

# Tema 03 - Hojas de estilo (Selectores)

## Parte 1

# 1. Introducción a CSS

Las hojas de estilo en cascada (CSS - Cascading Style Sheets) son un estándar que define la presentación de los documentos Web, es decir, el modo en el que se muestra un documento en pantalla o se suministra al usuario, ya sea por el monitor, en la pantalla del teléfono móvil o leído por un lector de pantalla. Lo más importante es que con CSS se mantienen las instrucciones de presentación separadas del contenido del documento HTML.



## 2. Añadir estilos a un documento con CSS

Ventajas:

- **Mayor control en el diseño de las páginas:** Se puede llegar a diseños fuera del alcance de HTML.
- **Menos trabajo:** Se puede cambiar el estilo de todo un sitio con la modificación de un único archivo.
- **Documentos más pequeños:** Las etiquetas <font> y la gran cantidad de tablas empleadas para dar una buena apariencia a los sitios web desaparecen ahora, por lo que se ahorra código en la configuración de la presentación del sitio.
- **Documentos mucho más estructurados:** Los documentos bien estructurados son accesibles a más dispositivos y usuarios.
- **Tiene buen soporte:** En este momento, casi todos los navegadores soportan casi toda la especificación CSS1 y la mayoría también las recomendaciones de nivel 2 y 2.1.

## 2. Añadir estilos a un documento con CSS

### ¿Cómo funciona CSS?

1. Hay que comenzar con un documento HTML. En teoría, el documento tendrá una estructura lógica y un significado semántico a través de los elementos HTML adecuados.
2. Luego hay que escribir las reglas de estilo para definir el aspecto ideal de todos los elementos. Las reglas seleccionan el elemento en cuestión por su nombre y, a continuación, listan las propiedades (fuente, color, etc.) y los valores que se le van a aplicar.
3. Por último, hay que vincular los estilos al documento. Las reglas de estilo pueden reunirse en un documento independiente y aplicarse a todo el sitio, o pueden aparecer en la cabecera y aplicarse sólo a ese documento.

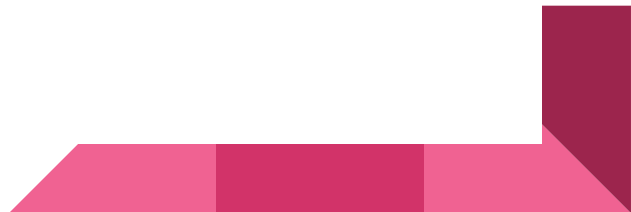
## 2. Añadir estilos a un documento con CSS

### CSS en línea

Los estilos en línea son declaraciones CSS que se integran en las etiquetas HTML mediante el atributo `style`. Este método tan solo afecta al elemento en el que se integra el código. El CSS en línea es complicado de entender y mantener ya que mezcla los estilos CSS con el código HTML.

*Ejemplo:*

```
<p style="color:green">Párrafo de color verde.</p>
```



## 2. Añadir estilos a un documento con CSS

### CSS incrustado en la cabecera

Otra manera muy simple de añadir estilo con CSS es utilizando la etiqueta `<style>` en la cabecera `<head>` del fichero HTML del sitio. La desventaja de este método es que a la hora de realizar cualquier cambio, se debe realizar en múltiples páginas diferentes y el código estará repetido. Su uso puede llegar a ser necesario en el caso de utilizar un gestor de contenido que no permita modificar el archivo CSS directamente.

Ejemplo:

```
<html>

<head>

    <title>CSS incrustado en la
cabecera</title>

    <style> p { color: green; } </style>

</head>

<body>

    <p>Párrafo de color verde.</p>

</body>

</html>
```

## 2. Añadir estilos a un documento con CSS

### CSS en hojas de estilo externas

Mediante hojas de estilo externas se consigue separar el archivo de estilos del fichero HTML. El archivo de estilos cuenta con la extensión .css y se referencia desde HTML mediante el elemento <link>. Este es el método más eficiente y más sencillo de mantener ya que el código CSS se encuentra separado del fichero HTML.

Atributos importantes:

- **rel:** indica la relación del documento enlazado con el actual. El uso más común para este atributo es especificar el enlace a una hoja de estilos externa: stylesheet
- **href:** establece con la URL de la hoja de estilos externa para dar formato a la página.

# 3. Selectores en CSS

## Selector universal

Se utiliza para seleccionar todos los elementos de la página.

El selector universal se indica mediante un asterisco (\*). A pesar de su sencillez, no se utiliza habitualmente, ya que es difícil que un mismo estilo se pueda aplicar a todos los elementos de una página.

**Sintaxis:** `* { atributo:valor; }`

**Ejemplo:** `* { color: grey; }`





# 3. Selectores en CSS

## Selector de etiqueta

Selecciona todos los elementos de la página cuya etiqueta HTML coincide con el valor del selector.

**Sintaxis:** etiqueta { atributo:valor }

**Ejemplo:** `p {color: green;}`

## Combinación de selectores

Si se quiere aplicar los mismos estilos a dos etiquetas diferentes, se pueden encadenar los selectores.

En este caso, CSS permite agrupar todas las reglas individuales en una sola regla con un selector múltiple. Para ello, se incluyen todos los selectores separados por una coma (,)

**Ejemplo:** `div, p { color: orange;}`



# 3. Selectores en CSS

## Selector de clase

Son los más utilizados junto con los selectores de ID que se verán a continuación. La principal característica de este selector es que en una misma página HTML varios elementos diferentes pueden utilizar el mismo valor en el atributo class.

Los selectores de clase son imprescindibles para diseñar páginas web complejas, ya que permiten disponer de una precisión total al seleccionar los elementos. Además, estos selectores permiten reutilizar los mismos estilos para varios elementos diferentes.

```
.aviso {  
    padding: 0.5em;  
    border: 1px solid #98be10;  
    background: #f6feda;  
}
```

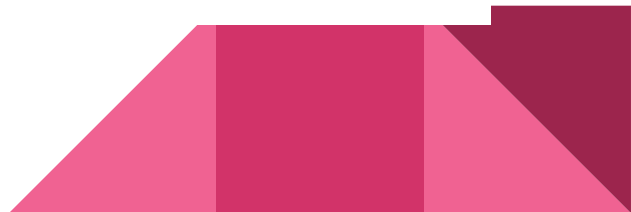


### 3. Selectores en CSS

#### Selector de clase

En ocasiones, es necesario restringir el alcance del selector de clase → Combinando el selector de tipo y el selector de clase, se obtiene un selector mucho más específico.

```
/* Todos los elementos de tipo "p" con atributo class="aviso" */  
p.aviso { ... }
```



## Ejercicio

Explica qué pasaría en los siguientes casos:

p.aviso { ... }

p, .aviso { ... }



# 3. Selectores en CSS

## Selector de ID

En ocasiones, es necesario aplicar estilos CSS a un único elemento de la página. Aunque puede utilizarse un selector de clase para aplicar estilos a un único elemento, existe otro selector más eficiente en este caso.

El selector de ID permite seleccionar un elemento de la página a través del valor de su atributo id. Este tipo de selectores sólo seleccionan un elemento de la página porque el valor del atributo id no se puede repetir en dos elementos diferentes de una misma página.

La sintaxis de los selectores de ID es muy parecida a la de los selectores de clase, salvo que se utiliza el símbolo de la almohadilla (#) en vez del punto (.) como prefijo del nombre de la regla CSS.

**Sintaxis:** #id { atributo:valor }

**Ejemplo:** `#cent {color: blue;}`


# 3. Selectores en CSS

## Selector de ID

Al igual que los selectores de clase, en este caso también se puede restringir el alcance del selector mediante la combinación con otros selectores.

A primera vista, restringir el alcance de un selector de ID puede parecer absurdo. Pero, en este caso, algunas páginas pueden disponer de elementos con un atributo id igual a aviso y que no sean párrafos, por lo que la regla anterior no se aplica sobre esos elementos.

```
/* Todos los elementos de tipo "p" con atributo id="aviso" */  
p#aviso { ... }
```



## Ejercicio

Explica qué pasaría en los siguientes casos:

`p#aviso { ... }`

`p, #aviso { ... }`



### 3. Selectores en CSS

Selector	Descripción
<b>*</b>	Selecciona todos los elementos del DOM
<b>etiqueta</b>	Selecciona todas las etiquetas indicadas
<b>.class</b>	Selección de los elementos con la clase .class
<b>#id</b>	Selección del elemento con id #id





# 3. Selectores en CSS

## Combinación de selectores básicos

CSS permite la combinación de uno o más tipos de selectores para restringir el alcance de las reglas CSS.

La combinación de selectores puede llegar a ser todo lo compleja que sea necesario.

```
.aviso, .especial { ... }
```

```
div.aviso, span.especial { ... }
```

```
ul#menuPrincipal, li.destacado, a#inicio { ... }
```



The background is a solid pink color. In the top right corner, there is a decorative pattern of overlapping geometric shapes, including triangles and squares, in various shades of pink and magenta.

# Classroom