

## Curso de CSS

¿Qué es CSS:

Antes de empezar con CSS, debemos saber las formas en las que podemos incluir contenido CSS en nuestros documentos HTML. Existen tres formas diferentes principales:

### CSS externo

En la cabecera del HTML, el bloque `<head></head>`, incluimos una relación al archivo CSS en cuestión:

```
HTML <link rel="stylesheet" type="text/css" href="index.css" />
```

De esta forma, los navegadores sabrán que deben aplicar los estilos de este archivo (`index.css`) al documento HTML actual. Se aconseja escribir esta línea lo antes posible (sobre todo, antes de scripts), obligando así al navegador a aplicar los estilos cuanto antes y eliminar la falsa sensación de página no cargada por completo.

Esta es la manera recomendada de utilizar un archivo CSS en nuestros documentos y la que utilizaremos principalmente.

También es posible incluir archivos CSS mediante la **regla `@import`** (las cuales deberían ir al principio del documento), pero esta regla sólo es recomendable en ciertas situaciones puesto que su utilización afecta al rendimiento de carga de la página, así que nos centraremos sólo en la anterior.

### CSS interno

Otra de las formas que existen para incluir estilos en un documento HTML es la de añadirlos directamente en la cabecera HTML del documento:

```
HTML <!DOCTYPE html>
<html>
<head>
  <title>Título de la página</title>
  <style type="text/css">
    div {
      background:#FFFFFF;
    }
  </style>
</head>
...
```

Este sistema puede servir en algunos casos, pero hay que tener en cuenta que utilizándolo, arruinamos la ventaja de tener los estilos en un documento independiente, por lo que siempre es preferible guardarlo en un archivo externo CSS.

### CSS embebido

Por último, la tercera forma de aplicar estilos en un documento HTML es hacerlo directamente en las propias etiquetas, a través del atributo `style`:

HTML

```
<p>¡Hola <span style="color:#FF0000">amigo lector</span>!</p>
```

Al igual que en el método anterior, se recomienda no utilizarse salvo casos muy específicos, ya que se pierde la independencia de la presentación y contenido. Sin embargo, es una opción que puede venir bien en algunos casos

### Estructura de CSS

Al igual que los documentos HTML, los documentos CSS son archivos de texto donde se escribe una serie de órdenes y el cliente (navegador) las interpreta y aplica a los documentos HTML asociados.

La estructura CSS se basa en reglas que tienen el siguiente formato:



- Selector: El selector es el elemento HTML que vamos a seleccionar del documento para aplicarle un estilo concreto. Por ejemplo, el elemento p. Realmente, esto es mucho más complejo, pero ya dedicaremos una serie de capítulos exclusivamente a este tema.
- Propiedad: La propiedad es una de las diferentes características que brinda el lenguaje CSS e iremos aprendiendo.
- Valor: Cada propiedad CSS tiene una serie de valores concretos, con los que tendrá uno u otro comportamiento.

Con todo esto le iremos indicando al navegador que, para cada etiqueta (selector especificado) debe aplicar las reglas (propiedad y valor) indicadas.

Vamos a verlo con un ejemplo para afianzar conceptos. Supongamos que este es el código HTML:

```

HTML <!DOCTYPE html>
<html>
<head>
  <title>Título de página</title>
  <link rel="stylesheet" type="text/css" href="index.css" />
</head>
<body>
  <div id="first">
    <p>Párrafo</p>
  </div>
  <div id="second">
    <span>Capa</span>
  </div>
</body>
</html>

```

## 1. Unidades:

Uno de los primeros cambios de estilo que podemos pensar realizar en un documento HTML es hacer variaciones en los colores de primer plano y de fondo. Esto es posible con las primeras dos propiedades que veremos a continuación:

Propiedad	Valor	Significado
color:	[color]	Cambia el color del texto que está en el interior de un elemento.
background-color:	[color]	Cambia el color de fondo de un elemento.

La propiedad color establece el color del texto, mientras que la propiedad background-color establece el color de fondo del elemento. Todas las propiedades CSS donde se hace referencia a colores, establecen la posibilidad de indicar 4 formas alternativas (con algunos derivados) para especificar el color deseado:

Nombre	Formato	Ejemplo
Palabra clave predefinida	[palabra]	red
Esquema RGB	rgb(rojo, verde, azul)	rgb(255, 0, 0)
Esquema RGB con canal alfa	rgba(rojo, verde, azul, alfa)	rgba(255, 0, 0, 0.25)
Esquema RGB hexadecimal	#RRGGBB	#ff0000
Esquema RGB hexadecimal abreviado	#RGB	#f00
Esquema HSL	hsl(color, saturación, brillo)	hsl(0, 100%, 100%)
Esquema HSL con canal alfa	hsla(color, saturación, brillo, alfa)	hsla(0, 100%, 100%, 0.25)

ColorPicker: Herramienta de conversión de colores entre formato RGB/RGBA, HEX y HSL/HSLA

A continuación iremos explicando cada uno de estos formatos para entender como se especifican los colores en CSS y utilizar el método que más se adapte a nuestras necesidades.

## Palabras clave de color

En el primer caso, se puede establecer el color utilizando palabras reservadas de colores, como red, blue, orange, white, navy, yellow u otras. Existen más de 140 colores diferentes y puedes verlos todos en SVG Colors.

Existen algunos valores especiales que puedes utilizar cuando quieras especificar un color, como colores transparentes o el color actual del texto, muy útil para SVG, por ejemplo.

2. Xxx

## Centrar verticalmente con CSS; 6 formas diferentes de lograrlo

1. 

```
<div id="padre">
  <div id="hijo">Texto alineado verticalmente.</div>
</div>
#hijo {
  line-height: 200px; //Damos 200px de alto para notar el efecto
}
```
2. 

```
<div id="padre">
  <div id="hijo">Contenido a centrar aquí.</div>
</div>
#padre {
  display: table;
  height:200px;
}
#hijo {
  display: table-cell;
  vertical-align: middle;
}
```
3. 

```
<div id="padre">
  <div id="hijo">Contenido alineado aquí.</div>
</div>
#padre {
  position: relative;
}
#hijo {
  position: absolute;
  top: 50%;
  left: 50%;
  height: 30%;
  width: 50%;
}
```

```

        margin: -15% 0 0 -25%;
    }
4. <div id="padre">
    <div id="hijo">Contenido aquí.</div>
</div>
#padre {
    position: relative;
}
#hijo {
    position: absolute;
    top: 0;
    bottom: 0;
    left: 0;
    right: 0;
    width: 50%;
    height: 30%;
    margin: auto;
}

5. <div id="padre">
    <div id="hijo">Contenido aquí.</div>
</div>
#padre {
    padding: 5% 0;
}
#hijo {
    padding: 10% 0;
}

6. <div id="padre">
    <div id="flotante"></div>
    <div id="hijo">Contenido aquí.</div>
</div>
#padre {height: 250px;}
#flotante {
    float: left;
    height: 50%;
    width: 100%;
    margin-bottom: -50px;
}
#hijo {
    clear: both;
    height: 100px;
}
<div id="padre">

```

```
<div id="hijo">Contenido alineado aquí.</div>
</div>
#padre {
    position: relative;
}
#hijo {
    position: absolute;
    top: 50%;
    left: 50%;
    height: 30%;
    width: 50%;
    margin: -15% 0 0 -25%;
}
```