



CEU

| *Centro de Estudios  
Profesionales*

LENGUAJE DE MARCAS  
1º DAW – 1º DAM

---

# TEMA 12

## Uso de clases



# Atributo “class” para especificar una clase de estilos de un elemento html

El atributo “class” se utiliza a menudo para señalar el nombre de una clase en una hoja de estilo “css” (Cascading Style Sheets) Hoja de estilo en cascada.

¿Qué es CSS?

CSS, al igual que HTML es uno de los lenguajes centrales de internet.

HTML→ Se usa para añadir texto a un sitio WEB y dotarlo de interactividad.

CSS→ Para definir el diseño y aspecto gráfico del contenido se utiliza CSS.

**Como hemos comentado el CSS se utiliza como lenguaje de programación para el diseño visual de páginas web.**

Cómo es la estructura de una instrucción CSS:



## Estructura de una instrucción CSS



Ya hemos visto algo de CSS en nuestros temas anteriores, cuando le dábamos estilos a elementos en común como las tablas, en la que elegíamos el selector: `table, tr, td` y luego le poníamos el estilo:

```
table,  
tr,  
td {  
    background-color: blue;  
    font-size: 16px;  
    border: 5px solid red;  
}
```

# ¿Cómo integrar CSS en tu archivo html?

Existen 3 formas de integrar estilos en tus archivos html, hasta ahora hemos visto 2 de ellos:



## 1. Integrar estilos internos, en el propio archivo html:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {color:blue; font-size:12px;}
</style>
</head>
<body>
<h1>Esto es un encabezado</h1>
<p>Esto es un párrafo</p>
</body>
</html>
```

Esta forma de integrar estilos a nivel general de nuestro archivo html, lo hemos visto en varios temas del curso.

## 2. Integrar estilos de forma inline (En línea):

Se usan para aplicar estilos particulares a un tag en concreto:

```
<h1 style="color:blue;font-size:12px;">Encabezado con estilos inline</h1>
```

## 3. Por último, el último método, que un no lo hemos visto en los temas, es aplicar hoja de estilos externa:

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="hoja_estilos.css">
</head>
<body>
<h1> Esto es un encabezado </h1>
<p>Esto es un párrafo</p>
</body>
```



```
</html>
```

Con el atributo link, especificamos la ruta de nuestro archivo de estilos que puede ser una ruta absoluta (cargando una hoja de estilos de internet) o una ruta relativa (cargando una hoja de estilos en local).

En el ejemplo anterior: la hoja de estilos estaría en el mismo directorio que nuestro archivo html:

```
<link rel="stylesheet" href="hoja_estilos.css">
```

Dentro del archivo “hoja\_estilos.css” tendríamos el siguiente estilo:

```
h1 {color:blue; font-size:12px;}
```

# Trabajando con estilos en capas

Ahora que ya sabemos cómo trabajar con estilos, vamos a crear los siguientes archivos para empezar a realizar prácticas en esta lección.

Crearemos un archivo denominado: **tema12.html** y creamos un directorio en la misma ubicación que el archivo anterior, denominado “**estilos**”, dentro crearemos un archivo denominada **estilos.css**:



---

Tendremos esta estructura:

En el archivo tema12.html:

```
<!DOCTYPE html>
<html lang="es">

<head>
  <meta charset="UTF-8">
  <!-- ETIQUETA PARA HACER RESPONSIVE LA PÁGINA, ADAPTACIÓN DE ESCALAS -->
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tema 12 html</title>
  <!-- ICONO DE VENTANA -->
  <link rel="icon" type="image/x-icon" href="imagenes/favicon.ico">
  <!-- HOJA DE ESTILOS EXTERNA-->
  <link rel="stylesheet" href="estilos/estilos.css">
  <style>
    .city_interna {
      background-color: rgb(46, 21, 157);
      color: white;
      border: 2px solid black;
      margin: 20px;
      padding: 20px;
    }
  </style>
</head>
</html>
```

y ahora en el archivo de estilos: estilos.css:



```
estilos > # estilos.css > .city_externa
1  .city_externa {
2      background-color: tomato;
3      color: white;
4      border: 2px solid black;
5      margin: 20px;
6      padding: 20px;
7  }
```

Creamos un nombre de clase denominado “city\_externa” con todos los atributos y sus propiedades.

Ahora en el body de nuestro archivo tema12.html, vamos a poner el siguiente código html:

```
<body>
  <!-- HOJA DE ESTILOS EXTERNA PARA CAPAS -->
  <div class="city_interna">
    <h2>Londres</h2>
    <p>Londres es la capital de Inglaterra.</p>
  </div>
  <!-- HOJA DE ESTILOS INTERNA PARA CAPAS -->
  <div class="city_externa">
    <h2>Madrid</h2>
    <p>Madrid es la capital de España.</p>
  </div>
</body>
```

Si visualizamos el resultado de nuestro ejercicio:



### Londres

Londres es la capital de Inglaterra.

### Madrid

Madrid es la capital de España.

Hemos apreciado que para poder utilizar el estilo de una case, se pone el atributo (class="nombre de la clase").

Apreciamos que ambas capas han recogido sus estilos del estilo que hemos definido, uno interno y otro externo.

Qué pasaría si el nombre de la clase fuera el mismo:





```
<style>
  .city_externa {
    background-color: rgb(46, 21, 157);
    color: white;
    border: 2px solid black;
    margin: 20px;
    padding: 20px;
  }
</style>
</head>
</head>

<body>
  <!-- HOJA DE ESTILOS EXTERNA PARA CAPAS -->
  <div class="city_externa">
    <h2>Londres</h2>
    <p>Londres es la capital de Inglaterra.</p>
  </div>
  <!-- HOJA DE ESTILOS INTERNA PARA CAPAS -->
  <div class="city_externa">
    <h2>Madrid</h2>
    <p>Madrid es la capital de España.</p>
  </div>
</body>
```

Cogería el estilo último en ejecutarse. Podéis probar a poner el estilo interno por delante de la carga del archivo de estilos externo y el resultado será distinto (Es lo que ya conocéis como sobreescritura de estilos), siempre se queda con la última carga.



# Trabajando con estilos en el texto

Ahora si queremos atribuir un estilo a una parte de nuestro texto, podemos utilizar el tag `<span>` que ya lo hemos utilizado anteriormente.

Escribamos el siguiente bloque de html en nuestro archivo:

```
<body>
  <!-- HOJA DE ESTILOS EXTERNA PARA CAPAS -->
  <div class="city_externa">
    <h2>Londres</h2>
    <p>Londres es la capital de Inglaterra.</p>
  </div>
  <!-- HOJA DE ESTILOS INTERNA PARA CAPAS -->
  <div class="city_interna">
    <h2>Madrid</h2>
    <p>Madrid es la capital de España.</p>
  </div>
  <!-- HOJA DE ESTILOS EXTERNA PARA TEXTO -->
  <h1>Mi <span class="note">Importante</span> cabecera</h1>
  <p>Esto es un <span class="note">Texto Resaltado</span> de un párrafo.</p>
</body>
```

Y ahora en el archivo de hoja de estilos, añadamos el siguiente estilo:



```
.city_externa {  
  background-color: tomato;  
  color: white;  
  border: 2px solid black;  
  margin: 20px;  
  padding: 20px;  
}
```

```
.note {  
  font-size: 120%;  
  color: red;  
}
```

Podéis observar que cada nombre de estilo o clase, se van poniendo en filas separados y en bloques.

El resultado sería el siguiente:

**Londres**

Londres es la capital de Inglaterra.

**Madrid**

Madrid es la capital de España.

**Mi Importante** cabecera

Esto es un Texto Resaltado de un párrafo.



# Selectores reservados

Ahora vamos a ver como aplicamos estilos de una forma más sencilla aplicándolo de forma general a selectores reservados, ya lo hemos visto en el capítulo de tablas.

Escribamos el siguiente código:

```
<!-- HOJAS DE ESTILO REPETIDAS-->
<h2 class="city_externa">CABECERA 1</h2>
<p>Esta es la cabecera uno</p>

<h2 class="city_externa">CABECERA 2</h2>
<p>Esta es la cabecera dos</p>

<h2 class="city_externa">CABECERA 3</h2>
<p>Esta es la cabecera tres</p>
```

El resultado sería el siguiente:

CABECERA 1

Esta es la cabecera uno

CABECERA 2

Esta es la cabecera dos

CABECERA 3

Esta es la cabecera tres



Pero hemos visto que hemos tenido que repetir 3 veces el nombre de la clase en cada cabecera. Si quisiéramos el mismo estilo para todas las cabeceras de tipo “h2” podríamos optimizar nuestro código de la siguiente forma:

```
<!-- HOJAS DE ESTILO REPETIDAS-->
<h2>CABECERA 1</h2>
<p>Esta es la cabecera uno</p>

<h2>CABECERA 2</h2>
<p>Esta es la cabecera dos</p>

<h2>CABECERA 3</h2>
<p>Esta es la cabecera tres</p>
```

Hemos quitado el atributo “class” en todas las cabeceras, y ahora para dar estilo a las cabeceras de tipo “h2” incorporaremos el selector h2 a nuestra hoja de estilos:



```
.city_externa h2 {  
    background-color: tomato;  
    color: white;  
    border: 2px solid black;  
    margin: 20px;  
    padding: 20px;  
}
```

```
.note {  
    font-size: 120%;  
    color: red;  
}
```

Podrás comprobar que el resultado es el mismo.

Si quisiéramos poner otro estilo diferente a los selectores “h2” haríamos lo siguiente:



```
.city_externa {  
    background-color: ■tomato;  
    color: ■white;  
    border: 2px solid ■black;  
    margin: 20px;  
    padding: 20px;  
}  
  
h2{  
    background-color: ■rgb(99, 255, 71);  
    color: ■white;  
    border: 2px solid ■black;  
    margin: 20px;  
    padding: 20px;  
}  
  
.note {  
    font-size: 120%;  
    color: ■red;  
}
```

La diferencia es que los selectores de directivas o tag html no llevan punto delante y los nombres de clases definidos por nosotros si llevan punto delante. Prueba a ponerle un punto delante a h2 en tu hoja de estilos y verás que no coge bien el estilo.

## Uso de múltiples clases

Todos los elementos Html pueden tener una o más clases.

Para definir múltiples clases, solamente tenemos que dejar un espacio en blanco entre cada nombre de clase.

Hagamos el siguiente ejemplo:



Escribamos el siguiente código en nuestro archivo de hoja de estilos:

```
.note {  
    font-size: 120%;  
    color: ■ red;  
}  
  
.main {  
    color: ■ red;  
    font-family: Arial, Helvetica, sans-serif;  
}  
  
.aux {  
    background-color: ■ aqua;  
}
```

Y ahora en nuestro archivo html:

```
<h2>CABECERA 3</h2>  
<p>Esta es la cabecera tres</p>  
<!-- HOJAS DE ESTILO REPETIDAS-->  
<h2 class="main aux">MULTIPLES CLASES</h2>  
<p class="main aux">Esta es la cabecera tres</p>  
</body>
```

El resultado sería el siguiente:

**MULTIPLES CLASES**

Esta es la cabecera tres





---

Como podemos observar, los dos elementos html usan los estilos de dos clases.

Puedes probar a poner una propiedad color en la clase “aux” verás como sobre escribe el atributo color de la clase main, porque se carga al final.

**NOTA:**

Hasta aquí la introducción de CSS, veremos más adelante muchos más temas con CSS avanzado.