CSS Combinators

A combinator is something that explains the relationship between the selectors.

A CSS selector can contain more than one simple selector. Between the simple selectors, we can include a combinator.

There are four different combinators in CSS3:

[Adjacent sibling selectors](https://developer.mozilla.org/en-US/docs/Web/CSS/Adjacent_sibling_selectors)

The '+' combinator selects nodes that immediately follow the former specified element.  
Syntax: *A* + *B*  
Example: div + p will match any [<p>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/p) that immediately follows a [<div>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/div).

[General sibling selectors](https://developer.mozilla.org/en-US/docs/Web/CSS/General_sibling_selectors)

The '~' combinator selects nodes that follow (not necessarily immediately) the former specified element, if both elements shared the same parent.  
Syntax: *A* ~ *B*  
Example: p ~ span will match all [<span>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/span) elements that follow a [<p>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/p) element inside the same element.

[Child selectors](https://developer.mozilla.org/en-US/docs/Web/CSS/Child_selectors)

The '>' combinator selects nodes that are direct children of the former specified element.  
Syntax: *A* > *B*  
Example: ul > li will match all [<li>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/li) elements that are inside a [<ul>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ul) element.

[Descendant selectors](https://developer.mozilla.org/en-US/docs/Web/CSS/Descendant_selectors)

The ' ' combinator selects nodes that are children (not necessary direct children) of the former specified element.  
Syntax: A B  
Example: div span will match any [<span>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/span) element that is inside a [<div>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/div) element.

Pseudo Element

PSEUDO-ELEMENTS are used to address sub-parts of elements. They allow you to set style on a part of an element's content beyond what is specified in the documents. In other words they allow logical elements to be defined which are not actually in the document element tree. Logical elements allow implied semantic structure to be addressed in CSS selectors.

Pseudo-elements may only be applied to external and document-level contexts - not to in-line styles. Pseudo-elements are restricted in where they can appear in a rule. They may only appear at the end of a selector chain (after the subject of the selector). They should come after any class or ID names found in the selector. Only one pseudo-element can be specified per selector. To address multiple pseudo-elements on a single element structure, multiple style selector/declaration statements must be made.

Pseudo-elements:

* [::after](https://developer.mozilla.org/en-US/docs/Web/CSS/::after)

::after pseudo-element matches a virtual last child of the selected element. It is typically used to add cosmetic content to an element by using the content CSS property. This element is inline by default.

* [::before](https://developer.mozilla.org/en-US/docs/Web/CSS/::before)

::before creates a pseudo-element that is the first child of the element matched. It is often used to add cosmetic content to an element by using the content property. This element is inline by default.

* [::first-letter](https://developer.mozilla.org/en-US/docs/Web/CSS/::first-letter)

The ::first-letter CSS pseudo-element selects the first letter of the first line of a block, if it is not preceded by any other content (such as images or inline tables) on its line.

* ::first-line

The **::first-line** CSS pseudo-element applies styles only to the first line of an element. The amount of the text on the first line depends of numerous factors, like the width of the element, width of the document, and the font size of the text. As all pseudo-elements, ::first-line does not match any real HTML element.

A first line has only meaning in a block-container box, therefore the ::first-line pseudo-element has only an effect on elements with a display value of block, inline-block, table-cell or table-caption. In all other cases, ::first-line has no effect.

<https://developer.mozilla.org/en-US/docs/Web/CSS/>

<https://www.w3schools.com>

https://www.w3.org