

Guía práctica de

# **Hojas de Estilo (CSS)**

Hojas de Estilo (CSS)

# Algunas Ventajas de usar Hojas de Estilo (CSS)

## Separación del contenido y la presentación.

Las Hojas de Estilo se encuentran (o deberían encontrarse) en archivos aparte de los archivos HTML. De esta forma, programador y diseñador puede realizar sus trabajos de forma independiente sin descuidar el resultado final.

## Flexibilidad

La flexibilidad que nos da poder cambiar el aspecto completo de nuestros documentos simplemente.

## Unificación del diseño

Siempre es más fácil modificar el diseño de todo un sitio a través de un archivo CSS, que tener que editar uno a uno todos los archivos HTML y cada uno de sus elementos. Nos da agilidad y minimiza el trabajo.

## Optimización de tiempos de carga

Al separar el contenido y la apariencia obtenemos archivos menos pesados lo cual reduce los tiempos de carga de un sitio web en el navegador. Por otra parte, la mayor parte de navegadores conservan en su caché los estilos del sitio de manera que sólo se cargan una vez al navegar por él. Esto también reduce el volumen de tráfico del servidor.

## Compatibilidad y Estandarización

CSS en todas sus versiones se hace compatible para las versiones anteriores y lo asegura una entidad tan importante como la W3C. Al ser un estándar se obtiene la ventaja de la compatibilidad entre los distintos navegadores. Al contrario de lo que ocurre con las soluciones propietarias de Microsoft que utiliza patrones y etiquetas propios y únicos.

## Importar Estilos

La forma más sencilla de añadir estilos a nuestra página web es utilizando las etiquetas `<style></style>` en un documento HTML cualquiera.

```
<html>
<head><title>Ejemplo 01</title>
      <style type="text/css">
        body{background-color:red;}
      </style>
</head>
<body></body></html>
```

Aunque por razones de claridad y cohesión es recomendable separar los estilos de la página HTML en un archivo diferente, con extensión .css que será importado desde la página principal o plantilla.

```
<html>
<head><title>Ejemplo 02</title>
<link rel="stylesheet" type="text/css" href="../ruta/hoja.css"/> </head>
<body></body>
</html>
```

# Selectores CSS

Los selectores nos permiten determinar a qué elemento o grupo de elementos aplicamos un determinado estilo.

A continuación un listado de selectores básicos de CSS:

Selector	Descripción
*	Selecciona a todos los objetos HTML dentro del ámbito actual del objeto.
.[nombre_clase]	Crea una clase css con el nombre dado.
[Selectores html]	Selecciona todos los elementos HTML del tipo indicado dentro del ámbito actual del objeto.
#[Selectores por ID]	Selecciona los elementos HTML que sean contenidos por otro cuyo id sea el indicado.

En el siguiente ejemplo se pueden observar distintos usos de los selectores anteriormente citados.

## Archivo HTML

```
<html>
<head><title>Test 01</title></head>
<link rel="stylesheet" type="text/css" href="../ruta/hoja.css"/>
<body>
  <h3>Test 01</h3>
  <p id="par1">Lorem ipsum</p>
  <span class="texto_etiqueta">Lorem ipsum</span>
  <input type="button" value="botón" />
</body>
</html>
```

## Hoja de Estilos (CSS)

```
h3 { font-size: 5em; font-weight: bold; }

#par1 { color: blue; border: 1px solid black; }

input { font-size: 2em; color: green; }

.texto_etiqueta { background-color: blue; color:white; font-weight: bold; }
```

# Agrupación de elementos CSS

En muchas ocasiones, los selectores comparten propiedades. Las hojas de estilos permiten agrupar estos selectores para poder asignar estas propiedades comunes una sola vez.

A continuación veamos un ejemplo de esto:

```
/* Sin utilizar las agrupaciones. (desaconsejado) */  
.header_text { font-family: arial; color: blue; font-size: 18px; }  
.body_text { font-family: arial; color: blue; font-size: 12px; }  
.footer_text { font-family: arial; color: blue; font-size: 10px; }
```

```
/* Utilizando agrupaciones (forma correcta) */  
.header_text, .body_text, .footer_text { font-family: arial; color: blue; }  
.header_text { font-size: 18px; }  
.body_text { font-size: 12px; }  
.footer_text { font-size: 10px; }
```

Para indicar que estamos definiendo una agrupación separamos los elementos con comas (,).

# Agrupación y Dependencia

La agrupación de selectores y la dependencia del contexto que ofrecen las hojas de estilo pueden ser combinadas para realizar cualquier selección imaginable.

Gracias a esta flexibilidad, trabajar con hojas de estilo puede ser tan fácil o complicado como nosotros queramos hacerlo. En ocasiones un uso excesivo de agrupaciones dependientes hace difícil la lectura de los estilos y no es aconsejado.

```
u b, b .body_text, .footer_text { color: red } ;
```

```
/* Se verán afectadas las etiquetas 'b' que se encuentren entre etiquetas 'u' (underline). */
```

```
/* Se verán afectadas las clases 'body_text' que se encuentren entre etiquetas 'b' (bold). */
```

```
/* Se verán afectadas las clases 'footer_text' */
```

La jerarquía de la dependencia se establece de izquierda a derecha. Se separan los elementos a través de un espacio.

# Selectores Dependientes del Contexto

Los selectores, dependen siempre del contexto en el que son incluidos. Esto es muy útil a la hora de delimitar el ámbito de ciertos estilos CSS dependiendo del uso y el lugar donde se encuentren

En el siguiente ejemplo vemos un estilo que sólo tiene efecto dependiendo del contexto en que es utilizado:

```
/*Se verán afectadas todas las etiquetas 'b' (bold). */  
b { color: #aeaeae; }
```

```
/*Se verán afectadas SOLAMENTE las etiquetas 'b' que se encuentren entre etiquetas 'u' (underline). */  
u b { color: #aeaeae; }
```

En un lugar de <u><b>La Mancha...</b></u>

La jerarquía de la dependencia se establece de izquierda a derecha. Se separan los elementos a través de un espacio.



# Selector Adyacente

Un selector muy útil es el selector adyacente. Este tipo de selector nos permite seleccionar el elemento adyacente (más cercano) al elemento actual.

El selector adyacente se especifica a través del operador “+” de suma. Observemos el siguiente ejemplo:

```
/* Selector adyacente de H1 */  
h1 + h2 { ... }
```

En este caso se selecciona únicamente aquel h2 que sea adyacente al h1, es decir que esté justamente detrás en la estructura de una página HTML. El Resto de h2 de la página no se verán afectados.

```
<!DOCTYPE html>
<html>
<head>
<style>
div+p
{
background-color:yellow;
}
</style>
</head>
<body>

<h1>Welcome to My Homepage</h1>

<div>
<h2>My name is Donald</h2>
<p>I live in Duckburg.</p>
</div>

<p>My best friend is Mickey.</p>

<p>I will not be styled.</p>

</body>
</html>
```

# Welcome to My Homepage

**My name is Donald**

I live in Duckburg.

My best friend is Mickey.

I will not be styled.

# Selector General de Hermanos (css3)

Con CSS 3 se introduce un nuevo selector, llamado Selector General de Hermanos. Este selector, se diferencia del Selector Adyacente en que selecciona todos los hermanos de un mismo padre.

Al contrario que el adyacente que únicamente seleccionaba elemento que se encontraba justo detrás, el selector de hermanos seleccionará todos y cada uno de los elementos hijos dentro de la estructura HTML (o XML) del documento.

```
/* Selector General de Hermanos */  
h1 ~ h2 { ... }
```

En este caso se seleccionan todos los elementos h2 que estén al nivel de hijos en la estructura XHTML de la página.

```
<!DOCTYPE html>
<html>
<head>
<style>
p~ul
{
background:#ff0000;
}
</style>
</head>
<body>

<div>A div element.</div>
<ul>  <li>Coffee</li>  <li>Tea</li>  <li>Milk</li> </ul>

<p>The first paragraph.</p>
<ul>  <li>Coffee</li>  <li>Tea</li>  <li>Milk</li> </ul>

<h2>Another list</h2>
<ul>  <li>Coffee</li>  <li>Tea</li>  <li>Milk</li> </ul>

</body>
</html>
```

A div element.

- Coffee
- Tea
- Milk

The first paragraph.

- Coffee
- Tea
- Milk

## Another list

- Coffee
- Tea
- Milk

## Selector de Atributos (css3)

Con la versión 3 de las hojas de estilo se introducen algunas mejoras en cuanto a los selectores se refiere. A parte de todo lo anterior, además proporcionan el acceso a elementos del documento accediendo a sus propiedades, como por ejemplo con el uso de los Selectores de Atributos

- Elemento[atributo<sup>^</sup>="valor"]  
**Selecciona todos los elementos que disponen de este atributo y cuyo valor comience por el especificado.**
- Elemento[atributo<sup>\$</sup>="valor"]  
**Selecciona todos los elementos que disponen de este atributo y cuyo valor termine por el especificado.**
- Elemento[atributo<sup>\*</sup>="valor"]  
**Selecciona todos los elementos que disponen de este atributo y cuyo valor contenga el valor especificado.**

```
/* Selecciona todos los enlaces que apuntan a una dirección de correo electrónico */  
a[href^="mailto:"] { ... }
```

```
/* Selecciona todos los enlaces que apuntan a una página HTML */  
a[href$=".html"] { ... }
```

```
/* Selecciona todos los títulos h1 cuyo atributo title contenga la palabra "capítulo" */  
h1[title*="capítulo"] { ... }
```

# Pseudo Elementos (css3)

Los pseudo elementos son otra herramienta más para determinar mejor la manera de filtrar la selección de elementos en la estructura de un documento bien organizado.

Los pseudo elementos ya existen desde versiones anteriores de CSS pero en la última versión se introduce un cambio en la forma de definirlos. En lugar del carácter ':' se usa '::'. Con esta única diferencia por lo demás su uso se mantiene igual.

- **Elemento::first-line** {...}

Selecciona la primera línea del texto de un elemento.

- **Elemento::first-letter** {...}

Selecciona la primera letra del texto de un elemento.

- **Elemento::before** {...}

Selecciona la parte anterior al contenido de un elemento para insertar un nuevo contenido generado.

- **Elemento::after** {...}

Selecciona la parte posterior al contenido de un elemento para insertar un nuevo contenido generado.



# Pseudo-Clases en CSS2

En CSS2 existen varias pseudo-clases que pueden ser usadas para acceder a ciertos estados de los elementos de la página. Estos se puede dividir en dos grupos, los que afectan únicamente a los enlaces (etiqueta 'a') y los que pueden afectar a cualquier otro elemento.

- Las pseudo-clases **:link** y **:visited**

```
a:link { color: red; }           /* Se aplica a todos los enlaces que todavía no hayan sido visitados */  
a:visited { color: green; }     /* Se aplica a todos los enlaces que ya hayan sido visitados al menos una vez */
```

- Las pseudo-clases **:hover**, **:active** y **:focus**

```
Elemento: hover { color: red; }      /* Se activa al pasar el ratón por encima de cualquier enlace */  
  
Elemento: active { color: green; }    /* Se activa cuando el usuario activa un elemento de la página. Por ejemplo al hacer  
click sobre el enlace, hasta que suelta de nuevo el botón del ratón */  
  
Elemento: focus { color: blue; }    /* Se aplica a todos los elementos que tiene el foco de la aplicación. Suele aplicarse  
a elementos <input> */
```

# Pseudo Clases (css3)

Además de las pseudo-clases existentes en la versión CSS2, la nueva versión de hojas de estilo introduce nuevas pseudo-clases muy interesantes. Las pseudo clases se indican con el carácter ‘:’.

Por desgracia, algunos navegadores aún no soportan todas o algunas de estas pseudo-clases formuladas hace ya algunos años por la W3C.

- **Elemento:nth-child(número) {...}**

Selecciona el elemento que cumpla la condición de ser el enésimo hijo de su padre. De esta forma podemos por ejemplo, seleccionar el tercer párrafo de un elemento.

- **Elemento:nth-last-child(número) {...}**

Es como el anterior, pero el número que se indica empieza a contar después del último hijo.

- **Elemento:first-child {...}** y **Elemento:last-child {...}**

Selecciona el primer y último elemento hijo del elemento principal.

- **Elemento:nth-of-type(número) {...}**

Selecciona el elemento que cumpla la condición de ser el enésimo hermano del elemento principal.

- **Elemento:nth-last-of-type(número) {...}**

Es como el anterior, pero el número que se indica empieza a contar después del último hermano.