

Cascading Stylesheets – Kombination von Selektoren

In Stylesheets werden Selektoren oft miteinander kombiniert um eine genauere Auswahl der HTML-Elemente zu ermöglichen. Folgende Kombinationsmöglichkeiten gibt es:

Selektor1, selektor2	Die Formatierung gilt sowohl für alle Elemente, die durch selektor1 als auch die durch selektor2 ausgewählt sind.
Selektor1selektor2 (ohne Leerzeichen)	Die Formatierung gilt für alle Elemente, die durch Selektor1 ausgewählt wurden, die aber gleichzeitig selektor2 entsprechen. Z.B.: <code>div.format1</code> → Alle DIV-Elemente mit der Klasse <code>format1</code>

Kombinatoren:

Nachfahrselektor: selektor1 selektor2 (Leerzeichen zwischen den Selektoren)	Alle Unterelemente in dem durch selektor1 ausgewählten Element mit der durch selektor2 festgelegten Eigenschaft werden formatiert.
Kindselektor: selektor1>selektor2	Nur die Kindelemente in dem durch selektor1 ausgewählten Element mit der durch selektor2 festgelegten Eigenschaft werden formatiert. Die Unterelemente der Kinder werden nicht formatiert (Enkel...)
Nachbarselektor: selektor1+selektor2	Das direkt nach dem durch selektor1 ausgewählte Element mit der durch selektor2 festgelegten Eigenschaft wird formatiert.
Geschwisterselektor selektor1~selektor2	Alle Nachbarelemente des durch selektor1 ausgewählten Elements mit der durch selektor2 festgelegten Eigenschaft wird formatiert.

Pseudoklassen

Pseudoklassen können an einen Selektor angehängt werden um die Selektion auf einen speziellen Zustand einzuschränken. Beispiele für Pseudoklassen sind:

Tabelle aus w3schools (https://www.w3schools.com/css/css_pseudo_classes.asp)

Selektor	Beispiel	Beschreibung des Beispiels
:active	<code>a:active</code>	Selects the active link
:checked	<code>input:checked</code>	Selects every checked <input> element
:disabled	<code>input:disabled</code>	Selects every disabled <input> element
:empty	<code>p:empty</code>	Selects every <p> element that has no children

:enabled	input:enabled	Selects every enabled <input> element
:first-child	p:first-child	Selects every <p> elements that is the first child of its parent
:first-of-type	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
:focus	input:focus	Selects the <input> element that has focus
:hover	a:hover	Selects links on mouse over
:in-range	input:in-range	Selects <input> elements with a value within a specified range
:invalid	input:invalid	Selects all <input> elements with an invalid value
:lang(language)	p:lang(it)	Selects every <p> element with a lang attribute value starting with "it"
:last-child	p:last-child	Selects every <p> elements that is the last child of its parent
:last-of-type	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
:link	a:link	Selects all unvisited links
:not(selector)	:not(p)	Selects every element that is not a <p> element
:nth-child(n)	p:nth-child(2)	Selects every <p> element that is the second child of its parent
:nth-last-child(n)	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
:nth-last-of-type(n)	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
:nth-of-type(n)	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
:only-of-type	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
	p:only-child	Selects every <p> element that is the only child of its parent
:visited	a:visited	Selects all visited links

Pseudoelemente

Mit Pseudoelementen wird ein bestimmter Teil einer Selektion formatiert.

Selektor	Beispiel	Beschreibung des Beispiels
::after	p::after	Insert content after every <p> element
::before	p::before	Insert content before every <p> element
::first-letter	p::first-letter	Selects the first letter of every <p> element
::first-line	p::first-line	Selects the first line of every <p> element
::selection	p::selection	Selects the portion of an element that is selected by a user

Gewichtung von Selektoren (Kaskadierung)

[CSS/Tutorials/Einstieg/Kaskade – SELFHTML-Wiki](#)

Wenn das Attribut eines HTML-Elementes mehrfach definiert wird, überschreiben sich die Definitionen. Dabei werden Kombinationen von Selektoren unterschiedlich gewichtet.

Bei gleich gewichteten Selektoren werden die im Stylesheet zuletzt definierten Eigenschaften übernommen.

Die Spezifität wird durch die Betrachtung der Bestandteile eines Selektors ermittelt. Für jeden Selektor werden drei Zähler (A, B und C) mit dem Startwert Null festgelegt. Jeder Bestandteil eines Selektors wirkt sich auf diese Zähler aus:

- Der Zähler A wird durch jedes Vorkommen eines ID-Selektors um eins erhöht.
- Der Zähler B wird durch jedes Vorkommen eines Attribut- oder Klassenselektors bzw. einer Pseudoklasse um eins erhöht.
- Der Zähler C wird durch jedes Vorkommen eines Typselektors oder eines Pseudoelements um eins erhöht.
- Der Universalselektor verhält sich neutral, er wird ignoriert.

Beim Vergleich werden zuerst die Werte in Zähler A verglichen. Der höhere Wert entscheidet. Sind beide Werte gleich, entscheidet der Zähler B ansonsten der Zähler C.

Im Browser-Inspector und in VS-Code können Sie sich im Stylesheet die Spezifität anzeigen lassen, indem Sie mit der Maus über den Selektor gehen.

