**2. Estructura básica de una regla CSS**

Una hoja de estilos está compuesta por un conjunto de reglas que especifican el aspecto que tendrán los elementos de un documento HTML. Cada una de estas reglas contiene una lista de propiedades acompañadas de uno o más valores. Será el valor, aplicado a una propiedad concreta, el que define el aspecto final del elemento.

Cada regla está formada por dos partes diferenciadas:

**Selector { Declaración }**

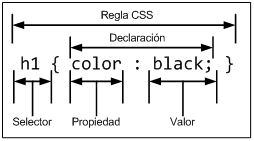
* **Selector**

Es un identificador que especifica sobre qué parte del documento se aplicará el estilo definido.

* **Declaración**

Establece una lista de definiciones (propiedades y valores) entre dos llaves que establecen el estilo a presentar. Esta lista debe contener al menos un elemento. Si existe más de un elemento deben separarse por (;).

CSS define una serie de términos que permiten describir cada una de las partes que componen los estilos CSS. El siguiente esquema muestra las partes que forman un estilo CSS muy básico:



Los diferentes términos se definen a continuación:

* Regla: cada uno de los estilos que componen una hoja de estilos CSS. Cada regla está compuesta de una parte de "selectores", un símbolo de "llave de apertura" ({), otra parte denominada "declaración" y por último, un símbolo de "llave de cierre" (}).
* Selector: indica el elemento o elementos HTML a los que se aplica la regla CSS.
* Declaración: especifica los estilos que se aplican a los elementos. Está compuesta por una o más propiedades CSS.
* Propiedad: característica que se modifica en el elemento seleccionado, como por ejemplo su tamaño de letra, su color de fondo, etc.
* Valor: establece el nuevo valor de la característica modificada en el elemento.

Un archivo CSS puede contener infinitas reglas CSS, cada regla puede contener infinitos selectores y cada declaración puede estar formada por un número infinito de pares propiedad/valor.

Un ejemplo sería:

**P** { **color**: **blue**; }

**SELECTOR** **PROPIEDAD** **VALOR**

**2.1. El selector**

2.1. El selector

2.1.1. Selectores de etiquetas html

2.1.2. Selectores de identificadores

2.1.3. Selectores de clases

[2.1.4. Selector universal](#_Toc185832899)

2.1.5. Selectores de pseudo-clases

2.1.6. Selectores de pseudo-elementos.

[2.1.7. Selectores por los atributos del elemento](#_Toc185832911)

[2.1.8. Selectores por contexto del elemento](#_Toc185832913)

[2.1.9. Agrupación y Combinación de Selectores](#_Toc185832914)

Está formado en principio por un identificador que sirve para determinar qué parte del documento utilizará el estilo definido en la Declaración. Este identificador sigue las siguientes reglas sintácticas:

* Comenzará por un carácter alfabético (A..Z o a..z).
* A continuación se incluirán más caracteres alfabéticos, guiones ( - ) y números (0-9).

Puede ir precedido de:

* **Carácter ' # ' (para formar los Selectores de Identificadores)**
* **Por un punto ’ . ' (para formar los Selectores de Clases).**

**COMPROBAR LOS EJERCICIOS UTILIZANDO HOJAS DE ESTILO EXTERNAS**

<link rel="StyleSheet" href="estilos.css" Type="text/css">

**2.1.1. Selectores de etiquetas HTML**

Definirán el tipo de elemento HTML que presentará el estilo definido en la regla (p, h2, li, etc.).

* Aplica formato a todos los elementos seleccionados sin distinción de clases, identificadores, etc. Es el más sencillo, pero con el inconveniente de que aplica a todos los elementos del tipo.
  + Ejemplo: p {color:red}

Un ejemplo sería:

|  |  |
| --- | --- |
| <html>  <head>  <**style** type="text/css">  **h2** { color:blue; font-size:30pt; font-family: times; }  **p** { color:red; font-family: "Comics Sans MS"; text-align: justify; font-style: italic}  <**/style**>  ...  </head>  <body>  <**h2**>Encabezado H2<**/h2**>    <**p**>  ESTO ES UNA PRUEBA <BR>  ESTO ES UNA PRUEBA <BR>  ESTO ES UNA PRUEBA<BR> <**/p**>  </body>  </html> |  |

**2.1.2. Selectores de Identificadores**

Las etiquetas HTML disponen de un atributo llamado “identificador”, cuya finalidad es la de asignar una identificación al elemento. De esta forma, CSS u otro lenguaje podrá seleccionarlo y distinguirlo de los demás.

Para definirlo se utiliza el atributo “**id**”, y con ello se asigna a esa etiqueta un nombre o expresión para distinguirlo. El uso de este atributo es opcional por lo que no tiene sentido definirlo si después no se prevé que CSS u otro lenguaje vaya a usarlo.

Un ejemplo sería:

<**p** **id**=”**despedida**”>

Mediante este tipo de selector dotamos de estilo a un único elemento de todo el documento.

El método para declarar un Selector de Identificador es el siguiente:

1. Primero debemos **definir una regla de estilo** que utilice un selector de id, es decir, un identificador **precedido** del **carácter almohadilla** ( **#** ).
2. Posteriormente **incluimos** el **atributo id** **a un único elemento de todo el documento e**n el cuál indicaremos el identificador (sin almohadilla #) que hemos usado en el selector.

* Si omitimos tipo elemento la regla afecta a todos los elementos incluidos dentro del elemento usado. Ej: #contenedor {} afectará a todos los elementos de contenedor. Si ponemos p#contenedor afectará sólo a los párrafos del contenedor

Ejemplo:

|  |  |
| --- | --- |
| <html>  <head>  <style type="text/css">  **#elemento1**{color:blue;  font-size:30pt; font-family: times;}  </style>  </head>  <body>  <**p** id="**elemento1**"> Párrafo que aplica el estilo. <**/p**>  <**p**> Párrafo que no aplica el estilo. <**/p**>  </body>  </html> |  |

EJEMPLO:

<html>

<head>

<style type= "text/css">

#cabecera { font-family:Times New Roman; font-size:30px; text-align:center; color:#0000ff;

background-color:#bbbbbb; }

</style>

</head>

<body>

<div id ="cabecera">

<h1>Título de la cabecera</h1>

</div>

</body>

</html>

**Utilizamos el carácter numeral (#) para indicar que se trata de un estilo de tipo Id. Luego, sólo una marca HTML dentro de una página puede definir un estilo de este tipo: <div id="cabecera">**

Hay que tener en cuenta que sólo una marca puede definir la propiedad id con el valor de cabecera.

**NOTA:** La diferencia fundamental en la definición de estilo por medio de Id con respecto a las clases (que veremos en el siguiente apartado) es, que solo podremos aplicar dicho estilo a una sola marca dentro de la página HTML.

**2.1.3. Selectores de Clases**

Las etiquetas HTML disponen de un atributo llamado “clase”, cuya finalidad es la de agrupar los elementos por clases o grupos. De esta forma, CSS u otro lenguaje podrá seleccionarlo y distinguirlo de los demás.

Para definirlo se utiliza el atributo “**class**”, y con ello se asigna un nombre o expresión para distinguirlo. El uso de este atributo es opcional por lo que no tiene sentido definirlo si después no se prevé que CSS u otro lenguaje vaya a usarlo.

Una etiqueta puede pertenecer a varias clases a la vez ya que éstas no son excluyentes.

Un ejemplo sería:

<**img src=**”**casa.jpg**” **class**=”**fotos**”>

<**img src=**”**chalet.jpg**” **class**=”**fotos casas**”>

Las clases son definiciones de estilo comunes a varios elementos, aunque éstos sean de distintos tipos. El método para utilizar las clases consiste en:

1. **Primero debemos declarar una regla de estilo en la cual el selector será el identificador que deseemos para la clase precedido de un punto ' . '.**
2. Después añadiremos el nombre de la clase (sin punto '.') que hemos decidido para la regla, en el atributo HTML class de todos aquellos elementos **que queramos que apliquen ese estilo.**

* Clases: Los objetos HTML tienen un atributo “class” que permite agrupar varios controles bajo la misma denominación. Pueden pertenecer a dos clases.

Ej.: <p class=”pNoticias pCabecera”>

Para acceder a ellos:

.nombreClase{}: afecta a todos los elementos de la clase

elemento.nombreClase{}: afecta únicamente al elemento especificado

EJEMPLO:

<html>

<head>

<style type="text/css">

**.GREENBOLD** {color: green; font-weight: bold;} </style>

</head>

<body>

<**h3** class="**GREENBOLD**">Este titulo es muy verde<**/h3**>

<**p** class="**GREENBOLD**">

Este párrafo usa la clase de estilo GREENBOLD. Se puede

utilizar el atributo class para especificar una clase de estilo

para cualquier elemento HTML.

<**/p**>

<**blockquote** class="**GREENBOLD**">

Este bloque usa la clase de estilo GREENBOLD. En consecuencia, es

verde y está en negrita. Puede ser útil para hacer que los bloques

resalten del resto de la página.

<**/blockquote**>

</body>

</html>

EJEMPLO:

**En muchas situaciones una regla de estilo puede ser igual para un conjunto de marcas HTML, en esos casos conviene plantear una regla de estilo con un nombre genérico que posteriormente se puede aplicar a varias marcas de HTML.**

<html> <head> <title>Problema</title>

<style type= “text/css”> body {

background-color:#eeeeee;

}

.resaltado{

color:#000000;

background-color:#ffff00;

font-style:italic;

}

</style>

</head> <body> <h1 class="resaltado">Titulo de nivel 1</h1> <p>Este parrafo muestra el resultado de aplicar la clase .resaltado en la última <span class="resaltado">palabra.</span> </p>

</body>

</html>

La sintaxis para definir una clase aplicable a cualquier marca HTML es:

.resaltado{

color:#000000;

background-color:#ffff00;

font-style:italic;

}

Es decir, la inicializamos con el carácter punto y seguidamente un nombre de clase. Dentro definimos las reglas de estilo.  
**El nombre de la clase no puede comenzar con un número.**  
Luego, para indicar que una marca sea afectada por esta regla:

<h1 class="resaltado">Titulo de nivel 1</h1>

Es decir, agregamos la propiedad class y le asignamos el nombre de la clase (sin el punto). Podemos inicializar tantas marcas HTML con esta regla como necesitemos:

<p>

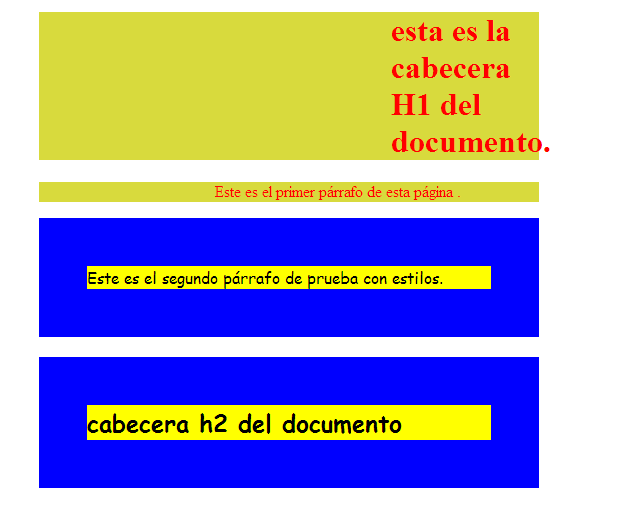
Este parrafo muestra el resultado de aplicar la clase .resaltado en la última <span class="resaltado">palabra.</span></p>

Definimos la **propiedad class a la marca span y le asignamos la misma clase aplicada anteriormente a la marca h1.**

EJERCICIO. Realiza el ejercicio que muestre la siguiente página, utiliza selectores de tipo class



EJERCICIO. Realiza el ejercicio que muestre la siguiente página, utiliza selectores de tipo class y de tipo id.



**2.1.4. Selector Universal**

Se indica por medio de un **carácter asterisco “\*”.** Su función es la de representar a cualquier elemento de la estructura de un documento HTML.

Ejemplo:

<style type="text/css">

\* { color: blue; } /\*Se aplica a cualquier etiqueta HTML\*/

...

<style>

EJEMPLO:

Si queremos inicializar las propiedades de todas las marcas HTML podemos utilizar el selector universal.

<html>

<head>

<title>Problema</title>

<style type="text/css">

\*{

color:green;

margin:0px;

padding:0px;

}

</style>

</head>

<body>

<h1>Título de nivel 1</h1>

<h2>Título de nivel 2</h2>

<p>Esto esta dentro de un párrafo</p>

</body>

</html>

**Esto significa que todas las marcas se imprimen de color verde con cero pixel de margin y padding, salvo que otra regla lo cambie, Imaginemos si definimos h1 { color:#ff0000} significa que tiene prioridad esta regla.**

**2.1.5. Selectores de pseudo-clases**

2.1.5. selectores de pseudo-clases

2.1.5.1. pseudo-clase a:link

2.1.5.2. pseudo-clase a:visited

2.1.5.3. pseudo-clase dinámica a:hover

2.1.5.4. pseudo-clase dinámica a:active

2.1.5.5. pseudo-clase dinámica a:focus

2.1.5.6. pseudo-clase dinámica :root

2.1.5.7. pseudo-clase dinámica :empty

2.1.5.8. pseudo-clase dinámica :target

2.1.5.9. pseudo-clase dinámica :first-child

2.1.5.10. pseudo-clase dinámica :last-child

Las pseudos-clases clasifican a las etiquetas según el estado de las mismas. Es decir, son agrupadas en función de los eventos que les hayan ocurrido (que el enlace haya sido visitado, que el cursor del ratón se encuentre encima o que deje de estar sobre él, etc.). Así pues, seleccionan los diferentes estados por los que puede pasar.

De las distintas etiquetas con pseudo-clases existentes, las más utilizadas son las referentes a los hipervínculos (**etiqueta <a>)**. Existen otros referentes a:

* El idioma (:lang)
* Al primer elemento después de otro (:first-child)

La sintaxis de las pseudo-clases es:

**etiqueta:pseudo-clase**

Las pseudo-clases pueden ser dinámicas en el sentido de que un elemento puede adquirirla o perderla a medida que el usuario interactúa con el documento. Estas pueden aplicarse a elementos distintos de hipervínculos.

Las pseudo\_clases son unas clases especiales, que se refieren a algunos estados especiales del elemento HTML, las que se utilizan fundamentalmente son las que se aplican a la marca <a>(ancla).

La sintaxis varía con respecto al concepto de clase visto anteriormente:

**a:pseudoclase {**

**propiedad: valor;**

**}**

Es decir separamos el nombre de la marca HTML con dos puntos.

Para la marca HTML **<a> tenemos 4 pseudoclases fundamentales**:

* link - Enlace sin introducir.
* visited - Enlace presionado
* hover - Enlace que tiene el puntero del mouse encima.
* active - Es la que tiene foco en ese momento.

**Es importante hacer notar que el orden en que definimos las pseudoclases es fundamental para su funcionamiento (debe respetarse el orden: link-visited-hover-active)**Este ejemplo es muy sencillo para ver el paso en los distintos estados que puede tener un enlace: p.html

<html>

<head>

<title>Problema</title>

<link rel="StyleSheet" href="estilos.css" type="text/css">

</head>

<body>

<a href="http://www.google.com">Google</a>

<a href="http://www.yahoo.com">Yahoo</a>

<a href="http://www.msn.com">Msn</a>

</body>

</html>

La hoja de estilo es: estilos.css

a:link{

background-color:#00ff00;

color:#ff0000;

}

a:visited{

background-color:#000000;

color:#ffffff;

}

a:hover{

background-color:#ff00ff;

color:#fffff;

}

a:active{

background-color:#ff0000;

color:#ffff00;

}

**COMPROBAR LOS COLORES SIGUIENTES**

Apenas ejecute la página los tres enlaces deben aparecer de color rojo con fondo verde:

a:link{

background-color:#00ff00;

color:#ff0000;

}

Si presionamos la tecla tab podremos ver que el enlace que tiene foco aparece de color amarillo con fondo rojo:

a:active{

background-color:#ff0000;

color:#ffff00;

}

Si pasamos la flecha del mouse sobre algún enlace veremos que aparece de color blanco con fondo lila:

a:hover{

background-color:#ff00ff;

color:#fffff;

}

Por último todos los enlaces que hayamos hecho click deberán aparecer de color blanco con fondo negro:

a:visited{

background-color:#000000;

color:#ffffff;

}

**2.1.5.1. Pseudo-clase a:link**

Para definir el estilo de los enlaces según se cargan en el documento. Es decir, todavía no se ha visitado.

Ejemplo:

|  |  |
| --- | --- |
| <html>  <head>  <style type="text/css">  **a:link**{ text-decoration: none; font-weight: bold; background-color: #dddddd; }  </style>  </head>  <body>  <p>Este <a href="enlace.htm">enlace</a> no aparece como suele ser habitual,  porque<br> se muestra sin subrayar, en negrita y con color de fondo.  </p>  </body>  </html> |  |

**2.1.5.2. Pseudo-clase a:visited**

Para definir el estilo de los enlaces ya visitados

Ejemplo:

|  |  |
| --- | --- |
| <html>  <head>  <style type="text/css">  **a:link**{ text-decoration: none; font-weight: bold; background-color: #dddddd; }  **a:visited** {background-color: yellow; color: #999999;}  </style>  </head>  <body>  <p><a href="enlaces.htm">Primer link</a></p>  <p><a href="enlaces2.htm">Segundo link</a></p>  </body>  </html> |  |

**2.1.5.3. Pseudo-clase dinámica a:hover**

Para definir el estilo de los enlaces sobre los que está el puntero del ratón.

Ejemplo:

|  |  |
| --- | --- |
| <html>  <head>  <style type="text/css">  **a:link** {font-family: serif; text-decoration: none}  **a:hover** {cursor: help; background-color: yellow; font-family: sans-serif}  </style>  </head>  <body>  <p>  <a href="ps\_din2.html">Link 2</a> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; <a href="ps\_din3.html">Link 3</a> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; <a href="link\_falso.html">Link de control</a>  </p>  </body>  </html> |  |

Ejemplo:

<html>

<head>

<style type="text/css">

a:hover {background-color:yellow;}

</style>

</head>

<body>

<a href="http://www.google.com">GOOGLE</a>

<a href="http://www.wikipedia.org">wikipedia.org</a>

<p><b>NotA:</b> :hover SELLECCIONA EL ESTILO Y EL ENLACE AL PASAR EL RATÓN</p>

</body>

</html**:hover se puede utilizar en todos los elementos no solo en los enlaces.**

Ejemplo de “:hover” para otro tipo de elemento:

|  |  |
| --- | --- |
| <html>  <head>  <style type="text/css">  **div** {border: 1px solid black}  **div:hover** {background-color: yellow}  </style>  </head>  <body>  <div><p>Situa el cursor sobre este cuadro  y el fondo pasará a ser amarillo.</p>  </div>  </body>  </html> |  |

**2.1.5.4. Pseudo-clase dinámica a:active**

Para definir el estilo de los enlaces sobre los que se está haciendo clic, durante el momento que transcurre entre presionar y soltar el botón del ratón sobre el elemento.

Ejemplo:

|  |  |
| --- | --- |
| <html>  <head>  <style type="text/css">  a:link {color: red}  a:visited {color: gray}  a:hover {color: blue}  a:active {color: fuchsia}  a:focus {  background-color: olive;  color: #fff;  border: 1px solid #ccc;  }  </style>  </head>  <body>  <p>  <a href="https://aulavirtual33.educa.madrid.org/ies.juandelacierva.madrid/">Enlace 1</a> <br>  <a href="https://aulavirtual33.educa.madrid.org/ies.juandelacierva.madrid/">Enlace 2</a> <br>  <a href="link\_falso.html">Enlace 3</a>  </p></body>  </html> |  |

Ejemplo de “:active” para otro tipo de elemento:

|  |  |
| --- | --- |
| <html>  <head>  <style type="text/css">  **div** {border: 1px solid black}  **div:hover** {background-color: yellow}  **div:active** {background-color: red}  </style>  </head>  <body>  <div><p>Situa el cursor sobre este cuadro  y el fondo pasará a ser amarillo.</p>  </div>  </body>  </html> |  |

**2.1.5.5. Pseudo-clase dinámica a:focus**

Para definir el estilo de los enlaces que están siendo seleccionados mediante el teclado.

Ejemplo:

<html>

<head>

<style type="text/css">

A:link {color: red}

A:visited {color: gray}

A:hover {color: blue}

A:active {color: fuchsia}

**A:focus** {

background: #cc0000;

color: #fff;

border: 1px solid #ccc;

}

</style>

</head>

<body>

<p>

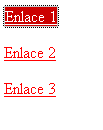
<a ref="ps\_din2.html">Enlace 1</a> <br>

<a href="ps\_din3.html">Enlace 2</a> <br>

<a href="link\_falso.html">Enlace 3</a>

</p></body>

</html>



Ejemplo de “:focus” para otro tipo de elemento:

<html>

<head>

<style type="text/css">

**input:focus**{

background-color:olive;

}

</style>

</head>

<body>

<p>Haga click dentro de la caja de texto cambiará de color:</p>

<form>

NOMBRE: <input type="text" name="nb" /><br />

APELLIDO: <input type="text" name="apel" />

<input type="submit" value="Submit" />

</form>

<p><b>Nota:</b>:focus estilo de los enlaces que estan siendo seleccionados .</p>

</body>

</html>

Ejemplo de “:focus” para otro tipo de elemento:

<html>

<head>

<style type="text/css">

**input:focus** {background-color: yellow}

</style>

</head>

<body>

<form>

<p>Escribe aquí: <input type="text"

size="30"></p>

</form>

</body>

</html>



**Nuevos selectores de CSS3**

**2.1.5.6. pseudo-clase dinámica :root.**

El selector **:root** de raíz coincide con el elemento raíz del documento. Si ponemos en lugar de :root body no existe ninguna diferencia; es decir afecta a todo el cuerpo del documento.

<!DOCTYPE html>

<html>

<head>

<style>

:root {

background: #ff0000; c

}

</style>

</head>

<body>

<h1>esto es cabecera</h1>

</body>

</html>

**2.1.5.7. pseudo-clase dinámica :empty**

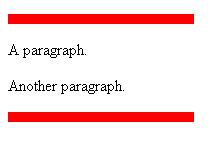
El selector **:empty** coincide con todos los elementos que no tienen contenido(incluidos los nodos de texto)

:empty {  
    css declarations;  
}

<!DOCTYPE html>

<html>

<head>

<style> 

:empty {

width: 100%;

height: 10px;

background: #ff0000;

}

</style>

</head>

<body>

<br/>

<p></p>

<p>A paragraph.</p>

<p>Another paragraph.</p>

<div></div>

</body>

</html>

**2.1.5.8. pseudo-clase dinámica :target**

URL con un # seguido de un nombre de ancla, enlaza con un cierto elemento dentro de un documento. El elemento está vinculado al elemento de destino.

Ej:target.html

<!DOCTYPE html>

<html>

<head>

<style>

:target {

border: 2px solid #D4D4D4;

background-color: #e5eecc;

}

</style>

</head>

<body>

<h1>This is a heading</h1>

<p><a href="#news1">Jump to New content 1</a></p>

<p><a href="#news2">Jump to New content 2</a></p>

<p>Click on the links above and the :target selector highlight the current active HTML anchor.</p>

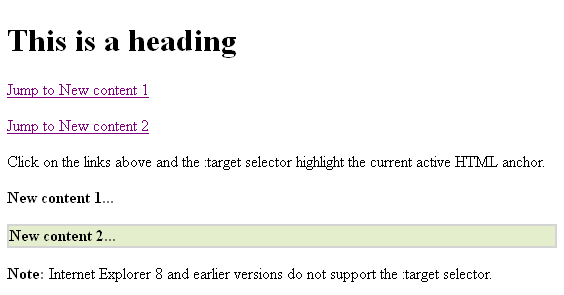
<p id="news1"><b>New content 1...</b></p>

<p id="news2"><b>New content 2...</b></p>

<p><b>Note:</b> Internet Explorer 8 and earlier versions do not support the :target selector.</p>

</body>

</html>



**2.1.5.9. pseudo-clase dinámica :first-child**

El selector: first-child se utiliza para seleccionar el selector especificado, sólo si es el primer hijo de su padre.

<!DOCTYPE html>

<html>

<head>

<style>

p:first-child {

background-color: yellow;

}

</style>

</head>

<body>

<p>este parrafo es el primer hijo del padre (body).</p>

<h1>bienvenido</h1>

<p>este parrafo no es el 1er hijo.</p>

<div>

<p>el 1er hijo del padre (div).</p>

<p>no es el 1er hijo.</p>

</div>

</body>

</html>

**2.1.5.10. pseudo-clase dinámica :last-child**

Selecciona a un determinado elemento siempre que sea el último hijo de un padre cualquiera.

<!DOCTYPE html>

<html>

<head>

<style>

p:last-child {

background: #ff0000;

}

</style>

</head>

<body>

<div>

<p>1er parrafo.</p>

<p>2º parrafo.</p>

<p>3er parrafo.</p>

<p>4º parrafo div1.</p>

</div>

<div>

<p>1er parrafo.</p>

<p>2º parrafo.</p>

<p>3er parrafo.</p>

<p>4º parrafo div2</p>

</div>

</body>

</html>

**2.1.6. Selectores de pseudo-elementos**

[2.1.6. Selectores de pseudo-elementos](#_Toc185832907)

[2.1.6.1. Pseudo-elemento::first-line](#_Toc185832908)

[2.1.6.2. Pseudo-elemento ::first-letter](#_Toc185832909)

[2.1.6.3. Pseudo-elementos::before y ::after](#_Toc185832910)

2.1.6.3 Pseudo-elemento::selection

Los pseudo-elementos existentes en CSS2 se mantienen invariables en CSS3 salvo un pequeño detalle: ahora se distinguen de la pseudo-clases usando **::** delante en lugar de : como hasta ahora. Se añade un pseudoelemento ::section que permite aplicar estilos diferenciados al texto que el usuario ha seleccionado mediante el ratón o el teclado

selector::pseudo-element {property:value;}

Los pseudo-elementos son **“partes” de elementos de una página** a los que HTML no otorga identidad propia y, por tanto, no pueden ser seleccionados por dicho lenguaje. Sin embargo, sí que **pueden ser identificados por las reglas de estilo** y ser formateados de manera distinta a la de los elementos a los que pertenecen.

Por ejemplo, en HTML no existe ninguna etiqueta para diferenciar **la primera letra o la primera línea** de cada párrafo (**subelementos de**l elemento **párrafo**). En cambio mediante los pseudo-elementos es posible acceder a ellas y dotarles de un estilo distinto al del párrafo.

La sintaxis de los pseudo-elementos es:

**etiqueta::pseudo-elemento**

**2.1.6.1. Pseudo-elemento ::first-line**

Para definir el estilo de la **primera línea visual del texto de un párrafo**. Esta primera línea va a referirse a la primera línea que produce el navegador al presentar el texto, de forma que acaba cuando se produce la ruptura de la línea para pasar a la siguiente. Debemos tener presente que no se trata de un elemento HTML real, es decir, no se define mediante una etiqueta, tan sólo es interpretada por el navegador.

Para especificar las propiedades de este pseudo-elemento el selector estará compuesto del identificador del elemento (etiqueta HTML) y a continuación los caracteres ":first-line".

Ejemplo:

|  |  |
| --- | --- |
| <html>  <head>  <style type="text/css">  body { font: 12pt Arial; }  p { color: red; }  **p::first-line** { color: blue; }  </style>  <body>  <p>Este es el primer párrafo del texto sobre el que hemos definido un estilo para la primera línea mediante un pseudo-elemento.</p>  <p>Este es el segundo párrafo <br>del texto sobre el que hemos  definido un estilo para la primera línea mediante un  pseudo-elemento.</p></body>  </html> |  |

Ejemplo:

<html>

<head>

<style type="text/css">

p::first-line{

background-color:yellow;

}

</style>

</head>

<body>

<h1>Pseudoelemento-first-line</h1>

<p>Define el estilo de la primera línea visual de un párrafo. Esta primera line se refiere a la primera línea que presenta el navegador, de forma que acaba cuando se pasa a la siguiente línea. No se trata de un elemento HTML, no se define mediante una etiqueta, tan solo es interpretada por el navegador.</p>

</body>

</html>

Sólo las siguientes propiedades se pueden aplicar al pseudo-elemento :first-line:

* la fuente, el color y el fondo
* word-spacing
* letter-spacing
* text-decoration
* vertical-align
* text-transform
* line-height
* text-shadow
* clear

**2.1.6.2. Pseudo-elemento ::first-letter**

Para definir el estilo de la primera letra de todo un bloque, por ejemplo, un párrafo de texto.

Ejemplo:

|  |  |
| --- | --- |
| <html>  <head>  <style type="text/css">  body { font-family: Arial; }  p { font-size: 12pt;  line-height: 12pt;  color: red; }  **p::first-letter** {  font-size: 250%;  font-weight: bold;  color: maroon;  background: salmon;float: left; }  </style>  <body>  <p>Este párrafo de texto tiene definido un estilo para la primera letra para que parezca una capitular caída. <br>El tamaño es del 250%, con color marrón y fondo color salmón.</p>  </body></html> |  |

Sólo las siguientes propiedades se pueden aplicar al pseudo-elemento :first-letter:

* la fuente, el color y el fondo
* los márgenes
* el relleno o padding
* los bordes
* text-decoration
* vertical-align (sólo si tiene definida la propiedad float con valor none)
* text-transform
* line-height
* text-shadow
* float
* clear

**2.1.6.3. Pseudo-elementos ::before y ::after**

Para definir el estilo de un contenido que se puede insertar antes o después de un elemento determinado. La **propiedad 'content'**, junto con estos pseudo-elementos, especifican lo que se inserta.

Ejemplo:

Podemos lograr que antes de cada elemento <H3> aparezca el texto "Tema:" sin necesidad de tener que escribirlo en cada título. También podemos hacer que cada párrafo termine con un pequeño ícono o poner "Fin" al pie de cada página usando las siguientes reglas.

En la última regla hemos especificado también **"display: block"** para que la palabra "Fin" comience en una nueva línea (como si fuese un nuevo párrafo).

|  |  |
| --- | --- |
| <html>  <head>  <title>Untitled</title>  <style type="text/css">  **H3::before** {content: "Tema:"}  **P::after** {content: url("cursorhelp.gif")}  **BODY::after** {content: "Fin"; display: block;}  </style>  </head>  <body>  <h3>1. Internet</h3>  <h3>2. Páginas y Sitios Web</h3>  <p>  Ahora vamos a escribir algo  </p>  </body>  </html> |  |

**EJEMPLO DE TODOS LOS PSEUDO-ELEMENTOS**

**<html>**

**<head>**

**<title>Uso de pseudo\_elementos</title>**

**<style type="text/css">**

**p.decorado::first-line{text-decoration:underline;color:red;}**

**p.decorado::first-letter{font-size:5em;border:solid 2px blue;}**

**p.decorado::after{content: "FIN DE TEXTO";}**

**p.yo::before{content:"periodista";}**

**p.el::before{content: "A.L";}**

**</style>**

**</head>**

**<body>**

**<h1>Ejemplo de un texto</h1>**

**<p class="decorado">**

**Para definir el estilo de un contenido que se puede insertar antes o después de un elemento determinado.**

**La propiedad 'content', junto con estos pseudo-elementos, especifican lo que se inserta. </p>**

**<h1>Entrevista</h1>**

**<p class ="yo"> ¿Que le pareció la pelicula?</p>**

**<p class="el"> en realidad no la vi</p>**

**<p class="yo"> pero si ayer la comentó en tu programa </p>**

**<p class="el"> me lo inventé todo</p>**

**</body>**

**</html>**

2**.1.6.3 Pseudo-elemento::selection**

El selector **::selection** aplica reglas a una porción de un documento que ha sido seleccionado mediante el mouse

<!DOCTYPE html>

<html>

<head>

<style>

::selection {

color: red;

background: yellow;

}

</style>

</head>

<body>

<h1>selecciona el texto en la página y veras como cambia el color</h1>

<p>esto es un párrafo.</p>

<div>elemento contenedor div.</div>

<a href="http://www.w3schools.com" target="\_blank">enlace W3Schools!</a>

<p><strong>Nota:</strong> The ::selection selector no es soportado por todos los navegadores</p>

</body>

</html>

**Nuevos selectores para uso de formularios**

**Input: In-range**

Selecciona el estilo dado a la etiqueta <input> sólo si el valor del elemento está dentro del rango especificado

input:in-range {   
    border: 2px solid yellow;  
}

<!DOCTYPE html>

<html>

<head><style>

input:in-range {

border: 2px solid yellow;

}</style></head>

<body>

<h3>A demonstration of the :in-range selector.</h3>

<input type="number" min="5" max="10" value="7">

<p>Try typing a number out of range (less than 5 or higher than 10), to see the styling disappear.</p>

<p><strong>Note</strong>: The :in-range selector is not supported in Internet Explorer.</p>

</body>

</html>

**:enable :disabled**

Establecer un color de fondo para todos los elementos <input> habilitados del type = "text":

input[type="text"]:enabled {  
    background: #ffff00;  
}

<!DOCTYPE html>

<html>

<head>

<style>

input[type=text]:enabled {

background: #ffff00;

}

input[type=text]:disabled {

background: #dddddd;

}

</style>

</head>

<body>

<form action="">

Nombre: <input type="text" value="Mickey"><br>

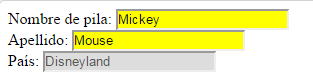
apellido: <input type="text" value="Mouse"><br>

Pais: <input type="text" disabled="disabled" value="Disneyland">

</form>

</body>

</html>



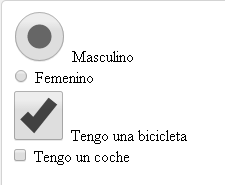
**:cheked**

El selector**: cheked** sólo para los botones de opción y casillas de verificación y elemento

<option> **.**

<!DOCTYPE html>

<html>

<head>

<style>

input:checked {

height: 50px;

width: 50px;

}

</style>

</head>

<body>

<form action="">

<input type="radio" checked="checked" value="male" name="gender"> masculino<br>

<input type="radio" value="female" name="gender"> femenino<br>

<input type="checkbox" checked="checked" value="Bike"> bicicleta<br>

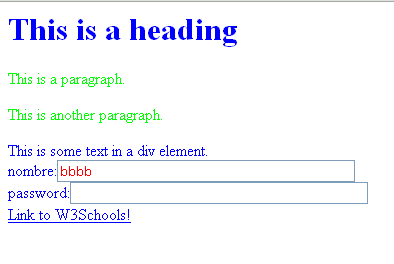
<input type="checkbox" value="Car"> coche

</body>

</html>

:not()

Sirve para aplicar estilos a aquellos elementos que no cumplen lo especificado entre paréntesis.

<head>

<style>

p {

color: #00ff00;

}

:not(p) {

color: #0000ff;

}

input:not([type="password"]){color:red;}

</style>

</head>

<body>

<h1>This is a heading</h1>

<p>This is a paragraph.</p>

<p>This is another paragraph.</p>

<div>This is some text in a div element.</div>

nombre:<input type="text" size=45><br/>

password:<input type="password" size=45>

<a href="http://www.w3schools.com" target="\_blank">Link to W3Schools!</a>

</body>

</html>

**2.1.7. Selectores por los atributos del elemento**

Para definir el estilo de aquellos elementos que tienen un determinado atributo o un valor de atributo dado. Aunque no son muy corrientes, pueden ser de utilidad en determinadas situaciones.

La forma en que se declaran es:

**Elemento[ atributo]** { estilo }

Se pueden combinar atributos al especificar la regla:

**Elemento[ atributo1] [ atributo2]…[atributoN]** { estilo }

Se pueden utilizar distintos patrones:

|  |  |
| --- | --- |
| **Patrón** | **Utilización** |
| **x[atributo]** | Para hacer referencia a cualquier elemento del tipo x que cuente con un determinado atributo (independientemente del valor que tenga el atributo) |
| **x[atributo="valor"]** | Para hacer referencia a cualquier elemento del tipo x que cuente con un determinado atributo cuyo valor sea el especificado |
| **x[atributo~="valor"]** | Algunos elementos pueden admitir varios valores separados por espacios para un atributo.  Para hacer referencia a cualquier elemento del tipo x que cuente con un determinado atributo, y alguno de cuyos valores sea el especificado.(alt+126 para sacar el carácter que hay antes del =) |
| **x[atributo|="valor"]** | Algunos elementos pueden admitir varios valores separados por guiones para un atributo.  Para hacer referencia a cualquier elemento del tipo x que cuente con un determinado atributo, y alguno de cuyos comience por el especificado. |

No existe soporte para este tipo de selectores en las versiones iguales o inferiores a la 6 de Internet Explorer.

Ejemplos:

*/\* Se muestran de color azul todos los enlaces que tengan*

*un atributo "class", independientemente de su valor \*/*

a[class] { color: blue; }

*/\* Se muestran de color azul todos los enlaces que tengan*

*un atributo "class" con el valor "externo" \*/*

a[class="externo"] { color: blue; }

*/\* Se muestran de color azul todos los enlaces que apunten*

*al sitio "http://www.ejemplo.com" \*/*

a[href="http://www.ejemplo.com"] { color: blue; }

*/\* Se muestran de color azul todos los enlaces que tengan*

*un atributo "class" en el que al menos uno de sus valores*

*sea "externo" \*/*

a[class~="externo"] { color: blue; }

*/\* Selecciona todos los elementos de la página cuyo atributo*

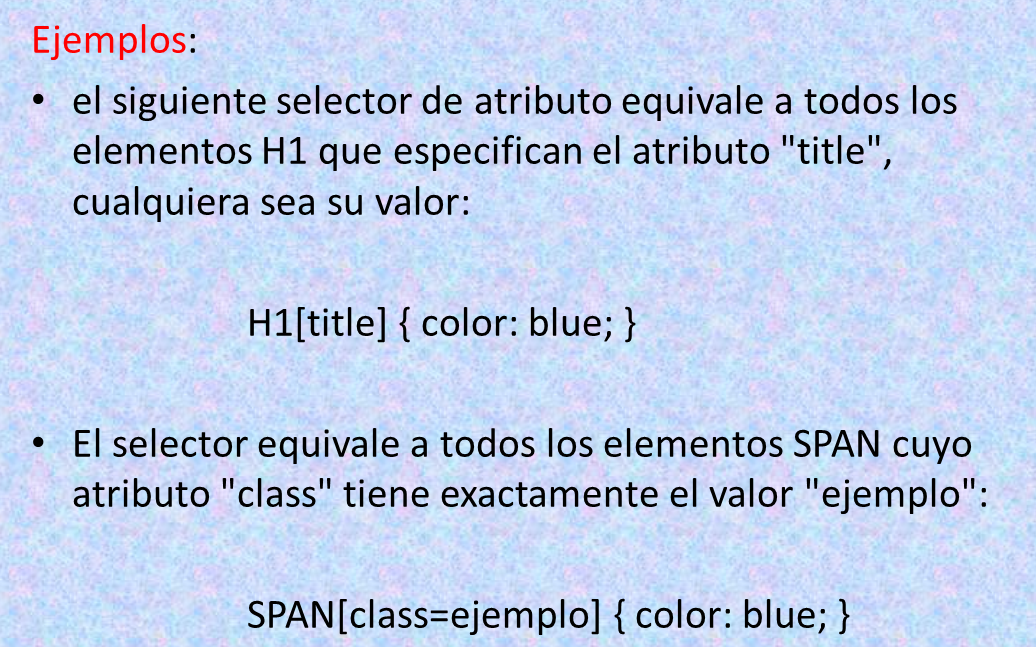
*"lang" sea igual a "en", es decir, todos los elementos en inglés \*/*

\*[lang=en] { ... }

*/\* Selecciona todos los elementos de la página cuyo atributo*

*"lang" empiece por "es", es decir, "es", "es-ES", "es-AR", etc. \*/*

\*[lang|="es"] { color : red }



Ejemplo:

<html>

<head>

<title>sintitulo</title>

<style type="text/css">

h2[title]{color:red}

h2[title="azul"]{color:blue}

</style>

</head>

<body>

<h2 title="algo"> en color rojo</h2>

<h2> En el color por defecto</h2>

<h2 title="azul"> en color azul</h2>

<p><br></p>

<p>En el primer caso se aplica el estilo a todos los elementos<b>h2</b><br>

con el atributo especificado(<b>title</b>).<br> En el segundo caso, solo a los que contengan el atributo especificado con el valor dado</p>

</body></html>

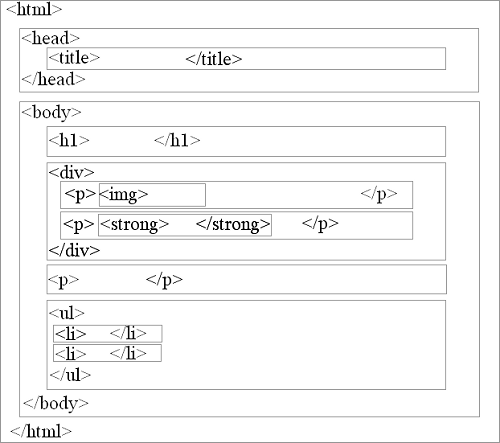
Ejemplo:

|  |  |
| --- | --- |
| <html>  <head>  <title>Untitled</title>  <style type="text/css">  **font[ face]{** font-weight:bold;}  **font[ face="Courier"]** { font-style:italic; }  **font[ face~="serif"]** { color:green; }  **font[ face|="serif"]** { color:yellow; }  </style>  </head>  <body>  <p><font face="Times New Roman, Times, serif">Este texto est&aacute; en letra times</font></p>  <p><font face="Helvetica, Arial, sans-serif">&Eacute;ste en arial</font></p>  <p><font face="Courier">Este est&aacute; escrito con courier</font></p>  <p><font face="serif">Esta  l&iacute;nea est&aacute;  tambi&eacute;n en una fuente serif  como la Times.</font></p>  Este texto se mostrar&aacute;, probablemente, con un tipo de letra serif que es el que suelen utilizar los navegadores por defecto, pero no lleva una etiqueta &lt;font&gt; por lo que no  deber&iacute;a modificar su apariencia  </body></html> |  |

**2.1.8. Selectores por contexto del elemento**

Para el uso correcto de este tipo de selectores debemos explicar previamente el concepto de **HERENCIA**.

Cada página HTML está compuesta por una serie de elementos (títulos, párrafos, listas, tablas, etc.) organizados en una estructura donde cada elemento está contenido por otro elemento, que a su vez puede estar contenido por otro.



En esta estructura existe un **elemento raíz** que es el que actúa de contenedor de todos los demás elementos. En HTML se puede considerar como elemento raíz al elemento **<BODY>** **o** al elemento **<HTML>**.

La importancia de este hecho es que **cada elemento hereda las propiedades del elemento que lo contiene** (llamado el **elemento padre**). Quiere decir que si especificamos la propiedad color: red para <BODY>, todos los elementos de la página heredarán esta característica y no será necesario especificar nuevamente la propiedad color en cada uno de ellos.

**Ejemplo**

<html>

<head>

<title>Untitled</title>

<style type="text/css">

**body** { color:red}

</style>

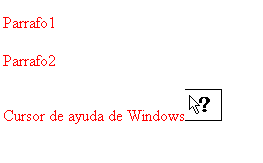
</head>

<body>

<p>Parrafo1</p>

<p>Parrafo2</p>

Cursor de ayuda de Windows<img src="cursorhelp.gif" alt="Cursor de ayuda de Windows"></body></html>



Ejemplo:

<html >

<head>

<title>Ejemplo de herencia de estilos</title>

<style type="text/css">

body { font-family: Arial; color: black; }

h1{ font-family: Verdana; }

p { color: red; }

</style>

</head>

<body>

<h1>Titular de la página</h1>

<p>Un párrafo de texto no muy largo.</p>

</body>

</html>

En el ejemplo anterior, se establece en primer lugar el color y tipo de letra del elemento <body>, por lo que todos los elementos de la página se mostrarían con ese mismo color y tipo de letra. No obstante, las otras reglas CSS modifican alguno de los estilos heredados,

Esta situación genera unas relaciones que pueden calificarse de familiares (de hecho, la nomenclatura para indicarlas utilizará los términos padre, hijo, ascendiente y descendiente entre otros).

Vamos a ver a continuación algunos ejemplos de lo que significa esta jerarquía en la organización de los elementos de una página html para lo que podremos guiarnos por el siguiente esquema de relaciones:

|  |  |
| --- | --- |
| **Relación** | **Descripción** |
| Un elemento es **padre** de otro si... | El segundo elemento se encuentra directamente entre la apertura y el cierre de las etiquetas del elemento padre.  Ej:  <p>Los <strong>rumiantes</strong> son <em>mamíferos</em></p>  El elemento <p> es padre de <strong> y de <em> |
| Un elemento es **hijo** de otro si... | La relación anterior es recíproca, por lo que los elementos <strong> y <em> del ejemplo son hijos del elemento <p> |
| Un elemento es **ascendiente** de otro si... | Si el segundo se encuentra entre las etiquetas de apertura y cierre del primero (aunque sea de forma indirecta).  Ej:  <div> <p>Los <strong>rumiantes</strong> son <em>mamíferos</em></p>  </div>  El elemento <div> es ascendiente  de <p> de <strong> y de <em> |
| Un elemento es **descendiente** de otro si... | Está incluido, **directa o indirectamente**, dentro de otro elemento.  En el ejemplo anterior <em> es descendiente de <p> y también de <div> |
| Un elemento es **adyacente** **o hermano** a otro si ... | Comparte el mismo padre y, además se encuentra inmediatamente consecutivo al anterior.  Ej:  <p>Los <strong>ungulados</strong> son <em>mamíferos</em> con <em>pezuñas</em></p>  En este ejemplo sólo el primer <em> sería adyacente a <strong> ya que el segundo, aunque tiene el mismo padre ya no es inmediatamente posterior. |

**El selector adyacente utiliza el signo + y su sintaxis es:**

**elemento1 + elemento2 { ... }**

La explicación del comportamiento de este selector no es sencilla, ya que selecciona todos los elementos de tipo elemento2 que cumplan las dos siguientes condiciones:

* elemento1 y elemento2 deben ser hermanos, por lo que su elemento padre debe ser el mismo.
* elemento2 debe aparecer inmediatamente después de elemento1 en el código HTML de la página.

En el siguiente ejemplo:

h1 + h2 { color: red }

<body>

<h1>Titulo1</h1>

<h2>Subtítulo</h2>

...

<h2>Otro subtítulo</h2>

...

</body>

Los estilos del selector h1 + h2 se aplican al primer elemento <h2> de la página, pero no al segundo <h2>, ya que:

* El elemento padre de <h1> es <body>, el mismo padre que el de los dos elementos <h2>. Así, los dos elementos <h2> cumplen la primera condición del selector adyacente.
* El primer elemento <h2> aparece en el código HTML justo después del elemento <h1>, por lo que este elemento <h2> también cumple la segunda condición del selector adyacente.
* Por el contrario, el segundo elemento <h2> no aparece justo después del elemento <h1>, por lo que no cumple la segunda condición del selector adyacente y por tanto no se le aplican los estilos de h1 + h2.

El siguiente ejemplo es muy útil para los textos que se muestran como libros:

p + p { text-indent: 1.5em; }

En muchos libros, suele ser habitual que la primera línea de todos los párrafos esté indentada, salvo la primera línea del primer párrafo. Con el selector p + p, se seleccionan todos los párrafos que estén dentro del mismo elemento padre que otros párrafos y que vayan justo después de otro párrafo. En otras palabras, el selector p + p selecciona todos los párrafos de un elemento salvo el primer párrafo.

Los patrones para indicar estas relaciones para elaborar los selectores serían los siguientes:

|  |  |
| --- | --- |
| **Patrón** | **Utilización** |
| **X Y** | Para apuntar hacia cualquier elemento del tipo **Y** que sea **descendiente** de un elemento del tipo **X** |
| **X > Y** | Para referirnos a cualquier elemento del tipo **Y** que sea **hijo** de un elemento del tipo **X** |
| **X + Y** | Para identificar cualquier elemento del tipo **Y** que sea **adyacente** a un elemento del tipo **X** |

Teniendo en cuenta estas relaciones podemos realizar una selección muy específica de elementos del documento para aplicarles un determinado estilo.

Ejemplo:

Podríamos querer que los fragmentos que estuvieran enfatizados mediante la etiqueta <em> aparecieran en color verde. Pero podríamos ser incluso más específicos indicando que sólo queremos que aparezcan en color verde los fragmentos enfatizados que formen parte de una lista. Esto lo haríamos colocando primero el selector li y, a continuación separado por un espacio, el selector em para luego especificar las propiedades

|  |  |
| --- | --- |
| <html>  <head>  <title>Untitled</title>  <style type="text/css">  **li em** {color: green}  </style></head>  <body>  <p>La palabra <em>"atención"</em> está enfatizada pero se presenta en su color normal porque:</p> <ol> <li>Está enfatizada pero... </li> <li><em>No pertenece a una lista</em></li> </ol> <p>La segunda condición también está enfatizada, pero en este caso pertenece a una lista y se ve en verde </p>  </body></html> |  |

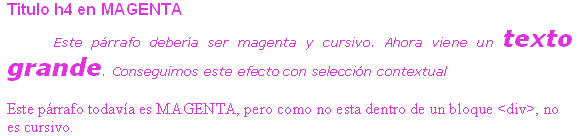
**Se pueden utilizar los criterios de selección contextual para buscar etiquetas, clases, ids o combinaciones de estos**.

Ejemplo:

Se crea la **clase MAGENTA** que lo colorea todo de magenta. Todos los párrafos MAGENTA que estén dentro de un <div> estarán en cursiva. Además los textos dentro de <b> anidados dentro de párrafos dentro de un <div> en MAGENTA usaran una fuente grande.

|  |
| --- |
| <html>  <head>  <title>Untitled</title>  <style type="text/css">  **.MAGENTA** {color: magenta;}  **div** **p.MAGENTA** {font-style: italic;}  **div p.MAGENTA b** {font-size: large;}  </style></head>  <body>  <div **class**=”MAGENTA”>  <h4> Titulo h4 en MAGENTA</h4>  <p **class**=”MAGENTA”>Este párrafo debería ser magenta y cursivo. Ahora viene un <b>texto grande</b>. Conseguimos este efecto con selección contextual</p>  </div>  <p **class**="MAGENTA">Este párrafo todavía es MAGENTA, pero como no esta dentro de un bloque &lt;div>, no es cursivo.</p>  </body></html> |

Resultado del ejemplo:



**EJEMPLO**

<html>

<head>

<style type="text/css">

div a {color:#00FF00; border-style:solid;} /\* Afecta a todas las <a> \*/

div > a {color:#0000FF;} /\* Afecta sólo a las <a> que cuelgan directamente de un div\*/

p+p {color:red;} /\* Afecta sólo a los dós últimos párrafos\*/

</style>

</head>

<body>

<p> Parrafo externo </p>

<div id "contenedor">

<p>Parrafo dentro de un div</p>

<div id="tituloPag">

<p> Párrafo más interno de otro div 1 </p>

<a href="pepe"> Enlace hijo de un div</a>

<p> Párrafo más interno de otro div 2 con un

<a href="pepe"> enlace nieto de un div</a>

</p>

<p> Párrafo más interno de otro div 3 </p>

<p> Párrafo más interno de otro div 4 </p>

</div>

</div>

</body>

</html>

**2.1.9. Agrupación y Combinación de Selectores**

[2.1.9. Agrupación y Combinación de Selectores](#_Toc185832914)

[2.1.9.1. Agrupación de selectores](#_Toc185832915)

[2.1.9.2. Combinación de selectores](#_Toc185832916)

[2.1.9.2.1. Combinación de selectores de elementos, con selectores de clases y con selectores de identificado](#_Toc185832917)

En ocasiones, con el fin de realizar una selección más optima, “ahorrar” código o que éste sea más fácil de entender para otros programadores (o para nosotros mismos), es recomendable realizar agrupaciones, combinaciones.

**2.1.9.1. Agrupación de selectores**

Cuando tenemos varios selectores que comparten las mismas declaraciones pueden agruparse en una lista separada por comas (**,**). Se trata de una fórmula abreviada para aplicar las mismas propiedades a distintos selectores en un solo paso.

Por ejemplo, si tenemos:

H1 {font-family: Arial, Sans-serif}

P {font-family: Arial, Sans-serif}

TABLE {font-family: Arial, Sans-serif}

#volanta {font-family: Arial, Sans-serif}

Podemos simplificarlo de este modo:

H1, P, TABLE, #volanta {font-family: Arial, Sans-serif}

Supongamos que queremos la misma fuente y color para las marcas h1, h2 y h3. Podemos implementarlo de la siguiente manera:

<html>

<head>

<title>Problema</title>

<style type="text/css">

h1,h2,h3 {

font-family:verdana;

color:#0000ff;

}

</style>

</head>

<body>

<h1>Título de nivel 1</h1>

<h2>Título de nivel 2</h2>

<h3>Título de nivel 3</h3>

</body>

</html>

Es decir, separamos por coma todas las marcas a las que se aplicará la misma regla de estilo:

h1,h2,h3 {

font-family:verdana;

color:#0000ff;

}

**2.1.9.2. Combinación de selectores**

Todos los selectores vistos anteriormente se pueden combinar para obtener así una selección de elementos más óptima. De esta forma se pueden seleccionar elementos que cumplan más de un criterio a la vez.

Para realizar una combinación de selectores, se deben seguir una serie de indicaciones y atenerse a las siguientes reglas de precedencia (que veremos en profundidad más adelante):

* Definir el contexto del elemento deseado
* Definir el nombre del elemento que se quiere seleccionar o utilizar el selector universal (\*)
* Especificar la clase o id del elemento deseado
* Definir los atributos que puedan estar presentes

***2.1.9.2.1. Combinación de selectores de elementos, con selectores de clases y con selectores de identificador***

Los tipos de selectores: de elemento, de clase y de identificador pueden combinarse obteniendo selectores más complejos:

* **elemento.clase**

Con este selector, que combina un Selector de Elemento delante de un Selector de Clase, distinguimos a unos determinados elementos (aquellos que definen el atributo class con el nombre de la clase) de entre todos de su mismo tipo.

* **elemento#id\_elemento**

Con este otro selector, que combina un Selector de Elemento con un Selector de Id se diferencia un elemento concreto (aquel que defina el atributo id con el nombre del identificador) del resto de elementos de su mismo tipo.

Ejemplo:

|  |  |
| --- | --- |
| <html>  <head>  <style type="text/css">  **p.clase1** { font-family:serif;color:red; }  **h2.clase2** { font-family:Courier;color:blue; font-weight:italic; }  **h1#ident1** { font-family:Arial;color:green; font-weight:bold;}  </style>  </head>  <body>  <h1>Cabecera de tipo 1 sin estilo </h1>  <h1 id="ident1">Cabecera de tipo 1 con el estilo único para este elemento</h1>  <h2 class="clase2">Cabecera de tipo 2 con el estilo de clase2</h2>  <p>Párrafo sin ningún estilo</p>  <p class="clase1">Párrafo con el estilo de la clase1</p>  </body></html> |  |

Otras combinaciones posibles de selectores para restringir el alcance de las reglas CSS serían:

|  |  |
| --- | --- |
| **.aviso .especial** { ... } | Esta regla solamente selecciona aquellos elementos con un class=”especial” que se encuentren dentro de cualquier elemento con un class=”aviso”. |
| **div.aviso span.especial** { ... } | Esta regla solamente selecciona aquellos elementos de tipo <span> con un atributo class=”especial” que estén dentro de cualquier elemento de tipo <div> que tenga un atributo class=”aviso”. |
| **ul#menuPrincipal li.destacado a#inicio** { ... } | Esta regla solamente selecciona a los enlaces con un id=”inicio” que se encuentra dentro de un elemento de tipo <li> con un class=”destacado”, que forma parte de una lista <ul> con un atributo id=” menuPrincipal”. |

* **Selectores descendientes**. Es una forma de seleccionar elementos que se encuentran dentro de otros elementos para aplicarles un estilo personalizado.

**Ejemplo**: **h1 em {color:red;}**

El estilo se aplicará a todas las etiquetas **em** dentro de un elemento **h1** sin importar el nivel de profundidad. Como vemos en el siguiente fragmento:

**<h1>**

**<a id=”INICIO”> titular de la<em> página</em></a>**

**</h1>**

* Se puede combinar este tipo de selectores con el selector universal para realizar restricciones.

**Ejemplo:**

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

**p \* a{ color:red; }**

**<p><a id=”INICIO”> enlace</a></p>**

**<p><strong><a href=”#”> enlace</a></strong></p>**

* **Si usamos la primera regla CSS se colorean en verde las dos líneas.**
* **Si usamos la segunda regla, solo colorea la segunda línea.**
* Si quisiéramos restringir el uso del selector de clase para que sólo sea válido en combinación de una determinada etiqueta, podemos combinarlo con el selector de tipo o etiqueta.

**Ojo, no debemos confundir las tres formas diferentes de combinación de selectores.**

**1er caso: p, .verde{color: green;}. La regla se aplica a cualquier etiqueta p o a cualquier etiqueta que tenga un atributo class con color verde.**

**2º caso: p .verde{color: green;}. Se aplica a cualquier etiqueta con un atributo class con valor verde que se encuentre dentro de un elemento de tipo p.**

**3er caso: p.verde{color: green;}. Solo se aplica en las etiquetas de tipo p que tengan atributo de tipo class con valor igual a verde**

EJEMPLO: CombinacionSelectores.html

!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xml:lang="es" xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html;charset=UTF-8" />

<title>Página CombinacionSelectores</title>

<style type="text/css">

p, .verde { color:red; }

p .verde{color:blue;}

p.verde{color:green;}

</style>

</head>

<body>

<h1>Combinación de selectores</h1>

<p class= "verde"> Si meliora dies, ut vina, poemata reddit, scire velim, c qui deperiit minor uno mense vel anno, inter quos referendus erit?</p>

<h1 class="verde">Titular del párrafo 2</h1>

<p>Si meliora dies, ut vina, poemata reddit, scire velim, chartis ptor abhiveteresque referri <span class="verde"> debet </span> Quid, qui deperiit minor uno mense vel anno, inter quos referendus erit?

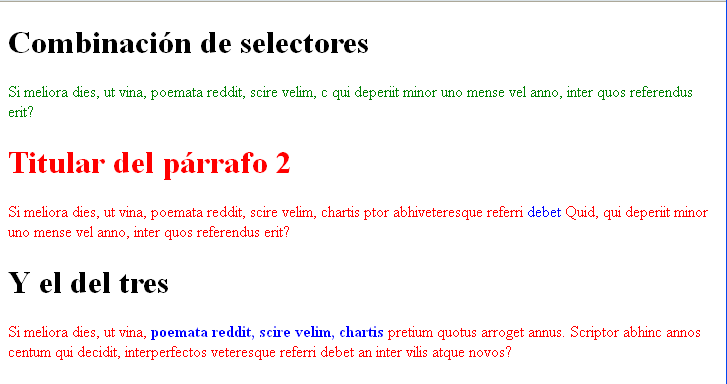
</p>

<h1>Y el del tres</h1>

<p>Si meliora dies, ut vina, <strong class="verde">poemata reddit, scire velim, chartis </strong>pretium quotus arroget annus. Scriptor abhinc annos centum qui decidit, interperfectos veteresque referri debet an inter vilis </p>

</body>

</html>



* Podemos aplicar también el formato de más de un selector de clase al mismo elemento html.

**Ejemplos:**

**.codigo {font-family: monospace; color:rojo;}**

**.versalita {Font-variant: small-caps;}**

**Suponiendo que tenemos el siguiente código html:**

**<p class=” codigo versalitas”> párrafo en rojo con fuentes monoespaciadas usando versalitas</p>**

Podemos usar también los selectores de clase múltiple. La regla css se expresa.

**.codigo.versalita { font-family: monospace; color:rojo; font-variant: small- caps;}**