

# **HABLA** CODE

## **Manual de CSS para principiantes**



**Carlos Vasquez**

# TABLA DE CONTENIDO

[Prefacio](#)

[Editores de código gratuitos](#)

[Introducción a CSS](#)

[Sintaxis CSS](#)

[Selectores CSS](#)

[Cómo usar CSS con HTML](#)

[Comentarios CSS](#)

[Fuentes CSS](#)

[Modelo de Caja | CSS](#)

[Donde Aprender Mas](#)

# ¡Gracias!

Antes de comenzar, me gustaría agradecer a las siguientes empresas por asociarse con hablacode y creer en nuestra misión de crear recursos de tecnología gratuitos.



# Prefacio

¡Bienvenidos!

He construido este guía para principiantes, para que sea fácil de aprender CSS (Cascading Style Sheets / Hojas de estilos en cascada). Debes de conocer los conceptos básicos de HTML para comprender CSS.

Este manual no cubre todo sobre CSS. Cubre lo que pensé que sería útil al aprender el desarrollo de front-end, que comienza con aprendiendo HTML y CSS.

A lo largo del tiempo, actualizaré el manual y agregaré más información relacionada con CSS. Compartiré otros recursos que le ayudarán a lo largo de su viaje para aprender desarrollo front-end.

Nuestro sitio web para ver y tratar ejemplos de CSS:

[https://hablacode.org/css\\_en\\_español/lección/introducción\\_a\\_css.html](https://hablacode.org/css_en_español/lección/introducción_a_css.html)

Curso HTML Desde Cero (video):

<https://youtu.be/q1Y5ipF85tA>

## Editores de código gratuitos

Si desea seguir adelante, descargue uno de los siguientes editores (gratis)

- [Visual Studio Code](#)
- [Sublime Text](#)

# Introducción a CSS

CSS es un lenguaje de hojas de estilo que se utiliza para describir la presentación de un documento escrito en un lenguaje de marcado como HTML.

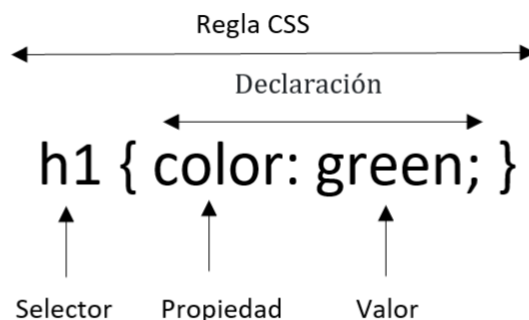
Si acabas de comenzar a aprender HTML o ya tienes experiencia, puedes notar que HTML no es más que texto en la web. Sin embargo, debe haber color, vida y estilo agregado a las páginas web y un formato común que aparece en todos los navegadores y celular digitales.

CSS es tan importante como HTML y JavaScript.

## Sintaxis CSS

CSS consiste de reglas de estilo que son interpretadas por el navegador y luego aplicadas a los elementos correspondientes como párrafos, encabezados, etc.

Una regla CSS tiene dos partes principales un selector y una o más declaraciones:



### Selector:

El selector especifica a qué elemento o elementos de la página HTML se aplica la regla CSS.

## Declaración:

Las declaraciones dentro del bloque determinan cómo se formatean los elementos en un navegador. Cada declaración consiste de una propiedad y un valor separados por dos puntos (:) y terminado en punto y coma (;), y los grupos de declaraciones están rodeados por llaves {}.

## Ejemplo:

```
h1 {  
  color: red;  
  font-size: 10px;  
}
```

En el ejemplo de arriba, hemos seleccionado el elemento h1 para que tenga el color del texto rojo y un tamaño de fuente de 10px.

Recuerda terminar cada declaración con un punto y coma (;). Esto te permitirá ingresar múltiples declaraciones, si no lo haces, tendrás errores CSS en tu página web.

---

# Selectores CSS

Los selectores CSS se utilizan para seleccionar el contenido que deseas diseñar. Como se mencionó anteriormente, los selectores son parte del

conjunto de reglas CSS. Los selectores CSS seleccionan elementos HTML de acuerdo con su id, tipo de clase, atributo, etc.

A continuación, explicaré los siguientes selectores.

## Selector Universal:

El selector universal es el `*` en CSS. Es el carácter de asterisco. Básicamente es un selector para seleccionar todos y cada uno de los tipos de elementos en una página HTML.

El siguiente ejemplo establecerá cualquier elemento html en tu documento HTML con el texto de la fuente 'times new roman', alineará el texto con el centro de la pagina y cambiará el color del texto azul.

```
* {  
  font-family: 'Times New Roman', Times, serif;  
  text-align: center;  
  color: blue;  
}
```

## Selector de tipo de elemento:

El selector de tipo coincide con el nombre de un tipo de elemento HTML. Un selector de tipo coincide con cada instancia del tipo de elemento en el árbol del documento.

A continuación, cada etiqueta de párrafo se alineará con el centro de la página y tendrá el color del texto azul.

```
p {  
  text-align: center;  
  color: blue;
```

```
}
```

## Selector de clases:

Un selector de clase es un nombre precedido por un (`.`).

Cualquier elemento que contenga la clase de título que he creado como ejemplo tendrá el elemento alineado con el centro de la página y tendrá el color de texto azul.

### Index.html

```
<h1 class="titulo">Hola Mundo</h1>
<p class="titulo">Bienvenido a CSS</p>
```

### Estilo.css

```
.titulo {
  text-align: center;
  color: blue;
}
```

## Selector de id

El selector de id es un nombre precedido por un carácter hash (`#`).

En el siguiente ejemplo, el div con el selector de id tendrá el color de fondo gris para ese contenedor y un padding de 20px.

### Index.html



```
<div id="titulo-div">
  <h1 class="titulo">Hola Mundo</h1>
  <p class="titulo">Bienvenido a CSS</p>
</div>
```

### Estilo.css

```
#titulo-div {
  background-color: gray;
  padding: 20px;
}
```

## La diferencia entre el id y el selector de clase

La diferencia entre un selector de clase y el selector de id es que el selector de id se puede usar para identificar un elemento, mientras que el selector de clase se puede usar para identificar más de un elemento.

---

# Cómo usar CSS con HTML

A continuación, les mostraré las tres formas en que pueden agregar CSS a su documento HTML.

## En línea CSS

La etiqueta `style=""` atributo se usa para crear una hoja de estilo interna.

Como puedes ver en el siguiente ejemplo. Al usar el atributo `style=""` en línea, tenemos que establecer la propiedad y el valor dentro de las comillas dobles

```
<h2 style="color: red; text-align: center;">Hello World</h2>
<p style="text-align: center;">Aprende a programar!</p>
```

## ¿Cuándo deberías usar la etiqueta de estilo en línea?

Tiene sentido utilizar una hoja de estilo interna si estás creando un página web que tenga solo un documento HTML. Si la página web es más grande, usa una hoja de estilo externo.

## Interno CSS

Una hoja de estilo interna se puede usar si una sola página HTML tiene un estilo único.

En el siguiente ejemplo, definimos nuestros estilos dentro, la etiqueta de estilo, la etiqueta de `<style>...</style>` debe estar dentro de la etiqueta de `<head>...</head>`.

```
<head>
  <style>
    body {
      background-color: green;
    }
  </style>
</head>
```

```
h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>
  <h1>Hola Mundo</h2>
</body>
```

Si la etiqueta de `<style>...</style>` se llena a ser demasiado grande, sugeriría usar una hoja de estilo externa.

## Externo CSS

Al usar CSS, es preferible mantener el CSS separado de tu HTML. Colocar CSS en un archivo permite al diseñador web diferenciar completamente entre HTML y CSS y también ayuda a que su documento HTML se vea mucho más limpio y fácil de navegar. CSS externo es un estilo que contiene solo código CSS y se guarda con una extensión `.css`. Luego, se hace referencia a este archivo CSS en su documento de HTML usando la etiqueta `<link>`.

La etiqueta `<link>` define la relación entre el documento actual y un recurso externo. Se utiliza con mayor frecuencia para relacionar a hojas de estilo externas. Es un elemento vacío, solo contiene atributos.

**index.html**

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<h1>Hola Mundo</h1>

<h2>Estoy formateado con una hoja de estilo vinculada.</h2>

</body>
</html>
```

### Estilo.css

```
h1 {
  color: green;
  text-align: center;
}

h2 {
  color: yellow;
  margin-top: 10px;
}
```

Si tuvieras que utilizar las 3 formas diferentes en tu proyecto, el estilo en línea anulará los estilos externos e internos. Esto se debe a que el navegador lee un documento HTML de arriba a abajo.

---

# Comentarios CSS

Los comentarios CSS se utilizan para agregar notas al código o para evitar que el navegador interprete partes específicas de la hoja de estilo.

Para escribir comentarios en CSS usa la siguiente sintaxis `/* */`. Se puede utilizar para una sola línea o para varias líneas.

```
/* un comentario de una línea */  
h1 {  
  color: green;  
  text-align: center;  
}
```

```
/*  
Un comentario  
que se estira  
sobre muchas  
líneas  
*/  
  
h1 {  
  color: green;  
  text-align: center;  
}
```

```
h1 {
  color: green;
  text-align: center;
}

/* h2 {
  color: yellow;
  margin-top: 10px;
} */
```

En el ejemplo de arriba el estilo CSS no será interpretado por el navegador porque esta comentado.

---

## Colores CSS

Usar colores en CSS es lo suficientemente fácil como simplemente establecer el valor en el nombre del color. Para que esto sea fácil, solo repasaremos los nombres de los colores, RGB y RGBA.

### Nombre de colores

En el siguiente ejemplo, hemos establecido el color del texto de la etiqueta de párrafo en verde. CSS proporciona 145 nombres de colores. Puedes encontrar fácilmente los otros 144 nombres de colores en el internet.

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

## RGB (Red, Green, Blue)

RGB, que significa Rojo, Verde y Azul en español.

- Cada valor puede ser un número entre 0 y 255 o un porcentaje de 0 a 100
- Un 0 para los valores de los colores será negro
- Un valor de 255 o 100% significa que está utilizando todo ese color
- Un 255 o 100% para los tres valores de color será blanco

En el siguiente ejemplo usamos porcentaje rgb para establecer el color anaranjado. También usamos números para configurar un elemento con el atributo de clase para que tenga un color de fondo amarillo.

```
p {  
  color: rgb(100%, 64.7%, 0); /* Naranja */  
}  
  
.con-principal {  
  background-color: rgb(244, 252, 21); /* Amarillo */  
}
```

## RGBA

Al igual que el RGB anterior, RGBA es lo mismo pero la **A** es para “**Alpha/Alfa**”, representa el nivel de transparencia que debe tener el color RGB.

- Puede ser un valor de 0 a 1
- Porcentaje de 0 a 100%
- Debe especificar **RGBA** en lugar de RGB

En el siguiente ejemplo hemos establecido un tipo de rojo pero con transparencia de **.4**.

```
#articulo-con {  
  background-color: rgba(228, 21, 21, .4);  
}
```

**Hola Mundo**

### Recurso:

Google tiene un buen [selector de color](#) que muestra los valores en **RGB, HEX, HSV y HSL**.

---

## Fuentes CSS



Elegir la fuente adecuada para tu sitio web es tan importante como elegir el color que debe tener tu página web. Elegir la fuente correcta permitirá a sus usuarios leer el contenido que proporcionas.

Hay miles de fuentes, pero la mayoría no son gratuitas.

Hay cinco Familias de fuentes genéricas

- **Serif**
- **Sans-serif**
- **Monospace**
- **Cursive**
- **Fantasy**

## La propiedad font-family

En CSS, usamos la propiedad **font-family** para especificar la fuente de un texto.

- Comienza con la fuente que deseas y termina con una fuente de family genérico
- Si el nombre de la fuente es una palabra, necesitas citas `"`
- Los nombres de las fuentes deben separarse con una coma

En el siguiente ejemplo, hemos establecido la fuente del texto en 'Arial' los siguientes dos valores son el sistema "alternativo" para garantizar la máxima compatibilidad entre navegadores/sistemas operativos.

```
p {  
  font-family: Arial, Helvetica, sans-serif;  
}
```

**Cómo se verá en el navegador:**

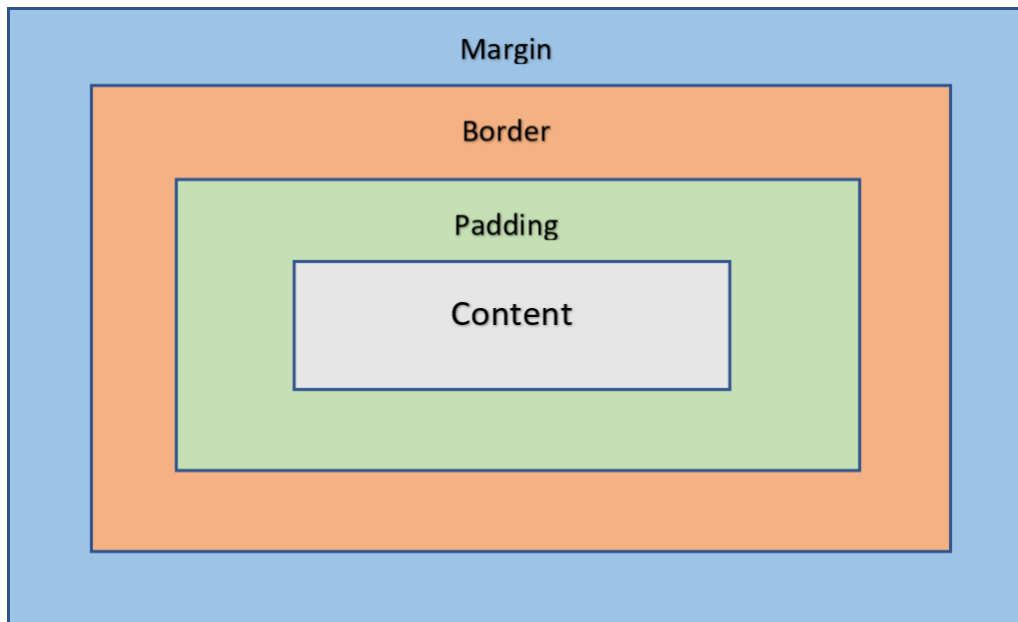
Hola Mundo

---

## Modelo de Caja | CSS

Desde la vista en perspectiva de los modelos de caja, así es como se ve un elemento en una página web, cajas dentro de cajas. El modelo de caja consta de cuatro partes:

- Margin / Margen (Espacio Exterior)
- Border / Frontera
- Padding / Relleno (Espacio interior)
- Content / Contenido (elemento)



Cada caja tiene cuatro lados arriba, abajo, izquierdo y derecho. Por lo tanto, podemos aplicar padding, border, y margin en cada uno de los lados separado o junto, lo que significa que, si solo queremos que un lado tenga margin, podemos configurarlo usando margin-left o margin-right y lo mismo se aplica a la parte de arriba y abajo margin-top o margin-bottom.

## Margin

Usamos margins para crear espacio entre elementos.

En el siguiente ejemplo, tenemos 2 contenedores, uno usa el modelo de caja y el otro no.

Puedes ver cuánto espacio hemos establecido entre el contenedor que no está usando el modelo de caja y también el lado izquierdo y la derecha de nuestro contenedor que usa el modelo de caja.

### index.html

```
<body>
  <div class="titulo-con">
    <h1>Usando el modelo de caja</h1>
  </div>
  <div class="parrafo-con">
    <h2>Sin usar el modelo de caja</h2>
  </div>
</body>
```

### estilo.css

```
body {
  background-color: rgb(205, 205, 205);
}

.titulo-con {
  background-color: green;
  color: #fff;
  margin-top: 50px;
  margin-left: 80px;
  margin-right: 150px;
  margin-bottom: 80px;
}

.parrafo-con {
  background-color: red;
  color: #fff;
}
```

## Resultado

**Usando el modelo de caja**

**Sin usar el modelo de caja**

Si quisiéramos que todos los lados tuvieran la misma cantidad de espacio, podemos usar la propiedad **margin** en lugar de especificar todos los lados.

```
.titulo-con {  
  margin: 50px;  
}
```

El código de arriba configurará todos los lados para que tengan 50px de espacio.

## Border / Frontera

Los borders son las líneas finales de un elemento.

En el siguiente ejemplo, tenemos un contenedor que tiene un relleno de 10x alrededor. También está usando la propiedad de border, y la propiedad de border toma los siguientes valores:

- Border-Width: Cuanto ancho quieres que tenga el border
- Border-style: El tipo de estilo de border (Requerido)
- Border-color: El color de la frontera

### index.html

```
<div class="titulo-con">  
  <h1>Usando el modelo de caja</h1>  
</div>
```

### estilo.css

```
.titulo-con {  
  background-color: rgb(229, 233, 148);  
  border: 3px solid rgb(247, 23, 23);  
  padding: 10px;  
}
```

## Resultado

**Usando el modelo de caja**

CSS ofrece los siguientes tipos diferentes de **border-style**

- dotted / punteado
- solid / solido
- double / doble
- dashed

Como mencionamos al comienzo de nuestra lección sobre el modelo de caja, podemos configurar los lados de la caja como queramos. Para la propiedad border, tenemos los siguiente:

- border-top / arriba
- border-bottom / abajo
- border-right / derecha
- border-left / izquierda

## Padding / Relleno

Para los ejemplos anteriores, usamos la propiedad de padding frecuentemente para ver las propiedades de css que estábamos revisando. Configuramos el padding de los elementos para crear un espacio vacío entre el contenido y el border.

El primer ejemplo no tiene relleno, y el segundo ejemplo tiene un relleno de 20px alrededor.

## Elemento sin padding

### Elemento con padding

```
.titulo-con {  
  background-color: rgb(229, 233, 148);  
  border: solid rgb(247, 23, 23);  
  padding: 20px;  
}
```

## Elemento con padding

También podemos configurar el relleno individualmente con las siguientes propiedades:

- padding-top / arriba
- padding-bottom / abajo
- padding-right / derecha
- padding-left / izquierda

---

## Donde Aprender Mas

Como mencioné anteriormente, esto es solo una introducción a CSS. Para obtener más información, puedes visitar los siguientes sitios

- Nuestra pagina web:  
[https://hablancode.org/css\\_en\\_español/lección/introducción\\_a\\_css.html](https://hablancode.org/css_en_español/lección/introducción_a_css.html)
- Suscribete a nuestro canal de Youtube ya que subiremos más videos: [Hablancode.org](https://www.youtube.com/channel/UC...)
- MDN Web Docs:  
<https://developer.mozilla.org/es/docs/Web/CSS>