**¿Cómo funciona la web?**

La internet o el mundo web en general está representado como un conjunto de computadoras interconectadas entre sí que se relacionan o pueden relacionarse mediante diferentes protocolos sin importar la distancia a la cual se encuentren unas de otras.

Cada una de estas computadoras pueden estar ofreciendo un servicio o información para ser consumido (un servidor) como así también pueden ser consumidores de dicho servicio o información (clientes), de esta manera, por ejemplo, si se tiene una página o aplicación web en un servidor, ésta se está ofreciendo como un servicio, donde cada una de las personas que accedan a la misma serán consideradas clientes.

**Direccionamiento Web**

Todo recurso en internet posee una dirección que lo representa y que permite que determinados clientes puedan acceder a él. Esta dirección es comúnmente conocida como **URL (Uniform Resource Locator).**

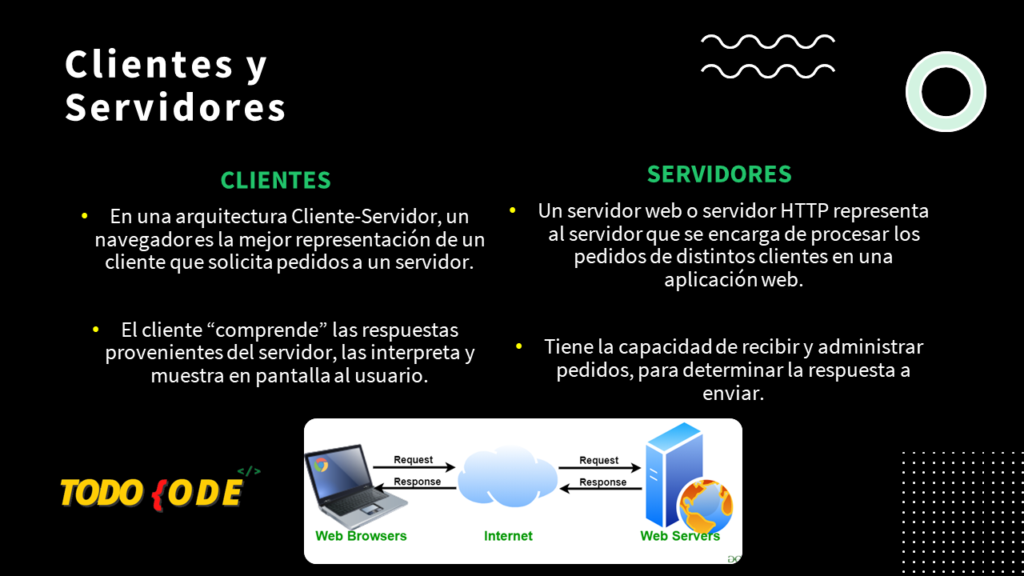
Toda **URL** de una aplicación, página o recurso web está compuesta por una serie de partes, las mismas te las presentamos a continuación:



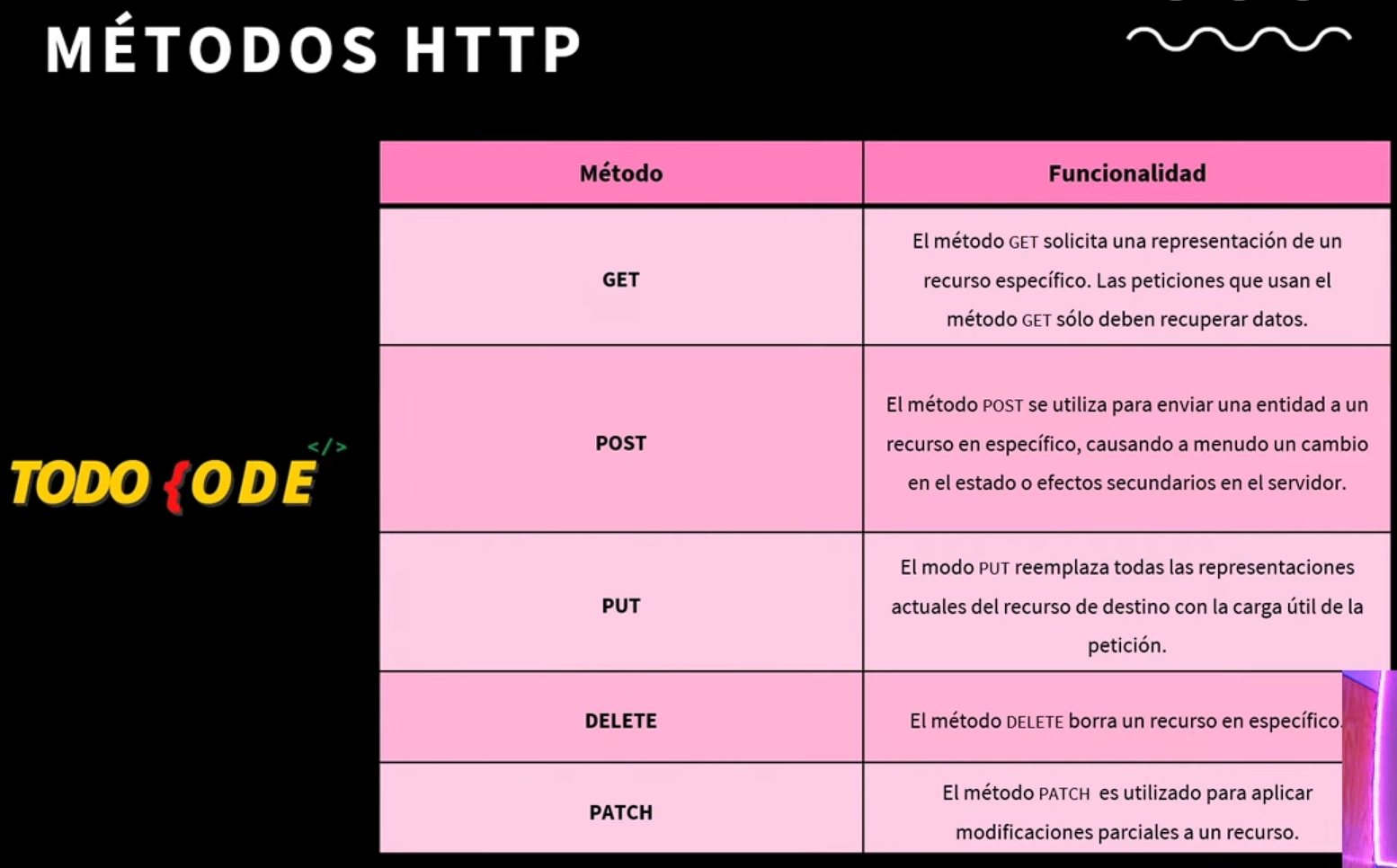
**Arquitectura Cliente-Servidor**

La arquitectura cliente-servidor establece que existe una serie de clientes que puede conectarse a servidores para consumir un determinado servicio. Este servicio puede ser de impresión (por ejemplo en una red local donde hay varias computadoras conectadas a una misma impresora y existe un servidor que la administra), de correo electrónico, de base de datos, de aplicaciones, entre muchos otros.

La principal característica de éste modelo es el hecho de que un mismo servicio puede estar siendo consumido al mismo tiempo por sin fin de clientes, siempre y cuando el servidor encargado de recibir las peticiones tenga la capacidad suficiente para tratarlas, por ejemplo, no es lo mismo un servidor que podamos armar de forma casera con una computadora que tenemos guardada hace varios años, que un servidor de Google o de Amazon que tienen todas las prestaciones técnicas de última generación.







**HTML**

El documento que lee el navegador está escrito en un lenguaje de marcado llamado HTML, que son las siglas de **HyperText Markup Language (Lenguaje de marcas de hipertexto**), o lo que es lo mismo, un lenguaje de etiquetas que permite incluir o hacer referencia a todo tipo de información.

* HTML 5 es un lenguaje maquetado o de etiquetas.
* NO ES un lenguaje de programación
* Se utiliza para crear mayormente la creación de pagina web o fronted de aplicaciones web.
* Los navegadores se comunican mediante lenguaje HTML para mostrar el contenido de una página web.
* Se usa en conjunto con CSS, el cual permite agregar colores, estilos, entre otras cuestiones que dejan a nuestra maqueta HTML con aspecto de diseño.

**SINTAXIS BASICA**

Doctype: Indica al navegador que versión de HTML estamos usando. Va al principio de un archivo HTML.

Elementos: Nos permite estructurar y dar significado a cada una de las partes de un documento HTML. Se representa en forma de etiqueta.

Las etiquetas delimitan el principio y fin de un elemento.

<!DOCTYPE html>

<html lang = “es”>

<head>

<title>Document</title>

</head>

<body>

</body>

</html>

**Categorías de etiquetas**

En HTML no se puede colocar cualquier palabra como etiqueta, sino que existen una colección de etiquetas, cada una con una misión y objetivo diferente. Esto es una lista de categorías, donde en cada una hay varias etiquetas:

1️⃣ Textuales: Partes, fragmentos o tipos de textos.

2️⃣ Agrupación: Agrupa partes o información de cierta naturaleza.

3️⃣ Multimedia: Imágenes, video, sonido y relacionados.

4️⃣ Tablas: Mostrar datos e información tabular relacionada.

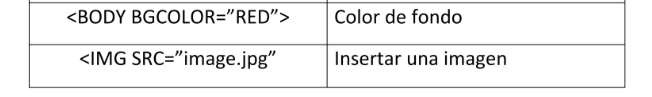
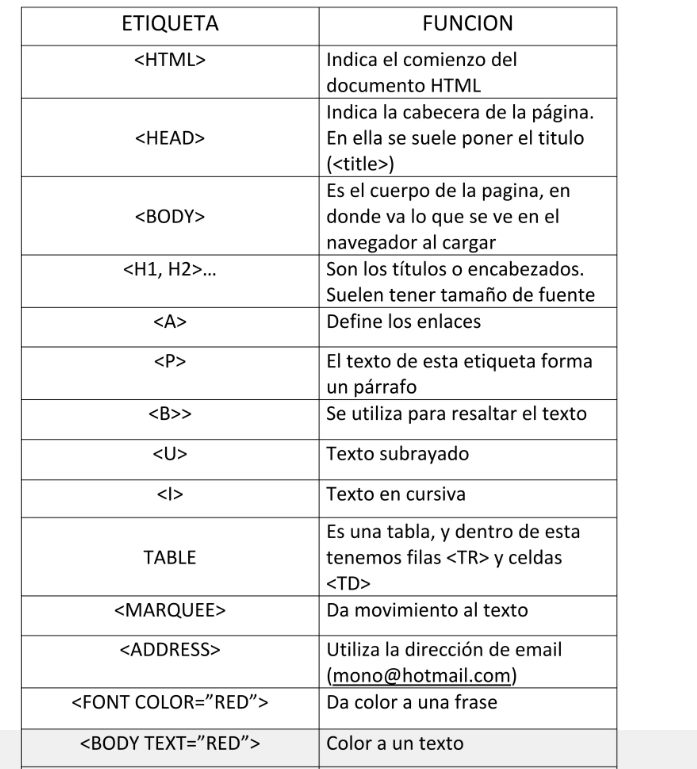
5️⃣ Formularios: Permite al usuario insertar información o datos.

6️⃣ Interactivas: Elementos con los que el usuario puede interactuar.

7️⃣ Semánticas: Variaciones semánticas del <div> y relacionados.

8️⃣ Scripting: Etiquetas relacionadas con Javascript.

Etiquetas



<br> “Espacios en blanco”

**HTML SEMANTICO**

<header>

**Cuándo usarlo:** Cuando necesitas un encabezado introductorio de una página o de una sección.

Contiene normalmente:

* Logo
* Título (h1–h6)
* Menú de navegación (a veces)

**Puede haber varios:** Sí. Uno para la página y otros dentro de section o article.

**No debe contener:** El contenido principal completo.

<nav>

**Cuándo usarlo:** Cuando el contenido es navegación, es decir, un conjunto de enlaces para moverse por el sitio o por una sección.

Ejemplos comunes:

* Menú principal
* Menú lateral
* Paginación

**No uses nav:** Para enlaces sueltos dentro de un texto.

<main>

**Cuándo usarlo:** Para envolver el contenido principal y único de la página.

Reglas importantes:

* Solo uno por documento
* No puede estar dentro de header, footer, nav o aside

**Piensa en main como:** “Lo más importante que el usuario vino a ver”.

<section>

**Cuándo usarlo:** Cuando necesitas agrupar contenido relacionado bajo un mismo tema.

**Debe tener sentido temático:** Normalmente incluye un título (h2, h3, etc.).

Ejemplo típico:

* Sección “Servicios”
* Sección “Contacto”
* Sección “Sobre nosotros”

No la uses solo para estilo.

<article>

**Cuándo usarlo:** Cuando el contenido es independiente y reutilizable.

**Buena regla:** Si puedes mover ese bloque a otro sitio y sigue teniendo sentido, es un article.

Ejemplos:

* Post de blog
* Noticia
* Comentario
* Tarjeta de producto

<aside>

**Cuándo usarlo:** Para contenido relacionado, pero no principal.

Ejemplos comunes:

* Barra lateral
* Publicidad
* Enlaces relacionados
* Información adicional

Puede estar:

* Dentro de main
* Dentro de article

<footer>

**Cuándo usarlo:** Para información final o complementaria de una página o sección.

Contiene normalmente:

* Autor
* Copyright
* Enlaces legales
* Información de contacto

**Puede haber varios:** Sí, uno por página y otros dentro de article o section.

**Enlaces en HTML**

Los enlaces (también conocidos como hipervínculos) son una de las características fundamentales de la Web. Los enlaces permiten a los usuarios navegar entre diferentes páginas y sitios web con solo hacer clic en una palabra o frase específica.

En HTML, los enlaces se crean utilizando la etiqueta <a> (anchor) en conjunto con el atributo «href» se utiliza para especificar la dirección URL del enlace.  
Ejemplo:

<a href="https://www.instagram.com/skrillex/">instagram</a>

Para referenciar un título o subtitulo como una especie de índice se puede usar la siguiente anotación

 <h1>índice</h1>

    <h2> <a href="#Elec">Ir a electronica</a></h2>

    <h2><a href="#Rock">Ir a Rock</a></h2>

Para definirlos se ve así:

<h1 id="EleC">Electronica</h1>

 <h1 id="Rock">Rock</h1>

**Imágenes, video, audio y más en HTML**

**Imágenes**

La etiqueta <img> se utiliza para mostrar imágenes en una página web. Se utiliza en conjunto con el atributo «src» (que significa source) para especificar la ruta o URL de la imagen a mostrar. También se pueden especificar atributos como «alt» para describir la imagen y «width» o «height» para especificar el tamaño. Un ejemplo puede ser este:

<img src="imagen.jpg" alt="descripcion de la imagen" width="200" height="300">

**Videos**

La etiqueta <video> se utiliza para mostrar videos en una página web. Tal como en las imágenes, se utiliza en conjunto con el atributo «src» para especificar la ruta o URL del video a mostrar. Además de esto, posee una serie de atributos «controls» que permiten mostrar controles de reproducción en el video, y el atributo «poster» para especificar una imagen o vista previa/portada antes de reproducir el video. Un ejemplo podemos verlo a continuación;

<video src="video.mp4" controls poster="imagen\_del\_video.jpg"></video>

**Audio**

La etiqueta <audio> se utiliza para mostrar audio en una página web. Al igual que con las imágenes y videos, se utiliza en conjunto con el atributo «src» y el atributo controls. Por ejemplo:

<audio src="audio.mp3" controls></audio>

**Contenido embebido**

La etiqueta <iframe> se utiliza para mostrar contenido embebido de otra página web dentro de la web donde estemos trabajando. Puede utilizar el atributo «src» para establecer la URL de la página a incorporar como así también los atributos «width» y «height» para especificar el tamaño del frame. Por ejemplo:

<iframe src="https://www.ejemplo.com" width="800" height="600"></iframe>

**Formularios en HTML**

Los formularios son uno de los principales puntos de interacción entre un usuario y una página web o aplicación. Su principal función es la de permitir a los usuarios el ingreso de datos y que la aplicación pueda trabajar con los mismos, sea, por ejemplo, para guardar dicha información en una base de datos, utilizarla para una búsqueda o simplemente realizar cambios en la interfaz gráfica a partir de la selección del usuario.

Las etiquetas básicas que pude contener cualquier formulario son:

1. **<form> </form>:** Apertura y cierre de un formulario.

**action:** permite indicar la acción que realizará el formulario cuando se lo quiera enviar.

**method:** indica el método mediante el cual se enviará el formulario (Ej: POST)

1. **<label></label>:** Apertura y cierre para una etiqueta de un elemento dentro del formulario.

**for:** se utiliza para indicar a qué input hará referencia la etiqueta.

1. **<input>:** Se utiliza para permitir cuadros de texto donde el usuario ingrese datos. Complementos:

**type:** permite indicar el tipo de entrada que permitiremos. Ej: email, text, password, etc.

**id:** permite asignar una id (identificación/nombre) que sea única para cada input.

1. **<textarea></textarea>:** Se utiliza para permitir cuadros de texto de gran tamaño de caracteres donde el usuario ingrese datos.

**id:** permite asignar una id (identificación/nombre) que sea única para cada textarea.

**¡Más Detalles en los formularios!**

Los formularios en HTML permiten recopilar información de los usuarios de una manera eficiente y organizada. Para lograr esto, se utilizan diferentes tipos de elementos de formularios, como checkboxes, radio buttons, select y botones.

* **Checkboxes:** Los checkboxes son elementos de selección múltiple que se utilizan para permitir a los usuarios seleccionar varias opciones a la vez. Se representan como un cuadro que se puede marcar o desmarcar. Se crean utilizando la etiqueta <input type=»checkbox»>.
* **Radio Buttons:** Los radio buttons son elementos de selección única que se utilizan para permitir a los usuarios seleccionar una sola opción. Se representan como un círculo que se puede marcar o desmarcar. Se crean utilizando la etiqueta <input type=»radio»>.
* **Select:** El elemento Select es una lista desplegable que permite a los usuarios seleccionar una opción de una lista predefinida. Se crea utilizando la etiqueta <select> y las opciones se definen utilizando la etiqueta <option>.
* **Botones:** Los botones son elementos interactivos que se utilizan para iniciar una acción, como enviar un formulario o realizar una búsqueda. Se crean utilizando la etiqueta <input type=»button»> o <button>.

**Ejercicio corto:**

<!DOCTYPE html>

<html lang="en">

<head>

  <title>Formulario</title>

</head>

<body>

  <h1>Complete los datos requeridos </h1>

  <form action="">

    <p><label for="nombre">Nombre: <input type="text" id = "nombre"></label></p>

    <p><label for="apellido">Apellido: <input type="text" id = "apellido"></label></p>

    <p><label for="edad">Edad: <input type="text" id = "edad"></label></p>

    <p><label for="comentario">Comentario: <textarea name="" id="comentario" ></textarea></label></p>

    <p><label for="estatura">Estatura:

      <select  id="estatura">

        <option value="< 1.50"> <1.50m </option>

        <option value="1.60 a 1.70">1.60m a 1.70m</option>

        <option value="1.71 a 1.80" >1.71m a 1.80m</option>

        <option value=">1.81">>1.81m</option>

      </select>

      </label></p>

    <p><b>Color de ojos</b></p>

    <input type="checkbox" value = "verde" >

    <label>Verde</label> <br>

    <input type="checkbox" value = "marron">

    <label >Marrón</label> <br>

    <input type="checkbox" value = "azul">

    <label >Azules</label> <br>

    <input type="checkbox" value = "negro">

    <label >Negro</label> <br>

  <p><b>Género</b></p>

<input type="radio" name="genero" value="hombre" id="hombre">

<label for="hombre">Hombre</label> <br>

<input type="radio" name="genero" value="Mujer" id="mujer">

<label for="mujer">Mujer</label> <br>

<input type="radio" name="genero" value="otro" id="otro">

<label for="otro">Prefiero no decirlo</label> <br>

    <br>

    <button type = "submit">Enviar</button>

  </form>

</body>

</html>

Esto me sirve para redirigir a otra página web



**Etiquetas para los imputs**

<fieldset>

Agrupa campos relacionados dentro de un formulario.

* Organización
* Accesibilidad
* Lectura del formulario

<legend>

Describe el contenido del fieldset.

Funciona como un título del grupo de campos.

**Tablas en HTMl**

Las tablas en HTML son una herramienta muy útil para presentar datos en un formato organizado y fácil de leer. Están compuestas por filas y columnas y se crean utilizando la etiqueta <table>. Además de ésta, cuenta con otra serie de etiquetas complementarias que nos permiten organizar de diferentes maneras la información dentro de las tablas, entre ellas tenemos:

* **tr:** La etiqueta <tr> significa, en inglés, «table row» (fila de tabla) y representa efectivamente una fila en una tabla HTML.
* **td:** La etiqueta <td> significa «table data cell» y representa, efectivamente, una celda dentro de una fila en una tabla HTML.
* **th:** La etiqueta <th> significa «table header». Su uso es similar a la etiqueta td, pero se utiliza para definir los encabezados de las columnas de una tabla HTML.

Además de las etiquetas **tr, td y th,** hay muchas otras etiquetas y atributos que se pueden utilizar para mejorar la apariencia y la funcionalidad de las tablas en HTML, como por ejemplo agregar bordes, alineación, colores de fondo, etc.

**Listas en HTML: Ordenadas y Desordenadas: “ul, ol, li” y más**

Las listas en HTML son una forma de organizar y presentar información de manera clara y estructurada.

Existen dos principales tipos de listas en HTML: las listas desordenadas **(ul)** y las listas ordenadas **(ol).**

**Listas Desordenadas**

Las listas desordenadas (unordered lists) se representan mediante la etiqueta **<ul>** y se usan para presentar una lista de elementos que no tienen un orden en específico por seguir. Cada elemento de la lista se marca con un símbolo de disc (•) o mejor conocido como «viñeta» antes de su texto.

Un ejemplo en código HTML podemos ver a continuación, donde en el lado izquierdo vemos el código y del lado derecho el resultado obtenido.

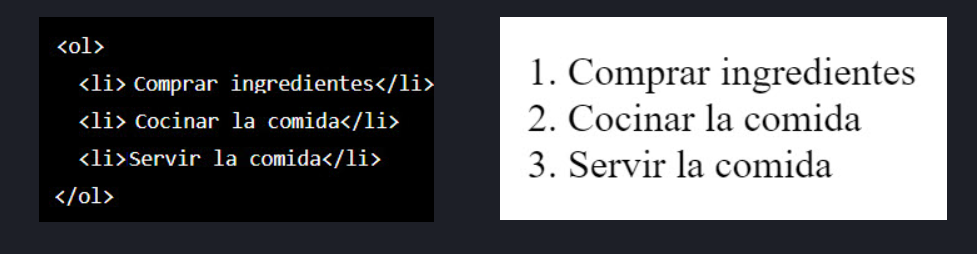


**Etiqueta <li>**

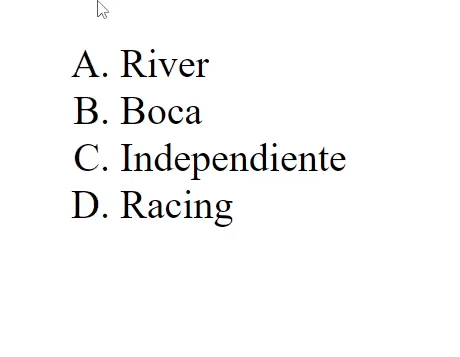
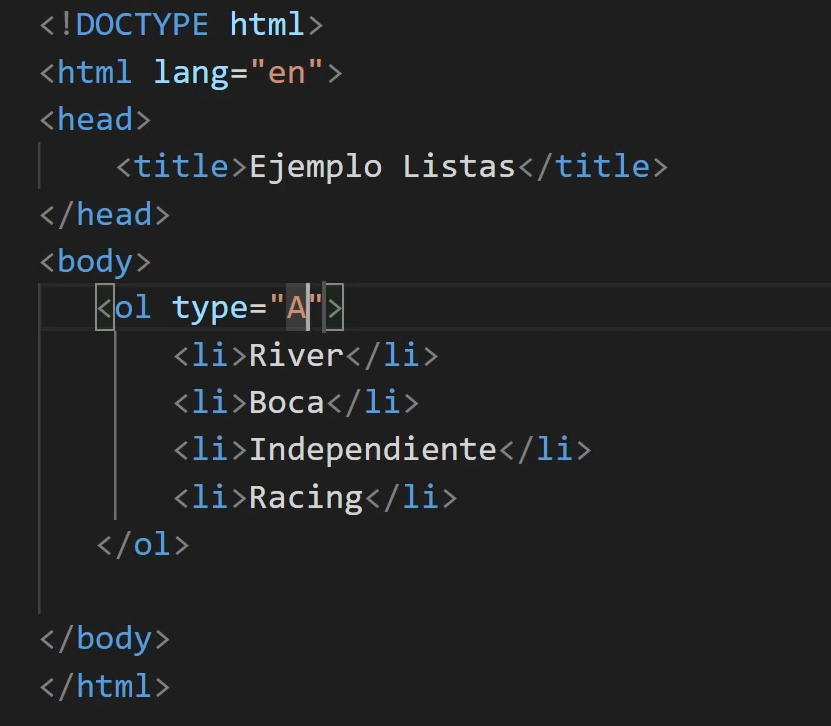
Además de la etiqueta **<ul>** de la lista desordenada, vemos que hay una más, que es **<li>.** Esta etiqueta significa «List item» (item de la lista) y se puede utilizar, tanto en listas ordenadas como desordenadas para citar a cada uno de sus elementos.

**Listas Ordenadas**

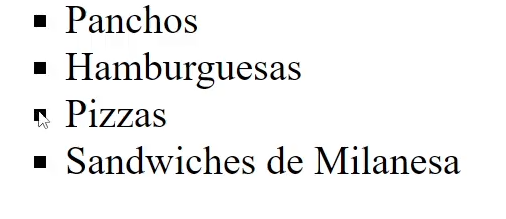
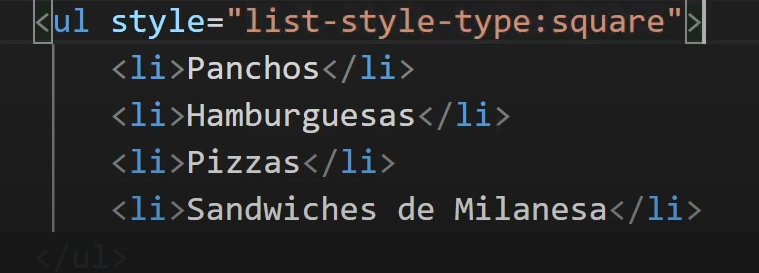
Las listas ordenadas (order lists) se representan mediante la etiqueta **<ol>** y se usan para presentar una lista de elementos que si tienen un orden específico. Cada elemento de la lista se marca con un número, letra o tipo de orden específico antes de su contenido.



**Tipos de orden con caracteres**



**Lista no ordenada usando un poco de css**

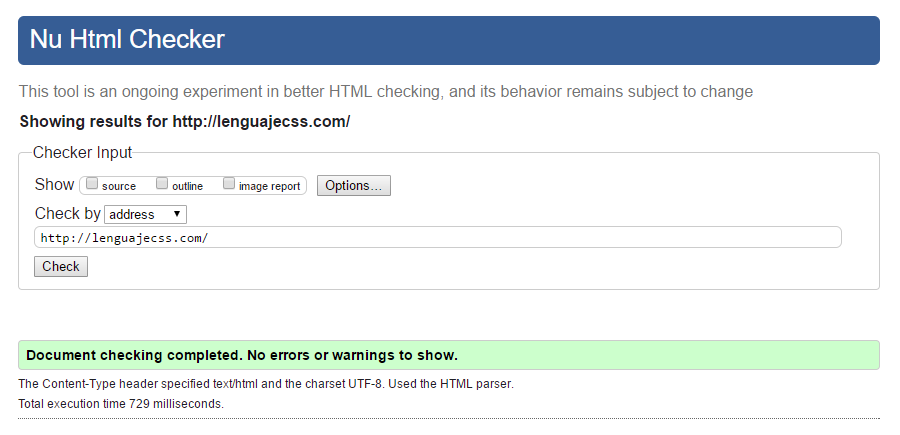


**Prueba de validación si mi archivo HTML es correcto.**

3 formas para validar el documento HTML

1. By URI: Indicando la URL de la página que se quiere validar. Obviamente, la página debe estar publicada en Internet.
2. By File upload: Seleccionando el archivo .html a analizar y subiéndolo al validador.
3. By Direct Input: Pegando directamente el código HTML que se quiere analizar en una casilla de texto.

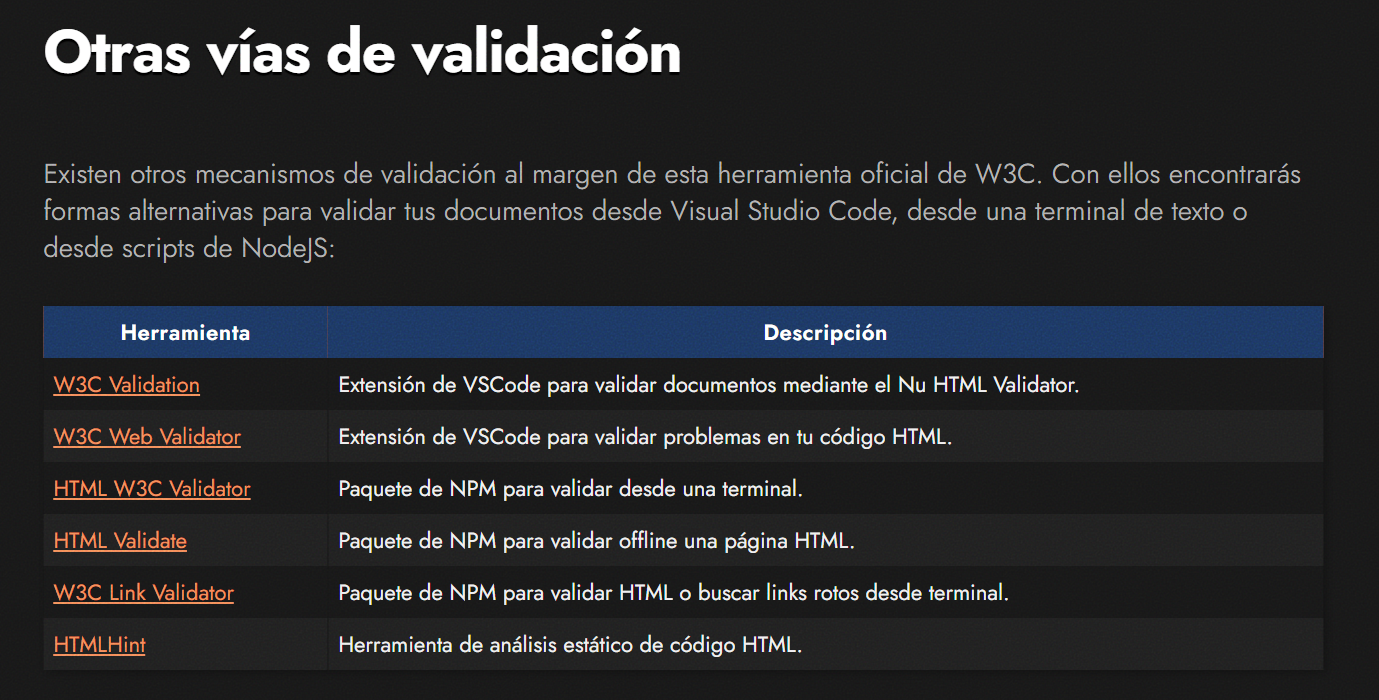
Página para validar el código <https://validator.w3.org>



Source: Nos muestra el código fuente, numerado por líneas. Útil para corregir problemas sobre la marcha o comprobar los errores en las líneas especificadas.

Outline: Nos muestra una especie de árbol o esquema de la página.

Image report: Genera un informe de las imágenes del sitio web, con una miniatura y sus características asociadas.



**CSS**

CSS por sus siglas en inglés significa “Cascading Style Sheets”, es decir, **hojas de estilo en cascada.** Es considerado un Lenguaje de Diseño gráfico o de estilos y es usado en conjunto con algún lenguaje de maquetado, como por ejemplo HTML.

CSS se encarga de diferenciar lo que es el contenido en si de una página web (HTML) y los estilos (colores, tamaños, diseño en general, etc).

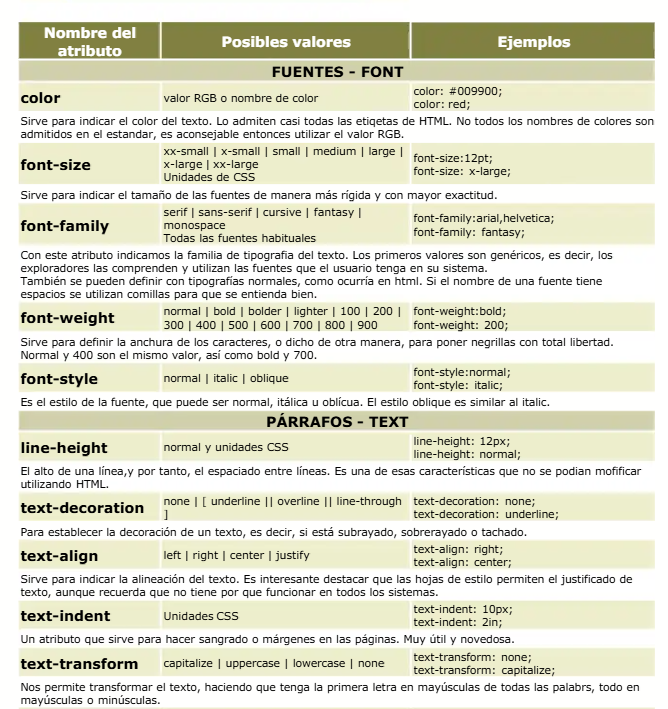
**Sintaxis de CSS**

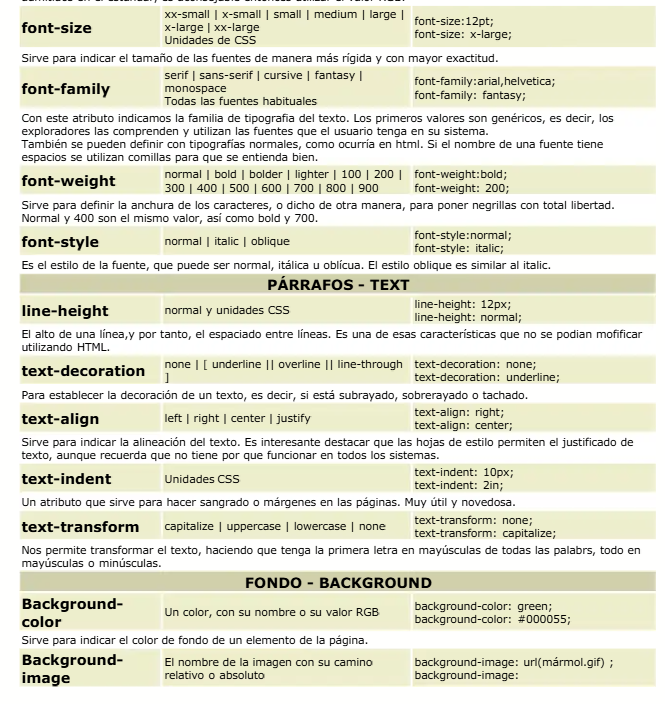
La sintaxis de CSS está basada principalmente en 3 partes:

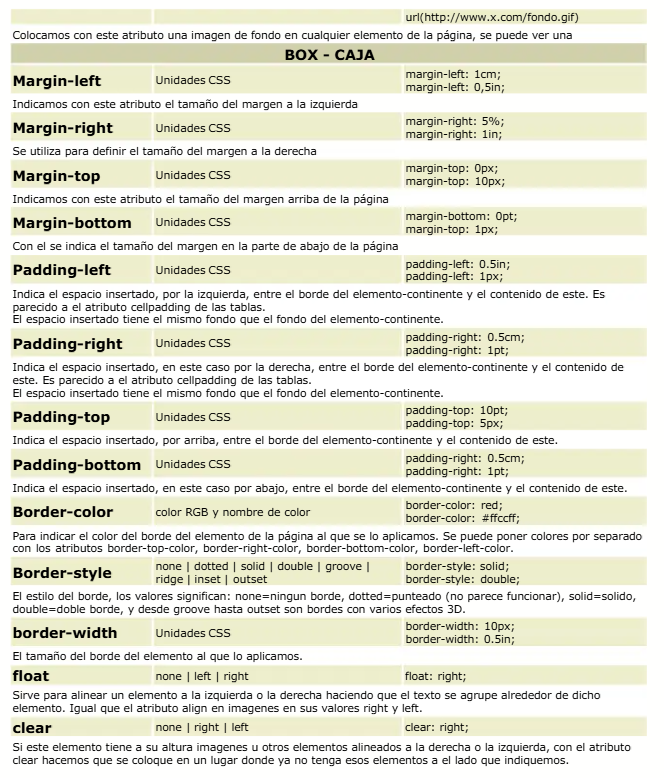
1. **Selectores:** Con ellos podemos especificar a qué elementos de una determinada página nos estamos refiriendo.
2. **Propiedades:** Están representadas por palabras reservadas que se utilizan para brindar diferentes tipos de estilos. Éstas se escriben en inglés y minúsculas. Ej: color
3. **Valores:** Por cada propiedad, existe un valor que se le puede ser asignado. Ej: red.

Un ejemplo a nivel código podemos ver a continuación, donde del lado izquierdo vemos la forma genérica de implementa y del lado derecho su implementación en código.





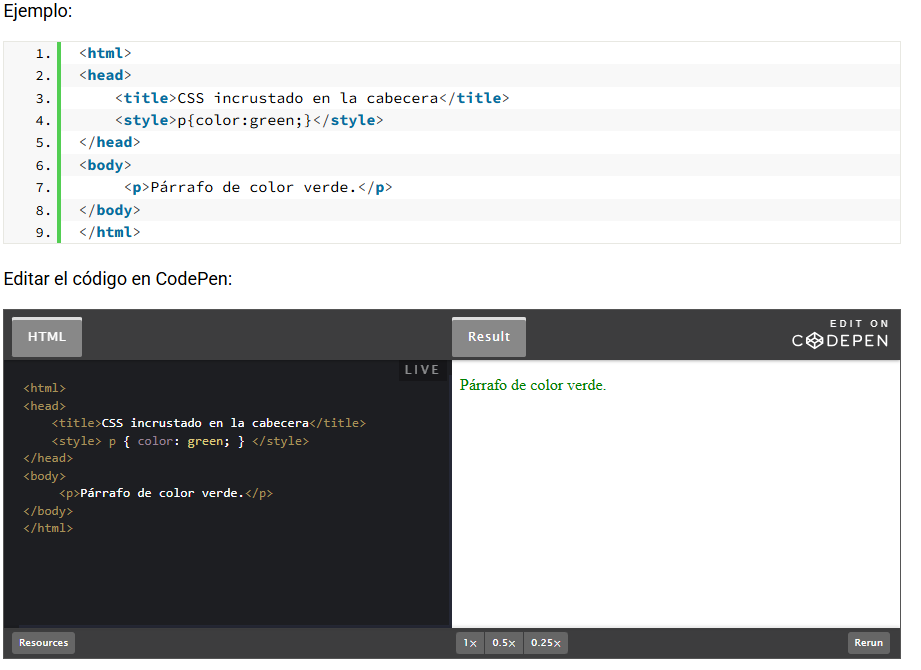




**¿Cómo incorporar CSS a HTML de forma INTERNA y EXTERNA?**

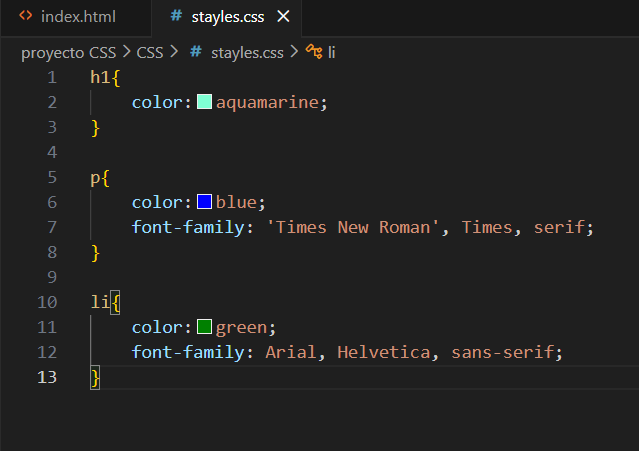
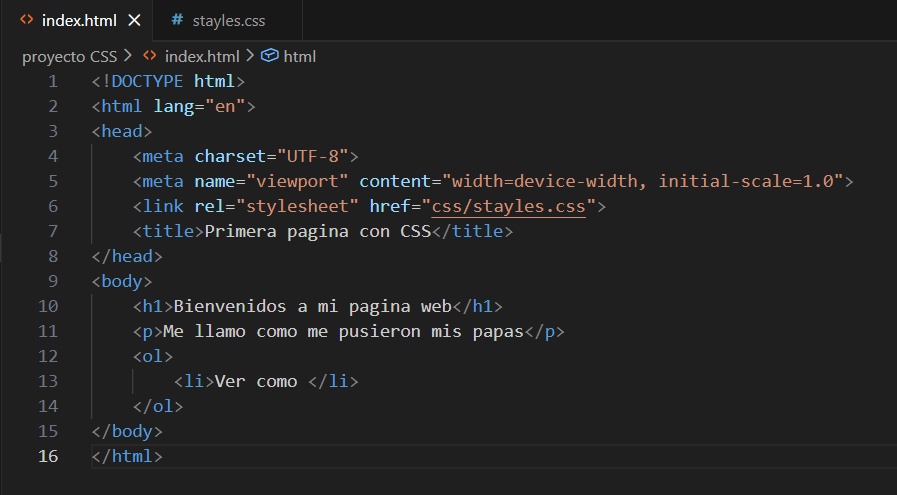
Existen dos formas básicas de hacerlo, de manera **INTERNA** y de manera **EXTERNA**.

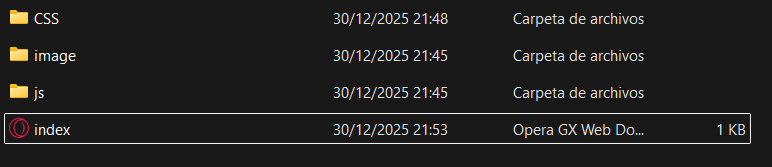
**Interna**

Unamanera muy simple de añadir estilo con CSS es utilizando la etiqueta <style> en la cabecera <head> del fichero HTML del sitio. La desventaja de este método es que, a la hora de realizar cualquier cambio, se debe realizar en múltiples páginas diferentes y el código estará repetido. Su uso puede llegar a ser necesario en el caso de utilizar un gestor de contenido que no permita modificar el archivo CSS directamente. 

**Externo**

Mediante hojas de estilo externas se consigue separar el archivo de estilos del fichero HTML. El archivo de estilos cuenta con la extensión .css y se referencia desde HTML mediante el elemento <link>. Este es el método más eficiente y más sencillo de mantener ya que el código CSS se encuentra separado del fichero HTML.





Se vera asi:



**¿Qué son los selectores en CSS?: Elementos, Clases, ID**

Como hemos visto anteriormente, los selectores nos ayudan a indicar el elemento sobre el que se van a aplicar los estilos. Los selectores pueden apuntar a elementos específicos, clases, identificadores o incluso atributos de un elemento.

**Selectores**

Existen muchos tipos de selectores y algunos de los más destacados son los que se detallan a continuación.

**Selector universal (\*)**

Es útil cuando deseas aplicar un estilo a todos los elementos en una página web sin excepción. Es una forma rápida y sencilla de establecer reglas generales que afectarán a todos los elementos en el documento. **Por ejemplo**, si deseas eliminar todos los márgenes y rellenos predeterminados de los elementos en tu página para empezar desde cero, puedes usar el selector universal para restablecerlos.

Sintaxis: \* {atributo: valor;} ejemplo: \* {color: grey;} /\* El estilo se aplicará a todos los elementos de la página\*/

**Selector etiqueta**

Sintaxis: etiqueta { atributo:valor }

Ejemplo: p {color: green;} /\* El estilo se aplicará a todos los elementos <p>.\*/

**Selector Clases**

Permite seleccionar elementos HTML que tienen una clase específica. Podemos creer la cantidad de clases que creamos necesarias, donde dentro de cada una de ellas podemos agrupar una serie de elementos que tengan cosas en común. Las clases se definen con el atributo «class» de un elemento HTML y se identifican en el archivo CSS con un punto (.) antes del nombre de la clase en cuestión que se quiere seleccionar. Por ejemplo, el selector «.titulos» seleccionaría todos los elementos HTML de una clase llamada «titulos».

Sintaxis: .clase { atributo:valor }

Ejemplo: .blend{color: red;} /\* El estilo se aplicará a cualquier elemento que tenga la clase .blend \*/

**Selector ID**

Permite seleccionar un elemento HTML que tiene una id en específico. Las ids se definen en el atributo «id» de un elemento HTML y se identifican con el numeral/almohadilla (#) antes del nombre de la id en el archivo CSS. Por ejemplo, el selector «#principal» seleccionaría el elemento HTML que cuente con la id llamada «principal».

Sintaxis: #id { atributo:valor }

Ejemplo: #cent {color: blue;} /\* El estilo se aplicará al elemento que tenga el id #cent \*/

**Selector descendiente**

Un elemento es descendiente de otro cuando se encuentra entre las etiquetas de apertura y de cierre del elemento padre.

Sintaxis: selector1 selector2 selectorN {atributo: valor;} /\* El estilo se aplica sobre el selector N \*/

Ejemplo: div p { color: black;} /\* El estilo se aplica a todos los párrafos que se encuentren dentro de una etiqueta div \*/

**Combinación de selectores**

La combinación de selectores nos permite dar un estilo a todos los selectores indicados.

Sintaxis: selector1, selector2, selector3{atributo: valor;} /\* El estilo se aplica sobre los selectores indicados \*/

Ejemplo: div, p { color: orange;} /\* El estilo se aplica a todos los divs y párrafos \*/

**Selector de hijos**

Se usa para seleccionar un elemento que es hijo de otro elemento.

Sintaxis: selector1 > selector2 {atributo: valor;} /\* El estilo se aplica sobre el selector 2 \*/

Ejemplo: div > p { color: white;} /\* El estilo se aplica a todos los párrafos que sean hijos de un div \*/

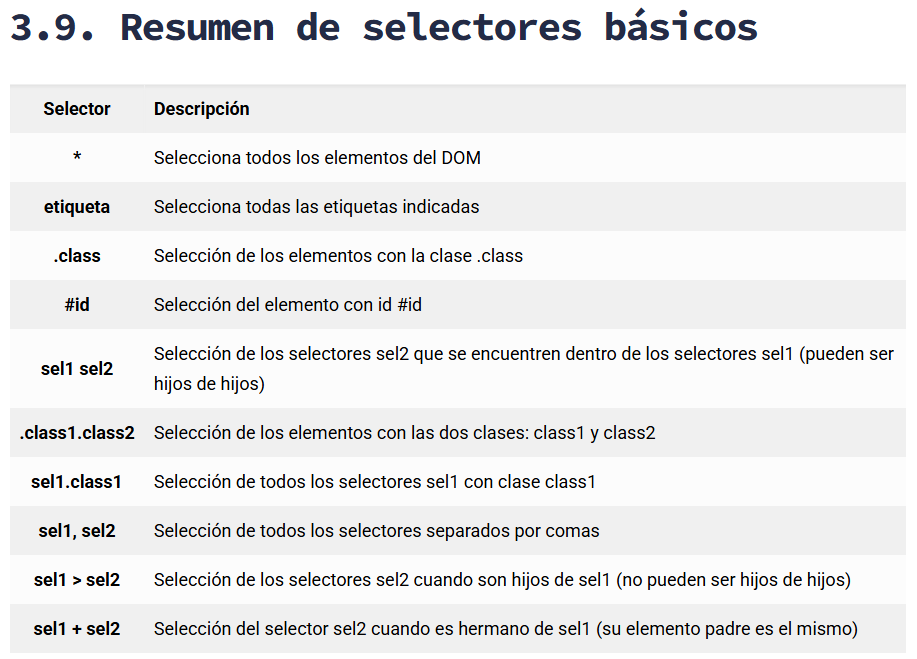
**Nota:** Diferencia entre selector de hijos y selector descendiente. Los hijos son los elementos que se encuentran un nivel directamente debajo de el mismo elemento section. Los descendientes son todos los elementos que se encuentran dentro del elemento section sean los mismos hijos o los hijos de los hijos.

**Selector adyacente**

Se usa para seleccionar elementos que son hermanos, es decir, su elemento padre es el mismo y están seguidos en el código HTML.

Sintaxis: selector1 + selector2{ atributo: valor; } /\* El estilo se aplica al selector 2 \*/

Ejemplo: div + p { color: black;} /\* El estilo se aplica a todos los párrafos que sean hermanos de un div \*/



**Pseudoclases en CSS**

Las pseudoclases son selectores especiales que permiten seleccionar elementos HTML basados en un estado, comportamiento o situación determinada. Estos selectores se utilizan para aplicar estilos a un elemento HTML de manera diferente dependiendo de cómo se encuentre o cómo se interactúe con él.

Las pseudoclases se definen en un archivo CSS añadiendo dos puntos antes de si misma. En el caso de existir selectores de etiqueta, id o clases, estas se escribirían a su izquierda.

En CSS existen muchas pseudoclases, y se han agregado más en las versiones más recientes de CSS. Algunas de las pseudoclases más comunes incluyen:

* :hover
* :active
* :focus
* :visited
* :first-child
* :last-child
* :nth-child()
* :not()
* :target
* :enabled
* :disabled
* :checked

Estos son solo algunas de las tantas pseudoclases existentes. Cabe destacar, que en cada versión y actualización de CSS pueden incorporarse nuevas, por lo tanto, es importante consultar siempre la documentación oficial de CSS para obtener una lista completa y actualizada de todas las pseudoclases disponibles.

1. **Interacción y Usuario (Estados)**

Estas responden directamente a lo que el usuario está haciendo con el mouse o el teclado.

:hover: Se activa cuando el puntero del ratón está sobre el elemento.

Uso: Cambiar el color de un botón o subrayar un enlace para indicar que se puede clicar.

:active: Se activa justo en el momento en que el usuario presiona el botón (durante el clic).

Uso: Dar un efecto de "hundido" o cambio de color rápido para confirmar que el clic fue detectado.

:focus: Se activa cuando un elemento tiene el "foco" (normalmente mediante el teclado con la tecla Tab o al hacer clic en un cuadro de texto).

Uso: Resaltar el borde de un input para que el usuario sepa dónde está escribiendo.

:visited: Se aplica a los enlaces (<a>) que el usuario ya ha visitado anteriormente en ese navegador.

Uso: Cambiar el color del enlace (típicamente a morado) para que el usuario sepa qué páginas ya leyó.

1. **Estructura y Posición (Hijos)**

Te permiten seleccionar elementos sin crear clases nuevas, basándote en su orden en el código.

:first-child: Selecciona el elemento si es el primer hijo de su contenedor.

:last-child: Selecciona el elemento si es el último hijo de su contenedor.

:nth-child(n): Es la más flexible. Selecciona el hijo número "n".

Uso: li:nth-child(3) elige el tercer elemento. También puedes usar even (pares) o odd (impares) para hacer tablas rayadas.

1. **Estados de Formulario**

Súper útiles cuando empieces a trabajar con JavaScript y validación de datos.

:enabled: Selecciona elementos de formulario que están activos y se pueden usar.

:disabled: Selecciona elementos que tienen el atributo disabled (están bloqueados y no se pueden clicar ni escribir en ellos).

:checked: Se aplica a los checkbox o radio buttons que el usuario ha marcado.

Uso: Cambiar el estilo de una etiqueta cuando el usuario marca una opción de "Acepto términos".

1. **Filtros y Navegación Especial**

:not(selector): Selecciona todos los elementos excepto los que coincidan con lo que pongas entre paréntesis.

Ejemplo: p:not(.especial) pintará todos los párrafos que NO tengan la clase .especial.

:target: Se activa cuando la URL de la página tiene un ID que coincide con el elemento (ejemplo: misitio.com/#seccion1).

Uso: Resaltar visualmente una sección cuando el usuario llega a ella a través de un enlace interno (ancla).

Practica para el borde de un texto



Colocar una imagen en un texto con la siguiente anotación

**Background**

background es una propiedad abreviada que permite definir varias propiedades de fondo en una sola línea.

background-image: define la imagen que se usa como fondo de un elemento HTML.

background-image: url("imagen.jpg");

La imagen no ocupa espacio en el DOM, es solo decorativa.

background-repeat: Controla si la imagen se repite.

Valores:

* repeat → se repite en ambas direcciones (valor por defecto)
* no-repeat → no se repite
* repeat-x → se repite horizontalmente
* repeat-y → se repite verticalmente
* Ejemplo: background-repeat: no-repeat;

background-position: Define la posición inicial de la imagen de fondo.

Valores comunes:

* center
* top
* bottom
* left
* right
* combinaciones: center top, right bottom
* valores numéricos: 50% 50%, 10px 20px

Ejemplo: background-position: center;

background-attachment: Controla si el fondo se mueve con el scroll.

Valores:

* scroll → se mueve con la página (default)
* fixed → fondo fijo (efecto parallax simple)
* local → se mueve con el contenido interno

Ejemplo: background-attachment: fixed;

background-origin: Define desde dónde comienza a dibujarse la imagen.

Valores:

* padding-box (default)
* border-box
* content-box

Ejemplo: background-origin: content-box;

background-clip: Define hasta dónde se extiende el fondo.

Valores:

* border-box → cubre hasta el borde
* padding-box
* content-box

Ejemplo: background-clip: padding-box;

**Múltiples fondos:** Un elemento puede tener más de un fondo.

background-image:

url("capa1.png"),

url("capa2.png");

Cada fondo puede tener su propio tamaño y posición:

background-size: cover, 100px;

background-position: center, top right;

background-blend-mode: Define cómo se mezclan el color y la imagen de fondo.

Valores comunes:

* multiply
* overlay
* screen
* darken
* lighten

Ejemplo: background-blend-mode: overlay;

**Margin**

Padding: es el espacio interno de un elemento.

Overflow: hidden o scroll

Outline

**Posción de texto**

Text-align: center o left o right o justify

Text-decoration

Text-shadow

**Estados de Links**

a:link{

color: blues;}

a:visited{

color: green}

a:hover{

color: yellow }

a:active{

color: purple}

**Listas**



**Tablas**



**Display, max-width y position**

**Float:** Toma el elemento que encuentra y lo hará flotar

**Inline block**

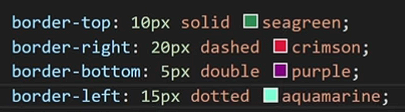
Practica

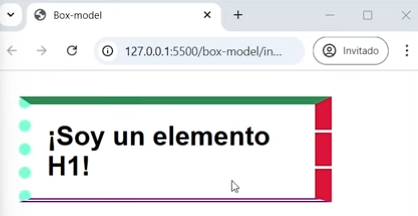
Para alinear los <div></div>

display: contents;

Flex y did

**Borde**





**SHORTHAND BÁSICO**

**1. width y height**

* Definen el tamaño del contenido del elemento.
* **No existe shorthand conjunto**.
* Se declaran por separado.

width: 200px;

height: 100px;

**2. padding (shorthand)**

Define el **espacio interno** entre el contenido y el borde.

**Sintaxis**

padding: valor;

**Valores**

* **1 valor** → todos los lados

padding: 10px;

* **2 valores** → arriba/abajo | izquierda/derecha

padding: 10px 20px;

* **3 valores** → arriba | lados | abajo

padding: 10px 20px 30px;

* **4 valores** → arriba | derecha | abajo | izquierda

padding: 10px 15px 20px 5px;

**3. margin (shorthand)**

Define el **espacio externo** del elemento.

**Funciona igual que padding**

margin: 10px;

margin: 10px 20px;

margin: 10px 20px 30px;

margin: 10px 15px 20px 5px;

**4. border (shorthand)**

Define el borde del elemento.

**Orden obligatorio**

border: ancho estilo color;

**Ejemplo**

border: 2px solid black;

Equivale a:

border-width: 2px;

border-style: solid;

border-color: black;

**5. Resumen clave (examen / entrevista)**

width / height → tamaño del contenido

padding → espacio interno

border → borde

margin → espacio externo

Orden del box model:

contenido → padding → border → margin