

Tema 3: Introducción a las Hojas de Estilo (CSS, Cascading Style Sheets)

**Lenguajes de Marcas y SGI
Administración de Sistemas Informáticos en Red
IES Los Manantiales**

1.- Qué son las Hojas de Estilo en Cascada (CSS, Cascading Style Sheets)

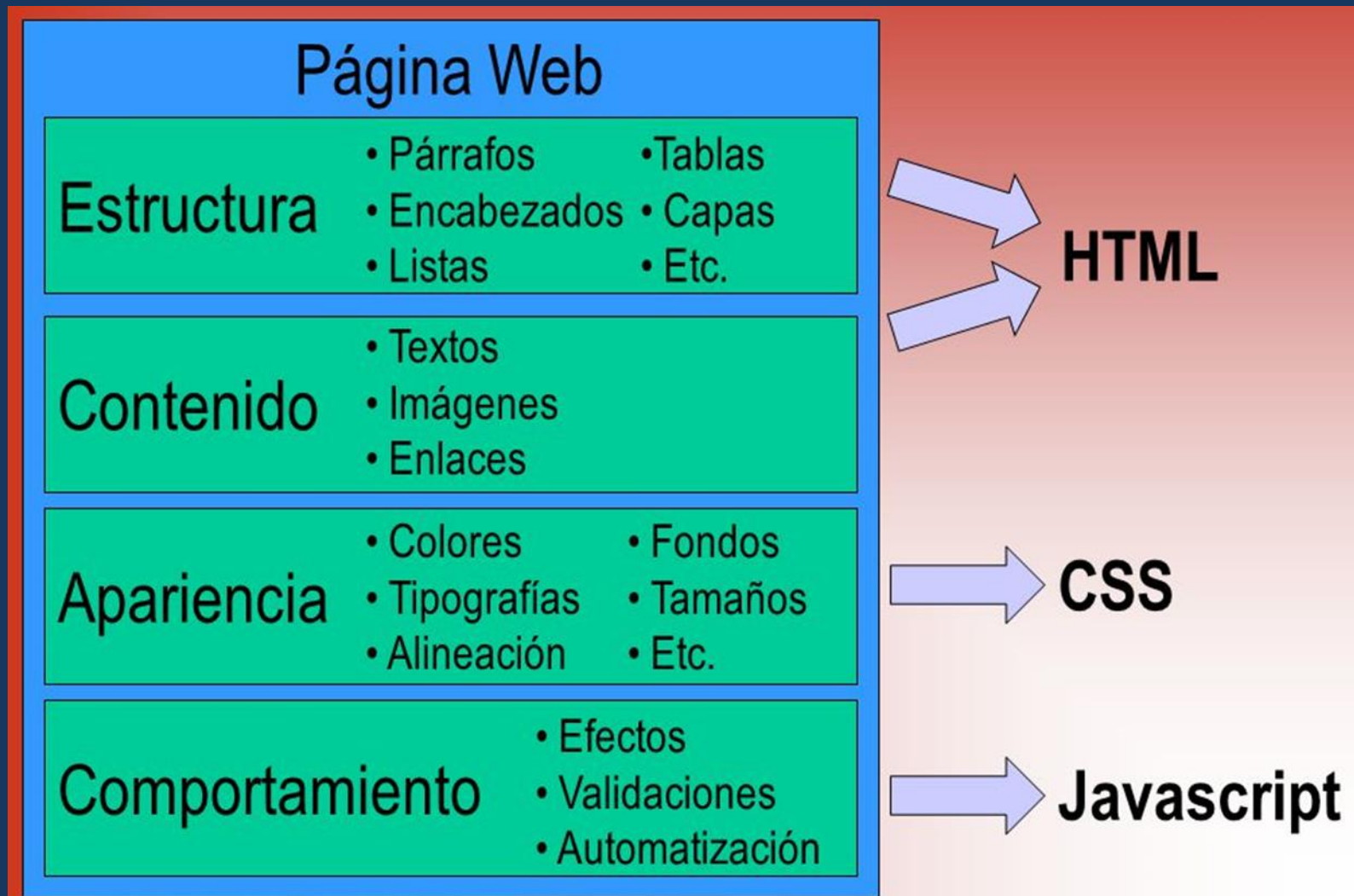
HTML es un lenguaje de marcas diseñado para mostrar contenido en la web, pero tiene muchas limitaciones en lo que se refiere al aspecto.

El objetivo principal de CSS es manejar el aspecto y formato de los documentos, liberando de esta forma a HTML de las tareas de presentación.

La idea principal consiste en separar la definición de los contenidos (HTML) de la definición del aspecto (CSS).

Con los estilos podemos configurar el tipo de letra, alineación, colores, bordes, enlaces, imágenes de fondo, etc.

1.- Qué son las Hojas de Estilo en Cascada



2.- Por qué aparecieron las CSS

Imaginemos que nuestro sitio web escrito en HTML tiene unas 50 páginas web. Supongamos también que cada página puede tener unos 50 elementos susceptibles de ser personalizados: encabezados, párrafos, listas, tablas, imágenes, etc. con distintos tamaños, colores, bordes, tipos de letra, etc.

Si queremos cambiar el aspecto de nuestra web, deberemos cambiar, por ejemplo, los valores de 2 atributos de cada etiqueta. Si tenemos 50 elementos en cada página y tenemos 50 páginas, el cálculo es sencillo:

$$2 \text{ atributos} \times 50 \text{ etiquetas} \times 50 \text{ páginas} = 5.000 \text{ cambios}$$

5.000 cambios es mucho trabajo.

3.- Cómo nos ayudan la CSS

Seguimos con los 5.000 cambios.

Si en mi documento HTML limito el contenido de las etiquetas y retiro los atributos de estilo, mi documento HTML sólo contiene la información a mostrar, pero no tiene nada de cómo se muestra.

CSS me permite definir el estilo que tendrá cada elemento de la web una sola vez, de forma que nos ahorramos tener que configurar todas las etiquetas que forman el sitio. Cada tipo de etiqueta tendrá su estilo definido previamente. Todo el contenido de un mismo tipo mantendrá el mismo aspecto.

Así, para cambiar el aspecto de una página, sólo tendré que cambiar una vez el estilo definido al comienzo del documento.

4.- Cómo se aplicaba el estilo antes de CSS

```
<html>
<head>
  <title>Ejemplo de estilos sin CSS</title>
</head>
<body>
  <h1><font color="red" face="Arial" size="5">Titular de la página</font></h1>
  <p><font color="gray" face="Verdana" size="2">Un párrafo de texto no muy largo.</font></p>
  <h1><font color="red" face="Arial" size="5">Otro Titular de página</font></h1>
  <p><font color="gray" face="Verdana" size="2">Siguiendo párrafo.</font></p>
</body>
</html>
```

5.- Métodos para definir estilos

- **CSS en línea**

Se aplica el estilo directamente en la etiqueta que se quiere modificar, dentro del body del documento HTML.

- **CSS interno**

Dentro del documento HTML, en la zona head, se especifican los estilos que darán formato a los distintos elementos HTML.

- **CSS externo**

Se utiliza un archivo externo con extensión .css que contendrá todos los estilos. Los documento .html deben hacer referencia al archivo .css del que dependen.

6.- CSS en línea

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Ejemplo de estilos con CSS en línea</title>
</head>
<body>
  <h1 style="color: red; font-family: Arial; font-size: large;">
    Ejemplo de CSS en línea</h1>
  <p style="color: gray; font-family: Verdana; font-size: medium;">
    Este párrafo contiene estilos CSS en línea, definidos dentro de
    las propias etiquetas.</p>
</body>
</html>
```

El método menos recomendable. Es una forma similar a la definición de estilos sin CSS, por lo que tiene los mismos problemas que ya conocemos.

Se usa para especificar estilos muy particulares de algunos elementos.

7.- CSS interno

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Ejemplo de estilos con CSS</title>
  <style>
    h1 { color: red; font-family: Arial; font-size: large; }
    p { color: gray; font-family: Verdana; font-size: medium; }
  </style>
</head>
<body>
  <h1>Ejemplo de CSS interno</h1>
  <p>Este párrafo contiene estilos CSS internos, definidos dentro de
    la cabecera, mediante la etiqueta style.</p>
</body>
</html>
```

Indicado para especificar estilos propios de la página. Modifican a los estilos del archivo externo.

8.- CSS externo

Definir estilos al comienzo del documento permite no tener que modificar los atributos de cada etiqueta de la página.

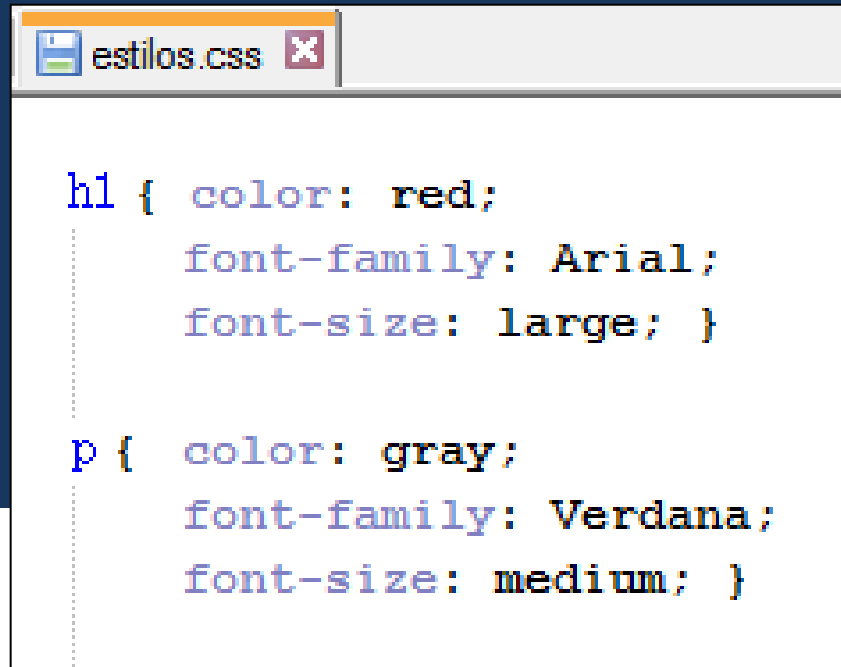
Sin embargo, si tenemos muchas páginas en nuestro sitio, CSS nos permite definir los estilos en un fichero externo e independiente. De esta forma, cada página de nuestro sitio hará referencia al fichero de estilos para “adoptarlos” del mismo.

Si quiero realizar cambios en la web, bastará con cambiar los estilos definidos en el fichero externo para que todo el sitio obtenga las modificaciones oportunas.

El archivo externo que tiene los estilos tendrá extensión .css

8.- CSS externo

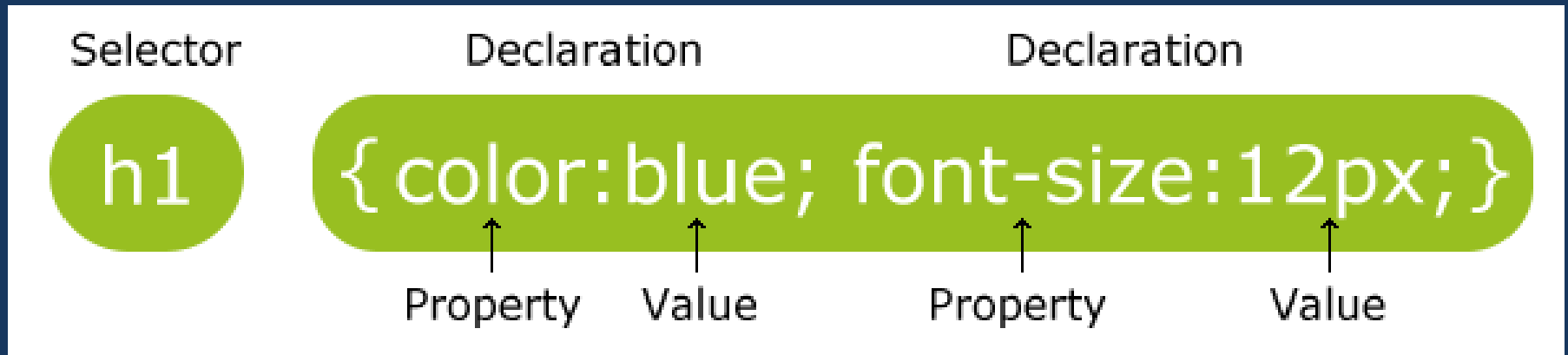
```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Ejemplo de estilos externos con CSS</title>
  <link rel="stylesheet" href="estilos.css">
</head>
<body>
  <h1>Ejemplo de CSS externo</h1>
  <p>Este párrafo contiene estilos CSS importados de un archivo
  externo llamado estilos.css mediante la etiqueta link.</p>
</body>
</html>
```



```
h1 { color: red;
      font-family: Arial;
      font-size: large; }

p { color: gray;
     font-family: Verdana;
     font-size: medium; }
```

9.- Sintaxis básica de CSS



Este es el formato básico de una regla CSS.

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.

10.- Etiquetas <div> y

Las etiquetas **div** se usan para agrupar elementos HTML en forma de bloque, pudiendo tratar a este bloque como un grupo.

Las etiquetas **span** se usan para delimitar un trozo de texto al que se le podrán aplicar determinados estilos.

```
p {color:red;}
```

```
span {color:green;}
```

```
<p>Datos estadísticos de la compañía <span>Los Manantiales</span>  
referentes al último trimestre.</p>
```

11.- Prioridad en estilos CSS

Es posible que para algunos elementos tengamos definidas reglas CSS de diferentes formas. Así, una etiqueta HTML puede verse afectada por estilos CSS que se han definidos en un archivo externo, en una declaración interna en la cabecera de la página e incluso de estilos en línea como atributos de la misma etiqueta.

A esto se le llama colisión de estilos. Por ello, debemos tener en cuenta que la prioridad que se aplica es la siguiente:

1º CSS en línea > 2º CSS interno > 3º CSS externo

Esto significa que, de haber varios estilos definidos para un elemento, se aplicará el más específico.

12.- Comentarios

```
/* Este es un comentario en CSS */
```

```
/* Este es un  
comentario CSS de varias  
lineas */
```

13.- Selectores

Los selectores nos permiten seleccionar los elementos sobre los que tendrán efecto los estilos definidos.

Dependiendo de dónde queramos aplicar el estilo a diseñar, tenemos que especificar qué elemento de la página va a recibir el estilo.

- Selector universal
- Selector de etiqueta
- Selector de clase
- Selector de id
- Selector descendiente
- Selector adyacente
- Selector de atributos

14.- Selector Universal

Este selector se aplica a todas las etiquetas de la página que puedan recibir el estilo especificado.

Es un selector muy general que se utiliza cuando queremos aplicar un estilo a todas las etiquetas.

Está representado por un asterisco (*)

```
* { color:red;  
    font-family: Verdana;  
    font-size: 14px; }
```

15.- Selector de Etiqueta

Son los estilos que hemos diseñado hasta ahora. Se especifica la etiqueta Html que recibe el estilo.

Se pueden agrupar etiquetas con los mismos estilos.

```
h1 { font-family: Verdana;  
      font-size: 24px;  
      color: red; }
```

```
h2 { font-family: Arial;  
      font-size: 18px;  
      color: marine; }
```

```
p { font-family: Verdana;  
      font-size: 14px;  
      color: gray; }
```

```
h1, h2, p { font-family : Arial;}
```

16.- Selector de Clase

Podemos definir clases con una serie de estilos definidos. Todos los elementos Html que pertenezcan a esa clase recibirán los estilos definidos para esa clase. En la definición se pone . y el nombre de la clase.

```
.especial { color : red;  
            font-family : Arial;}
```

```
<p class="especial">Lorem ipsum dolor sit amet</p>  
  
<h1 class="especial">Título en la Web</h1>
```

17.- Selector de ID

Se utilizan cuando el estilo se repite una sola vez en la página. Tienen más prioridad que los selectores de clase. Siempre empiezan por #.

```
#nombre {    color : red;  
           font-family : Arial;}
```

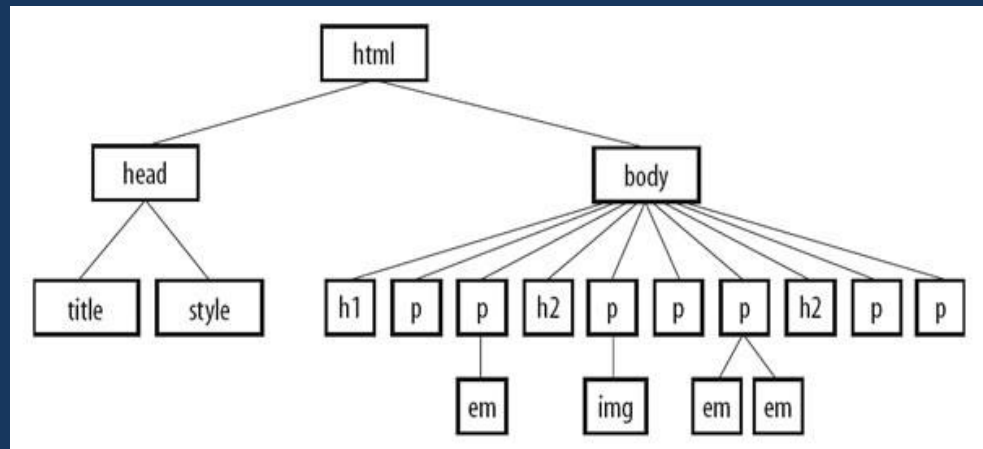
```
<p id="nombre">Lorem ipsum dolor sit amet</p>
```

id -> muy usados en programación JavaScript

18.- Selector Descendente

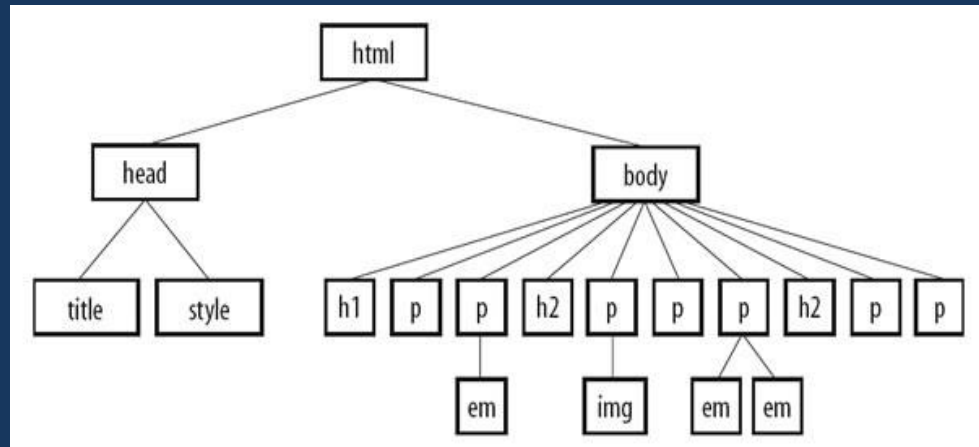
Árbol Genealógico HTML.

- Antepasado
- Descendiente
- Padre
- Hijo
- Hermano



Selecciona los elementos que se encuentran dentro de otros elementos. Un elemento es descendiente de otro si está contenido entre la etiqueta de apertura y cierre del otro.

18.- Selector Descendente



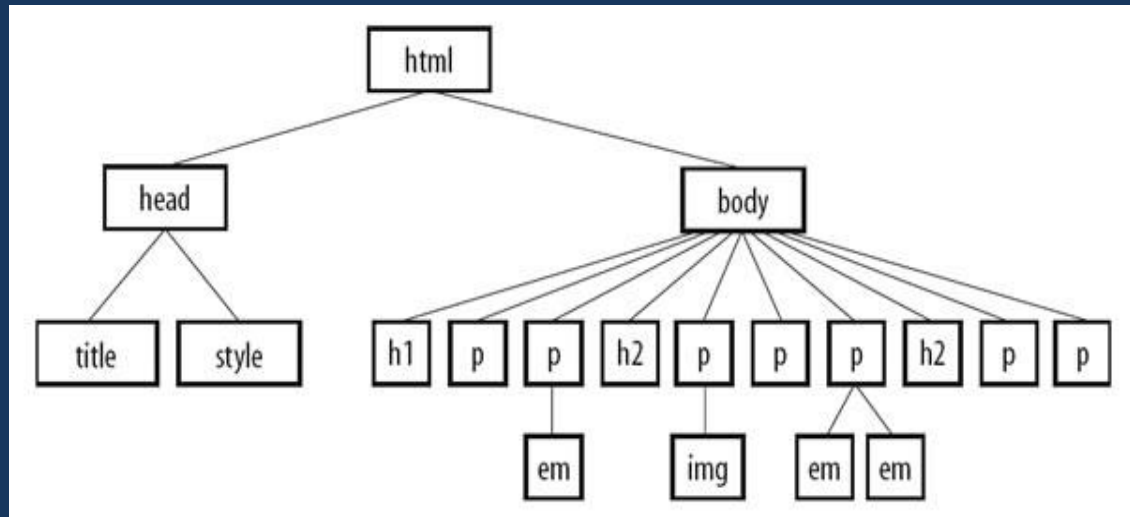
```
p em {color : red;}  -- selector descendente
```

Código HTML

```
<p><em>Este texto tiene énfasis</em></p>
```

19.- Selector Adyacente

E + M {declaración}



Selecciona a todos los elementos de tipo M, tales que E y M tienen el mismo padre y E precede inmediatamente a M en el código.

```
h1 + h2 {color : red; font-size: 50px;}
```

20.- Selector de Atributos

Sirve para especificar estilos a elementos que tengan atributos específicos.

```
h1 [class] {font-family:Arial;}  
  
h2 [class="titulos"] {color:blue;}
```

En el primer ejemplo, los elementos h1 pertenecientes a cualquier clase se verán afectados por el estilo.

21.- Resumen Selectores

Selectores	
*	Todos los elementos
div	<div>
div *	Elementos dentro de <div>
div span	 dentro de <div>
div, span	<div> y
div > span	 con <div> como padre
div + span	 precedido de <div> </div>
.foto	Elementos de la clase "foto"
div.class	<div> de la clase "foto"
#casa	Elementos con id "casa"
div#itemid	<div> con id "casa"
a[class]	<a> con el atributo class
a[class='x']	<a> cuando su clase es "x"
a[class~='x']	<a> cuando su clase es una lista de valores y uno de ellos es 'x'
a[lang='en']	<a> cuando lang comienza por "en"

22.- Estilos del navegador

Cada navegador parte con unos estilos previos predeterminados.

Internet Explorer:

<http://www.iecss.com/>

Firefox:

<http://mxr.mozilla.org/mozilla-central/source/layout/style/html.css>

Chrome y Safari:

<http://trac.webkit.org/browser/trunk/Source/WebCore/css/html.css>

Opera:

<http://www.iecss.com/opera-10.51.css>

Cómo podemos resetear los estilos:

```
<link rel="stylesheet" href="reset.css">  
<link rel="stylesheet" href="estilos.css">
```

23.- Resetear estilos.

Para cancelar los estilos del navegador podemos usar un Reset CSS que unifica la apariencia de la páginas en cualquier navegador.

```
/* reset browser styles */
html, body, h1, h2, h3, h4, h5, h6, p, ol, ul, li,
{
    padding: 0;
    margin: 0;
    font-size: 100%;
    font-weight: normal;
}
ol {
    margin-left: 1.4em;
    list-style: decimal;
}
ul {
    margin-left: 1.4em;
    list-style: square;
}
img {
    border: 0;
}
/* end reset browser styles */

#main {
    float: left;
    width: 580px;
}
#sidebar {
    margin-left: 600px;
    padding: 15px;
    background: #E9F5FE;
}
```

