

Überschriften, Abbildungen und Tabellen in einem HTML-Text automatisch nummerieren

Texte mit der CSS-Funktion "counter ()" strukturieren

Auch wenn kurze [knackige Texte](#) mit vielen Bildern, Videos und Animationen heute auf Internetseiten vorherrschen, gibt es immer noch Anlässe für längere Texte auf Webseiten: Wenn es z. B. um den Kern eines Produkts geht, um das Wesentliche einer Dienstleistung, um das Alleinstellungsmerkmal eines Unternehmens oder – ja, auch das gibt es noch – um wissenschaftliche Texte geht, darf ein Text im Internet gerne auch mal etwas umfangreicher ausfallen als allgemein vorgeschlagen wird – wenn [die Qualität](#) stimmt.

Durchdachte Strukturierung

Längere Texte sollten gut strukturiert sein durch

- Überschriften
- Nummerierung der Überschriften (z. B. 1. Kapitel, 1.1. Kapitel, 1.2. Kapitel usw.)
- typografische Auflockerungen wie Listen oder Textboxen
- Abbildungen und Tabellen

Wie Texte im Web aufgebaut werden sollten, habe ich in einem [früheren Artikel](#) bereits beschrieben.

Wäre es nicht schön, wenn sich die Nummerierung von Überschriften, Abbildungen und Tabellen automatisch anpassen, wenn neue Überschriften oder Bilder eingefügt werden?

Genau das geht mit ein wenig HTML und CSS.

Dazu verwende ich die CSS-Funktion `counter ()` und das [Pseudo-Element](#) `::before`. Vielmehr braucht es nicht – natürlich außer ein wenig Code drumherum 😊. Die CSS-Funktion `counter ()` wird von allen gängigen [Browsern unterstützt](#).

Komplexe Nummerierung eines HTML-Textes

Im Folgenden erstelle ich ein Template für die Nummerierung eines Textes mit Überschriften, Abbildungen und Tabellen, das für jeden beliebigen Text verwendet werden kann. Zunächst möchte ich die CSS-Funktion `counter ()` etwas genauer betrachten.

Die CSS-Funktion "counter ()" unter der Lupe

Ein Counter allein hat erst einmal keinerlei Auswirkung auf den HTML-Code einer Website. Erst mithilfe der Counter-Funktion können die Werte eines Counters ausgegeben werden. Sehen wir uns das einmal genauer an:

Die CSS-Funktion `counter ()` wird vorwiegend in den Pseudo-Elementen `::after` und `::before` angewendet. Um die Counter-Funktion zu initialisieren, muss zuvor ein Name für den Counter (Zähler) festgelegt werden. Dies geschieht immer in einem umfassenden HTML-Element; das kann das Body-Element, das Main-Element oder ein Div-Container sein.

Wenn wir z. B. eine nummerierte Liste mit den HTML-Elementen `ol` und `li` vor uns haben, wird der Name der Funktion im CSS-Selektor `ol` festgelegt:

```
ol { counter-reset: count; }
```

Die Eigenschaft `counter-reset` hat den Wert `count`. Dieser Wert ist der Name des Counters, in unserem Beispiel also `count`; der Name kann beliebig vergeben werden. `counter-reset` bewirkt zudem, dass die Nummerierung auf Null gesetzt wird.

Hinweis

Bei Listen entfernt man den standardmäßigen `list-style-type` entweder durch

```
ol { list-style-type: none; } oder durch
```

```
li { display: block; }.
```

Nun kann man die Nummerierung der Listenelemente frei bearbeiten.

Damit nun die Nummerierung initiiert werden kann, wird im CSS-Code das Listenelement `li` angesprochen und der Counter-Wert inkrementiert, also schrittweise erhöht:

```
li { counter-increment: count; }
```

Das bedeutet, dass der Counter mit Namen `count` nun in den Listen-Elementen schrittweise um 1 erhöht wird, also von 0, was durch den Counter-Reset festgelegt wurde, auf 1 und dann auf 2 usw.

Will man stattdessen in Zweierschritten erhöhen, lautet der Code:

```
li { counter-increment: count 2; } (Auch negative Werte sind erlaubt.)
```

Nun kann man die Nummerierung gestalten und erweitern. Dies geschieht im vorliegenden Beispiel im Pseudo-Element `::before`, in dem die CSS-Funktion `counter()` in die Eigenschaft `content` eingefügt wird (und nur hier funktioniert die Funktion bislang auch browserübergreifend, [siehe hier](#)).

Die CSS-Funktion `counter()` ist folgendermaßen aufgebaut:

```
li::before { content: counter(count); }
```

Der Aufbau der CSS-Anweisung allgemein ausgedrückt:

```
Selektor (= li::before) { Eigenschaft (= content): Wert (= counter(counter-name)); }
```

Um die Nummerierung zu ändern – z. B. in Großbuchstaben –, verwendet man die Funktion so:

```
li::before { content: counter(count, upper-latin); }
```

Es ist auch möglich, Wörter, Zeichen, Bilder und Symbole einzufügen:

```
li::before { content: "Kapitel " counter(count, upper-latin) " \n2606"; }
```

Vor den Counter (A, B, C ...) wird nun das Wort *Kapitel* ausgegeben und danach das Sternsymbol:

Kapitel A ☆

Im nächsten Schritt wende ich den Counter für Überschriften, Abbildungen und Tabellen an.

Nummerierung mit der CSS-Counter-Funktion und dem Pseudo-Element `::before`

Die Anforderungen

Stellen wir uns vor, wir möchten einen Text mit sechs Überschriftsebenen, die auch verschachtelt

sein sollen, und mit Abbildungen sowie Tabellen erstellen. Die Nummerierung soll auf allen Ebenen automatisch vergeben werden nach dem Schema:

A. 1. 2. 3. 4. 5. bei den Überschriften, wobei A. für die Überschrift der ersten Ebene und 5. für die Überschrift der sechsten Ebene steht; die Abbildungen und Tabellen sollen durchgehend nummeriert werden.

Weitere Möglichkeiten zur Nummerierung finden Sie [hier](#).

Die Nummerierung der Überschriften

Hinweis

Ich empfehle für die Praxis die Verwendung von CSS-Klassen statt der Elementbezeichnungen, also statt `h1` die Klasse `.heading-1` usw. Zur besseren Übersicht bleibe ich hier bei `h1`, `h2` ...

Zuerst starte ich die Inkrementierung für jede Überschriftsebene mit diesem Code:

```
h1 { counter-increment: h1; } ... h6 { counter-increment: h6; }
```

Anschließend setze ich mit folgendem Code alle Überschriftsebenen auf Null, damit die Nummerierung bei 1 beginnen kann (d. h. wenn eine Ebene nicht vorhanden ist, wird eine Null angezeigt):

Im Body als umfassendes Element wird der Counter initiiert:

```
body { counter-reset: h1 h2 h3 h4 h5 h6; }
```

Und wird dann für jede Überschriftenebene fortgeführt, damit die Nummerierung für jede Ebene stets neu beginnt:

```
h1 { counter-reset: h2 h3 h4 h5 h6; }  
h2 { counter-reset: h3 h4 h5 h6; }
```

bis hin zu

```
h5 { counter-reset: h6; }
```

Wichtig ist, dass nur ein Leerzeichen und kein Komma zwischen die Werte `h1` bis `h6` gesetzt wird.

Für die oberste Ebene `h1` habe ich nun diesen Code:

```
h1 { counter-increment: h1; counter-reset: h2 h3 h4 h5 h6; }
```

Um die Nummerierung sichtbar zu machen, füge ich vor die Überschrift die CSS-Counterfunktion `counter()` ein:

```
h1::before { content: counter(h1, upper-alpha) ". "; }
```

Das ergibt für die erste Überschrift `h1` im Text A. und für die zweite Überschrift der Ebene `h1` B.

.

Bei jeder Ebene wird nun der Counter der übergeordneten Ebene(n) eingefügt und zusätzlich der Counter dieser Ebene:

```
h2::before { content: counter(h1, upper-alpha) ". "  
counter(h2, decimal) ". "; }
```

Das Ergebnis ist: A. 1.

Die unterste Ebene `h6` weist dementsprechend folgenden Code auf:

```
h6::before { content: counter(h1,upper-alpha) ". "
counter(h2,decimal) ". " counter(h3,decimal) ". "
counter(h4,decimal) ". " counter(h5,decimal) ". "
counter(h6,decimal) ". "; }
```

Und das war es auch schon mit den Überschriften!

Nummerierung der Abbildungen und Tabellen

Analog zum Vorgehen bei den Überschriften initiiere ich den Counter in einem umfassenden Element. Da das Body-Element bereits für die Überschriften reserviert ist, bietet sich für die Tabellen z. B. das Element `main` an. Für die Abbildungen erstelle ich einen Container mit der Klasse `.wrapper`.

Die Nummerierung soll in der Beschriftung der Abbildungen und Tabellen auftauchen. Dazu benötige ich folgenden Code:

Tabellen

Für die Tabellenbeschreibung verwende ich die Klasse `.table-subline` in einem Absatz `p`, um nicht mit den Table-Elementen `tfoot` oder `caption` zu kollidieren. Ich wähle "table-counter" als Namen für die Counter-Funktion:

```
main { counter-reset: table-counter; }
p.table-subline { counter-increment: table-counter; }
p.table-subline::before { content: "Tab. " counter(table-counter)
": "; }
```

Die Ausgabe sieht nun so aus: *Tab. 1:* .

Abbildungen

Für die Abbildungen verwende ich das HTML-Element `figcaption` und wähle "fig-counter" als Namen für die Counter-Funktion:

```
.wrapper { counter-reset: fig-counter; }
figcaption { counter-increment: fig-counter; }
figcaption::before { content: "Fig. " counter(fig-counter) ": "; }
```

Die Ausgabe sieht nun so aus: *Fig. 1:* .

Überschriften, Abbildungen und Tabellen zusammengefügt

Das Ergebnis können Sie sich auf dieser [Demo-Seite](#) ansehen. Sie können die Vorlage auch selbst bearbeiten. Dafür habe ich für Sie das Template auf [codepen.io](#) eingestellt.

Hinweis

Im nächsten Artikel werde ich ein praktisches Beispiel für die Anwendung der CSS-Counterfunktion entwickeln: **Ein Template für eine Speisekarte!**

Links

Mehr Informationen und viele Beispiele für die Anwendung der Counter-Funktion finden Sie auf den Websites des World Wide Web Consortium (W3C) [hier](#) und [hier](#) und auf der Website von Mozilla Developer [hier](#) und [hier](#).

Von Rachel Andrew gibt es einen lesenswerten Artikel zum Thema im [Smashing Magazine](#).

Zur Browserunterstützung der Counter-Funktion siehe [caniuse.com](#).