



# CSS

Estilos básicos, clases e ID

Javier Ribal del Río

2025-11-28

## Table of contents

<b>1</b>	<b>Problemas de HTML</b>	<b>1</b>
<b>2</b>	<b>Formato con CSS</b>	<b>1</b>
2.1	Atributo <code>style</code> . . . . .	1
2.1.1	Problemas del atributo <code>style</code> . . . . .	2
2.2	Clases e ID . . . . .	3
2.2.1	Clases . . . . .	3
2.2.2	ID . . . . .	4
2.3	Propiedades CSS . . . . .	4
2.3.1	<code>border</code> y <code>border-radius</code> . . . . .	4
2.3.2	<code>border-top</code> , <code>border-right</code> , <code>border-bottom</code> , <code>border-left</code> . . . . .	5
2.3.3	<code>margin</code> y <code>padding</code> . . . . .	5
2.4	Archivos externos CSS . . . . .	6
<b>3</b>	<b>Etiqueta <code>&lt;meta&gt;</code></b>	<b>6</b>
<b>4</b>	<b>Git &amp; GitHub:</b>	<b>7</b>

### Contenido

- Problemas del HTML
- Formato mediante atributo `style`
- Selectores básicos (Clases e ID)
- Formato mediante etiqueta `style`
- Formato mediante hojas de estilo
- Etiqueta `<meta>`
- Git y GitHub: metodología de trabajo

## 1 Problemas de HTML

Hasta el momento hemos trabajado con HTML, que a pesar de ser muy práctico para la creación de la estructura interna de la web, presenta ciertas limitaciones a la hora de modificar el formato de la web, para ello se introduce CSS, una herramienta que nos va a permitir modificar el formato cómodamente.

## 2 Formato con CSS

### 2.1 Atributo `style`

Supongamos que tenemos el siguiente elemento



< p > Hola Mundo< p >. Por ahora el texto se mostrará en color negro, el color predeterminado de HTML para las etiquetas < p >

Hola mundo

Si se desea modificar el formato intrínseco de esta etiqueta < p > debemos incluir la propiedad **style** que, como su nombre indica nos permitirá modificar su estilo. Dentro de este atributo se ha de insertar **propiedad CSS** (no confundir con los atributos html). La propiedad CSS especifica que aspecto visual del contenido de la etiqueta. Por ejemplo, si se desea modificar el color de la letra se debe de utilizar la propiedad **color** (nótese que no lleva u).

```
<p style="color:red">Hola Mundo</p>

<!-- Equivalente en Hexadecimal -->

<p style="color:#ff0000">Hola Mundo</p>
```

Hola Mundo

Sí deseamos modificar más de una propiedad CSS deberemos de separarlas dentro de la etiqueta con un punto y coma

```
<p style="color:#ff0000; font-size: 44px">Hola Mundo</p>
```

Hola Mundo

### 2.1.1 Problemas del atributo **style**

El atributo **style** resulta útil para formatear una etiqueta individualmente, pero resulta tedioso y repetitivo para más de una etiqueta. Observe el siguiente ejemplo

```
<p>Autores del culteranismo marcados en rojo<p>

<ul>
  <li>Francisco de Quevedo</li>
  <li style="color:red; text-decoration: underline; font-size:30px;">Luis de Góngora</li>
  <li>Baltasar Gracián</li>
  <li style="color:red; text-decoration: underline; font-size:30px;">Pedro Soto de Rojas</li>
  <li>Tirso de Molina</li>
  <li>Calderón de la Barca</li>
</ul>
```

Autores del culteranismo marcados en rojo

Francisco de Quevedo

Luis de Góngora

Baltasar Gracián

Pedro Soto de Rojas

Tirso de Molina

Calderón de la Barca

#### Problemas:

- Repetición de mucho código
- Alta posibilidad error
- Poca escalabilidad



- Si añadimos por ejemplo a Luis Carrillo y Sotomayor (autor culteranista) hay que volver a repetir todo el atributo
- Si deseasemos cambiar el color de rojo a verde deberemos modificar cada uno de los atributos

## 2.2 Clases e ID

### 2.2.1 Clases

La solución a este problema son las clases que nos permiten *identificar* aquellos componentes con características de formato similares. Se agregan como atributos a cada etiqueta

```
<p>Autores del culteranismo marcados en rojo<p> <!--En este caso no se marcan en rojo-->
```

```
<ul>
  <li>Francisco de Quevedo</li>
  <li class="culteranista">Luis de Góngora</li>
  <li>Baltasar Gracián</li>
  <li class="culteranista">Pedro Soto de Rojas</li>
  <li>Tirso de Molina</li>
  <li>Calderón de la Barca</li>
</ul>
```

#### 2.2.1.1 Etiqueta <style>

Ahora para marcar en rojo los culterantistas se ha de indicar el formato de todos en la class **culteranista**, para ello nos definiremos una hoja de estilo que será común para todo el documento. La hoja de estilos se define en el **<head>** de la siguiente forma:

```
<head>
<!-- Fragmento HTML -->
<style>
  /*Fragmento CSS*/
  .culteranistas{ /*Nótese el punto*/
    color: red;
    text-size: 40px;
    background-color: blue;
    text-decoration: underline; /*Este último ';' no es obligatorio como tal, pero es recomendable, en
  }
</style>
<!--Fin de la etiqueta Style, seguimos con html-->
</head>
```

Ahora se visualizará correctamente, además se añadirá el fondo azul

#### 2.2.1.2 Composición de clases

Suponga que tenemos dos clases, una primera clase que modifica el tipo de letra a **monospace** que aplicaremos a los autores culteranistas y una segunda que modifica el grosor de la letra que se aplicará sobre los autores de lirica.

```
<head><style>
  .culteranista {
    font-family: monospace;
  }
  .lirica {
    font-weight: bold;
  }
```



```
.autores {  
    list-style: square;  
}  
</style></head>  
  
<body>  
    <ul class="autores">  
        <li class="lirica">Fray Luis de León</li>  
        <li class="culteranista">Hortensio Félix de Paravicino</li>  
        <li class="culteranista lirica">Lope de Vega</li>  
        <li class="lirica">San Juan de la Cruz</li>  
    </ul>  
</body>
```

Fray Luis de León  
Hortensio Félix de Paravicino  
Lope de Vega  
San Juan de la Cruz

## 2.2.2 ID

Los ID son como las clases pero solo se pueden utilizar una vez por sitio web. Es útil para cosas que no se deben repetir más de una vez por sitio web.

```
<style>  
#principal{  
    text-align: center  
    color:blue  
}  
</style>  
<!-- [...] -->  
<div id="principal">  
    <h4>Teatro Romántico</h4>  
    <p>El retablo romántico se caracteriza por exaltar la libertad creadora,  
    la expresión intensa de los sentimientos individuales, el gusto por lo  
    misterioso y lo sublime, y la reivindicación de la imaginación frente a  
    las reglas clásicas.</p>  
</div>
```

Teatro Romántico

El retablo romántico se caracteriza por exaltar la libertad creadora, la expresión intensa de los sentimientos individuales, el gusto por lo misterioso y lo sublime, y la reivindicación de la imaginación frente a las reglas clásicas.

## 2.3 Propiedades CSS

### 2.3.1 border y border-radius

La propiedad border es una **propiedad abreviada (shorthand)** que en realidad combina **tres propiedades más pequeñas**:

- border-width → grosor del borde



- **border-style** → estilo del borde (solid, dashed, dotted, etc.)
- **border-color** → color del borde

Por ejemplo, estas dos definiciones son **equivalentes**:

```
/* Forma abreviada */
.caja {
  border: 2px solid red;
}

/* Forma "desdoblada" en 3 mini propiedades */
.caja-desdoblada {
  border-width: 2px;
  border-style: solid;
  border-color: red;
}
```

Lope de Vega

### 2.3.2 **border-top**, **border-right**, **border-bottom**, **border-left**

CSS permite aplicar bordes distintos en cada lado. Cada uno funciona igual que border pero aplicado a un solo lado:

```
border-top
border-right
border-bottom
border-left

.caja-bordes {
  border-top: 3px solid red;
  border-right: 2px dashed blue;
  border-bottom: 4px dotted green;
  border-left: 5px double orange;
}
```

También pueden desdoblarse:

```
.caja-top-desdoblado {
  border-top-width: 4px;
  border-top-style: solid;
  border-top-color: purple;
}
```

### 2.3.3 **margin** y **padding**

Las propiedades **margin** y **padding** controlan el espacio alrededor de los elementos, pero cada una actúa en una zona distinta:

- **margin** → espacio **externo**, fuera del borde del elemento
- **padding** → espacio **interno**, entre el contenido y el borde

Visualmente:

CONTENT



```
<div style="background:#f0f0f0; padding:20px;">
    <!-- MARGIN (gris claro alrededor del borde) -->
    <!-- BORDER -->
    <div style="border:3px solid black">
        <!-- PADDING (espacio interno) -->
        <div style="background:#d0ffd0; padding:15px;">
            <!-- CONTENT -->
            <span style="background-color:white">CONTENT</span>
        </div>
    </div>
</div>
```

#### 2.3.3.1 Regla *Trouble*

Cuando `margin` o `padding` usan **4 valores**, el orden siempre es el mismo:

`top, right, bottom, left`

Para recordarlo fácilmente se utiliza la **regla mnemotécnica Trouble**:

- **T** → **Top** (arriba)
- **R** → **Right** (derecha)
- **B** → **Bottom** (abajo)
- **L** → **Left** (izquierda)

Es decir:

```
margin: top right bottom left;
padding: top right bottom left;
```

💡 Truco para los ingenieros: *La regla de la mano derecha*

El orden de los `margin` y de los `padding` siempre es en sentido horario.

Por lo que podemos utilizar la regla de la mano derecha de la inducción electro magnética asumiendo que el *vector intensidad* apunta hacia la pantalla

## 2.4 Archivos externos CSS

Incluir en la cabecera el CSS presenta ventajas como tener la página web en un único archivo. Sin embargo en la práctica normalmente trabajaremos con más de 2 archivos por lo que incluir uno más o uno menos no marca la diferencia. Por lo que podemos separar el `css` en un archivo externo `.css`. De esta forma podremos reutilizar las hojas de estilos.

```
<head>
    <title>Mi Página</title>
    <link rel="stylesheet" href="styles.css">
</head>
```

## 3 Etiqueta `<meta>`

Las etiquetas `<meta>` en HTML sirven para proporcionar metadatos sobre la página web. Estos metadatos no se muestran al usuario, pero son importantes para navegadores, buscadores, redes sociales y dispositivos.



```
<head>
  <meta charset="UTF-8"> <!-- Selecciona el conjunto de caracteres -->
  <meta name="viewport" content="width=device-width, initial-scale=1.0"> <!-- "Hace la web Responsive"
  <title>Mi Página</title>
  <link rel="stylesheet" href="styles.css">
</head>
```

## 4 Git & GitHub:

Los comandos de git nos permiten utilizar la cli para gestionar nuestro repositorio

```
git init # Inicializa repositorio (= crear repo con GitHub)
git clone <url-repo>

git add <files> # Añade los archivos a la staging area
git commit -m "<mensaje de commit>" # commit

git push # push
git pull # pull

git status # estado actual del repo
git log # historia reciente del log con poco detalle de profundidad

git diff <file> # muestra la diferencia entre la versión modificada ente un archivo y otro
```