Contenido

[**¿Cómo funciona la web?** 2](#_Toc201400860)

[**HTML** 2](#_Toc201400861)

[**ESTRUCTURA BÁSICA HTML** 2](#_Toc201400862)

[**ETIQUETAS** 3](#_Toc201400863)

[**<head>, <title> y <body>** 3](#_Toc201400864)

[**PARRAFOS Y ENCABEZADOS** 3](#_Toc201400865)

[**LISTAS** 4](#_Toc201400866)

[**¿Cómo hacer una lista ordenada con HTML?** 5](#_Toc201400867)

[**¿Cómo hacer una lista desordenada en HTML?** 7](#_Toc201400868)

[**Enlaces** 8](#_Toc201400869)

[**Imágenes** 9](#_Toc201400870)

[**Rutas** 9](#_Toc201400871)

[**Formularios** 10](#_Toc201400872)

[**CSS** 11](#_Toc201400873)

[**¿Cómo agregar CSS?** 11](#_Toc201400874)

[**Selectores** 12](#_Toc201400875)

[**Propiedades de fuente y texto** 13](#_Toc201400876)

[**Tipografías externas** 14](#_Toc201400877)

[**BoxModel** 15](#_Toc201400878)

[Width y Height 16](#_Toc201400879)

[Padding y Marging 16](#_Toc201400880)

# **¿Cómo funciona la web?**

* Las personas que desarrollaron la web querían que la información se pudiera ver desde cualquier lugar del mundo WWW (Word Wide Web).
* La información se almacena en un servidor y accedemos a ella a través de un enlace (URL).
* Desde la computadora se envía una petición al servidor por medio del enlace, el servidor recibe la solicitud y da una respuesta.
* Una página web, es un archivo de texto con código y se abren desde un navegador, que es el programa que lee archivo, escrito en HTML y CSS, lo interpreta y transforma en lo que el código dice y vemos en pantalla.



Ilustración 1. Como funciona la WEB

# **HTML**

* **HTML,** Hypertext Markup Language, lenguaje de marcado de hipertexto.
* Los lenguajes de marcado agarran un texto y lo presentan de forma distinta, es decir, necesitan de un programa que interprete que es lo que están queriendo decir y mostrarlo.
* Con HTML armamos la estructura de la página web a través de **etiquetas**.

## **ESTRUCTURA BÁSICA HTML**



Ilustración 2. Estructura básica de HTML

## **ETIQUETAS**

Las "tags" HTML, o "etiquetas" HTML, son códigos utilizados para "marcar" el texto de una página web, con el fin de dar instrucciones al navegador sobre cómo mostrarlo. Es decir, las etiquetas HTML son el lenguaje utilizado para estructurar y definir el contenido en un documento HTML.

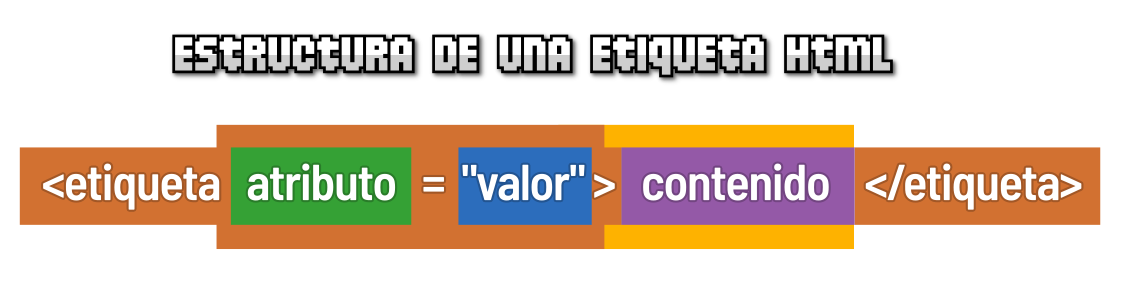


Ilustración 3. Estructura básica de una etiqueta

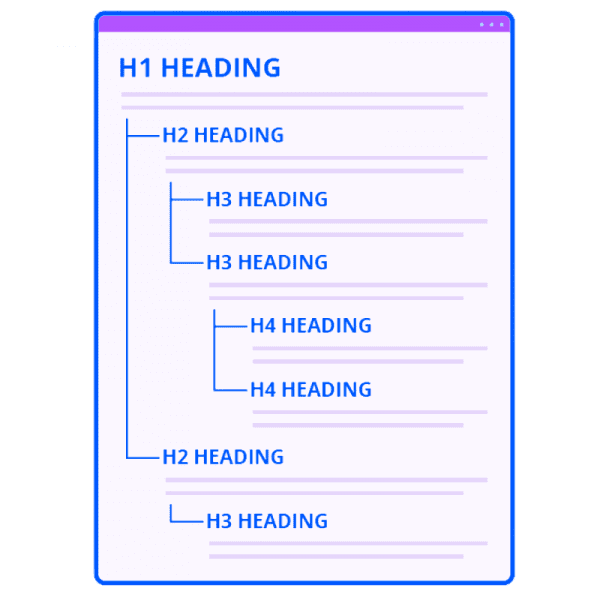
## **<head>, <title> y <body>**

<head>, <title> y <body> son elementos que se utilizan para definir la estructura de una página web:

* **<head>** Es la primera sección de la estructura de HTML, y se encuentra después de <html>. En esta sección se incluye información importante sobre la página, como el título, el conjunto de caracteres, los estilos, los scripts, entre otros. Los metadatos que se incluyen en <head> no son visibles para los usuarios, sino que describen referencias de la página.
* **<title>** Es la etiqueta que se utiliza para escribir el título de la página web. Este título aparece en la barra de título del navegador, en las páginas de resultados de los buscadores, y en la parte superior de la hoja al imprimirla o guardar la página como marcador.
* **<body>** Es la etiqueta que se utiliza para desarrollar el cuerpo de la página web, es decir, para incluir el contenido real del documento. Solo puede haber un elemento <body> en un documento.

## **PARRAFOS Y ENCABEZADOS**

En HTML, las etiquetas <h1> a <h6> representan encabezados y se utilizan para describir el contenido que se encuentra debajo de ellos. La etiqueta de párrafo o <p> en HTML ayuda a los usuarios a definir un párrafo.



## **LISTAS**

En HTML, podemos enumerar elementos en forma ordenada o desordenada: Una lista ordenada utiliza números o algún tipo de notación que indica una serie de elementos.

Por ejemplo, una lista ordenada puede comenzar con el número 1 y continuar por el 2, 3, 4, etc. Tu lista ordenada también puede comenzar con la letra A y pasar por B, C, D, etc.



Ilustración 4. Lista ordenada de alumnos

Por otro lado, tenemos listas desordenadas, como una lista de tareas, por ejemplo. ¿Soy tan apasionado por la programación que me salté el desayuno?



Ilustración 5. Lista de tareas desordenada

### **¿Cómo hacer una lista ordenada con HTML?**

En HTML, podemos crear una lista ordenada usando la etiqueta <ol>. La etiqueta ol representa una lista ordenada. Dentro de cada uno de los elementos de la lista ordenada <ol> y <ol />, tenemos que definir los elementos de la lista. Podemos definir los elementos de la lista usando la etiqueta <li>.

Aquí está la estructura HTML completa para

una lista ordenada:

<ol>

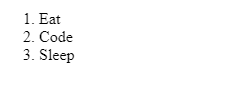
<li>Eat</li>

<li>Code</li>

<li>Sleep</li>

</ol>

El resultado de la lista ordenada anterior es:



Entonces, tenemos la lista de elementos ordenados con un número que comienza con 1 y se incrementa a 2 y 3.

**¿Qué pasa si no quieres ordenar tu lista por número?**

En su lugar, deseas ordenar usando el alfabeto como A, B, C o a, b, c. Puedes hacerlo especificando el valor del atributo type de la etiqueta <ol>.

Puedes ordenar la lista usando las **letras A, B,** C pasando A como valor de **type (tipo**).

<ol type="A">

<li>Eat</li>

<li>Code</li>

<li>Sleep</li>

</ol>

La salida se ve así:



De manera similar, puedes usar **letras minúsculas** como a para el valor del tipo y así enumerar los elementos con a, b, c, etc.

<ol type="a">

<li>Eat</li>

<li>Code</li>

<li>Sleep</li>

</ol>

Aquí está la salida:



Si desea utilizar **números romanos**, utilice el valor I para una lista ordenada con números romanos:

<ol type="I">

<li>Eat</li>

<li>Code</li>

<li>Sleep</li>

</ol>

La salida se ve así:



**¿Cómo usar el atributo start en listas HTML?**

El elemento <ol> tiene un atributo interesante llamado start. Puedes especificar un valor del atributo de inicio ("start") para iniciar la lista ordenada desde un número específico.

Digamos que deseas comenzar la lista con el número 30 en lugar de 1. Puedes especificar el número 30 como el valor del atributo start de esta manera:

<ol start="30">

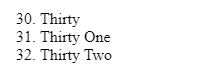
<li>Thirty</li>

<li>Thirty One</li>

<li>Thirty Two</li>

</ol>

La salida se ve así:



### **¿Cómo hacer una lista desordenada en HTML?**

Pasemos ahora a las listas desordenadas. Usamos la etiqueta <ul> para crear una lista desordenada. Como de costumbre, necesitamos usar las etiquetas <li> dentro de <ul> y <ul/> para crear los elementos de la lista.

Los elementos de la lista (li) dentro de la lista desordenada (ul) vienen con el estilo predeterminado de viñetas, por lo que cada uno de los elementos de la lista está precedido por un punto negro.

Vamos a crear una lista de mis recursos en línea favoritos para aprender sobre programación web:

My Favorite Web Development Learning Sites

<div>

<ul>

<li>freeCodeCamp</li>

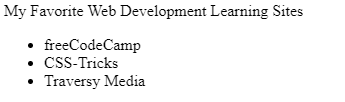
<li>CSS-Tricks</li>

<li>Traversy Media</li>

</ul>

</div>

La salida se ve así:



**Nota: Podemos personalizar el estilo de viñetas de una lista desordenada usando la propiedad de estilo CSS llamada list-style.**

## **Enlaces**

Los enlaces HTML, también conocidos como hipervínculos, son elementos que permiten enlazar un documento con otro, de modo que, al hacer clic en ellos, el usuario se dirige al nuevo documento.

Para crear un enlace en HTML, se utiliza la etiqueta <A> (de anchor) y se le especifica el atributo href, que indica la página a la que se dirige el enlace. El texto que se quiere convertir en enlace se coloca dentro de la etiqueta <A>.



Ilustración 6. Estructura de un Enlace en HTML

Algunos aspectos a tener en cuenta sobre los enlaces HTML son:

* Los enlaces se identifican por su color azul y subrayado, y al hacer clic en ellos se vuelven rojos.
* Se pueden personalizar los enlaces utilizando CSS para cambiar su color, tamaño, fuente, etc.
* Para enlazar a sitios web externos, se utilizan URLs completas.
* Para enlazar entre páginas de un mismo sitio web, se utilizan URLs relativas, que indican una ubicación relativa a la página actual.
* Para crear enlaces de correo electrónico, se utiliza el protocolo mailto.
* Para convertir una imagen en un enlace, se debe envolver la etiqueta de imagen <img> dentro de la etiqueta de ancla <a>.
* Los enlaces y anclajes definidos por el elemento A no deben estar anidado

## **Imágenes**

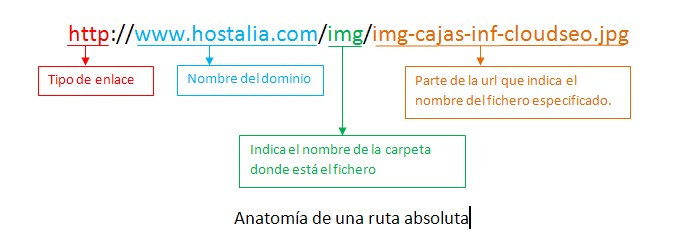
Para poner una imagen simple en una página web, utilizamos el elemento <img>. Se trata de un elemento vacío (lo que significa que no contiene texto o etiqueta de cierre) que requiere de por lo menos de un atributo para ser utilizado: src (a veces denominado por su nombre completo, source).

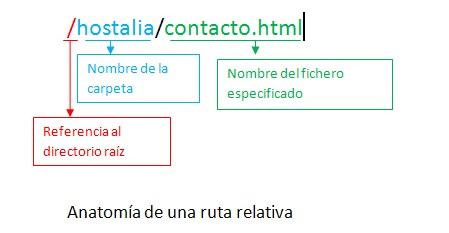
Más información: <https://desarrolloweb.com/articulos/599.php>

## **Rutas**

Las rutas de archivos HTML funcionan como mapas del camino que hay que recorrer entre los directorios de nuestro proyecto. De manera general, estas definen las rutas de carpetas y archivos dentro del mismo directorio separándolas por barras oblicuas. Para acceder a otros niveles del directorio, usamos los puntos.

Una **ruta absoluta** HTML proporciona la URL completa de un recurso, mientras que una **ruta relativa** HTML hace referencia a un recurso en función del directorio HTML actual. La elección entre un tipo de ruta de archivo HTML depende de la estructura del proyecto y del entorno de implementación.

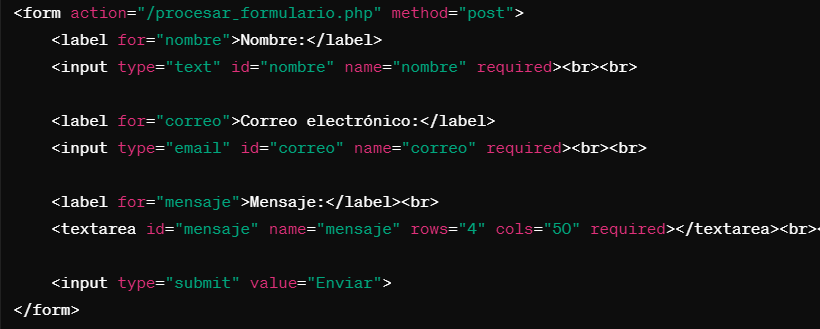




## **Formularios**

Un formulario HTML **es una caja de texto que puede contener casillas de verificación, botones y campos alfanuméricos**. Suelen encontrarse en los sitios web para recolectar los datos personales de los usuarios interesados en la oferta del sitio. Estos datos posteriormente se envían a una base de datos para ser procesados.

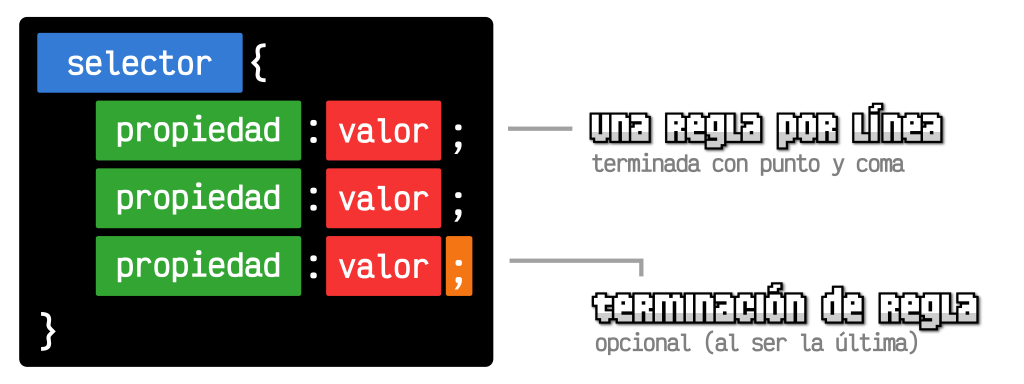
**Ejemplo,**



Más información: <https://www.apinem.com/formulario-html/>

# **CSS**

* Un elemento de HTML tiene propiedades que se definen con **CSS** (Cascading Style Sheets)
* CSS funciona mediante reglas de estilo, podemos definir color, tamaño etc.



## **¿Cómo agregar CSS?**

Se puede agregar CSS a HTML de tres formas principales:

* **En línea**: Se agrega el atributo "style" a un elemento HTML.
* **Interno**: Se agrega el código CSS dentro de una etiqueta <style> en la sección <head> del documento HTML.
* **Externo**: Se vincula el documento HTML con un archivo CSS externo.

Para agregar CSS en HTML, se puede:

1. Crear un archivo CSS con la extensión ".css".
2. Escribir las reglas CSS en el archivo.
3. Vincular el archivo CSS al documento HTML usando una etiqueta de "enlace" en la cabecera del documento HTML.

Para agregar CSS a una plantilla, se puede:

1. Abrir la plantilla en el administrador de diseño.
2. En el menú lateral derecho, en Opciones de encabezado y cuerpo, seleccionar un archivo en el menú desplegable Agregar, junto a Hojas de estilo vinculadas.

**Se puede inyectar CSS con JavaScript.**

## **Selectores**

Tabla con los principales tipos de selectores en CSS:

| **Selector** | **Uso** | **Cómo se usa** | **Ejemplo** |
| --- | --- | --- | --- |
| **Selector universal (\*)** | Selecciona todos los elementos de la página. | \* { propiedad: valor; } | \* { margin: 0; padding: 0; } |
| **Selector de etiqueta** | Aplica estilos a todos los elementos de un tipo. | etiqueta { propiedad: valor; } | p { color: blue; } |
| **Selector de clase (.)** | Aplica estilos a todos los elementos con una clase específica. | .nombreClase { propiedad: valor; } | .destacado { font-weight: bold; } |
| **Selector de ID (#)** | Aplica estilos a un único elemento con un ID específico. | #nombreID { propiedad: valor; } | #titulo { font-size: 24px; } |
| **Selector de atributo** | Selecciona elementos con un atributo específico. | [atributo] { propiedad: valor; } | [type="text"] { border: 1px solid black; } |
| **Selector de descendencia** | Aplica estilos a elementos dentro de un contenedor. | contenedor elemento { propiedad: valor; } | div p { color: red; } |
| **Selector de hijo directo (>)** | Selecciona solo los hijos directos de un elemento. | padre > hijo { propiedad: valor; } | ul > li { list-style: none; } |
| **Selector de hermano general (~)** | Selecciona los elementos hermanos después de un elemento. | elemento1 ~ elemento2 { propiedad: valor; } | h1 ~ p { color: gray; } |
| **Selector de hermano adyacente (+)** | Selecciona solo el primer hermano inmediato después de un elemento. | elemento1 + elemento2 { propiedad: valor; } | h1 + p { font-style: italic; } |
| **Selector de grupo (,)** | Aplica estilos a varios elementos a la vez. | elemento1, elemento2 { propiedad: valor; } | h1, h2 { text-align: center; } |
| **Selector :hover** | Aplica estilos cuando el usuario pasa el mouse sobre un elemento. | elemento:hover { propiedad: valor; } | a:hover { color: red; } |
| **Selector :nth-child(n)** | Selecciona el enésimo hijo de un tipo dentro de su padre. | elemento:nth-child(n) { propiedad: valor; } | tr:nth-child(odd) { background: gray; } |
| **Selector ::before y ::after** | Inserta contenido antes o después de un elemento. | elemento::before { propiedad: valor; content: " "; } | p::before { content: "→ "; } |

## **Propiedades de fuente y texto**

Tabla con las principales propiedades de fuente y texto en CSS:

| **Propiedad** | **Uso** | **Cómo se usa** | **Ejemplo** |
| --- | --- | --- | --- |
| **font-family** | Define la familia de fuentes para un texto. | font-family: nombreFuente, alternativa; | font-family: Arial, sans-serif; |
| **font-size** | Especifica el tamaño de la fuente. | font-size: tamaño; | font-size: 16px; |
| **font-weight** | Define el grosor de la fuente. | font-weight: valor; | font-weight: bold; |
| **font-style** | Define el estilo de la fuente (normal, cursiva, etc.). | `font-style: normal | italic |
| **font-variant** | Convierte el texto en mayúsculas pequeñas. | font-variant: small-caps; | font-variant: small-caps; |
| **letter-spacing** | Controla el espacio entre letras. | letter-spacing: valor; | letter-spacing: 2px; |
| **word-spacing** | Controla el espacio entre palabras. | word-spacing: valor; | word-spacing: 5px; |
| **line-height** | Define la altura de línea del texto. | line-height: valor; | line-height: 1.5; |
| **text-align** | Alinea el texto (izquierda, centro, derecha, justificado). | `text-align: left | center |
| **text-decoration** | Agrega o elimina decoraciones al texto (subrayado, tachado, etc.). | `text-decoration: none | underline |
| **text-transform** | Cambia la capitalización del texto. | `text-transform: uppercase | lowercase |
| **white-space** | Controla cómo se manejan los espacios en blanco en el texto. | `white-space: normal | nowrap |
| **direction** | Define la dirección del texto. | `direction: ltr | rtl;` |
| **text-shadow** | Agrega sombras al texto. | text-shadow: desplazamientoX desplazamientoY desenfoque color; | text-shadow: 2px 2px 5px gray; |

## **Tipografías externas**

En CSS, puedes usar tipografías externas de dos maneras principales:

**1. Google Fonts (Uso en línea)**

Google Fonts proporciona una forma sencilla de usar fuentes sin necesidad de subir archivos.

**¿Cómo usar Google Fonts?**

1. Ir a [Google Fonts](https://fonts.google.com/) y seleccionar una fuente.
2. Copiar el <link> y agregarlo en el <head> de tu HTML.
3. Usar font-family en CSS.

**Ejemplo:**

<head>

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&display=swap" rel="stylesheet">

</head>

body {

font-family: 'Poppins', sans-serif;

}

**2. @font-face (Uso con archivos locales o externos)**

Si tienes un archivo de fuente (.ttf, .woff, .woff2, etc.), puedes cargarlo manualmente con @font-face.

**Ejemplo con archivo local:**

@font-face {

font-family: 'MiFuente';

src: url('fonts/MiFuente.woff2') format('woff2'),

url('fonts/MiFuente.woff') format('woff');

font-weight: normal;

font-style: normal;

}

body {

font-family: 'MiFuente', sans-serif;

}

**Ejemplo con fuente externa (CDN)**

Algunas fuentes pueden cargarse desde una URL externa:

@font-face {

font-family: 'Open Sans';

src: url('https://example.com/fonts/OpenSans.woff2') format('woff2');

}

**Formatos de Fuente Soportados**

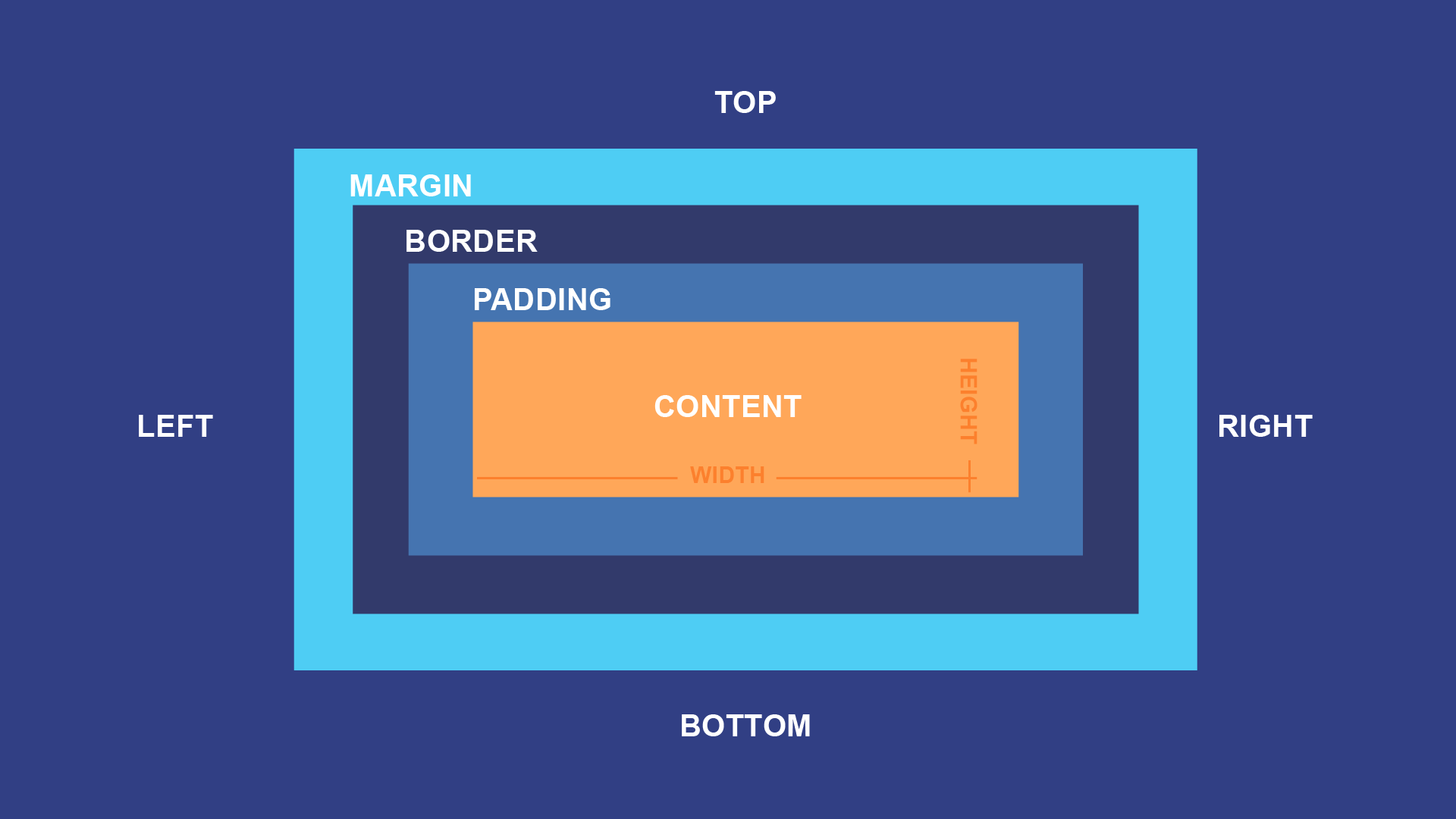
| **Formato** | **Descripción** |
| --- | --- |
| .ttf | TrueType Font, formato antiguo pero compatible. |
| .woff | Web Open Font Format, optimizado para la web. |
| .woff2 | Versión mejorada de WOFF, más comprimida. |
| .otf | OpenType Font, más avanzado que TTF. |
| .eot | Embedded OpenType, usado en Internet Explorer. |

**¿Cuál usar?**

* **WOFF2** es el más recomendado para rendimiento web.
* **WOFF** es compatible con navegadores más antiguos.
* **TTF/OTF** funcionan, pero no están optimizados para la web.

## **BoxModel**

El box model CSS es un contenedor que contiene múltiples propiedades, incluidos bordes, margen, relleno y el contenido en sí. Se utiliza para crear el diseño y el diseño de páginas web



### **Width y Height**

La propiedad **width** establece el ancho de un elemento mediante un valor numérico (con su unidad) o un porcentaje (con respecto al elemento padre). Si no se establece la anchura de un elemento, el elemento ocupa todo el ancho disponible.

La propiedad **height**establece la altura de un elemento mediante un valor numérico (con su unidad), pero no en porcentaje. Si no se establece la altura de un elemento, el elemento ocupa el espacio necesario para mostrar su contenido (sin contar los elementos flotantes).



### **Padding y Marging**

En CSS, tanto margin como padding se utilizan para crear espacio alrededor de los elementos, pero cada uno tiene un propósito diferente:

* margin (margen):

Crea espacio fuera del borde del elemento, separándolo de otros elementos o del contenedor. Se utiliza para crear espacio entre elementos.

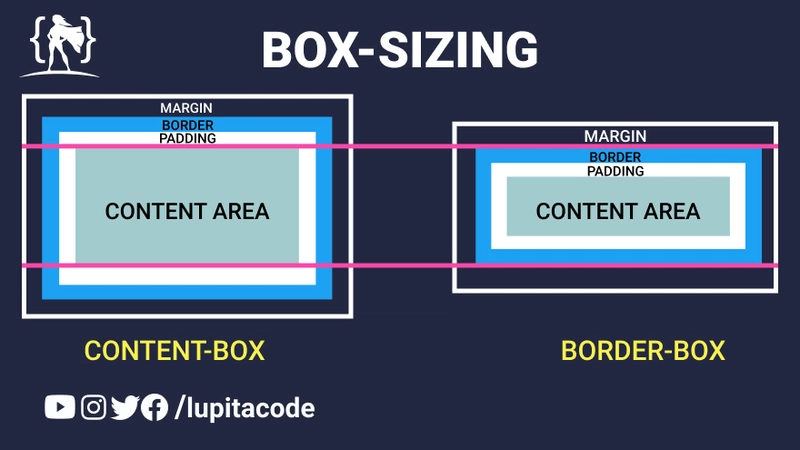
* padding (relleno): Crea espacio dentro del borde del elemento, separándolo del contenido. Se utiliza para crear espacio entre el borde y el contenido del elemento.

### **Border**

El border CSS es una propiedad que te permite definir los bordes de los elementos HTML. Puedes utilizarla para especificar el grosor, el estilo y el color del borde de un elemento en una sola declaración. Esto facilita la creación de diseños visualmente atractivos y coherentes en tu sitio web.

### **Box Sizing**

La propiedad box-sizing en CSS determina cómo se calcula el ancho y alto total de un elemento, incluyendo el contenido, relleno y borde. La propiedad box-sizing es fundamental para el modelo de caja en CSS y permite definir si el relleno y el borde se suman al ancho y alto definidos (valor por defecto content-box) o si se incluyen dentro de esos valores (valor border-box).



## **Colors**

En CSS, los colores se pueden especificar de varias maneras: usando nombres de colores predefinidos, códigos hexadecimales, valores RGB, HSL o HWB. La propiedad color se utiliza para el color del texto, mientras que background-color define el color de fondo de un elemento.

Formas de especificar colores en CSS:

**1. Nombres de colores:**

CSS tiene una lista de nombres de colores predefinidos que se pueden usar directamente, como "red", "blue", "green", etc.

**2. Códigos hexadecimales:**

Se representan con un símbolo # seguido de seis dígitos hexadecimales (0-9 y A-F), que indican la intensidad de rojo, verde y azul (ej. #FF0000 es rojo).

**3. Valores RGB:**

Representan los colores como la combinación de rojo, verde y azul, con valores enteros entre 0 y 255, o porcentajes entre 0% y 100% (ej. rgb(255, 0, 0) es rojo).

**4. Valores HSL:**

Definen el color por tono (Hue), saturación (Saturation) y luminosidad (Lightness) (ej. hsl(0, 100%, 50%) es rojo).

**5. Valores HWB:**

Definen el color por tono (Hue), blanco (Whiteness) y negro (Blackness).

**6. Funciones de color:**

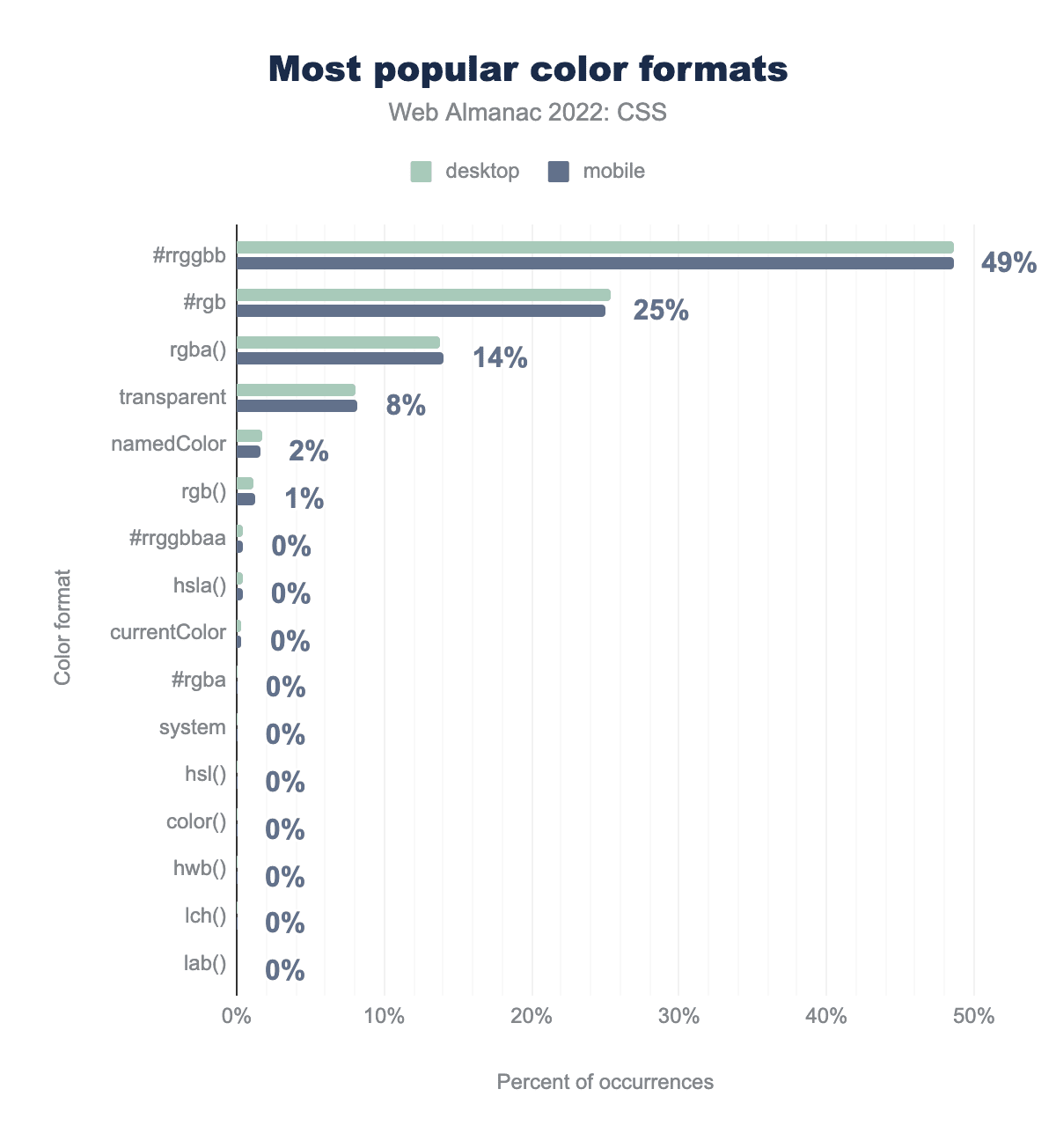
CSS también ofrece funciones como rgb(), rgba(), hsl(), hsla(), hwb(), lab(), entre otras, que permiten especificar colores y también su opacidad (transparencia).

Ejemplos:

**Color de texto:** color: blue; o color: #0000FF; o color: rgb(0, 0, 255);.

**Color de fondo:** background-color: green; o background-color: #00FF00; o background-color: hsl(120, 100%, 50%);.

**Transparencia:** rgba(255, 0, 0, 0.5) (rojo con 50% de transparencia) o hsla(0, 100%, 50%, 0.7) (rojo con 70% de transparencia).



## **CSS Units (Unidades de medida)**

En CSS, las unidades de medida se utilizan para especificar dimensiones, como longitudes y tamaños. Se dividen en dos categorías principales: unidades absolutas y unidades relativas.

Unidades Absolutas:

**px (píxeles):** La unidad más común, representa un punto en la pantalla. Se considera una unidad fija, aunque su tamaño físico puede variar dependiendo de la resolución de la pantalla.

**cm (centímetros):** Una unidad basada en el sistema métrico.

**mm (milímetros):** Una unidad basada en el sistema métrico, 1 mm = 1/10 cm.

**in (pulgadas):** Una unidad basada en el sistema imperial, 1 pulgada = 2.54 cm.

**pt (puntos):** Una unidad tradicionalmente usada en tipografía, 1 punto = 1/72 de pulgada.

**pc (picas):** Otra unidad tipográfica, 1 pica = 12 puntos = 1/6 de pulgada.

Unidades Relativas:

**em:** Relativa al tamaño de la fuente del elemento actual. Si no se especifica, es relativa al tamaño de fuente del elemento padre.

**rem:** Relativa al tamaño de la fuente del elemento raíz (<html>). Esto permite un control más consistente sobre el tamaño de los elementos en relación con el tamaño base de la página.

**% (porcentaje):** Relativa a un valor padre, como el ancho o alto del contenedor.

**vw:** Relativa al 1% del ancho de la ventana gráfica.

**vh:** Relativa al 1% de la altura de la ventana gráfica.

**vmin:** Relativa al 1% del valor menor entre el ancho y el alto de la ventana gráfica.

**vmax:** Relativa al 1% del valor mayor entre el ancho y el alto de la ventana gráfica.

**ex:** Relativa a la altura de la "x" minúscula de la fuente actual.

**ch:** Relativa al ancho del carácter "0" (cero) de la fuente actual.

**fr:** Se utiliza con la nueva especificación de diseño CSS Grid, representa una fracción del espacio disponible en la cuadrícula.

En general, las unidades relativas son preferibles para diseños responsive, ya que permiten que los elementos se adapten a diferentes tamaños de pantalla y resoluciones. Las unidades absolutas son útiles para situaciones donde se necesita un tamaño preciso, como en diseño para impresión.