

APRENDIENDO LLMM

HTML | CSS | SVG

¿Qué es HTML?

HTML es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de las siglas que corresponden a "HyperText Markup Language", es decir, "Lenguaje de Marcas de Hipertexto".

No obstante, este tipo de definiciones no nos dice mucho porque la definición es técnica. Para algunas personas al leer esto, piensan que HTML incluye el diseño gráfico de las páginas web, sin embargo, eso no es cierto ya que HTML sólo sirve para indicar como va ordenado el contenido de una página web. Esto lo hace por medio de las marcas de hipertexto las cuales son etiquetas conocidas en inglés como tags.

Hasta el momento ya tenemos una idea sobre lo que hace HTML pero aún no sabemos cómo funcionan las etiquetas por ello si quieres aprender cual es su uso y como se utiliza puedes haz click sobre la siguiente imagen:



¿Qué es SVG?

SVG son las siglas de Scalable Vector Graphics, que podríamos traducir libremente al español como gráficos basados en vectores escalables. En otras palabras, podemos decir que es un formato gráfico basado en XML para crear archivos vectoriales en 2D, con un lenguaje de marcado por medio de etiquetas. Este es un formato vectorial poco conocido pero muy útil para su uso online por su flexibilidad y por su capacidad para ofrecer gráficos con calidad.

Desde la revista de diseño, creatividad y cultura visual Graffica, comentan diversas ventajas de trabajar con este tipo de imagen, por ello si quieres aprender cual es su uso y como se utiliza puedes haz click sobre la siguiente imagen:



¿Qué es CSS?

Las siglas CSS (Cascading Style Sheets) significan «Hojas de estilo en cascada» y parten de un concepto simple pero muy potente: aplicar estilos (colores, formas, márgenes, etc...) a uno o varios documentos (generalmente documentos HTML, páginas webs) de forma masiva.

Se le denomina estilos en cascada porque se aplican de arriba a abajo (siguiendo un patrón denominado herencia que trataremos más adelante) y en el caso de existir ambigüedad, se siguen una serie de normas para resolverla.

Si quieres saber cual es su utilizacion y los diferentes atributos que utiliza puedes hacer click en la siguiente imagen:



SOBRE EL AUTOR



Nombre

Sebastián

Apellidos

Gómez Macarro

Fecha de nacimiento

14 de Abril de 1999

Formación

Actualmente cursa 1º
DAW

NICK

Mr.Gómez

ENLACES USADOS

[W3School.com/CSS](https://www.w3schools.com/css/)

[W3School.com/HTML](https://www.w3schools.com/html/)

[GitHub.io/SVG](https://github.com/)

DOCUMENTACIÓN

[PáginaWeb PDF](#)

<HTML>

¿Cómo se utiliza?

Para programar en HTML se utilizan "etiquetas". Éstas le dan formato al texto que luego aparecerá en tu sitio web, crean tablas, insertan imágenes, etc. Una etiqueta siempre va entre los símbolos "<>", de esta forma: . Todo lo demás que no esté entre los "<>" es lo que aparecerá en el sitio web.

Existen dos tipos de etiquetas: de apertura y de cierre. Las etiquetas de apertura tienen sólo una palabra clave, por ejemplo: .Las etiquetas de cierre (las más comunes) tienen dos palabras clave. La última va al final de la sentencia que quieres que aparezca en el sitio web y tiene el símbolo "/", por ejemplo: Hola, mi nombre es Dane .

Etiquetas mas utilizadas

Tan solo veremos algunas etiquetas utilizadas, empezando por las mas importantes que forman la estructura. De cada una además solo veremos utilidades concretas de estas etiquetas HTML. Si deseas mas información sobre estas tendrás un enlace al final de esta página para acceder a dicha información

Estructura básica de un documento HTML

<html> (Abrimos el documento)

<head> (Abrimos el head)

* En la cabecera o «head» se inserta todo lo referente al title, styles vinculados, scripts, etc.*

</head> (Cerramos el head)

<body> (Abrimos el body)

En el cuerpo del documento o «body», es donde va el contenido general, el grueso de la página.

</body> (Cerramos el body)

</html> (Cerramos el documento)

Las etiquetas HTML más relevantes

<title>: para definir el título de una página. Importantísimo para el SEO.



<link> y **<style>**: ambas van dentro del **<head>**, y sirven para aplicar estilos CSS a nuestro documento. Con **<link>** vinculamos de forma externa un archivo.css, y con **<style>** podemos escribir directamente código CSS dentro del head.



<h1>, **<h2>**, **<h3>**....<, **<h6>**: encabezados, numerados del 1 al 6 por orden de RELEVANCIA. El uso de estos encabezados es determinante para el SEO, porque es la mejor forma que tienen Google de entender la estructura de nuestro contenido y poder saber qué es lo relevante.



* Un pequeño consejo: no se te ocurra, meter dos o más encabezados **<h1>**. Lo mejor, es poner sólo uno y que contenga tu keyword, y luego estructurar el contenido de forma coherente con **<h2>** y **<h3>**, incluyendo en ello la keyword y variantes, no siempre la misma exactamente.

<table>, **<tr>** y **<td>**: son las tablas, las filas y las celdas. Están un poco obsoletas aunque a veces me han sacado de un apuro.



<div>: es un elemento «contenedor» de otras cosas. Realmente es lo que más se usa. Sabiendo manejar un **<div style=>lo que necesites</div>** puedes hacer casi todo. Podrás ver un ejemplo práctico al final del post.



<a>: para añadir enlaces. Dentro de ella tendremos dos atributos importantes, que son: **href=>indica_la_URL_o_RUTA** y **target=>self o _blank**, según quieras que el enlace se abra en la misma página o en una pestaña nueva.



**** es la etiqueta con la que se insertan las imágenes. Este es uno de los pocos casos que no necesita etiqueta de cierre, sino que vale con: **ruta de la imagen** **alt=>texto alternativo**



* Puedes obtener más información sobre CSS en el siguiente enlace*

[W3School.com/CSS](https://www.w3school.com/css)



¿Cómo se utiliza?

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:

```
<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 documento 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:

```
<!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`:

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

Propiedades mas utilizadas

Las principales propiedades y por tanto las mas usadas en Css son las siguientes:

Font-family:

Define la familia tipográfica. Es conveniente poner una lista de dos o tres tipografías separadas por coma, porque si el usuario no tiene instalada la tipografía que nosotros elegimos, el navegador opta por mostrar la siguiente que debería ser una similar, si tampoco la tiene instalada, mostrará la tipografía por defecto.

Tenemos 3 opciones para definir la elección de la familia tipográfica:1) usar una de las fuentes más comunes (web safe fonts) que muy probablemente tenga instalada el usuario (Arial, Verdana, Trebuchet, Georgia, san serif, Times), 2) subir nuestra propia fuente con una propiedad nueva que se llama `@font-face`, o 3) usar una fuente de Google Font, que en este momento es lo más recomendable.

Font-size:

Define el tamaño de la fuente y el valor se puede escribir en pixels o en ems. En este momento se recomienda usar ems. Los dos son valores relativos, el pixel es un valor relativo a la resolución de la pantalla, pero el em es relativo al tamaño de la fuente definida por el usuario. Si el usuario no cambió la configuración, el valor por defecto de los textos en todos los navegadores es de 16px. Entonces 1em = 16px.

Color:

Define el color de la tipografía. Los colores se pueden escribir de 3 formas distintas: con sistema hexadecimal, por ejemplo: `#FF0000` (es rojo). Con los nombres de los colores (más limitado) por ejemplo: black, red, green. O usando RGB, esta paleta permite agregar el canal alfa para hacer transparencias.

Width:

Define el ancho de un elemento, el valor se puede escribir en pixels, ems o porcentaje.

Max-width o min-width:

Definen el ancho máximo o mínimo de un elemento. Muy importante en sitios adaptables

Height:

Define el alto de un elemento, el valor se puede escribir en pixels, ems o porcentaje.

Max-height o min-height:

Definen el alto máximo o mínimo de un elemento. Muy importante en sitios adaptables.

Padding:

Es la distancia desde el borde de un elemento hasta su contenido.

Margin:

Es la distancia entre un elemento y otro (desde el borde de un elemento hacia afuera)

Border:

Define el borde de un elemento, su color, su estilo y grosor.

Background:

Define los fondos de un objeto. El fondo puede ser una imagen o un color. El color puede ser pleno o degradado. La imagen se puede repetir formando una trama (es lo que ocurre por defecto) o se puede especificar que no repita y que se coloque en determinada posición.

* Puedes obtener más información sobre CSS en el siguiente enlace*

W3School.com/CSS



¿Cuál es su uso?

SVG puede ser un formato bastante práctico para ciertos recursos web. Este formato no sólo puede ser redimensionado sin perder la calidad o definición de las imágenes que forman parte del SVG, sino que se suelen generar archivos ligeros. Por supuesto, mientras más complicada sea la ilustración o forma, el archivo SVG tendrá mayor peso, así que en ciertas situaciones puede que sea más conveniente optar por otro formato más apto.

¿Cómo se utiliza SVG?

Existen varias maneras de implementar SVG en nuestros sites:

En línea en el código HTML:

```
<svg>.....</svg>
```

Este código lo obtenemos al abrir el archivo SVG con un editor Html como Notepad++, por ejemplo.

```
<div class="wrapper">
  <p>Ejemplo de colocación de un SVG dentro del código HTML</p>
  <svg viewBox="0 0 32 24.191" style="enable-background:new 0 0 32 24.191;" xml:space="preserve">
    <g>
    </g>
  </svg>
</div>
```

Como imagen:

Insertar SVG imagen

```
<div class="wrapper">
  <button class="btnComprar">
    
    <span>COMPRAR</span>
  </button>
</div>
```

Como objeto dentro de nuestro código HTML:

SVG como objeto

```
<div class="wrapper">
  <button class="btnComprar">
    <object type="image/svg+xml" data="images/my_SVG_file.svg" class="imgBoton"></object>
    <span>COMPRAR</span>
  </button>
</div>
```

Utilizando un iframe:

Utilizar SVG utilizando un iframe

```
<iframe src="images/my_SVG_file.svg">
  <!-- fallback here -->
</iframe>
```

En nuestro CSS como background-image:

SVG como imagen background

```
nav{
  width:35px;
  height:40px;
  background:transparent url(images/my_SVG_file.svg) no-repeat 0 0;
}
```

La decisión de usar un método u otro dependerá de varios factores

Puedes obtener más información sobre SVG en el siguiente enlace

[GitHub.io/SVG](https://github.com/yourusername/SVG)