**Modulo 2**

**Desarrollo Front End Estático**

**Bienvenida al Módulo II**

Te damos la bienvenida al módulo II de #Yo Programo, en él comenzaremos a trabajar con lenguajes etiquetado y de programación que les dan forma a todas las páginas web, incorporaras los conocimientos elementales para poder diseñar tu propio porfolio web full stack que te servirá para volcar los conocimientos adquiridos y luego publicarlo así las empresas pueden conocerte.

Te invitamos en este módulo a que comiences a diseñar la maqueta web de lo que será tu proyecto integrador.

Proyecto Integrador

Tu trabajo final o proyecto integrador será la construcción de tu propio perfil, En esta ocasión te vamos a dejar un diseño base como idea inicial para tu portfolio, pero recuerda que puedes crear el diseño que desees aplicando lo aprendido, puedes descargarlo haciendo clic aquí.

**¿Qué es HTML?**

HTML significa Hyper Text Markup Language, es el lenguaje más utilizado en la Web para desarrollar páginas web. HTML fue creado por Berners-Lee a finales de 1991, pero "HTML 2.0" fue la primera especificación HTML estándar que se publicó en 1995.

HTML 4.01 era una versión principal de HTML y se publicó a finales de 1999. Aunque la versión HTML 4.01 se usa ampliamente actualmente tenemos la versión HTML-5, que es una extensión de HTML 4.01 y esta versión se publicó en 2012.

**¿Por qué aprender HTML?**

Originalmente, HTML se desarrolló con la intención de definir la estructura de documentos como encabezados, párrafos, listas, etc., para facilitar el intercambio de información científica entre investigadores. Ahora, HTML se está utilizando ampliamente para formatear páginas web con la ayuda de diferentes etiquetas disponibles en lenguaje HTML.

HTML es una OBLIGACIÓN para que los estudiantes y los profesionales se conviertan en grandes ingenieros de software, especialmente cuando están trabajando en el dominio de desarrollo web.

Enumeraremos algunas de las ventajas claves de aprender HTML:

Crear un sitio web: podemos crear un sitio web o personalizar una plantilla web existente si conocemos bien HTML.

Convertirnos en diseñadores web: si deseamos comenzar una carrera como diseñador web profesional, el diseño de HTML y CSS es una habilidad imprescindible.

Comprender la Web: si deseamos optimizar nuestro sitio web para aumentar su velocidad y rendimiento, es bueno conocer HTML para obtener los mejores resultados.

Aprender otros lenguajes: una vez que comprendamos los conceptos básicos de HTML, otras tecnologías relacionadas como javascript, php o angular se volverán más fáciles de entender.

**Aplicaciones de HTML**

Como se mencionó anteriormente, HTML es uno de los lenguajes más utilizados en la web. Vamos a enumerar algunos ejemplos y aplicaciones:

Desarrollo de páginas web: HTML se utiliza para crear páginas que se representan en la web. Casi todas las páginas de la web tienen etiquetas html para mostrar sus detalles en el navegador.

Navegación por Internet: HTML proporciona etiquetas que se utilizan para navegar de una página a otra y se utiliza mucho en la navegación por Internet.

Interfaz de usuario responsiva: las páginas HTML hoy en día funcionan bien en todas las plataformas, dispositivos móviles, pestañas, computadoras de escritorio o portátiles debido a la estrategia de diseño responsivo.

Las páginas HTML de soporte sin conexión, una vez cargadas, pueden estar disponibles sin conexión en la máquina sin necesidad de Internet.

Desarrollo de videojuegos: HTML5 tiene soporte nativo para una experiencia rica y ahora también es útil en el campo del desarrollo de videojuegos.

**Introducción a HTML**

**¿Qué es HTML?**

Por sus siglas en inglés HTML (Hypertext Markup Language) se lo define como un lenguaje de marcado estándar que nos permite crear y presentar páginas web de una manera versátil y eficiente.

El HTML describe la ESTRUCTURA y el CONTENIDO de una página web y es un lenguaje que consiste en etiquetas agrupadas o estructuradas de una manera lógica en función de lo que necesitamos como vista. Estas etiquetas le dicen al “navegador web” cómo debe mostrar el contenido.

HTML es un lenguaje y como tal tiene su propio vocabulario (palabras) y su propia gramática (reglas).

En la actualidad y desde hace varios años, el HTML está definido por el W3C. “El Consorcio World Wide Web (W3C) es una comunidad internacional que desarrolla estándares abiertos para asegurar el crecimiento a largo plazo de la Web”.

**¿Qué NO es HTML?**

HTML se diferencia de un lenguaje de programación, porque no define el COMPORTAMIENTO (lógica) de las páginas web.

HTML utiliza "marcas" para etiquetar texto, imágenes y otro contenido para mostrarlo en un navegador Web. Las marcas HTML incluyen "elementos" especiales como <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, <div>, <span>, < img>, <aside>, <audio>, <canvas>, <datalist>, <details>, <embed>, <nav>, <output>, <progress>, <video>, <ul>, <ol> , <li> y muchos otros.

Es decir, con el lenguaje HTML en sí mismo, sólo podremos presentar texto e imágenes básicas en el navegador web. Para la lógica y la presentación estética de una página web, utilizaremos otros lenguajes complementarios al HTML, tales como Javascript y CSS.

Un elemento HTML se distingue de otro texto en un documento mediante "etiquetas", que consiste en el nombre del elemento rodeado por "<" y ">". El nombre de un elemento dentro de una etiqueta no distingue entre mayúsculas y minúsculas. Es decir, se puede escribir en mayúsculas, minúsculas o una mezcla. Por ejemplo, la etiqueta <title> se puede escribir como <Title>, <TITLE> o de cualquier otra forma.

Las etiquetas vienen de a pares, la primera etiqueta del par se llama etiqueta o tag de apertura y se ve como esta: <p>. El tag o etiqueta de cierre se escribe igual que el de apertura, pero con la barra de división y se ve como esto: </p>.

Si quieres especificar que se trata de un párrafo, podrías encerrar el texto con la etiqueta de párrafo (<p>)

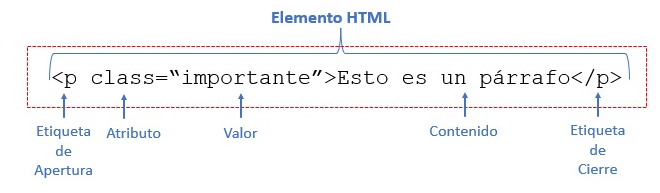
Veamos un ejemplo y su anatomía en la siguiente imagen



Las partes principales del elemento son:

1. **La etiqueta de apertura** : consiste en el nombre del elemento (en este caso p), encerrado por paréntesis angulares (<>) de apertura y cierre. Establece dónde comienza o comienza dónde a tener efecto el elemento —en este caso, dónde es el comienzo del párrafo—.
2. **La etiqueta de cierre** : es igual que la etiqueta de apertura, excepto que incluye una barra de cierre (/) antes del nombre de la etiqueta. Establece dónde termina el elemento —en este caso dónde termina el párrafo—.
3. **El contenido** : este es el contenido del elemento, que en este caso es sólo texto.
4. **El elemento** : la etiqueta de apertura, más la etiqueta de cierre, más el contenido equivalente al elemento.

Los elementos y las etiquetas no son las mismas cosas. Las etiquetas comienzan o terminan un elemento en el código fuente, mientras que los elementos son parte del DOM (Document Object Model) tema que abordaremos más adelante en detalle.



Las **etiquetas de apertura** contienen información adicional acerca del elemento, el cual no desea que aparezca en el contenido real del elemento. Aquí **clase** es el nombre del atributo e  **"importante"** es el valor del atributo. En este caso, el atributo class permite darle al elemento un nombre identificativo, que se puede utilizar luego para encontrarlo dentro de toda la página y poder aplicarle estilos entre otras cosas.

Un atributo debe tener siempre:

1. Un espacio entre este y el nombre del elemento (o del atributo previo, si el elemento ya posee uno o más atributos).
2. El nombre del atributo, seguido por un signo de igual (=).
3. Comillas de apertura y de cierre, encerrando el valor del atributo.

Los atributos siempre se incluyen en la etiqueta de apertura de un elemento, nunca en la de cierre.

**Elementos anidar**

Puedes también colocar elementos dentro de otros elementos —esto se llama anidamiento—. Si, por ejemplo, quieres resaltar una palabra del texto (en el ejemplo la palabra «muy»), podemos encerrarla en un elemento <strong>, que significa que dicha palabra se debe enfatizar:



Los elementos deben abrirse y cerrarse ordenadamente, de forma tal que se encuentren claramente.

Te mostramos el mismo ejemplo pero con los elementos mal anidados, en el ejemplo de abajo, La etiqueta de apertura del elemento <p> primero, luego la del elemento <strong>, por lo tanto, debes cerrar esta etiqueta primero, y luego la de <p>. Esto es incorrecto:

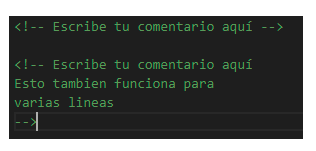


Si esto te llega a pasar, te vas a dar cuenta porque en el navegador hace cosas raras y no lo que vos esperabas ver. También el editor de código HTML que utilizas te avisará que hay algo incorrecto.

**Comentarios HTML:**

Las etiquetas de comentario son usadas para insertar comentarios en el código HTML.

Ejemplo:



Observemos que el signo de exclamación de la etiqueta de apertura no está presente en el de cierre. Los comentarios no son mostrados en el navegador web, sirven de ayuda para documentar el código fuente dentro del mismo documento HTML.

**Atributos**

Hemos visto algunas etiquetas HTML y su uso, como por ejemplo las etiquetas de encabezado  **<h1>, <h2>,**  etiqueta de párrafo  **<p>**  y otras etiquetas. Las etiquetas HTML también pueden tener atributos, que son bits adicionales de información.

Un atributo se utiliza para definir las características de un elemento HTML y se coloca dentro de la etiqueta de apertura del elemento. Todos los atributos se componen de dos partes: un **nombre** y un **valor**

* El  **nombre**  es la propiedad que desea establecer. Por ejemplo, el elemento de párrafo  **<p>**  en el ejemplo lleva un atributo cuyo nombre es **align**, que puede usar para indicar la alineación del párrafo en la página.
* El  **valor**  es lo que desea que se establezca como valor de la propiedad y siempre entre comillas. El siguiente ejemplo muestra tres valores posibles del atributo de alineación:  **left, center**  y  **right**.

Los nombres y valores de los atributos no se distinguen entre mayúsculas y minúsculas. Sin embargo, el Consorcio World Wide Web (W3C) recomienda atributos / valores de atributo en minúsculas en su recomendación HTML 4.



¿Dónde escribo el HTML?

Antes de continuar con el aprendizaje de HTML, vamos a hablar un poco de **los editores de código** .

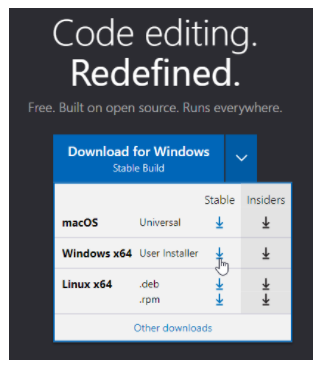
Los editores de código son programas muy sencillos que requieren poco espacio de almacenamiento y no mucho rendimiento del sistema. El dispositivo y el sistema operativo donde se utiliza básicamente depende de los gustos y necesidades del usuario. Al igual que otros tipos de software, estos programas se clasifican en software propietario, de código abierto (open source) y freeware (libre de pago), esto podría condicionar nuestra elección por uno u otro editor de código.

Existen muchos editores de código, pero aquí sólo nombraremos algunos, como por ejemplo, Visual Studio Code, Sublime Text, Notepad + + y Atom, entre otros.

Además del editor que puedes descargar y utilizar desde tu computadora, también existen editores en línea. Te vamos a dejar un editor HTML online para que puedas probarlo sin salir del curso y hagas tus experimentos, pero siempre te vamos a recomendar que lo hagas también en tu computadora con el editor que uses. Te dejamos una imágenes con algunos tips para que lo conozcas, puedes obtener más información en  [este link](https://stackblitz.com/) . 

Nosotros nos centraremos y utilizaremos para los ejemplos del editor de código a Visual Studio Code. Este editor de código es gratuito, estable, open source, robusto y posee una buena velocidad de trabajo. Estas son algunas de las principales características de Visual Studio Code que es desarrollado y mantenido por Microsoft. A continuación veremos brevemente cómo instalarlo:

1.- Desde la [página oficial de Visual Studio Code](https://code.visualstudio.com/) , descargaremos el archivo para la instalación según el sistema operativo que utilicemos. En este caso mostraremos cómo hacerlo en Windows.

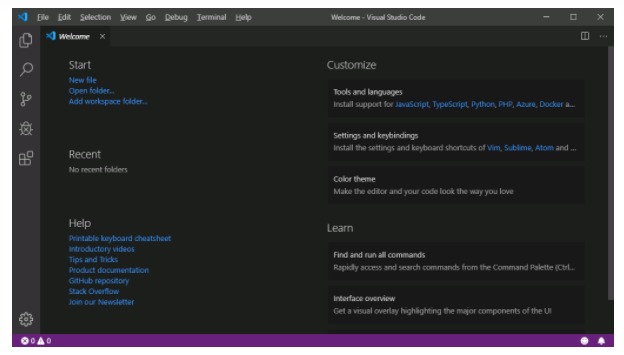


2.- Una vez descargado el archivo, hacemos clic en la barra inferior a la izquierda, donde se descargó el archivo .exe y ejecutamos el mismo.

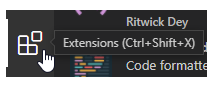


3.- Se abrirá un cuadro de diálogo y procederemos a la instalación hasta llegar a la finalización de la misma.

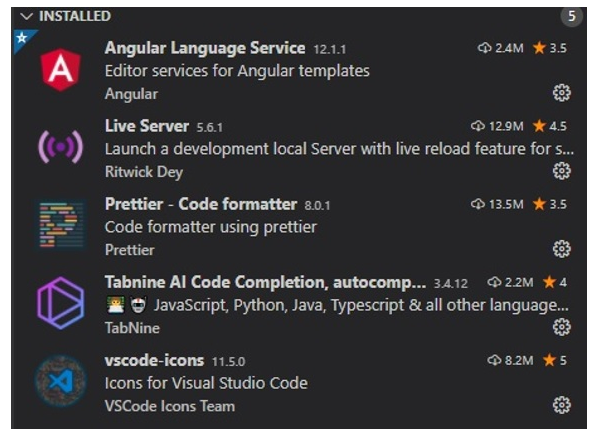
4.- Luego, una vez instalado procederemos a abrir la aplicación:



Podemos hacer clic en extensiones del menú lateral izquierdo, para agregar las extensiones que sean de nuestro agrado y nos ayuden como complemento a la hora de escribir o editar código.



Por ejemplo, podremos instalar las siguientes extensiones:



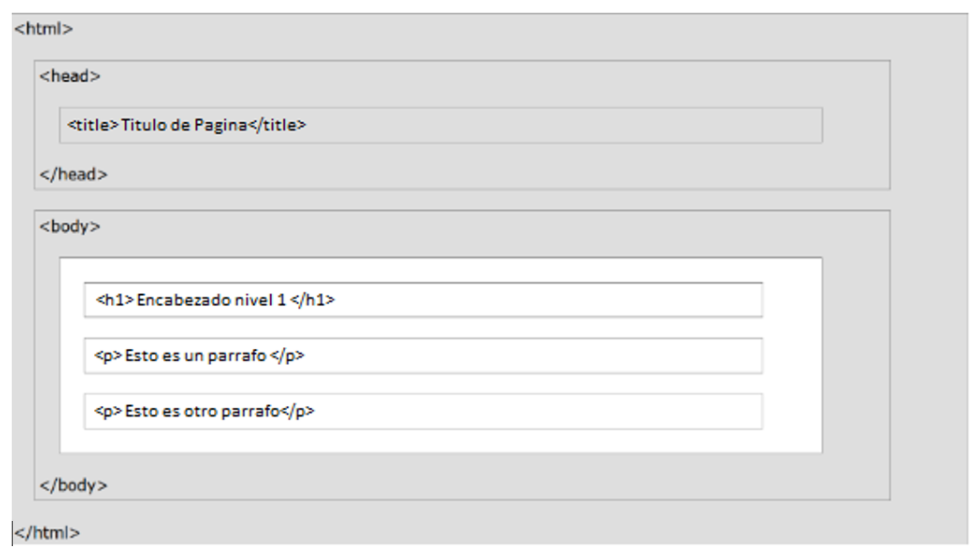
Haciendo [clic aquí](https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/editor-de-codigo/) podrás obtener más información de éste y otros editores de código.

Ahora, una vez instalado el editor de código, continuamos con el desarrollo del aprendizaje de HTML.

**Estructura documento HTML**

**Estructura básica de una página HTML**

La siguiente figura es una visualización abstracta de la estructura de una página HTML. En ella podemos observar lo que representa cada etiqueta y el contenido de texto que mostraría cada elemento. También podemos visualizar mejor la analogía con la estructura de cajas de cartón. Podríamos decir con exactitud que la caja <html> contiene en su totalidad al resto de las cajas.



Te dejamos una  [Referencia de Elementos HTML - HTML: Lenguaje de etiquetas de hipertexto | MDN (mozilla.org)](https://developer.mozilla.org/es/docs/Web/HTML/Element)  para que puedas explorar.

Ahora veamos esa estructura escrita en el código:



Observamos:

* **<! DOCTYPE html>**  - el tipo de documento. Es un preámbulo requerido. Anteriormente, cuando HTML era joven (cerca de 1991/2), los tipos de documentos actuaban como vínculos a un conjunto de reglas que el código HTML de la página debía seguir para ser considerado bueno, lo que podía significar la verificación automática de errores y algunas otras cosas de utilidad. Sin embargo, hoy día es simplemente un artefacto antiguo que a nadie le importa, pero que debe ser incluido para que todo funcione correctamente.
* **<html> </ html** > - el elemento <html>. Este elemento encierra todo el contenido de la página entera y, a veces, se conoce como el elemento raíz.
* **<head> </head>**  - el elemento <head>. Este elemento actúa como un contenedor de todo aquello que desea incluir en la página HTML que no es contenido visible por los visitantes de la página. Incluye cosas como palabras clave (keywords), una descripción de la página que quieres que aparezca en resultados de búsquedas, código CSS para dar estilo al contenido, declaraciones del juego de caracteres, etc.
* **<meta charset = "utf-8">**  - <meta>. Este elemento establece el juego de caracteres que tu documento usará en utf-8, que incluye casi todos los caracteres de todos los idiomas humanos. Básicamente, puede manejar cualquier contenido de texto que pueda incluir. No hay razón para no establecerlo, y puede evitar problemas en el futuro.
* **<title> </ title >** - el elemento <title> establece el título de tu página, que es el título que aparece en la pestaña o en la barra de título del navegador cuando la página es cargada, y se usa para describir la página cuando es añadida a los marcadores o como favorita.
* **<body> </body>**  - el elemento <body>. Encierra todo el contenido que deseas mostrar a los usuarios web que visitan tu página, ya sea texto, imágenes, videos, juegos, pistas de audio reproducibles, y demás.
* **<h1>,…, <h6>**  Los elementos de encabezado implementan seis niveles de encabezado del documento, <h1> es el más importante, y <h6>, el menos importante. Un elemento de encabezado describe brevemente el tema de la sección que presenta. La información de encabezado puede ser usada por los agentes usuarios, por ejemplo, para construir una tabla de contenidos para un documento automáticamente. Sus etiquetas son <h1>,…, <h6> y </h1>,…, </h6>.
* **<p>**  El elemento <p> (párrafo) es apropiado para distribuir el texto en párrafos. Sus etiquetas son <p> y </p>.
* **<b>** El elemento HTML <b> indica que el texto debe ser representado con una variable bold, o negrita, de la tipografía que se esté usando. Sin darle al texto importancia adicional. Sus etiquetas son <b> y </b>.
* **<strong>**   El elemento <strong> destaca el texto. Sus etiquetas son <strong> y </strong>. El elemento <strong> le da al texto más énfasis que el elemento <b>, con una importancia más alta semánticamente.
* **<i>**  El elemento HTML <i> muestra el texto marcado con un estilo en cursiva o itálica. Sus etiquetas son <i> e </i>.
* **<em>**   El elemento HTML <em> es apropiado para marcar con énfasis en el texto. El elemento <em> puede ser anidado, con cada nivel de anidamiento indicando un mayor grado de énfasis. Sus etiquetas son <em> y </em>.
* **<br>**  El elemento HTML <br> produce un salto de línea en el texto (retorno de carro). Es útil para escribir un poema o una dirección, donde la división de las líneas es significante.  No lo utilices para incrementar el espacio entre líneas de texto; para ello utiliza la propiedad margin de CSS o el elemento <p>.
* **<li>**  El elemento HTML <li> o elemento de lista declara cada uno de los elementos de una lista. Sus etiquetas son <li> e </li>.
* **<ol>**  El elemento <ol> permite definir listas o viñetas ordenadas con numeración o alfabéticamente. Sus etiquetas son <ol> y </ol>.
* **<ul>**  El elemento HTML <ul> de "lista desordenada" - lista no ordenada. crea una lista no ordenada. Sus etiquetas son <ul> y </ul>.
* **<div>**  El elemento HTML <div> es exclusivamente usado como contenedor para otros elementos HTML. En conjunto con CSS, el elemento <div> puede ser usado para agregar formato a un bloque de contenido. Sus etiquetas son <div> y </div>.
* **<img>**  El elemento HTML <img> posee los atributos src y alt pero no tiene etiqueta de cierre. Se puede representar así <img src = "imagen.png" alt = "Mi descripción de imagen"> Un elemento <img> no encierra contenido. También a este tipo de elemento se lo conoce como elemento vacío. El propósito del elemento <img> es desplegar una imagen en la página web, en el lugar que corresponde según la estructura del documento.
* El nombre de archivo de la imagen de origen está especificado por el atributo src. Los navegadores web no siempre muestran la imagen a la que el elemento hace referencia. Es el caso de los navegadores no gráficos (incluidos aquellos usados ​​por personas con problemas de visión), si el usuario elige no mostrar la imagen, o si el navegador es incapaz de mostrarla porque no es válida o soportada. En ese caso, el navegador la reemplazará con el texto definido en el atributo alt
* **<a>** El Elemento HTML Anchor <a> crea un enlace a otras páginas de Internet, archivos o ubicaciones dentro de la misma página, direcciones de correo, o cualquier otra URL que especifiquemos en sus atributos. Se puede representar así <a href="https://developer.mozilla.org/es/docs/Web/HTML/Element/a"> </a> donde la dirección del enlace está especificada por el atributo href.
* Dentro del atributo href la URL puede escribirse de forma absoluta (incluyendo el dominio) o relativa (sin incluir el dominio) solo para enlaces dentro del mismo dominio. Tanto de una forma u otra, la ruta de carpetas debe especificarse.

Siguiendo con la descripción del atributo href del elemento <a>, podemos dividir los enlaces o links en [3 tipos](https://3con14.biz/html/enlaces/13-tipos-de-enlaces.html):

1. Enlaces internos: son los que se dan entre páginas web del mismo dominio.
2. Enlaces externos: son los que se dan entre páginas web de distinto dominio.
3. Enlaces de posición (o marcadores):

A. De un lugar a otro dentro de la misma página.

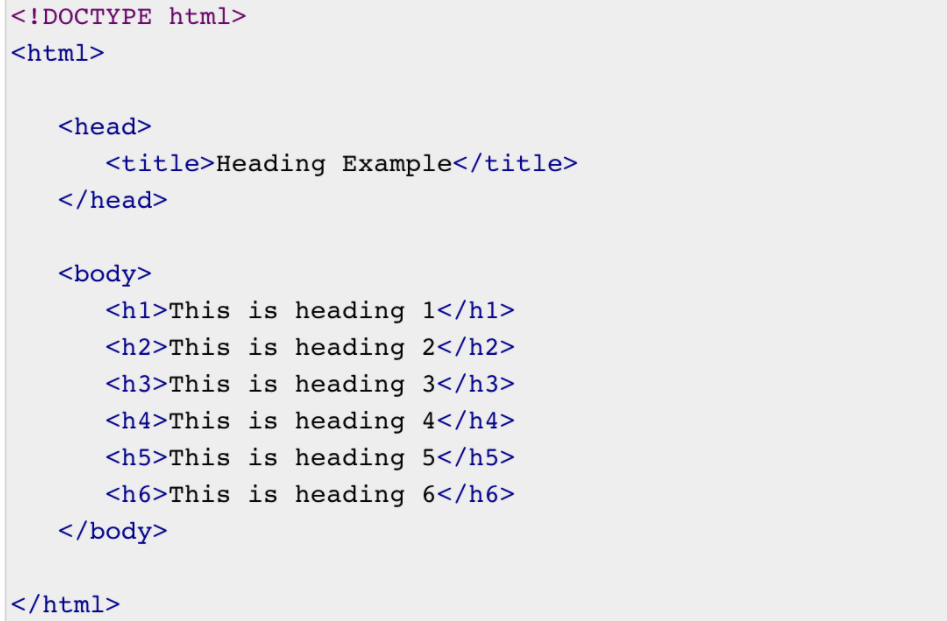
B. De un lugar a otro lugar concreto de otra página del mismo dominio.

C. De un lugar a otro lugar concreto de una página de otro dominio.

**Encabezados**

**Etiquetas de encabezado**

Cualquier documento comienza con un encabezado. Puede utilizar diferentes tamaños para sus títulos. HTML también tiene seis niveles de encabezados, que utilizan los elementos <h1>, <h2>, <h3>, <h4>, <h5> y <h6>. Mientras muestra cualquier encabezado, el navegador agrega una línea antes y una línea después de ese encabezado.



**Hagamos algunas prácticas**

Te dejamos un [IDE](https://www.redhat.com/es/topics/middleware/what-is-ide#:~:text=Un%20entorno%20de%20desarrollo%20integrado,gr%C3%A1fica%20de%20usuario%20(GUI).) con la estructura básica de una pagina HTML para que puedas escribir el código siguiendo las instrucciones de abajo:

1. Escribir las 6 líneas de código que se muestra en la imagen de arriba con las distintas etiquetas de encabezado para experimentar cómo se ve.
2. Te dejamos un texto sin formato y deberás darle formato agregando las etiquetas de encabezado.
   1. Descarga [este código HTML](https://argentinaprograma.inti.gob.ar/pluginfile.php/49694/mod_page/content/15/Etiquetas_Encabezado-CodigoPorCompletar.txt?time=1630728872464) , este texto deberás pegarlo dentro de la etiqueta <body> (cuerpo) del archivo  **index.html**  para darle formato al texto, agregando las etiquetas de encabezado, deberás dejarlo igual a  [este documento pdf](https://argentinaprograma.inti.gob.ar/pluginfile.php/49694/mod_page/content/15/Etiqueta%20encabezado%20-%20resultado%20esperado.pdf) .

Si no recuerdas algo búscalo en la web, repasa los temas anteriores, escribí una pregunta en el foro del módulo.

**Formato**

Si usas un procesador de texto, debes estar familiarizado con la capacidad de hacer que el texto esté en **negrita**, *cursiva* o subrayado; estas son solo tres de las diez opciones disponibles para indicar cómo puede aparecer el texto en HTML y XHTML:

**Texto en negrita**

Todo lo que aparece dentro del elemento  **<b> ... </b>**  se muestra en negrita

*Texto en cursiva*

Todo lo que aparece dentro del elemento  **<i> ... </i>**  se muestra en cursiva

**Texto subrayado**

Todo lo que aparece dentro del elemento  **<u> ... </u>**  se muestra subrayado

Todo lo que aparece dentro del elemento  **<strike> ... </strike>**  se muestra tachado

Fuente monoespaciada

El contenido de un elemento  **<tt> ... </tt>**  está escrito en fuente monoespaciada. La mayoría de las fuentes se conocen como fuentes de ancho variable porque diferentes letras tienen diferentes anchos (por ejemplo, la letra 'm' es más ancha que la letra 'i'). Sin embargo, en una fuente monoespaciada, cada letra tiene el mismo ancho.

Existen muchos otros tipos de fuentes, te invitamos a profundizar más sobre ello en [este link](https://www.w3schools.com/html/html_formatting.asp).

**Meta Tags**

HTML te permite especificar metadatos: información adicional importante sobre un documento de diversas formas. Los elementos META se pueden utilizar para incluir pares de nombre - valor que describen las propiedades del documento HTML, como el autor, la fecha de caducidad, una lista de palabras clave, el autor del documento, etc.

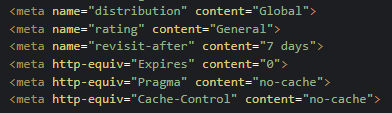
La etiqueta  **<meta>**  se utiliza para proporcionar dicha información adicional. Esta etiqueta es un elemento vacío y, por lo tanto, no tiene una etiqueta de cierre, pero lleva información dentro de sus atributos.

Puede incluir una o más meta etiquetas en su documento en función de la información que desee mantener, pero en general, las meta etiquetas no indican la apariencia física del documento, por lo que desde el punto de vista de la apariencia, no importa si se incluye o no.

Agregar meta etiquetas a sus documentos

Puedes agregar metadatos a sus páginas web colocando etiquetas <meta> dentro del encabezado del documento que está representado por las etiquetas  **<head>**  y  **</head>**. Una meta etiqueta puede tener los siguientes atributos además de los atributos principales:





**listas, imágenes y enlaces HTML**

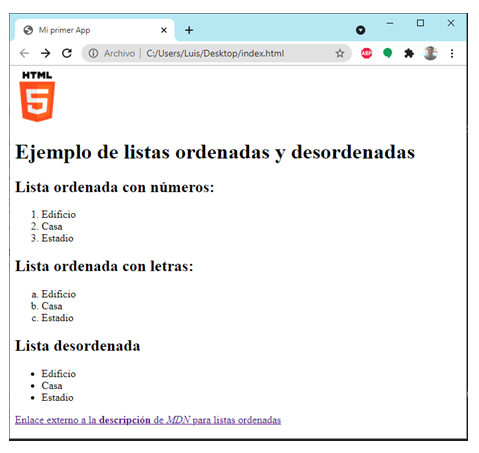
A continuación desarrollamos un ejemplo en donde utilizamos los elementos <ol>, <ul>, <img> y <a>.

En el caso del elemento <ol> si le asignamos el atributo type = “a” le indicamos al explorador web que vamos a realizar una lista ordenada alfabéticamente en minúscula. Vale aclarar que lo que se muestra ordenado es el prefijo que se asigna al ítem y no los propios nombres de cada ítem. Si no indicamos el atributo type, por defecto, la lista sale ordenada numéricamente.

En el caso del elemento <img>, se utilizó como referencia, una imagen del sitio <https://3con14.biz/html/> y se le aplicaron atributos de ancho y alto para reducir su tamaño relativo a la pantalla (los explicaremos con más profundidad en el módulo de CSS).

Con el elemento <a> insertamos un enlace a una página externa, en este caso, de [Mozilla Developer Network](https://developer.mozilla.org/es/docs/Web/HTML/Element/ol)





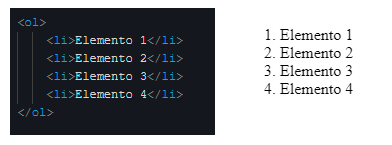
Las listas en HTML son muy importantes, no solo se utilizan para listar, sino también se utilizan para el menú, cuando ingresas en una web y tiene un menú desplegable, seguramente usaron listas que tienen un estilo y comportamiento específico, pero en el fondo son listas.

En HTML tenemos 3 tipos de listas a utilizar para diferentes usos:

1. Listas Ordenadas.
2. Listas Desordenadas.
3. Lista de Definición.

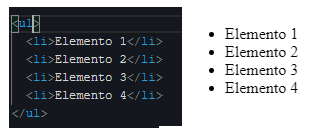
**Listas Ordenadas**

Las listas ordenadas son listas en las que el orden de los elementos **SI importa** y se indican con la etiqueta <ol> </ol>, dentro se le anida otra etiqueta <li> </li> quedando de la siguiente manera:



**Listas Desordenadas**

Las listas desordenadas son listas en las que el orden de los elementos **NO importa** y se indican con la etiqueta <ul> </ul>, dentro, se le anida otra etiqueta <li> </li> quedando de la siguiente manera:



¿Recordarás que las etiquetas pueden tener Atributos?

En las listas ordenadas **existen varios Atributos**(por ejemplo: start, type, reverse, etc) que se aplican al elemento <ol> </ol>. Investiga qué se puede hacer con esos Atributos, algunas aplican solo para las listas ordenadas y otras para listas desordenadas. Usa tus habilidades de búsqueda ya que en los ejercicios siguientes lo vas a necesitar.

**Listas de Definiciones**

Estas listas de definiciones no son tan comunes, por lo que te dejaremos este [documento PDF](https://argentinaprograma.inti.gob.ar/pluginfile.php/49697/mod_page/content/12/Practica_ListasDefiniciones_ResultadoEsperado.pdf) para que observes cómo se ven estos tipos de lista y te invitamos a que busques en internet cuáles son las etiquetas HTML que se utilizan para crear esta lista.

**Práctica** :

En el IDE de abajo, te dejamos un código HTML para que puedas completar, tienes que agregarle las etiquetas de la **lista que corresponda** para que quede igualito a [este documento PDF](https://argentinaprograma.inti.gob.ar/pluginfile.php/49697/mod_page/content/12/Practica_ListasOrdenadas_ResultadoEsperado.pdf?time=1635946973745) , para eso debes hacer lo siguiente:

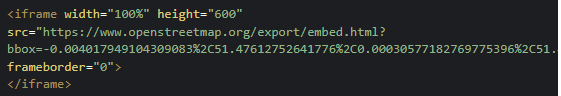
1. Descarga el documento PDF.
2. Usando el IDE de tu computadora o el que te dejamos debajo, tienes que agregar  las etiquetas y atributos necesarios en el código HTML para que se pueda visualizar igual al documento PDF.

Te dejamos el IDE, y ya sabes que si necesitas ayuda nos vemos en el foro.

**Iframe**

Puedes definir un marco en línea con la etiqueta HTML  **<iframe>**. La etiqueta <iframe> no está relacionada de alguna manera con la etiqueta <frameset>, sino que puede aparecer en cualquier parte de su documento. La etiqueta <iframe> define una región rectangular dentro del documento en la que el navegador puede mostrar un documento separado, incluidas las barras de desplazamiento y los bordes. Un marco en línea se utiliza para incrustar otro documento dentro del documento HTML actual.

El atributo  **src**  se utiliza para especificar la URL del documento que ocupa el marco en línea:



**Fuentes**

Las fuentes juegan un papel muy importante para hacer que un sitio web sea más fácil de usar y aumentar la legibilidad del contenido. La cara y el color de la fuente depende completamente de la computadora y el navegador que estás utilizando para ver tu página, pero puedes usar la etiqueta HTML  **<font>**  para agregar estilo, tamaño y color al texto de su sitio web. Puedes usar una etiqueta  **<basefont>**  para configurar todo tu texto en el mismo tamaño, cara y color.

La etiqueta de **font** tiene tres atributos llamados **size, color** y **face**para personalizar tus fuentes. Para cambiar cualquiera de los atributos de fuente en cualquier momento dentro de tu página web, simplemente usa la etiqueta <font>. El texto que sigue permanecerá cambiado hasta que cierres con la etiqueta </font>. Puedes cambiar uno o todos los atributos de fuente dentro de una etiqueta <font>.

**Nota:** La  *fuente* y *BASEFONT* son etiquetas que están en desuso y se supone que será eliminado en una futura versión de HTML. Por lo tanto, no deben usarse, en su lugar se sugiere usar estilos CSS para manipular tus fuentes. Pero aún con el propósito de aprender, este capítulo explicará en detalle las etiquetas de fuente y fuente base.

Establecer tamaño de fuente

Puedes establecer el tamaño de la fuente del contenido mediante el atributo de **size**. El rango de valores aceptados es de 1 (menor) a 7 (mayor). El tamaño predeterminado de una fuente es 3:



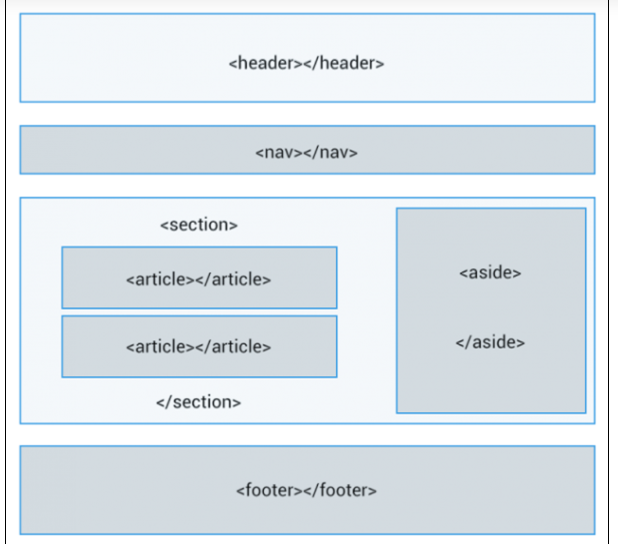
**HTML5 lo nuevo**

Hasta ahora hemos leído y hemos visto ejemplos sobre HTML básico, pero, en la actualidad existe en plena vigencia la versión HTML 5.

Es decir, HTML5 es la versión más reciente de HTML homologada por la W3C.

Una de las principales ventajas de HTML5 es la inclusión de [elementos semánticos](https://www.eniun.com/html5-estructura-basica-elementos-semanticos/), o marcadores semánticos, que nos ayudan a definir las distintas divisiones de una página web. Es decir, los elementos semánticos de HTML5, nos ayudan a identificar cada sección del documento y organizar el cuerpo de una página web de una manera eficiente y estandarizada.

En la figura siguiente, observamos el esquema básico de los elementos semánticos introducidos por la última y por la versión actual de HTML5:



**Según el W3C una Web Semántica:**

"*Permite que los datos sean compartidos y reutilizados en todas las aplicaciones, las empresas y las comunidades*."

**¿Por qué utilizar elementos semánticos?**

En HTML4 los desarrolladores usaban sus propios nombres como identificación de los elementos, tales como: cabecera, superior, inferior, pie de página, menú, navegación, principal, contenedor, contenido, artículo, barra lateral, etc. , se hacía imposible que los motores de búsqueda identificaran el contenido correcto de la página web.

**Elementos semánticos = elementos con significado.**

Un [elemento semántico](https://www.w3schools.com/html/html5_semantic_elements.asp) describe claramente su significado tanto para el navegador web como para el desarrollador.

* Ejemplos de elementos no semánticos: <div> y <span> - No dice nada sobre su contenido.
* Ejemplos de elementos semánticos: <form>, <table> y <article> - Definen claramente su contenido.

**Descripción de elementos semánticos HTML5**

**Elemento HTML <section>**

El  [elemento](https://www.w3schools.com/html/html5_semantic_elements.asp) <section> define una sección en un documento. Según la documentación HTML del W3C: " *Una sección es una agrupación temática de contenido, normalmente con un encabezado* ".

Una página web normalmente se puede dividir en secciones para la introducción, el contenido y la información de contacto.

**Elemento HTML <article>**

El elemento <article> especifica contenido autónomo e independiente. Un artículo debe tener sentido por sí solo y debe ser posible distribuirlo independientemente del resto del sitio web.

Ejemplos de dónde se puede utilizar un elemento <article>:

* Publicación en el foro
* Entrada en el blog
* Artículo de periódico

**Elemento HTML <header>**

El elemento <header> sirve para especificar contenido de tipo introductorio o un conjunto de enlaces de navegación.

Un elemento <header> normalmente contiene:

1. Uno o más elementos de encabezado (<h1> - <h6>).
2. Logo o icono.
3. Información de autoría.

**Nota:** puede tener varios elementos <header> en un documento HTML. Sin embargo, el elemento <header> no puede ser colocado dentro de elementos <footer>, <address> o dentro de otro elemento <header>.

**Elemento HTML <footer>**

El elemento <footer> define un pie de página para un documento o sección.

Un elemento <footer> normalmente contiene:

1. Información de autoría.
2. Información registrada.
3. Información de contacto.
4. Mapa del sitio.
5. Volver a los enlaces superiores.
6. Documentos relacionados.

Puede tener varios elementos <footer> en un documento.

**Elemento HTML <nav>**

El elemento HTML [<nav>](https://developer.mozilla.org/es/docs/Web/HTML/Element/nav) representa una sección de una página cuyo propósito es proporcionar enlaces de navegación, ya sea dentro del documento actual o en otros documentos. Ejemplos comunes de secciones de navegación son: menús, tablas de contenido e índices.

Debemos tener en cuenta que NO todos los enlaces de un documento deben estar dentro de un elemento <nav>. El elemento <nav> está destinado sólo para el bloque principal de enlaces de navegación.

Los navegadores, como los lectores de pantalla para usuarios con capacidades diferentes, pueden utilizar este elemento para determinar si se debe omitir la representación inicial de este contenido.

**Elemento HTML <aside>**

El elemento HTML [<aside>](https://developer.mozilla.org/es/docs/Web/HTML/Element/aside) representa una sección de una página que consiste en contenido que está indirectamente relacionado con el contenido principal del documento. Estas secciones son a menudo representadas como barras laterales o como inserciones y contienen una explicación al margen como una definición de glosario, elementos relacionados indirectamente, como publicidad, la biografía del autor, o en aplicaciones web, la información de perfil o enlaces a blogs relacionados .

**Colores**

Los colores son muy importantes para darle una buena apariencia a tu sitio web. Puedes especificar colores a nivel de página usando la etiqueta <body> o puedes establecer colores para etiquetas individuales usando el atributo **bgcolor**.

La etiqueta <body> tiene los siguientes atributos que se pueden usar para establecer diferentes colores:

* **bgcolor**: establece un color para el fondo de la página.
* **text**: establece un color para el cuerpo del texto.
* **alink**: establece un color para los enlaces activos o los enlaces seleccionados.
* **link**: establece un color para el texto vinculado.
* **vlink**: establece un color para los enlaces visitados, es decir, para el texto vinculado en el que ya ha hecho clic.

Métodos de codificación de colores HTML

Existen tres métodos diferentes para configurar los colores en tu página web:

* **Nombres de colores**: puede especificar nombres de colores directamente como **green**, **blue** o**red.**
* **Códigos hexadecimales**: un código de seis dígitos que representa la cantidad de rojo , verde y azul que compone el color.
* **Valores decimales o porcentuales de color**: este valor se especifica mediante la propiedad rgb ().

Ahora veremos estos esquemas de coloración uno por uno.

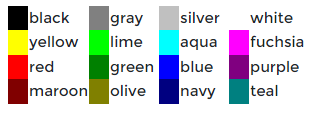
Colores HTML - Nombres de colores

Puedes especificar un nombre de color directo para establecer el texto o el color de fondo. W3C ha enumerado 16 nombres de colores básicos que se validarán con un validador HTML, pero hay más de 200 nombres de colores diferentes compatibles con los principales navegadores.

**Nota**: te dejamos para que consultes una lista completa de nombres de [colores HTML.](https://www.tutorialspoint.com/html/html_color_names.htm)

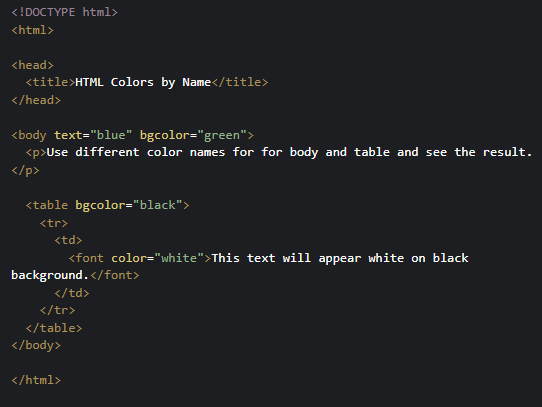
Estándar W3C 16 colores

Aquí está la lista de nombres de 16 colores estándar del W3C y se recomienda usarlos.



Ejemplo

Estos son los ejemplos para establecer el fondo de una etiqueta HTML por nombre de color:



**Introducción CSS**

El gran impulso de los lenguajes de hojas de estilos se produjo con el boom de Internet y el crecimiento exponencial del lenguaje HTML. Entre la fuerte competencia de navegadores web y la falta de un estándar para la definición de los estilos, se dificultaba la creación de documentos con la misma apariencia en diferentes navegadores.

A mediados de la década de 1990 el W3C, encargado de crear todos los estándares relacionados con la web, propuso la creación de un lenguaje de hojas de estilos específicos para el lenguaje HTML y se presentaron propuestas. Las dos propuestas que se tuvieron en cuenta fueron la CHSS (Cascading HTML Style Sheets) y la SSP (Stream-based Style Sheet Proposal). La propuesta CHSS fue realizada por [Håkon Wium Lie](https://en.wikipedia.org/wiki/H%C3%A5kon_Wium_Lie" \t "_blank) y SSP fue propuesto por [Bert Bos](https://es.wikipedia.org/wiki/Bert_Bos) . Entre finales de 1994 y 1995 Lie y Bos se unieron para definir un nuevo lenguaje que tomaba lo mejor de cada propuesta y lo llamaron [CSS](https://developer.mozilla.org/es/docs/Web/CSS) , Cascading Style Sheets.

En 1995, el W3C decidió gestionar el desarrollo y estandarización de CSS y agregó el proyecto a su grupo de trabajo de HTML. A finales de 1996, el W3C publicó la primera recomendación oficial, conocida como " [CSS nivel 1](https://www.ecured.cu/CSS) ".

**¿Qué es CSS?.**  
Css es un leguaje que trabaja junto con HTML para proporcionar estilos visuales a los elementos de un documento web.

**Características:**

* .- Ahorra trabajo. Se puede controlar el diseño de varias páginas HTML a la vez.
* .- Se pueden almacenar en archivos \* .css

CSS 3 es la última versión estándar.

¿Para qué utilizar CSS?

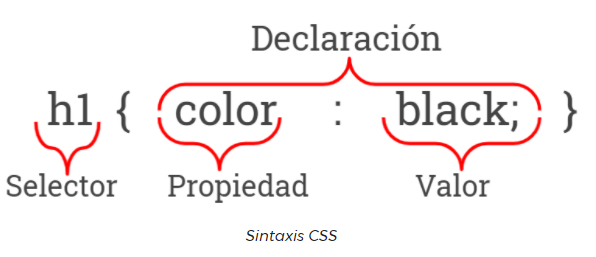
Para definir estilos en los documentos web, incluyendo el diseño, la disposición de los elementos y para responder a las variaciones en la pantalla en cuanto a diferentes dispositivos y tamaños de pantalla.

Ventajas

1. Control centralizado de la presentación de un sitio web completo con lo que se agiliza considerablemente la actualización y mantenimiento.
2. Los Navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad. Por ejemplo, las personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.
3. Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestra o incluso una elección del usuario. Por ejemplo, para ser impresa, mostrada en un dispositivo móvil, o ser "leída" por un sintetizador de voz.
4. El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño (siempre y cuando no se utiliza estilo en línea).

**Reglas CSS**

CSS define un conjunto de reglas que permiten describir cada una de las partes que componen los estilos CSS.



**Selector:** indica el elemento o elementos HTML a los que se aplica la regla CSS.

**Declaración:** especifica los estilos que se aplican a los elementos.

**Propiedad:** permite modificar el aspecto de una característica del elemento.

**Valor:** indica el nuevo valor de la característica modificada en el elemento.

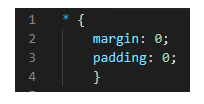
Selectores

Nos indican qué elemento HTML hay que aplicar el estilo. Una misma regla puede aplicar a varios selectores y, a un mismo selector se le pueden aplicar varias reglas.

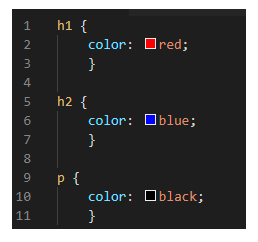
Para mayor información referirse a [este enlace](https://www.w3.org/wiki/CSS_/_Selectores_CSS#Lista_de_selectores_css) .

Selectores básicos

**Selector universal** : se utiliza para seleccionar todos los elementos de la página:



**Selector de tipo o etiqueta:** selecciona todos los elementos de la página cuya etiqueta HTML coincide con el valor del selector:

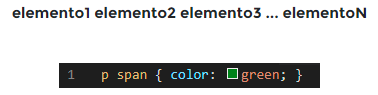


Selector descendente

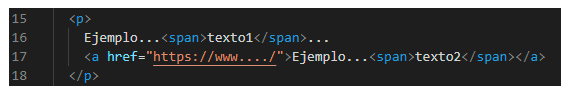
Selecciona los elementos que se encuentran dentro de otros elementos. Un elemento es descendente de otro cuando se encuentra entre las etiquetas de apertura y de cierre de otro elemento.



La sintaxis formal del selector descendente se muestra a continuación:



Si el código HTML de la página es el siguiente:



El selector p span selecciona tanto texto1 como texto2. El motivo es que en el selector descendente, un elemento no tiene que ser descendiente directo del otro. La única condición es que un elemento debe estar dentro de otro elemento, sin importar el nivel de profundidad en el que se encuentre.

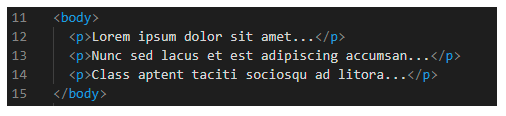
No se les aplica la regla CSS anterior si el resto de los elementos <span> de la página no están dentro de un elemento <p>.

Los selectores descendentes nos permiten aumentar la precisión del selector de tipo o etiqueta. Así, utilizando el selector descendente podemos aplicar diferentes estilos a los elementos del mismo tipo.

Para conocer más te invitamos hacer clic en el este [enlace](https://uniwebsidad.com/libros/css/capitulo-2/selectores-basicos).

Selector de clase o selector Class

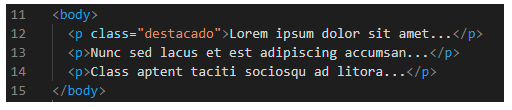
Si consideramos el siguiente código HTML de ejemplo:



**¿Cómo podemos aplicar estilos CSS sólo al primer párrafo?**

El selector universal (\*) no se puede utilizar porque seleccionaría todos los elementos de la página. El selector de tipo o etiqueta (p) tampoco se puede utilizar porque seleccionaría todos los párrafos. Por último, el selector descendente (body p) tampoco se puede utilizar porque todos los párrafos se encuentran en el mismo sitio.

Una de las soluciones más sencillas para aplicar estilos a un solo elemento de la página consiste en utilizar el atributo class de HTML sobre ese elemento para indicar directamente la regla CSS que se le debe aplicar:

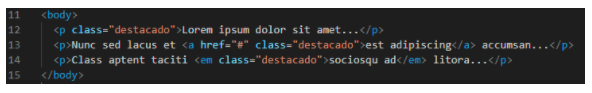


A continuación, creamos en el archivo CSS una nueva regla llamada destacado con todos los estilos que vamos a aplicar al elemento. Para que el navegador no confunda este selector con los otros tipos de selectores, prefijamos el valor del atributo class con un punto (.) Tal como vemos en la siguiente línea:



El selector destacado se interpreta como "cualquier elemento de la página cuyo atributo class sea igual a destacado", por lo que solamente el primer párrafo del anterior ejemplo cumple esa condición.

Este tipo de selectores de clase son los más utilizados junto con los selectores de ID que veremos a continuación. La principal característica de este selector es que en una misma página HTML varios elementos diferentes pueden utilizar el mismo valor en el atributo class:

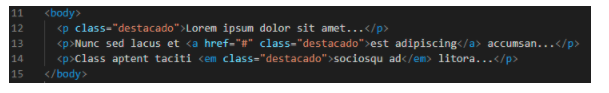


Los selectores de clase resultan imprescindibles para diseñar páginas web complejas, ya que nos permiten disponer de una precisión total al seleccionar los elementos. Además, estos selectores nos permiten reutilizar los mismos estilos para varios elementos diferentes:



El elemento <span> tiene un atributo class = "error", por lo que se le aplica las reglas CSS indicadas por el selector .error. Por su parte, el elemento <div> tiene un atributo class = "aviso", por lo que su estilo es el que define las reglas CSS del selector .aviso.

En ciertos casos, es necesario restringir el alcance del selector de clase. Si utilizamos el ejemplo anterior:



**¿Cómo podemos aplicar estilos solamente al párrafo cuyo atributo class sea igual a destacado?**

Combinando el selector de tipo y el selector de clase, se obtiene un selector mucho más específico:



Interpretamos al selector p.destacado como "aquellos elementos de tipo <p> que dispongan de un atributo class con valor destacado". De la misma forma, el selector a.destacado solamente seleccionaría los enlaces cuyo atributo class sea igual a destacado.

De lo anterior deducimos que el atributo .destacado es equivalente a \* .destacado, por lo que todos los diseñadores obvian el símbolo \* al escribir un selector de clase normal.

No debemos confundir el selector de clase con los selectores anteriores:

/ \* Todos los elementos de tipo "p" con atributo class = "aviso" \* /

p.aviso {...}

/ \* Todos los elementos con atributo class = "aviso" que estén dentro

   de cualquier elemento de tipo "p" \* /

p .aviso {...} / \* notar el espacio entre py la clase \* /

/ \* Todos los elementos "p" de la página y todos los elementos con

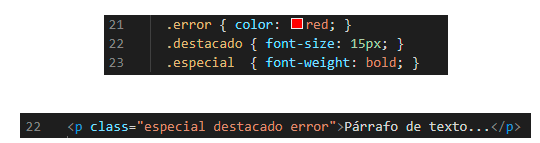
   atributo class = "aviso" de la página \* /

p, .aviso {...} / \* notar la, (coma) entre py la clase \* /

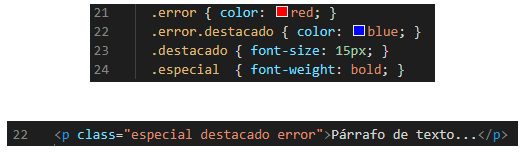
Para completar, es posible aplicar los estilos de varias clases CSS sobre un mismo elemento. La sintaxis es similar, pero los diferentes valores del atributo class se separan con espacios en blanco.



Al párrafo anterior le aplicamos los estilos definidos en las reglas .especial, .destacado y .error, por lo que en el siguiente ejemplo, el texto del párrafo se vería de color rojo, en negrita y con un tamaño de letra de 15 píxel:



Si un elemento dispone de un atributo class con más de un valor, es posible utilizar un selector más avanzado:



En el ejemplo anterior, el color de la letra del texto es azul y no rojo. Esto se debe a que hemos utilizado un selector de clase múltiple .error.destacado, que se interpreta como "aquellos elementos de la página que están disponibles de un atributo class con al menos los valores error y destacado".

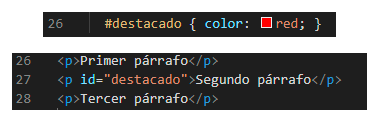
Para saber más podes hacer clic en [este enlace](https://uniwebsidad.com/libros/css/capitulo-2/selectores-basicos).

**Selectores de ID**

En ocasiones, es necesario aplicar estilos CSS a un único elemento de la página. Aunque puede aplicar un selector de clase para un único elemento, existe otro selector más eficiente en este caso.

El selector de ID permite seleccionar un elemento de la página a través del valor de su atributo id. Este tipo de selectores sólo selecciona un elemento de la página porque el valor del atributo id no se puede repetir en dos elementos diferentes de una misma página.

La sintaxis de los selectores de ID es muy parecida a la de los selectores de clase, salvo que se utiliza el símbolo de la almohadilla (#) en vez del punto (.) Como prefijo del nombre de la regla CSS:



En el ejemplo anterior, el selector #destacado solamente selecciona el segundo párrafo (cuyo atributo id es igual a destacado).

La principal diferencia entre este tipo de selector y el selector de clase tiene que ver con HTML y no con CSS. En una misma página, el valor del atributo id debe ser único, de forma que dos elementos diferentes no pueden tener el mismo valor de id. Sin embargo, el atributo class no es obligatorio que sea único, de forma que muchos elementos HTML diferentes pueden compartir el mismo valor para su atributo class.

De esta forma, la recomendación general es la de utilizar el selector de ID cuando se quiere aplicar un estilo a un solo elemento específico de la página y utilizar el selector de clase cuando queremos aplicar un estilo determinado a varios elementos diferentes de la página HTML.

Al igual que los selectores de clase, en este caso también podemos restringir el alcance del selector mediante la combinación con otros selectores. El siguiente ejemplo aplica la regla CSS solamente al elemento de tipo <p> que tenga un atributo id igual al indicado:



A primera vista, restringir el alcance de un selector de ID puede parecer absurdo. En realidad, un selector de tipo p # aviso sólo tiene sentido cuando el archivo CSS se aplica sobre muchas páginas HTML diferentes.

En este caso, algunas páginas pueden disponer de elementos con un atributo id igual a aviso y que no sean párrafos, por lo que la regla anterior no se aplica sobre esos elementos.

No debe confundirse el selector de ID con los selectores anteriores:

/ \* Todos los elementos de tipo "p" con atributo id = "aviso" \* /

p # aviso {...}

/ \* Todos los elementos con atributo id = "aviso" que estén dentro

de cualquier elemento de tipo "p" \* /

p #aviso {...}

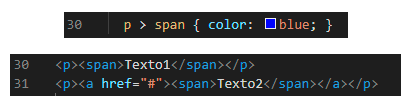
/ \* Todos los elementos "p" de la página y todos los elementos con

atributo id = "aviso" de la página \* /

p, #aviso {...}

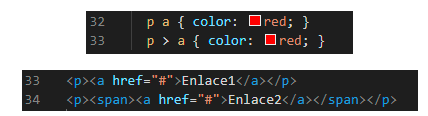
**Selectores avanzados**

Se trata de un selector similar al selector descendente, pero muy diferente en su funcionamiento. Se utiliza para seleccionar un elemento que es hijo directo de otro elemento y se indica mediante el "signo de mayor que" (>):



En el ejemplo anterior, el selector p> span se interpreta como "cualquier elemento <span> que sea hijo directo de un elemento <p>", por lo que el primer elemento <span> cumple la condición del selector. Sin embargo, el segundo elemento <span> no la cumple porque es descendiente pero no es hijo directo de un elemento <p>.

El siguiente ejemplo muestra las diferencias entre el selector descendente y el selector de hijos:

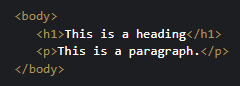


El primer selector es de tipo descendente y por tanto se aplica a todos los elementos <a> que se encuentran dentro de los elementos <p>. En este caso, los estilos de este selector aplican se a los dos enlaces.

Por otra parte, el selector de hijos obliga a que el elemento <a> sea hijo directo de un elemento <p>. Por lo tanto, los estilos del selector p> a no se aplican al segundo enlace del ejemplo anterior.

Para conocer más sobre selectores avanzados podes hacer clic en [este enlace](https://uniwebsidad.com/libros/css/capitulo-2/selectores-avanzados).

**Ejemplo Selectores**

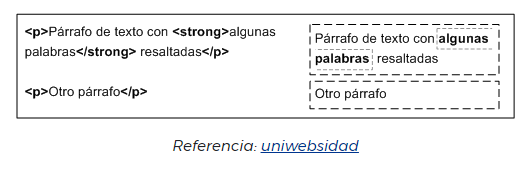


**Modelo de Cajas**

Tal como dijimos en la unidad de HTML, si hacemos una analogía con una estructura de cajas de cartón, podemos decir que hay ciertas cajas que van dentro de otras y ciertas cajas que van una al lado de otra.

El "modelo de caja" es probablemente la característica más importante del lenguaje de hojas de estilos CSS, ya que condiciona el diseño de todas las páginas web. El modelo de cajas es el comportamiento de CSS que hace que todos los elementos de las páginas sean representados mediante cajas rectangulares.

Cada vez que se inserta un elemento HTML, se crea una nueva caja rectangular que encierra los contenidos de ese elemento. La siguiente imagen muestra tres cajas rectangulares que crean los tres elementos HTML de una porción de página de ejemplo.



Los navegadores web crean y colocan las cajas de forma automática, pero CSS permite modificar todas sus características. Cada una de las cajas está formada por seis partes que se describen a continuación:

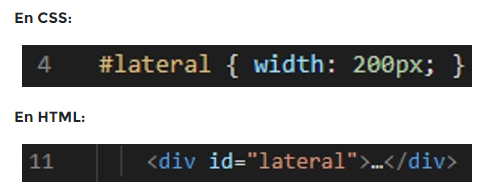
1. **Content** (contenido): se trata del contenido HTML del elemento (las palabras de un párrafo, una imagen, el texto de una lista de elementos, etc.)
2. **Padding** (relleno): espacio libre opcional existente entre el contenido y el borde.
3. **Border** (borde): línea que encierra completamente el contenido y su relleno.
4. **Background-image** (Imagen de fondo): imagen que se muestra por detrás del contenido y el espacio de relleno.
5. **Background-color** (color de fondo): color que se muestra por detrás del contenido y el espacio de relleno.
6. **Margin** (margen): separación opcional existente entre la caja y el resto de cajas adyacentes.

**Dimensiones de las cajas**

Anchura

Para los elementos de bloque y las imágenes, la propiedad width (anchura) permite establecer la anchura directamente mediante una medida.

Anchura = width



Si se utilizan unidades de medida, los valores indicados no pueden ser negativos. Si en vez de una unidad de medida se indica un porcentaje, la referencia de ese valor es la anchura del elemento que lo contiene. El valor heredado indica que la anchura del elemento se hereda de su elemento padre.

Si se establece la anchura de un elemento con la unidad de medida em, el valor indicado toma como referencia el tamaño de letra del propio elemento.

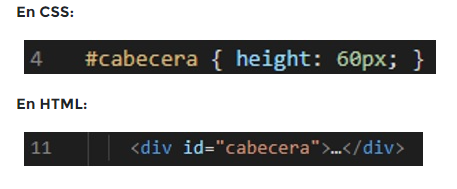
El valor auto es el valor por defecto de la anchura de todos los elementos. En este caso, el navegador determina la anchura de cada elemento teniendo en cuenta, entre otros, el tipo de elemento que se trata (de bloque o en línea), el sitio disponible en la pantalla del navegador y los contenidos de los elementos.

**Altura**

Al igual que sucede con ancho, la propiedad altura no admite valores negativos. Si se indica un porcentaje, se toma como referencia la altura del elemento padre. Si el elemento padre no tiene una altura definida explícitamente, se asigna el valor auto a la altura.

El valor heredado indica que la altura del elemento se hereda de su elemento padre. El valor automático, que es el que se utiliza si no se establece altura de forma explícita un valor a esta propiedad, indica que el navegador debe calcular automáticamente el elemento, teniendo en cuenta sus contenidos y el sitio disponible en la página.

Altura = height

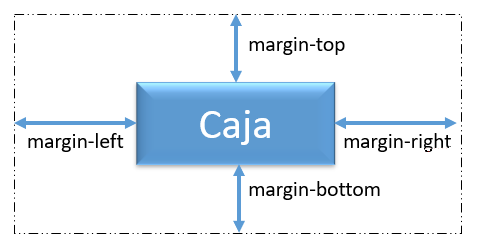


**Margen**

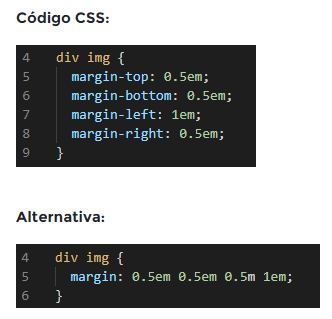
CSS define cuatro propiedades para controlar cada uno de los márgenes horizontales y verticales de un elemento.

1. margin-top => Margen superior
2. margin-right => Margen derecho
3. margin-bottom => Margen inferior
4. margin-left => Margen izquierdo

Cada una de las propiedades establece la separación entre el borde lateral de la caja y el resto de cajas adyacentes:



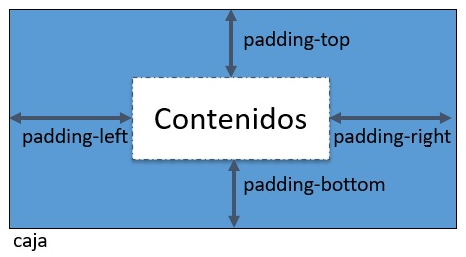
Los márgenes verticales (margin-top y margin-bottom) sólo se pueden aplicar a los elementos de bloque y las imágenes, mientras que los márgenes laterales (margin-left y margin-right) se pueden aplicar a cualquier elemento.



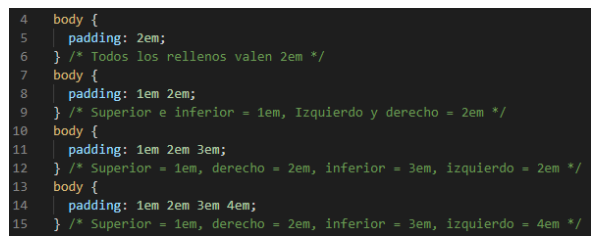
**Relleno o acolchado**

Relleno o acolchado. CSS define cuatro propiedades para controlar cada uno de los espacios de relleno horizontales y verticales de un elemento:

1. padding-top => Relleno superior
2. padding-right => Relleno derecho
3. padding-bottom => Relleno inferior
4. padding-left => Relleno izquierdo



**Algunos ejemplos de acolchado:**

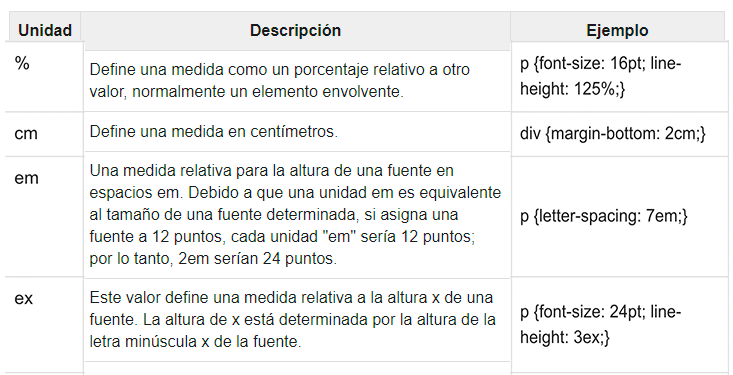


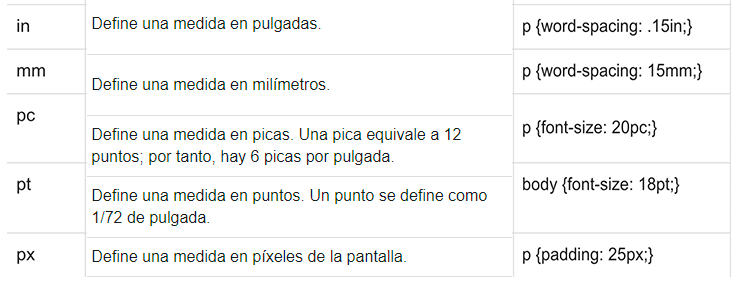
**Unidades de medida**

Antes de comenzar el ejercicio real, nos gustaría dar una breve idea sobre las unidades de medida CSS.

CSS admite una serie de medidas que incluyen unidades absolutas como pulgadas, centímetros, puntos, etc., así como medidas relativas como porcentajes y unidades em. Necesita estos valores al especificar varias medidas en sus reglas de estilo, por ejemplo,  **border = "1px solid red"**  .

Hemos enumerado todas las unidades de medida CSS junto con los ejemplos adecuados:

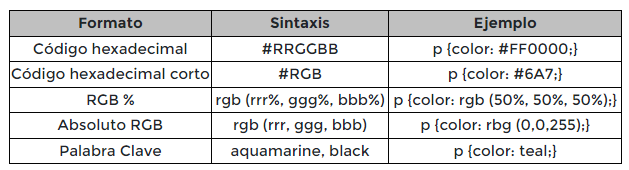




**Colores**

CSS usa valores de color para especificar un color. Por lo general, se utilizan para establecer un color para el primer plano de un elemento (es decir, su texto) o para el fondo del elemento. También se puede usar para afectar el color de los bordes y otros efectos decorativos.

Puede especificar sus valores de color en varios formatos. La siguiente tabla enumera todos los formatos posibles:



**Fuentes**

Veamos cómo configurar las fuentes de un contenido, disponible en un elemento HTML. Puedes establecer las siguientes propiedades de fuente de un elemento:

* La propiedad **font-family** se utiliza para cambiar la cara de una fuente.
* La propiedad de **font-style** se usa para hacer una fuente en cursiva u oblicua.
* La propiedad **font-variant** se utiliza para crear un efecto de versalitas.
* La propiedad de **font-weight** se utiliza para aumentar o disminuir la negrita o la luz de una fuente.
* La propiedad de **font-size** se utiliza para aumentar o disminuir el tamaño de una fuente.
* La propiedad de **font** se utiliza como forma abreviada para especificar otras propiedades de fuente.

Establecer la familia de fuentes

A continuación se muestra el ejemplo, que demuestra cómo configurar la familia de fuentes de un elemento. El valor posible podría ser cualquier nombre de familia de fuentes:

