inset | outset

Startwert

none

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente.

Beschreibung

Definiert den Rahmenstil für den linken Rahmen eines Elements. Damit der Rahmen tatsächlich dargestellt wird, muss der Rahmenstil einen anderen Wert haben als `none` oder `hidden`.

Beispiele

```
p {border-left-style: solid;}
```

```
pre {border-left-style: double;}
```

border-left-width

Vererbt: N Anim.: J

Werte

`thin` | `medium` | `thick` | `<Länge>`

Startwert

`medium`

Berechneter Wert

Eine absolute Längenangabe, or 0 if the style of the border is `none` or `hidden`; otherwise, as declared.

Angewendet auf

Alle Elemente.

Beschreibung

Definiert die Breite des linken Rahmens für ein Element. Das funktioniert nur, wenn der Rahmenstil einen anderen Wert hat als `none` oder `hidden`. Hat der Rahmenstil den Wert `none`, wird die Rahmenbreite effektiv auf 0 zurückgesetzt. Negative Längenangaben sind nicht erlaubt.

Beispiele

```
p {border-left-width: 3em;}
```

```
pre {border-left-width: 4px;}
```

border-radius

Vererbt: N Anim.: J

Werte

[<Länge> | <Prozentwert>]{1,4} [/ [<Länge> | <Prozentwert>]{1,4}]?

Startwert

0

Berechneter Wert

Vier Werte, jeweils ein absoluter <Prozentwert> oder eine absolute <Länge>.

Prozentwerte

Beziehen sich auf die relevanten Ausmaße der Rahmenbox.

Angewendet auf

Alle Elemente außer interne Tabellenelemente

Beschreibung

Eine Kurzschrifteigenschaft, die den Radius für das Abrunden der Rahmenecken eines Elements definiert. Die Ecken haben jeweils die deklarierte Höhe und Breite.

```
.callout {border-radius: 10px;}
```

Alle Ecken eines Elements, dessen class-Attribut den Wert `callout` hat, werden mit einer Abrundung von 10 Pixeln Breite und 10 Pixeln Höhe versehen. Diese wird vom Anfang der Abrundung bis zur seitlichen Außenkante des Elements gemessen. Das kann man sich vorstellen, als zeichnete man in den Ecken des Elements Kreise mit einem Radius von 10 Pixeln (20 Pixel Durchmesser) und würde die Ecken um diese Kreise »herumbiegen«.

Werden weniger als vier Werte angegeben, werden die angegebenen Werte nach dem bekannten Muster wiederholt (siehe `margin`, `padding` etc.). Allerdings gibt es einen kleinen Unterschied. Anstelle der Reihenfolge oben, rechts, unten, links (englisch *Top-Right-Bottom-Left*, kurz TRBL oder »trouble«) lautet das Muster hier links oben, rechts oben, rechts unten,

links unten (oder »LoRoRuLu« bzw. englisch *TopLeft-TopRight-BottomRight-BottomLeft*, kurz TLTRBRBL oder »tilter burble«). Ansonsten ist das Wiederholungsmuster identisch.

Die Berechnung von Prozentwerten geschieht anhand der Größe der Rahmenbox des Elements (das heißt der Box, die durch die Außenkanten des Rahmens des Elements definiert werden) auf der entsprechenden Achse.

Beispiele

```
a[href] {border-radius: 0.5em 50%;}
```

```
.callout {border-radius:
```

```
10px 20px 30px 40px /
```

```
1em 2em 3em 4em;}
```

border-right

Vererbt: N Anim.: T

Werte

[*<border-width>* || *<border-style>* || *<border-color>*]

Startwert

Siehe die einzelnen Eigenschaften.

Berechneter Wert

Siehe die einzelnen Eigenschaften (*border-width* etc.).

Angewendet auf

Alle Elemente.

Animierbar

Rahmenbreite und -farbe, nicht Rahmenstil.

Beschreibung

Eine Kurzschrifteigenschaft, die Breite, Farbe und Stil für den rechten Rahmen eines Elements definiert. Wie bei `border` kann das Fehlen eines Rahmenstils dazu führen, dass der Rahmen nicht dargestellt wird.

Beispiele

```
img {border-right: 30px dotted blue;}
```

```
h3 {border-right: cyan 1em inset;}
```

border-right-color

Vererbt: N Anim.: J

Werte

<Farbe>

Startwert

`currentColor`

Berechneter Wert

Eine Farbe.

Angewendet auf

Alle Elemente.

Beschreibung

Definiert die Farbe für die sichtbaren Teile des rechten Rahmens für ein Element. Damit der Rahmen tatsächlich dargestellt wird, muss der Rahmenstil einen anderen Wert haben als `none` oder `hidden`.

Beispiele

```
img {border-right-color: blue;}
```

```
h3 {border-right-color: cyan;}
```

border-right-style

Vererbt: N Anim.: N

Werte

none | hidden | dotted | dashed | solid | double | groove |
ridge | inset | outset

Startwert

none

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente.

Beschreibung

Definiert den Rahmenstil für den rechten Rahmen eines Elements. Damit der Rahmen tatsächlich dargestellt wird, muss der Rahmenstil einen anderen Wert haben als none oder hidden.

Beispiele

```
img {border-right-style: dotted;}
```

```
h3 {border-right-style: inset;}
```

border-right-width

Vererbt: N Anim.: J

Werte

thin | medium | thick | <Länge>

Startwert

medium

Berechneter Wert

Eine absolute Längenangabe, or 0 if the style of the border is none or hidden; otherwise, as declared.

Angewendet auf

Alle Elemente.

Beschreibung

Definiert die Breite des rechten Rahmens für ein Element. Das funktioniert nur, wenn der Rahmenstil einen anderen Wert hat als none oder hidden. Hat der Rahmenstil den Wert none, wird die Rahmenbreite effektiv auf 0 zurückgesetzt. Negative Längenangaben sind nicht erlaubt.

Beispiele

```
img {border-right-width: 30px;}
```

```
h3 {border-right-width: 1em;}
```

border-spacing

Vererbt: J Anim.: J

Werte

<Länge> <Länge>?

Startwert

0

Berechneter Wert

Zwei absolute Werte vom Typ <Länge>.

Angewendet auf

Elemente mit dem display-Wert table oder table-inline.

Beschreibung

Wird das Tabellen-Layoutmodell der getrennten Rahmen verwendet, legt diese Eigenschaft den Abstand zwischen den Zellenrahmen fest. Die erste der beiden Längenangaben definiert den horizontalen Abstand, die zweite

den vertikalen. Obwohl die Eigenschaft nur auf Tabellen angewendet wird, wird sie von allen Elementen innerhalb der Tabelle geerbt.

Beispiele

```
table {border-spacing: 0;}
```

```
table {border-spacing: 3px 5px;}
```

Hinweis

Diese Eigenschaft wird ignoriert, es sei denn, `border-collapse` hat den Wert `separate`.

border-style

Vererbt: N Anim.: N

Werte

[none | hidden | solid | dotted | dashed | double | groove | ridge | inset | outset]{1,4}

Startwert

Siehe die einzelnen Eigenschaften.

Berechneter Wert

Siehe die einzelnen Eigenschaften (`border-top-style` etc.).

Angewendet auf

Alle Elemente.

Beschreibung

Eine Kurzschrifteigenschaft, mit der die Rahmenstile aller Seiten des Elements gemeinsam oder einzeln definiert werden können. Damit die Rahmenseiten angezeigt werden, muss `border-style` einen anderen Wert haben als `none` (Standardwert) oder `hidden`. Dies gilt sowohl für die einzelnen Rahmenseiten wie auch für die gemeinsame Definition. Hat `border-style` den Wert `none`, werden die Angaben für `border-width` ignoriert und auf `0` gesetzt. Unbekannte Werte aus der Werteliste sollten durch das Benutzerprogramm als `solid` interpretiert werden.

```
h1 {border-style: solid;}
```

```
img {border-style: inset;}
```

border-top

Vererbt: N Anim.: T

Werte

[*<border-width>* || *<border-style>* || *<border-color>*]

Startwert

Siehe die einzelnen Eigenschaften.

Berechneter Wert

Siehe die einzelnen Eigenschaften (*border-width* etc.).

Angewendet auf

Alle Elemente.

Animierbar

Rahmenbreite und -farbe, nicht Rahmenstil.

Beschreibung

Eine Kurzschrifteigenschaft, die Breite, Farbe und Stil für den oberen Rahmen eines Elements definiert. Wie bei *border* kann das Fehlen eines Rahmenstils dazu führen, dass der Rahmen nicht dargestellt wird.

Beispiele

```
ul {border-top: 0.5in solid black;}
```

```
h1 {border-top: dashed 1px gray;}
```

border-top-color

Vererbt: N Anim.: J

Werte

<Farbe>

Startwert

currentColor

Berechneter Wert

Eine Farbe.

Angewendet auf

Alle Elemente.

Beschreibung

Definiert die Farbe für die sichtbaren Teile des oberen Rahmens eines Elements. Damit der Rahmen angezeigt wird, muss der Rahmenstil einen anderen Wert als none oder hidden haben.

Beispiele

```
ul {border-top-color: black;}
```

```
h1 {border-top-color: gray;}
```

border-top-left-radius

Vererbt: N Anim.: J

Werte

[<Länge> | <Prozentwert>]{1,2}

Startwert

0

Berechneter Wert

Zwei Werte, entweder beide ein <Prozentwert> oder eine absolute <Länge>.

Prozentwerte

Beziehen sich auf die relevanten Ausmaße der Rahmenbox.

Angewendet auf

Alle Elemente außer interne Tabellenelemente.

Beschreibung

Definiert den Radius für das Abrunden der linken oberen Rahmenecke eines Elements. Wurden zwei Werte angegeben, steht der erste für den horizontalen und der zweite für den vertikalen Radius. Siehe `border-radius` für eine Beschreibung davon, wie die Werte die Form der Abrundung bestimmen.

Beispiele

```
h1 {border-top-left-radius: 10%;}
```

```
h2 {border-top-left-radius: 1em 10px;}
```

border-top-right-radius

Vererbt: N Anim.: J

Werte

[<Länge> | <Prozentwert>]{1,2}

Startwert

0

Berechneter Wert

Zwei Werte, entweder beide ein <Prozentwert> oder eine absolute <Länge>.

Prozentwerte

Beziehen sich auf die relevanten Ausmaße der Rahmenbox.

Angewendet auf

Alle Elemente außer interne Tabellenelemente.

Beschreibung

Definiert den Radius für das Abrunden der rechten oberen Rahmenecke eines Elements. Wurden zwei Werte angegeben, steht der erste für den horizontalen und der zweite für den vertikalen Radius. Siehe `border-`

radius für eine Beschreibung davon, wie die Werte die Form der Abrundung bestimmen.

Beispiele

```
h1 {border-top-right-radius: 10%;}
```

```
h2 {border-top-right-radius: 1em 10px;}
```

border-top-style

Vererbt: N Anim.: N

Werte

none | hidden | dotted | dashed | solid | double | groove |
ridge | inset | outset

Startwert

none

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente.

Beschreibung

Definiert den Rahmenstil für den oberen Rahmen eines Elements. Damit der Rahmen tatsächlich dargestellt wird, muss der Rahmenstil einen anderen Wert haben als none oder hidden.

Beispiele

```
ul {border-top-style: solid;}
```

```
h1 {border-top-style: dashed;}
```

border-top-width

Vererbt: N Anim.: J

Werte

thin|medium|thick|<Länge>

Startwert

medium

Berechneter Wert

Eine absolute <Länge> oder 0, falls border-style den Wert none oder hidden hat, ansonsten wie deklariert.

Angewendet auf

Alle Elemente.

Beschreibung

Definiert die Breite des oberen Rahmens für ein Element. Das funktioniert nur, wenn der Rahmenstil einen anderen Wert hat als none oder hidden. Hat der Rahmenstil den Wert none, wird die Rahmenbreite effektiv auf 0 zurückgesetzt. Negative Längenangaben sind nicht erlaubt.

Beispiele

```
ul {border-top-width: 0.5in;}
```

```
h1 {border-top-width: 1px;}
```

border-width

Vererbt: N Anim.: J

Werte

[thin|medium|thick|<Länge>]{1,4}

Startwert

Siehe die einzelnen Eigenschaften.

Berechneter Wert

Siehe die einzelnen Eigenschaften (`border-top-style` etc.).

Angewendet auf

Alle Elemente.

Beschreibung

Eine Kurzschrifteigenschaft, die die Breite des gesamten Rahmens eines Elements oder unterschiedliche Breiten für die vier Rahmenseiten festlegt. Der Wert von `border-width` wird nur berücksichtigt, wenn der Rahmenstil für die Rahmenseite(n) einen anderen Wert hat als `none` oder `hidden`. Hat der Rahmenstil den Wert `none`, wird die Rahmenbreite effektiv auf 0 zurückgesetzt. Negative Längenangaben sind nicht erlaubt.

Beispiele

```
h1 {border-width: 2ex;}
```

```
img {border-width: 5px thick thin 1em;}
```

bottom

Vererbt: N Anim.: T

Werte

`<Länge> | <Prozentwert> | auto`

Startwert

auto

Berechneter Wert

Für statische Elemente `auto`, für Längenangaben die entsprechende absolute `<Länge>`, für `<Prozentwert>` der angegebene Wert, ansonsten `auto`.

Prozentwerte

Beziehen sich auf die Höhe des umgebenden Blocks.

Angewendet auf

Positionierte Elemente.

Beschreibung

Definiert den Abstand zwischen der unteren Außenkante des Außenabstands eines positionierten Elements und der Unterkante seines umgebenden Blocks.

Beispiele

```
div#footer {position: fixed; bottom: 0;}
```

```
sup {position: relative; bottom: 0.5em;
```

```
vertical-align: baseline;}
```

Hinweis

Für relativ positionierte Elemente gilt: Haben `bottom` und `top` den Wert `auto`, ist der berechnete Wert für beide `0`. Hat nur eine der beiden Eigenschaften den Wert `auto`, wird der negative Wert des anderen verwendet. Hat keine den Wert `auto`, erhält `bottom` den negativen Wert von `top`.

box-decoration-break

Vererbt: N Anim.: N

Werte

`slice|clone`

Startwert

`slice`

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente.

Beschreibung

Legt fest, ob die »Dekorationen«, also Hintergrund, Innenabstände, Rahmen, abgerundete Ecken, Rahmenbilder und Schattenwürfe einer Box, die in mehreren Teilen dargestellt werden, auf jeden Teil einzeln angewendet werden oder auf die gesamte Box, bevor diese aufgeteilt wird.

Der häufigste Fall ist ein Inline-Element, das sich über mehrere Zeilenumbrüche erstreckt. Beim Standardverhalten, `slice`, werden die einzelnen Teile des Inline-Elements dargestellt, als befände sich das gesamte Element auf einer einzelnen Zeile und würde dann an den Zeilenumbrüchen »zerschnitten«. Wird dagegen der Wert `clone` verwendet, werden die einzelnen Teile des Elements als separate Elemente mit gleichen Stildefinitionen behandelt.

`box-decoration-break` wird auch auf Blockelemente angewendet, die auf mehrere Spalten oder Seiten verteilt werden.

Beispiele

```
span {box-decoration-break: clone;}
```

```
a {box-decoration-break: slice;}
```

box-shadow

Vererbt: N Anim.: J

Werte

`none` | [`inset?` && `<Länge>{2,4}` && `<Farbe>?`]#

Startwert

`none`

Berechneter Wert

Werte vom Typ `<Länge>` als absolute Werte, Werte vom Typ `<Farbe>` wie intern berechnet, ansonsten wie deklariert.

Angewendet auf

Alle Elemente.

Beschreibung

Definiert einen oder mehrere Schatten, die sich an der Form der Elementbox orientieren. Es können entweder äußere (»Schlagschatten«, Standardverhalten) oder innere Schattenwürfe definiert werden. Für innere Schattenwürfe muss das optionale Schlüsselworts `inset` angegeben werden.

Es können vier Längenwerte angegeben werden. Dies sind (in dieser Reihenfolge): horizontale Verschiebung (Horizontal Offset), vertikale Verschiebung (Vertical Offset), Weichzeichnungsgrad (Blur) und Ausdehnung (Spread). Positive Werte verschieben den Schatten nach rechts bzw. nach unten, negative Werte in die jeweils andere Richtung. Positive Werte für die Ausdehnung vergrößern den Schatten, negative Werte verkleinern ihn. Der Weichzeichnungsgrad darf nicht negativ sein.

Alle Schatten werden durch die Rahmenkante des Elements beschnitten. Daher wird ein äußerer Schatten nur außerhalb der Rahmenkante dargestellt. Auch wenn der Hintergrund des Elements ganz oder teilweise transparent ist, wird *kein* äußerer Schatten »hinter« dem Element angezeigt. Auf ähnliche Weise sind innere (»inset«) Schattenwürfe nur innerhalb der Rahmenkante sichtbar.

Beispiele

```
h1 {box-shadow: 5px 10px gray;}
```

```
table th {box-shadow:
```

```
inset 0.5em 0.75em 5px -2px
```

```
rgba(255,0,0,0.5);}
```

box-sizing

Vererbt: N Anim.: N

Werte

`content-box` | `padding-box` | `border-box`

Startwert

content-box

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente, die Werte für `width` oder `height` akzeptieren.

Beschreibung

Definiert, ob die Werte für `width` (Breite) und `height` (Höhe) eines Elements sich auf die Inhaltsbox (`content-box`, das historische Verhalten), die Innenabstandsbox (`padding-box`) oder auf die Border-Box (`border-box`) beziehen. Im letzteren Fall definiert `width` den Abstand zwischen den seitlichen Rahmenkanten. Entsprechend definiert `height` den Abstand zwischen der oberen und unteren Rahmenkante. Entgegen dem historischen »additiven« Verhalten werden in diesem Fall die Werte für Innenabstände und Rahmenbreite von diesem Wert subtrahiert. Hier ein Beispiel:

```
body {box-sizing: border-box;

width: 880px; padding: 0 20px;}
```

Durch die obigen Regeln ist der Inhaltsbereich schließlich 840 Pixel (880 – 20 – 20) breit.

Beispiel

```
body {box-sizing: padding-box;}
```

caption-side

Vererbt: J Anim.: N

Werte

top | bottom

Startwert

top

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Elemente mit dem `display`-Wert `table-caption`.

Beschreibung

Legt fest, wo die Tabellenbeschriftung bezogen auf die Tabellenbox platziert wird. Die Beschriftung wird dargestellt, als sei unmittelbar vor der Tabelle ein Blockelement eingefügt worden.

Beispiel

```
caption {caption-side: top;}
```

Hinweis

Die Werte `left` und `right` waren Teil von CSS2, wurden aber später mangels ausreichender Unterstützung wieder aufgegeben.

clear

Vererbt: N Anim.: N

Werte

`left` | `right` | `both` | `none`

Startwert

`none`

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Blockelemente.

Beschreibung

Definiert, auf welcher Seite (oder welchen Seiten) eines Elements kein Float stehen darf. Führt das normale Layout eines mit `clear` versehenen Elements dazu, dass an der per `clear` definierten Seite ein Float stünde, so wird das mit `clear` versehene Element nach unten verschoben, bis es sich unterhalb des Floats befindet. In CSS1 und CSS2 wird hierfür der obere Außenabstand des `clear`-Elements automatisch vergrößert. In CSS 2.1 wird der für den `clear`-Vorgang nötige Raum oberhalb des oberen Außenabstands des Elements hinzugefügt. Dabei wird der obere Außenabstand selbst nicht verändert. In beiden Fällen befindet sich die obere Rahmenaußenkante des `clear`-Elements direkt unterhalb der Außenkante des unteren Außenabstands eines Floats auf der definierten Seite.

Beispiele

```
h1 {clear: both;}
```

```
p + h3 {clear: right;}
```

clip-path

Vererbt: N Anim.: T

Werte

```
none | <URI> | [[ inset() | circle() | ellipse() | polygon() ] |  
[ border-box | padding-box | content-box | margin-box |  
fill-box | stroke-box | view-box ]]
```

Startwert

none

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente (in SVG alle Grafikelemente sowie alle Container-elemente außer `defs`).

Animierbar

`inset()`, `circle()`, `ellipse()` und `polygon()` values

Beschreibung

Definiert eine Beschneidungsform, innerhalb deren ein Element sichtbar ist bzw. außerhalb deren ein Element unsichtbar ist. Anhand von `url()`-Werten kann auf eine SVG-Datei oder ein `SVG-clip-path`-Element verwiesen werden, das als Beschneidungsform verwendet wird.

Beispiele

```
p.clipped {clip-path: url(shapes.svg#cloud02);}
```

```
p.rounded {clip-path:
```

```
    ellipse(100px 50px at 75% 25%);}
```

```
p.diamond {clip-path:
```

```
    polygon(50% 0, 100% 50%, 50% 100%, 0 50%);}
```

clip-rule

Vererbt: N Anim.: N

Werte

`nonzero` | `evenodd`

Startwert

`nonzero`

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle SVG-Grafikelemente (`circle`, `ellipse`, `image`, `line`, `path`, `polygon`, `polyline`, `rect`, `text` und `use`), ausschließlich wenn sie Kindelemente eines `clipPath`-Elements sind.

Beschreibung

Verändert die Art, auf die sich überschneidende Teile eines Pfads die resultierende Form füllen würden. Der Wert `nonzero` sorgt dafür, dass die gesamte Form gefüllt wird. Bei der Verwendung von `nonzero` kann es passieren, dass Teile des Innern der Form vollständig transparent sind.

Beispiele

```
span.fullshape {clip-rule: nonzero;}
```

```
span.knockouts {clip-rule: evenodd;}
```

color

Vererbt: J Anim.: J

Werte

<Farbe>

Startwert

Hängt vom jeweiligen Benutzerprogramm ab.

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente.

Beschreibung

Definiert die Vordergrundfarbe eines Elements. In HTML ist dies der Text eines Elements. Auf Rasterbilder (im Gegensatz zu Vektorgrafiken) hat `color` keinen Einfluss. Außerdem wird der Wert von `color` als Farbe für Rahmen verwendet, sofern dieser nicht von einer Eigenschaft für Rahmenfarben (z. B. `border-color`, `border-bottom-color` etc.) überschrieben wird.

Wird für die Farbe ein Schlüsselwort (z. B. `navy`) oder ein hexadezimaler RGB-Wert (z. B. `#008800` oder `#080`) verwendet, ist der berechnete Wert der entsprechende `rgb()`-Wert. Das Schlüsselwort `transparent` wird als `rgba(0,0,0,0)` berechnet; für `currentColor` ist der berechnete

Wert inherit. Für alle anderen Werte entspricht der berechnete Wert dem angegebenen.

Beispiele

```
strong {color: rgb(255,128,128);}
```

```
h3 {color: navy;}
```

```
p.warning {color: #ff0000;}
```

```
pre.pastoral {color: rgba(0%,100%,0%,0.33334);}
```

content

Vererbt: N Anim.: N

Werte

normal | [<String> | <URI> | <Zähler> | attr(<Identifizier>) | open-quote | close-quote | no-open-quote | no-close-quote]+

Startwert

normal

Berechneter Wert

Für <URI>-Werte ein absoluter URI, für Attributreferenzen der resultierende String, ansonsten wie deklariert.

Angewendet auf

Die Pseudoelemente ::before und ::after.

Beschreibung

Definiert den erzeugten Inhalt, der vor oder nach einem Element platziert wird. Standardmäßig ist dies sehr wahrscheinlich Inline-Inhalt. Allerdings kann die Art der erzeugten Box anhand der Eigenschaft display verändert werden.

Beispiele

```
p::before {content: "Paragraph...";}
```

```
a[href]::after {content: "(" attr(href) ")};
```

```
font-size: smaller;}
```

counter-increment

Vererbt: N Anim.: N

Werte

[<Identifizier> <Integerwert>?]+ | none

Startwert

Abhängig vom Benutzerprogramm.

Berechneter Wert

Entspricht dem deklarierten Wert.

Angewendet auf

Alle Elemente.

Beschreibung

Mit dieser Eigenschaft können Zähler (CSS-Counter) um einen beliebigen positiven oder negativen <Integerwert> (auch 0) inkrementiert (oder dekrementiert) werden. Wurde kein Wert angegeben, wird standardmäßig 1 verwendet.

Beispiele

```
h1 {counter-increment: section;}
```

```
*.backward li {counter-increment: counter -1;}
```

Symbole

— für eigene Identifier (CSS-Variablen) 50

; in Stilregeln 5

: in Stilregeln 6

!— ... – für HTML-Kommentare 8

! für Wertedefinitionen 78

? für Wertedefinitionen 78

. für Klassen-Selektor 53

... für Werte 77

[]

für Attribut-Selektoren 54

zum Gruppieren 78

{ } für Wertedefinitionen 79

*

für Wertedefinitionen 78

Selektor 51

/*...*/ für CSS-Kommentare 8

&& für Wertedefinitionen 78

#

für ID-Selektor 54

für Wertedefinitionen 78

+

benachbarte Geschwisterelemente, Selektor 52

für Wertedefinitionen 78

| für Wertedefinitionen 78

|| für Wertedefinitionen 78

~ für allg. Selektor f. Geschwisterelemente 53

A

Abrundungs-Radius, Border-Box 99, 109, 115

absolute Längeneinheiten 43

absolute Positionierung 19, 20, 21

Abstand, Rahmenbild 103

Abstand, Umriss 186

Abstände

Buchstabenabstand 162

Hintergrundbilder 95

Rahmenabstände 113

Wortabstände 222

:active, Pseudoklasse 64

::after, Pseudoelement 68

align-content, Eigenschaft 79

align-items, Eigenschaft 80

align-self, Eigenschaft 81

all, Eigenschaft 81

Alpha (Deckkraft v. Farben) 38

alternative Stylesheets 3

Anführungszeichen, verschachtelte 198

animation, Eigenschaft 82

animation-delay, Eigenschaft 83

animation-direction, Eigenschaft 83
animation-duration, Eigenschaft 84
animation-fill-mode, Eigenschaft 85
animation-iteration-count, Eigenschaft 85
animation-name, Eigenschaft 86
animation-play-state, Eigenschaft 87
animation-timing-function, Eigenschaft 87
animierbare Eigenschaften 77
Aspekt, Wert 142
At-Regeln 6
Attributwerte 48
Ausrichtung
 von Flex-Elementen 22
 Tabellenzellen, Inhalt 35
 Text 205
 vertikal 218
auto, Wert
 bei Block-Level-Layout 15
 bei Grid-Layout 26
 bei Inline-Layout 15
automatisches Tabellenlayout 31, 33

B

backface-visibility, Eigenschaft 88
background, Eigenschaft 88
background-attachment, Eigenschaft 90
background-blend-mode, Eigenschaft 90
background-clip, Eigenschaft 91
background-color, Eigenschaft 92

background-image, Eigenschaft 93
background-origin, Eigenschaft 93
background-position, Eigenschaft 94
background-repeat, Eigenschaft 95
background-size, Eigenschaft 96
::before, Pseudoelement 68
benachbarte Geschwisterelemente, Selektor 52

Bilder

- als Aufzählungszeichen 165
- Float-Formen, Erstellung 200
- Hintergrundbild 93
- Maskierung 173
- Rahmenbild 102, 107

Block-Boxen 12

Blocklayout 14

border, Eigenschaft 97
border-bottom, Eigenschaft 97
border-bottom-color, Eigenschaft 98
border-bottom-left-radius, Eigenschaft 99
border-bottom-right-radius, Eigenschaft 99
border-bottom-style, Eigenschaft 100
border-bottom-width, Eigenschaft 100
border-collapse, Eigenschaft 101
border-color, Eigenschaft 102
border-image, Eigenschaft 102
border-image-outset, Eigenschaft 103
border-image-repeat, Eigenschaft 104
border-image-slice, Eigenschaft 105

border-image-source, Eigenschaft 106
border-image-width, Eigenschaft 106
border-left, Eigenschaft 107
border-left-color, Eigenschaft 108
border-left-style, Eigenschaft 108
border-left-width, Eigenschaft 109
border-radius, Eigenschaft 109
border-right, Eigenschaft 111
border-right-color, Eigenschaft 111
border-right-style, Eigenschaft 112
border-right-width, Eigenschaft 112
border-spacing, Eigenschaft 113
border-style, Eigenschaft 114
border-top, Eigenschaft 114
border-top-color, Eigenschaft 115
border-top-left-radius, Eigenschaft 115
border-top-right-radius, Eigenschaft 116
border-top-style, Eigenschaft 117
border-top-width, Eigenschaft 117
border-width, Eigenschaft 118
bottom, Eigenschaft 119
box-decoration-break, Eigenschaft 119
Box-Layout, flexibel 21, 23
box-shadow, Eigenschaft 120
box-sizing, Eigenschaft 15, 121
Breite
 siehe Größe 36

C

calc(), Werte 48
caption-side, Eigenschaft 122
@charset, Regel 6
:checked, Pseudoklasse 65
clear, Eigenschaft 123
clip-path, Eigenschaft 123
clip-rule, Eigenschaft 124
color, Eigenschaft 40, 125
content, Eigenschaft 126
counter-increment, Eigenschaft 126
counter-reset, Eigenschaft 127
@counter-style, Regel 7
CSS-Variablen 49
currentColor, Schlüsselwort 40
cursor, Eigenschaft 127

D

Darstellungsrollen, für Elemente 12

Dauer

Animation 84

Übergänge (transitions) 215

Deklarationsblock (in Stilregeln) 5

Dekorationen

Box 119

Text 206

direction, Eigenschaft 129

:disabled, Pseudoklasse 65

display, Eigenschaft 12, 129

3D-Transformationen 197

Durchschuss 17

E

Eigenschaften

- universelle Werte 79

- Vererbung und Animation 77

- vollständige Liste 79, 224

- Wertesyntax, Konventionen 77, 79

Eigenschaftswerte (in Stilregeln) 5

einfacher Attribut-Selektor 54

eingebettete Stylesheets

- Präzedenz von Stilen 8, 11

- über 2

Einrückung, Text 207

Element-Boxen 14

Elemente, Klassifizierung 11

Elementlayout, Grundlagen 13, 17

Element-Selektor 51

Em-Boxen 16

Em-Höhe (Längeneinheit) 43

Em-Höhe des Wurzelements (rem-Einheit) 44

:empty, Pseudoklasse 57

empty-cells, Eigenschaft 131

:enabled, Pseudoklasse 65

ersetzte Elemente 12

exakter Attributwert, Selektor 54

externe Stylesheets 2

- Präzedenz von Stilen 8, 11

- über 2

F

Farbe

Hintergrundfarbe 92

Rahmenfarbe 98, 102, 108, 111, 115

Schlüsselwörter 40

Umriss (outline) 185

Farbnotation, hexadezimal paarweise 38

Farbwerte 38, 41

Farbwinkel (Farbe) 39

Feature-Queries 75

festе (fixed) Grid-Tracks 27

festе Positionierung 20

festes Tabellenlayout 30

filter, Eigenschaft 131

:first-child, Pseudoklasse 57

::first-letter, Pseudoelement 69

::first-line, Pseudoelement 69

:first-of-type, Pseudoklasse 58

fit-content, Wert f. Grid-Layout 27

flex, Eigenschaft 132

flex-basis, Eigenschaft 133

Flexbox-Layout 21, 23

flex-direction, Eigenschaft 134

flex-flow, Eigenschaft 135

flex-grow, Eigenschaft 135

flexible Grid-Tracks 27

flex-shrink, Eigenschaft 136

flex-wrap, Eigenschaft 137

float, Eigenschaft 137
Floating 18
:focus, Pseudoklasse 66
font, Eigenschaft 138
@font-face, Regel 7
font-family, Eigenschaft 139
font-feature-settings, Eigenschaft 140
font-kerning, Eigenschaft 141
font-size, Eigenschaft 141
font-size-adjust, Eigenschaft 142
font-stretch, Eigenschaft 143
font-style, Eigenschaft 143
font-synthesis, Eigenschaft 144
font-variant, Eigenschaft 145
font-weight, Eigenschaft 146
Form, Cursor 127
Frequenzen, Werte 46

G

geordnete Listen
siehe die einzelnen list-Eigenschaften 36
Geschwisterelemente, allgemein, Selektor 53
Gewicht, Schrift 146
Grafiken
siehe Bilder 36
grid, Eigenschaft 147
grid-area, Eigenschaft 148
grid-auto-columns, Eigenschaft 148
grid-auto-flow, Eigenschaft 149

grid-auto-rows, Eigenschaft 150
Grid-Bereich 24
grid-column, Eigenschaft 150
grid-column-end, Eigenschaft 151
grid-column-gap, Eigenschaft 152
grid-column-start, Eigenschaft 152
Grid-Container 23
Grid-Elemente 23
grid-gap, Eigenschaft 153
Grid-Layout 23, 28
Grid-Linie 24
grid-row, Eigenschaft 154
grid-row-end, Eigenschaft 154
grid-row-gap, Eigenschaft 155
grid-row-start, Eigenschaft 156
grid-template-areas, Eigenschaft 156
grid-template-columns, Eigenschaft 157
grid-template-rows, Eigenschaft 158
Grid-Track 24
Grid-Zelle 24
Größe
 Außenabstand (margin) 167
 Box 121
 Elemente 159, 220
 Größenänderung v. Elementen, Funktionalität 199
 Hintergrund 96
 Innenabstand (padding) 190
 Maske 176

Maximum, Definition 177
Minimum, Definition 179
Rahmen 100, 109, 112, 117, 118
Rahmenbild 106
Schrift 141
Umriss (outline) 187
Zeilenhöhe 163

Grundlagen der visuellen Darstellung 13, 17

H

»haftende« (sticky) Positionierung 20

Halb-Durchschuss 17

height, Eigenschaft 159

Helligkeit (Farbe) 39

Höhe

siehe Größe 36

horizontale Ausrichtung

siehe Ausrichtung 36

:hover, Pseudoklasse 66

HSL-Farbwerte 39

.htaccess-Datei 4

HTML, Darstellungsrollen 12

HTML, mit Stilen versehen 1, 5

HTML-Elemente, Klassifizierung 11

HTML-Kommentare 8

HTTP Link-Header 4

httpd.conf, Datei 4

hyphens, Eigenschaft 159

I

ID, Selektor 54
Identifizier, Werte 48
@import, Direktive 2
@import, Regel 6
!important, Direktive 9
inherit, Schlüsselwort 37
initial, Schlüsselwort 37
Inline-Boxen 13
Inline-Layout (Inline-Boxen) 15
Inline-Stile 1
Interaktions-Pseudoklassen, Liste 64, 67
intrinsische Grid-Tracks 27
isolation, Eigenschaft 160

J

justify-content, Eigenschaft 161

K

Kaskade, Funktionsweise 8, 10

@keyframes, Regel 7

Kind-Selektor 52

Klassen (class), Selektor 53

Klassifizierung der Elemente 11

Kommentare 8

Konfliktlösung

siehe Kaskade, Funktionsweise 36

Kontext-Selektor 52

L

:lang, Pseudoklasse 58

lang, Attribut 56

<Länge>, Wert 25
Längenangaben
 siehe Größe 36
Längeneinheiten 42, 45
language, Selektor f. Attribut 56
:last-child, Pseudoklasse 58
:last-of-type, Pseudoklasse 59
left, Eigenschaft 161
letter-spacing, Eigenschaft 162
line-break, Eigenschaft 163
line-height, Eigenschaft 16, 163
<link>, Element 3
:link, Pseudoklasse 66
Link-Header (HTTP) 4
Listeneinträge 13
list-style, Eigenschaft 164
list-style-image, Eigenschaft 165
list-style-position, Eigenschaft 165
list-style-type, Eigenschaft 166
M
margin, Eigenschaft 167
margin-bottom, Eigenschaft 168
margin-left, Eigenschaft 169
margin-right, Eigenschaft 169
margin-top, Eigenschaft 170
mask, Eigenschaft 170
mask-clip, Eigenschaft 171
mask-composite, Eigenschaft 172

mask-image, Eigenschaft 173
mask-mode, Eigenschaft 173
mask-origin, Eigenschaft 174
mask-position, Eigenschaft 175
mask-repeat, Eigenschaft 176
mask-size, Eigenschaft 176
mask-type, Eigenschaft 177
max-content, Wert f. Grid-Layout 26
max-height, Eigenschaft 177
max-width, Eigenschaft 178
@media, Regel 7
media, Pseudo-Attribut (xmlstylesheet) 4
Media-Queries 70, 75
 Features 72, 75
 Grundkonzepte 70, 72
 Werte 72, 75
Medien-Attribut (link) 3
Millimeter (mm, Längeneinheit) 42
min-content, Wert f. Grid-Layout 26
min-height, Eigenschaft 179
minmax, Wert f. Grid-Layout 26
min-width, Eigenschaft 179
mix-blend-mode, Eigenschaft 180
N
Nachfahrenelemente, Selektor 52
@namespace, Regel 6
Negation, Pseudoklasse 63
nicht-ersetzte Elemente 11

:not, Pseudoklasse 63
not, Schlüsselwort in Media-Queries 71
:nth-child, Pseudoklasse 59
:nth-last-child, Pseudoklasse 60
:nth-last-of-type, Pseudoklasse 60
:nth-of-type, Pseudoklasse 61
NULL-Breite (Längeneinheit) 44

O

object-fit, Eigenschaft 181
object-position, Eigenschaft 182
only, Schlüsselwort f. Media-Queries 72
:only-child, Pseudoklasse 62
:only-of-type, Pseudoklasse 62
opacity, Eigenschaft 182
order, Eigenschaft 183
orphans, Eigenschaft 184
outline, Eigenschaft 185
outline-color, Eigenschaft 185
outline-offset, Eigenschaft 186
outline-style, Eigenschaft 187
outline-width, Eigenschaft 187
overflow, Eigenschaft 188
overflow-wrap, Eigenschaft 188
overflow-x, Eigenschaft 189
overflow-y, Eigenschaft 190

P

padding, Eigenschaft 190
padding-bottom, Eigenschaft 191

padding-left, Eigenschaft 192
padding-right, Eigenschaft 192
padding-top, Eigenschaft 193
page, Eigenschaft 194
page-break-after, Eigenschaft 194
page-break-before, Eigenschaft 195
page-break-inside, Eigenschaft 196
perspective, Eigenschaft 197
perspective-origin, Eigenschaft 197
Pica (pc, Längeneinheit) 43
Pixel (px, Längeneinheit) 44
Position
 Aufzählungszeichen (Listen) 165
 Hintergrund 94
 Maskierung 175
 Objekt 182
 siehe auch Ursprung 94
 Werte 47
position, Eigenschaft 198
Positionierung 19, 21
Positionierung, v. Elementen 19, 21
Präzedenz von Stilen 8, 11
<Prozentwert> 25
Prozentwerte 41
Pseudoelemente, Liste 68
Pseudoklassen, Liste 56
 Interaktion 64, 67
 Negation 63

strukturell 56, 67

Punkt (pt, Längeneinheit) 43

Q

Quelle, Rahmenbild 106

quotes, Eigenschaft 198

R

Radius, Border-Box 99

Radius, Rahmen-Box 109, 115

Rahmen-Slices, Abstände 105

Regelstruktur in Stylesheets 5

relative Längeneinheiten 43

relative Positionierung 19

repeat, Wert f. Grid-Layout 27

resize, Eigenschaft 199

RGB-Farbnotation 38

Richtung

Animation 83

right, Eigenschaft 200

:root, Pseudoklasse 62

S

Sättigung (Farbe) 39

Schatten

Box 120

Text 209

Schlüsselwörter 37, 78

Schreibrichtung

Grundrichtung 129

Schrumpfungsfaktor 136

Seitenbox, size-Eigenschaft für 203
Selektor (in Stilregeln) 5
Selektoren, vollständige Liste 51, 56
shape-image-threshold, Eigenschaft 200
shape-margin, Eigenschaft 201
shape-outside, Eigenschaft 202
size, Eigenschaft 203
Spalten, Tabelle
 siehe Tabellenlayout 36
Spezifität (Bestimmung d. Stil-Präzedenz) 9
statische Positionierung 19
Stile, hinzufügen 1, 5
Strings 47
Struktur von Stylesheets 5
strukturelle Pseudoklassen, Liste 56
<style>, Element 2
style, Attribut 1
Stylesheets einbinden 3
Stylesheets, Arten 1, 5
Stylesheet-Struktur 5
Substrings am Anfang eines Attributwerts, Selektor 55
Substrings am Ende eines Attributwerts, Selektor 55
Substrings an beliebiger Stelle im Attributwert, Selektor 56
@supports, Regel 7

T

Tabellen-Beschriftungen 122
Tabellenlayout 29, 36
 automatisch 31, 33

- festes (fixed) Layout 30
- vertikale Ausrichtung in Zellen 35
- Zellrahmen, zusammengefasst 33
- Tabellenspalten 36
- Tabellenzellen, leer 131
- table-layout, Eigenschaft 204
- tab-size, Eigenschaft 204
- :target, Pseudoklasse 67
- teilweise Attributwerte, Selektor 55
- Teilwerte (fr) 45
- text-align, Eigenschaft 205
- text-align-last, Eigenschaft 206
- text-decoration, Eigenschaft 206
- text-indent, Eigenschaft 207
- text-orientation, Eigenschaft 208
- text-rendering, Eigenschaft 209
- text-shadow, Eigenschaft 209
- Textstil
 - siehe* Einträge zu font-Eigenschaften, letter-Eigenschaften, word-Eigenschaften
- text-transform, Eigenschaft 210
- top, Eigenschaft 211
- transform, Eigenschaft 212
- Transformierung, Text 212
- transform-origin, Eigenschaft 212
- transform-style, Eigenschaft 213
- transition, Eigenschaft 214
- transition-delay, Eigenschaft 214
- transition-duration, Eigenschaft 215

transition-property, Eigenschaft 216
transition-timing-function, Eigenschaft 216
transparent, Schlüsselwort 41
Typ-Selektor 51

U

umgebender Block, Definition 20
ungeordnete Listen
 siehe Listeneinträge 36
unicode-bidi, Eigenschaft 217
universeller Selektor (*) 51
unset, Schlüsselwort 38
URIs (URI-Werte) 45
Ursprung
 für 3D-Transformationen 197
 für Hintergründe 93
 für Masken 174
 für Transformierungen 212

V

vererbte Eigenschaften 77
Vererbung (bei Stil-Präzedenz) 9
Vererbung von Stilen 9
vertical-align, Eigenschaft 218
vertikale Ausrichtung
 siehe Ausrichtung 36
Verzögerung
 Animation 83
 Übergang (transition) 214
VGA, Farb-Schlüsselwörter 40

Viertel-Millimeter (q, Längeneinheit) 42

Viewport-Breite (vw), Einheit 44

Viewport-Höhe (vh), Einheit 45

Viewport-Maximumeinheit (vmax) 45

Viewport-Mindesteinheit (vmin) 45

visibility, Eigenschaft 218

:visited, Pseudoklasse 67

visuelle Darstellung, Grundlagen 13, 17

W

Wachstumsfaktor 135

Werte 37, 50

- Attributwerte 48

- Berechnungswerte 48

- CSS-Variablen 49

- Farbwerte 38, 41

- Frequenzen 46

- Identifizier 48

- Länge 42, 45

- Media-Queries 72, 75

- Positionen 47

- Prozentwerte 41

- Schlüsselwörter 37

- Strings 47

- Syntaxkonventionen 77, 79

- Teilwerte (fr, fraction) 45

- URI-Werte 45

- Winkel 46

- Zahlen 41

Zeitangaben 46
white-space, Eigenschaft 219
widows, Eigenschaft 220
width, Eigenschaft 220
Wiederholung
 Hintergrund 95
 Maskierungsbild 176
 Rahmenbild 104
Winkeleinheiten 46
word-break, Eigenschaft 221
word-spacing, Eigenschaft 222
writing-mode, Eigenschaft 222

X

x-Höhe (Längeneinheit) 44
xml-styleSheet, Verarbeitungsanweisung 3

Z

Zahlenwerte 41
Zähler dekrementieren 126
Zähler, inkrementieren 126
Zähler, zurücksetzen 127
Zeilen, Tabelle
 siehe Tabellenlayout 36
Zeitangaben, Werte 46
Zellen, Tabelle
 siehe Tabellenlayout 36
Zentimeter (Längeneinheit) 42
z-index, Eigenschaft 223
Zoll (Inch, Längeneinheit) 43

zusammengefassten Zellrahmen, Modell 33