

Definición de estilos en entornos web: CSS

En este tema aprenderemos los conceptos básicos para trabajar con hojas de estilo en cascada (*Cascading Style Sheets*) o CSS que serán las que nos ayuden a dar formato a documentos XHTML o HTML.

Estas hojas de estilo se crearon para poder separar el contenido de un documento de su aspecto, lo que permite tener documentos mejor estructurados y limpios.

Para separar el contenido de la presentación se usan una serie de reglas sobre los elementos que hay que declarar previamente, ya sea en la misma página o en un documento externo.

Las hojas de estilo las podemos aplicar a un documento entero, a una parte o simplemente a una etiqueta.

Podremos modificar el aspecto de una página sin tocar su código.

Tenemos el siguiente código HTML

```
<html>
  <head>
    <title>Aplicando CSS</title>
  </head>
  <body>
    <h1>Las Cuatro Estaciones</h1>
    <ul>
      <li>Primavera</li>
      <li>Verano</li>
      <li>Otoño</li>
      <li>Invierno</li>
    </ul>
  </body>
</html>
```

Las Cuatro Estaciones

- Primavera
- Verano
- Otoño
- Invierno

Vamos a ponerle a este documento un color de fondo, al h1 le pondremos una fuente Verdana y color, y todos los elementos de la lista estarán subrayados.

inline- Podemos añadir directamente el atributo `style` en cada uno de los elementos:

```
<body style="background-color: cadetblue;">
  <h1 style="font-family: Verdana, Geneva, Tahoma, sans-serif; color:#FFFFFF;">Las Cuatro
Estaciones</h1>
  <ul>
    <li style="text-decoration: underline;">Primavera</li>
    <li style="text-decoration: underline;">Verano</li>
    <li style="text-decoration: underline;">Otoño</li>
    <li style="text-decoration: underline;">Invierno</li>
  </ul>
</body>
```

Podemos añadir los estilos directamente en el documento, añadiendo el siguiente código dentro del elemento `<head>`

```
<style>
  body{
    background-color: cadetblue;
  }
  h1{
    font-family: Verdana, Geneva, Tahoma, sans-serif;
    color:#FFFFFF;
  }
  li{
    text-decoration: underline;
  }
</style>
```

O bien en un documento externo:

Dentro de un documento *hoja_estilos.css* con el contenido

```
body{
  background-color: cadetblue;
}
h1{
  font-family: Verdana, Geneva, Tahoma, sans-serif;
  color:#FFFFFF;
}
li{
  text-decoration: underline;
}
```

Al que referenciaremos dentro del documento HTML añadiendo la siguiente línea en el contenido del elemento `<head>`

```
<head>
  <title>Aplicando CSS</title>
  <link rel="stylesheet" href="hoja_estilos.css"/>
</head>
```

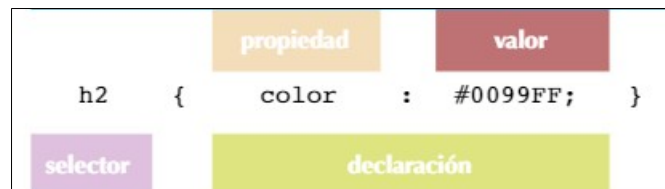
En cualquier caso, el resultado será el mismo:



Estructura y sintaxis de CSS

El funcionamiento de CSS consiste en aplicar una serie de reglas a los elementos de un documento HTML/XHTML.

Una **regla** es una declaración sobre un estilo de uno o varios elementos. Está formada por un **selector** y una **declaración**.



Prioridades y orden de CSS

La cantidad de reglas que se pueden aplicar a un documento HTML es ilimitado y pueden entrar en conflicto.

Si tenemos una regla que utiliza el selector **body** y otra que utiliza el selector **h1**, ambas afectarán al elemento **<h1>**, pero ¿cuál elegirá el navegador en caso de que más de una regla afecte a la misma propiedad de un selector?

Antes de responder a esta pregunta vamos a ver tres nuevos términos:

Herencia: la herencia provoca que los valores de las propiedades aplicadas a un elemento contenedor, afecten a los elementos contenidos en él.

Los estilos CSS se aplican en cascada y los factores que participan en la decisión de qué estilo aplicar son los siguientes:

Origen e importancia: los estilos inherentes del navegador tienen menos prioridad que los definidos por el diseñador, por lo que en caso de conflicto, serán éstos los que se apliquen. Dentro de los estilos definidos por el diseñador el orden sería:

1º inline (mayor prioridad)

2º en un elemento `<style>` (prioridad intermedia)

3º en un archivo css externo (menor prioridad)

Nivel de especificidad: cuanto más específico, mayor prioridad.

Orden de aparición: los últimos estilos en leerse o procesarse se imponen a los existentes

Además, CSS dispone de la declaración **!important** que impondrá las reglas que marque:

Si tenemos el documento

```
<html>
  <head>
    <title>Aplicando CSS</title>
    <link rel="stylesheet" href="hoja_estilos.css"/>
  </head>
  <body style="background-color: cadetblue;">
    <h1 style="font-family: Verdana, Geneva, Tahoma, sans-serif; color:#FFFFFF;">Las Cuatro
Estaciones</h1>
    <ul>
      <li style="text-decoration: underline;">Primavera</li>
      <li style="text-decoration: underline;">Verano</li>
      <li style="text-decoration: underline;">Otoño</li>
      <li style="text-decoration: underline;">Invierno</li>
    </ul>
  </body>
</html>
```

Y el CSS externo

```
body{
  background-color: pink;
}
```

El resultado será el que marque el estilo inline, definido por el atributo style de body y el resultado será

Las Cuatro Estaciones

- Primavera
- Verano
- Otoño
- Invierno

Pero si añadimos !important en el Css externo

```
body{
  background-color: pink!important;
}
```

El resultado ahora será:

Las Cuatro Estaciones

- [Primavera](#)
- [Verano](#)
- [Otoño](#)
- [Invierno](#)

Valores y unidades en CSS

Unidades de longitud absoluta (más utilizadas)

| | |
|----|---------------------------------------|
| px | píxel |
| pt | punto (1/72 pulgadas, 1 pulgada=96px) |

Unidades de longitud relativa (más utilizadas)

| | |
|-----|------------------------------------|
| em | Tamaño de letra del elemento padre |
| rem | Tamaño de letra del elemento raíz |

Ejemplo

En el siguiente ejemplo mostramos 3 textos de diferentes tamaños, dos de ellos utilizando **em**. El tamaño asignado a **pTriple** y el asignado a **pTropecientos** da el mismo resultado ya que multiplican por 3 el tamaño del tipo de letra del elemento

```
<html>
  <head>
    <title>Unidades de longitud</title>
    <style>
      #pTriple
      {
        font-size: 3em;
      }
      #pDoble
      {
        font-size: 2em;
      }
      #pTropecientos
      {
        font-size: 300%;
      }
    </style>
  </head>
  <body>
    <p>Buenas tardes de tamaño normal</p>
    <p id="pDoble">Buenas tardes al doble</p>
    <p id="pTriple">Buenas tardes al triple</p>
    <p id="pTropecientos">Buenas tardes al
300%</p>
  </body>
</html>
```

Buenas tardes de tamaño normal

Buenas tardes al doble

Buenas tardes al triple

Buenas tardes al 300%

Tipos de letras

Desde el punto de vista del diseño web existen ciertas tipografías que se pueden utilizar de manera segura ya que son comunes a todos los sistemas. Se denominan **tipos de letras seguros** y son: **Arial, Courier New, Georgia, Times New Roman, Trebuchet MS y Verdana**. De manera específica CSS define 5 tipos de letra que no tienen porque ser la misma en todos los sistemas. Estos **tipos genéricos** son: **serif, sans-serif, monospace, cursive y fantasy**

Ejemplo

```
<body>
  <p> Este texto está escrito con el tipo de letra elegido por el navegador</p>
  <p style="font-family:Verdana ;">Este texto está escrito en Verdana</p>
  <p style="font-family:monospace;">Este texto está escrito en monospace</p>
</body>
```

Este texto está escrito con el tipo de letra elegido por el navegador

Este texto está escrito en Verdana

Este texto está escrito en monospace

Se puede referenciar una fuente no genérica en la hoja de estilos CSS, si está instalada en el ordenador, se mostrará correctamente, si no es así, se mostrará la fuente por defecto. También se pueden importar fuentes de letra externas a la página web utilizando la regla **@import**, que nos permite importar un recurso a partir de una url.

```
@import url("https://fonts.googleapis.com/css2?family=Roboto&display=swap");
body{
  background-color: pink!important;
}
h1{
  font-family: roboto;
  color:#FFFFFF;
}
```

```
<html>
  <head>
<link rel="stylesheet" href="hoja_estilos.css"/>
  </head>
  <body>
    <h1>Soy Roboto</h1>
    <p> Este texto está escrito con el tipo de letra elegido por el navegador</p>
    <p style="font-family:Verdana ;">Este texto está escrito en Verdana</p>
    <p style="font-family:monospace;">Este texto está escrito en monospace</p>
  </body>
</html>
```

Soy Roboto

Este texto está escrito con el tipo de letra elegido por el navegador

Este texto está escrito en Verdana

Este texto está escrito en monospace

Puedes encontrar más fuentes de Google en <https://fonts.google.com/>






Selectores

Los selectores son el mecanismo que determina sobre qué elemento o elementos se debe aplicar un estilo. Cualquier elemento HTML puede ser un selector.

Podemos agrupar los selectores como básicos, combinadores, pseudoclases y pseudoelementos.

Selectores básicos

Los más usados son:

| Nombre | Descripción | Sintaxis |
|--------------------|--|----------------------------------|
| Universal | Todos los elementos | * |
| De tipo | Todos los elementos de un tipo determinado | nombreElemento |
| De clase | Todos los elementos de una determinada clase. Para poder usar este selector, se usa el atributo class de las etiquetas HTML a las que quiere aplicarse el estilo | .nombreClase |
| Por ID | El elemento con el identificador indicado | #identificador |
| De atributo | Los elementos que tienen un determinado atributo con unas características específicas | [atributo] elemento[atributo] |
| |  El atributo tiene un determinado valor | elemento[atributo="cadena"] |
| |  El atributo contiene un determinado valor | elemento[atributo *="cadena"] |
| |  El atributo comienza por un determinado valor | elemento[atributo ^="cadena"] |
| |  El atributo termina con un determinado valor | elemento[atributo \$="cadena"] |
| |  El atributo contiene un determinado valor en una lista de valores | elemento[atributo -="cadena"] |

Una **clase** es una definición de un estilo que, en principio, no está asociado a una etiqueta HTML, pero que podemos asociar a etiquetas concretas.

Ejemplo

```

<body>
  <h1>Resumen anual</h1>
  <form action="#">
    <input type="text" placeholder="Introduce el nombre del departamento">
    <input type="number" placeholder="Límite de ventas">
  </form>
  <p id="introduccion">Durante la temporada los resultados han sido satisfactorios</p>
  <table>
    <tr>
      <th>
        Ingresos
      </th>
      <th>
        Gastos
      </th>
      <th>
        Diferencia
      </th>
    </tr>
    <tr>
      <td>100</td>
      <td>50</td>
      <td class="positivo">50</td>
    </tr>
    <tr>
      <td>200</td>
      <td>190</td>
      <td class="positivo">10</td>
    </tr>
    <tr>
      <td>80</td>
      <td>180</td>
      <td class="negativo">-100</td>
    </tr>
  </table>
</body>

```

En los estilos tendremos

```
/*Selector universal*/
*{
    margin:5px; /*Establecemos un margen de 5px*/
}
/*Selector de elemento*/
h1{
    color:rgb(50,50,50);
    font-style: italic;
}
td{
    border-style: solid;
    border-color: black;
}
/*Selector de identificados*/
#introduccion{
    border-style: dotted;
    border-color:rgb(200,20,20);
    padding:2em; /*Establecemos un espacio entre el contenido y su borde de 2em*/
}
/*Selectores de clase*/
.positivo{
    color:green;
    font-weight: bolder;
}
.negativo{
    color:red;
    font-weight:bolder;
}
/*Selectores de atributo*/
input[type]{
    background-color: rgb(126, 144, 126);
    height: 25px;
}
input[type]{
    color:rgb(8, 37, 84);
    font-weight: bolder;
}
```

En el navegador veremos

Resumen anual

hhh

6

Durante la temporada los resultados han sido satisfactorios

Ingresos Gastos Diferencia

| | | |
|-----|-----|------|
| 100 | 50 | 50 |
| 200 | 190 | 10 |
| 80 | 180 | -100 |