

Was sechs Deklarationen und in der »Übersetzung« einen komplizierten Satz ergibt, entspricht folgender Kurzschreibweise:

```
p {
  font: italic small-caps 700 12px/16px Arial, sans-serif;
}
```

Eine sehr potente Kurzschreibweise – ersetzt sie doch sechs Deklarationen durch eine einzige Zeile! Allerdings wird hier im Gegensatz zu den ersten Beispielen auch die Komplexität von Kurzschreibweisen deutlich, welche gleichzeitig eine höhere Fehleranfälligkeit impliziert.

In der Kurzschreibweise `font` ist ein Parameter für die CSS-Eigenschaft `line-height` zu finden, was nicht intuitiv ist. Zusätzlich ist hier eine spezielle Schreibweise notwendig: Dem Wert für `line-height` muss ein `/` (Schrägstrich) vorangestellt sein, damit ein Browser diese »Zeilenhöhe« als solche erkennt. Die Reihenfolge der Parameter für `font` ist obendrein nicht für alle Werte frei wählbar. Eine ausführliche Beschreibung von `font` finden Sie unter nebenstehendem Link.



CSS-Eigenschaften `font`:  
[www.h5c3.de/link-31-1](http://www.h5c3.de/link-31-1)

Ich schätze, Sie ahnen, in welche Richtung ich Ihren Blick lenken möchte. Gerade beim Einstieg in CSS müssen Sie sich das Leben nicht unnötig schwer machen:

#### Praxistipp

- Kurzschreibweisen sind gut und sinnvoll, solange diese einfach zu lesen sind und keine spezielle Notation oder Reihenfolge erfordern.
- Ist eine Kurzschreibweise sehr komplex und erfordert eine spezielle Syntax oder festgelegte Reihenfolge, sind die einzelnen Eigenschaften vorzuziehen. Das reduziert Fehlerquellen und macht Quellcode einfacher lesbar – und ist damit langfristig auch besser pflegbar.

### 31.2.2 Kurzschreibweisen für die vier Ausrichtungen

Eine andere Art der Kurzschreibweise definiert mit einem Ausdruck die einzelnen Parameter für oben, rechts, unten und links.

```
div {
  padding-top: 20px;
  padding-right: 50px;
  padding-bottom: 20px;
  padding-left: 10px;
}
```

Über diese Regel werden für `div`-Elemente Innenabstände gesteuert, oben und unten von 20 px, rechts von 50 px und links von 10 px.

```
div {
  padding: 20px 50px 20px 10px;
}
```

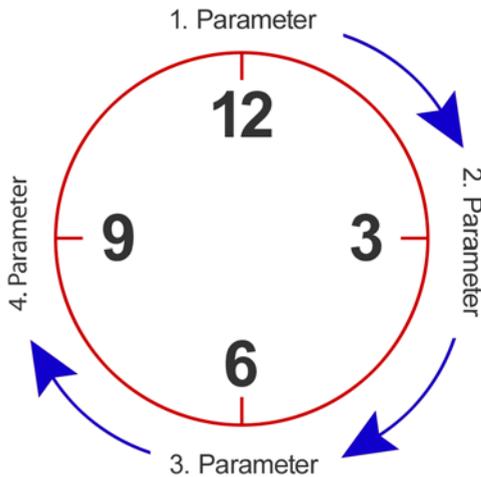


Abb. 31–3

Die einzelnen Werte in der Kurzschreibweise werden ab 12:00 im Uhrzeigersinn angegeben.

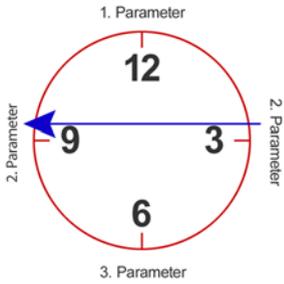
Auch hier ist das Prinzip zu erkennen, dass der gemeinsame Stamm der einzelnen CSS-Eigenschaften die Kurzschreibweise ausmacht.

Im Gegensatz zu den Kurzschreibweisen für verschiedene CSS-Eigenschaften sind diese wenig fehleranfällig. Selbst wenn verschiedene Größeneinheiten (px, em, % usw.) oder unterschiedliche Farbangaben (RGB-Werte, Hexadezimalwerte usw.) gemischt werden, erhöht dies nicht die Fehlerwahrscheinlichkeit, im Gegenteil: CSS-Quelltexte bleiben hierdurch wesentlich übersichtlicher.

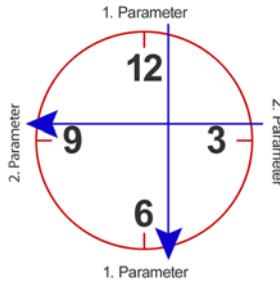
Ein Augenmerk sei an dieser Stelle auf die automatische Vervollständigung der einzelnen Werte gesetzt. Kurzschreibweisen für die vier Richtungen erwarten einen bis vier Parameter. Die Vervollständigung fehlender Parameter findet nach folgenden Regeln statt: Fehlt der 4. Parameter, wird der 2. Parameter für die 4. Position übernommen; fehlt außerdem der 3. Parameter, wird der 1. Parameter für die 3. Position übernommen. Mit anderen Worten: Fehlende Werte werden von der gegenüberliegenden Seite übernommen. Ist nur ein Parameter angegeben, wird dieser für alle Dimensionen gesetzt.

Werte werden automatisch »aufgefüllt«.

`padding: 10px 20px 40px;`



`padding: 10px 20px;`



`padding: 10px;`

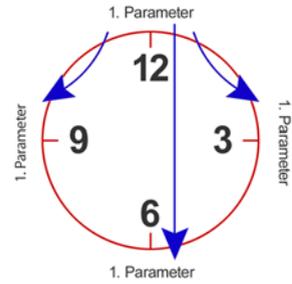


Abb. 31-4

Automatische Wertübername bei drei, zwei und einem Wert

### 31.2.3 Zusammenspiel von Kurzschreibweisen und einzelnen Eigenschaften

In der Kombination von Einzel- und Kurzschreibweisen können so effektive Styles aufgebaut werden.

Abb. 31-5

Mehrere CSS-Regeln werden browserintern »zusammengerechnet«.

```
P {
  border: 8px solid blue;
}
P {
  border-left-color: red;
  border-bottom-style: dotted;
}
```



Über Kurzschreibweisen können Sie generelle Anweisungen formulieren. Soll nur ein einzelner Parameter verändert werden, ist es nicht notwendig, die komplette Regel neu zu definieren. Über speziellere Eigenschaften können einzelne Aspekte der allgemeinen Regel »überschrieben« werden. In diesem Beispiel wird der Rahmen links über `border-left-color` rot, der untere Rahmen per `border-bottom-style` gepunktet dargestellt, ohne dass die anderen Anweisungen der Kurzschreibweise davon betroffen sind. Dieses Beispiel zeigt gleichzeitig eine typische Herangehensweise, ganz nach der Devise: »So allgemein wie möglich, so speziell wie nötig.«

### 31.3 @-Regeln

Eine besondere Art der CSS-Regel stellen @-Regeln (im Englischen »At-rules«) dar. Hier wird das @-Zeichen mit bestimmten Schlüsselwörtern kombiniert und steuert besondere Funktionen: beispielsweise das Einbinden von CSS-Dateien aus CSS-Dateien heraus (@import), das Eingrenzen eines CSS-Abschnitts für bestimmte Medientypen (@media) oder das Einbinden von Schriften (@font-face).

Im Einzelnen sind diese Regeln in späteren Kapiteln beschrieben.

### 31.4 CSS-Regeln zusammengefasst

CSS-Regeln können einfach sein, aber auch zu sehr komplexen Gebilden werden. Die allgemeinen Anforderungen an CSS-Regeln sind hier noch mal zusammengefasst:

- Eigenschaft und Wert werden durch einen Doppelpunkt getrennt, dadurch wird der Eigenschaft der Wert zugewiesen.
- Werte, auch Schlüsselwörter werden bis auf wenige Ausnahmen ohne Anführungszeichen im CSS-Code angegeben.
- Mehrere Deklarationen werden durch ein Semikolon voneinander getrennt.
- Hinter der letzten Deklaration und vor der schließenden geschweiften Klammer ist kein Semikolon notwendig.
- Leerzeichen und Zeilenumbrüche vor und hinter Selektoren, den CSS-Eigenschaften und den Werten sind optional, dienen jedoch dazu, CSS-Code übersichtlich zu gestalten.
- Einfachselektoren müssen durch Leerzeichen getrennt werden.
- Mehrere Einzelwerte in einem Werteblock werden ebenfalls mit Leerzeichen voneinander getrennt.
- Mehrere Werteblocke werden durch ein Komma voneinander getrennt.
- CSS-Regeln dürfen nicht verschachtelt werden. Eine Ausnahme stellt die @media-Regel dar.