**Pseudo selectores**

[Pseudoclases y pseudoelementos - Aprende sobre desarrollo web](https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Selectores_CSS/Pseudo-clases_y_pseudo-elementos)

Se llaman así por no seleccionar elementos, sino ciertas partes de estos o solo bajo determinadas circunstancias. Hay de dos tipos: pseudo-clases y pseudo-elementos.

:*pseudo-class*

*::pseudo-elemento*

[Selectors Level 4](https://drafts.csswg.org/selectors-4/)

<https://drafts.csswg.org/selectors-4/>

[Editor’s Draft](https://www.w3.org/standards/types#ED), 24 August 2022

**Pseudoclases**

[CSS Selectors Reference](http://www.w3schools.com/cssref/css_selectors.asp)

[CSS Selector Tester](http://www.w3schools.com/cssref/trysel.asp)

Una [pseudo-clase](https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes) CSS consta de una clave precedida de dos puntos (:) que añadiremos al final del selector para indicar que daremos estilo a los elementos seleccionados solo cuando estos se encuentren en un estado determinado. Por ejemplo, podríamos querer dar estilo a un elemento cuando este se muestre al pasarle el puntero del ratón, o una caja de selección al estar habilitada o deshabilitada o cuando un elemento es hijo directo de su padre en el árbol DOM.

Muchos elementos del lenguaje de marcas HTML disponen de estados especiales o usos asociados a ellos. Podemos aplicar estilos diferentes a estos elementos basándonos en esos estados o usos en nuestra hojas de estilo CSS.

Las pseudo clases se utilizan para añadir efectos a algunos selectores.

**selector:pseudo-clase {propiedad:valor;}**

También se pueden usar clases con las pseudo-clases:

**selector.class:pseudo-clase {propiedad:valor;}**

a.rojo:hover {color:red;}

a.verde:hover {color:green;}  
  
<a class="rojo" href="http://www.google.es">Google</a>

<a class="verde" href="http://www.yahoo.es">Yahoo</a>

Por ejemplo, el elemento <a> tiene cuatro estados:

* **normal**: Su estado normal
* **visited**: Cuando ya hemos visitado el link al que hace referencia
* **hover**: Cuando tenemos el cursor situado encima del mismo
* **active**: Cuando hacemos click sobre él

Una pseudo clase es un estado o uso predefinido de un elemento al que se le puede aplicar un estilo independientemente de su estado por defecto. Existen cuatro tipos diferentes de pseudo clases:

* **Links (:link y :visited)**: Estas pseudoclases se usan para dar estilo al enlace tanto en su estado normal por defecto como cuando ya ha sido visitado, mientras mantenemos el cursor encima de él o cuando hacemos click en él
* **Dinámicas (o de acción de usuario)**: Estas pseudoclases pueden ser aplicadas a cualquier elemento para definir cómo se muestran cuando el cursor está situado sobre ellos, o haciendo click en ellos o bien cuando son seleccionados
* **Estructurales**: Permiten dar estilo a elementos basándonos en una posición numérica exacta del elemento
* **Otras**: Algunos elementos pueden ser estilizados de manera diferente basándonos en el lenguaje o qué tipo de etiqueta no son

**Enlaces <a>**

CSS permite definir las propiedades de color, texto, fuente de un enlace de forma diferente dependiendo del estado del mismo, es decir, si el enlace se ha visitado, no se ha visitado, si es el enlace activo o si el cursor está sobre dicho enlace. Esto permite añadir efectos a un sitio web. Para controlar estos efectos se usan lo que se ha denominado como pseudo-clases.

Una pseudo-clase te permite tener en cuenta diferentes condiciones o eventos al definir una propiedad para una etiqueta HTML.

**a { color: blue; }**

Todo enlace tiene diferentes estados. Por ejemplo, visitado o no visitado. Puedes usar una pseudo-clase para asignar diferentes estilos a los enlaces visitados y no visitados.

**a:link { color: blue; }**

**a:visited { color: red; }**

Usa a:link y a:visited para enlaces visitados y no visitados, respectivamente. A los enlaces activos se les aplica la pseudo-clase a:active, y a:hover cuando el cursor se coloca o pasa sobre el enlace.

**a:hover debe aparecer después de a:link y a:visited**

a:active debe aparecer después de a:hover

* + 1. **Pseudo-clase a:link**

La pseudoclase :link se usa para enlaces que dirigen a páginas que el usuario no ha visitado. En el ejemplo de código que sigue, los enlaces no visitados tendrán un color azul claro.

**a:link { color: #6699CC; }**

* + 1. **Pseudo-clase a:visited**

La pseudoclase :visited se usa para enlaces que dirigen a páginas que el usuario ya ha visitado. Por ejemplo, el código siguiente hará que todos los enlaces visitados sean de color púrpura oscuro:

**a:visited { color: #660099; }**

* + 1. **Pseudo-clase a:any-link**
    2. [:any-link - CSS | MDN](https://developer.mozilla.org/es/docs/Web/CSS/:any-link)
    3. Coincide con todos los elementos que coincidan con [:link](https://developer.mozilla.org/es/docs/Web/CSS/:link) o [:visited](https://developer.mozilla.org/es/docs/Web/CSS/:visited).

**:any-link /\* Selecciona cualquier elemento que coincida con :link o :visited \*/**

* + 1. **Pseudo-clase a:active**

La pseudoclase :active se usa para enlaces que están activos.

El código de este ejemplo hace que el color de fondo para los enlaces activos sea amarillo:

**a:active { background-color: #FFFF00; }**

* + 1. **Pseudo-clase a:hover**

La pseudoclase a:hover se usa cuando el puntero del ratón pasa por encima de un enlace.

Esta pseudo-clase se puede usar para crear efectos interesantes. Por ejemplo, si queremos que nuestros enlaces sean de color naranja y estén en cursiva cuando el cursor pase sobre ellos, el código CSS que utilizaremos será el siguiente:

**a:hover { color: orange; font-style: italic; }**

**Ejemplos de estilos:**

**a {text-decoration:none; color:#000000}**

**a:link {}**

**a:visited {}**

**a:hover {color:#0000cc}**

**a:active {color:#0000cc}**

**Efecto cuando el cursor está encima de un enlace**

El espaciado entre los caracteres se puede ajustar usando la propiedad letter-spacing. Esta propiedad se puede aplicar a los enlaces para crear un efecto especial:

**a:hover { letter-spacing: 10px; font-weight:bold; color:red; }**

La propiedad text-transform, que sirve para intercambiar entre letras en mayúscula y minúscula. Esto se puede usar también para crear un determinado efecto en los enlaces:

**a:hover {**

**text-transform: uppercase;**

**font-weight:bold;**

**color:blue;**

**background-color:yellow;**

**}**

**a { text-decoration:none; }**

De forma alternativa, puedes establecer la propiedad text-decoration junto con otras propiedades para las cuatro pseudo clases.

**a:link {**

**color: blue;**

**text-decoration:none;**

**}**

**a:visited {**

**color: purple;**

**text-decoration:none;**

**}**

**a:active {**

**background-color: yellow;**

**text-decoration:none;**

**}**

**a:hover {**

**color:red;**

**text-decoration:none;**

**}**

Las pseudoclases se pueden usar con otras etiquetas además de con <a>,salvo en algunos navegadores antiguos.

[3.4. Pseudo-clases (CSS avanzado)](https://uniwebsidad.com/libros/css-avanzado/capitulo-3/pseudo-clases)

**Pseudoclases estructurales**

Algunas veces, los diseñadores necesitan dar estilo al primer elemento nodo hijo de otro elemento. Esto es muy habitual por ejemplo con las listas. El pseudo elemento :first-child ha estado disponible desde CSS2, sin embargo CSS3 ofrece mucha más potencia a la hora de trabajar con este tipo de problema.

* + 1. **:first-child**

Fija el aspecto del primer elemento de un tipo de selector si es el primer nodo hijo de su elemento padre.

li:first-child {

color: red;

}

* 1. Primer hijo de cada ul:

ul>:first-child  
{  
background:yellow;  
}

* + 1. **:first-of-type**

Fija el aspecto del primer elemento de un tipo de selector si es el primer nodo hijo que aparece con su tipo de selector en su elemento padre.

* + 1. **:nth-child(#)**

Fija el aspecto de una ocurrencia específica del elemento nodo hijo especificado. Por ejemplo, el tercer elemento nodo hijo de una lista no ordenada sería li:nth-child(3). Además se pueden usar pequeñas expresiones como parámetro para por ejemplo seleccionar todos los elementos impares: li:nth-child(2n+1);, los pares: li:nth-child(2n);. Los elementos impares también pueden ser seleccionados con li:nth-child(odd); y los pares con li:nth-child(even);

El número que acompaña a la n especifica el salto, el segundo que va con la suma es dónde se empieza a contar.

Por ejemplo:

:nth-child(-n+5) empieza a contar desde el 5 hacia atrás, es decir, los 5 primeros.

:nth-child(n+6) son todos menos los 5 primeros.

* + 1. **:nth-of-type(#)**

Fija la apariencia de una ocurrencia específica del elemento con el tipo de selector especificado en un elemento padre. Por ejemplo la segunda lista no ordenada sería ul:nth-of-type(2). También permite los mismos parámetros que :nth-child(#).

* + 1. **:nth-last-of-type(#)**

Fija el aspecto de una ocurrencia específica del elemento con el tipo de selector especificado pero desde abajo. Por ejemplo la octava lista no ordenada empezando por abajo sería ul:nth-last-of-type(8). También le es aplicable la parametrización de :nth-child(#).

* + 1. **:nth-last-child(#)**

Aplica el parámetro empezando por los últimos hijos.

* + 1. **:last-child**

Fija el aspecto del último hijo del tipo de selector de un elemento padre.

* + 1. **:last-of-type**

Fija el aspecto de la última instancia de un selector particular en el elemento padre.

* 1. **Ejemplo:**

Las cuatro últimas filas de una tabla.

tr:nth-last-child(-n+4)

Muestra los elementos de la lista de menor a mayor tamaño de fuente y el primer y último elemento de la lista deben ser azules.

<style>

li:first-child { font-size: .9em; }

li:first-of-type { color: blue; }

li:nth-of-type(4) { font-size: 2em; }

li:nth-last-of-type(3) { font-size: 1.5em; }

li:last-of-type { color: blue; }

li:last-child { font-size: 2.5em; }

</style>

<ul>

<li>Primero</li>

<li>Segundo</li>

<hr> primero de su tipo

<li>Tercero</li> cuarto hijo de su padre, pero tercero de su tipo

<li>Cuarto</li>

<li>Quinto</li>

</ul>

* + 1. **:root**

1. La  [pseudo-clase](https://developer.mozilla.org/en/CSS/Pseudo-classes) de CSS :root selecciona el elemento raíz de un documento. Aplicado a HTML, :root representa el elemento [<html>](https://developer.mozilla.org/es/docs/Web/HTML/Elemento/html) y es idéntico al selector html, excepto que su nivel de [especificidad](https://developer.mozilla.org/en/CSS/Specificity) es mayor.

:root{

font-family:Arial, Helvetica, sans-serif;

background-color:#666;

color:#ccc;

}

**:only-child**

Representa a un elemento que no tiene hermanos.

**:only-of-type**

Representa a un elemento que no tiene hermanos del mismo tipo.

**:empty**

Para elementos que no tienen hijos ni texto.

div:empty{

height:100px;

width:100px;

background-color:#CFF;

}

<div></div>

<div>Este div contiene texto</div>

<http://www.w3.org/TR/2001/CR-css3-selectors-20011113/#structural-pseudos>

<http://programandowebs.es/2012/09/estilos-css-en-filas-pares-e-impares-de-una-tabla-html/>

**Aplicando estilo a idiomas**

**:lang()**

La pseudo clase :lang() nos permite especificar estilos que dependen del idioma especificado por la propiedad language:

<style>

p:lang(en) {

color: blue;

font-style: italic;

font-weight: bold;

}

</style>

<p>Esto es Castellano</p>

<p lang="en">This is English</p>

**No aplicar estilo a un elemento concreto**

**:not()**

Si no queremos dar estilo a un selector específico, podemos hacerlo gracias al pseudo elemento :not:

p:not(.notme) {

color: blue;

font-style: italic;

}

<p>Párrafo1</p>

<p class="notme">Este no tiene el estilo</p>

<p>Párrafo 2</p>

El párrafo del medio no debería de tener la fuente en color azul y cursiva.

**Aplicar estilo al objetivo de un enlace**

[**:target**](http://www.w3schools.com/cssref/sel_target.asp)

#news:target

Aplica el estilo al elemento con id="news" desde un enlace.

Ejemplo:

<http://www.w3schools.com/cssref/tryit.asp?filename=trycss3_target>

(Ver en el ejemplo anterior las animaciones y los display table)

<p><a href="#newcomment">Añade un comentario</a></p>

<div id="newcomment">

<form>

<p>Escribe tu comentario:<br />

<textarea></textarea></p>

</form>

</div>

**Formularios**

<http://www.w3.org/TR/2001/CR-css3-selectors-20011113/#structural-pseudos>

<http://www.w3.org/TR/2001/CR-css3-selectors-20011113/#selectors>

[**:focus**](http://www.w3schools.com/cssref/sel_focus.asp)

* 1. :focus, aplicado a los campos de formulario, nos permite introducir modificaciones en el momento en el que un campo de formulario tiene el foco.

input:focus { background-color: #eaeaea; }

La pseudoclase :focus{} tiene un correspondencia en Javascript con el evento onfocus en combinación con onblur. El primero actúa cuando el elemento obtiene el foco y el otro cuando lo pierde.

Los elementos capaces de recibir el foco son:

<a>, <area>, <label>, <input>, <select>, <textarea> y <button>. Sin embargo el elemento <label> es una simple etiqueta y realmente quién recibe el foco es el control que tiene asociado.

El hecho de incluir los cuadros de control de formulario dentro de un <form> es que automáticamente se le asigna un tabindex dentro del formulario según la posición que ocupe. Esto para los controles <input>, <select>, <textarea> y <button>, puesto que para otros elementos que no sean estrictamente controles como <a> esta tabulación automática no funciona. Por eso es conveniente incluir el atributo tabindex en los elementos del formulario además de encerrar todo el conjunto en un <form>.

En todo caso si queremos usar un evento CSS para modificar algo en el estilo hemos de usar un :focus. Este evento CSS mantiene las modificaciones durante todo el tiempo que el elemento mantenga el foco, pues los otros :hover o :active sólo lo hacen mientras estemos pasando el ratón encima del elemento o entre tanto se mantenga la tecla pulsada respectivamente.

input:focus {

border: 2px solid black;

background-color: lightgray;

}

input:hover {

border: 2px solid green;

}

**:focus-within**

* 1. <https://developer.mozilla.org/en-US/docs/Web/CSS/:focus-within>
  2. Aplica el estilo al contenedor del elemento que tenga el foco.

[**:invalid**](http://www.w3schools.com/cssref/sel_invalid.asp)

[**:valid**](http://www.w3schools.com/cssref/sel_valid.asp)

Con las opciones anteriores se podrán cambiar los estilos si la entrada es válida o no. La [pseudo-clase](https://developer.mozilla.org/es/docs/Web/CSS/Pseudo-classes) :valid CSS representa cualquier elemento de tipo [<input>](https://developer.mozilla.org/es/docs/Web/HTML/Elemento/input) o [<form>](https://developer.mozilla.org/es/docs/Web/HTML/Elemento/form) que cumpla con los ajustes de [validación](https://developer.mozilla.org/en/HTML/HTML5/Constraint_validation)  especificados para dicho elemento.

input:invalid {

border-color: red;

}

input:valid {

border-color: green;

}

[**:in-range**](http://www.w3schools.com/cssref/sel_in-range.asp)

Cuando un elemento tiene su atributo de valor dentro de las limitaciones de rango de este elemento.

[**:out-of-range**](http://www.w3schools.com/cssref/sel_out-of-range.asp)

Cuando un elemento tiene su atributo de valor fuera de las limitaciones específicas de rango para dicho elemento.

Estas 2 últimas pseudo-clases sólo se aplican a elementos que tienen limitaciones de rango

[**:indeterminate**](https://developer.mozilla.org/es/docs/Web/CSS/:indeterminate)

La [pseudo-clase](https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes) de [CSS](https://developer.mozilla.org/en-US/docs/Web/CSS) :indeterminate representa:

* cualquier elemento <input type="checkbox"> cuya propiedad indeterminate de DOM sea establecida con valor true desde JavaScript
* elementos <input type="radio"> donde todos los elementos de su mismo grupo estén desmarcados
* elementos [<progress>](https://developer.mozilla.org/es/docs/Web/HTML/Elemento/progress) en estado indeterminado

[**:checked**](http://www.w3schools.com/cssref/sel_checked.asp)

La [pseudo-clase](https://developer.mozilla.org/es/docs/Web/CSS/es/docs/Web/CSS/Pseudo-classes) de CSS :checked representa cualquier **radio** (<input type="radio">), **checkbox**(<input type="checkbox">) o **option** ([<option>](https://developer.mozilla.org/es/docs/HTML/Elemento/option) en un [<select>](https://developer.mozilla.org/es/docs/HTML/Elemento/select))que esté seleccionado.

[**:disabled**](http://www.w3schools.com/cssref/sel_disabled.asp)

[**:enabled**](http://www.w3schools.com/cssref/sel_enabled.asp)

Elementos que tengan o no el atributo disabled.

[**:required**](http://www.w3schools.com/cssref/sel_required.asp)

[**:optional**](http://www.w3schools.com/cssref/sel_optional.asp)

Elementos que tengan o no el atributo required.

[**:read-only**](http://www.w3schools.com/cssref/sel_read-only.asp)

[**:read-write**](http://www.w3schools.com/cssref/sel_read-write.asp)

Elementos que tengan o no el atributo readonly.

* 1. Probar el siguiente ejemplo en Firefox, Chrome o Safari:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Ejemplo 2 CSS-4</title>

<style>

input:invalid { border-color: red; }

input:valid { border-color: blue; }

input:required {background-color: yellow;}

input:optional {background-color: pink;}

input:-moz-read-only { /\* Firefox \*/

background-color: grey;

color: yellow;

}

input:read-only {

background-color: grey;

color: yellow;

}

input:nth-of-type(-n+3) {border-width: 5px;} /\*En formularios es type y no child por las label y br\*/

input:nth-child(-n+3) {border-style: dotted;}

:focus {background-color: white !important; } /\*Sin important habría que poner input:focus o una clase\*/

</style>

</head>

<body>

<form onsubmit="alert('Todo correcto.');return false;">

<input type="text" readonly="readonly" value="Solo lectura"><br>

<input type="text" pattern="[0-9]{3,5}" name="zipcode" value="" placeholder="3 a 5 números" /><br />

<input type="text" required pattern="[0-9]{8}[A-Za-z]" name="dni" placeholder="dni"><br />

<input type="text" required

pattern="((0[1-9]|5[0-2])|[1-4][0-9])[0-9]{3}" name="cp" placeholder="cp"><br>

<input type="text" pattern="^[a-zA-Z][a-zA-Z0-9-\_\.]{1,20}$" required name="nombre" placeholder="nombre"><br>

<button type="submit" >Enviar</button>

</form>

</body>

</html>

[HTML5Pattern](http://html5pattern.com/)

:default

[:default - CSS: Cascading Style Sheets | MDN](https://developer.mozilla.org/en-US/docs/Web/CSS/:default)

Para identificar un elemento del interfaz del usuario que es el elemento por defecto entre varios elementos similares.

input:default {

box-shadow: 0 0 2px 1px coral;

}

input:default + label {

color: coral;

}

<input type="radio" name="season" id="spring">

<label for="spring">Spring</label>

<input type="radio" name="season" id="summer" checked>

<label for="spring">Summer</label>

<input type="radio" name="season" id="fall">

<label for="spring">Fall</label>

<input type="radio" name="season" id="winter">

<label for="spring">Winter</label>

**:blank**

[:blank - CSS | MDN](https://developer.mozilla.org/es/docs/Web/CSS/:blank)

Selecciona elementos de entrada vacíos.

**Pseudoelementos**

* 1. Los[pseudo-elementos](https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-elements) son parecidos a las pseudo-clases, con alguna diferencia. Estos son claves — ahora precedidas por (::) — que se añaden al final del selector para elegir cierta parte de un elemento.
  2. Los pseudo-elementos nos permiten seleccionar partes de contenido dentro de una etiqueta. Por ejemplo, podemos seleccionar la primera línea empleando el pseudo-elemento ::first-line, o la primera letra, empleando el pseudo-elemento ::first-letter, o el primer hijo, mediante :first-child.

**selector::pseudo-elemento {  
    propiedad: valor;  
}**

[**::first-letter**](http://www.w3schools.com/cssref/sel_firstletter.asp)

[**::first-line**](http://www.w3schools.com/cssref/sel_firstline.asp)

p::first-letter {

font-weight: bold;

font-size: 300%;

background-color: red;

}

p::first-line {

font-weight: bold;

}

Otro uso de los pseudoelementos es generar contenido por medio del CSS. Podemos utilizar los pseudoelementos **::before** o **::after** para especificar el contenido que se debe insertar antes o después del elemento que seleccionemos. Entonces podemos especificar qué es lo que queremos insertar.

La regla siguiente para insertar una imagen decorativa después de cada uno de los enlaces de la página:

[**::after**](http://www.w3schools.com/cssref/sel_after.asp)

[**::before**](http://www.w3schools.com/cssref/sel_before.asp)

a::after{

content: " " url(diskon.ico);

}

.element::before {

content: "\01F525";

}

Ejemplo:

<!DOCTYPE html>

<html>

<head>

<style>

p::after

{

content:"- Remember this";

background-color:yellow;

color:red;

font-weight:bold;

}

</style>

</head>

<body>

<p>My name is Donald</p>

<p>I live in Ducksburg</p>

</body>

</html>

[**::selection**](http://www.w3schools.com/cssref/sel_selection.asp)

Remarca la parte del elemento que seleccione el usuario.

Se puede utilizar con las propiedades: color, background, cursor, and outline.

El pseudo-elemento **::selection**fue redactado como selector del nivel 3 de CSS pero fue quitado antes que alcanzara el estatus, sin embargo es soportado por la mayoría de los navegadores.

::-moz-selection {

color: red;

background: pink;

}

::selection {  
    color: red;   
    background: yellow;  
}

**Usar pseudo-clases con pseudo-elementos:**

article p:first-child::first-line {

font-size: 120%;

font-weight: bold;

}

**Otros pseudo selectores:**

<https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Selectores_CSS/Pseudo-clases_y_pseudo-elementos#Las_pseudoclases>

<https://developer.mozilla.org/es/docs/Learn/CSS/Building_blocks/Selectores_CSS/Pseudo-clases_y_pseudo-elementos#Pseudoelementos>

<https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes#specifications>

**:has()**

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

Ejemplos:

label:has(+ :required)

p:has(> em)

**:where()**

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

**:is()**

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

**Selectores avanzados de CSS**

**Selectores universales**

Selecciona todos los elementos para aplicarles estilos.

Ejemplos:

\* { border: 1px solid #000000;}

Todos los elementos de la página.

div \* {     background-color: yellow; }

Todos los elementos dentro de un div.

**Selectores de atributos**

[Selectores de atributo - CSS | MDN](https://developer.mozilla.org/es/docs/Web/CSS/Selectores_atributo)

Permiten seleccionar elementos en función de los atributos que contienen.

**[*attr*]**

Selecciona los elementos que tienen el atributo attr.

**[attr=value]**

Selecciona los elementos cuyo atributo *attr* tenga exactamente el valor *value*.

**[attr~=value]**

Selecciona los elementos cuyo atributo attr tenga por valor una lista de palabras separadas por espacios, una de las cuales sea value.

**[attr|=value]**

Selecciona los elementos cuyo atributo attr tenga exactamente el valor value o empiece por value seguido de un guión - (U+002D). Se puede usar para coincidencias de subcódigos en otros idiomas.

**[attr^=value]**

Selecciona los elementos cuyo atributo attr tenga un valor prefijado por value.

**[attr$=value]**

Selecciona los elementos cuyo atributo attr cuyo valor tiene el sufijo (seguido) de value.

**[attr\*=value]**

Selecciona los elementos cuyo atributo attr tenga un valor que contenga value.

**[attr operator value i]**

Agregar una i (o I) antes del corchete de cierre hace que el valor sea comparado sin distinguir entre mayúsculas y minúsculas (para caracteres dentro del rango ASCII).

**[attr operator value s]**

Agregar una s (o S) antes del corchete de cierre hace que el valor sea comparado distinguiendo entre mayúsculas y minúsculas (para caracteres dentro del rango ASCII).

Ejemplo:

img[alt]{

border: 1px solid #000000;

}

Todos los elementos img con un atributo alt.

**Ejemplos:**

img[src="imagen.gif"] {

border: 1px solid #000000;

}

Todos los elementos img con un src concreto.

input[type="button"] {

border: 6px solid blue;

}

Todos los elementos de tipo <input type=”button”>

input[type="submit"] {

width: 130px;

background-color: rgb(234, 103, 19);

color: rgb(255, 255, 255);

font-size: 24px;

font-weight: bold;

border: 0px none;

height: 35px;

}

Todos los <input type="submit">

input[type="submit"].naranja {

width: 130px;

background-color: rgb(234, 103, 19);

color: rgb(255, 255, 255);

font-size: 24px;

font-weight: bold;

border: 0px none;

height: 35px;

}

...

<input type="submit" value="Ver" class="naranja">

Todos los <input type="submit"> que tengan el atributo class="naranja".

**Selectores de hijos**

Seleccionar elementos concretos que son hijos de otros elementos concretos.

h3 > strong {

color: blue;

}

Color azul el texto de los elementos strong hijos de elementos h3, pero ninguno de los demás elementos strong.

h3 strong (descendiente y no tiene por qué ser hijo)

**Selectores de descendientes**

Son muy similares a los selectores de hijos, excepto que estos últimos seleccionan solo a los descendientes directos; los selectores de descendientes seleccionan los elementos pertinentes en cualquier punto de la jerarquía del elemento, y no sólo los descendientes directos.

<div>

<em>hello</em>

<p>In this paragraph I will say

<em>goodbye</em>.

</p>

</div>

En este fragmento, el elemento div es el padre de todos los demás. Tiene dos hijos, un em y un p. El elemento p tiene un único elemento hijo, que es otro em.

Podéis utilizar un selector de hijos para seleccionar sólo el elemento em que hay inmediatamente dentro de div.

div > em {

color: red;

background-color: yellow;

}

Si utilizamos un selector de descendientes de la siguiente manera:

div em {

color: red;

background-color: yellow;

}

entonces se seleccionarán los dos elementos em.

Ejemplo:

h3 strong {

color: blue;

}

h2 > strong {

color: blue;

}

<h3>hola<strong>uno</strong>hola<strong>uno</strong>

<p>hola2<strong>dos</strong></p>

</h3>

<h2>hola<strong>uno</strong>hola<strong>uno</strong>

<p>hola2<strong>dos</strong></p>

</h2>

**Selectores de hermanos adyacentes**

Permiten seleccionar un elemento concreto que aparece directamente después de otro elemento concreto en el mismo nivel de la jerarquía del elemento.

**Ejemplos:**

* 1. ***Aplica el estilo a los párrafos que estén inmediatamente después a un <h2>.***

h2 + p {

color: red;

background-color: yellow;

}

* 1. Aplica el estilo a los hermanos que se encuentren después del párrafo con class=”mitexto2” (pueden ser más de 1).

p.mitexto2 ~ p {

color: #990000;

}