CSS selectors

CSS selectors define the pattern to select elements to which a set of CSS rules are then applied.

CSS selectors can be grouped into the following categories based on the type of elements they can select.

Basic selectors

Universal selector

Selects all elements. Optionally, it may be restricted to a specific namespace or to all namespaces.

Syntax: `*` `ns|*` `*|*`

Example: `*` will match all the elements of the document.

Type selector

Selects all elements that have the given node name.

Syntax: `elementname`

Example: `input` will match any `<input>` element.

Class selector

Selects all elements that have the given `class` attribute.

Syntax: `.classname`

Example: `.index` will match any element that has `class="index"`.

ID selector

Selects an element based on the value of its `id` attribute. There should be only one element with a given ID in a document.

Syntax: `#idname`

Example: `#toc` will match the element that has `id="toc"`.

Attribute selector

Selects all elements that have the given attribute.

Syntax: `[attr]` `[attr=value]` `[attr~=value]` `[attr|=value]` `[attr^=value]` `[attr$=value]` `[attr*=value]`

Example: `[autoplay]` will match all elements that have the `autoplay` attribute set (to any value).

Grouping selectors

Selector list

mdn web docs

Syntax: `A, B`

Example: `div, span` will match both `<span>` and `<div>` elements.

Combinators

Descendant combinator

The " " (space) combinator selects nodes that are descendants of the first element.

Syntax: `A B`

Example: `div span` will match all `<span>` elements that are inside a `<div>` element.

Child combinator

The `>` combinator selects nodes that are direct children of the first element.

Syntax: `A > B`

Example: `ul > li` will match all `<li>` elements that are nested directly inside a `<ul>` element.

General sibling combinator

The `~` combinator selects siblings. This means that the second element follows the first (though not necessarily immediately), and both share the same parent.

Syntax: `A ~ B`

Example: `p ~ span` will match all `<span>` elements that follow a `<p>`, immediately or not.

Adjacent sibling combinator

The `+` combinator matches the second element only if it *immediately* follows the first element.

Syntax: `A + B`

Example: `h2 + p` will match the first `<p>` element that *immediately* follows an h2 element.

Column combinator

The `||` combinator selects nodes which belong to a column.

Syntax: `A || B`

Example: `col || td` will match all `<td>` elements that belong to the scope of the `<col>`.

Pseudo-classes and pseudo-elements

Pseudo classes

The `:` pseudo allow the selection of elements based on state information that is not contained in the document tree.

Example: `a:visited` will match all `<a>` elements that have been visited by the user.

Pseudo elements

The `::` pseudo represent entities that are not included in HTML.

Example: `p::first-line` will match the first line of all `<p>` elements.

Structure of a selector

The term 'selector' can refer to one of the following:

Simple selector

A selector with a single component, such as a single id selector or type selector, that's not used in combination with or contains any other selector component or combinator. A given element is said to match a simple selector when that simple selector accurately describes the element. All basic selectors, attributes, and single pseudo-classes and pseudo-elements are simple selectors.

Compound selector

A sequence of simple selectors that are not separated by a combinator. A compound selector represents a set of simultaneous conditions on a single element. A given element is said to match a compound selector when the element matches all the simple selectors in the compound selector.

In a compound selector, the type selector or a universal selector in a compound selector must come first in the sequence of selectors. Only one type selector or universal selector is allowed in the sequence. Since whitespace represents the descendant combinator, no whitespace is allowed between the simple selectors in a compound selector.

Example: `a#selected {...}`

Complex selector

A sequence of one or more simple and/or compound selectors that are separated by combinators. A complex selector represents a set of simultaneous conditions on a set of elements. These conditions apply in the context of relationships described by the combinators. A given element is said to match a complex selector when the element matches compound selectors and the combinators between the compound selectors.

Examples: `a#selected > .icon {...}`, `.box h2 + p {...}`, `a .icon {...}`

Relative selector

A selector representing an element relative to one or more anchor elements preceded by a combinator. Relative selectors that don't begin with an explicit combinator have an implied descendant combinator.

Examples: `+ div#topic > #reference {...}`, `> .icon {...}`, `dt:has(+ img) ~ dd {...}`

Selector list

A comma-separated list of simple, compound, or complex selectors. If the constituent selector type of a selector list is important but unspecified, it is called a *complex selector list*. A given element is said to match a selector list when the element matches any (at least one) of the selectors in that selector list. Read more about when a selector list is deemed invalid and how to construct a forgiving selector list.

Example: `#main, article.heading {...}`

Specifications

Specification
Selectors Level 4

See the pseudo-class and pseudo-element specification tables for details on those.

See also

- `:has()` pseudo class
- CSS Specificity
- Selector list

This page was last modified on Feb 22, 2023 by MDN contributors.