

CURSO DE HTML Y CSS

Dorian Desings



HTML



CSS



PERE MANEL VERDUGO ZAMORA
pereverdugo@gmail.com

Temario

- Primeros pasos
 - Editor de código
 - Plugins
 - Vocabulario web
 - Conceptos e historia de HTML
 - Estructura de un documento
 - Crear un servidor de desarrollo
- Elementos de sección
 - Encabezados
 - Párrafos
 - Web Semántica
 - Article vs Section
- Elementos a nivel de texto
 - Enlaces
 - Resaltado de texto
- Atributos
 - Atributos obligatorios
 - Atributos personalizados
- Listas
 - Listas ordenadas
 - Listas desordenadas
 - Listas de definición
 - Listas anidadas
- Tablas
 - Construcción
 - Semántica y atributos
 - Estilos para las tablas
- Formularios
 - Estructura
 - Tipos de input
 - Elementos seleccionables
 - Atributos
- Contenido embebido
 - Imágenes
 - Audio
 - Vídeo
- Etiquetas meta y accesibilidad
 - SEO
 - FaceBook
 - Twitter
 - Accesibilidad
- Introducción a GIT
 - Crear un repositorio
 - Subir el curso a github
- Crear y publicar un sitio web
 - Publicar nuestra web de forma gratuita.

#YOAPRENDOCONDORIAN

- Comprar y configurar un dominio y hosting
- Configurar la subida por ftp
- Sigüientes pasos
 - Cómo leer la documentación
 - Qué estudiar después
 - Cómo mantenerse actualizado

Preguntas

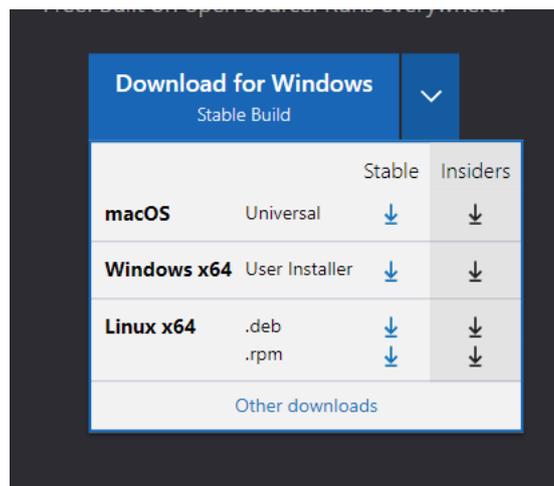
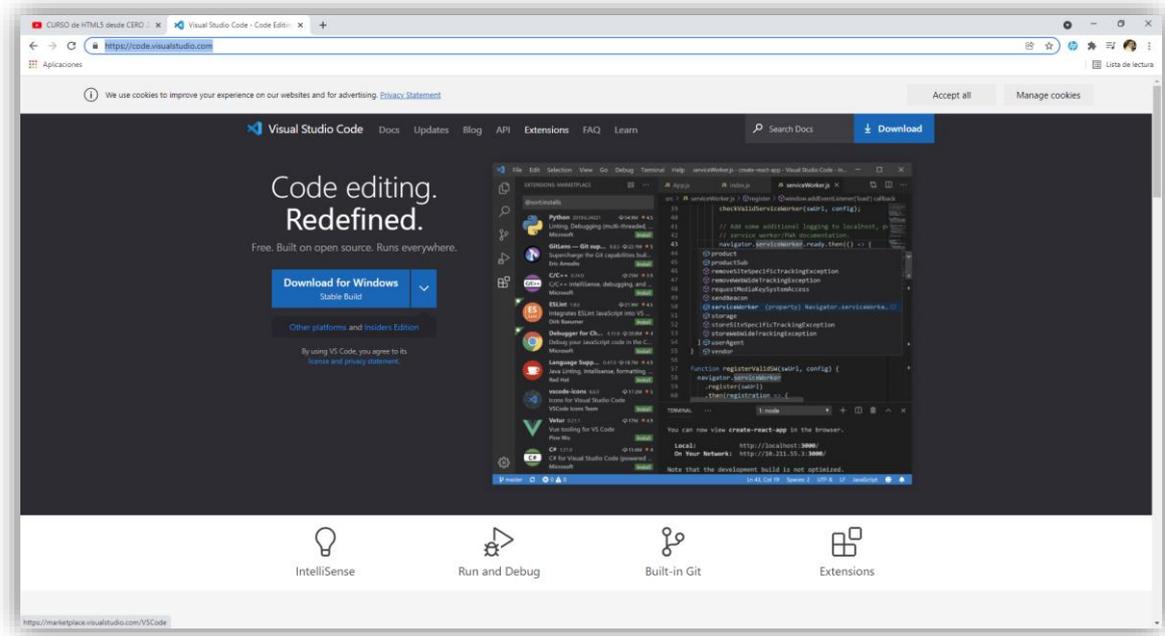
- ¿Cuánto dura el curso? De 7 a 8 horas.
- ¿Está completo? No, no existe ningún curso completo, pero si es el más actualizado.
- ¿Dónde puedo consultar dudas? En <https://discord.com/invite/nE2dZrmZAS>

Video completo



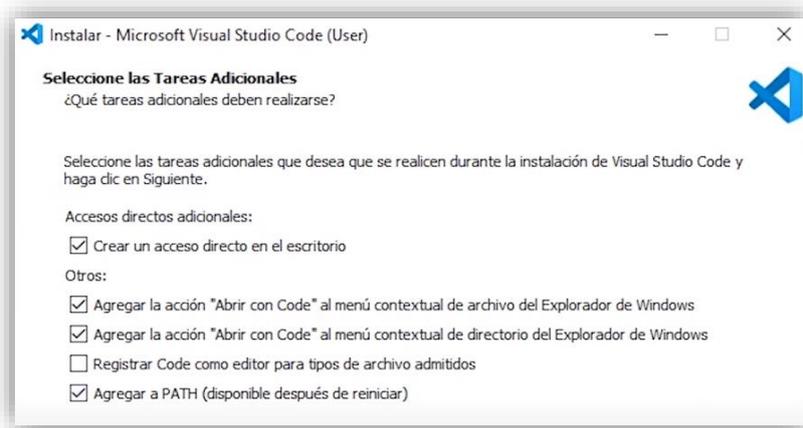
Instalación de Visual Studio Code

Lo podrás descargar desde: <https://code.visualstudio.com/>



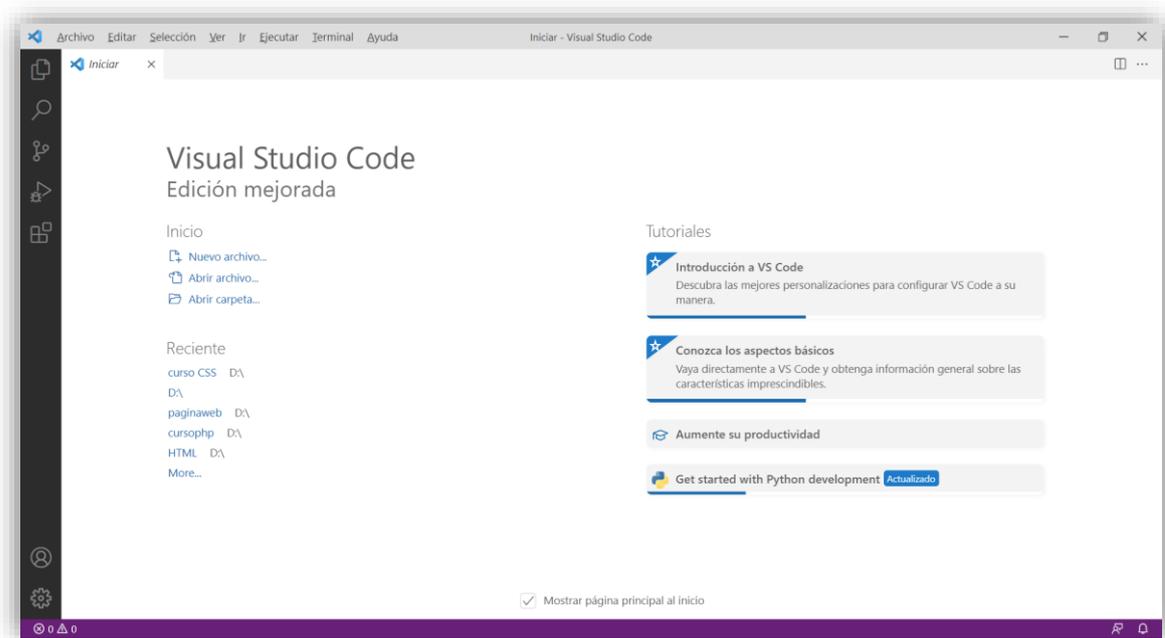
Se puede descargar para macOS, Windows y Linux.

Una vez lo tengamos descargado procedemos a su instalación.



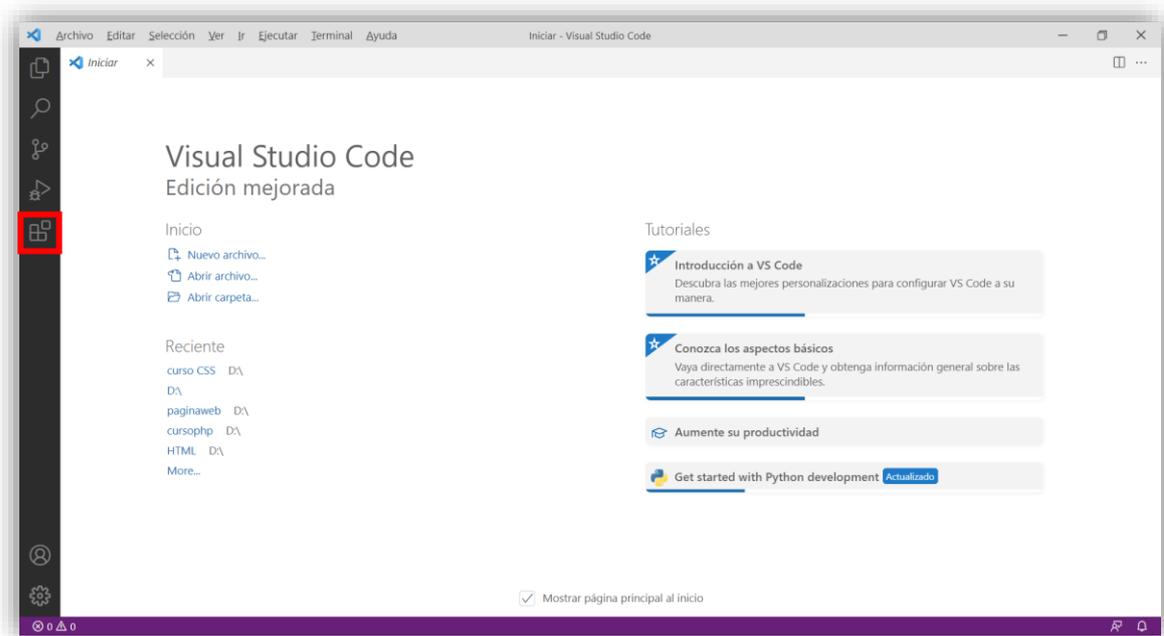
Saldrá una ventana para seleccionar las tareas adicionales.

Una vez lo tengamos instalado procederemos a ejecutarlo:

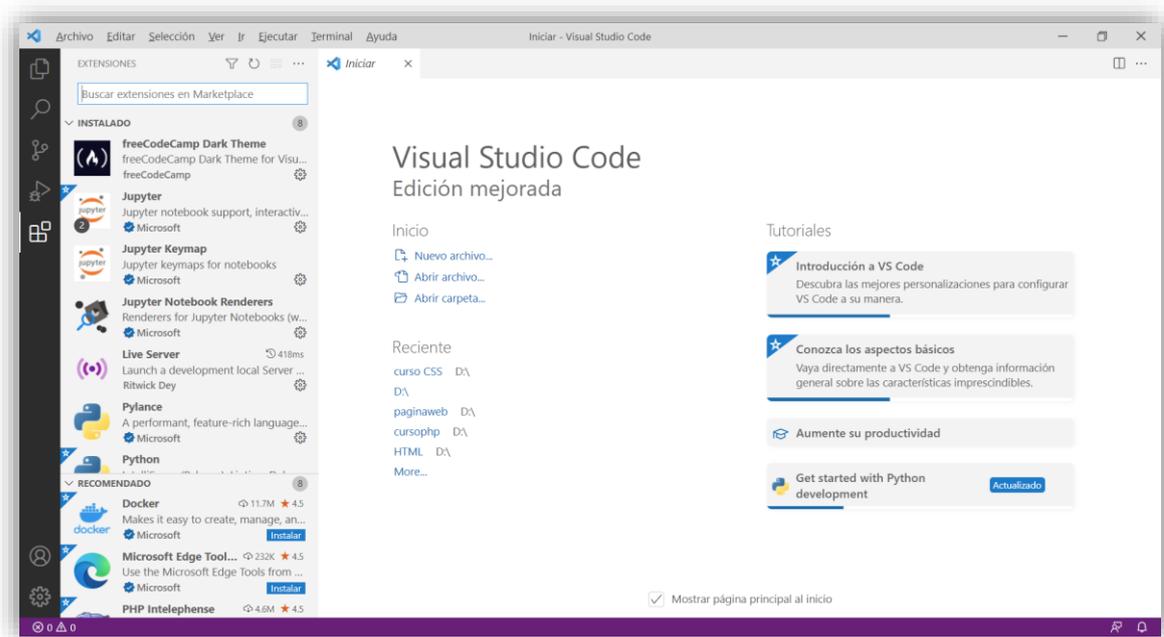


A continuación vamos a instalar una serie de paquetes, también llamados extensiones desde el siguiente botón.

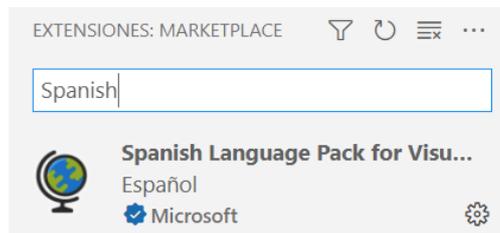
Herramientas para Visual Studio Code

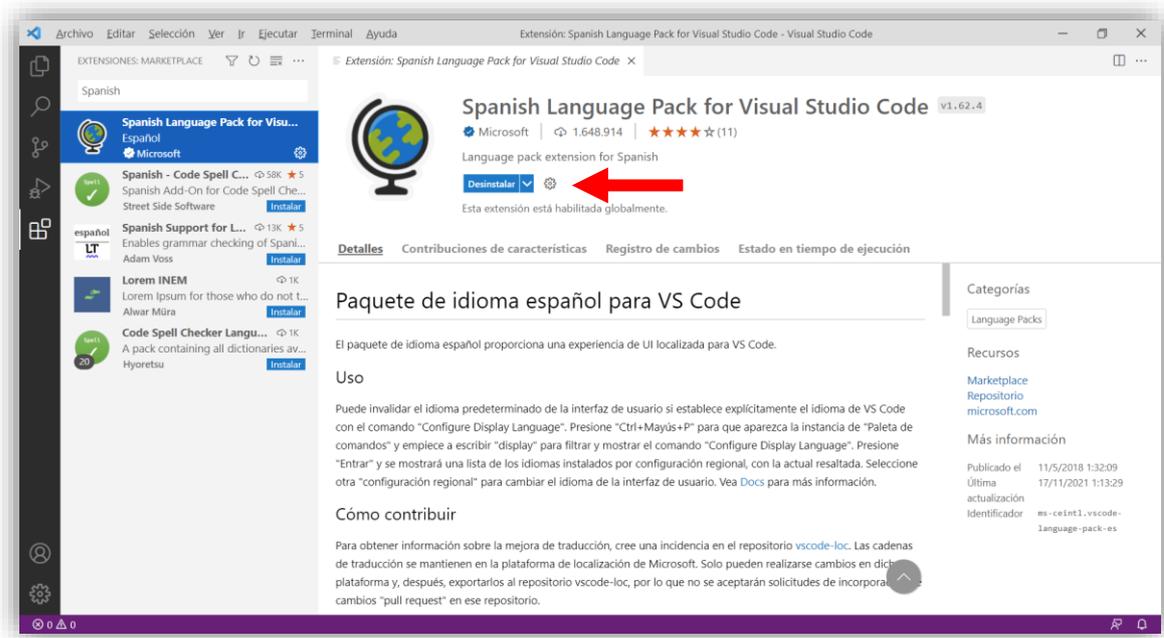


Desde este botón podremos descargar las siguientes herramientas:



Si el editor está en inglés procederemos a instalar el paquete de español.

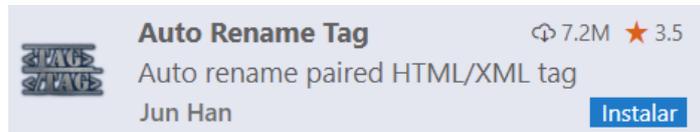




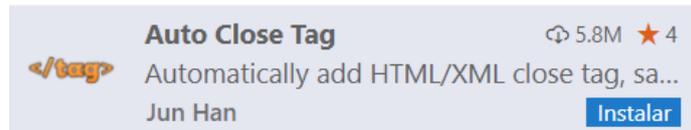
Yo ya lo tengo instalado, de lo contrario empezaremos a su instalación.

Para que se apliquen los cambios, en la parte inferior derecha nos indica que hay que restaurar.

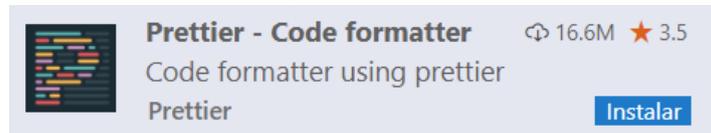
Volvemos a las extensiones.



Descargaremos e instalamos Auto Rename Tag.



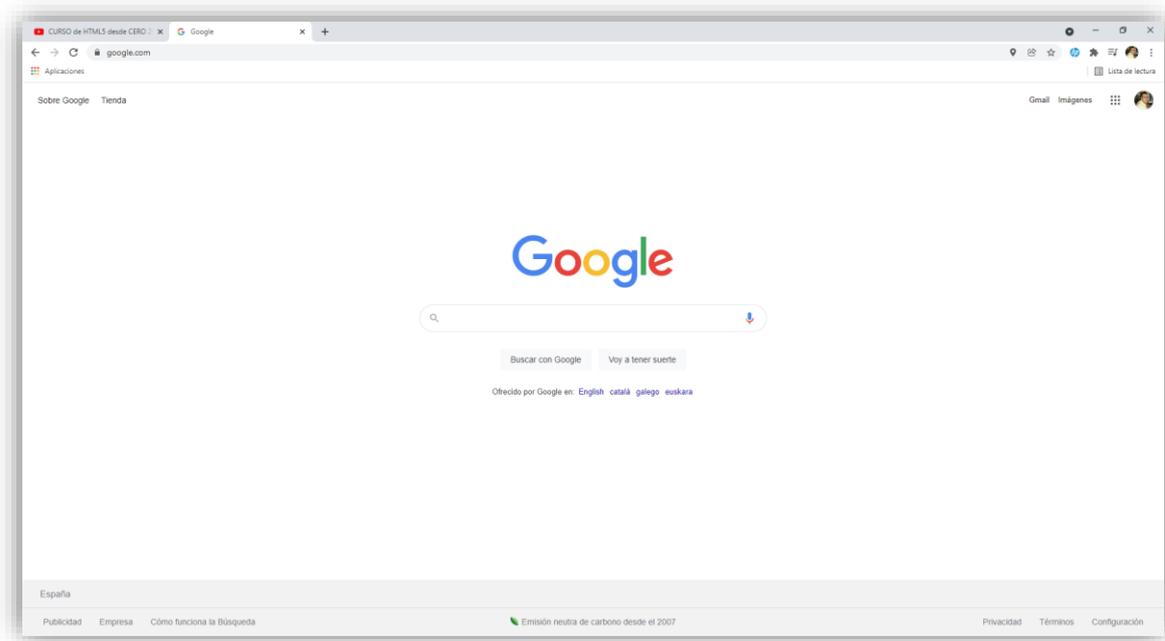
Descargamos e instalamos Auto Close Tag.



Descargamos e instalamos Prettier – Code formatter, formateará el código para que se vea mucho más limpio.

Vocabulario web

- IP: Es el identificador numérico de una página web, es único y representa la dirección donde está el ordenador que contiene esa página web.
- Dominio web/ URL: Uniform Resources Locator. Es el nombre asociado a la IP que utilizamos para solicitar recursos, en nuestro caso un sitio web, si en la barra de dirección escribimos 216.58.210.163 iremos a la página de Google.com



- • DNS: Domain Name System. Es un servidor cuya principal función es traducir el nombre de dominio a su identificador único.
- Sitio web: Es un conjunto de uno a varios recursos web alojados en el mismo dominio.
- Servidor web: Es un ordenador cuyo objetivo es servir recursos web.
- Hosting: Es el almacenamiento del servidor web. El disco duro donde el servidor guarda los recursos.
- Petición: Es la acción de pedir recursos a un servidor.

¿Qué es HTML?

- Es un lenguaje de marcado de hipertexto (Hyper Text Markup Lenguaje).
- HTML no es un lenguaje de programación, es un lenguaje de estructura.
- Es la base con la que están creados TODAS las páginas web del mundo.
- Cada etiqueta le dice al navegador y a los motores de búsqueda cuál es la estructura de los documentos, elementos, organización, etc.

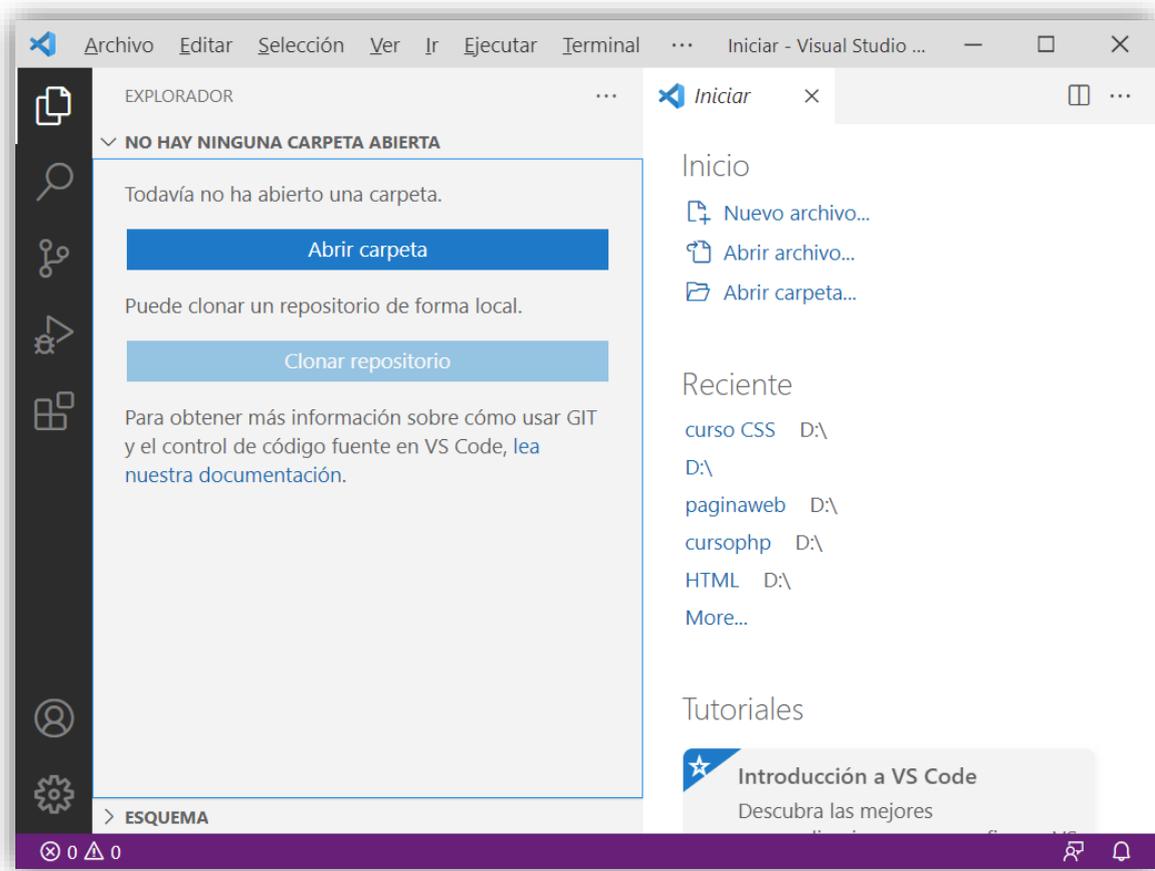
Historia de HTML

- 1989 – Inicio de su desarrollo.
- 1991 – lanzamiento de la web (se basa en 3 conceptos. http, html y url).
- 1992 – Lanzamiento de HTML (Nació de SMGL. Creado por Tim Bernes Lee).
- 1994 – Creación de la W3C (Consortio que define los estándares de la web).
- 1998 – HTML 4 (versión que más duró en HTML).
- 1999 – HTML 4.1 – XHTML (actualización del estándar HTML4).
- 2004 – Creación de la WHATWG.
- 2008 – HTML5 (Lanzado por WHATWG de forma independiente).
- 2014 – Estándar HTML5 (Lanzado por W·C de forma oficial)

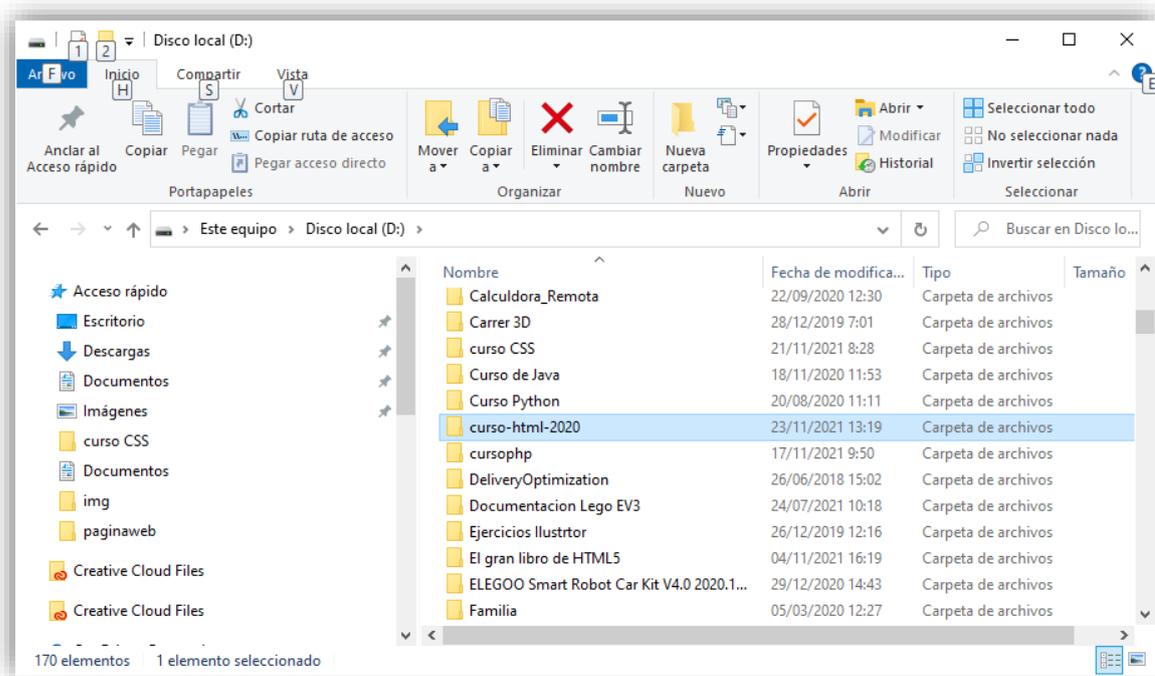
Un video muy interesante es la Guerra de los navegadores:



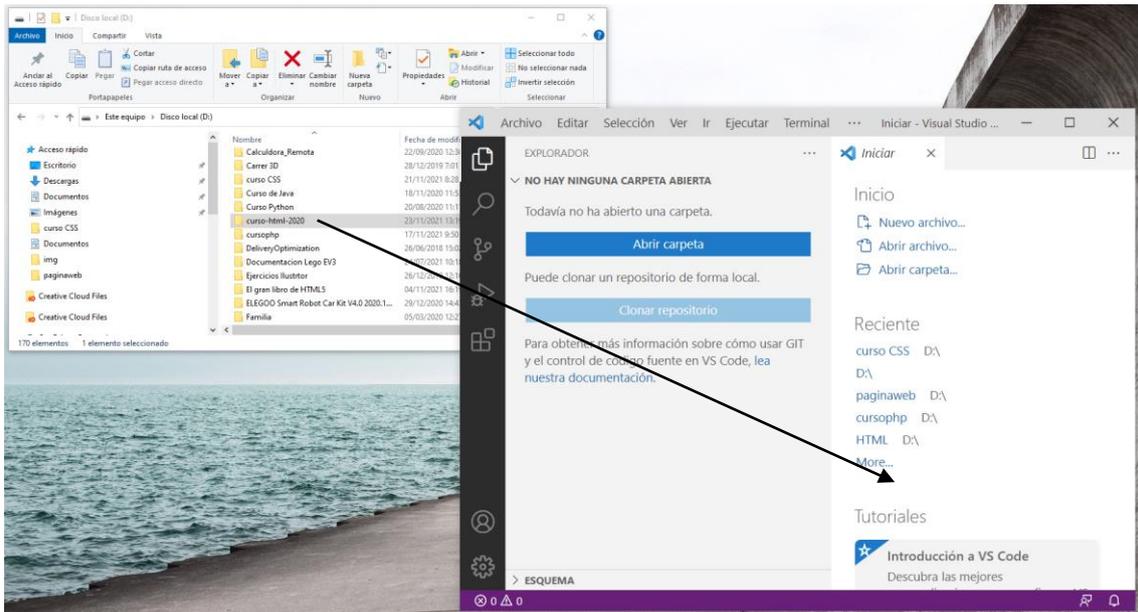
Sintaxis de HTML



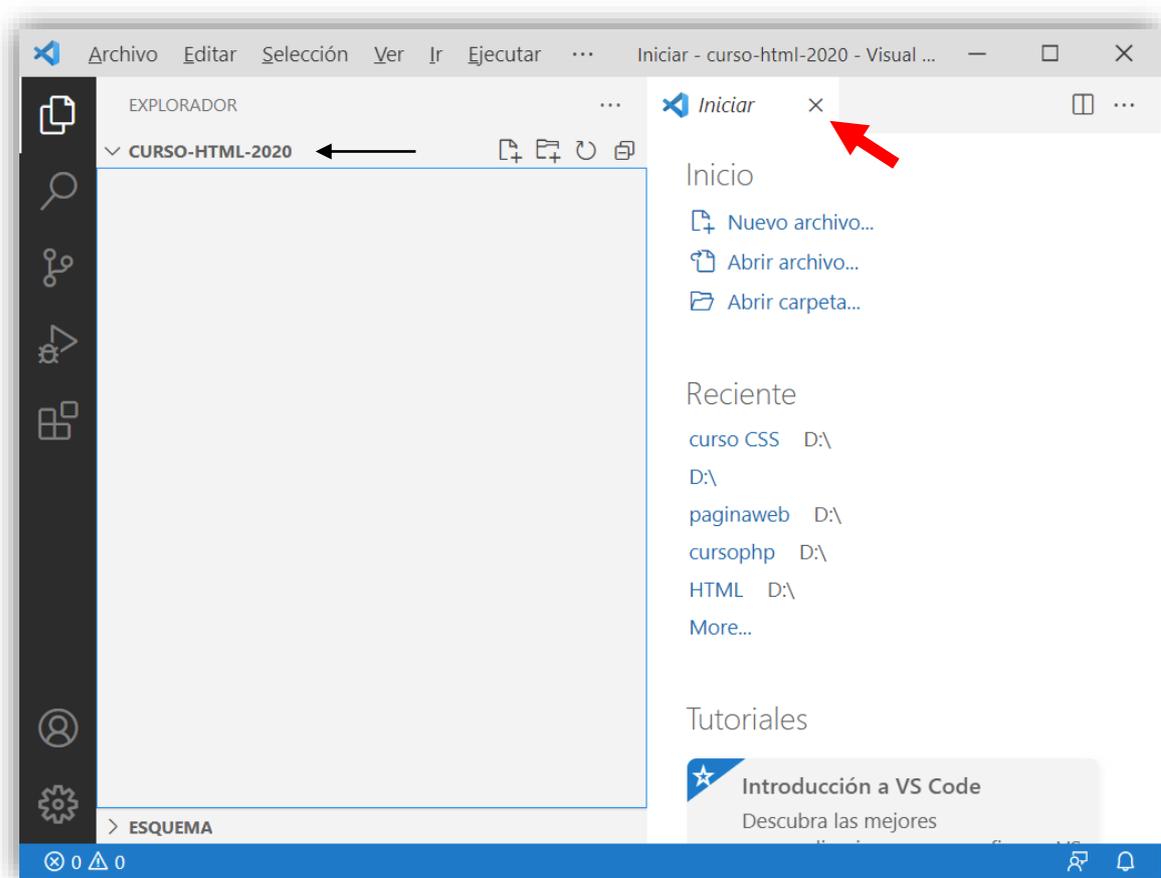
Vamos a crear una carpeta donde guardaremos todos los proyectos de este curso.



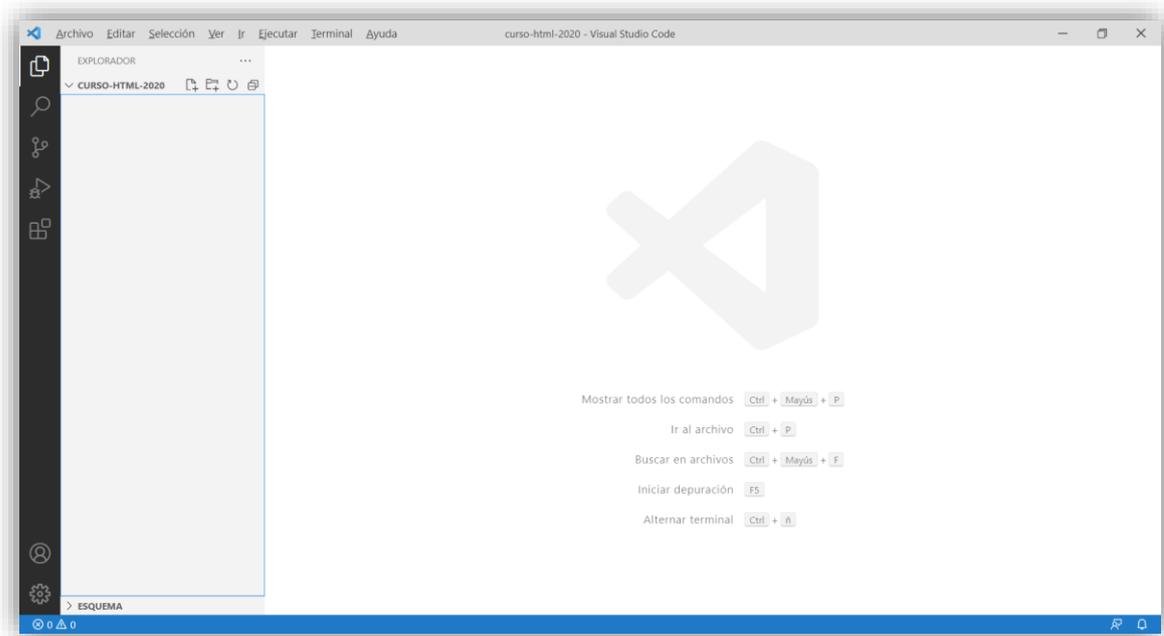
Le llamaremos curso-html-2020, a la hora de dar nombre a la carpeta para vuestros proyectos evitar letras en mayúsculas y espacios.



Seleccionaremos la carpeta y la arrastraremos a Visual Studio Code.

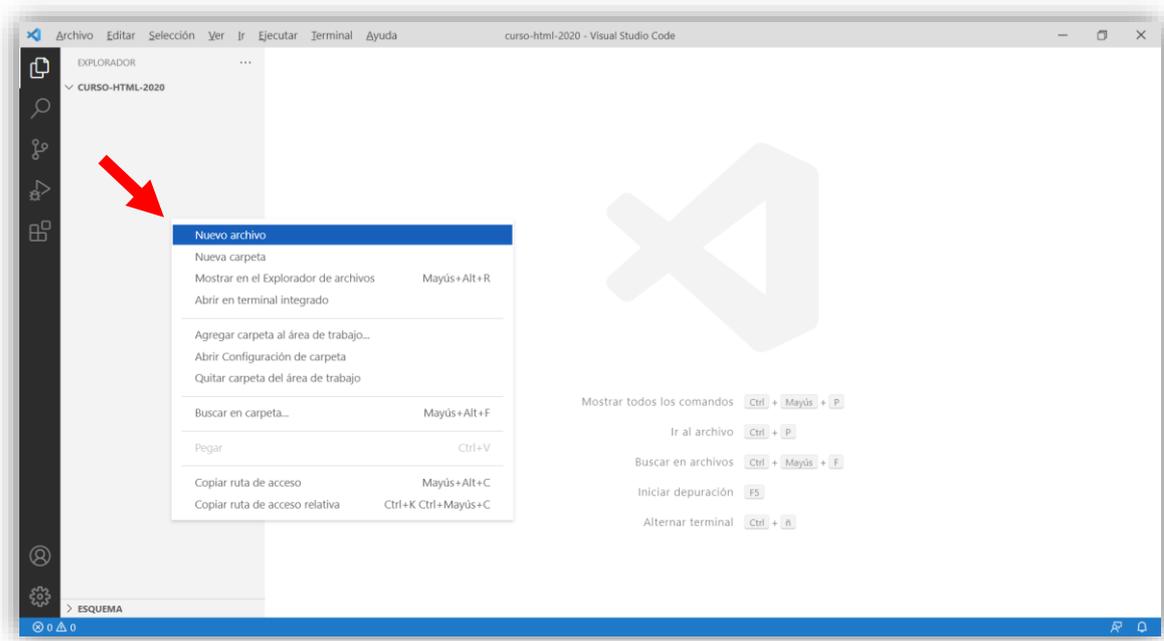


Ya la tenemos en nuestro proyecto.
Ya podemos cerrar la ventana de bienvenida.

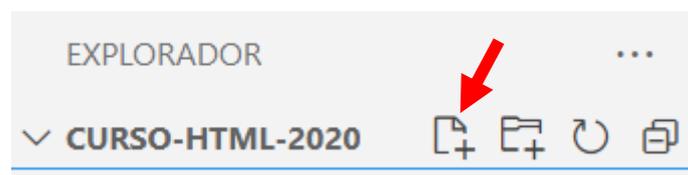


Vamos a crear nuestro primer documento HTML.

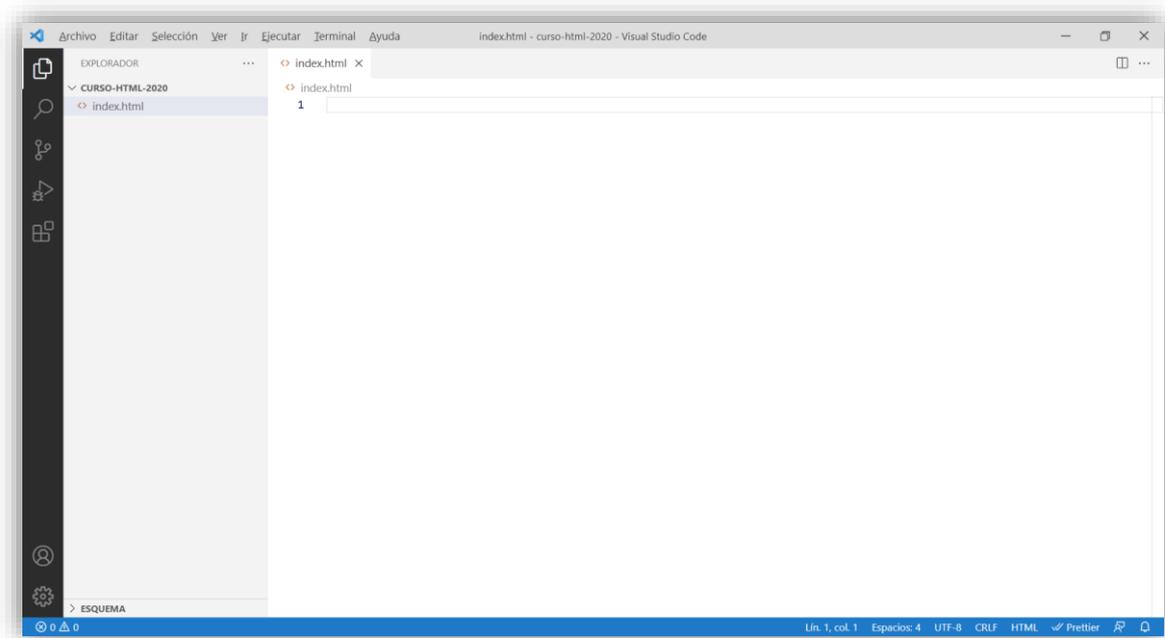
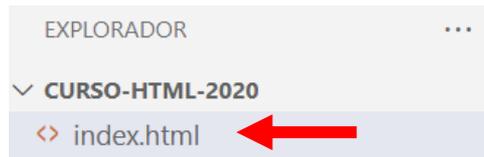
- 1.- Del menú Archivo / Nuevo archivo...
- 2.- Botón derecho sobre el explorador.



3.- En el explorador documento nuevo.



Un nombre correcto de archivo será:



```
<> index.html ×  
<> index.html > ...  
1 <etiqueta></etiqueta>  
2
```

<etiqueta_de_apertura></etiqueta_de_cierre>

Tienen que tener el mismo nombre y la de cierre la diferencia por la barra '/'.

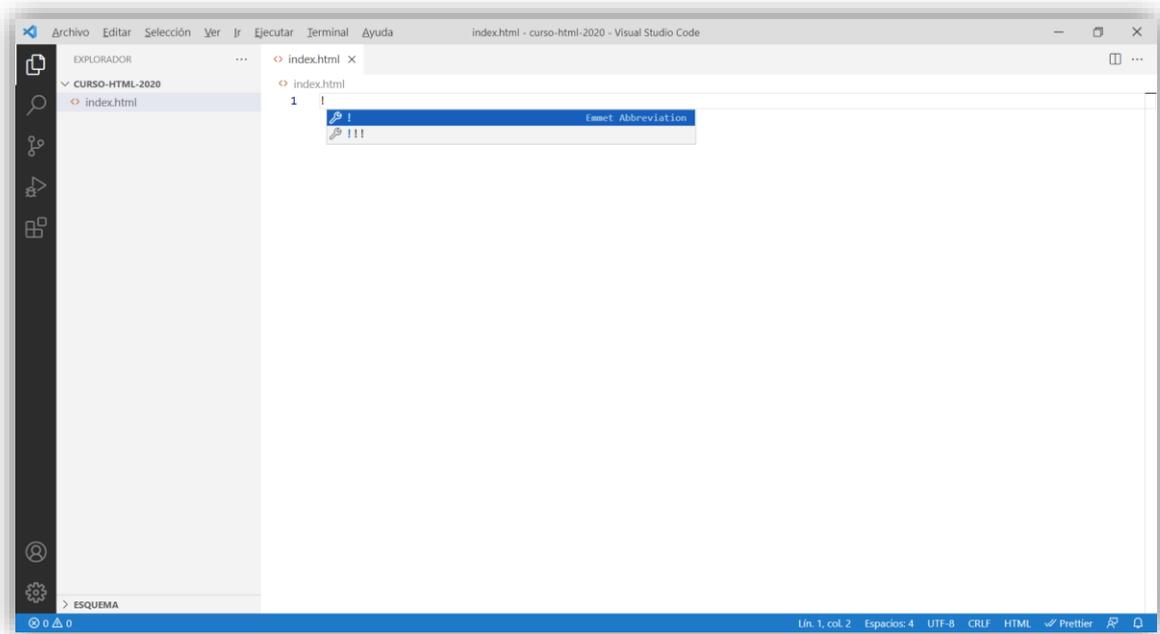
```
1 <etiqueta>  
2 |   Contenido de la etiqueta  
3 </etiqueta>
```

En medio de las dos irá el contenido de la etiqueta.

Existen varios tipos de etiquetas para poner distintos tipos de contenido.

Estructura de un sitio web

Visual Studio Code nos agrega automáticamente esta estructura para ello escribimos ! punto hacia abajo seguido de la tecla intro.



```
<> index.html x
<> index.html > html > head > meta
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10 |
11 </body>
12 </html>
```

Para no confundir vamos a eliminar las líneas con la etiqueta meta.

```
<> index.html x
<> index.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Document</title>
5 </head>
6 <body>
7 |
8 </body>
9 </html>
```

#YOAPRENDOCONDORIAN

<!DOCTYPE html> : Sirve para definir que estándar de documento estamos creando, que estamos siguiendo el estándar de HTML 5.

<html Lang="en"> le decimos que el documento está en inglés, lo vamos a cambiar a español.

<html Lang="es"> </html>

<head></head> El navegador obtiene una información con respecto a los metadatos!

<title></title> El título que pongamos se mostrará en la pestaña de nuestro navegador.

<body></body> Aquí irá todo el contenido de nuestra web.

```
<> index.html > ...
1  <!-- Establece el tipo de estandar del documento (HTML5) -->
2  <!DOCTYPE html>
3  <!-- Inicio del documento HTML -->
4  <html lang="es">
5  <!-- Datos que pasamos al navegador don información de nuestra página web -->
6  <head>
7  |   <title>Document</title>
8  </head>
9  <!-- Representa todo el contenido de nuestra página web -->
10 <body>
11
12 </body>
13 </html>
```

Para poner comentarios de una línea utilizaremos <!-- Comentario -->.

Un método abreviado para escribir un comentario es ctrl + ç.

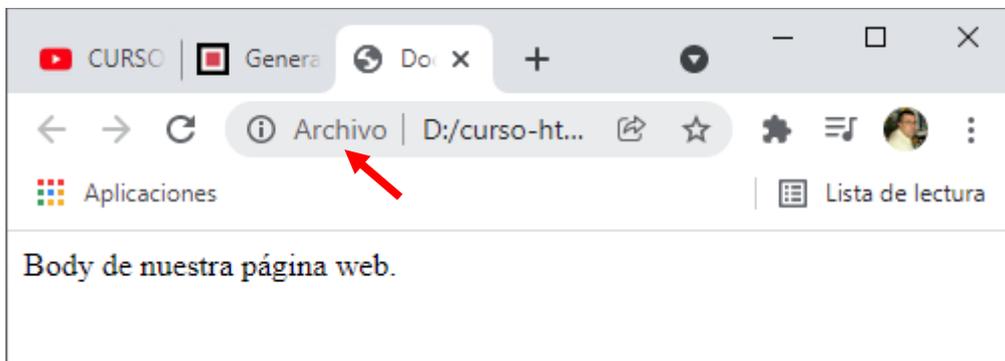
Creando un servidor de desarrollo

```
<> index.html > ...
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4  |   <title>Document</title>
5  </head>
6  <body>
7  |   Body de nuestra página web.
8  </body>
9  </html>
```

Escribimos un texto entre las etiquetas <body> y </body>.

A continuación guardamos el archivo.

Ahora mucha gente lo que hace es irse a la carpeta donde esta el index.html y abrirlo.



Observamos el siguiente mensaje, esto nos indica que algunas opciones no funcionará correctamente y vamos a tener conflictos a la hora de visualizar nuestra página web.

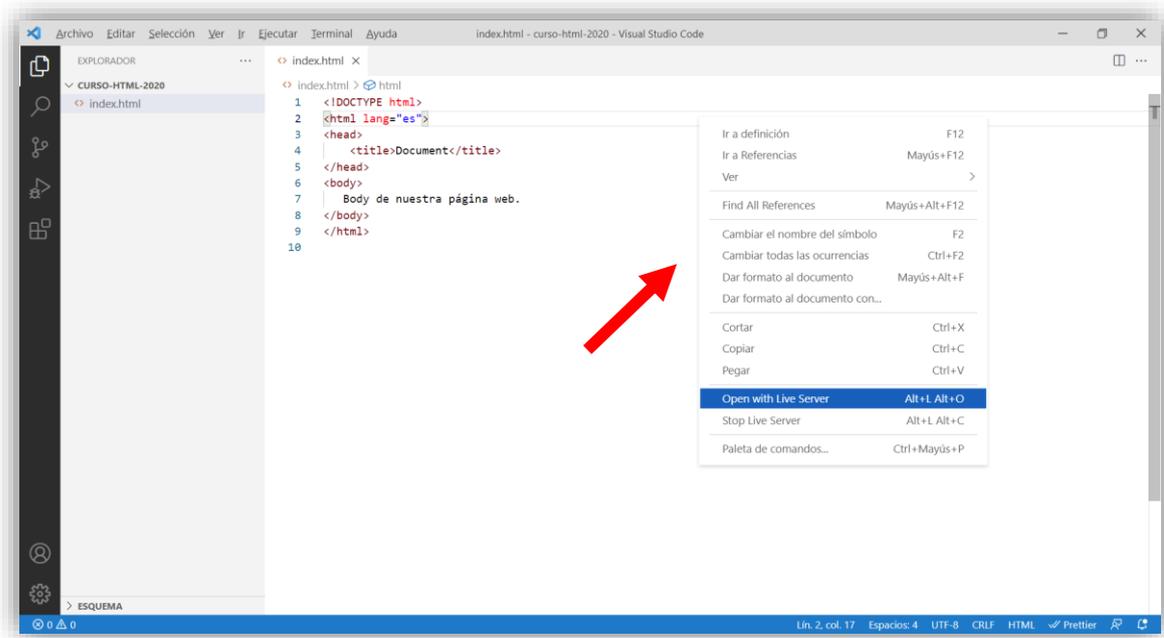
Volvemos a nuestro panel de extensiones de Visual Studio Code.

Vamos a buscar Live Server que instalaremos.

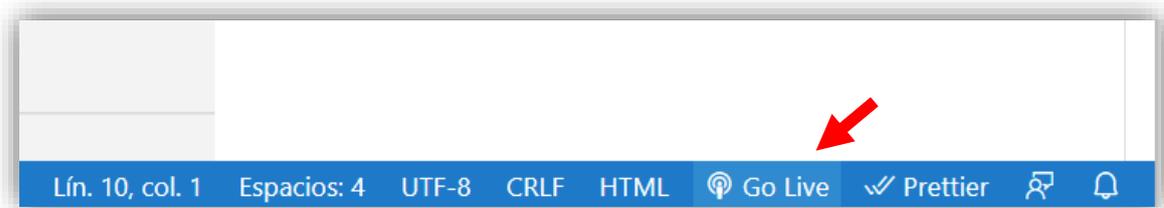


Una vez instalado.

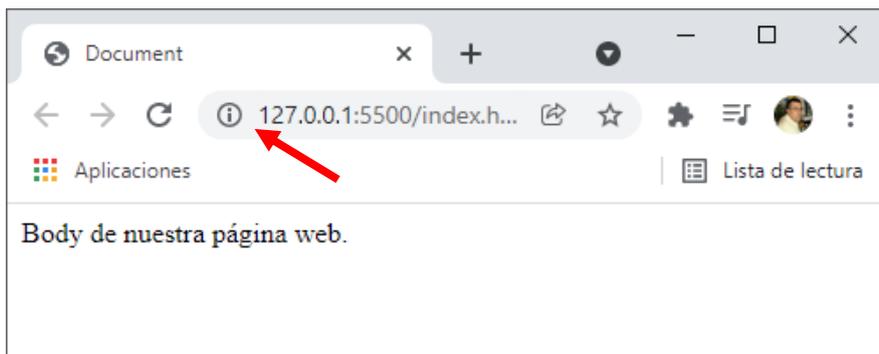
Botón derecho del ratón en la ventana del código y del menú seleccionaremos "Open with Live Server".



Esto hará que en parte inferior tengamos acceso a esta opción.



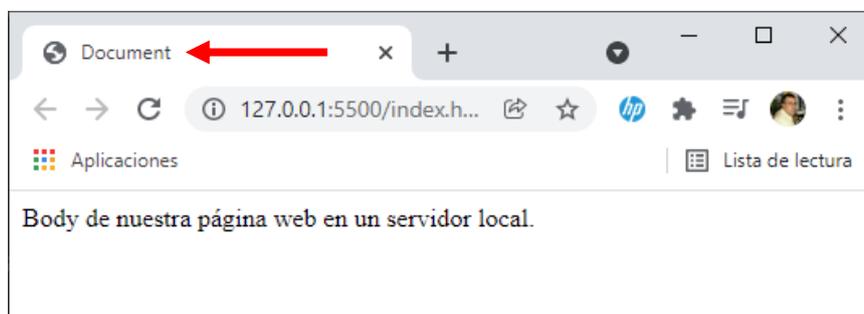
Ahora podrás observar como esta página se ejecuta desde un servidor.



Nos permite la recarga instantánea de los cambios.

```
<> index.html > html > body
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4  |   <title>Document</title>
5  </head>
6  <body>
7  |   Body de nuestra página web en un servidor local.
8  </body>
9  </html>
10
```

Hemos modificado el texto, ahora vamos a guardar los cambios.



Se ha actualizado automáticamente.

Document es el título que tenemos en la etiqueta <title></title>.

Nos podemos encontrar que algunos caracteres como acentos no se mostrarán bien, para esto hay que decirle que estamos trabajando con este tipo de caracteres.

```
<> index.html > html > head > meta
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <title>Document</title>
6  </head>
7  <body>
8  |   Body de nuestra página web en un servidor local.
9  </body>
10 </html>
```

La codificación UTF-8 es la codificación de los caracteres latinos.

Títulos y párrafos

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.

Ejemplo:

```
<> titulos-y-parrafos.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  |   <title>Títulos y párrafos</title>
8  </head>
9  <body>
10 |   <h1>Harry Potter</h1>
11 |   <h2>Sinopsis</h2>
12 |   <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Corrupti, ad!
    |   Consequuntur nobis recusandae, veritatis aliquid facere sapiente perspiciatis
    |   laborum sequi corrupti fuga delectus minus et pariatur accusamus consectetur beatae
    |   veniam?</p>
13 </body>
14 </html>
```

Presionamos Go Live.

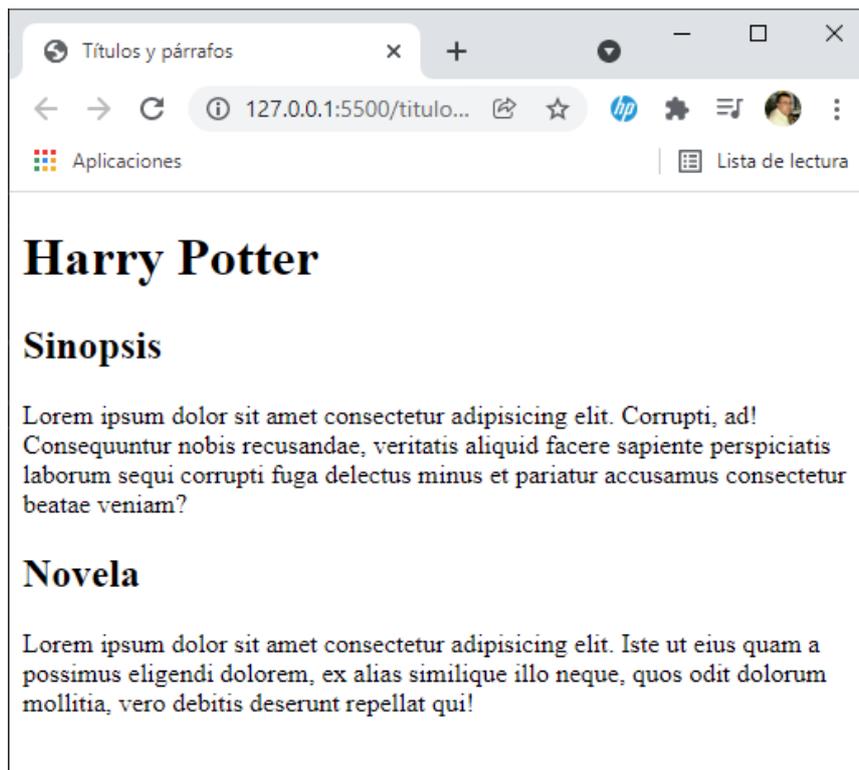


```

<> titulos-y-parrafos.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  |   <title>Títulos y párrafos</title>
8  </head>
9  <body>
10 |   <h1>Harry Potter</h1>
11 |   <h2>Sinopsis</h2>
12 |   <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Corrupti, ad!
    |   Consequuntur nobis recusandae, veritatis aliquid facere sapiente perspiciatis
    |   laborum sequi corrupti fuga delectus minus et pariatur accusamus consectetur beatae
    |   veniam?</p>
13 |   <h2>Novela</h2>
14 |   <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Iste ut eius quam a
    |   possimus eligendi dolorem, ex alias similique illo neque, quos odit dolorum
    |   mollitia, vero debitis deserunt repellat qui!</p>
15 </body>
16 </html>

```

Hemos agregado lo que está enmarcado.



```

<> titulos-y-parrafos.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  |   <title>Títulos y párrafos</title>
8  </head>
9  <body>
10 |   <h1>Harry Potter</h1>

```

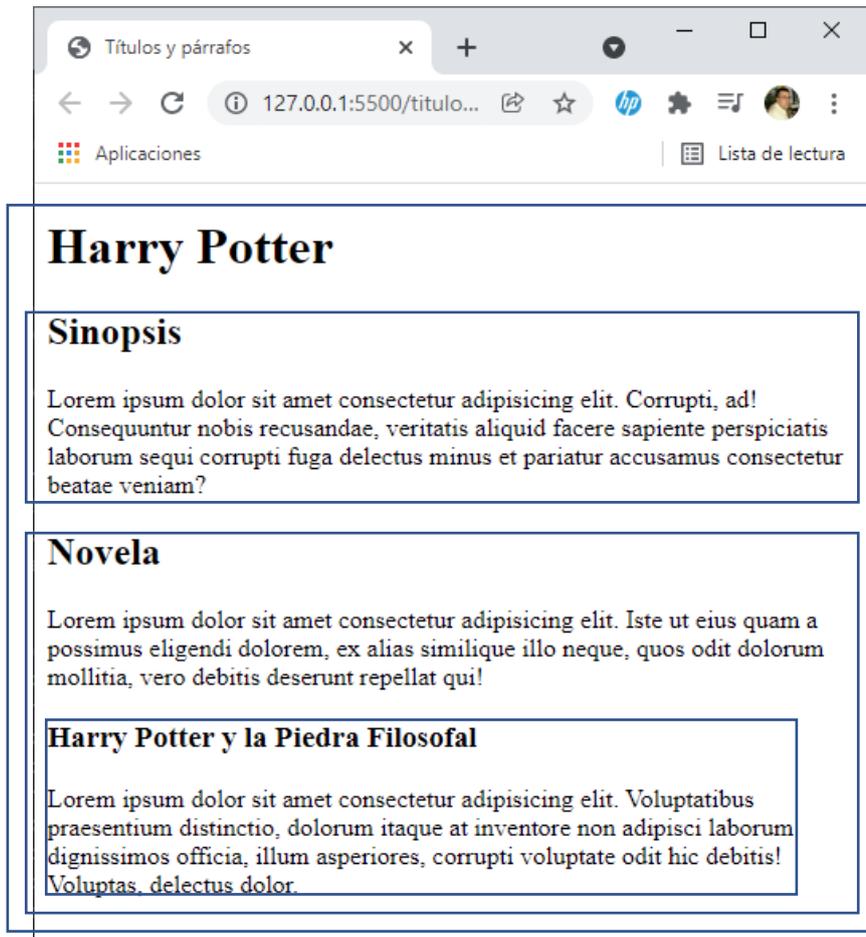
#YOAPRENDOCONDORIAN

```

11 <h2>Sinopsis</h2>
12 <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Corrupti, ad! Consequuntur
nobis recusandae, veritatis aliquid facere sapiente perspiciatis laborum sequi corrupti
fuga delectus minus et pariatur accusamus consectetur beatae veniam?</p>
13 <h2>Novela</h2>
14 <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Iste ut eius quam a possimus
eligendi dolorem, ex alias similique illo neque, quos odit dolorum mollitia, vero
debitis deserunt repellat qui!</p>
15 <h3>Harry Potter y la Piedra Filosofal</h3>
16 <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Voluptatibus praesentium
distinctio, dolorum itaque at inventore non adipisci laborum dignissimos officia, illum
asperiores, corrupti voluptate odit hic debitis! Voluptas, delectus dolor.</p>
17 </body>
18 </html>

```

Hemos añadido lo que está enmarcado.



Los `<h2>` pertenecen al `<h1>` y el `<h3>` pertenece al `<h2>`.

Header, Main y Footer

Etiquetas de sección de contenido.

Vamos a crear un nuevo documento sección-de-contenido.html y copiaremos el código del ejercicio anterior.

```

<!DOCTYPE html>
<html lang="en">
<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Títulos y párrafos</title>

</head>
<body>

  <!-- <header></header> <main></main> <footer></footer> -->

  <header>
    <!--menú de navegación, logo, redes sociales... -->
  </header>

  <main>

    <h1>Harry Potter</h1>
    <h2>Sinopsis</h2>

    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Corrupti, ad! Consequuntur nobis recusandae, veritatis aliquid facere sapiente perspiciatis laborum sequi corrupti fuga delectus minus et pariatur accusamus consectetur beatae veniam?</p>

    <h2>Novela</h2>

    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Iste ut eius quam a possimus eligendi dolorem, ex alias similique illo neque, quos odit dolorum mollitia, vero debitis deserunt repellat qui!</p>

    <h3>Harry Potter y la Piedra Filosofal</h3>

    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Voluptatibus praesentium distinctio, dolorum itaque at inventore non adipisci laborum dignissimos officia, illum asperiores, corrupti voluptate odit hic debitis! Voluptas, delectus dolor.</p>

  </main>

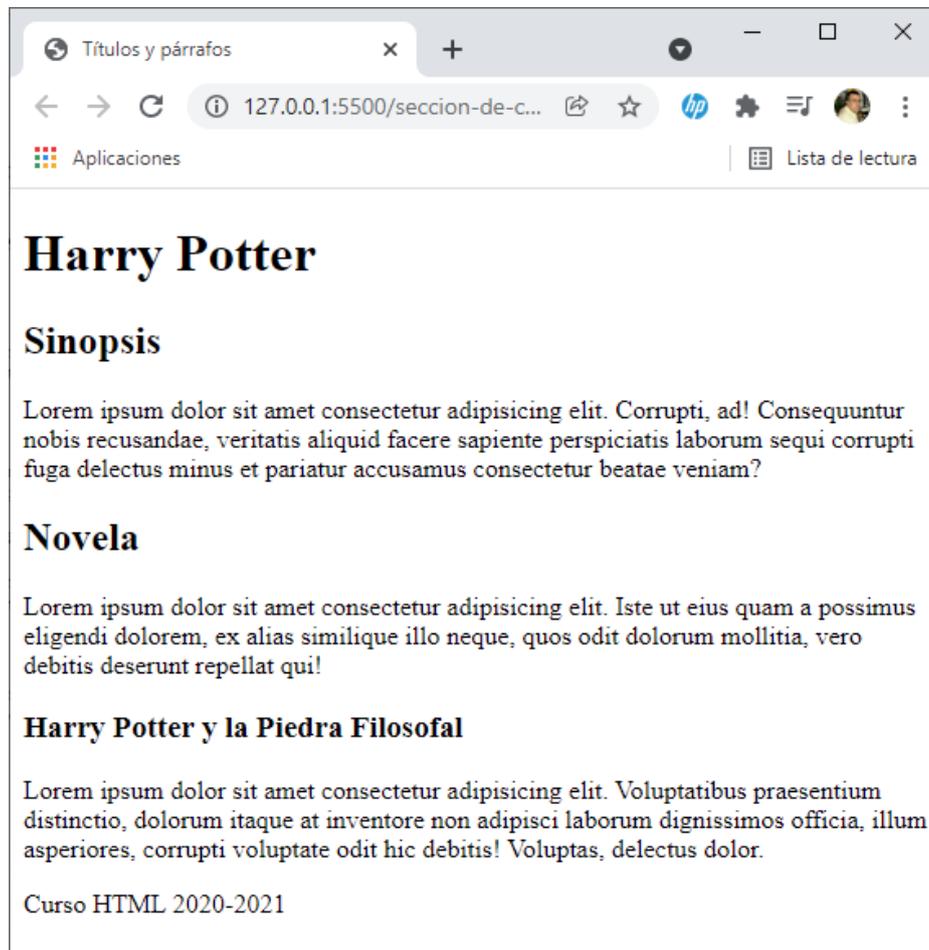
  <footer>
    Curso HTML 2020-2021
  </footer>

</body>

```

</html>

Este será el resultado:



Vamos a agregar contenido en <header></header>

```

11 <header>
12   <!--menú de navegación, logo, redes sociales... -->
13   Inicio
14   Novelas
15   Peículas
16   Contacto
17 </header>

```

Este será el resultado:



Section vs article I

section: es un contenedor genérico que agrupa contenido que está relacionado. Cuando creamos bloques cuyo contenido es parte de un bloque total usaremos section.

article: es un contenedor que representa contenido independiente, es decir, podemos leer ese fragmento en cualquier otro sitio y tendría sentido por sí mismo.

aside: se utiliza para representar contenido relacionado pero que forma parte del contenido principal.

Anidamiento: Un section puede contener articles, por ejemplo, si tenemos varios artículos que hablan sobre etiquetas HTML, deben ir dentro de un section, ya que es contenido relacionado entre sí, y los article serían contenido independiente porque podríamos leer uno sin haber leído el resto, y seguiría teniendo sentido.

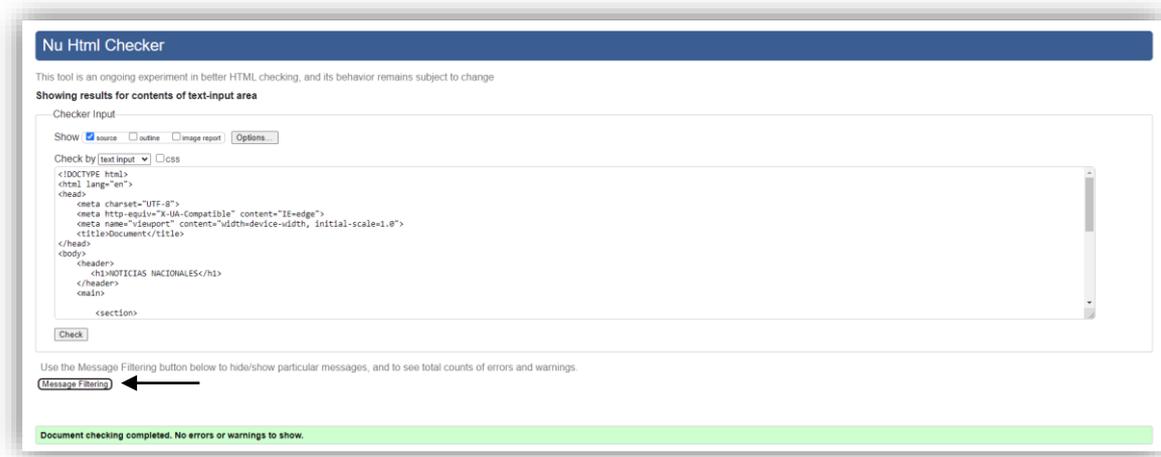
El article es definido como un componente de la página de contenido independiente, esto implica que esta etiqueta pueda tener un header y un footer propios.

También existe el caso en el que un article contenga varias secciones, el artículo independiente podría ser navegadores y este contener dentro secciones como navegadores más utilizados en 2020.

Section vs article II

Este ejemplo es un section con varios articles.

Hay un enlace donde podemos introducir el código html y este nos dirá si hay errores o está bien, <https://validator.w3.org/nu/#textarea>



Copiamos el texto lo pegamos en esta caja de texto y seleccionamos el botón.

En este caso el mensaje es: "Comprobación de documentos completada. No hay errores ni advertencias que mostrar".

```
<!DOCTYPE html>
```

```
<html lang="es">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Document</title>
```

```
</head>
```

```
<body>
```

```
  <header>
```

```
    <h1>NOTICIAS NACIONALES</h1>
```

```
  </header>
```

```
  <main>
```

```
    <section>
```

```
      <h2>Noticias del día</h2>
```

```
      <article>
```

```
        <h3>Noticia 1</h3>
```

```
        <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Quae ipsa fuga dicta illum unde placeat neque, non natus deserunt eligendi commodi, sit eaque expedita? Reprehenderit esse ea repellat veniam ex. </p>
```

```
      </article>
```

```
    <h3>Noticia 2</h3>
```

<p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Consectetur sequi facilis voluptas optio reiciendis quas, distinctio veniam culpa saepe et autem ea velit, vitae illo pariatur voluptatum obcaecati cupiditate? Officiis?</p>

<article>

<h3>Noticia 3</h3>

<p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Eaque rerum ut ipsam numquam recusandae voluptates possimus perferendis exercitationem voluptatum dolor nisi porro doloremque itaque, corporis cum odit esse corrupti at.</p>

</article>

</section>

</main>

<footer>4-11-2020</footer>

</body>

</html>



Section vs article III

```

<? seccion-de-contenido-ll.html > ...
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Document</title>
8   </head>
9   <body>
10    <article>
11      <h2>Navegadores más utilizados en 2020</h2>
12      <section>
13        <header><h3>Chrome</h3></header>
14        <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Nesciunt reprehenderit officia inventore sint
15        quisquam ratione minus aliquam provident optio eveniet iure, commodi nostrum eligendi! Perferendis commodi
16        nihil accusantium quisquam quo?</p>
17        <footer>Creado por google</footer>
18      </section>
19      <section>
20        <header><h3>Firefox</h3></header>
21        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Beatae suscipit error magnam amet culpa dolore
22        velit totam id fugiat neque nisi possimus ipsa exercitationem hic fuga, quas odio sequi pariatur.</p>
23        <footer></footer>
24      </section>
25    </article>
26  </body>
27 </html>

```

Este ejemplo también está correcto.

Aside

```
<> seccion-de-contenido-II.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8  </head>
9  <body>
10 <header>
11     <h1>Cursos de desarrollo web gratuitos</h1>
12 </header>
13 <aside>Visita mi canal de YouTube</aside>
14 <main>
15 </main>
16 <footer>4-11-2020</footer>
17 </body>
18 </html>
```

Como esta etiqueta hace referencia a un texto que no está relacionado, este puede estar tanto fuera como dentro de la etiqueta main.

Elementos de bloque y elementos de línea

Los elementos de bloque van a ocupar todo el ancho disponible aunque su contenido no lo haga, por lo que los elementos que pongamos a continuación saltarán a la siguiente línea.

```
<> elemetos-de-bloque.html > ...
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8  </head>
9  <body>
10     <h1>Soy un elemento de bloque</h1>
11     <p>Soy otro elemento de bloque</p>
12 </body>
13 </html>
```

Este será el resultado:



Cada elemento de bloque ocupa una línea.

Práctica de sección de contenido

Inicio Personajes Comics Solicitar Información

Los Vengadores (The Avengers)

Introducción

Los Vengadores nacen en 1963 de la mente de Stan Lee y Jack Kirby, como respuesta a la famosa Liga de la Justicia de DC cómic que, por aquel entonces, acaparaba todos los números uno en los rankings de ventas. Aunque hay que destacar que, a diferencia de La Liga de la Justicia, donde sus integrantes eran lo más de lo más de DC (Superman, Batman, Wonder Woman...), Los Vengadores que Stan y Jack escogieron para su colección no eran superestrellas de la editorial.

No en vano se trataba del grupo más dispar que se había formado hasta la fecha, con Iron Man (que debía sus poderes a la tecnología), Thor (una deidad asgardiana), Hulk (una mole de poder y furia incontrolable), El Hombre Hormiga (científico transformado en héroe) y La Avispa (pareja del Hombre Hormiga que obtiene sus poderes de su ciencia). Por aquel entonces, Los Vengadores era una apuesta muy arriesgada con héroes muy raros para la época y tachados de solitarios, obligados por las circunstancias a trabajar en equipo.

Un poco de historia

1960

Cuando el dios asgardiano Loki busca venganza contra su hermano Thor, sus maquinaciones involuntariamente llevan al adolescente Rick Jones a recoger Ant-Man, Avispa y Iron Man para ayudar a Thor y Hulk, a quien Loki usó como peón. Después de que el grupo derrotó a Loki, Ant-Man declaró que los cinco trabajaron bien juntos y sugirieron que formaran un equipo; Avispa llamó al grupo, Vengadores

1970

El equipo encontró nuevos personajes como Arkon en el número (abril de 1970), y Red Wolf en el número 80 (septiembre de 1970). Las aventuras del equipo aumentaron en alcance cuando el equipo cruzó a una dimensión alternativa y luchó contra el Escuadrón Supremo, y luchó en la Guerra Kree-Skrull, una batalla épica entre las razas alienígenas Kree y Skrull y el protagonista invitado Kree, el Capitán Marvel. Los Vengadores se disolvieron brevemente cuando los Skrulls se hacen pasar por el Capitán América, Thor y Iron Man usaron su autoridad como fundadores del equipo y lo disolvieron. Los verdaderos Vengadores fundadores, menos Avispa, reformaron el equipo en respuesta a las quejas de Jarvis.

1980

El primer gran avance fue el colapso de Henry Pym, con sus frecuentes cambios de vestimenta y nombre siendo sintomático de un problema de identidad y un complejo de inferioridad. Después de abusar de su esposa, no haber recuperado la confianza de los Vengadores con una estratagema y haber sido engañado por el villano Egghead, Pym fue encarcelado. Pym luego burló a Egghead y derrotó a la última encarnación de los Maestros del Mal por su cuenta, y

demonstró su inocencia. Pym se reconcilió con Avispa, pero decidieron permanecer separados. Pym se retiró de los súper héroes, pero regresó años más tarde.

1990

El gobierno de los Estados Unidos revocó la carta del Estado de Nueva York de los Vengadores en un tratado con la Unión Soviética. Los Vengadores luego recibieron un estatuto de las Naciones Unidas y los Vengadores se dividieron en dos equipos con un equipo suplente de reserva que respaldaba a los equipos principales.

2000

Los Vengadores recibieron la autoridad internacional de las Naciones Unidas. Los miembros que se unieron durante ese período incluyeron Jack de Corazones y el segundo Ant-Man. Un nuevo Capitán Bretaña fue agregado al equipo. La historia de "Vengadores Desunidos" siguió. Titulado Chaos, la historia presentaba la muerte de algunos miembros y la pérdida de credibilidad del equipo. Se revela que la culpable es la Bruja Escarlata, que se había vuelto loca después de agonizar por el recuerdo de sus hijos perdidos y que posteriormente perdió el control de sus poderes que alteran la realidad. Con el equipo en desorden y la mansión de los vengadores arruinada, los miembros sobrevivientes acordaron disolverse.

2010

Después de que los Vengadores Oscuros de Osborn son expuestos como criminales y su ataque contra Asgard fue frustrado, la próxima iteración de la lista de los Vengadores consiste en Thor, Hawkeye, Spider-Man, Wolverine, Capitán América, Spider-Woman, Iron Man y la líder del equipo Maria Hill. Steve Rogers, evitando brevemente su personaje de Capitán América, responde a las preocupaciones de Luke Cage acerca de que el equipo vuelva a los viejos métodos al otorgar el reconocimiento de "Nuevos Vengadores" a Cage como un equipo oficial independiente de los Vengadores más tradicionales de Stark. Bucky Barnes como el Capitán América se unió a los principales Vengadores, mientras que Iron Fist, Power Woman y la Mole se unieron al equipo de Cage, Spider-Man y Wolverine manteniendo la membresía doble en ambos equipos. Rogers era una presencia ocasional y Victoria Hand se añadió como un enlace del gobierno para los Nuevos Vengadores con el respaldo de Rogers.

¿Cómo se unieron los héroes más poderosos de la Tierra? Pues en el primer número del cómic de Los Vengadores, se nos narra cómo Loki, hermano de Thor de insaciable ambición, tiende una trampa a Hulk para que le consideren culpable de un accidente de tren (aunque en realidad consigue salvar a todos los viajeros). Tras esto, Loki envía un mensaje para que lo intercepte su hermano y, así, tenderle una emboscada. Por suerte para Thor, el mensaje también llega a oídos de El Hombre Hormiga, El Hombre de Hierro y La Avista. Juntos, descubren que todo ha sido una encerrona y acaban con los planes de Loki. Así nacieron Los Vengadores.

De esta forma, se considera que la formación original de Los Vengadores está compuesta por Hulk, Thor, Iron Man, La Avispa y El Hombre Hormiga. Siendo en el cómic número cuatro el primero donde aparece ya El Capitán América. De esta forma se completa así el tridente clásico de Los Vengadores: Iron Man, Thor y El Capitán América.

Un grupo en constante cambio Uno de los principales atractivos de la formación, en los cómics claro, es la incesante marea de superhéroes que pasan por Los Vengadores. No en vano, hasta la fecha, son más de 75 los superhéroes que han formado parte del grupo y que podemos dividir en tres categorías: miembros completos (los que acuden a todas las misiones), reservas

(que van cuando algún superhéroe no puede) y honorarios (que nunca formaron parte oficial de Los Vengadores pero se les otorga este cargo como reconocimiento a alguna ayuda prestada).

Marvel's Avengers

Resultado:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <article>
    <h1>Inicio Personajes Comics Solicitar Información </h1>
    <h2>Los Vengadores (The Avengers) </h2>
    <section>
      <header>
        <h3>Introducción </h3>
      </header>
      <p>Los Vengadores nacen en 1963 de la mente de Stan Lee y Jack Kirby, como respuesta a la famosa Liga de la Justicia de DC cómic que, por aquel entonces, acaparaba todos los números uno en los rankings de ventas. Aunque hay que destacar que, a diferencia de La Liga de la Justicia, donde sus integrantes eran lo más de lo más de DC (Superman, Batman, Wonder Woman...), Los Vengadores que Stan y Jack escogieron para su colección no eran superestrellas de la editorial.</p>
      <p>No en vano se trataba del grupo más dispar que se había formado hasta la fecha, con Iron Man (que debía sus poderes a la tecnología), Thor (una deidad asgariana), Hulk (una mole de poder y furia incontrolable), El Hombre Hormiga (científico transformado en héroe) y La Avispa (pareja del Hombre Hormiga que obtiene sus poderes de su ciencia). Por aquel entonces, Los Vengadores era una apuesta muy arriesgada con héroes muy raros para la época y tachados de solitarios, obligados por las circunstancias a trabajar en equipo.</p>
    </section>
    <header>
      <h3>Un poco de historia</h3>
    </header>
```

<section>

<header>

<h4>1960</h4>

</header>

<p>Cuando el dios asgardiano Loki busca venganza contra su hermano Thor, sus maquinaciones involuntariamente llevan al adolescente Rick Jones a recoger Ant-Man, Avispa y Iron Man para ayudar a Thor y Hulk, a quien Loki usó como peón. Después de que el grupo derrotó a Loki, Ant-Man declaró que los cinco trabajaron bien juntos y sugirieron que formaran un equipo; Avispa llamó al grupo, Vengadores</p>

</section>

<section>

<header>

<h4>1970</h4>

</header>

<p>El equipo encontró nuevos personajes como Arkon en el número (abril de 1970), y Red Wolf en el número 80 (septiembre de 1970). Las aventuras del equipo aumentaron en alcance cuando el equipo cruzó a una dimensión alternativa y luchó contra el Escuadrón Supremo, y luchó en la Guerra Kree-Skrull, una batalla épica entre las razas alienígenas Kree y Skrull y el protagonista invitado Kree, el Capitán Marvel. Los Vengadores se disolvieron brevemente cuando los Skrulls se hacen pasar por el Capitán América, Thor y Iron Man usaron su autoridad como fundadores del equipo y lo disolvieron. Los verdaderos Vengadores fundadores, menos Avispa, reformaron el equipo en respuesta a las quejas de Jarvis.</p>

</section>

<section>

<header>

<h4>1980</h4>

</header>

<p>El primer gran avance fue el colapso de Henry Pym, con sus frecuentes cambios de vestimenta y nombre siendo sintomático de un problema de identidad y un complejo de inferioridad. Después de abusar de su esposa, no haber recuperado la confianza de los Vengadores con una estratagema y haber sido engañado por el villano Egghead, Pym fue encarcelado. Pym luego burló a Egghead y derrotó a la última encarnación de los Maestros del Mal por su cuenta, y demostró su inocencia. Pym se reconcilió con Avispa, pero decidieron permanecer separados. Pym se retiró de los súper héroes, pero regresó años más tarde.</p>

</section>

<section>

<header>

<h4>1990</h4>

</header>

<p>El gobierno de los Estados Unidos revocó la carta del Estado de Nueva York de los Vengadores en un tratado con la Unión Soviética. Los Vengadores luego recibieron un estatuto de las Naciones Unidas y los Vengadores se dividieron en dos equipos con un equipo suplente de reserva que respaldaba a los equipos principales.</p>

</section>

<section>

<header>

<h4>2000</h4>

</header>

<p>Los Vengadores recibieron la autoridad internacional de las Naciones Unidas. Los miembros que se unieron durante ese período incluyeron Jack de Corazones y el segundo Ant-Man. Un nuevo Capitán Bretaña fue agregado al equipo. La historia de “Vengadores Desunidos” siguió. Titulado Chaos, la historia presentaba la muerte de algunos miembros y la pérdida de credibilidad del equipo. Se revela que la culpable es la Bruja Escarlata, que se había vuelto loca después de agonizar por el recuerdo de sus hijos perdidos y que posteriormente perdió el control de sus poderes que alteran la realidad. Con el equipo en desorden y la mansión de los vengadores arruinada, los miembros sobrevivientes acordaron disolverse.</p>

</section>

<section>

<header>

<h4>2010</h4>

</header>

<p>Después de que los Vengadores Oscuros de Osborn son expuestos como criminales y su ataque contra Asgard fue frustrado, la próxima iteración de la lista de los Vengadores consiste en Thor, Hawkeye, Spider-Man, Wolverine, Capitán América, Spider-Woman, Iron Man y la líder del equipo Maria Hill. Steve Rogers, evitando brevemente su personaje de Capitán América, responde a las preocupaciones de Luke Cage acerca de que el equipo vuelva a los viejos métodos al otorgar el reconocimiento de “Nuevos Vengadores” a Cage como un equipo oficial independiente de los Vengadores más tradicionales de Stark. Bucky Barnes como el Capitán América se unió a los principales Vengadores, mientras que Iron Fist, Power Woman y la Mole se unieron al equipo de Cage, Spider-Man y Wolverine manteniendo la membresía doble en ambos equipos. Rogers era una presencia ocasional y Victoria Hand se añadió como un enlace del gobierno para los Nuevos Vengadores con el respaldo de Rogers.</p>

<p>¿Cómo se unieron los héroes más poderosos de la Tierra? Pues en el primer número del cómic de Los Vengadores, se nos narra cómo Loki, hermano de Thor de insaciable ambición, tiende una trampa a Hulk para que le consideren culpable de un accidente de tren (aunque en realidad consigue salvar a todos los viajeros). Tras esto, Loki envía un mensaje para que lo intercepte su hermano y, así, tenderle una emboscada. Por suerte para Thor, el mensaje también llega a oídos de El Hombre Hormiga, El Hombre de Hierro y La Avista. Juntos,

#YOAPRENDOCONDORIAN

descubren que todo ha sido una encerrona y acaban con los planes de Loki. Así nacieron Los Vengadores.

De esta forma, se considera que la formación original de Los Vengadores está compuesta por Hulk, Thor, Iron Man, La Avispa y El Hombre Hormiga. Siendo en el cómic número cuatro el primero donde aparece ya El Capitán América. De esta forma se completa así el tridente clásico de Los Vengadores: Iron Man, Thor y El Capitán América.

Un grupo en constante cambio Uno de los principales atractivos de la formación, en los cómics claro, es la incesante marea de superhéroes que pasan por Los Vengadores. No en vano, hasta la fecha, son más de 75 los superhéroes que han formado parte del grupo y que podemos dividir en tres categorías: miembros completos (los que acuden a todas las misiones), reservas (que van cuando algún superhéroe no puede) y honorarios (que nunca formaron parte oficial de Los Vengadores pero se les otorga este cargo como reconocimiento a alguna ayuda prestada).

</section>

</section>

<footer>Marvel's Avengers</footer>

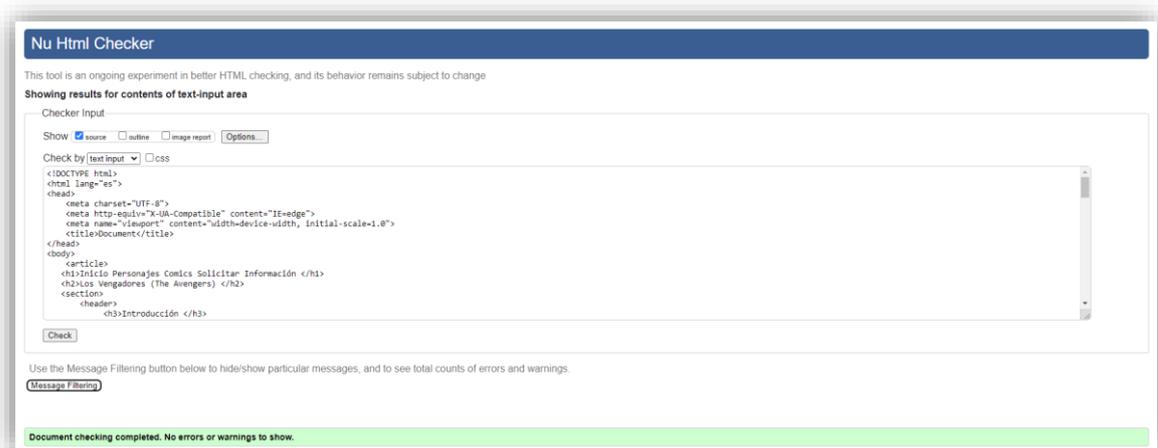
</section>

</article>

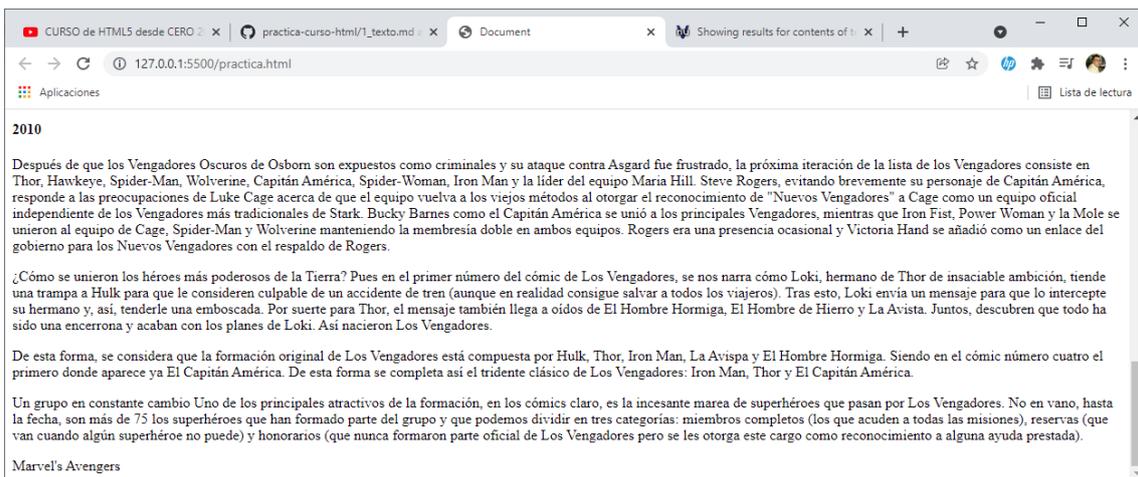
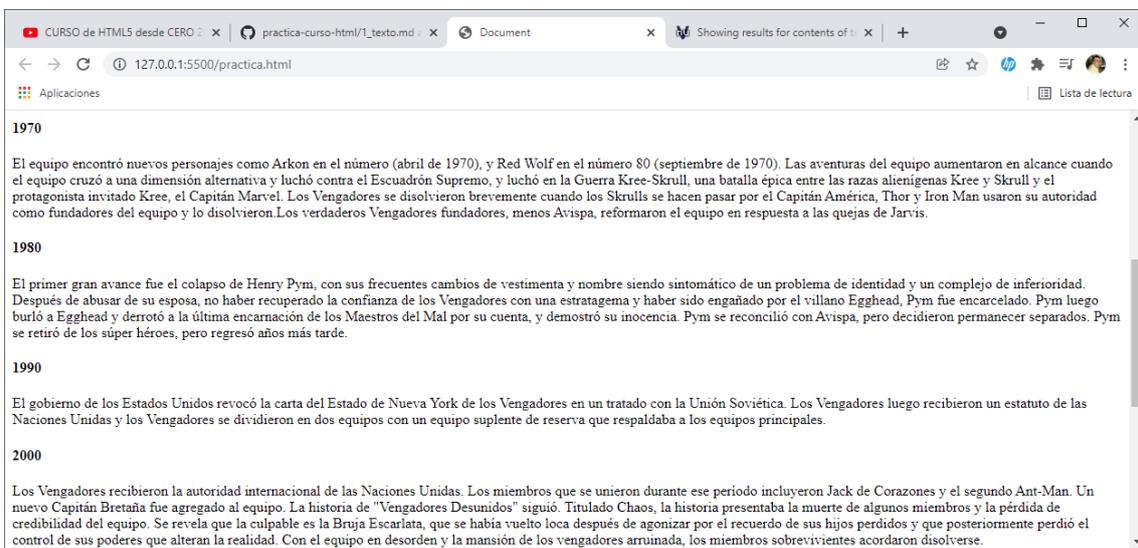
</body>

</html>

Comprobación:



Resultado final:



Elementos de línea

em -> emphasis

strong -> más énfasis

small -> Menos énfasis que el texto normal

#YOAPRENDOCONDORIAN

br -> Forzar salto de línea

wbr -> Salto de línea si hiciera falta

time -> Se usa para representar un contenido de hora/fecha

i -> italic

b -> bold

u -> underline

sup -> Superíndice

sub -> Subíndice

``:

`<>` elementos-de-linea.html `>` ...

```

1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  |   <title>Document</title>
8  </head>
9  <body>
10 |   <h1>Elementos de línea</h1>
11 |   <p><em>El dinero</em> es importante pero la salud es lo más importante</p>
12 </body>
13 </html>

```



Con esta etiqueta damos más importancia a parte del texto, en este caso "El dinero".

``:

```
<> elementos-de-línea.html > ...
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10  <h1>Elementos de línea</h1>
11  <p><em>El dinero</em> es importante pero <strong>la salud</strong> es lo más importante</p>
12 </body>
13 </html>
```



<small></small>:

```
<> elementos-de-línea.html > ...
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10  <h1>Elementos de línea</h1>
11  <p><em>El dinero</em> es importante pero <strong>la salud</strong> es lo más importante. <small>Saludos</small></p>
12 </body>
13 </html>
```



Es el texto con menos relevancia de todo el párrafo.

:

```

<> elementos-de-línea.html > ...
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10  <h1>Elementos de línea</h1>
11  <p><em>El dinero</em> es importante pero <strong>la salud</strong> es lo más importante. <small>Saludos!</small></p>
12
13  <p>Lorem ipsum dolor sit, amet<br>consectetur adipisicing elit. Eaque, nostrum necessitatibus. Quam dolores praesentium
    aperiam, esse autem dolorum, modi atque eligendi beatae vero laborum ratione quod ad provident cupiditate sed!</p>
14 </body>
15 </html>

```



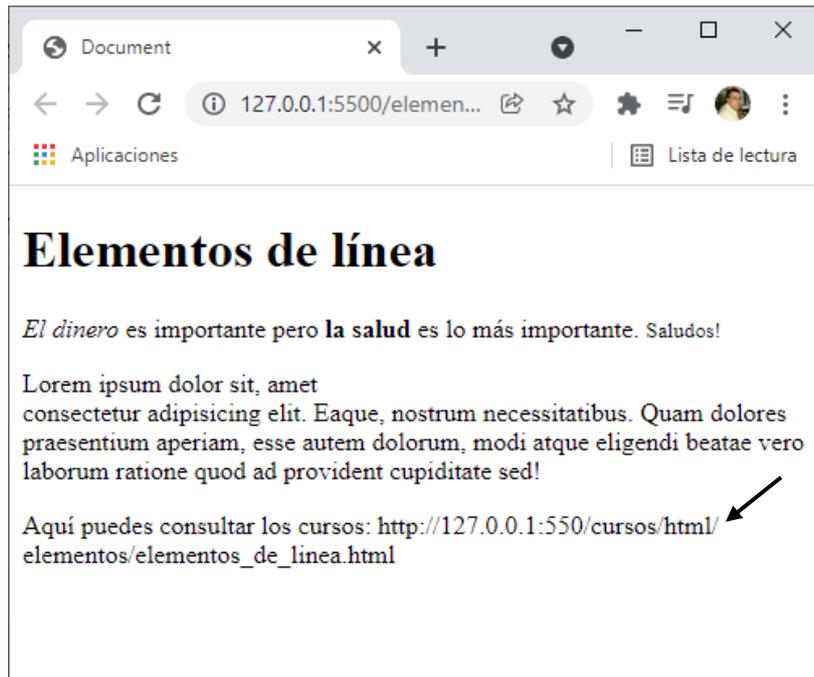
Provoca un salto de línea, no utilizar la etiqueta
 para separar texto, para esta el CSS.

<wbr>: le dice al navegador por las partes del texto donde quiero que corte el párrafo, es igual que si ponemos un – el navegador entiende que por esta parte puede partir el párrafo.

```

<> elementos-de-línea.html > ...
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10  <h1>Elementos de línea</h1>
11  <p><em>El dinero</em> es importante pero <strong>la salud</strong> es lo más importante. <small>Saludos!</small></p>
12
13  <p>Lorem ipsum dolor sit, amet<br>consectetur adipisicing elit. Eaque, nostrum necessitatibus. Quam dolores praesentium
    aperiam, esse autem dolorum, modi atque eligendi beatae vero laborum ratione quod ad provident cupiditate sed!</p>
14
15  <p>Aquí puedes consultar los cursos: <a href="http://127.0.0.1:550/cursos/html/elementos/elementos_de_linea.html">http://127.0.0.1:550/cursos/html/elementos/elementos_de_linea.html</a><br></p>
16 </body>
17 </html>

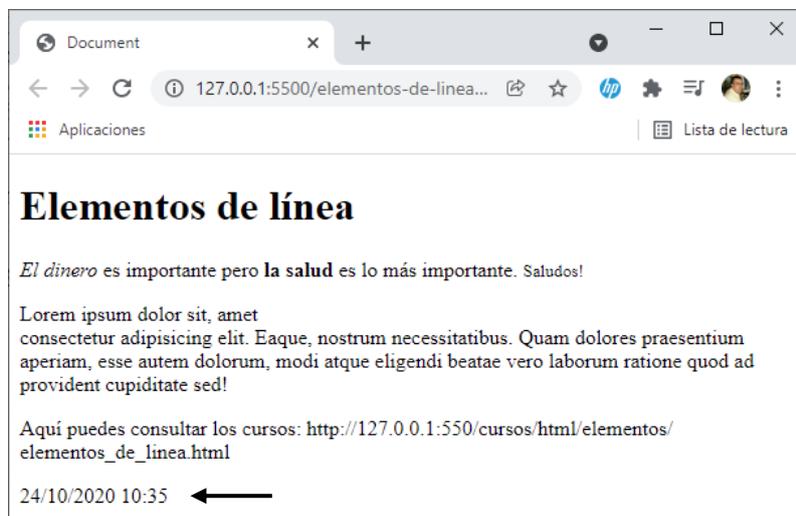
```



Elementos de línea II

<time>:

```
<> elementos-de-linea.html > ...
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10  <h1>Elementos de línea</h1>
11  <p><em>El dinero</em> es importante pero <strong>la salud</strong> es lo más importante. <small>Saludos!</small></p>
12
13  <p>Lorem ipsum dolor sit, amet<br>consectetur adipisicing elit. Ea que, nostrum necessitatibus. Quam dolores praesentium
14  aperiam, esse autem dolorum, modi atque eligendi beatae vero laborum ratione quod ad provident cupiditate sed!</p>
15
16  <p>Aquí puedes consultar los cursos: http://127.0.0.1:5500/cursos/html/elementos/elementos\_de\_linea.html</p>
17  <time>24/10/2020 10:35</time>
18 </body>
19 </html>
```

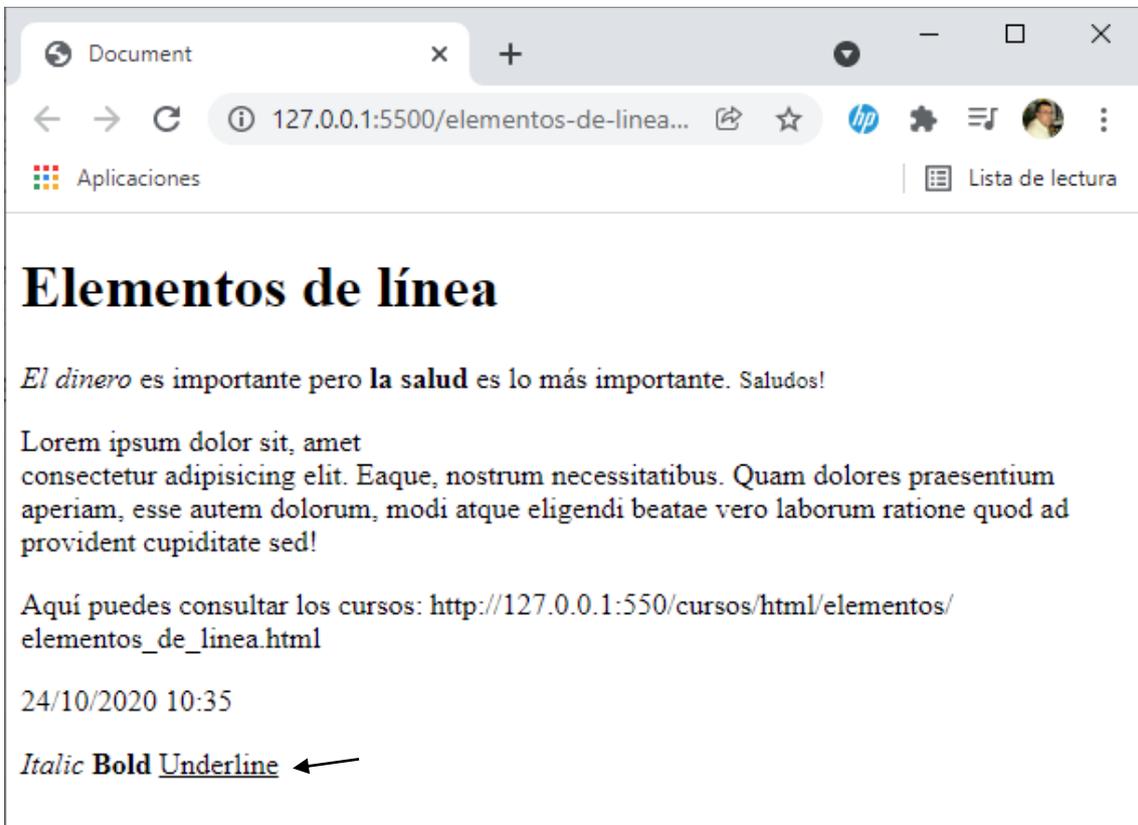


También se utiliza si también solo se pone la fecha o la hora.

```

<> elementos-de-linea.html > ...
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10  <h1>Elementos de línea</h1>
11  <p><em>El dinero</em> es importante pero <strong>la salud</strong> es lo más importante. <small>Saludos!</small></p>
12
13  <p>Lorem ipsum dolor sit, amet<br>consectetur adipisicing elit. Eaque, nostrum necessitatibus. Quam dolores praesentium
14  aperiam, esse autem dolorum, modi atque eligendi beatae vero laborum ratione quod ad provident cupiditate sed!</p>
15
16  <p>Aquí puedes consultar los cursos: <a href="http://127.0.0.1:550/cursos/html/elementos/elementos_de_linea.html">http://127.0.0.1:550/cursos/html/elementos/elementos_de_linea.html</a>.</p>
17
18  <time>24/10/2020 10:35</time>
19  <p>
20    <i>Italic</i>
21    <b>Bold</b>
22    <u>Underline</u>
23  </p>
24 </body>
    </html>

```



Estos estilos se suelen hacer desde CSS.

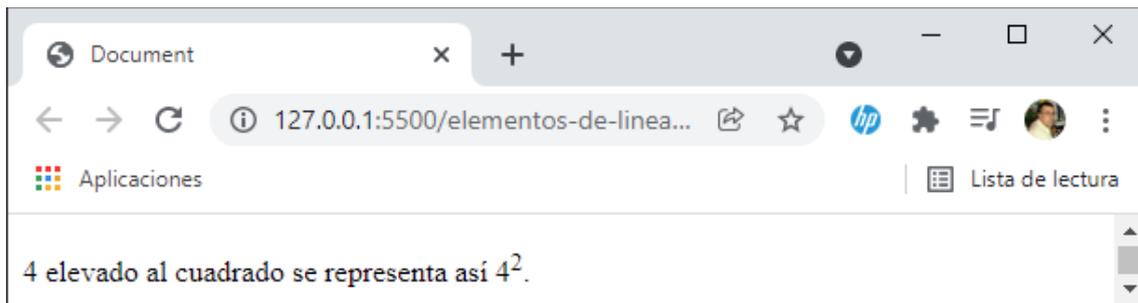
Un caso excepcional el la etiqueta <i></i> y sirve para añadir iconos.

```

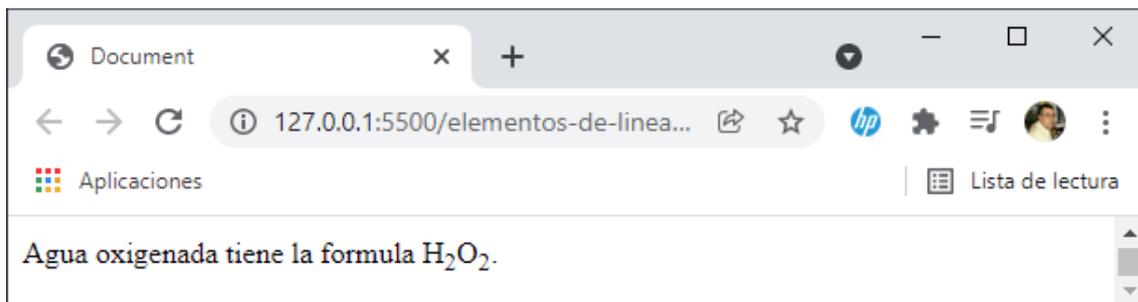
<i class="glyphicon glyphicon-cloud"></i>
<i class="glyphicon glyphicon-remove"></i>
<i class="glyphicon glyphicon-user"></i>
<i class="glyphicon glyphicon-envelope"></i>

```

<p>4 elevado al cuadrado se representa así 4².</p>



<p>Agua oxigenada tiene la formula H₂O₂.</p>



Introducción a los atributos

Atributos: Los atributos son valores adicionales que configuran los elementos y/o ajustan su comportamiento.

En términos generales hay dos tipos de atributos:

- Comunes: Su sintaxis es -> atributo="valor".
- Booleanos: Su sintaxis es -> atributo.

Atributos ya utilizados

```
<html lang="es">
```

Este es un ejemplo de atributo común, le estamos diciendo que esta página se contenido será en español.

```
<meta charset="UTF-8">
```

Es un atributo que declara la codificación de caracteres que contiene la página.

Desde Visual Studio Code podemos acceder a la información de todas las etiquetas colocando el cursor encima de dicha etiqueta.

```

The head element represents a collection of metadata for the Document.
MDN Reference
</head>
<meta http-equiv="X-UA-Compatible" content="IE=edge" initial-scale=1.0">

```

Atributos globales

Algunos atributos globales que están disponibles para la mayoría de etiquetas HTML.

class -> Éste atributo se usa para dar estilos a través de CSS.

id -> Este identificador único que se utiliza para seleccionar el elemento desde JavaScript y para hacer navegación a través de anclas.

title -> Es un atributo que ayuda a la accesibilidad mostrando una descripción del elemento al que pertenece el mensaje en un tooltip.

data-* -> Es un atributo que nos permite guardar algún valor en la etiqueta HTML.

```

<> atribulos-comunes.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8  <style>
9      .parrafo-1{
10         color: red; ←
11     }
12 </style>

```

```

13 </head>
14 <body>
15
16 </body>
17 <p class="parrafo-1">Clases</p>
18 <p>Identificador</p>
19 <p>Título con tooltip</p>
20 <p>Datos pasados en la etiqueta</p>
21 </html>

```

El atributo class permite desde CSS hacer referencia a este párrafo, en este caso para ponerlo en color rojo.



id: Este identificador para hacer referencia a JavaScript y la navegación a través de anclas, es decir además de ir a una determinada página a una zona en concreto.

```

<> atribulos-comunes.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <style>
9     .parrafo-1{
10      color: red;
11    }
12  </style>
13 </head>
14 <body>
15
16 </body>
17 <p class="parrafo-1">Clases</p>
18 <p id="parrafo"> Identificador</p>
19 <p>Título con tooltip</p>
20 <p>Datos pasados en la etiqueta</p>
21

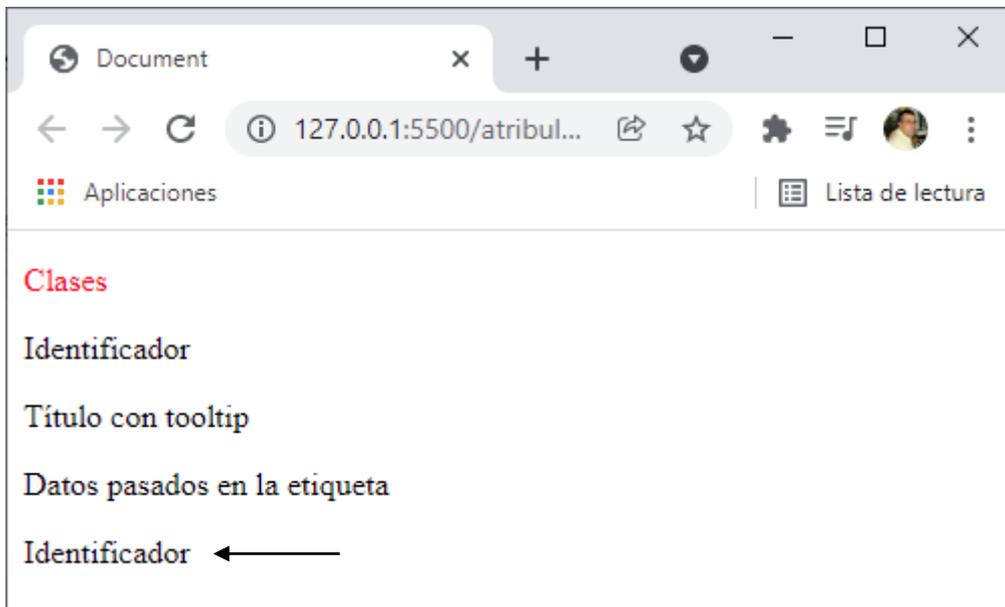
```

```

22 <script>
23   const p = document.getElementById('parrafo')
24   document.write(p.textContent)
25 </script>
26 </html>

```

Desde JavaScript queremos recuperar el contenido del párrafo con el id="párrafo", que a continuación se muestra en el navegador.



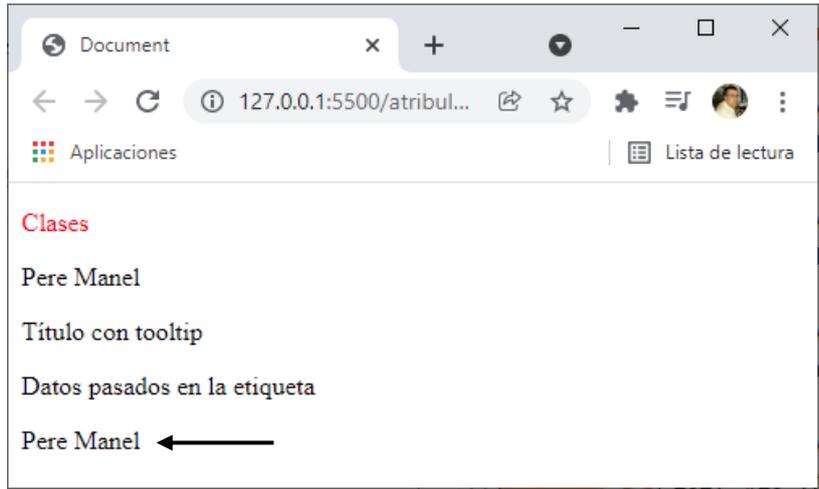
Si cambiamos la frase donde está el id:

```

<> atribulos-comunes.html > html > p#parrafo
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <style>
9     .parrafo-1{
10      color: red;
11    }
12  </style>
13 </head>
14 <body>
15
16 </body>
17 <p class="parrafo-1">Clases</p>
18 <p id="parrafo"> Pere Manel</p>
19 <p>Título con tooltip</p>
20 <p>Datos pasados en la etiqueta</p>
21
22 <script>
23   const p = document.getElementById('parrafo')
24   document.write(p.textContent)
25 </script>
26 </html>
--

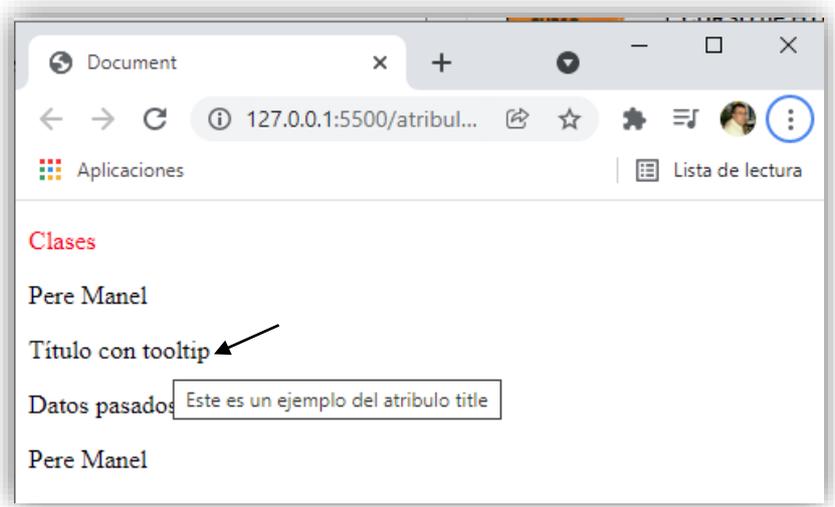
```

Este será el resultado:



El id no se puede duplicar y tiene que ser único.
title:

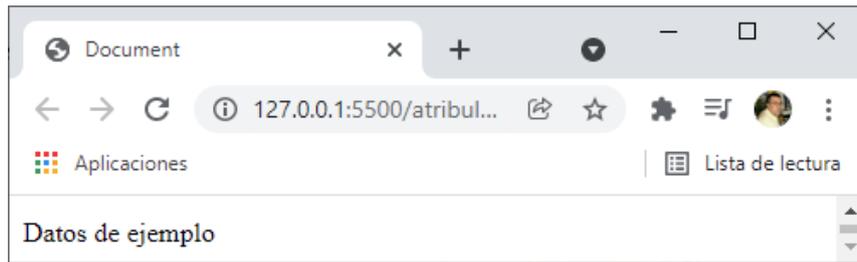
```
<p title="Este es un ejemplo del atribulo title">Título con tooltip</p>
```



Si nos colocamos encima del texto aparece un mensaje de título con tooltip.

```
20 <p id="parrafo-2" data-ejemplo="Datos de ejemplo">Datos pasados en la etiqueta</p>
21
22 <script>
23   const parrafo2 = document.getElementById('parrafo-2')
24   document.write(parrafo2.dataset.ejemplo)
25 </script>
26 </html>
```

Data-ejemplo guarda "Datos de ejemplo" gracias al id="párrafo-2" podemos recuperar la información de data-ejemplo.



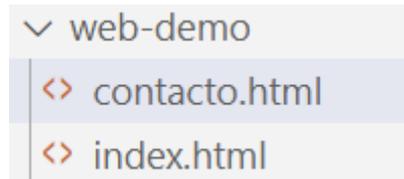
Introducción a enlaces

Enlaces: Conocidos también por links popularmente. Su objetivo es conectar una página web con otra página web,, con un recurso tanto interno como externo, o con otro sitio web.

Tiene el atributo obligatorio href, donde le especificamos la ruta del recuso o sitio que queremos obtener.

También tiene el atributo target, que configura cómo queremos visualizar el recuso o sitio que solicitamos.

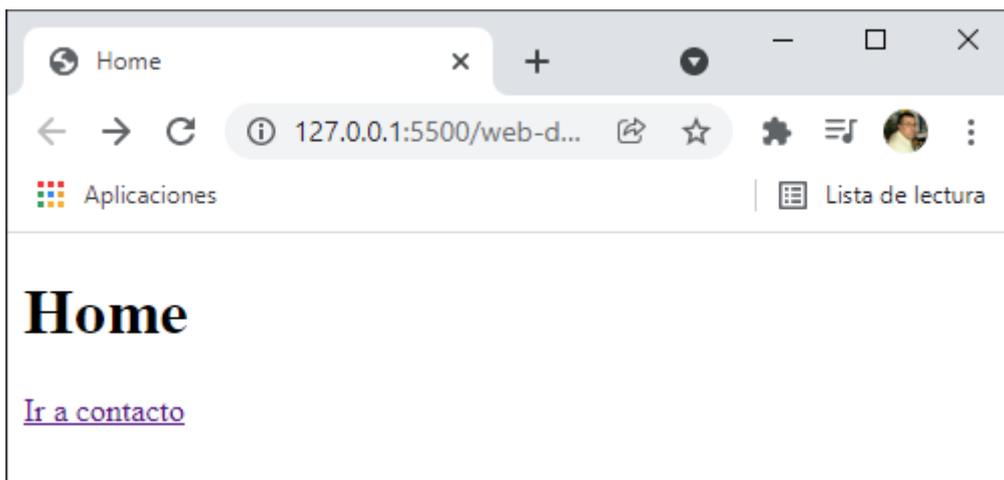
Para este capítulo hemos creado una carpeta con sus correspondientes archivos html.



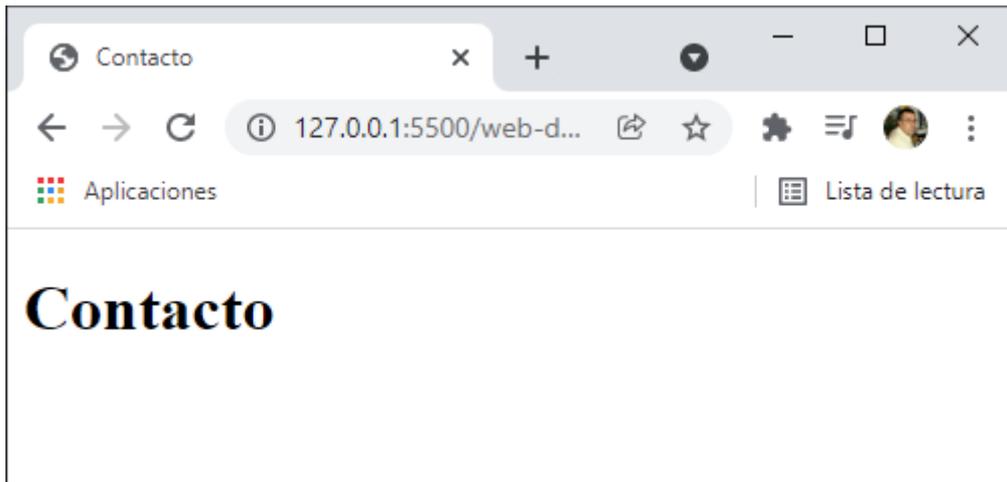
La carpeta se llama web-demo y los archivos index.html y contacto.html.

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Home</title>
8 </head>
9 <body>
10  <h1>Home</h1>
11  <a href="contacto.html">Ir a contacto</a>
12 </body>
13 </html>
```

Este será el resultado:



Si hacemos un clic sobre el texto Ir a contacto este será el resultado:

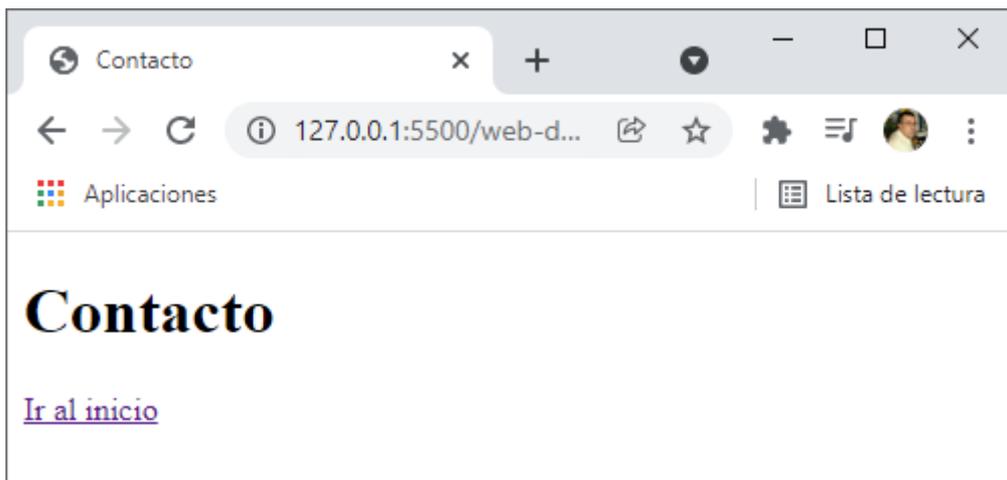


Nos hemos ido a la otra página.

Para poder volver a la otra página tenemos que hacer otro enlace en la página contacto.html.

```
web-demo > <> contacto.html > ...
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  |   <title>Contacto</title>
8  </head>
9  <body>
10 |   <h1>Contacto</h1>
11 |   <a href="index.html">Ir al inicio</a>
12 </body>
13 </html>
```

Este será el resultado:



Ya tenemos el enlace para volver a la página index.html.

Rutas absolutas y relativas

Rutas absolutas:

Tienen un protocolo, http o https y son únicas en la red. Se suelen utilizar para rutas externas.

Rutas relativas:

Pueden ser relativas al punto donde nos encontramos o relativas a la raíz del proyecto.

No usan protocolo.

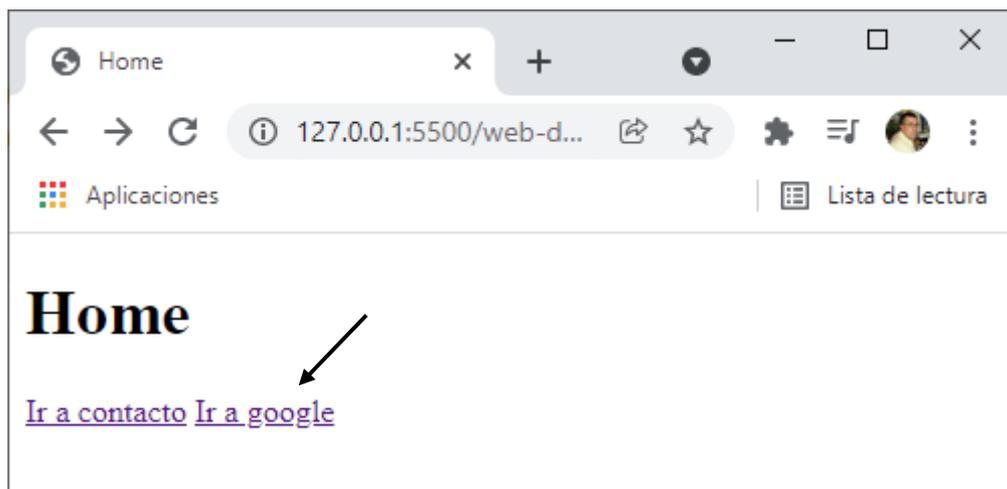
Si el recurso se encuentra al mismo nivel (en la misma carpeta) pondremos únicamente el nombre del archivo.

Si necesitamos salir de la carpeta actual usaremos ../ y si se pone uno por cada nivel (carpeta) de la que queramos salir.

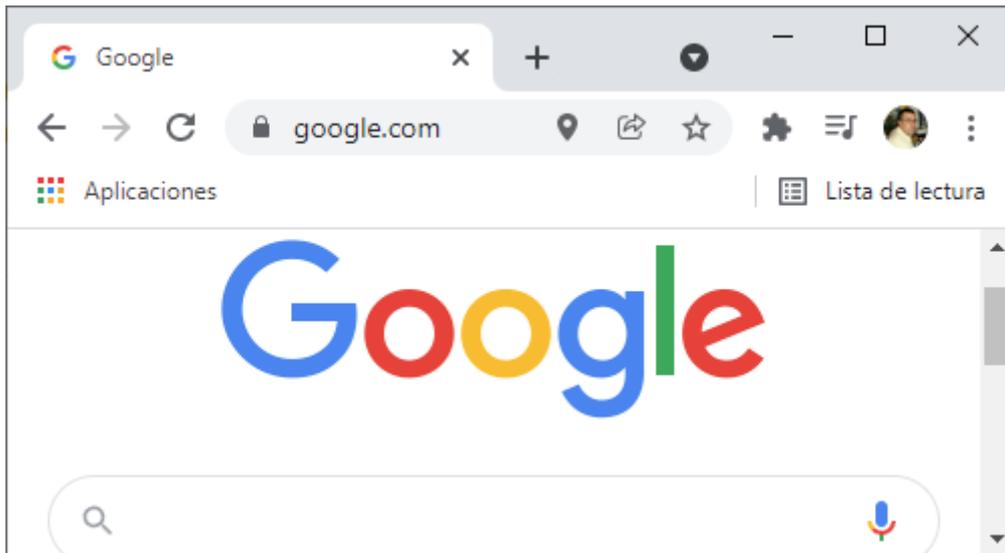
Ruta absoluta:

```
web-demo > <> index.html > ...
 1  <!DOCTYPE html>
 2  <html lang="es">
 3  <head>
 4  |   <meta charset="UTF-8">
 5  |   <meta http-equiv="X-UA-Compatible" content="IE=edge">
 6  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
 7  |   <title>Home</title>
 8  </head>
 9  <body>
10  |   <h1>Home</h1>
11  |   <a href="contacto.html">Ir a contacto</a>
12  |   <a href="https://google.com">Ir a google</a>
13  </body>
14  </html>
```

Este será el resultado:



Si hacemos clic sobre este enlace este será el resultado:



Búsqueda relativa:

```
<a href="contacto.html">Ir a contacto</a>
```

En este caso el documento se encuentra en la misma carpeta donde está el documento donde está el enlace.

```
<a href="web-demo/contacto.html">Ir a contacto</a>
```

En este caso el documento que estamos buscando se encuentra en una carpeta llamada web-demo y el documento que tiene el enlace se encuentra fuera de esta carpeta.

```
<a href="../contacto.html">Ir a contacto</a>
```

En este caso el archivo contacto.html se encuentra fuera de la carpeta donde se encuentra el archivo que contiene el enlace, por cada nivel que tengamos que salir utilizaremos ../

Las rutas relativas a la raíz.

```
<a href="/contacto.html">Ir a contacto</a>
```

Desde el directorio raíz buscamos la página contacto.html.

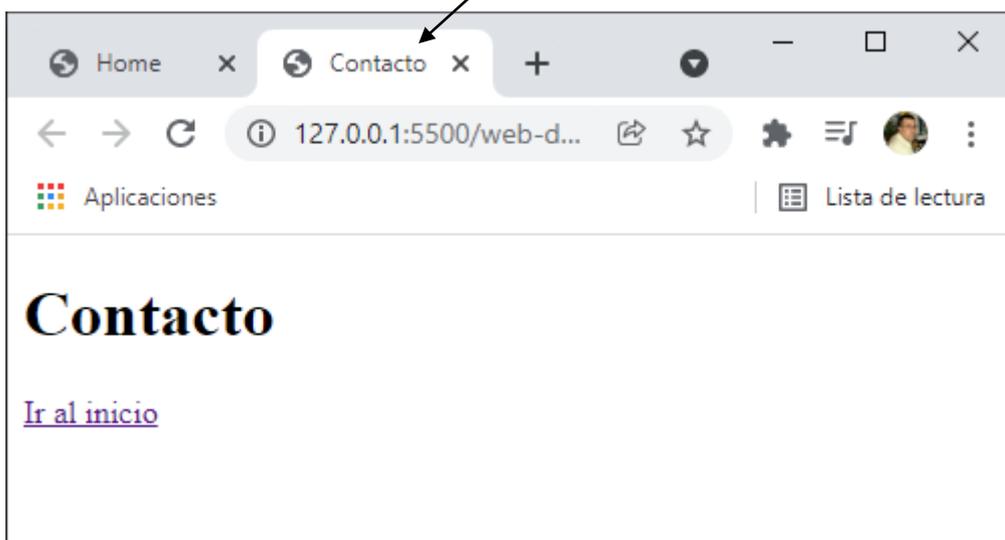
```
web-demo > <> contacto.html > ...
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  |   <title>Contacto</title>
8  </head>
9  <body>
10 |   <h1>Contacto</h1>
11 |   <a href="/">Ir al inicio</a>
12 </body>
13 </html>
```

Al poner solo / entiende que tiene que cargar la página index.html que se encuentra en el directorio raíz. Por esto la importancia de que nuestra página principal se llame index.html.

Atributos de los enlaces

target: define donde se abrirá el recurso solicitado. Por norma general siempre que uséis rutas absolutas pondréis como valor "_blank". "_self" valor por defecto, así que no lo pondremos.

```
web-demo > <> index.html > ...
 1  <!DOCTYPE html>
 2  <html lang="es">
 3  <head>
 4  |   <meta charset="UTF-8">
 5  |   <meta http-equiv="X-UA-Compatible" content="IE=edge">
 6  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
 7  |   <title>Home</title>
 8  </head>
 9  <body>
10  |   <h1>Home</h1>
11  |   <a href="contacto.html" target="_blank">Ir a contacto</a>
12 </body>
13 </html>
```



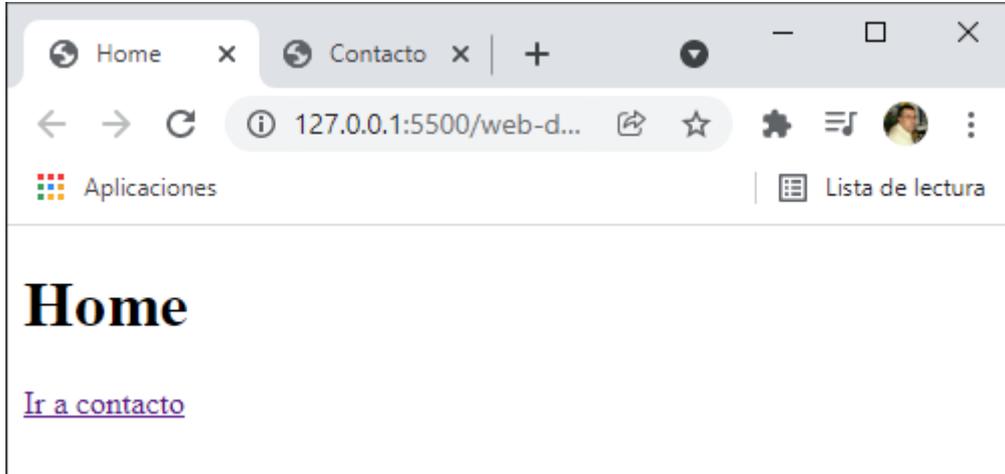
El documento se abre en una pestaña nueva.

```
web-demo > <> index.html > ...
 1  <!DOCTYPE html>
 2  <html lang="es">
 3  <head>
 4  |   <meta charset="UTF-8">
 5  |   <meta http-equiv="X-UA-Compatible" content="IE=edge">
 6  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
 7  |   <title>Home</title>
 8  </head>
 9  <body>
10  |   <h1>Home</h1>
11  |   <a href="contacto.html" download>Ir a contacto</a>
```

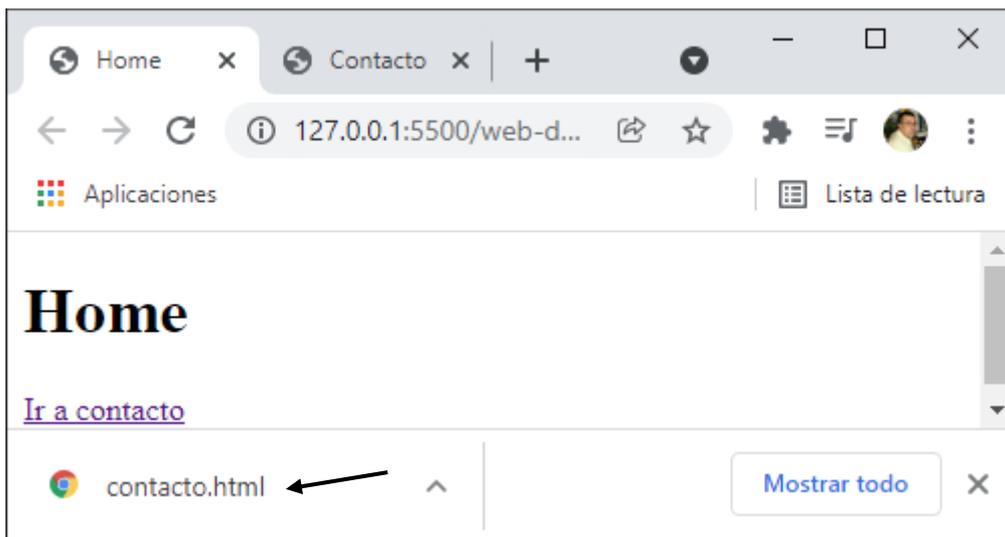
```
12 </body>  
13 </html>
```

download: Atributo booleano, sirve para descargar el recurso solicitado.

IMPORTANTE, el recurso debe estar en tu mismo servidor.



Al hacer clic sobre el enlace.



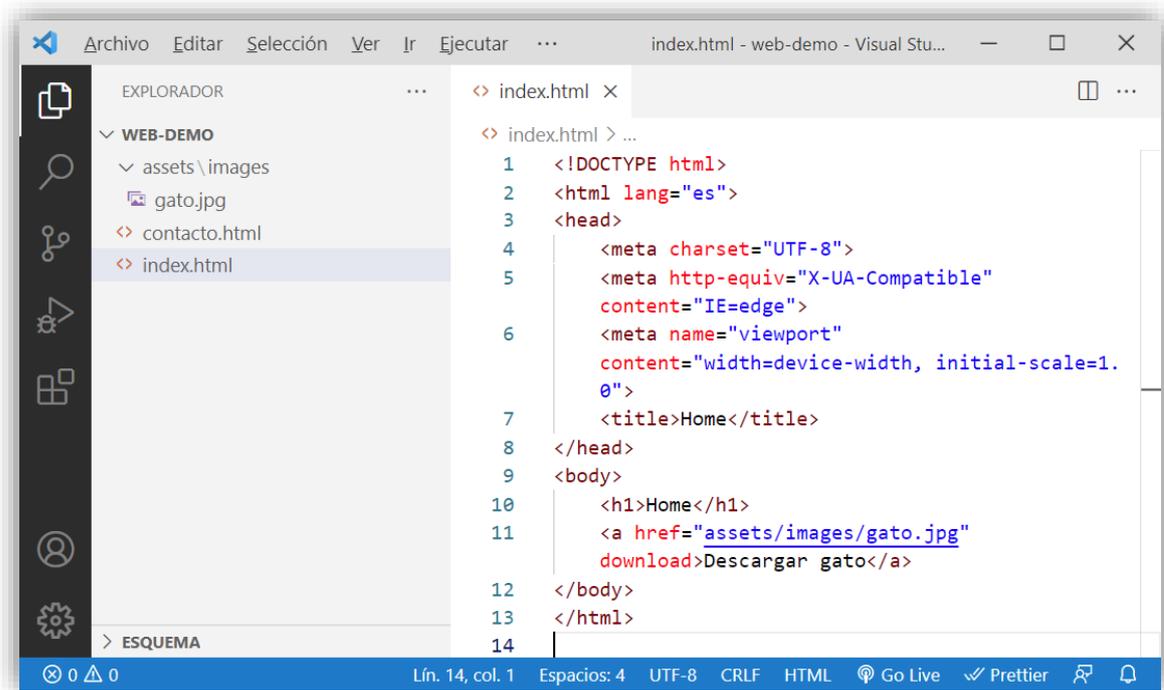
Se realiza la descarga.

Si no trabajaras con un servidor, esta opción nos daría error.

Si este atributo lo utilizamos en un archivo externo de nuestro servidor este no funcionará y nos mostrará la página.

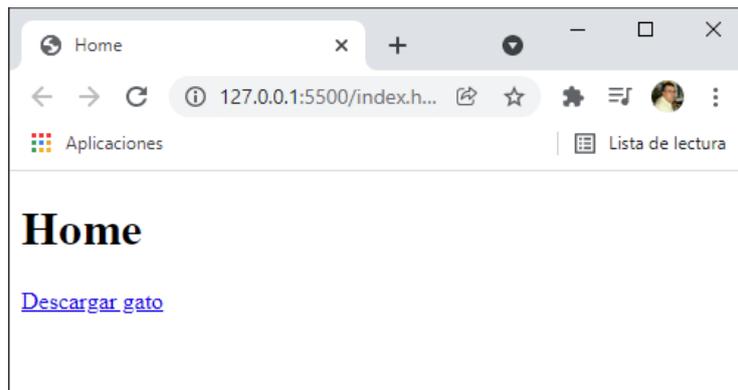
En nuestra carpeta WEB-DEMO hemos creado la carpeta assests y en ella la carpeta images.

Dentro de esta carpeta tenemos una imagen llamada gato.jpg.

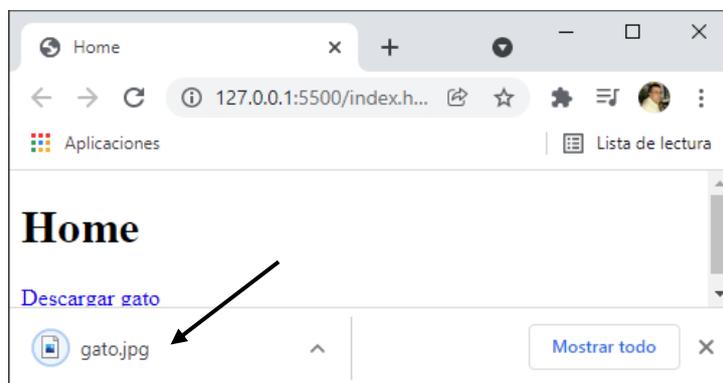


```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible"
6     content="IE=edge">
7   <meta name="viewport"
8     content="width=device-width, initial-scale=1.
9     0">
10  <title>Home</title>
11 </head>
12 <body>
13   <h1>Home</h1>
14   <a href="assets/images/gato.jpg"
15     download>Descargar gato</a>
16 </body>
17 </html>
```

Este será el resultado:



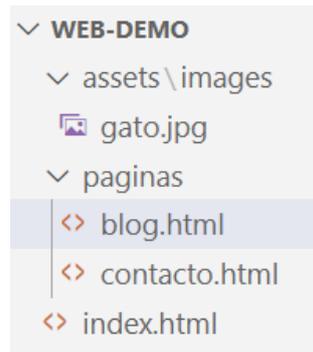
Si hacemos clic sobre el enlace.



En lugar de mostrarlo lo que hace es descargarlo.

Navegación con anclas

Tenemos la siguiente estructura:



Vamos a escribir el siguiente código:

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Blog</title>
</head>
<body>
  <h1 id="inicio">Blog</h1>
  <nav>
    <p><a href="#post-1">Post 1</a></p>
    <p><a href="#post-2">Post 2</a></p>
    <p><a href="#post-3">Post 3</a></p>
    <p><a href="#post-4">Post 4</a></p>
    <p><a href="#post-5">Post 5</a></p>
  </nav>
  <article id="post-1">
    <h2>Post 1</h2>
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit in tenetur quidem officiis repellendus, ab sunt neque mollitia at, dolore quia perspiciatis voluptatem ad aliquam rem eveniet numquam hic nostrum!</p>
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit in tenetur quidem officiis repellendus, ab sunt neque mollitia at, dolore quia perspiciatis voluptatem ad aliquam rem eveniet numquam hic nostrum!</p>
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit in tenetur quidem officiis repellendus, ab sunt neque mollitia at, dolore quia perspiciatis voluptatem ad aliquam rem eveniet numquam hic nostrum!</p>
    <a href="#inicio">Volver</a>
  </article>

```

```
</article>
<article id="post-2" >
  <h2>Post 2</h2>
```

<p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Consequuntur enim hic perferendis non animi ex eius corrupti earum eligendi, ab adipisci magni a deserunt quasi repellendus reprehenderit maxime eos expedita</p>

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit in tenetur quidem officii repellendus, ab sunt neque mollitia at, dolore quia perspiciatis voluptatem ad aliquam rem eveniet numquam hic nostrum!</p>

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit in tenetur quidem officii repellendus, ab sunt neque mollitia at, dolore quia perspiciatis voluptatem ad aliquam rem eveniet numquam hic nostrum!</p>

Volver

```
</article>
<article id="post-3">
  <h2>Post 3</h2>
```

<p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. In voluptatum omnis, at et, tenetur cum dolorum iste enim ipsum quod optio rerum tempora! Quaerat nam dolores adipisci vitae ducimus at.</p>

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit in tenetur quidem officii repellendus, ab sunt neque mollitia at, dolore quia perspiciatis voluptatem ad aliquam rem eveniet numquam hic nostrum!</p>

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit in tenetur quidem officii repellendus, ab sunt neque mollitia at, dolore quia perspiciatis voluptatem ad aliquam rem eveniet numquam hic nostrum!</p>

Volver

```
</article>
<article id="post-4" >
  <h2>Post 4</h2>
```

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Doloribus exercitationem atque necessitatibus eius non, soluta autem animi iure. Esse eos eius architecto quidem? Tenetur, delectus ullam quas odio pariatur dolorem.</p>

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit in tenetur quidem officii repellendus, ab sunt neque mollitia at, dolore quia perspiciatis voluptatem ad aliquam rem eveniet numquam hic nostrum!</p>

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit in tenetur quidem officii repellendus, ab sunt neque mollitia at, dolore quia perspiciatis voluptatem ad aliquam rem eveniet numquam hic nostrum!</p>

Volver

```

</article >
<article id="post-5">
  <h2>Post 5</h2>

  <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Amet itaque dicta natus dolor
  corrupti, maiores harum quas maxime quia consequuntur. Nemo ullam totam at sint
  dignissimos ex natus repellendus ad.</p>

  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit in tenetur quidem officiis
  repellendus, ab sunt neque mollitia at, dolore quia perspiciatis voluptatem ad aliquam rem
  eveniet numquam hic nostrum!</p>

  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit in tenetur quidem officiis
  repellendus, ab sunt neque mollitia at, dolore quia perspiciatis voluptatem ad aliquam rem
  eveniet numquam hic nostrum!</p>

  <a href="#inicio">Volver</a>

</article>
</body>
</html>

```

```

9  <body>
10 <h1 id="inicio">Blog</h1>
11 <nav>
12   <p><a href="#post-1">Post 1</a></p>
13   <p><a href="#post-2">Post 2</a></p>
14   <p><a href="#post-3">Post 3</a></p>
15   <p><a href="#post-4">Post 4</a></p>
16   <p><a href="#post-5">Post 5</a></p>
17 </nav>
18 <article id="post-1">
19   <h2>Post 1</h2>
20   <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
21   <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
22   <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
23   <a href="#inicio">Volver</a>
24 </article>

```

En las líneas 12, 13, 14, 15 y 16 tenemos el menú de navegación que nos envía a su correspondiente id.

Igualmente en la línea 23 tenemos un enlace “Volver” que nos situra al principio del documento al id inicio.

```

<article id="post-2" > ←
  <h2>Post 2</h2>
  <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Consequuntur
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit in tenet
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit in tenet
  <a href="#inicio">Volver</a> →
</article>

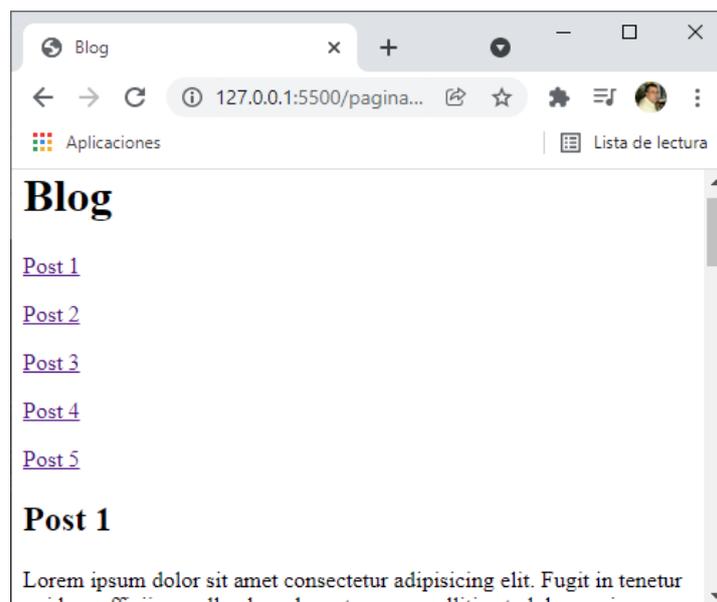
<article id="post-3"> ←
  <h2>Post 3</h2>
  <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. In voluptatum
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit in tenet
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit in tenet
  <a href="#inicio">Volver</a> →
</article>

<article id="post-4" > ←
  <h2>Post 4</h2>
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Doloribus exer
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit in tenet
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit in tenet
  <a href="#inicio">Volver</a> →
</article >

<article id="post-5"> ←
  <h2>Post 5</h2>
  <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Amet itaque d
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit in tenet
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit in tenet
  <a href="#inicio">Volver</a> →
</article>

```

Este será el resultado:



Ya podemos ir por los diferentes Post y además encontramos al final de cada Post un enlace para volver a principio de la página.

Si queremos desde otra página ir a una pero en una determinada posición de la misma y además no se encuentra ubicada en la misma carpeta:

```
<> index.html > ...
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6  |
7  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8  |   <title>Home</title>
9  </head>
10 <body>
11 |   <h1>Home</h1>
12 |   <nav>
13 |     <p><a href="/paginas/blog.html#post-1">Post 1</a></p>
14 |     <p><a href="/paginas/blog.html#post-2">Post 2</a></p>
15 |     <p><a href="/paginas/blog.html#post-3">Post 3</a></p>
16 |     <p><a href="/paginas/blog.html#post-4">Post 4</a></p>
17 |     <p><a href="/paginas/blog.html#post-5">Post 5</a></p>
18 |   </nav>
19 </body>
20 </html>
```

Con este ejemplo nos vamos a la página blog.html que se encuentra en la carpeta paginas, pero además a una determinada posición de la página.

Introducción a las listas

Las listas en HTML sirven para listar contenido. En función del tipo de contenido de nuestra lista tenemos tres tipos de listas:

ul -> unordered list: Se utilizan cuando el orden de los elementos no influyen (lista de la compra)

ol -> ordered list: Se utilizan cuando el orden de los elementos es importante (top 10)

Cada elemento de la lista se representa con la etiqueta , tanto en las ul como en las ol

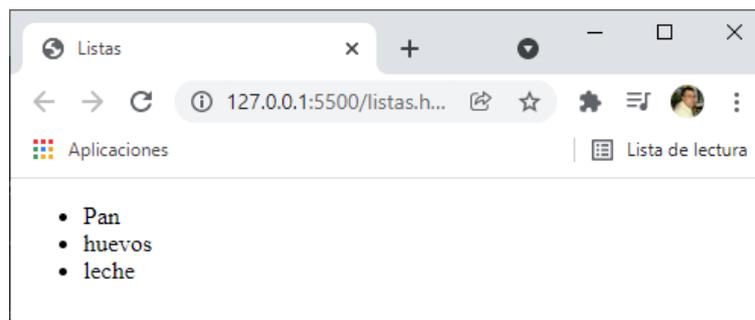
dl -> definition list: Se utilizan para hacer una lista de definiciones (diccionario)

Listas desordenadas

```
<> listas.html > ...
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Listas</title>
8  </head>
9  <body>
10     <ul>
11         <li>Pan</li>
12         <li>huevos</li>
13         <li>leche</li>
14     </ul>
15 </body>
16 </html>
```

Empezamos con la etiqueta y después cada elemento entre .

Este será el resultado:



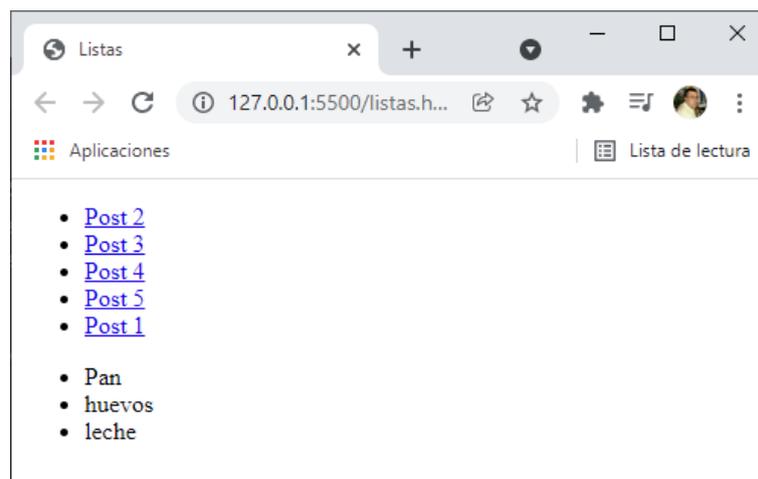
Ahora vamos a comentar como es la forma correcta de hacer un menú.

```
<> listas.html > ...
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Listas</title>
8 </head>
9 <body>
10    <nav>
11    <ul>
12        <li><a href="#post-2">Post 2</a></li>
13        <li><a href="#post-3">Post 3</a></li>
14        <li><a href="#post-4">Post 4</a></li>
15        <li><a href="#post-5">Post 5</a></li>
16        <li><a href="#post-1">Post 1</a></li>
17    </ul>
18 </nav>
19 <ul>
20     <li>Pan</li>
21     <li>huevos</li>
22     <li>leche</li>
23 </ul>
24 </body>
25 </html>
```

Esta es la semántica correcta para hacer un menú desde CSS ya cambiaremos el estilo.

Este es el resultado:



Listas ordenadas

```
<h2>Orden en el aprendizaje del desarrollo web</h2>
<ol>
  <li>HTML</li>
  <li>CSS</li>
  <li>JavaScript</li>
</ol>
</body>
</html>
```

Este será el resultado:



Listas de definición

dl -> definition list: Se utilizan para hacer una lista de definiciones (diccionario)

Cada elemento de una lista de definición lleva 2 etiquetas:

dt -> Definición term: El término que vamos a definir.

dd -> Definición descripción: La descripción del término.

```
<h2>Lenguajes para el desarrollo web</h2>
<dl>
  <dt>HTML</dt>
  <dd>HyperText Markup Language: Es un lenguaje de marcado
  que se usa para estructurar datos en una página web.</dd>
</dl>
<dl>
  <dt>CSS</dt>
  <dd>Cascade Style Sheets: Es un lenguaje de diseño que se
  utiliza para dar estilos al HTML.</dd>
</dl>
<dl>
  <dt>JavaScript</dt>
  <dd>Es un lenguaje de programación para dar interactividad
  a un sitio web.</dd>
</dl>
```

Este será el resultado:

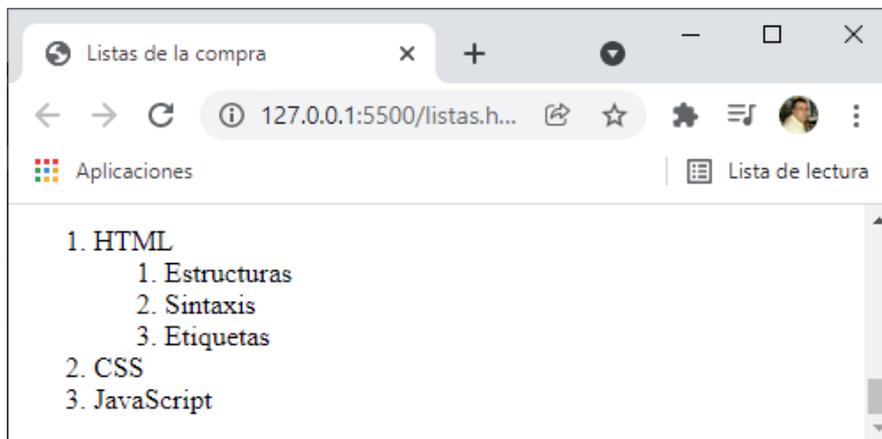


Listas anidadas y atributos

Se pueden construir listas anidadas teniendo en uno de los li otro ul/ol según sea necesario.

```
<ol>
  <li>
    HTML
    <ol>
      <li>Estructuras</li>
      <li>Sintaxis</li>
      <li>Etiquetas</li>
    </ol>
  </li>
  <li>CSS</li>
  <li>JavaScript</li>
</ol>
```

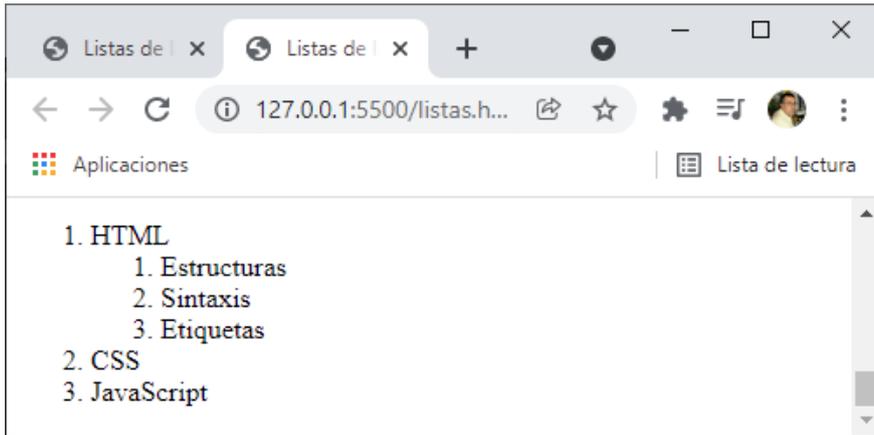
Este será el resultado:



Vamos a realizar unos cambios.

```
<ol>
  <li>HTML</li>
  <ul>
    <ol>
      <li>Estructuras</li>
      <li>Sintaxis</li>
      <li>Etiquetas</li>
    </ol>
  </ul>
  <li>CSS</li>
  <li>JavaScript</li>
</ol>
```

Este será el resultado:



Atributos de las listas:

ol ->

- type: Estilo de numeración (1, A, a, I, i)
- start: Inicio de las secuencias (un número)

ul ->

- type: Estilo de los ítems (disc, square, circle)

```
<h2>Atributos en las Listas</h2>  
<ul type="square">  
  <li>Item</li>  
  <li>Item</li>  
  <li>Item</li>  
</ul>  
<ol type="a" start="3">  
  <li>Item</li>  
  <li>Item</li>  
  <li>Item</li>  
</ol>
```

Este será el resultado:



Introducción a las tablas

Las tablas en HTML sirven para mostrar contenido tabulado.

La estructura básica de una se compone de

table -> Etiqueta que encierra una tabla.

tr -> table row, etiqueta que construye una fila.

td -> table data, etiqueta que construye una celda.

Truco: el número de celdas dentro de un td establecerá el número de columnas de la tabla.

Estructura básica de una tabla

```

<!DOCTYPE html>
<html lang="es">
<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Tablas</title>
</head>
<body>

  <h1>Tablas</h1>
  <table>

    <tr>

      <td>8:30 - 9:30</td>
      <td>Matemáticas</td>
      <td>Matemáticas</td>
      <td>Sociales</td>
      <td>Matemáticas</td>
      <td>Matemáticas</td>
    </tr>
    <tr>

      <td>9:30 - 10:25</td>
      <td>Lengua</td>
      <td>Lengua</td>
      <td>atemáticas</td>
      <td>Ciencias</td>
      <td>Lengua</td>
    </tr>
    <tr>

      <td>10:25 -11:20</td>
      <td>Educación Física</td>
      <td>Ciencias</td>
      <td>Lengua</td>
    </tr>
  </table>

```

```
<td>Lengua</td>
  <td>Educación Física</td>
</tr>
<tr>
  <td>11:20 - 11:45</td>
  <td>Recreo</td>
  <td></td>
  <td></td>
  <td></td>
  <td></td>
</tr>
<tr>
  <td>11:45 - 12:40</td>
  <td>Inglés</td>
  <td>Sociales</td>
  <td>Inglés</td>
  <td>Tutoría</td>
  <td>Inglés</td>
</tr>
<tr>
  <td>12:40 - 13:35</td>
  <td>Música</td>
  <td>Educación Física</td>
  <td>Plástica</td>
  <td>Religión</td>
  <td>Música</td>
</tr>
<tr>
  <td>13:35 - 14:40</td>
  <td>Tecnología</td>
  <td>Francés</td>
  <td>Tecnología</td>
  <td>Frances</td>
  <td>Tecnología</td>
</tr>
</table>
</body>
</html>
```

Este será el resultado:

The screenshot shows a web browser window with the title 'Tablas' and the URL '127.0.0.1:5500/tablas.html'. The page content is a table with the following data:

Tablas					
8:30 - 9:30	Matemáticas	Matemáticas	Sociales	Matemáticas	Matemáticas
9:30 - 10:25	Lengua	Lengua	atemáticas	Ciencias	Lengua
10:25 - 11:20	Educación Física	Ciencias	Lengua	Lengua	Educación Física
11:20 - 11:45	Recreo				
11:45 - 12:40	Inglés	Sociales	Inglés	Tutoria	Inglés
12:40 - 13:35	Música	Educación Física	Plástica	Religión	Música
13:35 - 14:40	Tecnología	Francés	Tecnología	Frances	Tecnología

Estructura completa de una tabla

Los títulos de las tablas se establecen con la etiqueta caption, es una etiqueta opcional, y según las especificaciones esa etiqueta se colora justo después de la etiqueta table.

Las cabeceras de las tablas se establecen con la etiqueta thead.

Dentro tendremos una etiqueta tr normal, pero en el caso de las celdas, las estableceremos con la etiqueta th en lugar de td.

El contenido de la tabla debe ir dentro de una etiqueta tbody para representar el cuerpo de la tabla.

De forma opcional podemos colocar untfoot a la tabla para establecer un pie de tabla, esto es algo que algunas tablas tienen como suma de cantidades o total.

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tablas</title>
</head>
<body>
  <h1>Tablas</h1>
  <table>
    <caption>HORARIO DE CLASE</caption>
    <thead>
      <tr>
        <th></th>
        <th>L</th>
        <th>M</th>
        <th>X</th>
    
```

```
<th>J</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
  <td>8:30 - 9:30</td>
  <td>Matemáticas</td>
  <td>Matemáticas</td>
  <td>Sociales</td>
  <td>Matemáticas</td>
  <td>Matemáticas</td>
</tr>
<tr>
  <td>9:30 - 10:25</td>
  <td>Lengua</td>
  <td>Lengua</td>
  <td>atemáticas</td>
  <td>Ciencias</td>
  <td>Lengua</td>
</tr>
<tr>
  <td>10:25 - 11:20</td>
  <td>Educación Física</td>
  <td>Ciencias</td>
  <td>Lengua</td>
  <td>Lengua</td>
  <td>Educación Física</td>
</tr>
<tr>
  <td>11:20 - 11:45</td>
  <td>Recreo</td>
  <td></td>
  <td></td>
  <td></td>
  <td></td>
</tr>
<tr>
  <td>11:45 - 12:40</td>
  <td>Inglés</td>
  <td>Sociales</td>
  <td>Inglés</td>

```

```
<td>Tutoria</td>
<td>Inglés</td>
</tr>
<tr>
<td>12:40 - 13:35</td>
<td>Música</td>
<td>Educación Física</td>
<td>Plástica</td>
<td>Religión</td>
<td>Música</td>
</tr>
<tr>
<td>13:35 - 14:40</td>
<td>Tecnología</td>
<td>Francés</td>
<td>Tecnología</td>
<td>Frances</td>
<td>Tecnología</td>
</tr>
</tbody>
<tfoot>
<tr>
<td>Total de asignaturas</td>
<td>12</td>
</tr>
</tfoot>
</table>
</body>
</html>
```

Este será el resultado:

HORARIO DE CLASE					
	L	M	X	J	V
8:30 - 9:30	Matemáticas	Matemáticas	Sociales	Matemáticas	Matemáticas
9:30 - 10:25	Lengua	Lengua	Matemáticas	Ciencias	Lengua
10:25 - 11:20	Educación Física	Ciencias	Lengua	Lengua	Educación Física
11:20 - 11:45	Recreo				
11:45 - 12:40	Inglés	Sociales	Inglés	Tutoría	Inglés
12:40 - 13:35	Música	Educación Física	Plástica	Religión	Música
13:35 - 14:40	Tecnología	Francés	Tecnología	Francés	Tecnología
Total de asignaturas 12					

Atributos de las tablas

Para hacer que las celdas ocupen más de una fila o más de una columna tenemos 2 atributos:

rowspan: sirve para que una celda ocupe más de una fila, el valor por defecto es de 1.

colspan: sirve para que una celda ocupe más de una columna, el valor por defecto es 1.

En este capítulo vamos a ver como agrupar celdas tanto en filas como en columnas.

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tablas</title>
</head>
<body>
  <h1>Tablas</h1>
  <table border=1>
    <caption>HORARIO DE CLASE</caption>
    <thead>
      <tr>
        <th></th>
        <th>L</th>
        <th>M</th>

```

```
<th>X</th>
<th>J</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:30 - 9:30</td>
<td>Matemáticas</td>
<td>Matemáticas</td>
<td>Sociales</td>
<td>Matemáticas</td>
<td>Matemáticas</td>
</tr>
<tr>
<td>9:30 - 10:25</td>
<td>Lengua</td>
<td>Lengua</td>
<td>atemáticas</td>
<td>Ciencias</td>
<td>Lengua</td>
</tr>
<tr>
<td>10:25 - 11:20</td>
<td>Educación Física</td>
<td>Ciencias</td>
<td>Lengua</td>
<td>Lengua</td>
<td>Educación Fisica</td>
</tr>
<tr>
<td>11:20 - 11:45</td>
<td colspan="5"><p>RECREO</p></td>
</tr>
<tr>
<td>11:45 - 12:40</td>
<td>Inglés</td>
<td>Sociales</td>
<td>Inglés</td>
<td>Tutoria</td>
<td>Inglés</td>
</tr>
```

```
</tr>
<tr>
  <td>12:40 - 13:35</td>
  <td rowspan="2">Tecnología</td>
  <td>Educación Física</td>
  <td>Plástica</td>
  <td>Religión</td>
  <td rowspan="2">Tecnología</td>
</tr>
<tr>
  <td>13:35 - 14:40</td>
  <td>Francés</td>
  <td>Tecnología</td>
  <td>Frances</td>
</tr>
</tbody>
<tfoot>
  <tr>
    <td colspan="5">Total de asignaturas</td>
    <td>12</td>
  </tr>
</tfoot>
</table>
</body>
</html>
```

Este será el resultado:

The screenshot shows a web browser window with the title 'Tablas'. The address bar shows '127.0.0.1:5500/tablas.html'. The page content includes a title 'Tablas' and a table titled 'HORARIO DE CLASE'. The table has columns for days of the week (L, M, X, J, V) and rows for time slots and subjects. The last row shows 'Total de asignaturas' with the value '12'.

	L	M	X	J	V
8:30 - 9:30	Matemáticas	Matemáticas	Sociales	Matemáticas	Matemáticas
9:30 - 10:25	Lengua	Lengua	Matemáticas	Ciencias	Lengua
10:25 - 11:20	Educación Física	Ciencias	Lengua	Lengua	Educación Física
11:20 - 11:45	RECREO				
11:45 - 12:40	Inglés	Sociales	Inglés	Tutoría	Inglés
12:40 - 13:35	Tecnología	Educación Física	Plástica	Religión	Tecnología
13:35 - 14:40		Francés	Tecnología	Francés	
Total de asignaturas					12

Seleccionar columnas

Cuando necesitamos seleccionar una columna, tenemos la etiqueta colgroup, que nos permite seleccionar una columna en concreto. Dentro pondremos tantas etiquetas col como columnas tengamos, cada etiqueta col equivale a una columna siguiendo el mismo orden que tiene la tabla.

Si necesitamos que una etiqueta col agrupe más de una columna, tenemos el atributo span, que funciona exactamente igual que rowspan y colspan.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tablas</title>
  <style>
    col:nth-child(1){
      background-color: lightcoral;
    }
    col:nth-child(2){
      background-color: yellow;
    }
  </style>
```

```
</head>
<body>

  <h1>Tablas</h1>
  <table border=1>

    <caption>HORARIO DE CLASE</caption>
    <colgroup>

      <col span="5">
      <col>

    </colgroup>
    <thead>

      <tr>

        <th></th>
        <th>L</th>
        <th>M</th>
        <th>X</th>
        <th>J</th>
        <th>V</th>

      </tr>
    </thead>
    <tbody>
    <tr>

      <td>8:30 - 9:30</td>
      <td>Matemáticas</td>
      <td>Matemáticas</td>
      <td>Sociales</td>
      <td>Matemáticas</td>
      <td>Matemáticas</td>

    </tr>
    <tr>

      <td>9:30 - 10:25</td>
      <td>Lengua</td>
      <td>Lengua</td>
      <td>atemáticas</td>
      <td>Ciencias</td>
      <td>Lengua</td>

    </tr>
    <tr>

      <td>10:25 -11:20</td>
      <td>Educación Física</td>
      <td>Ciencias</td>
      <td>Lengua</td>
```

```
<td>Lengua</td>
  <td>Educación Física</td>
</tr>
<tr>
  <td>11:20 - 11:45</td>
  <td colspan="5"><p>RECREO</p></td>
</tr>
<tr>
  <td>11:45 - 12:40</td>
  <td>Inglés</td>
  <td>Sociales</td>
  <td>Inglés</td>
  <td>Tutoría</td>
  <td>Inglés</td>
</tr>
<tr>
  <td>12:40 - 13:35</td>
  <td rowspan="2">Tecnología</td>
  <td>Educación Física</td>
  <td>Plástica</td>
  <td>Religión</td>
  <td rowspan="2">Tecnología</td>
</tr>
<tr>
  <td>13:35 - 14:40</td>
  <td>Francés</td>
  <td>Tecnología</td>
  <td>Frances</td>
</tr>
</tbody>
<tfoot>
  <tr>
    <td colspan="5">Total de asignaturas</td>
  <td>12</td>
</tr>
</tfoot>
</table>
```

</body>

</html>

Este será el resultado:



The screenshot shows a web browser window with the title 'Tablas' and the URL '127.0.0.1:5500/tablas.html'. The main content is a table titled 'HORARIO DE CLASE' (Class Schedule). The table has columns for days of the week (L, M, X, J, V) and rows for time slots and subjects. The 'V' column is highlighted in yellow. The table content is as follows:

	L	M	X	J	V
8:30 - 9:30	Matemáticas	Matemáticas	Sociales	Matemáticas	Matemáticas
9:30 - 10:25	Lengua	Lengua	Matemáticas	Ciencias	Lengua
10:25 - 11:20	Educación Física	Ciencias	Lengua	Lengua	Educación Física
11:20 - 11:45	RECREO				
11:45 - 12:40	Inglés	Sociales	Inglés	Tutoría	Inglés
12:40 - 13:35	Tecnología	Educación Física	Plástica	Religión	Tecnología
13:35 - 14:40		Francés	Tecnología	Francés	
Total de asignaturas					12

Cómo (NO) dar estilos a una tabla

En este capítulo no pretendemos dar una introducción a CSS pero como ejemplo para que podrás ver el efecto de este lenguaje gráfico.

Código:

```
<style>
  table{
    font-family: sans-serif;
    border: 2px solid grey;
    border-collapse: collapse;
    text-align: center;
    margin-left: auto;
    margin-right: auto;
  }
  caption{
    font-size: 2rem;
    font-weight: bold;
    margin-bottom: 1rem;
  }
  th,
  td{
    border: 2px solid grey;
    padding: .5rem;
  }
  tr:nth-child(4) td:nth-child(2){
    background-color: lightgreen;
    letter-spacing: 90px;
    padding-left: 50px;
    text-transform: uppercase;
  }
  td:first-child,
  th,
  tfoot{
    background-color: lightgreen;
  }
</style>
```

Está escrito dentro de las etiquetas <head> y </head>

Este es el resultado:

Tablas

127.0.0.1:5500/tablas.html

Aplicaciones | Lista de lectura

Tablas

HORARIO DE CLASE

	L	M	X	J	V
8:30 - 9:30	Matemáticas	Matemáticas	Sociales	Matemáticas	Matemáticas
9:30 - 10:25	Lengua	Lengua	Matemáticas	Ciencias	Lengua
10:25 - 11:20	Educación Física	Ciencias	Lengua	Lengua	Educación Física
11:20 - 11:45	R E C R E O				
11:45 - 12:40	Inglés	Sociales	Inglés	Tutoría	Inglés
12:40 - 13:35	Tecnología	Educación Física	Plástica	Religión	Tecnología
13:35 - 14:40		Francés	Tecnología	Francés	
Total de asignaturas					12

Más etiquetas importantes de bloque

address -> Se utiliza para aportar información de contacto para el artículo más cercano o para todo el body.

blockquote -> Se utiliza para marcar las citas a otros autores o documentos. Se puede incluir el atributo cite para poner un enlace al documento original o fuente.

pre -> Se utiliza para tener código preformateado que necesita ser presentado igual que se escribió.

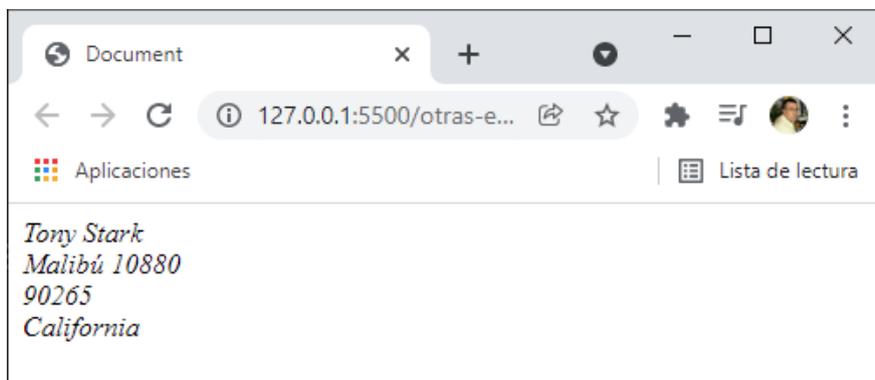
div -> Se utiliza como división del documento, semánticamente no significa nada, en un contenedor genérico que se usa generalmente para dar estilos a través de CSS o para usar algo denominado "delegación de eventos" en JavaScript.

hr -> horizontal rule, se utiliza para decirle al navegador que vamos a cambiar de tema.

Ejemplo de address:

```
<> otras-etiquetas-bloque.html > ...
 1  <!DOCTYPE html>
 2  <html lang="es">
 3  <head>
 4  |   <meta charset="UTF-8">
 5  |   <meta http-equiv="X-UA-Compatible" content="IE=edge">
 6  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
 7  |   <title>Document</title>
 8  </head>
 9  <body>
10  |   <address>
11  |       Tony Stark<br>
12  |       Malibú 10880<br>
13  |       90265<br>
14  |       California
15  |   </address>
16 </body>
17 </html>
```

Resultado:



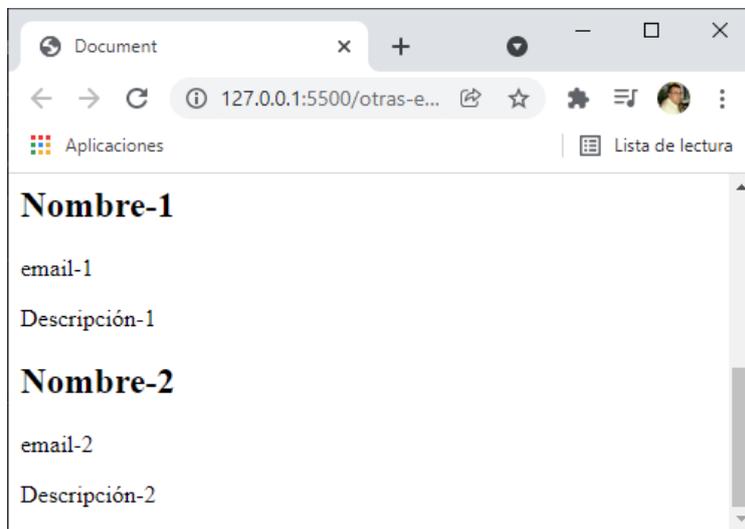
Etiqueta <blockquote>:

Etiqueta <div>:

Es la forma de dividir nuestro documento en contenedores, a la hora de mostrarlo en el navegador no se refleja, hasta que le agreguemos estilos con css.

```
<div>
  <h2>Nombre-1</h2>
  <p>email-1</p>
  <p>Descripción-1</p>
</div>
<div>
  <h2>Nombre-2</h2>
  <p>email-2</p>
  <p>Descripción-2</p>
</div>
```

Este será el resultado:

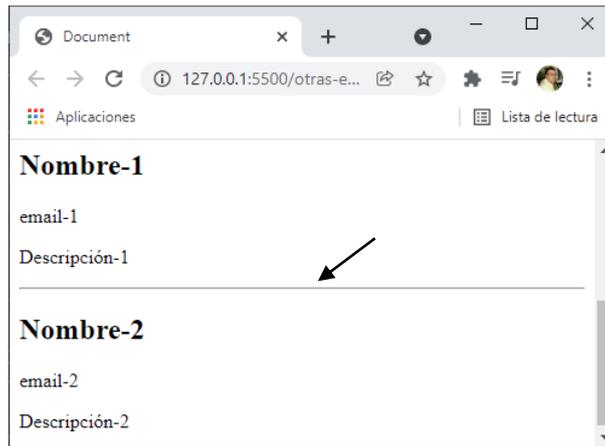


Etiqueta <hr>:

Informa al navegador que el tema que viene a continuación es totalmente distinto al anterior, aunque además dibuje una línea.

```
<div>
  <h2>Nombre-1</h2>
  <p>email-1</p>
  <p>Descripción-1</p>
</div>
<hr> ←
<div>
  <h2>Nombre-2</h2>
  <p>email-2</p>
  <p>Descripción-2</p>
</div>
```

Este será el resultado:



Más etiquetas de línea

span -> Es un contenedor de línea, equivalente a div, se suele usar para encerrar palabras o pequeños textos y darles estilo a través de CSS o localizarlos desde javascript. Semánticamente no significa nada.

q -> Es el equivalente a blockquote, significa quote, por eso el de bloque se llama block – quote. Sirve para poner citas pero en línea.

code -> Sirve para encerrar código que queremos representar visualmente, suele ir unido con la etiqueta pre.

Entidades especiales en HTML – Código ASCII

<http://ascii.cl/es/codigos-html.html>

Ejemplo etiqueta

```
<p>Lorem ipsum dolor sit amet <span>consectetur adipiscing</span> elit. Itaque consequuntur minus, repellat fuga architecto facere saepe est. Asperiores adipisci ab reiciendis itaque tempore. Totam voluptates numquam nulla consequuntur culpa aliquid.</p>
```

Con la etiqueta span hemos seleccionado dos palabras, si lo vemos en el navegador esto no se reflejará en nada, pero si entre las etiquetas <head> y </head> agregamos un estilo, haciendo referencia a span

```
<style>
  span{
    color: red;
  }
</style>
```

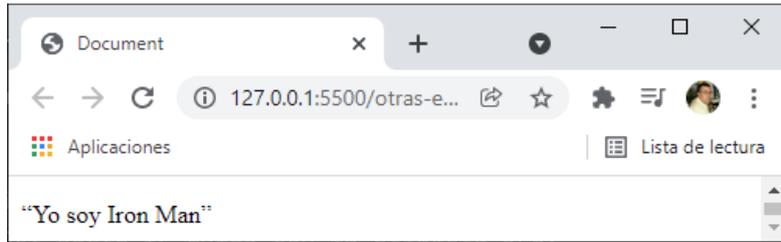
Este será el resultado:



Etiqueta <q>:

```
<q>Yo soy Iron Man</q>
```

Este será el resultado:



Lo cierra entre comillas.

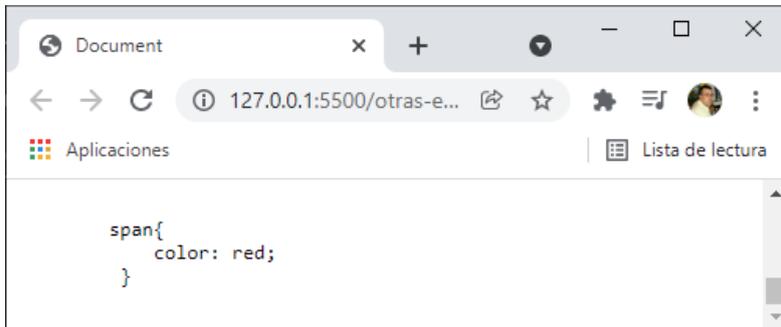
```
<q cite="">Yo soy Iron Man</q>
```

También admite cite como blockquote que no se reflejará en el navegador, pero si dará información al navegador.

Etiqueta <code>:

```
<pre>
  <code>
    span{
      color: red;
    }
  </code>
</pre>
```

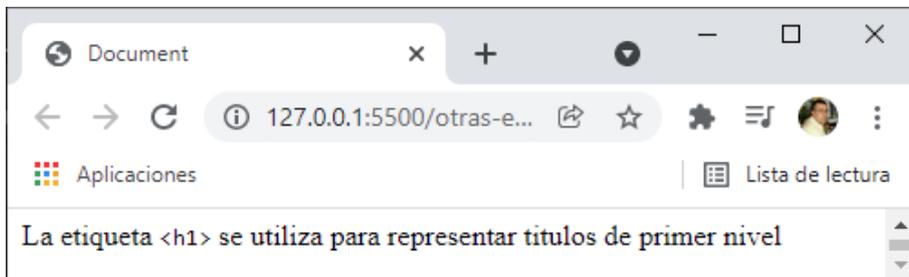
Este será el resultado:



Otro ejemplo:

```
<p>La etiqueta <code>&#60;h1&#62;</code> se utiliza para representar titulos de primer nivel</p>
```

Este será el resultado:



Introducción a Formularios

La estructura básica de un formulario se compone de 4 elementos.

form -> es la etiqueta que engloba nuestro formulario.

label -> Sirve para escribir el nombre del campo a rellenar.

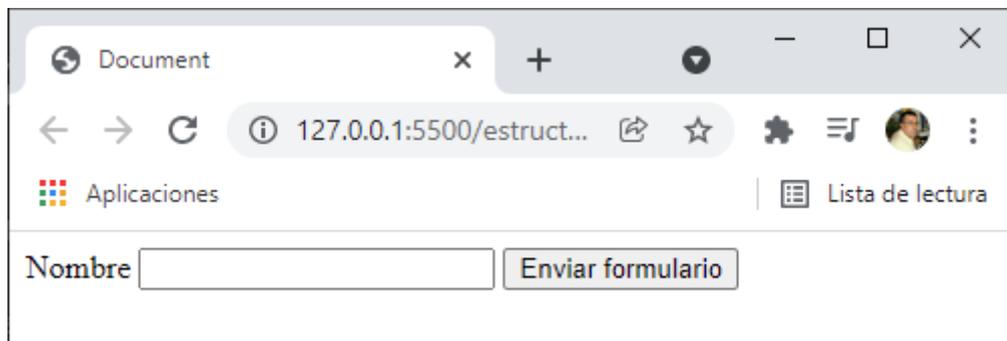
input -> Sirve para crear un campo que permitirá al usuario interactuar.

button -> crea un botón que permitirá enviar el formulario

Estructura básica de un formulario

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10  <form>
11    <label>Nombre</label>
12    <input>
13    <button>Enviar formulario</button>
14  </form>
15 </body>
16 </html>
```

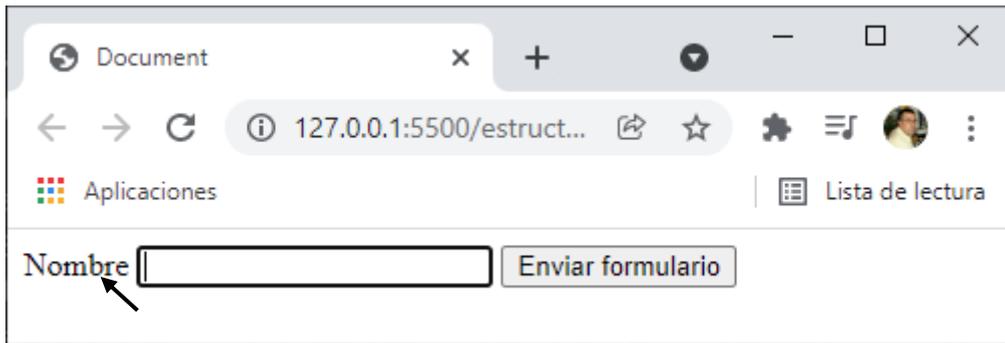
Este será el resultado:



Asociar input y label

```
<form>
  <label for="name">Nombre</label>
  <input id="name">
  <button>Enviar formulario</button>
</form>
```

En la etiqueta label le decimos para "name" y en el input un id con "name", ahora si tocamos el nombre desde la página web también se puede rellenar el campo input.



Otra forma de hacerlo:

```
<form>
  <label>
    Nombre
    <input>
  </label>
  <button>Enviar formulario</button>
</form>
```

El resultado es el mismo.

button vs type button

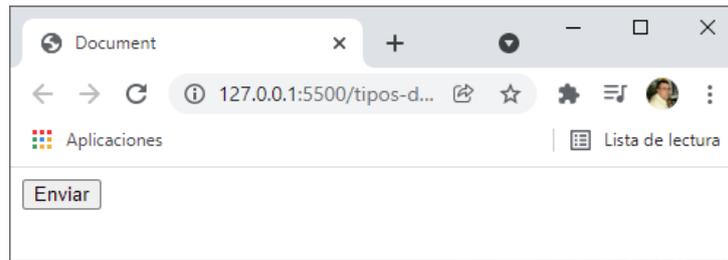
input type:

- button -> se comporta igual que un botón <button>
- color -> Se utiliza para especificar un color
- date -> Se utiliza para introducir una fecha
- datetime -> Obsoleto
- datetime-local -> fecha y hora, no funciona en firefox
- email -> Se utiliza para introducir un email
- hidden -> Campo oculto, puede contener valor pero no se mostrar
- month -> Se utiliza para introducir un mes
- number -> Se utiliza par valores numéricos
- password -> Se utiliza para contraseñas
- range -> Se utiliza para establecer un rango
- reset -> Se utiliza para resetear el formulario
- search -> Se utiliza para las barras de búsqueda
- submit -> Se utiliza para enviar el formulario
- tel -> Se utiliza para introducir números telefónicos
- text -> Valor por defecto
- time -> Se utiliza para introducir una hora
- url -> Se utiliza para introducir URLs

week -> Se utiliza para introducir el número de la semana del año button:

```
<form>
  <input type="button" value="Enviar">
</form>
```

Este será el resultado:



Es lo mismo que si escribimos:

```
<form>
  <button>Enviar</button>
</form>
```

Aunque a simple vista parece que hacen lo mismo hay diferencias el botón realizado con un input no puede enviar el formulario ni resetearlo.

El botón que realicen mediante un input sirve para poder ejecutar algún código de JavaScript.

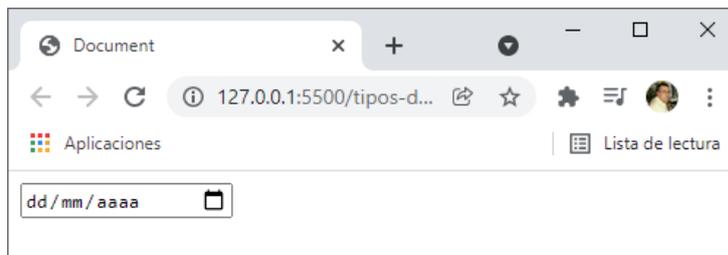
Si con un input queremos enviar un formulario este tiene que ser de tipo "submit".

```
<form>
  <input type="submit" value="Enviar input">
</form>
```

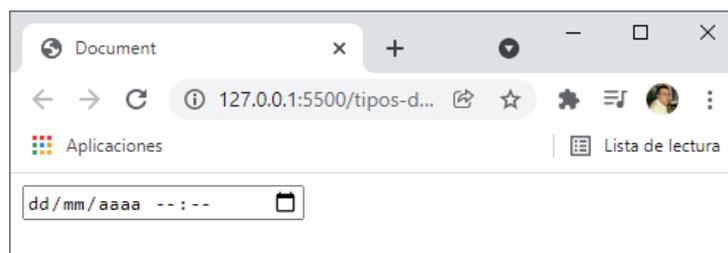
Input para fechas

```
<form>
  <input type="date">
</form>
```

Este será el resultado:



```
<form>
  <input type="datetime-local">
</form>
```



Permite poner la fecha y la hora pero no es compatible con Firefox.

#YOAPRENDOCONDORIAN

Se recomienda usar `datetime-local` y `time` para seleccionar fecha y hora.

```
<form>
  <input type="date">
  <input type="time">
</form>
```

A screenshot of a web browser window. The address bar shows the URL `127.0.0.1:5500/tipos-d...`. Below the address bar, there are two input fields: the first is a date picker with the placeholder text `dd/mm/aaaa` and a calendar icon; the second is a time picker with the placeholder text `--:--` and a clock icon.

```
<form>
  <input type="month">
</form>
```

A screenshot of a web browser window. The address bar shows the URL `127.0.0.1:5500/tipos-d...`. Below the address bar, there is a single input field for selecting a month, with the placeholder text `----- de ----` and a calendar icon.

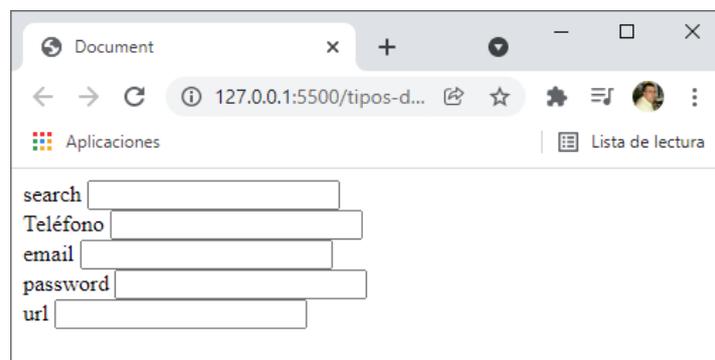
```
<form>
  <input type="week">
</form>
```

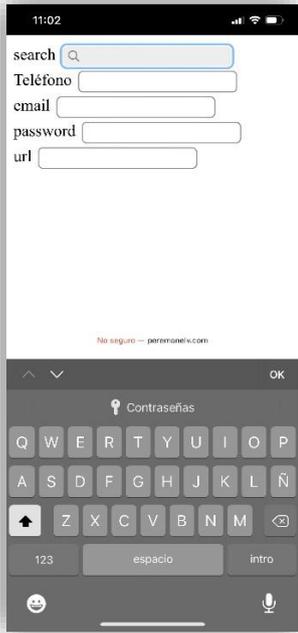
A screenshot of a web browser window. The address bar shows the URL `127.0.0.1:5500/tipos-d...`. Below the address bar, there is a single input field for selecting a week, with the placeholder text `Semana --, ----` and a calendar icon.

Inputs para móviles

```
<form>
  <div>
    <label> search
    |   <input type="search">
    </label>
  </div>
  <div>
    <label> Teléfono
    |   <input type="tel">
    </label>
  </div>
  <div>
    <label> email
    |   <input type="email">
    </label>
  </div>
  <div>
    <label> password
    |   <input type="password">
    </label>
  </div>
  <div>
    <label> url
    |   <input type="url">
    </label>
  </div>
</form>
```

Este será el resultado desde la página web del ordenador:





Search



Teléfono



email



password



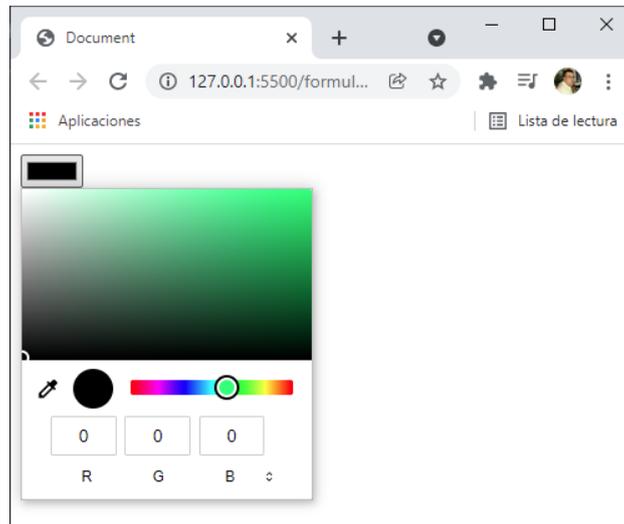
url

En el móvil se puede ver las diferencias al introducir datos en los distintos campos.

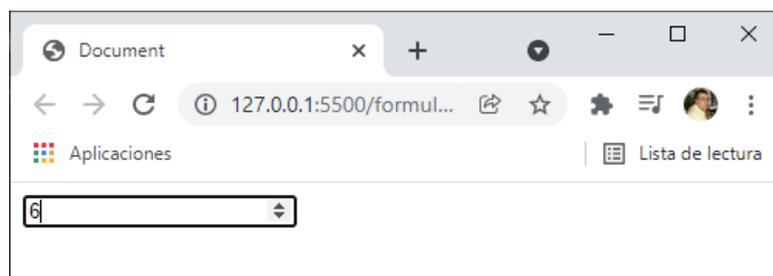
Inputs extras

```
<form>  
|   <input type="color">  
</form>
```

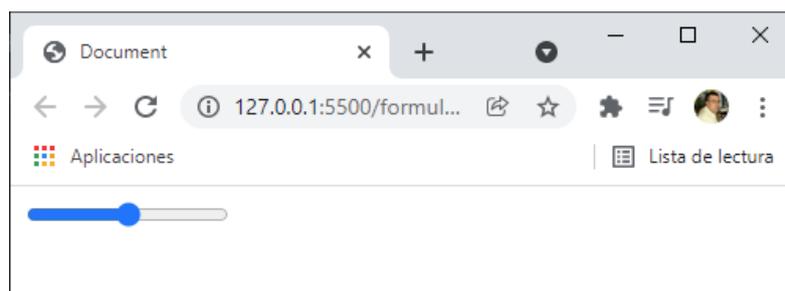
Nos permite seleccionar un tipo de color, si hacemos una página donde el usuario puede elegir el color.



```
<form>  
|   <input type="number">  
</form>
```



```
<form>  
|   <input type="range">  
</form>
```

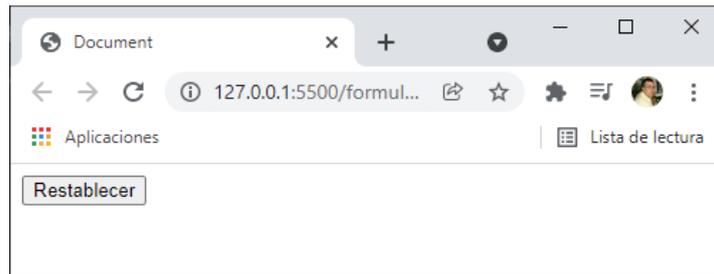


```
<form>
  <input type="range" step="10" min="0" max="100" >
</form>
```

Aquí controlamos el valor mínimo, máximos y el salto.

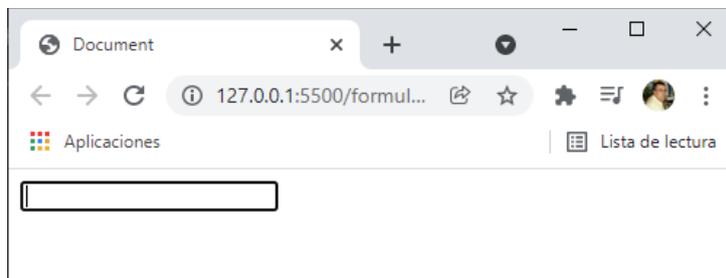
```
<form>
  <input type="reset">
</form>
```

Si le damos a este botón reestablece los datos del formulario a los valores por defecto.



```
<form>
  <input type="text">
</form>
```

Es el valor por defecto.



Input radio

radio -> Permite seleccionar una única opción de una lista de opciones relacionadas.

checkbox -> Permite seleccionar varias opciones de una lista de opciones relacionadas.

select -> Crea una lista de opciones donde podemos seleccionar una o varias opciones.

Cada opción irá dentro de una etiqueta <option> </option>.

Si tenemos muchas opciones podemos ordenarlas por categoría a través de la etiqueta <optgroup> con el atributo label para nombrar la categoría.

```
<body>
  <h1>Elementos seleccionables</h1>
  <form>
    <h2>Género</h2>
    <label>Masculino
      |   <input type="radio" name="genero" checked>
    </label>
    <label>Femenino
      |   <input type="radio" name="genero">
    </label>
    <label>Desconocido
      |   <input type="radio" name="genero">
    </label>
  </form>
</body>
```

Como solo podemos seleccionar uno tenemos que poner el atributo name y en los tres el mismo nombre.

Si queremos que aparezca uno marcado por defecto utilizaremos checked.

Este será el resultado:



```
<form>
  <h2>Género</h2>
  <label>Masculino
    |   <input type="radio" name="genero" value="masculino" checked>
  </label>
  <label>Femenino
    |   <input type="radio" name="genero" value="femenino">
  </label>
  <label>Desconocido
    |   <input type="radio" name="genero" value="desconocido">
  </label>
</form>
```

Si queremos que retorne un valor utilizaremos el atributo value.

Input checkbox

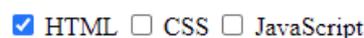
```
<body>
  <h1>Elementos seleccionables</h1>
  <form>
    <h2>Lenguajes que conoces</h2>
    <label>
      <input type="checkbox" name="languages" value="html">
      HTML
    </label>
    <label>
      <input type="checkbox" name="languages" value="css">
      CSS
    </label>
    <label>
      <input type="checkbox" name="languages" value="javascript">
      JavaScript
    </label>
  </form>
</body>
```

Este será el resultado:



Si queremos que alguna opción aparezca marcada por defecto:

```
<label>
  <input type="checkbox" name="languages" value="html" checked>
  HTML
</label>
```



Se puede seleccionar más de una opción.

Elemento select básico

```
<body>
  <h1>Elementos seleccionables</h1>
  <form>
    <select name="lenguajes">
      <option value="html">HTML</option>
      <option value="css">CSS</option>
      <option value="javascript">JavaScript</option>
    </select>
  </form>
</body>
```

Este será el resultado:



Si queremos que se pueda seleccionar más de uno.

```
<body>
  <h1>Elementos seleccionables</h1>
  <form>
    <select name="lenguajes" multiple>
      <option value="html">HTML</option>
      <option value="css">CSS</option>
      <option value="javascript">JavaScript</option>
    </select>
  </form>
</body>
```

Este será el resultado:

Con ayuda de la tecla Ctrl se puede seleccionar más de una opción.

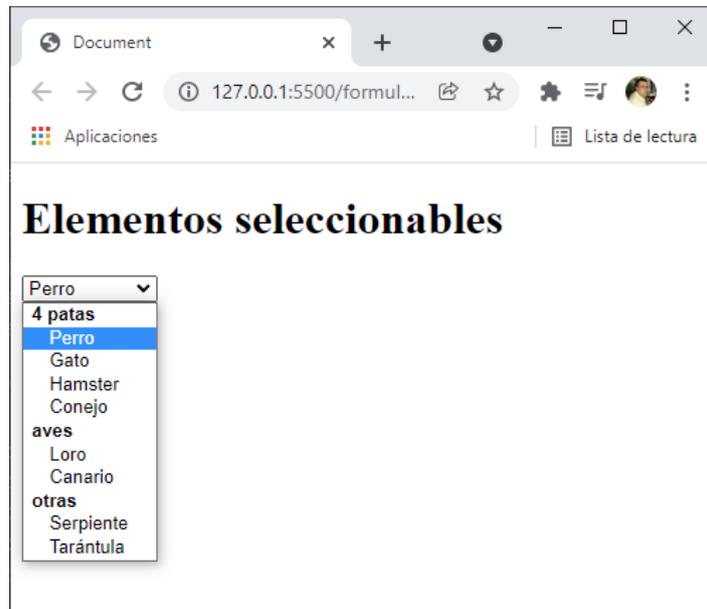


Elementos select avanzado

Si tenemos muchas opciones podemos ordenarlas por categorías a través de la etiqueta `<optgroup>` con el atributo `label` para nombrar la categoría.

```
<form>
  <select name="mascotas">
    <optgroup label="4 patas">
      <option value="perro">Perro</option>
      <option value="gato">Gato</option>
      <option value="hamster">Hamster</option>
      <option value="conejo">Conejo</option>
    </optgroup>
    <optgroup label="aves">
      <option value="loro">Loro</option>
      <option value="canario">Canario</option>
    </optgroup>
    <optgroup label="otras">
      <option value="serpiente">Serpiente</option>
      <option value="tarantula">Tarántula</option>
    </optgroup>
  </select>
</form>
```

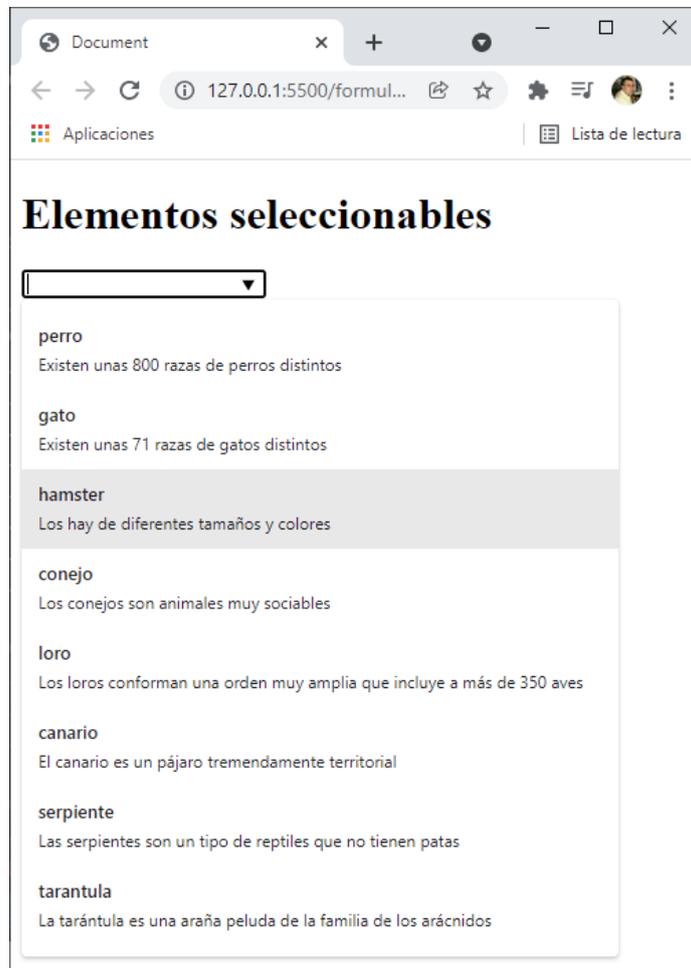
Este será el resultado:



Datalist

```
<body>
  <h1>Elementos seleccionables</h1>
  <form>
    <input type="list" list="mascotas">
    <datalist id="mascotas">
      <option value="perro">Existen unas 800 razas de perros distintos</option>
      <option value="gato">Existen unas 71 razas de gatos distintos</option>
      <option value="hamster">Los hay de diferentes tamaños y colores</option>
      <option value="conejo">Los conejos son animales muy sociables</option>
      <option value="loro">Los loros conforman una orden muy amplia que incluye a más de 350 aves</option>
      <option value="canario">El canario es un pájaro tremendamente territorial</option>
      <option value="serpiente">Las serpientes son un tipo de reptiles que no tienen patas</option>
      <option value="tarantula">La tarántula es una araña peluda de la familia de los arácnidos</option>
    </datalist>
  </form>
</body>
</html>
```

Este será el resultado.



Más elementos para formularios

fieldset -> Se utiliza para agrupar elementos dentro de un formulario.

legend -> Representa una etiqueta para el contenido del fieldset.

file -> Este input se utiliza para cargar archivos y enviarlos desde un formulario

meter -> Representa un valor dentro de un rango conocido

progress -> Representa el progreso de una tarea

textarea -> Se utiliza para introducir texto en un formulario

```
<body>
  <h1>Formulario</h1>
  <form>
    <fieldset>
      <legend>Datos personales:</legend>
      <div>
        <label for="name">Nombre:</label>
        <input type="text" id="name">
      </div>
      <div>
        <label for="surname">Apellidos:</label>
        <input type="text" id="surname">
      </div>
    </fieldset>
  </form>
</body>
</html>
```

Este será el resultado:



Document

127.0.0.1:5500/formul...

Aplicaciones | Lista de lectura

Formulario

Datos personales:

Nombre:

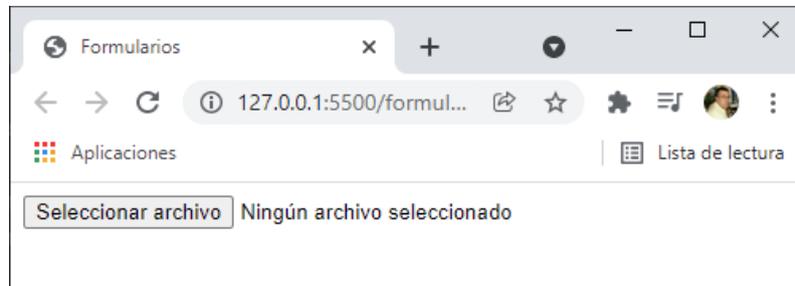
Apellidos:

Para poder enviar un archivo desde un formulario.

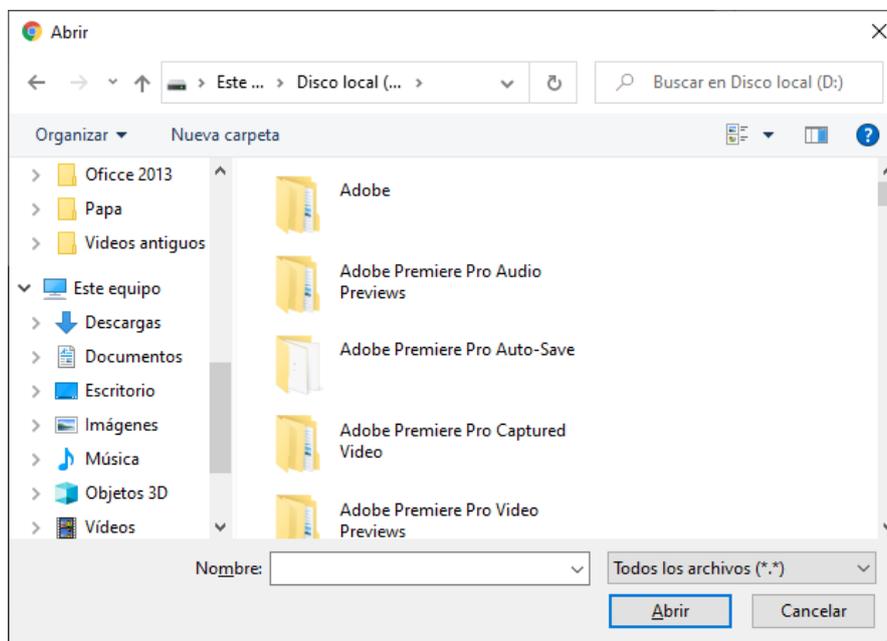
#YOAPRENDOCONDORIAN

```
<body>
  <form>
    <input type="file">
  </form>
</body>
```

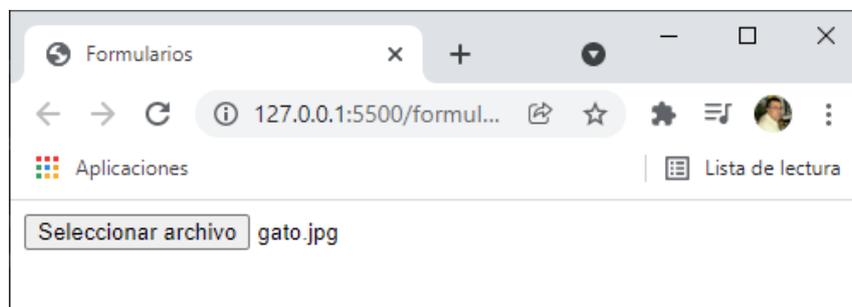
Este será el resultado:



Si le damos al botón:

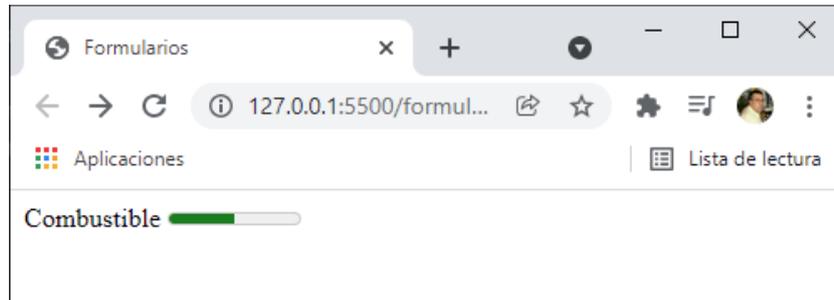


Podremos cargar el archivo.

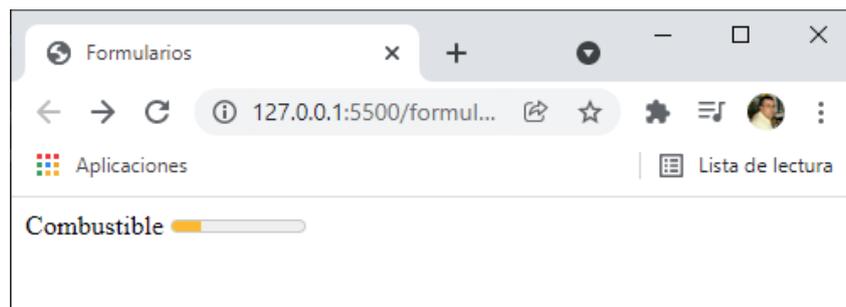


```
<form>
  <label for="fuel">Combustible</label>
  <meter id="fuel" min="0" max="100" value="50" low="30" high="75" optimun="50">
  </meter>
</form>
```

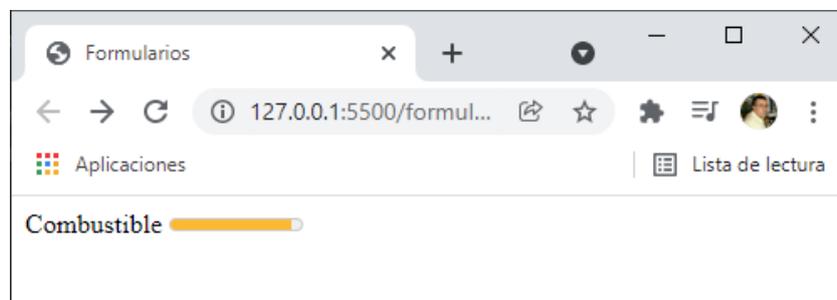
Valor mínimo 0, máximo 100 valor actual 50 por debajo de 30 color amarillo, por encima de 75 amarillo, y en 50 optimo color verde.



```
<form>
  <label for="fuel">Combustible</label>
  <meter id="fuel" min="0" max="100" value="22" low="30" high="75" optimun="50">
  </meter>
</form>
```

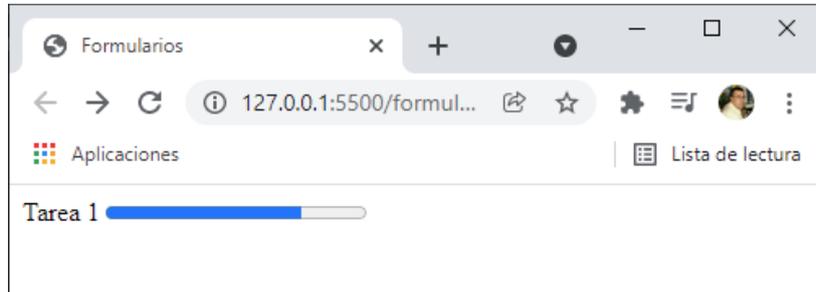


```
<form>
  <label for="fuel">Combustible</label>
  <meter id="fuel" min="0" max="100" value="92" low="30" high="75" optimun="50">
  </meter>
</form>
```



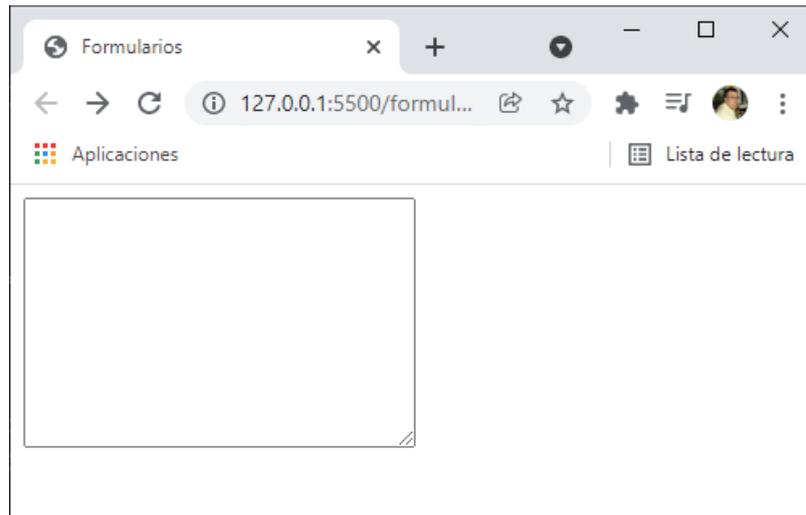
```
<form>
  <label for="task">Tarea 1</label>
  <progress id="task" max="100" value="75"></progress>
</form>
```

Este será el resultado:



```
<form>
  <textarea name="" id="" cols="30" rows="10"></textarea>
</form>
```

Este será el resultado:



Para introducir texto con varias líneas.

Atributos para formularios

placeholder -> Da una pista de lo que el usuario tiene que introducir.

required -> Hacer que un campo sea obligatorio.

disabled -> Desactiva el campo, no se podrá escribir en el autofocus.

readonly -> Hace que el campo solo sea de lectura.

min – max -> Establece el mínimo y el máximo de un campo numérico.

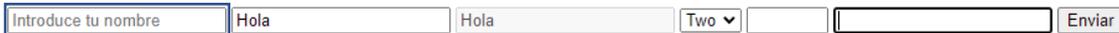
minlength – maxlength -> Establece el mínimo y máximo de caracteres en un campo de texto.

selected -> Equivale a checked en los select, sirve para establecer una opción por defecto.

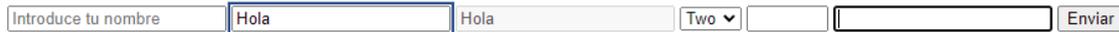
autofocus -> Para poner el foco por defecto al cargar el formulario.

```
9   <body>
10   <form>
11     <input type="text" placeholder="Introduce tu nombre" required>
12     <input type="text" value="Hola" readonly>
13     <input type="text" value="Hola" disabled>
14     <select name="numbers">
15       <option value="one">One</option>
16       <option value="two" selected>Two</option>
17     </select>
18     <input type="number" min="5" max="10">
19     <input type="text" minlength="3" maxlength="5" required autofocus>
20     <input type="submit">
21   </form>
22 </body>
```

Línea 11 con placeholder ponemos una información para que el usuario sepa lo que debe introducir en este campo, required no está diciendo que este campo es obligatorio.

A screenshot of the web form showing the first input field with the placeholder text "Introduce tu nombre" highlighted with a blue border. The other fields are "Hola", "Hola", "Two", an empty text field, and an "Enviar" button.

Línea 12 readonly le estamos diciendo que este campo es solo de lectura.

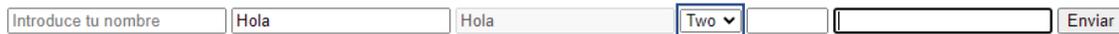
A screenshot of the web form showing the second input field with the value "Hola" highlighted with a blue border. The other fields are "Introduce tu nombre", "Hola", "Two", an empty text field, and an "Enviar" button.

Línea 13 disabled, le estamos diciendo que este campo no esté activo aunque lo seleccionemos.

A screenshot of the web form showing the third input field with the value "Hola" highlighted with a blue border. The other fields are "Introduce tu nombre", "Hola", "Two", an empty text field, and an "Enviar" button.

Línea 14, 15, 16 y 17 Estamos creando una lista que contendrá dos valores one y two.

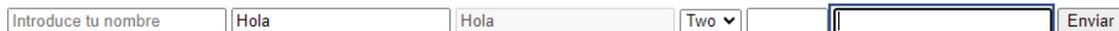
Línea 16 la opción por defecto que sea Two.

A screenshot of the web form showing the dropdown menu with "Two" selected and highlighted with a blue border. The other fields are "Introduce tu nombre", "Hola", "Hola", an empty text field, and an "Enviar" button.

Línea 18 Introducir un número pero no puede ser menor de 5 ni mayor de 10.

A screenshot of the web form showing the number input field highlighted with a blue border. The other fields are "Introduce tu nombre", "Hola", "Hola", "Two", an empty text field, and an "Enviar" button.

Línea 19 El texto no puede ser menor de 3 caracteres ni mayor de 5 caracteres, es un campo obligatorio y además al ejecutar el formulario habrá realizado un autofocus en este campo, ya podemos escribir para rellenarlo.

A screenshot of the web form showing the autofocus text input field highlighted with a blue border. The other fields are "Introduce tu nombre", "Hola", "Hola", "Two", an empty text field, and an "Enviar" button.

Línea 20 para enviar el formulario.

Envío GET vs POST

```

1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  |   <title>Get vs post</title>
8  </head>
9  <body>
10 |   <form action="enviar.html" method="GET">
11 |       <label for="name">Nombre</label>
12 |       <input type="text" id="name" name="name">
13 |       <label for="surname">Apellido</label>
14 |       <input type="text" id="surname" name="surname">
15 |       <input type="submit">
16 |
17 |   </form>
18 </body>
19 </html>

```

Le damos al botón enviar.

```

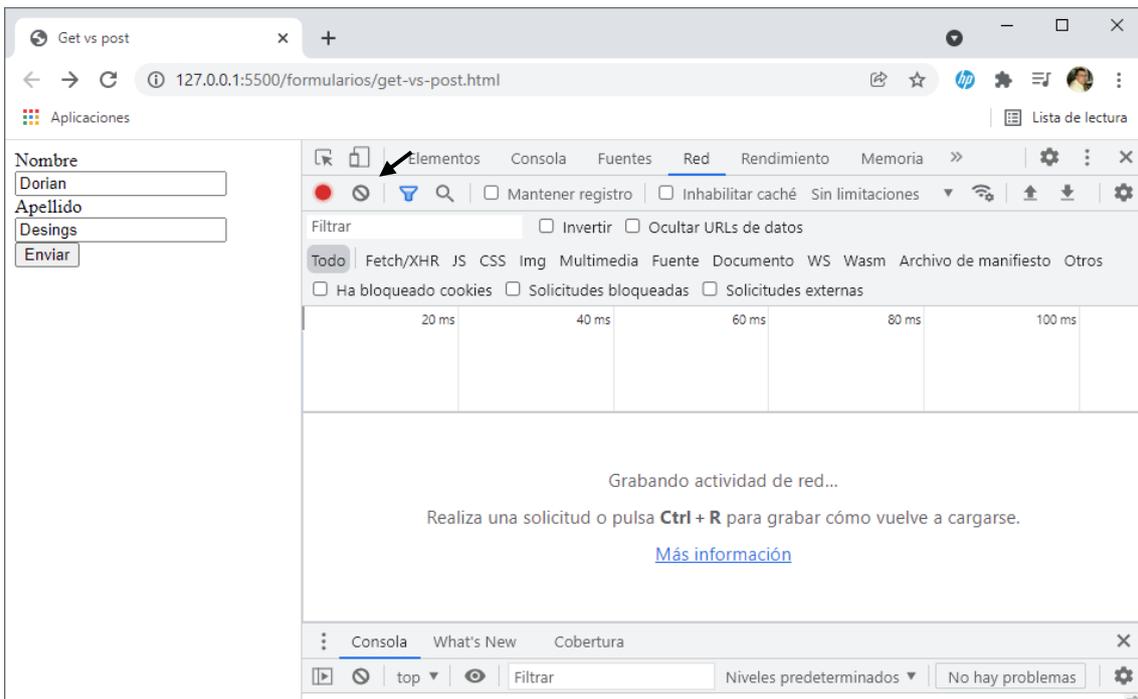
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  |   <title>Get vs post</title>
8  </head>
9  <body>
10 |   <form action="enviar.html" method="post">
11 |       <label for="name">Nombre</label>
12 |       <input type="text" id="name" name="name">
13 |       <label for="surname">Apellido</label>
14 |       <input type="text" id="surname" name="surname">

```

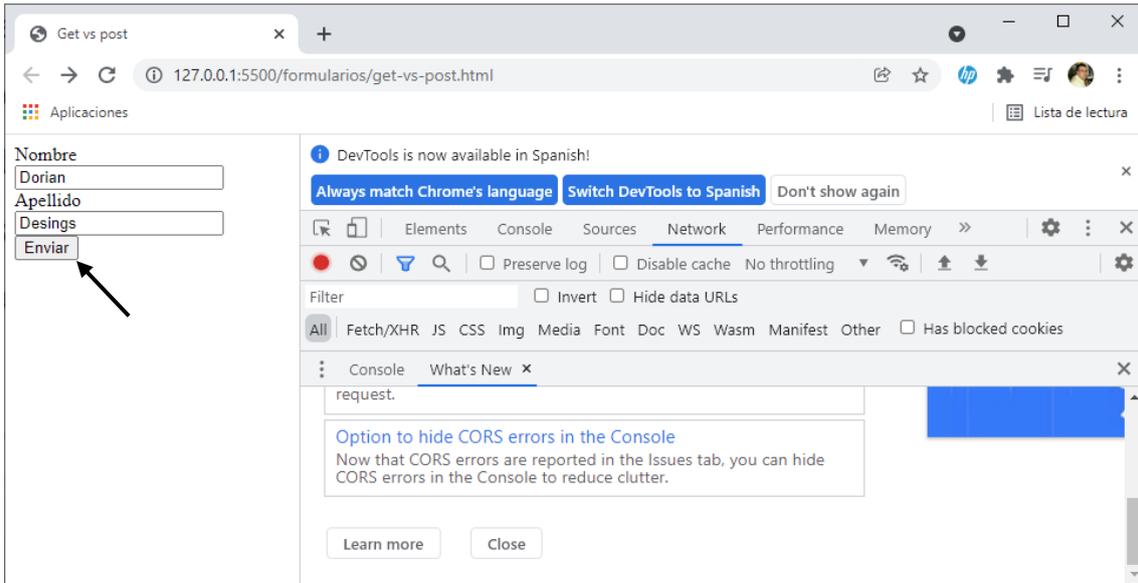
```
15 |     <input type="submit">
16 |
17 | </form>
18 </body>
19 </html>
```



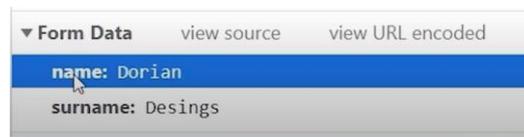
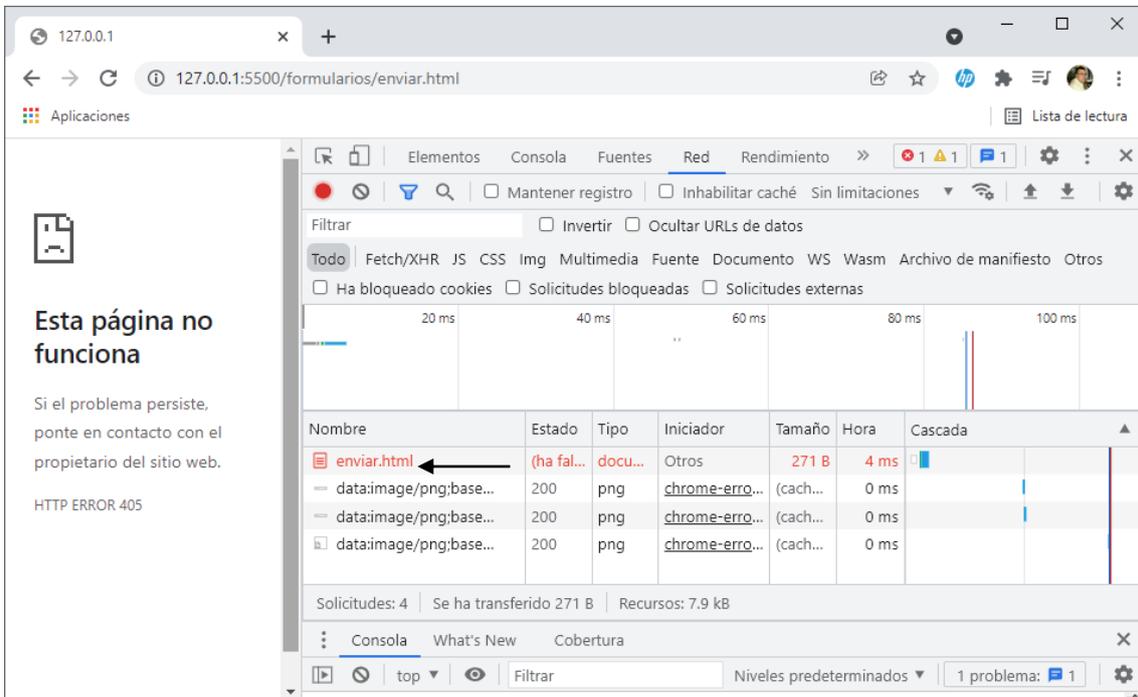
Le damos al botón de enviar.
Botón derecho / Inspeccionar.
Nos vamos a la pestaña Network (red)



Le damos a clear (borrar).



Enviamos los datos de nuevo.



Estos son los datos que estamos enviando.

¿Qué es el contenido embebido?

- El contenido embebido es todo el contenido que nos traemos desde fuera.
- Estos archivos son los que más peso (tamaño) añaden a un sitio web.
- Los tipos más conocidos son:
 - Imágenes
 - Audio
 - Vídeo

Iframes Imágenes:

- Los formatos de imágenes para web los podemos clasificar en 2 tipos:
 - Vectoriales
 - svg(recomendado siempre que se pueda)
 - Mapa de bits
 - jpg
 - png 8 y 24 (si necesitáis transparencia)
 - gif (si necesitáis una imagen animada)
 - webp (el formato que menos pesa)

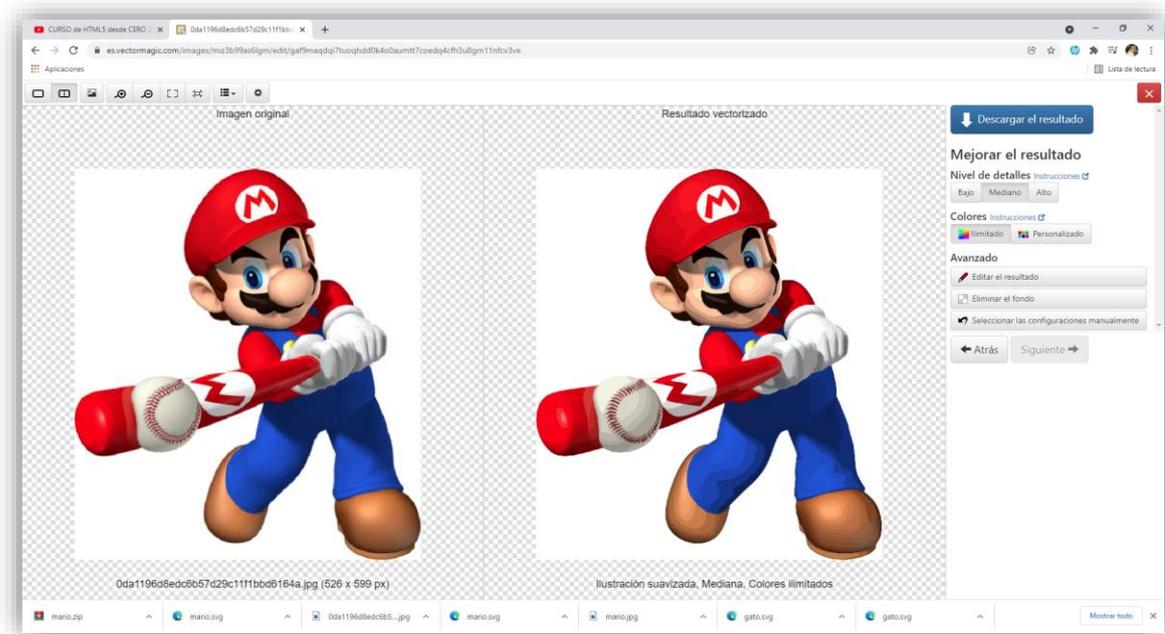
Imágenes de bits vs imágenes vectoriales

Una imagen de tipo bit si hacemos zoom veremos en un momento que se empieza a distorsionar.

Hay herramientas que nos permiten vectorizar imágenes con formato svg.

Los formatos svg son coordenadas para que el navegador pueda pintar.

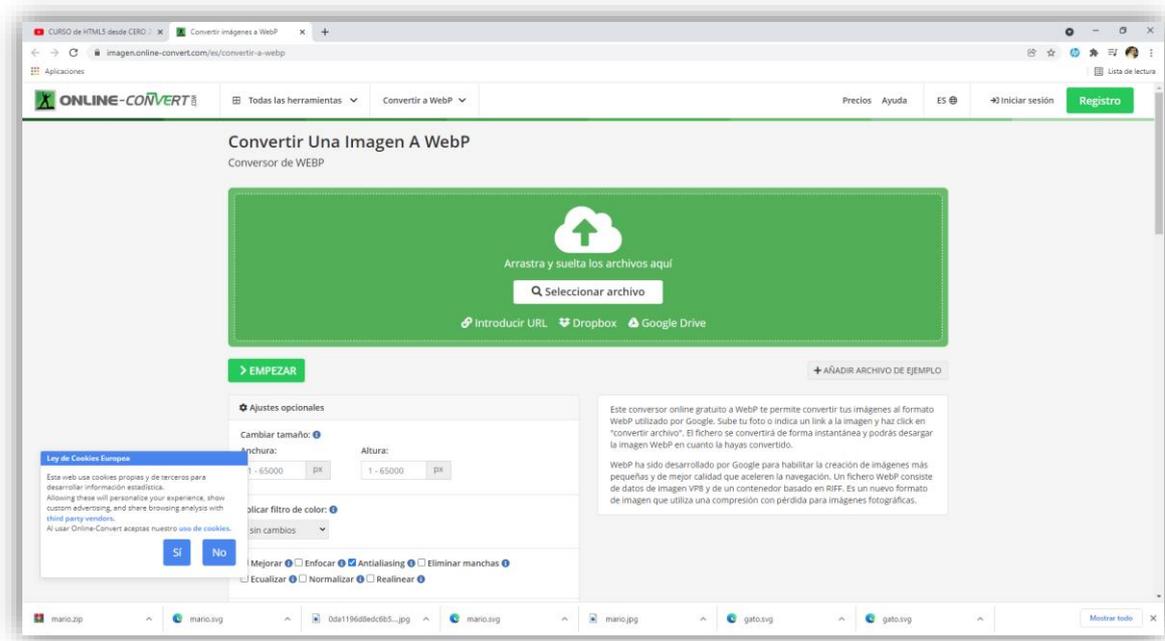
Una página web que lo convierte es: <https://es.vectormagic.com/>



Imágenes de bits WEBP

En el siguiente enlace encontraremos un convertor a WebP:

<https://imagen.online-convert.com/es/convertir-a-webp>

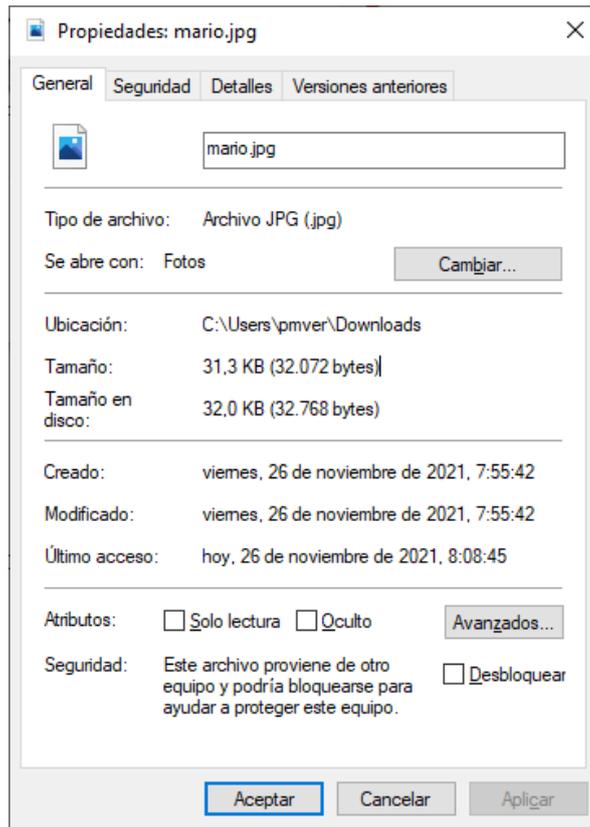


Soporta tanto jpg, png como gif.

Vamos a realizar un ejemplo:

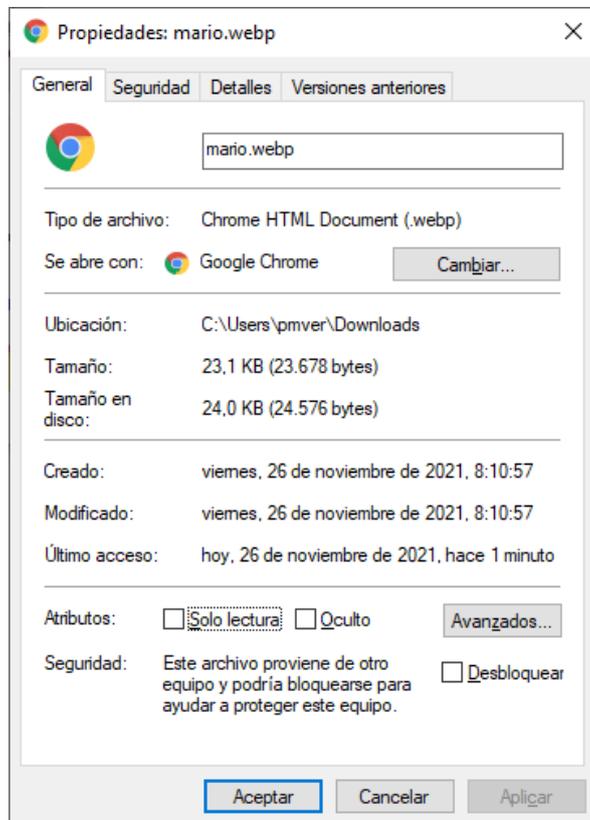


Tengo esta imagen con un tamaño de:

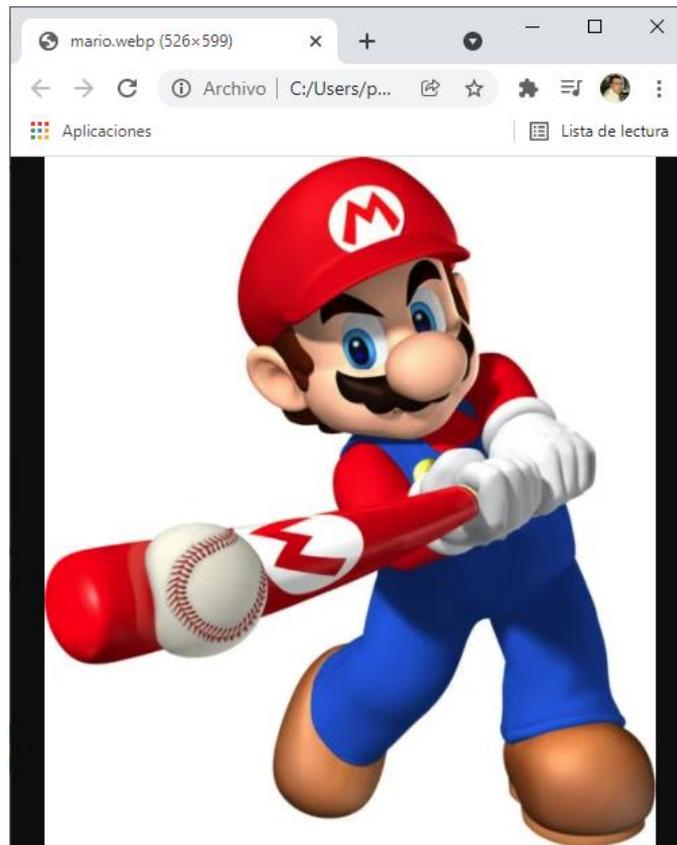


Ahora la vamos a convertir.

Nuestra segunda imagen tiene un tamaño:



De 32.072 bytes ahora la tenemos con un tamaño de 23.678 bytes, con el nuevo formato .webp.



Insertar imágenes en HTML

```
<> contenido-embebido.html > ...
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Contenido embebido</title>
8      <style>
9          img{
10             max-width: 100%;}
11     </style>
12 </head>
13 <body>
14     <h1>Contenido embebido</h1>
15     
16 </body>
17 </html>
```

Hemos agregado un estilo con css para modificar el tamaño de la imagen.

Este será el resultado:



```

```

Podemos seleccionar un archivo webp y el resultado sería exactamente el mismo.

```

```

El formato webp también permite exportar archivos gif.



```

```

Si en un momento no se puede mostrar la imagen que muestre el texto de la propiedad alt.

Contenido embebido



```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Contenido embebido</title>
  <style>
    |   img{
    |       |   max-width: 100%;}
  </style>
</head>
<body>
  <h1>Contenido embebido</h1>
  
</body>
</html>
```

Este será el resultado:



Otra forma de insertar una imagen svg es haciendo doble clic sobre la imagen para ver su contenido.

#YOAPRENDOCONDORIAN

assets > images > 733547.svg

```

1  <?xml version="1.0" encoding="iso-8859-1"?>
2  <!-- Generator: Adobe Illustrator 19.0.0, SVG Export Plug-In . SVG Version:
3  6.00 Build 0) -->
4  <svg version="1.1" id="Capa_1" xmlns="http://www.w3.org/2000/svg"
5  xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px"
6  viewBox="0 0 512 512" style="enable-background:new 0 0 512 512;"
7  xml:space="preserve">
8  <path style="fill:#1976D2;" d="M448,0H64C28.704,0,0,28.704,0,64v384c0,35.296,
9  28.704,64,64,64h384c35.296,0,64-28.704,64-64V64
10  C512,28.704,483.296,0,448,0z"/>
11 <path style="fill:#FAFAFA;" d="M432,256h-80v-64c0-17.664,14.336-16,
12 32-16h32V96h-64l0,0c-53.024,0-96,42.976-96,96v64h-64v80h64
13 v176h96V336h48L432,256z"/>
14 </svg>

```

La vamos a seleccionar y copiarla.

```
<> contenido-embedido.html > ...
```

```

1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4  <meta charset="UTF-8">
5  <meta http-equiv="X-UA-Compatible" content="IE=edge">
6  <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  <title>Contenido embebido</title>
8  <style>
9  |   img{
10 |     max-width: 100%;}
11 </style>
12 </head>
13 <body>
14 <h1>Contenido embebido</h1>
15 <svg version="1.1" id="Capa_1" xmlns="http://www.w3.org/2000/svg"
16 xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px"
17 viewBox="0 0 512 512" style="enable-background:new 0 0 512 512;"
18 xml:space="preserve">
19 <path style="fill:■ #1976D2;" d="M448,0H64C28.704,0,0,28.704,0,64v384c0,35.296,28.
20 704,64,64,64h384c35.296,0,64-28.704,64-64V64
21 C512,28.704,483.296,0,448,0z"/>
22 <path style="fill:□ #FAFAFA;" d="M432,256h-80v-64c0-17.664,14.336-16,
23 32-16h32V96h-64l0,0c-53.024,0-96,42.976-96,96v64h-64v80h64
24 v176h96V336h48L432,256z"/>
25 </svg>
26 </body>
27 </html>

```

Y veremos que el resultado es el mismo.

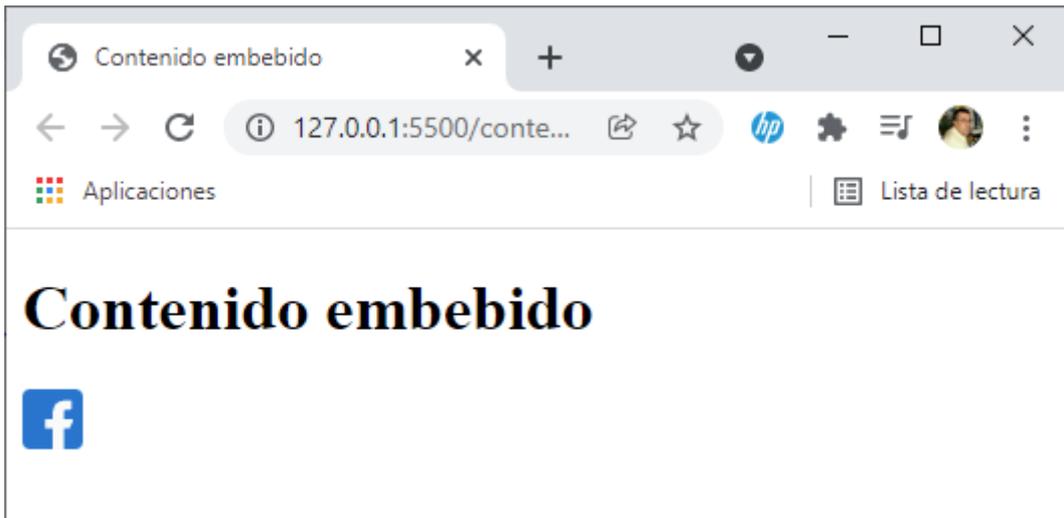


Si desde CSS o JavaScript tenemos que realizar modificaciones es mejor ponerlo de la segunda forma, sino solo lo que pretendemos en insertar la imagen con la primera opción también es válido.

Si solo pretendemos modificar el tamaño la primera opción es válida.

```
<> contenido-embebido.html > ...
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  |   <title>Contenido embebido</title>
8  |   <style>
9  |   |   img{
10 |   |   |   max-width: 100%;
11 |   |   |   width: 30px;
12 |   |   }
13 |   </style>
14 </head>
15 <body>
16 |   <h1>Contenido embebido</h1>
17 |   
18 </body>
19 </html>
```

Este será el resultado:



Device Pixel Ratio

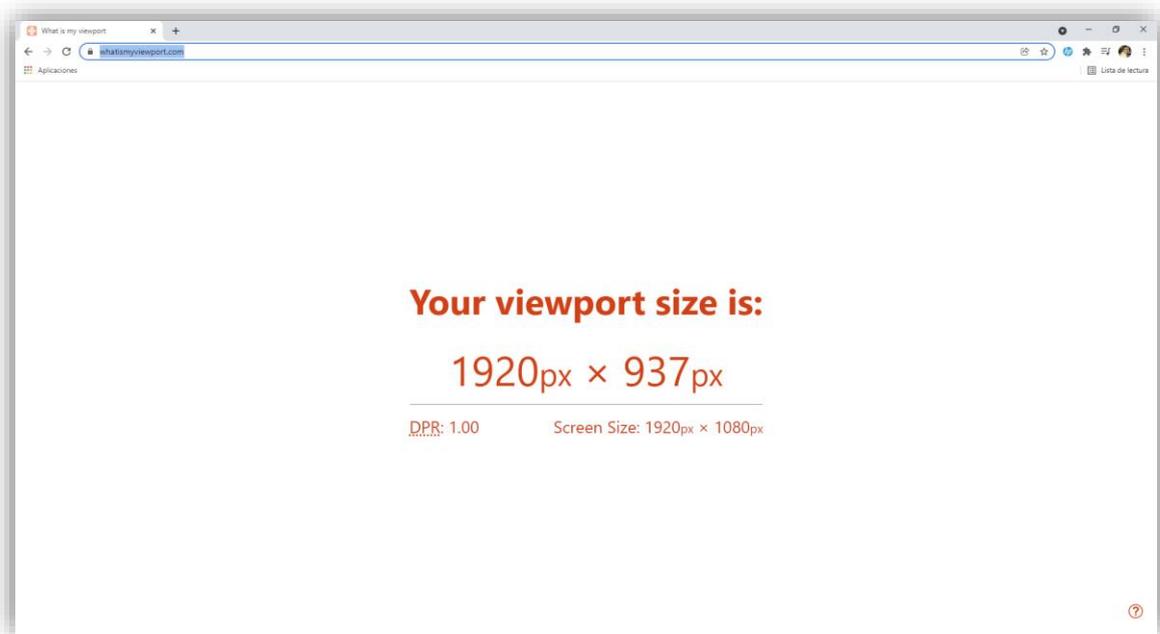
DPR (Device Pixel Ratio)

Es la relación que existe entre los píxeles reales que tienen el dispositivo y los píxeles que tenemos disponible para “pintar” contenido.

DPR = pixeles reales / pixeles disponibles.

Desde el siguiente enlace podremos saber la resolución de nuestra pantalla o dispositivo.

<https://whatismyviewport.com/>



Nuestra pantalla es 1920 x 1080 de alto pero el navegador se come una parte por la parte superior y otra parte por la inferior de aquí que en lugar de 1080 sea 937.

Si vistamos esta página desde un móvil.



Este es el espacio que tenemos para dibujar si lo queremos ver desde el móvil.

<> contenido-embebido.html > ...

```

1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Contenido embebido</title>
8      <style>
9          |   img{
10         |       |   max-width: 100%;
11         |       |   }
12         |   </style>
13 </head>
14 <body>
15     <h1>Contenido embebido</h1>
16     
17 </body>
18 </html>

```

Este será el resultado:



En el próximo capítulo vamos a ver como podemos ajustar una imagen a una pantalla de ordenador o a un móvil.

Atributo srcset

Para este capítulo vamos a trabajar con dos imágenes una portada para el ordenador y otra para el móvil con un tamaño más pequeño.

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Contenido embebido</title>
8      <style>
9          img{
10             max-width: 100%;
11         }
12     </style>
13 </head>
14 <body>
15     <img srcset="
16         /assets/images/portada.webp,
17         /assets/images/portada-mobile.webp 3x" alt="">
18 </body>
19 </html>
```

En la línea 15 para cargar la imagen ponemos srcset en lugar de src, agregamos las dos imágenes separadas por coma y como la segunda imagen es la del móvil ponemos 3x.



De este modo si cargamos la página web para modo escritorio cargará la primera imagen y si cargamos la página web en un móvil cargará la segunda imagen que tiene otras dimensiones.



Portada de escritorio



Portada de móvil

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Contenido embebido</title>
8
9 </head>
10 <body>
11   
15 </body>
16 </html>
```

Si algún navegados como el Internet Explorer no soporta srcset agregaremos src como respaldo para que pueda cargar la imagen portada.jpg.

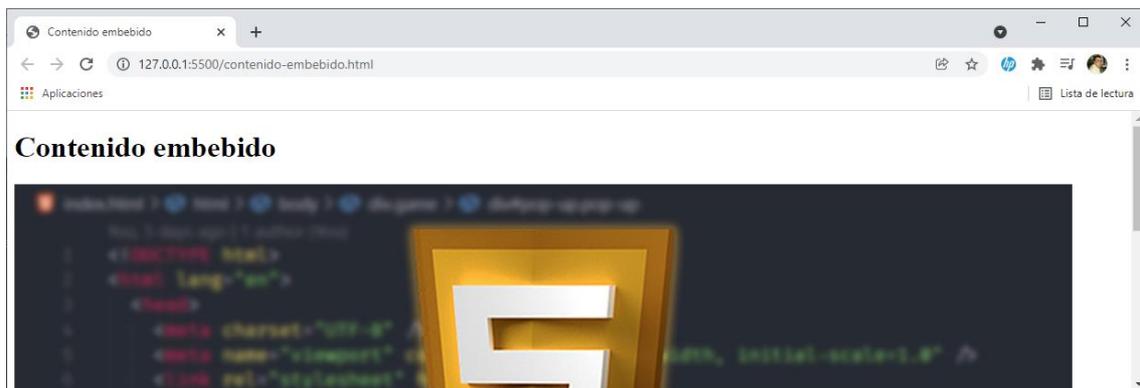
Etiqueta Picture

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Contenido embebido</title>
8
9 </head>
10 <body>
11   <h1>Contenido embebido</h1>
12   <picture>
13     <source srcset="assets/images/portada-mobile.webp" media="(max-width: 1200px)">
14     
15   </picture>
16 </body>
17 </html>
```

En la línea 13 le estamos diciendo que si la ventana del navegador tiene un máximo de 1200px muestre la portada-mobile.webp de lo contrario muestra portada.jpg que es la de respaldo y tiene que ser obligatoria que esté.



Vamos a hacerla más ancha hasta que supere los 1200 px.

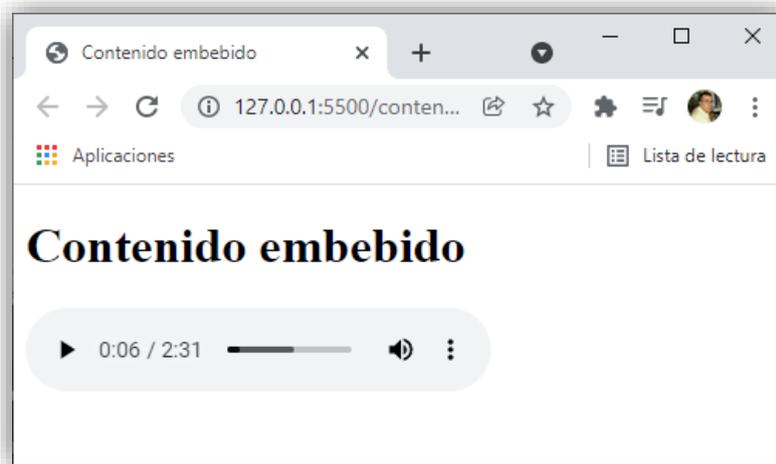


A cargado la segunda imagen.

Etiqueta Audio

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Contenido embebido</title>
8
9 </head>
10 <body>
11   <h1>Contenido embebido</h1>
12   <audio src="assets/audio/oomiee.mp3" controls></audio>
13 </body>
14 </html>
```

Este será el resultado:



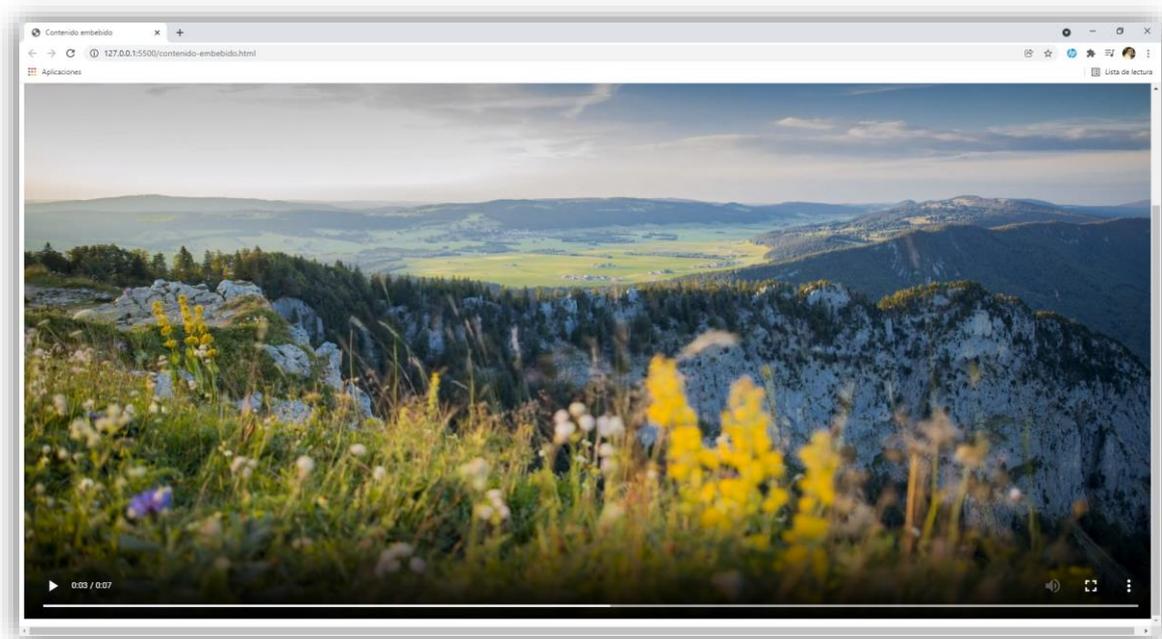
```
</body>
|   <h1>Contenido embebido</h1>
|   <audio src="assets/audio/oomiee.mp3" controls autoplay muted></audio>
</body>
```

Con autoplay el reproductor se reproduce automáticamente al cargar la página y muted que está en silencio, en muchos casos el autoplay no se reproduce para no molestar al usuario.

Etiqueta video

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Contenido embebido</title>
8
9 </head>
10 <body>
11   <h1>Contenido embebido</h1>
12   <video src="assets/video/Creux De Van.mp4" controls></video>
13 </body>
14 </html>
```

Este será el resultado:



Para que el video no se desborde.

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Contenido embebido</title>
8   <style>
9     video{
10       max-width: 100%;
11     }
```

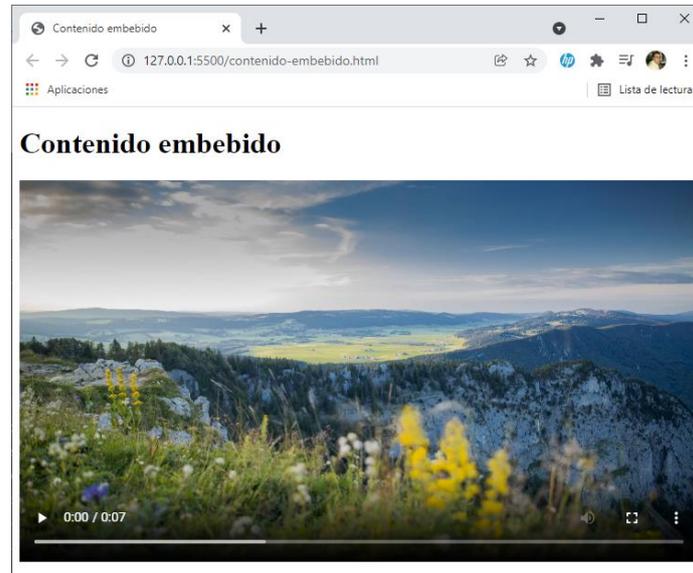
#YOAPRENDOCONDORIAN

```

12     </style>
13 </head>
14 <body>
15     <h1>Contenido embebido</h1>
16     <video src="assets/video/Creux De Van.mp4" controls></video>
17 </body>
18 </html>

```

Agregamos un estilo css.



```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Contenido embebido</title>
8     <style>
9         video{
10             max-width: 100%;
11         }
12     </style>
13 </head>
14 <body>
15     <h1>Contenido embebido</h1>
16     <video src="assets/video/Creux De Van.mp4" controls autoplay muted></video>
17 </body>
18 </html>

```

En la línea 16 si queremos que se reproduzca automáticamente añadiremos autoplay y que no se reproduzca el sonido muted.

El atributo loop que también esta disponible en audio hace que se reproduzca continuamente.

```

<body>
    <h1>Contenido embebido</h1>
    <video src="assets/video/Creux De Van.mp4" controls poster="assets/images/portada.jpg"></video>
</body>

```

La propiedad poster hace que mientras no se reproduzca aparezca una imagen.

Contenido embebido

127.0.0.1:5500/contenido-embecido.html

Contenido embebido



0:00 / 0:07

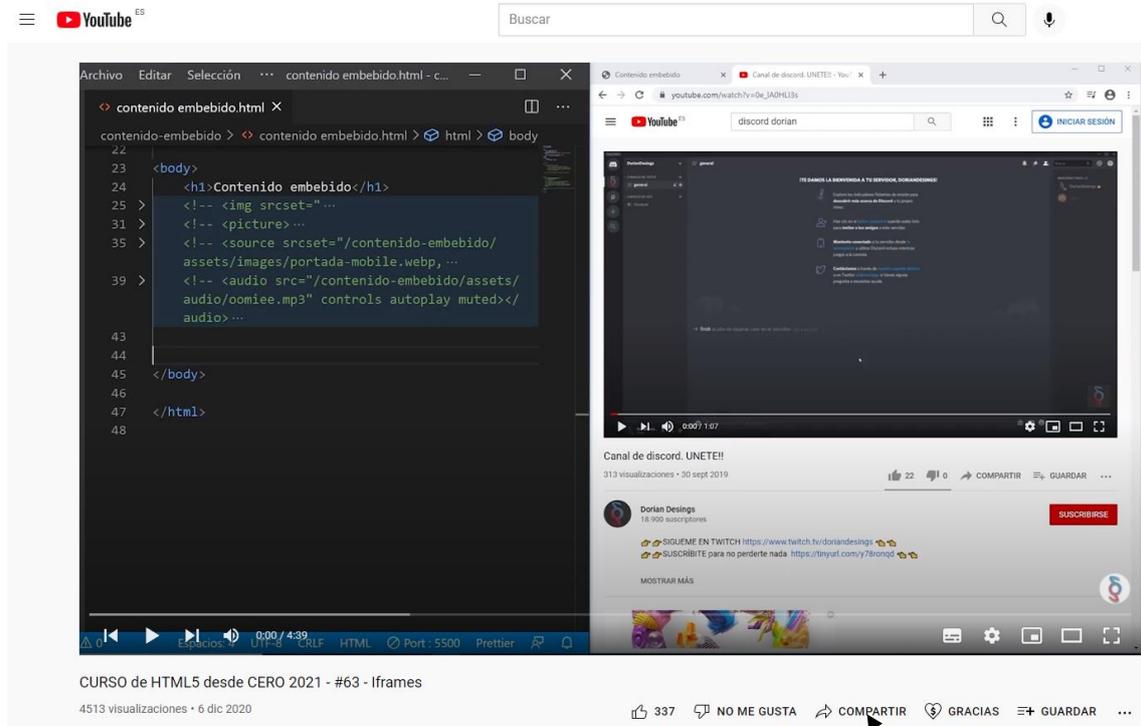
The image shows a browser window with an embedded video player. The browser's address bar displays '127.0.0.1:5500/contenido-embecido.html'. The video player's title card features a large, 3D-style white letter 'S' inside a gold shield-like border. Below the logo, the text 'HTML5 DESDE CERO' is written in a bold, yellow, sans-serif font. At the bottom of the title card, the years '2020-2021' are visible in a lighter, semi-transparent font. The video player interface includes a play button, a progress bar showing '0:00 / 0:07', and standard volume and full-screen controls.

Iframes

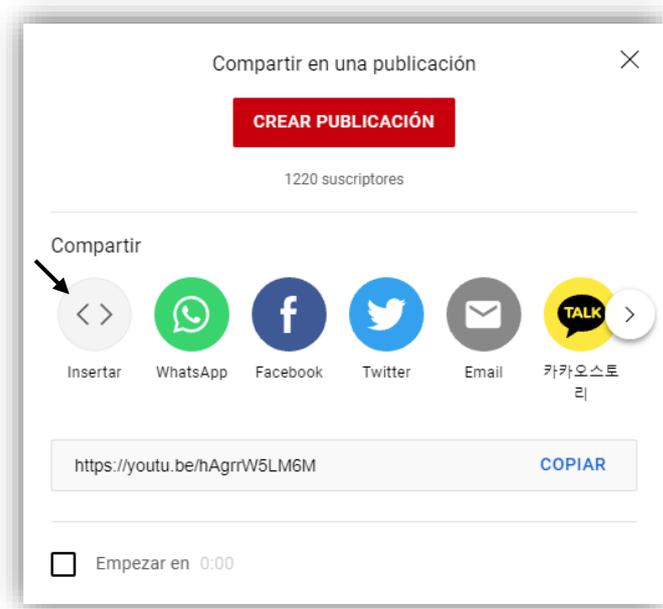
Coger una web e introducirla en la nuestra.

Para entender correctamente este capítulo nos vamos a YouTube y queremos ver un video en nuestra página web.

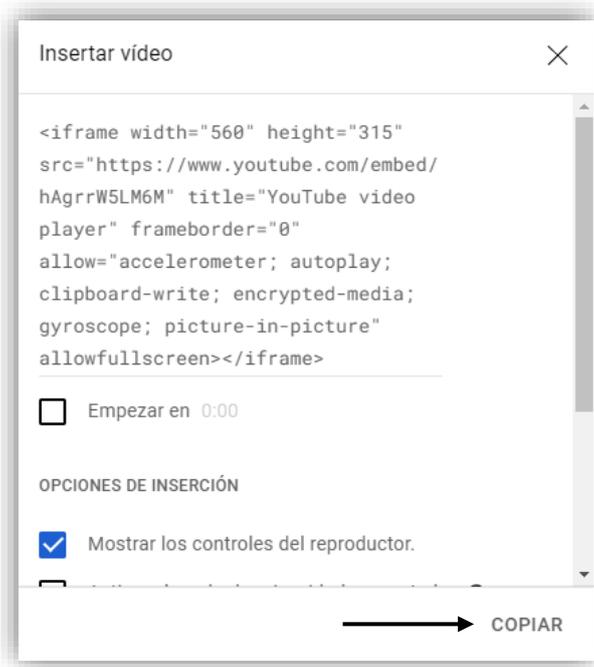
Una vez tenemos el video seleccionado nos iremos a compartir.



En la parte inferior seleccionaremos compartir.



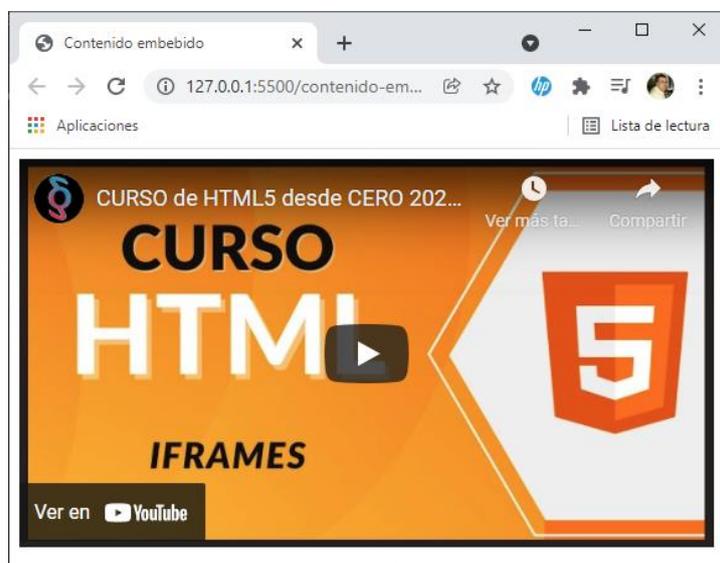
Seleccionaremos insertar.



Seleccionaremos el botón COPIAR.
Ahora lo vamos a pegar un una nueva página.

```
<> contenido-embedido.html > ...
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Contenido embebido</title>
8 </head>
9 <body>
10   <iframe width="560" height="315" src="https://www.youtube.com/embed/hAgrW5LM6M" title="YouTube video player"
    frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture
    allowfullscreen"></iframe>
11 </body>
12 </html>
```

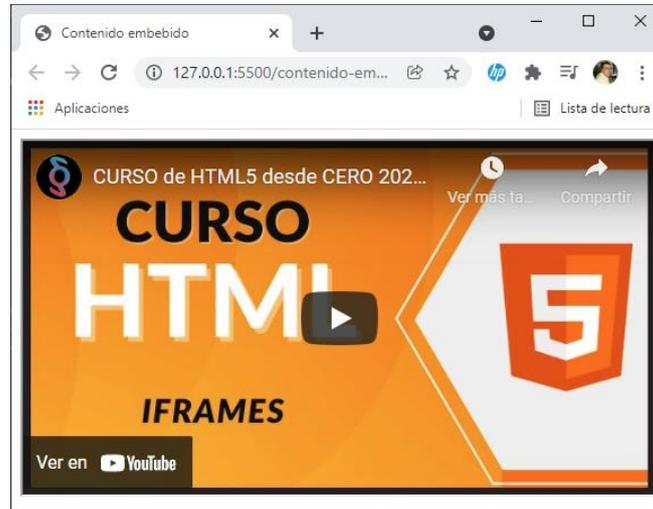
Este será el resultado:



#YOAPRENDOCONDORIAN

```
<body>
|   <iframe width="560" height="315" src="https://www.youtube.com/embed/hAgrw5LM6M" </iframe>
</body>
```

Hemos eliminado parte del código insertado en el iframe, y podrás comprobar que este funciona correctamente.



No es una buena idea cargar nuestras páginas con muchos iframes, esto ralentiza el cargado de nuestra página.

Etiqueta Figure

Se utiliza siempre que se quiera romper el flujo del contenido con una imagen.

```
<body>
|   <h1>Gatos</h1>
|   <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Blanditiis, accusantium iure eius autem ad ipsum
|   aspernatur, expedita, sapiente quisquam perspiciatis doloribus aut sint alias! Ullam velit nobis non natus iure!</p>
|   <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Aliquam ullam accusamus culpa tenetur natus libero
|   architecto. Voluptates temporibus commodi, doloremque voluptatum sequi, veritatis praesentium, pariatur aspernatur ipsa
|   impedit animi laudantium.</p>
|
|   <figure>
|     
|     <figcaption>Gato con una mirada muy tierna</figcaption>
|   </figure>
|   <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Magni molestias, quas deserunt consequuntur omnis
|   molestiae blanditiis culpa cumque facere consectetur eligendi sit, aut dolorem! Commodi quos vel quaerat tempore et.</p>
</body>
```

Dentro de la etiqueta `<figure>` `</figure>` insertamos una imagen agregando su url, con esta opción le estamos diciendo que entre párrafos vamos a insertar una imagen, además con `<figcaption>` `</figcaption>` podemos agregar un título para la imagen (pie de foto).



Etiquetas meta

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

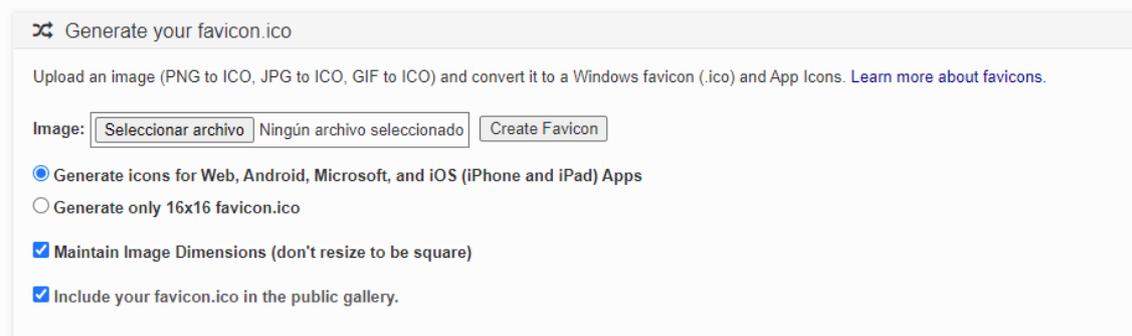
Cuando trabajamos con diseños que se adaptan a móviles esto permite que funcione todo el tema de adaptabilidad.

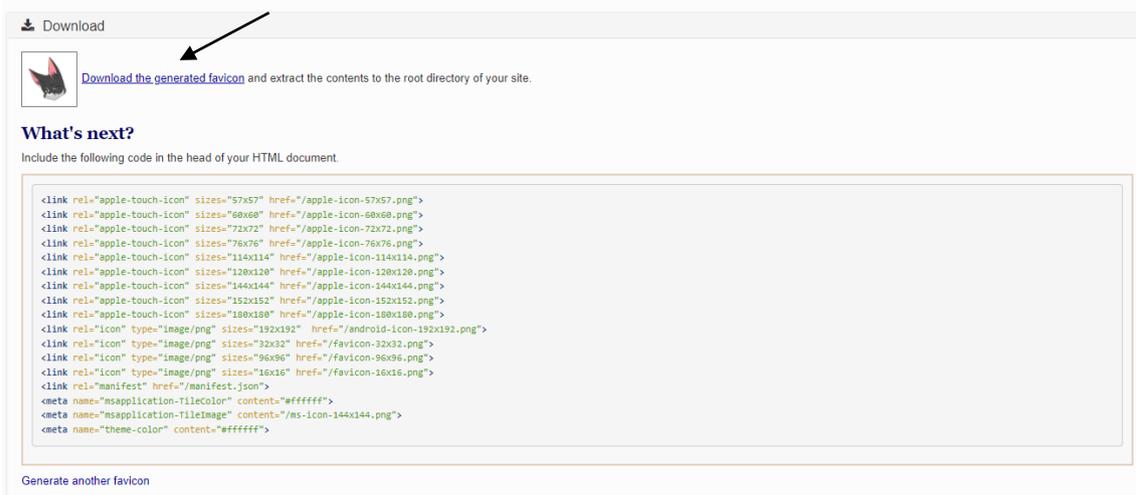
```
<meta name="description" content="El sitio MDN Web Docs proporciona información sobre tecnologías Open Web, incluidas HTML, CSS y API, tanto para sitios web como para aplicaciones web progresivas.">
```

```
<meta name="autor" content="DorianDesings">
```

Favicon – Creación y uso

Vamos a ir al siguiente enlace: <https://www.favicon-generator.org/>





Descargamos el siguiente enlace contiene todos los iconos para todos los dispositivos. Una vez descargados los guardamos en una carpeta llamada favicons dentro de nuestro proyecto web.

A generado el siguiente código que copiaremos.

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <link rel="apple-touch-icon" sizes="57x57" href="/apple-icon-57x57.png">
8     <link rel="apple-touch-icon" sizes="60x60" href="/apple-icon-60x60.png">
9     <link rel="apple-touch-icon" sizes="72x72" href="/apple-icon-72x72.png">
10    <link rel="apple-touch-icon" sizes="76x76" href="/apple-icon-76x76.png">
11    <link rel="apple-touch-icon" sizes="114x114" href="/apple-icon-114x114.png">
12    <link rel="apple-touch-icon" sizes="120x120" href="/apple-icon-120x120.png">
13    <link rel="apple-touch-icon" sizes="144x144" href="/apple-icon-144x144.png">
14    <link rel="apple-touch-icon" sizes="152x152" href="/apple-icon-152x152.png">
15    <link rel="apple-touch-icon" sizes="180x180" href="/apple-icon-180x180.png">
16    <link rel="icon" type="image/png" sizes="192x192" href="/android-icon-192x192.png">
17    <link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
18    <link rel="icon" type="image/png" sizes="96x96" href="/favicon-96x96.png">
19    <link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png">
20    <link rel="manifest" href="/manifest.json">
21    <meta name="msapplication-TileColor" content="#ffffff">
22    <meta name="msapplication-TileImage" content="/ms-icon-144x144.png">
23    <meta name="theme-color" content="#ffffff">
24
25    <title>Contenido embebido</title>
26 </head>
27 <body>
28
29 </body>
```

Lo hemos pegado dentro de las etiquetas <head> </Head>

#YOAPRENDOCONDORIAN

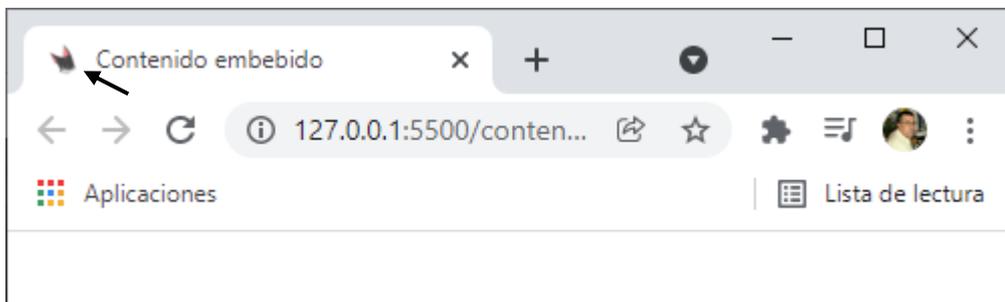
```

<link rel="apple-touch-icon" sizes="57x57" href="favicons/apple-icon-57x57.png">
<link rel="apple-touch-icon" sizes="60x60" href="favicons/apple-icon-60x60.png">
<link rel="apple-touch-icon" sizes="72x72" href="favicons/apple-icon-72x72.png">
<link rel="apple-touch-icon" sizes="76x76" href="favicons/apple-icon-76x76.png">
<link rel="apple-touch-icon" sizes="114x114" href="favicons/apple-icon-114x114.png">
<link rel="apple-touch-icon" sizes="120x120" href="favicons/apple-icon-120x120.png">
<link rel="apple-touch-icon" sizes="144x144" href="favicons/apple-icon-144x144.png">
<link rel="apple-touch-icon" sizes="152x152" href="favicons/apple-icon-152x152.png">
<link rel="apple-touch-icon" sizes="180x180" href="favicons/apple-icon-180x180.png">
<link rel="icon" type="image/png" sizes="192x192" href="favicons/android-icon-192x192.png">
<link rel="icon" type="image/png" sizes="32x32" href="favicons/favicon-32x32.png">
<link rel="icon" type="image/png" sizes="96x96" href="favicons/favicon-96x96.png">
<link rel="icon" type="image/png" sizes="16x16" href="favicons/favicon-16x16.png">
<link rel="manifest" href="/manifest.json">
<meta name="msapplication-TileColor" content="#ffffff">
<meta name="msapplication-TileImage" content="/ms-icon-144x144.png">
<meta name="theme-color" content="#ffffff">

```

Le decimos que los iconos están en la carpeta favicons.

Cuando ejecutemos nuestra página web, este será el resultado:

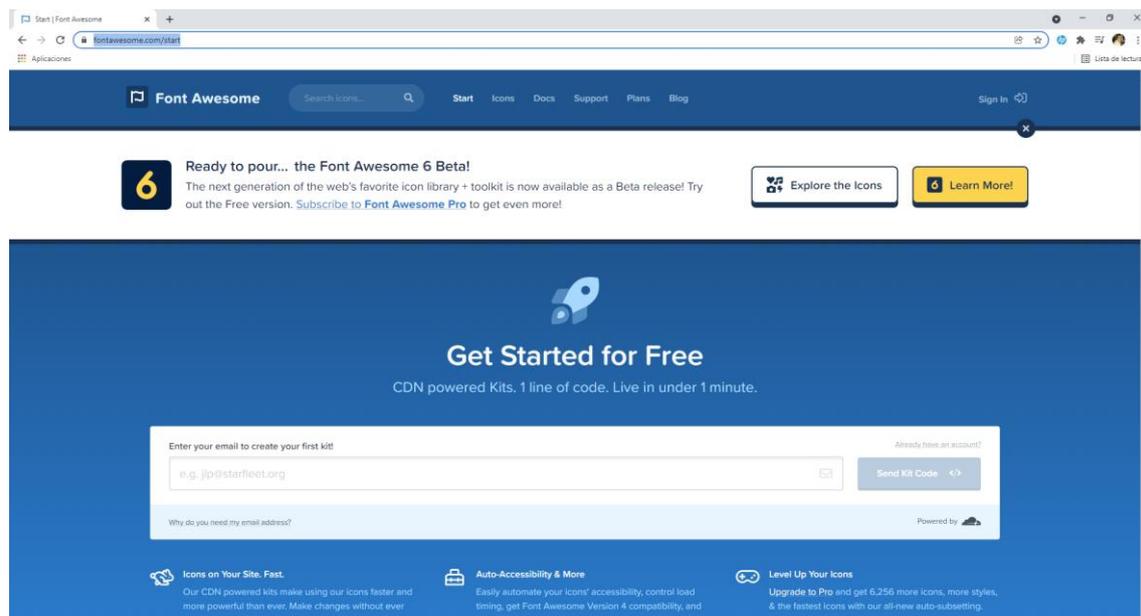


Estos últimas 4 líneas las podemos eliminar y para este ejercicio funcionaria igual.

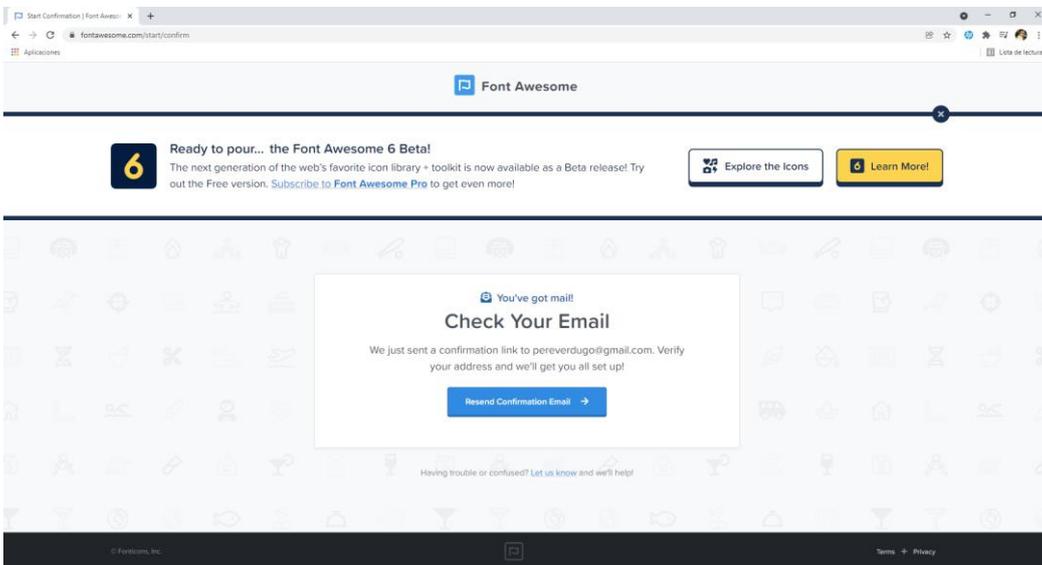
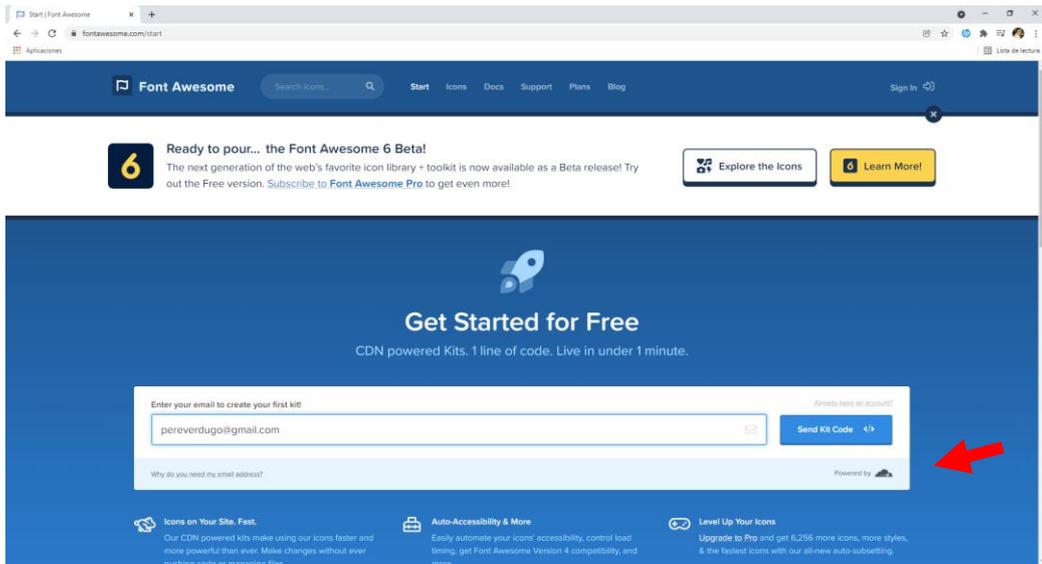
Fontawesome

Vamos a una página web donde encontraremos una librería de iconos.

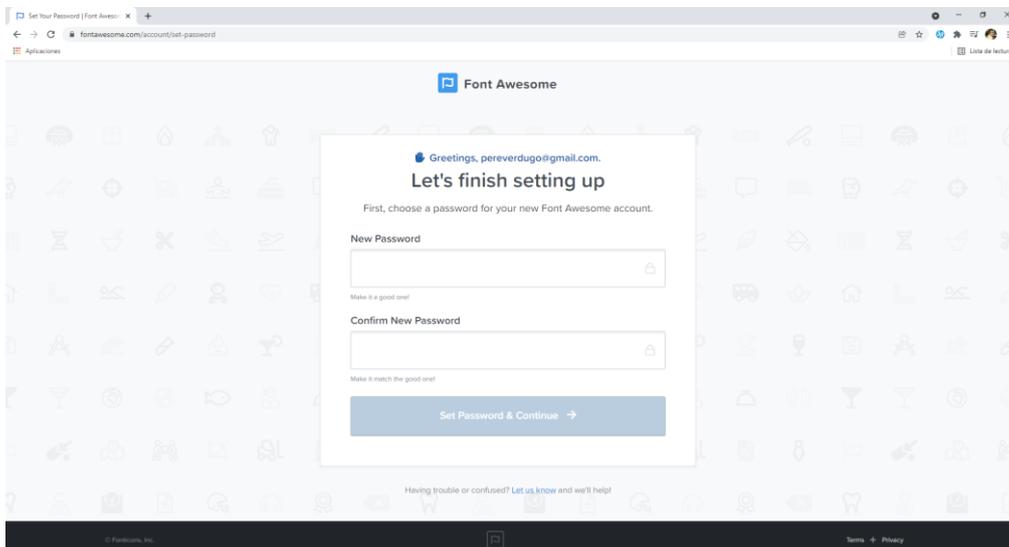
<https://fontawesome.com/start>



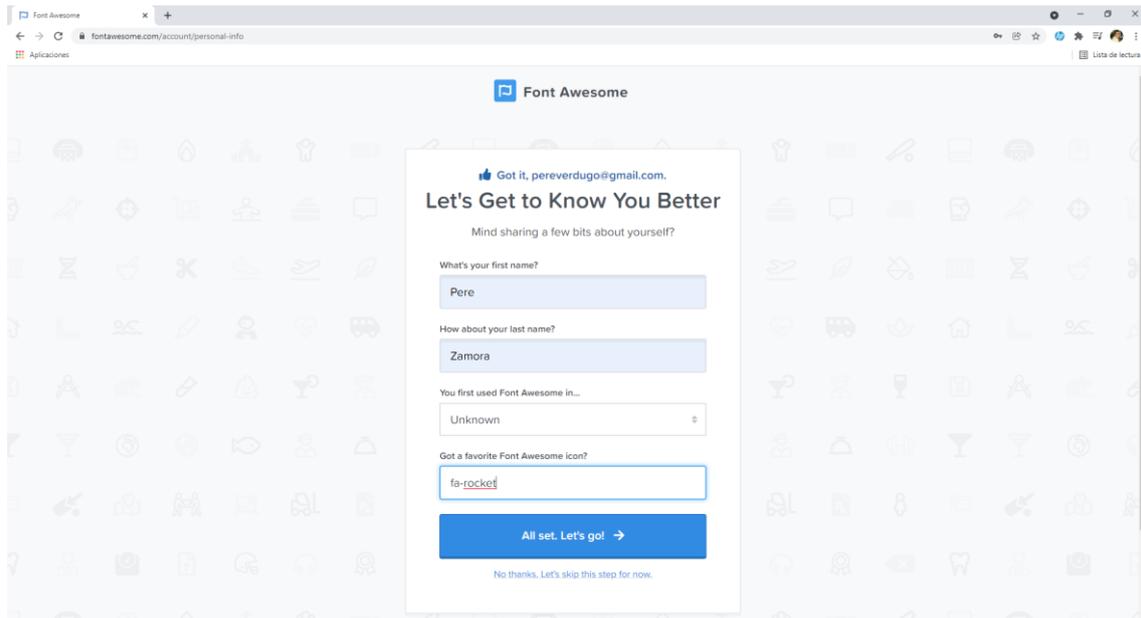
Nos pide que introduzcamos nuestro correo electrónico.



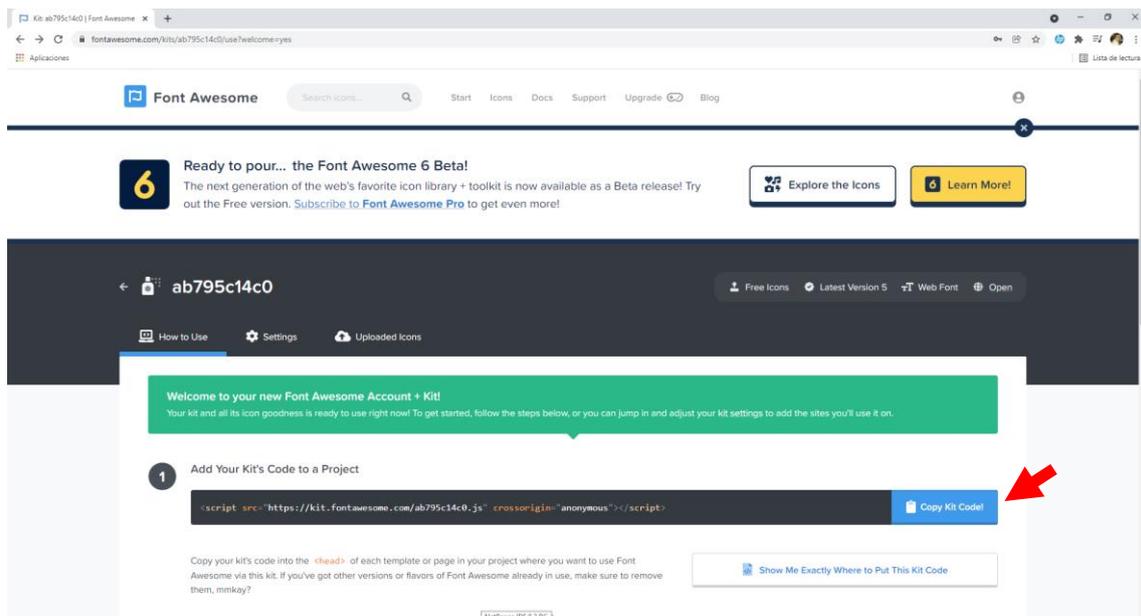
Nos dice que vallamos a mirar nuestro email, para confirmar nuestro correo.



Nos pide que introduzcamos un password.



Introducimos los datos.



Seleccionamos el botón de copiar código.

`<script src="https://kit.fontawesome.com/ab795c14c0.js" crossorigin="anonymous"></script>`

```
<> iconos.html > html > head > script
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <script src="https://kit.fontawesome.com/ab795c14c0.js" crossorigin="anonymous"></script>
8   <title>Document</title>
9 </head>
10 <body>
```

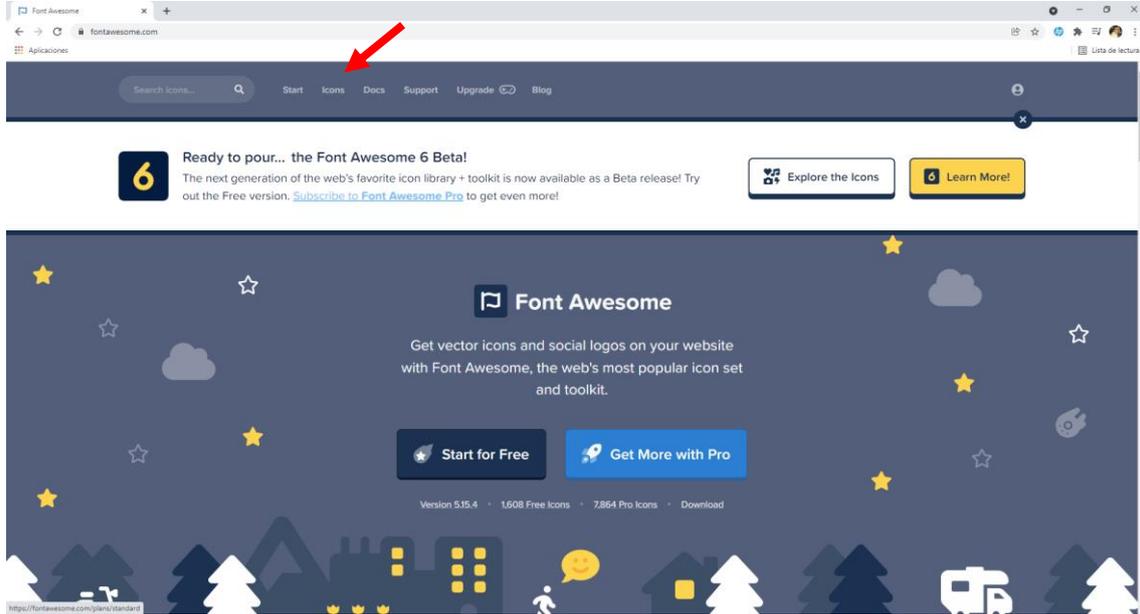
#YOAPRENDOCONDORIAN

```

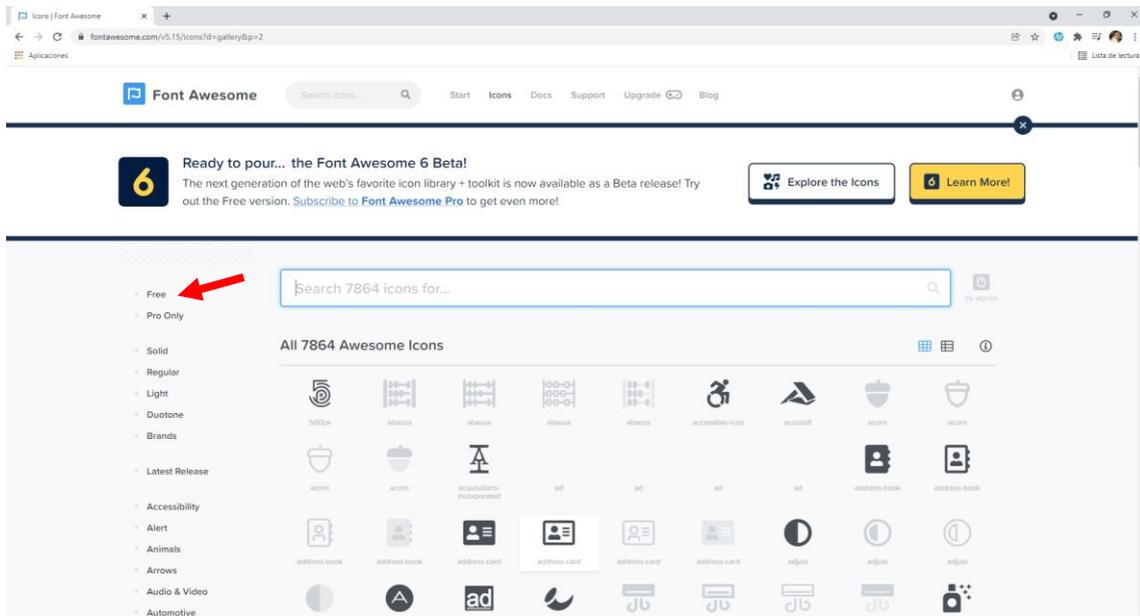
11
12 </body>
13 </html>

```

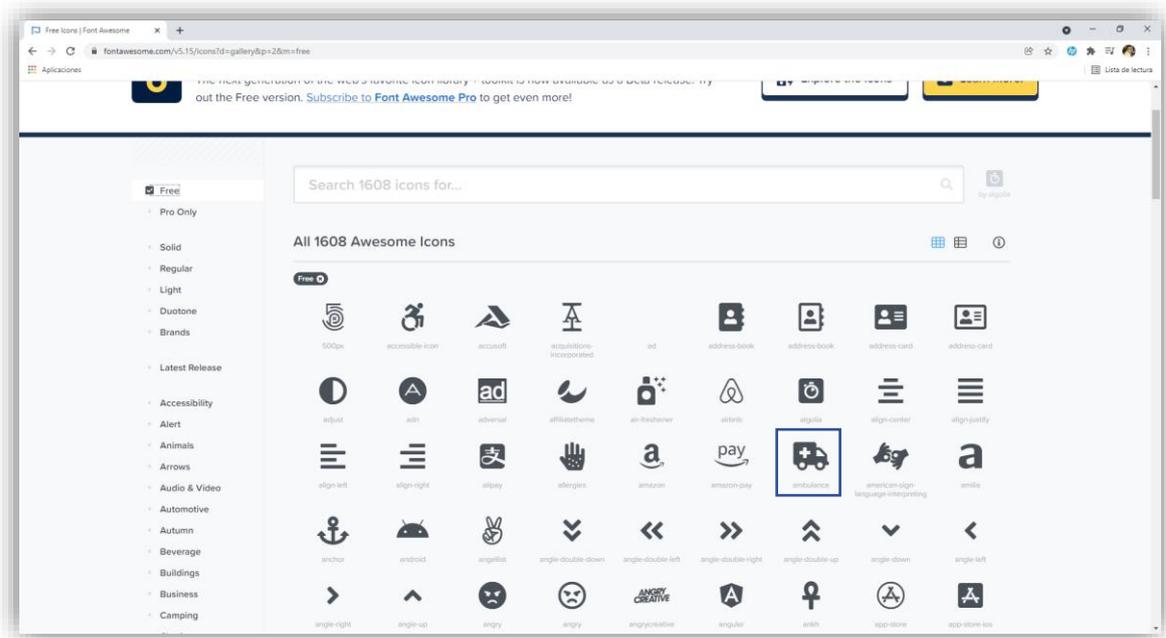
Esta línea nos da acceso a la librería.
 Nos vamos a la pagina principal de los iconos



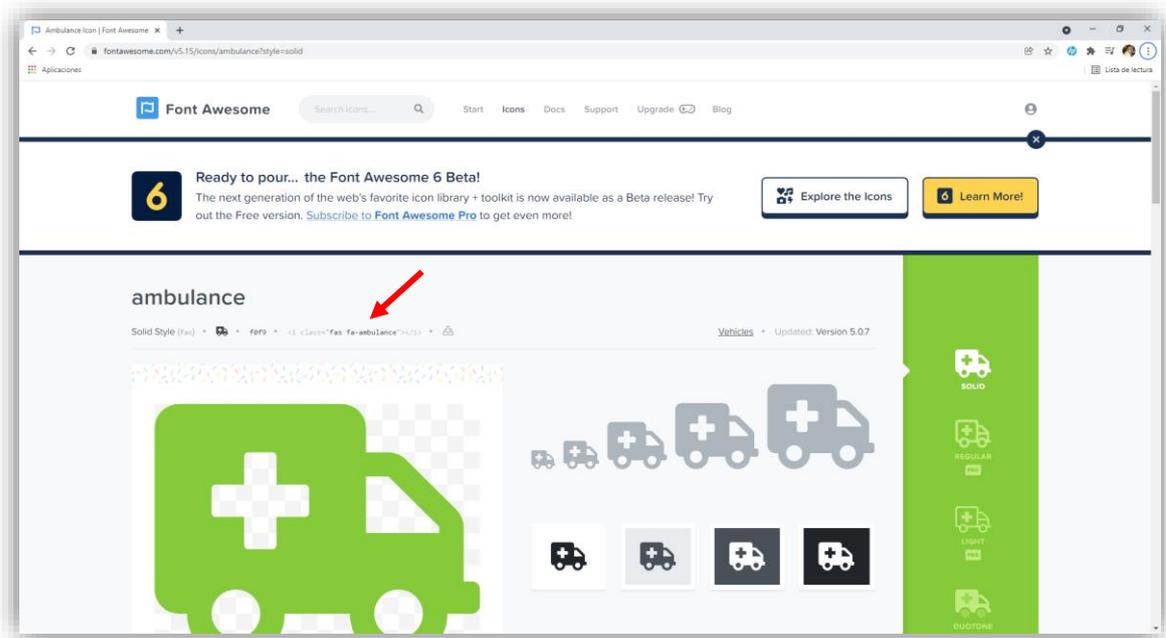
Seleccionamos Icons.



Solo podemos utilizar los iconos que están más oscuros, si pulsamos a free veremos los que tenemos acceso de forma gratuita.



Vamos a seleccionar el camión ambulancia.

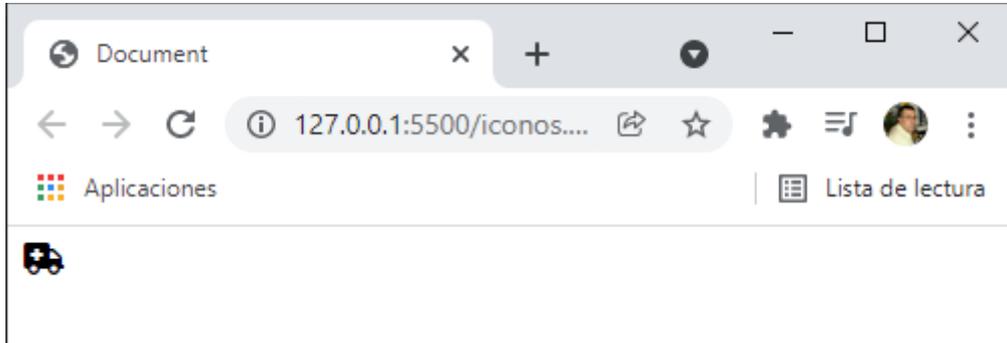


Copiamos el código.

```
<> iconos.html > html > body > i.fas.fa-ambulance
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <script src="https://kit.fontawesome.com/ab795c14c0.js" crossorigin="anonymous"></script>
8   <title>Document</title>
```

```
9 </head>
10 <body>
11 <i class="fas fa-ambulance"></i>
12 </body>
13 </html>
```

Pegamos el código.



Ya tenemos nuestro icono.

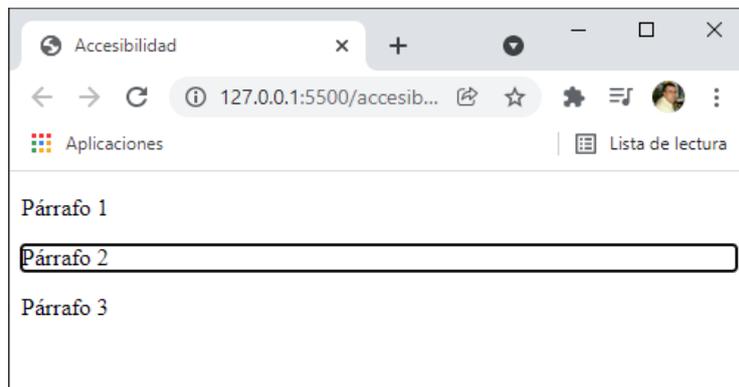
Atributos de accesibilidad

Es el hecho de hacer una página web accesible para todo el mundo, independientemente de si pueden verla o no.

Esto está pensado para persona invidentes.

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta charset="UTF-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>Accesibilidad</title>
8 </head>
9 <body>
10 <p tabindex="1">Párrafo 1</p>
11 <p tabindex="2">Párrafo 2</p>
12 <p tabindex="3">Párrafo 3</p>
13 </body>
14 </html>
```

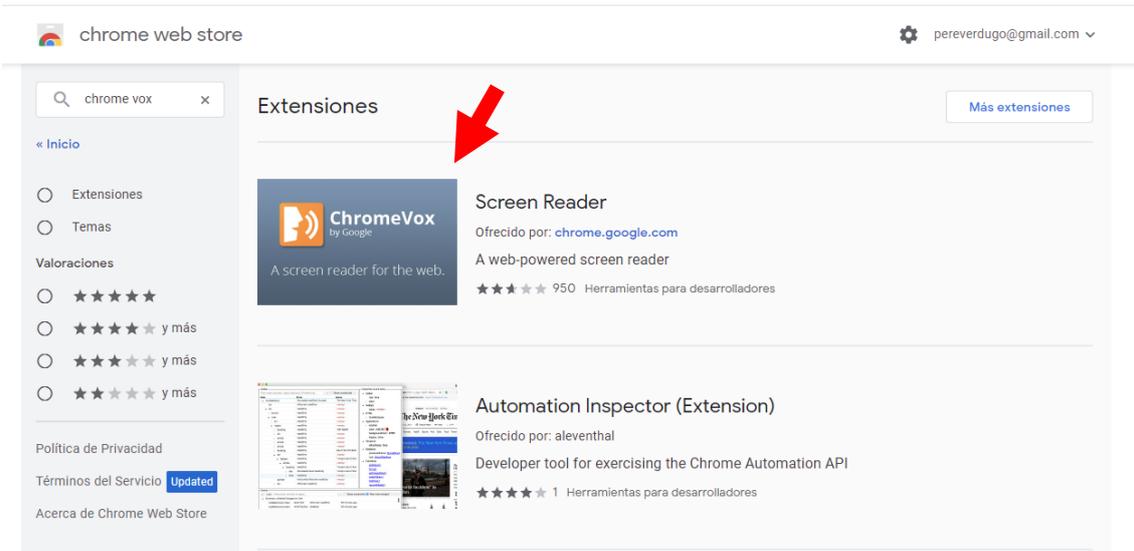
tabindex -> permite movernos por el navegador mediante la tecla tab.



#YOAPRENDOCONDORIAN

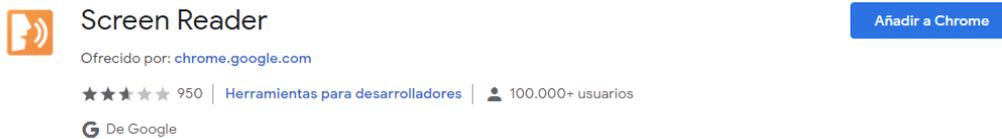
Para que un invidente pueda escuchar el contenido de una página web.

Nos iremos a Chrome web store y buscaremos Chrome vox.

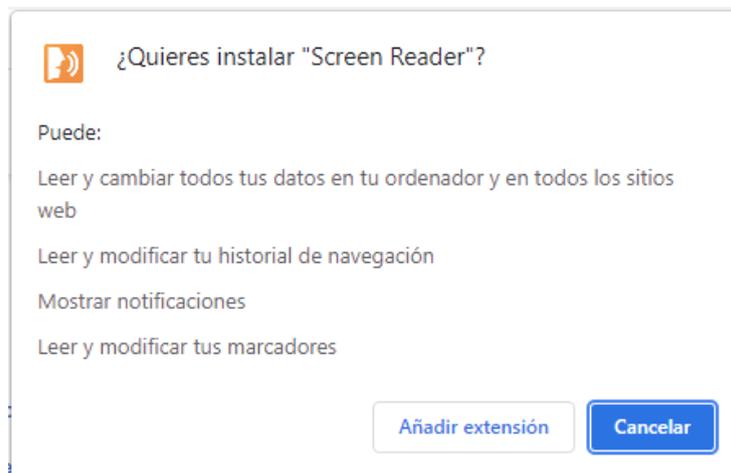


Haremos un clic.

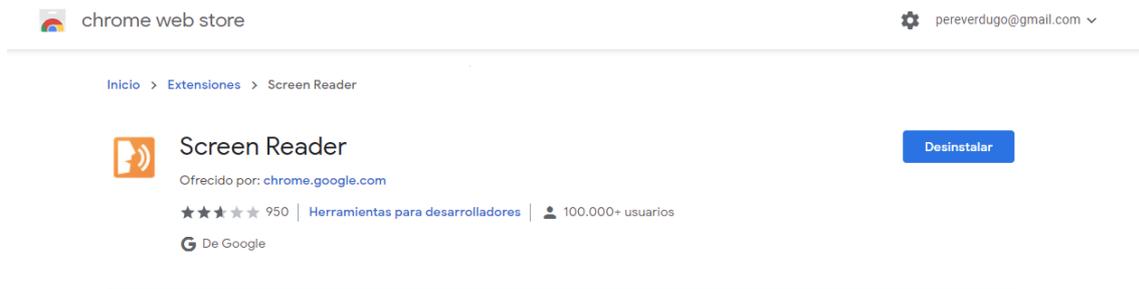
Inicio > Extensiones > Screen Reader



Seleccionaremos "Añadir a Chrome".



Seleccionaremos Añadir extensión.



Cuando queremos la podemos desinstalar.

```
<body>
|   <a href="#" aria-label="Leer más sobre la noticia del agua en marte">leer más</a>
|
| </body>
```

Se agregamos la etiqueta aria-label la voz leerá el mensaje que tiene entre comillas en lugar de decir leer más.

```
<body>
|   <a href="#" role="link">leer más</a>
|
| </body>
```

Da información sobre la zona que selecciona.

Open Graph Protocol

El protocolo Open Graph permite que cualquier página web se convierta en un objeto enriquecido en un gráfico social. Por ejemplo, esto se usa en Facebook para permitir que cualquier página web tenga la misma funcionalidad que cualquier otro objeto en Facebook.

Si bien existen muchas tecnologías y esquemas diferentes que podrían combinarse, no existe una sola tecnología que proporcione suficiente información para representar de manera rica cualquier página web dentro del gráfico social. El protocolo Open Graph se basa en estas tecnologías existentes y ofrece a los desarrolladores una cosa para implementar. La simplicidad del desarrollador es un objetivo clave del protocolo Open Graph que ha informado muchas de las decisiones de diseño técnico.

Para convertir sus páginas web en objetos gráficos, debe agregar metadatos básicos a su página. Hemos basado la versión inicial del protocolo en RDFa, lo que significa que colocará <meta>etiquetas adicionales en el <head>de su página web. Las cuatro propiedades requeridas para cada página son:

- og:title - El título de su objeto como debería aparecer dentro del gráfico, por ejemplo, "La Roca".
- og:type- El tipo de su objeto, por ejemplo, "video.movie". Dependiendo del tipo que especifique, es posible que también se requieran otras propiedades.
- og:image - Una URL de imagen que debe representar su objeto dentro del gráfico.
- og:url - La URL canónica de su objeto que se utilizará como su ID permanente en el gráfico, por ejemplo, "<https://www.imdb.com/title/tt0117500/>".

#YOAPRENDOCONDORIAN

```
<html prefix="og: https://ogp.me/ns#">
<head>
<title>The Rock (1996)</title>
<meta property="og:title" content="The Rock" />
<meta property="og:type" content="video.movie" />
<meta property="og:url" content="https://www.imdb.com/title/tt0117500/" />
<meta property="og:image" content="https://ia.media-imdb.com/images/rock.jpg" />
...
</head>
...
</html>
```

Introducción CSS 2021

Nivel básico

- Primeros pasos
- Estado de CSS
- Fundamentos de CSS
- Box Model
- Posicionamiento
- Bordes
- Outline
- Sombras
- Colores
- Textos
- Variables CSS

Nivel intermedio

- Metodología BEM
- Responsive design
- Imágenes
- Vídeos
- Background
- Tablas
- Listas
- Formularios
- Pseudoclases
- Pseudoelementos
- Flexbox
- Grid

Nivel avanzado

- Funcionamiento de CSS en el navegador
- Propiedades animables
- Estado inicial de los elementos
- CSSOM
- Transformaciones
- Transiciones 2D
- Eventos de transiciones (JS)
- Animaciones 2D
- Eventos de animaciones (JS)
- Transiciones 3D
- Animaciones 3D

Requisitos para el curso

- Conocer HTML
- Editor de código (VSCODE)
- Navegadores (Chrome, Firefox)
- Práctica, mucha práctica

¿Qué es CSS? Estado actual e historia

- Cascade Style Sheets (Hojas de estilo en cascada)

Estos documentos se leen desde arriba hacia abajo, esto quiere decir que los archivos que se encuentran en la parte inferior pisan a los de la parte superior.

- Es un lenguaje de diseño.
- Lo interpreta el navegador.

Tiene motor de render que es el que se encarga de leer el CSS y de pintar en pantallas las instrucciones que nosotros le damos.

- Nació en 1996.
- La versión CSS 2 salió en 1998.
- En 2011 se tomó la decisión de modularizar CSS.
- El contenido de CSS es acumulativo.

Instalaciones necesarias

En el apartado de HTML con respecto a las instalaciones necesarias, vamos a trabajar con lo siguiente:

Utilizaremos 2 navegadores: Chrome y Firefox.

Como editor Visual Studio Code con varias extensiones.

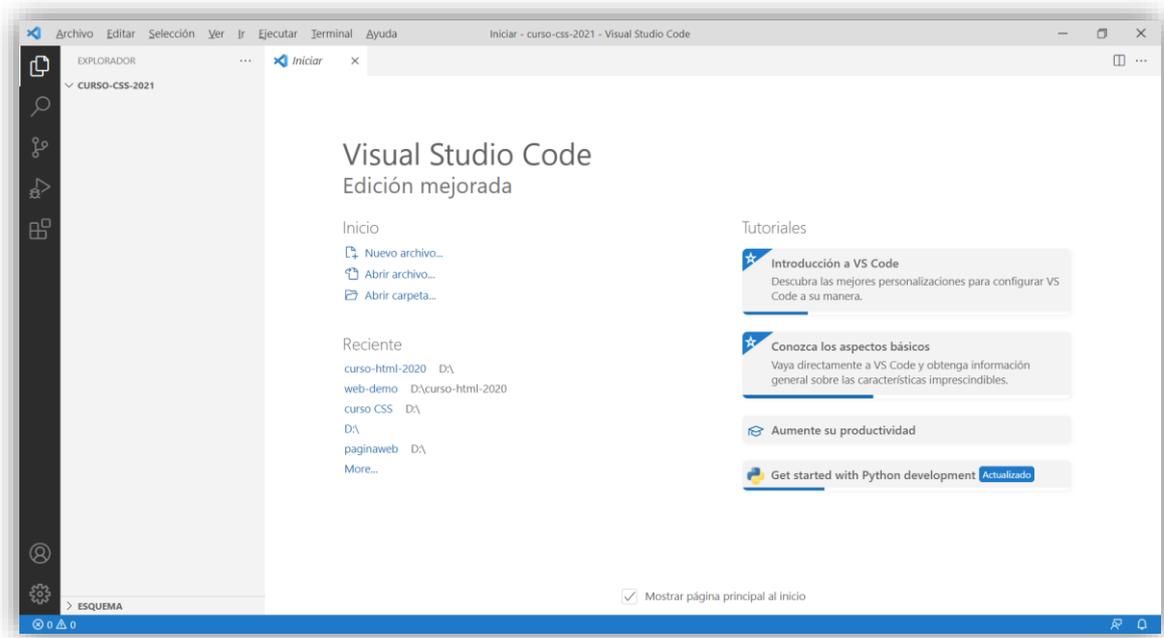
- Spanis Language Pack for Visual Studio Code.
- Live Server.
- Prettier - Code formatter.
- Mike McBride

En la introducción de HTML se explica con más detalles este apartado. (Página 3)

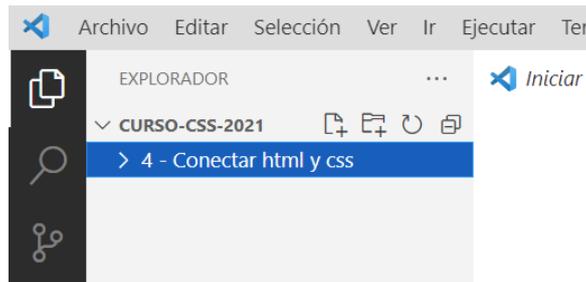
Unir HTML y CSS

Vamos a crear un carpeta llamada curso-css-2021.

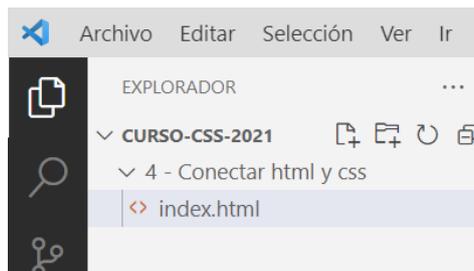
Ejecutamos Visual Studio Code y arrastramos esta carpeta.



Ahora vamos a crear al carpeta 4 – Conectar html y css.



En este carpeta vamos a crear nuestro archivo index.html.



Y la vamos a rellenar con la plantilla que lleva por defecto. ! Enter.



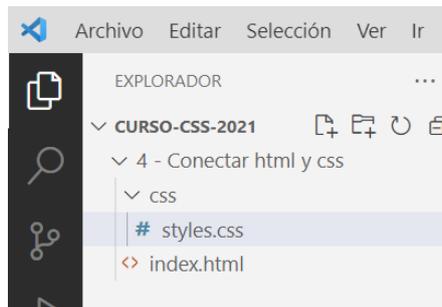
```
<body>
  <h1>Conectar HTML y CSS</h1>
</body>
```

Agregamos la siguiente línea.
Y un párrafo con relleno.

```
9 <body>
10   <h1>Conectar HTML y CSS</h1>
11   <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Aspernatur similique tempore possimus molestiae debitis hic!
    Placeat odit aliquam fuga ipsa corporis voluptatum molestiae, culpa ea quia, optio quae consequuntur perferendis!</p>
12 </body>
```

Hay cuatro formas de incluir css en html.

1.- Hacer una carpeta que se llame css y dentro creamos un archivo que se llame styles.css.

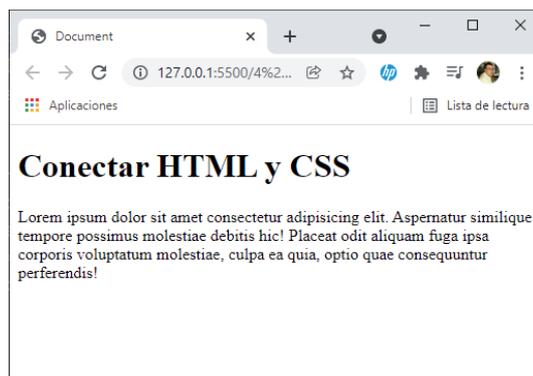


Desde html tenemos que conectar esta hoja.

```
4 - Conectar html y css > <> index.html > html > head > link
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <link rel="stylesheet" href="css/styles.css">
8   <title>Document</title>
9 </head>
```



Este será el resultado:



#YOAPRENDOCONDORIAN

Vamos a añadir el siguiente código en css.

4 - Conectar html y css > css > # styles.css > ...

```
1 body{
2     background-color: steelblue;
3 }
```

Este será el resultado:



2.- Otra forma es realizar el estilo en la mismo documento html.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7
8     <style>
9         body{
10             background-color: steelblue;
11         }
12     </style>
13     <title>Document</title>
14 </head>
15 <body>
16     <h1>Conectar HTML y CSS</h1>
17     <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Aspernatur similique tempore possimus molestiae debitis hic! Placeat odit aliquam fuga ipsa corporis voluptatum molestiae, culpa ea quia, optio quae consequuntur perferendis!</p>
18 </body>
19 </html>
```

El problema es que este estilo solo afecta a una solo documento html, si se hacen por separado este estilo puede aplicarse a varios documentos html.

Si queremos tener el estilo en varios documentos html habrá que escribirlos en cada uno de ellos, en cambio si el documento es externo con solo llamarlo será suficiente.

3.- forma no muy recomendable.

```
<h1 style="background-color: springgreen;">Conectar HTML y CSS</h1>
```

Aplicar el estilo a una determinada etiqueta.



Meter estilos en una etiqueta es lo que se hacía a los inicios.

En otros casos se puede hacer desde JavaScript.

4.- Otra forma no recomendable es importándola.

```
7 | <!--<link rel="stylesheet" href="css/styles.css"-->
8 | @import url(css/styles.css); | ←
9 | <title>Document</title>
10| </head>
```

Sintaxis de CSS

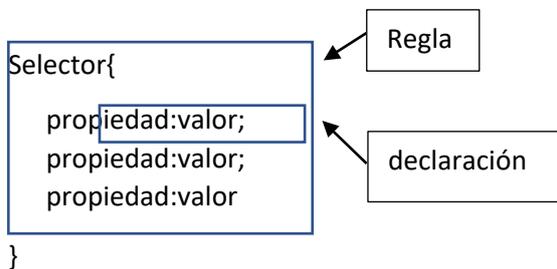
Selector: El elemento al que vamos a aplicar estilos.

Propiedad: Lo que vamos a cambiar.

Valor: El nuevo valor que vamos a cambiar a la propiedad.

Propiedad:Valor -> Se le denomina declaración.

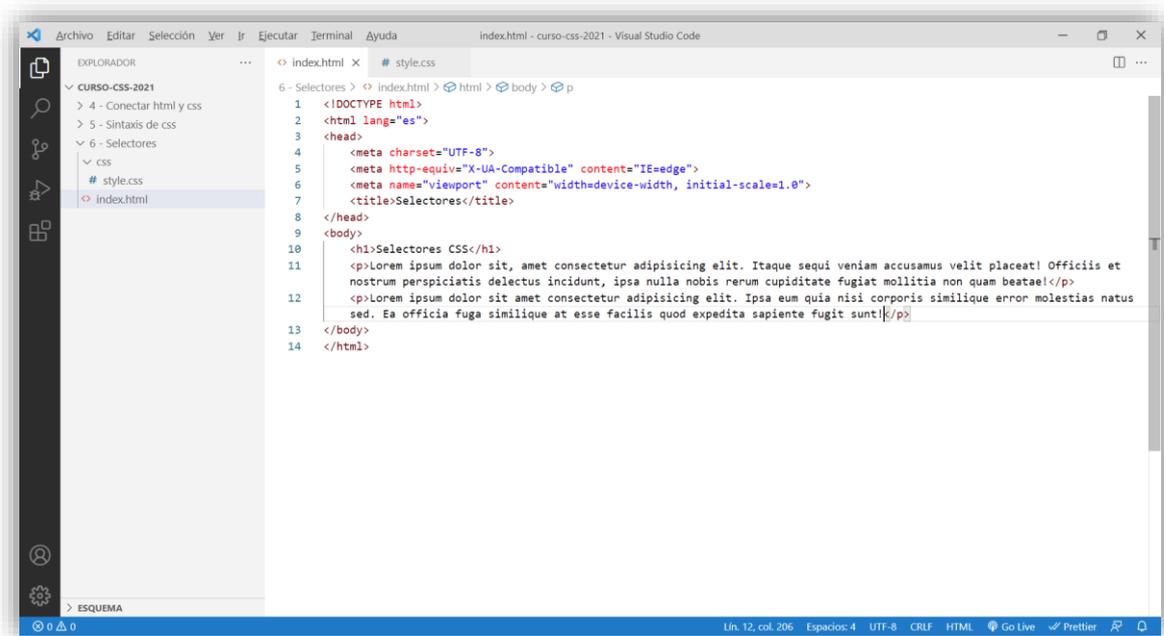
Al conjunto de selector + declaración(es) se le denomina regla.



Se pueden poner todas las propiedades que necesites para conseguir el estilo deseado.

Cada propiedad la separaremos con punto y coma.

Tipos de selectores en CSS



Para este capítulo vamos a crear una carpeta llamada “6 – Selectores”

Dentro de esta carpeta otra llamada “CSS” y dentro de este un archivo llamado style.css.

En la carpeta 6 – Selectores vamos a crear un archivo index.html con el contenido que se muestra.

Hay dos tipos de selectores Simples y Compuestos.

Simples:

- Selectores elementales

Selectores de atributo Compuestos:

- Selectores agrupados
- Selectores combinadores
- Pseudoclases – Pseudoelementos

Selectores elementales

Tenemos dos tipos:

- Selector universal -> *
- Selector de tipo o etiqueta -> Nombre de la etiqueta

Vamos a relacionar los dos documentos desde el HTML.

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/style.css">
  <title>Selectores</title>
</head>
```

Selector universal:

#YOAPRENDOCONDORIAN

```
*{
  background-color: lightcoral;
}
```

Este será el resultado:

**Selector de etiqueta:**

```
h1{
  background-color: lightcyan;
}
```

Este será el resultado:



```
p{
  background-color: lightgreen;
}
```

Este será el resultado:



Selectores de id y de clase

id -> id del elemento

clase -> clase del elemento

otros atributos.

En nuestro documento html en la etiqwta h1 hemos agregado un id="title".

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
  initial-scale=1.0">
  <link rel="stylesheet" href="css/style.css">
  <title>Selectores</title>
</head>
<body>
  <h1 id="title">Selectores CSS</h1>
  <p>Lorem ipsum dolor sit, amet consectetur adipisicing
  elit. Itaque sequi veniam accusamus velit placeat!
  Officiis et nostrum perspiciatis delectus incidunt, ipsa
  nulla nobis rerum cupiditate fugiat mollitia non quam
  beatae!</p>
  <p>Lorem ipsum dolor sit amet consectetur adipisicing
  elit. Ipsa eum quia nisi corporis similique error
  molestias natus sed. Ea officia fuga similique at esse
  facilis quod expedita sapiente fugit sunt!</p>
</body>
</html>
```

En el documento style.css hemos escrito lo siguiente:

```
#title{
  background-color: lightgreen;
}
```

Para hacer referencia del id que hemos escrito en el documento html desde css escribiremos una almohadilla seguido del nombre del id.

Este será el resultado:



El id tiene que escribirlo igual en scc ya que para él las mayúsculas y minúsculas son distintas.

En un documento html los id no se pueden repetir con el mismo nombre, por este motivo las clases que si se pueden repetir son mejor para agregar estilos.

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width,
7          initial-scale=1.0">
8      <link rel="stylesheet" href="css/style.css">
9      <title>Selectores</title>
10 </head>
11 <body>
12     <h1 class="title">Selectores CSS</h1>
13     <p>Lorem ipsum dolor sit, amet consectetur adipisicing
14     elit. Itaque sequi veniam accusamus velit placeat!
15     Officiis et nostrum perspiciatis delectus incidunt, ipsa
16     nulla nobis rerum cupiditate fugiat mollitia non quam
17     beatae!</p>
18     <p>Lorem ipsum dolor sit amet consectetur adipisicing
19     elit. Ipsa eum quia nisi corporis similique error
20     molestias natus sed. Ea officia fuga similique at esse
21     facilis quod expedita sapiente fugit sunt!</p>
22 </body>
23 </html>
```

Desde scc lo llamaremos con punto.

```
.title{
  background-color: lightgreen;
}
```

El resultado será el mismo.

En una clase se pueden poner varios títulos.

```
<h1 class="title title2 title3">Selectores CSS</h1>
```

```

1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width,
7          initial-scale=1.0">
8      <link rel="stylesheet" href="css/style.css">
9      <title>Selectores</title>
10 </head>
11 <body>
12     <h1 class="title">Selectores CSS</h1>
13     <p class="texto">Lorem ipsum dolor sit, amet consectetur
14     adipiscing elit. Itaque sequi veniam accusamus velit
15     placeat! Officiis et nostrum perspiciatis delectus
16     incidunt, ipsa nulla nobis rerum cupiditate fugiat
17     mollitia non quam beatae!</p>
18     <p class="texto">Lorem ipsum dolor sit amet consectetur
19     adipiscing elit. Ipsa eum quia nisi corporis similique
20     error molestias natus sed. Ea officia fuga similique at
21     esse facilis quod expedita sapiente fugit sunt!</p>
22 </body>
23 </html>

```

Repetimos la class="texto" dos veces cuando apliquemos un estilo a esta clase se aplicarán a los dos párrafos.

```

.texto{
  background-color: limegreen;
}

```

Este será el resultado:



Con un solo selector podemos aplicar estilos a un montón de elementos.

Selectores de atributos

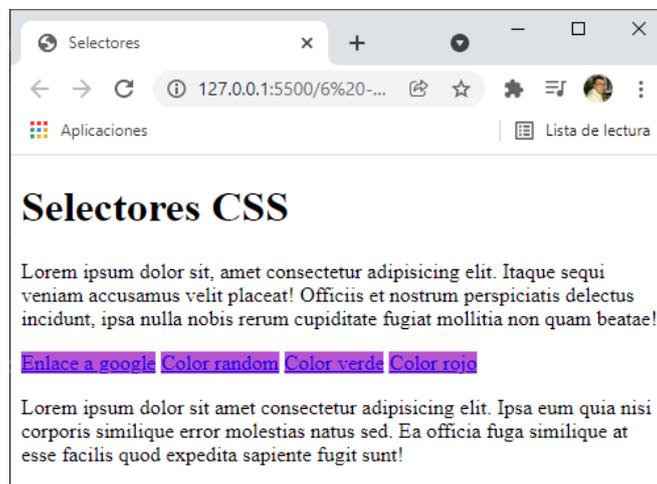
Otros tipos de atributos que va con [] utilizaremos [href]

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/style.css">
  <title>Selectores</title>
</head>
<body>
  <h1 class="title">Selectores CSS</h1>
  <p class="texto">Lorem ipsum dolor sit, amet consectetur adipisicing
elit. Itaque sequi veniam accusamus velit placeat! Officiis et nostrum
perspiciatis delectus incidunt, ipsa nulla nobis rerum cupiditate
fugiat mollitia non quam beatae!</p>
  <a href="http://google.com">Enlace a google</a>
  <a href="#color-modo-random">Color random</a>
  <a href="#seccion-verde">Color verde</a>
  <a href="#color-rojo">Color rojo</a>
  <p lang="en-GB">Lorem ipsum dolor sit amet consectetur adipisicing
elit. Ipsa eum quia nisi corporis similique error molestias natus sed.
Ea officia fuga similique at esse facilis quod expedita sapiente fugit
sunt!</p>
</body>
</html>
```

En CSS

```
[href]{
  background-color: #mediumorchid;
}
```

Este será el resultado:



También podemos poner [atributo=valor]

```
[href="http://google.com"]{
  background-color: mediumvioletred
}
```

Este será el resultado:



Por cascada puede modificarse, el estilo que está por debajo anula al estilo que está por encima.

Otro modo [atributo^=valor] Comprobamos si el atributo empieza por este valor.

```
13 <a href="http://google.com">Enlace a google</a>
14 <a href="color-modo-random">Color random</a>
15 <a href="#seccion-verde">Color verde</a>
16 <a href="color-rojo">Color rojo</a>
```

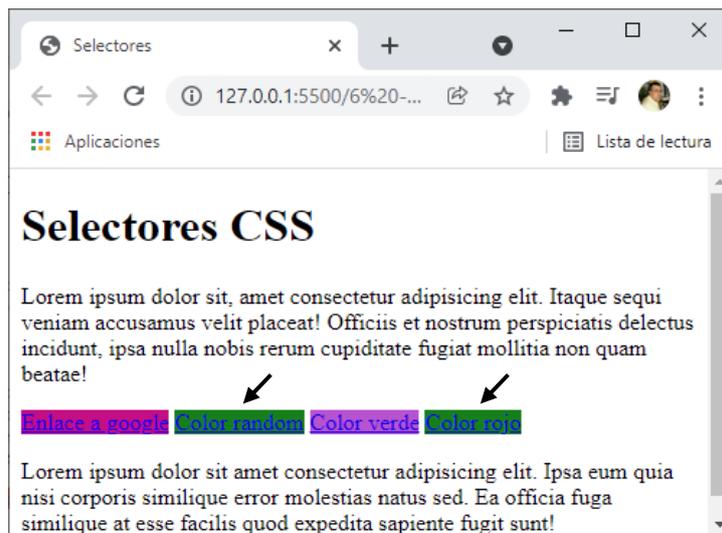
En el documento html le quitamos las almohadillas a las líneas 14 y 16.

En estilos agregamos el siguiente código:

```
[href^=color]{
  background-color: green;
}
```

Como color es solo una palabra puedes evitar poner comillas, pero es mejor acostumbrarse a poner las siempre.

Este será el resultado:



Otro modo [atributo*=valor] que tenga el valor en cualquier lugar, no necesariamente al principio.

```

13 | <a href="http://google.com">Enlace a google</a>
14 | <a href="#color-modo-random">Color random</a>
15 | <a href="#seccion-verde">Color verde</a>
16 | <a href="#color-rojo">Color rojo</a>

```

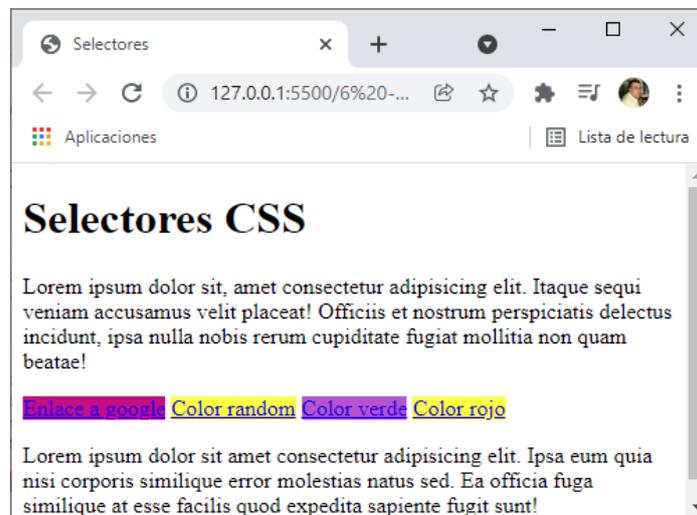
Ponemos de nuevo la almohadilla en las líneas 14 y 16

```

[href*="color"]{
  background-color: yellow;
}

```

Este será el resultado:



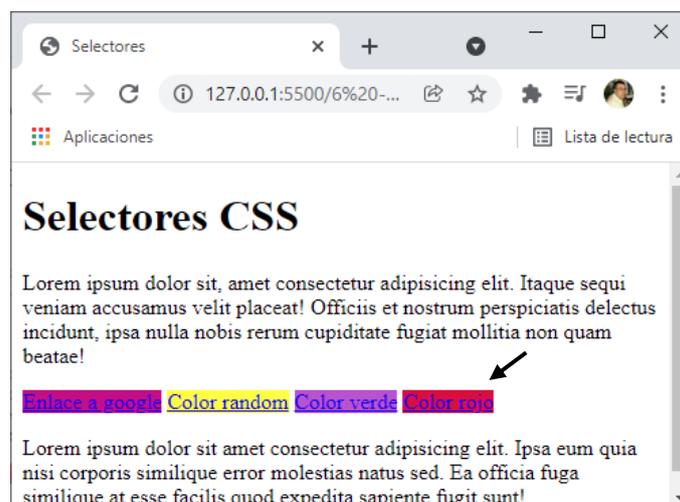
Otro modo [atributo \$=valor] si termina con un determinado valor.

```

[href$="rojo"]{
  background-color: crimson ;
}

```

Este será el resultado:



```
17 <p lang="en">Lorem ipsum dolor sit amet consectetur adipisicing elit.
    Ipsa eum quia nisi corporis similique error molestias natus sed. Ea
    officia fuga similique at esse facilis quod expedita sapiente fugit
    sunt!</p>
```

Para este ejemplo queremos hacer referencia a un párrafo que está escrito en inglés.

Otro modo [atributo|=valor] acepta que la palabra a buscar tenga guión.

```
[lang|= "en"]{
  background-color: magenta;
}
```

Este será el resultado:



```
17 <p lang="en-GB">Lorem ipsum dolor sit amet consectetur adipisicing
    elit. Ipsa eum quia nisi corporis similique error molestias natus sed.
    Ea officia fuga similique at esse facilis quod expedita sapiente fugit
    sunt!</p>
```

Si agregamos guión y algo más también funciona.

Selectores agrupados

Selectores compuestos:

- Selectores Agrupados
- Selectores Combinadores
- Pseudoclase – Pseudoelementos (Más adelante)

```
<!DOCTYPE html>
```

```
<html lang="es">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<link rel="stylesheet" href="css/style.css">

<title>Selectores</title>

</head>
<body>

  <h1 class="title">Selectores CSS</h1>
  <p class="subtitle">Subtitulo</p>
  <p class="text">

    Lorem ipsum dolor sit, amet consectetur adipisicing elit. Itaque sequi veniam accusamus
    velit placeat! Officiis et nostrum perspiciatis delectus incidunt, ipsa nulla nobis rerum
    cupiditate fugiat mollitia non quam beatae!

  </p>
  <div>

    <p class="text-2">

      Lorem ipsum dolor sit amet consectetur adipisicing elit. Molestias consequatur neque
      eaque aliquid porro, debitis ratione doloremque! Vel a, numquam dolores nesciunt ratione
      error consectetur assumenda, ullam molestiae quis molestias!

    </p>
    <h2>Título secundario</h2>
    <p class="text-3">

      Lorem ipsum, dolor sit amet consectetur adipisicing elit. Commodi veniam reiciendis ea
      earum quis nemo. Laborum sint soluta quisquam repellendus eius necessitatibus itaque
      pariatur? Delectus enim similique tenetur quasi voluptas.

    </p>
    <p class="text-4">

      Lorem ipsum, dolor sit amet consectetur adipisicing elit. Illo excepturi modi praesentium
      iure tenetur sunt similique, repellendus laborum quaerat quis eum. Tenetur, animi sequi.
      Eveniet fuga mollitia molestias error unde.

    </p>
  </div>
</body>
</html>
```



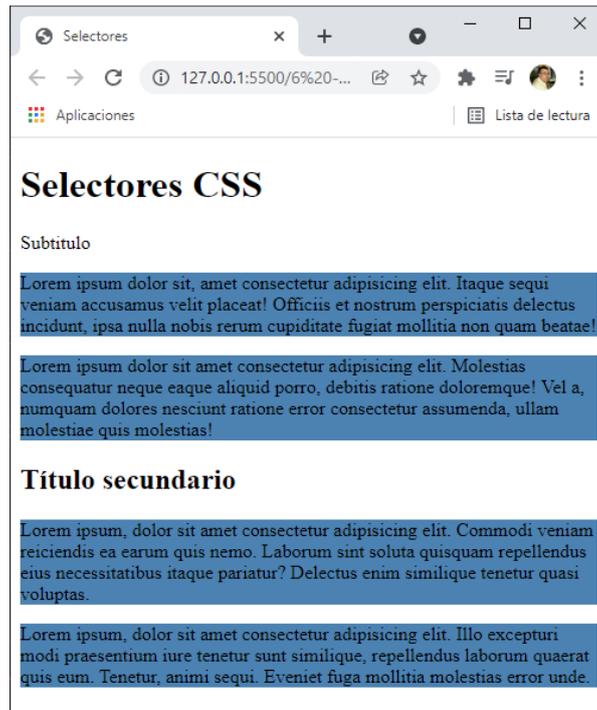
```
.text{
  background-color: steelblue;
}

.text-2{
  background-color: steelblue;
}

.text-3{
  background-color: steelblue;
}

.text-4{
  background-color: steelblue;
}
```

Con este código CSS este será el resultado:



Estamos repitiendo mucho código y es algo que tenemos que evitar.

```
.text,
.text-2,
.text-3,
.text-4{
  background-color: steelblue;
}
```

Con esta opción de selectores agrupados se obtiene el mismo resultado.

Selectores Descendente

Tenemos dos títulos secundarios con h2 si agrego el siguiente código css.

```
h2{
  background-color: teal;
}
```

Este será el resultado:



Pero en nuestro ejemplo solo queremos que afecte solo al segundo título secundario.

```
<div>
  <p class="text-2">
    Lorem ipsum dolor sit amet consectetur adipisicing elit. Molestias
    consequatur neque eaque aliquid porro, debitis ratione doloremque! Vel
    a, numquam dolores nesciunt ratione error consectetur assumenda, ullam
    molestiae quis molestias!
  </p>
  <h2>Titulo secundario</h2>
  <p class="text-3">
    Lorem ipsum, dolor sit amet consectetur adipisicing elit. Commodi veniam
    reiciendis ea earum quis nemo. Laborum sint soluta quisquam repellendus
    eius necessitatibus itaque pariatur? Delectus enim similique tenetur
    quasi voluptas.
  </p>
  <p class="text-4">
    Lorem ipsum, dolor sit amet consectetur adipisicing elit. Illo excepturi
    modi praesentium iure tenetur sunt similique, repellendus laborum
    quaerat quis eum. Tenetur, animi sequi. Eveniet fuga mollitia molestias
    error unde.
  </p>
</div>
```

Como el titulo secundario número dos se encuentra en una etiqueta <div> y nos queremos hacer referencia a este segundo título.

```
div h2{
  background-color: teal
}
```

Especificamos las dos etiquetas una como primer nivel y la segunda como segundo nivel, este será el resultado:



A esto se le denomina nombrar hijos.
 Entramos en <div> Y <h2> sería el hijo de <div>.

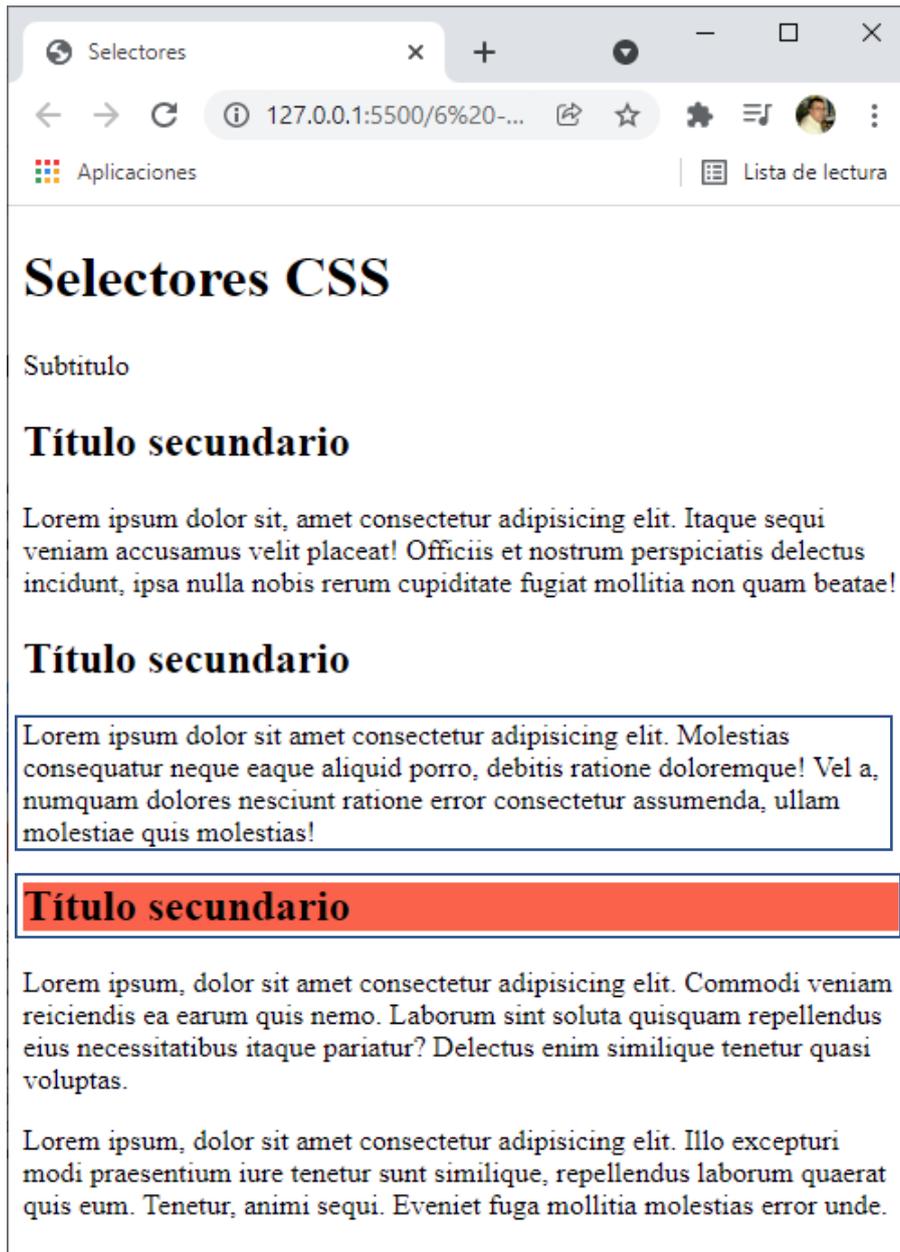
Selector de hermano

Selector de hermano siguiente.

```
.text-2 + .title-2{
  background-color: tomato;
}
```

```
<p class="text-2">
  Lorem ipsum dolor sit amet consectetur adipisicing elit. Molestias
  consequatur neque eaque aliquid porro, debitis ratione doloremque! Vel
  a, numquam dolores nesciunt ratione error consectetur assumenda, ullam
  molestiae quis molestias!
</p>
<h2 class="title-2">Titulo secundario</h2>
<p class="text-3">
  Lorem ipsum, dolor sit amet consectetur adipisicing elit. Commodi veniam
  reiciendis ea earum quis nemo. Laborum sint soluta quisquam repellendus
  eius necessitatibus itaque pariatur? Delectus enim similique tenetur
  quasi voluptas.
</p>
```

Este será el resultado:



Se puede utilizar etiquetas como clases.

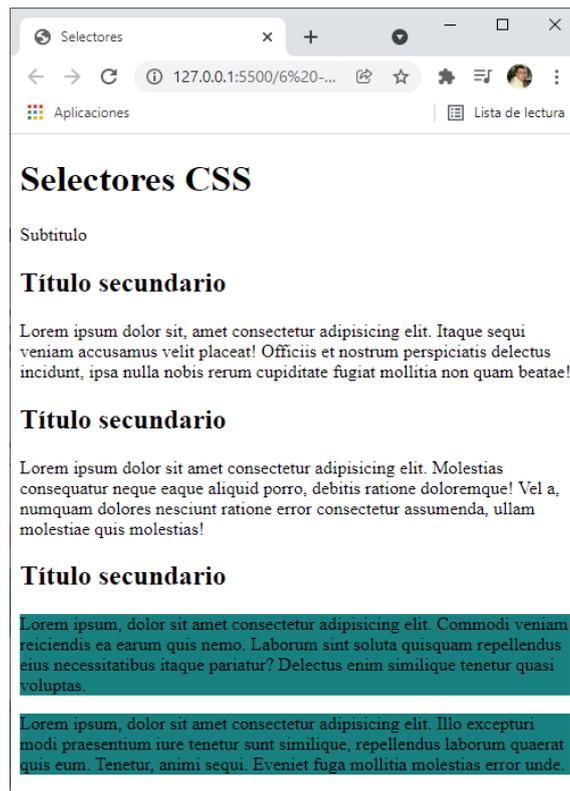
Todos los hermanos siguientes:

```
.text-2 ~ p{  
  background-color: teal ;  
}
```



```
<p class="text-2">
  Lorem ipsum dolor sit amet consectetur adipisicing elit. Molestias
  consequatur neque eaque aliquid porro, debitis ratione doloremque! Vel
  a, numquam dolores nesciunt ratione error consectetur assumenda, ullam
  molestiae quis molestias!
</p>
<h2 class="title-2">Título secundario</h2>
<p class="text-3">
  Lorem ipsum, dolor sit amet consectetur adipisicing elit. Commodi veniam
  reiciendis ea earum quis nemo. Laborum sint soluta quisquam repellendus
  eius necessitatibus itaque pariat? Delectus enim similique tenetur
  quasi voluptas.
</p>
<p class="text-4">
  Lorem ipsum, dolor sit amet consectetur adipisicing elit. Illo excepturi
  modi praesentium iure tenetur sunt similique, repellendus laborum
  quaerat quis eum. Tenetur, animi sequi. Eveniet fuga mollitia molestias
  error unde.
```

Este será el resultado:



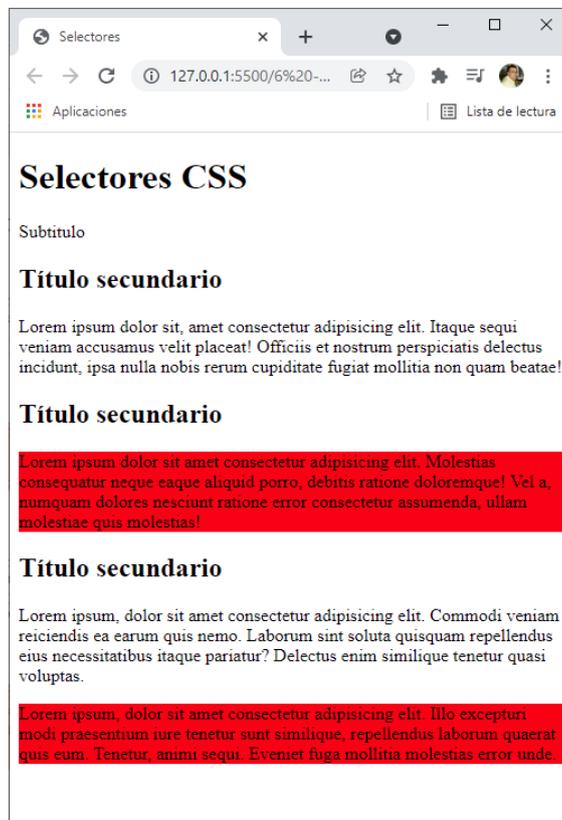
Selector de hijo directo

```

<div class="container">
  <p class="text-2">
    Lorem ipsum dolor sit amet consectetur adipisicing elit. Molestias
    consequatur neque eaque aliquid porro, debitis ratione doloremque! Vel
    a, numquam dolores nesciunt ratione error consectetur assumenda, ullam
    molestiae quis molestias!
  </p>
  <h2 class="title-2">Título secundario</h2>
  <div>
    <p class="text-3">
      Lorem ipsum, dolor sit amet consectetur adipisicing elit. Commodi
      veniam reiciendis ea earum quis nemo. Laborum sint soluta quisquam
      repellendus eius necessitatibus itaque pariatu? Delectus enim
      similique tenetur quasi voluptas.
    </p>
  </div>
  <p class="text-4">
    Lorem ipsum, dolor sit amet consectetur adipisicing elit. Illo excepturi
    modi praesentium iure tenetur sunt similique, repellendus laborum
    quaerat quis eum. Tenetur, animi sequi. Eveniet fuga mollitia molestias
    error unde.
  </p>
</div>

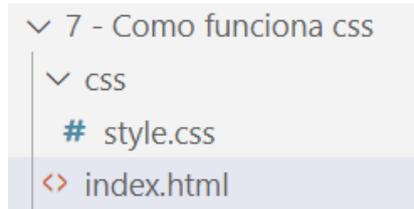
.container > p{
  background-color: red;
}

```



Especificidad y Cascada

Para este capítulos vamos a crear la siguiente estructura:



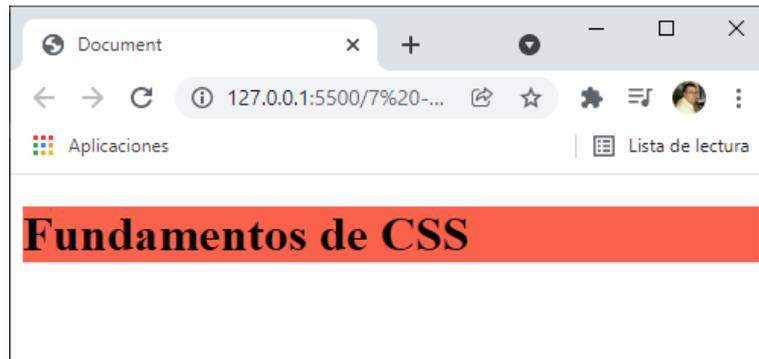
El código html.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/style.css">
  <title>Document</title>
</head>
<body>
  <h1 id="title" class="title">Fundamentos de CSS</h1>
</body>
</html>
```

Código css.

```
h1{
  background-color: red;
}
.title{
  background-color: blue;
}
#title{
  background-color: green;
}
h1.title{
  background-color: yellow;
}
h1#title{
  background-color: teal;
}
h1.title#title{
  background-color: tomato;
}
```

Este será el resultado:



La prioridad de los selectores.

Podemos pensar que como es el último que se aplica, pero lo vamos a mover.

```
h1.title#title{
  background-color: tomato;
}
h1{
  background-color: red;
}
.title{
  background-color: blue;
}
#title{
  background-color: green;
}
h1.title{
  background-color: yellow;
}
h1#title{
  background-color: teal;
}
```

Y el resultado sigue siendo el mismo.

Suma unos valores y con este resultado da una prioridad u otra.

¿Cómo lo valora?

Etiqueta y pseudoelementos 001

Clases, atributos y pseudoclasas 010

Ids 100

Estilos en línea 1000

!important GANA A TODO y no se usa nunca.

```
#title{
  background-color: green !important;
}
```

Para evitar confusiones siempre utilizar clases.

#YOAPRENDOCONDORIAN

```
.title{
  background-color: blue;
}
```



Hay una página web que nos puede ayuda para decirnos si estamos realizando bien el codigo.

<https://jonassebastianohlsson.com/specificity-graph/>

A screenshot of a YouTube video player. The video is titled "CURSO de CSS desde CERO 2021 - #14 - Especificidad y Cascada". The main content of the video is a web browser displaying the "CSS Specificity Graph Generator" tool. The tool shows a line graph with "Specificity" on the y-axis and "Location in stylesheet" on the x-axis. The graph has several sharp peaks. Below the graph, there is a code editor showing CSS rules. The video player interface includes a search bar, a chat window on the right with a list of video chapters, and video controls at the bottom.

Si la gráfica tiene muchos picos significa que algo no estamos haciendo bien.

Cascada -> Funciona siempre que la especificidad sobre el elemento sea la misma.

Herencias en CSS

Es la capacidad que tienen algunos elementos de heredar algunas propiedades de sus elementos contenedores (padres, abuelos, etc).

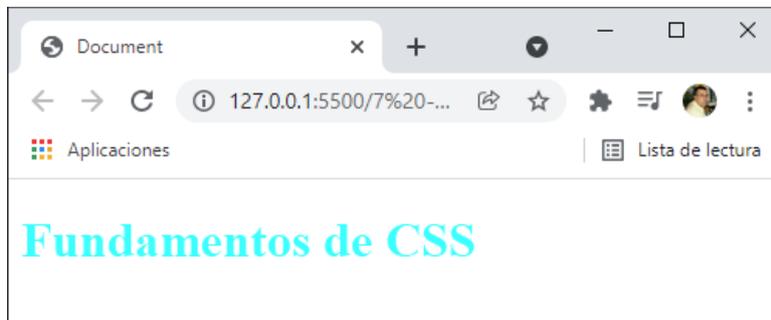
Código html.

```
<body>
|   <h1 id="title" class="title">Fundamentos de <span>CSS</span></h1>
| </body>
```

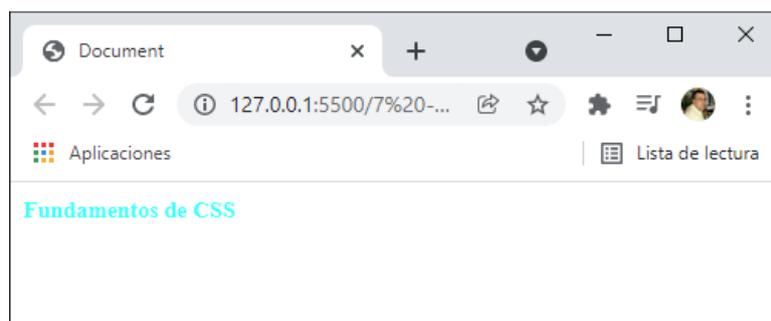
Código scs.

```
h1{
|   color: cyan;
| }
}
```

La etiqueta `` es una etiqueta hijo de `<h1>`, si en `h1` le asignamos un color de fuente la etiqueta `` también la heredará, este será el resultado:



```
h1{
|   color: cyan;
|   font-size: 15px;
| }
}
```



Vamos a comentar un ejemplo que no se hereda, dentro de un párrafo vamos a poner un enlace, vamos a cambiar de color el párrafo y observaremos que el enlace se mantiene con el mismo color, código html.

```
<p class="text">Lorem ipsum, dolor sit amet consectetur adipisicing elit.
| Quisquam cumque ipsa quis <a href="#">Enlace ejemplo</a> est inventore! Cum
| similique aliquid reiciendis sit, veritatis pariatur vitae laboriosam possimus
| nulla quam commodi omnis recusandae at.</p>
```

Código css.

```

}p{
|   color: ■ red;
}

```

Este será el resultado:

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Quisquam cumque ipsa quis [Enlace ejemplo](#) est inventore! Cum similique aliquid reiciendis sit, veritatis pariatur vitae laboriosam possimus nulla quam commodi omnis recusandae at.

Los enlaces no heredan de las propiedades de su padre.

En este caso para cambiar el color a un enlace:

```

p a{
|   color: ■ green;
}

```

Este será el resultado:

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Quisquam cumque ipsa quis [Enlace ejemplo](#) est inventore! Cum similique aliquid reiciendis sit, veritatis pariatur vitae laboriosam possimus nulla quam commodi omnis recusandae at.

Otro ejemplo:

Código html.

```

<p class="text">Lorem ipsum, dolor sit amet consectetur adipisicing elit. Quisquam cumque ipsa quis <a class="enlace" href="#">Enlace ejemplo</a> est inventore! Cum similique aliquid reiciendis sit, veritatis pariatur vitae laboriosam possimus nulla quam commodi omnis recusandae at.</p>

```

Código scc:

```

.enlace{
|   color: ■ pink;
}

```

Este será el resultado:

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Quisquam cumque ipsa quis [Enlace ejemplo](#) est inventore! Cum similique aliquid reiciendis sit, veritatis pariatur vitae laboriosam possimus nulla quam commodi omnis recusandae at.

inherit -> Se utiliza cuando queremos forzar la herencia.

Código html.

```

<p class="text">Lorem ipsum, dolor sit amet consectetur adipisicing elit. Quisquam cumque ipsa quis <a class="enlace" href="#">Enlace ejemplo</a> est inventore! Cum similique aliquid reiciendis sit, veritatis pariatur vitae laboriosam possimus nulla quam commodi omnis recusandae at.</p>

```

#YOAPRENDOCONDORIAN

Codigo css.

```
p{
  color: red;
}

.enlace{
  color: inherit;
}
```

Este será el resultado:

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Quisquam cumque ipsa quis [Enlace ejemplo](#) est inventore! Cum similique aliquid reiciendis sit, veritatis pariatur vitae laboriosam possimus nulla quam commodi omnis recusandae at.

Ahora vamos a cambiar el párrafo a color verde.

```
p{
  color: green;
}

.enlace{
  color: inherit;
}
```

Este será el resultado:

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Quisquam cumque ipsa quis [Enlace ejemplo](#) est inventore! Cum similique aliquid reiciendis sit, veritatis pariatur vitae laboriosam possimus nulla quam commodi omnis recusandae at.

Ahora vamos a realizar un ejemplo en la que no queremos que herede.
initial -> para que no herede el color.

Código html.

```
<ul class="list">
  <li class="list-item">Item 1</li>
  <li class="list-item">Item 2</li>
  <li class="list-item">Item 3</li>
  <li class="list-item-extra">Item 4</li>
</ul>
```

Código css.

#YOAPRENDOCONDORIAN

```

body{
  background-color: yellow;
}

.list{
  color: indianred;
}

.list-item-extra{
  color: initial;
}

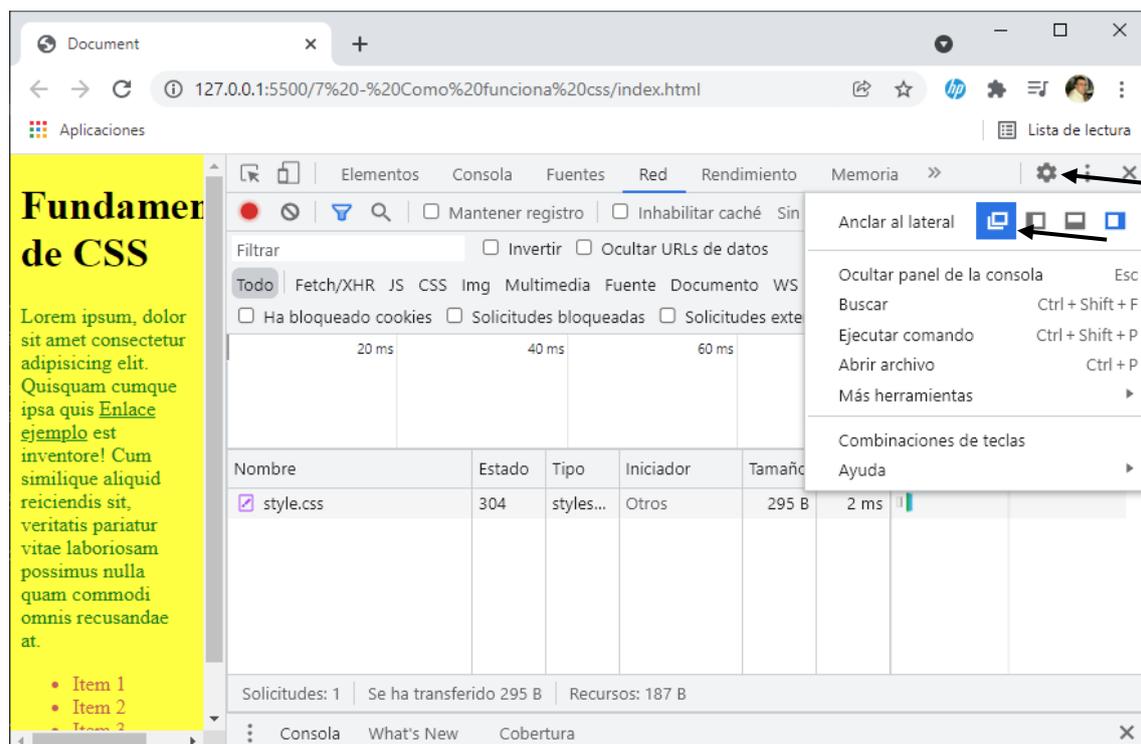
```

Este será el resultado:



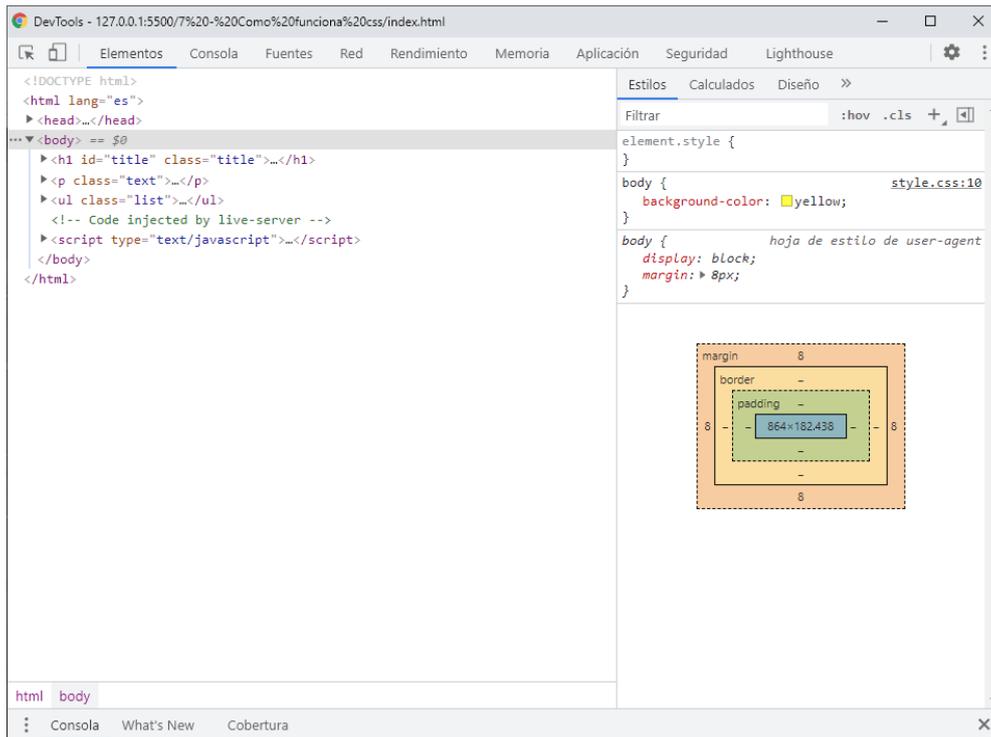
Estilos computados

Desde el navegador Chrome o Firefox pulsamos F12.



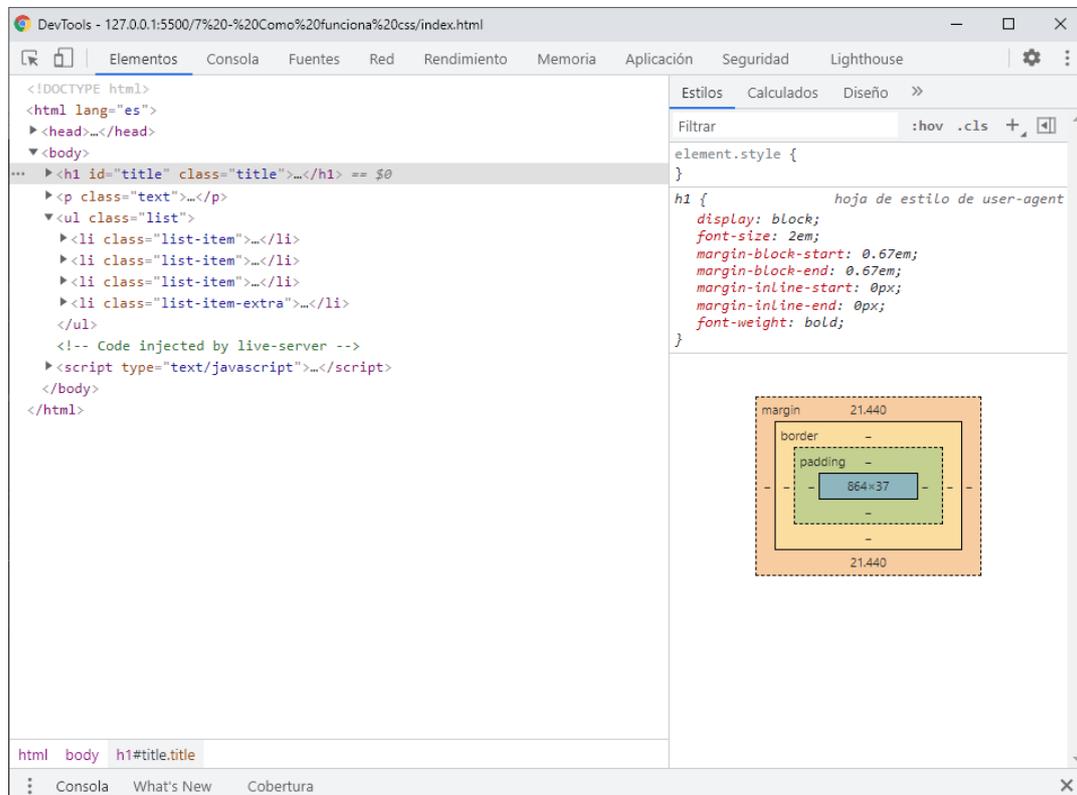
#YOAPRENDOCONDORIAN

Si seleccionamos Elementos.



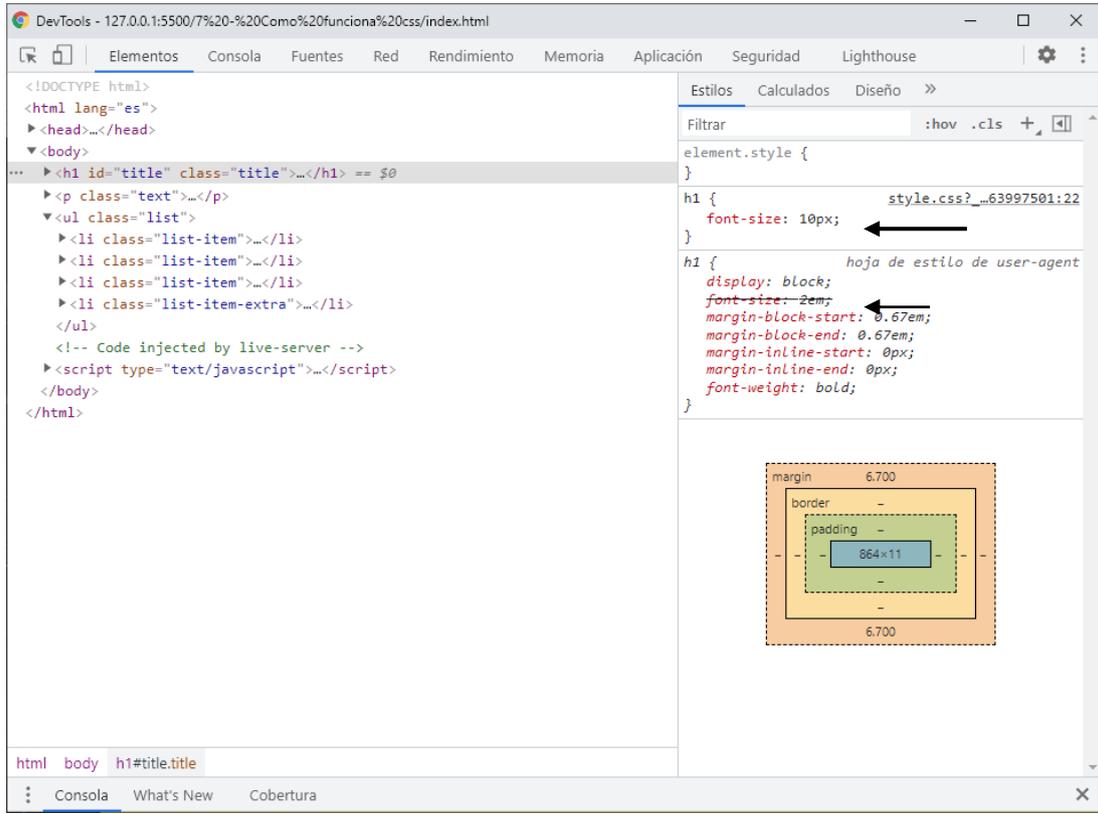
Podemos analizar el código y saber que estilos se han aplicado, además de los estilos que se le han aplicado por defecto.

Si seleccionamos la etiqueta <h1> podremos observar los valores.

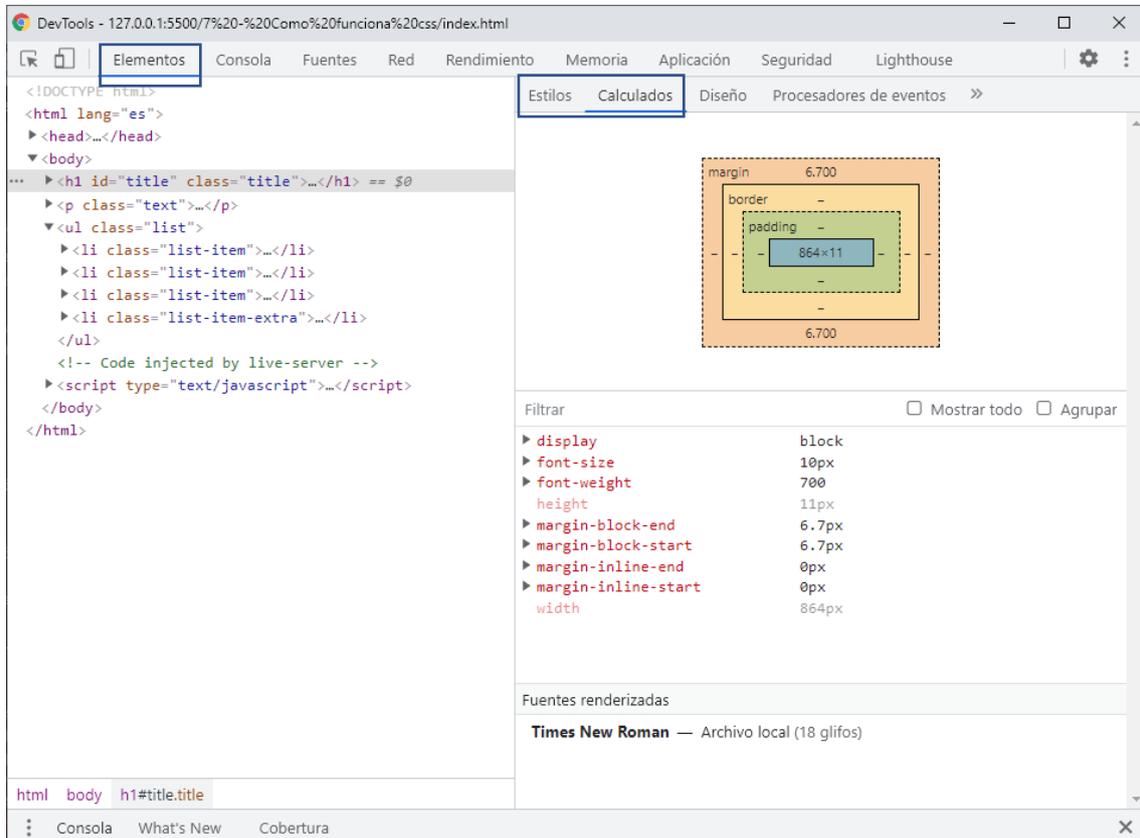


Vamos a aplicar un estilo de 10 pixeles a la etiqueta h1.

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



El valor por defecto aparece tachado porque ha sido reemplazado por el nuevo estilo.



#YOAPRENDOCONDORIAN

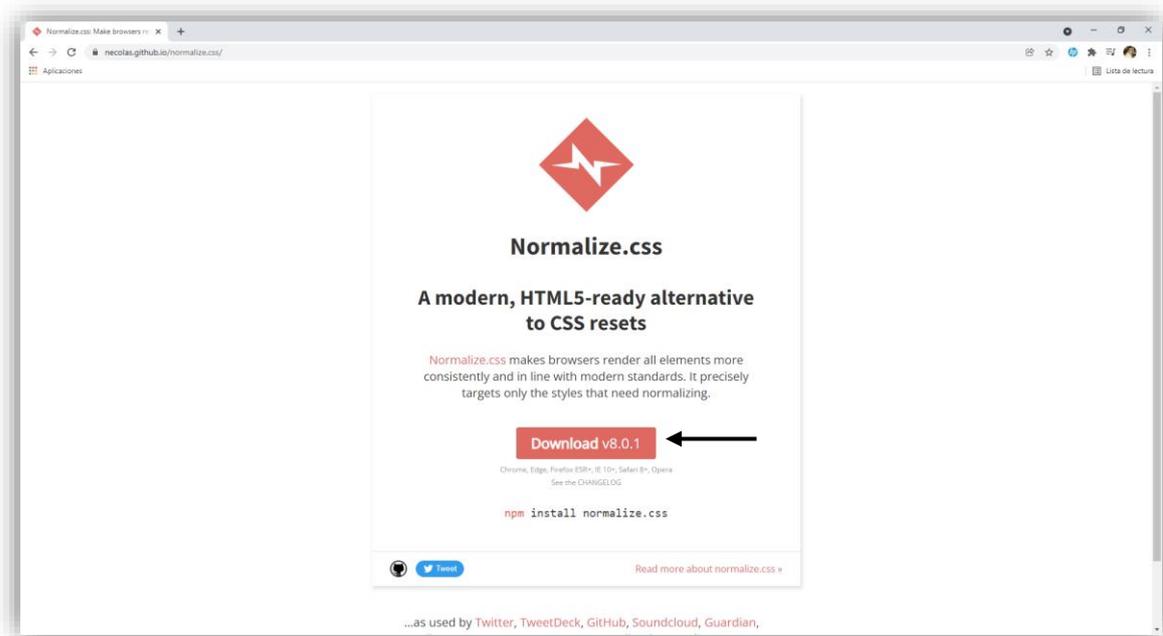
De la pestaña elementos en estilos y calculados obtendremos muchas respuestas.

Normalizar estilos del navegador

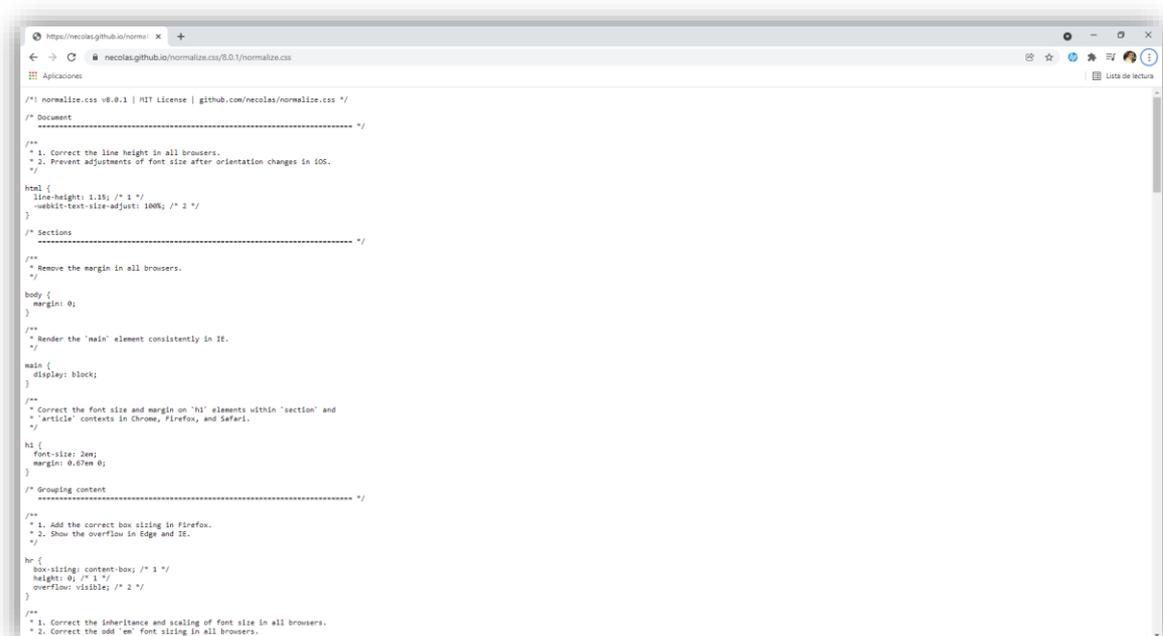
Nos encontramos que algunas veces los navegadores no están actualizados y por este motivo podemos observar algunas diferencias entre ellos.

Vamos a acceder al siguiente enlace:

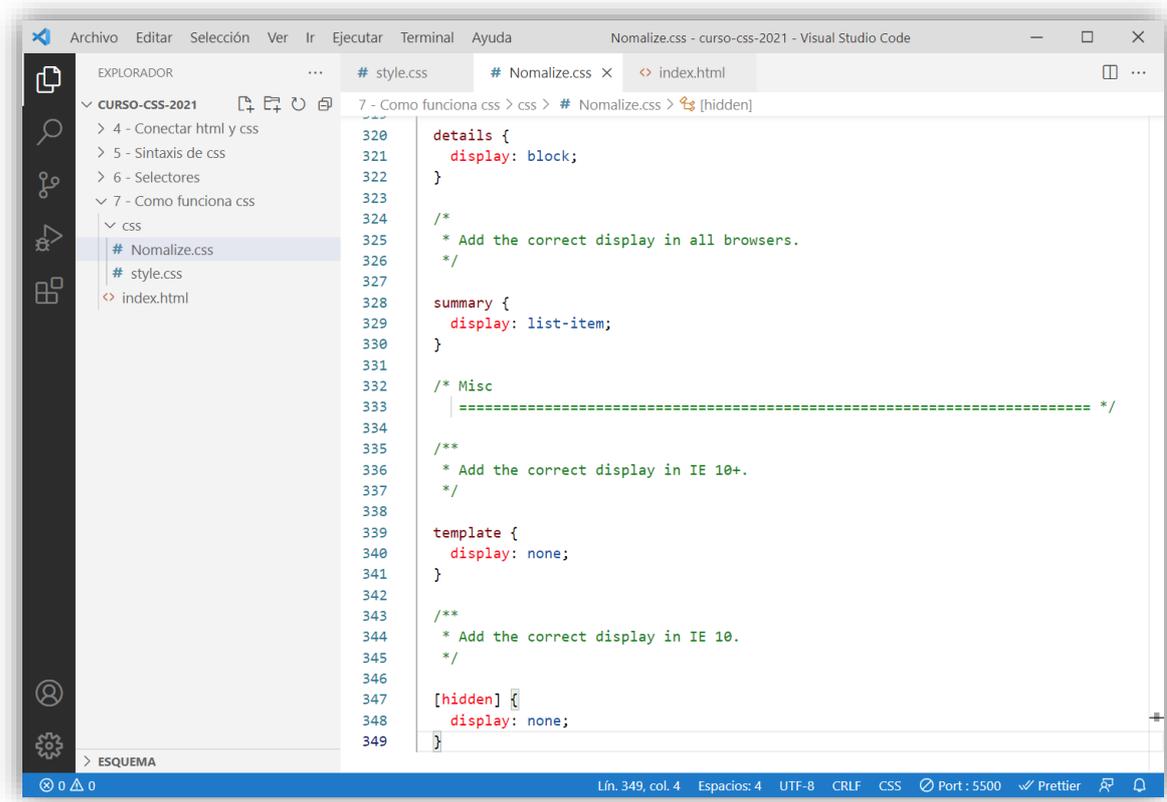
<https://necolas.github.io/normalize.css/>



Le damos a descargar.



Seleccionamos todo el código y lo copiamos.



Creamos un nuevo documento en la carpeta css con el nombre Normalize.css.

Ahora nos vamos al código html.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/Normalice.css"> ←
  <link rel="stylesheet" href="css/style.css">
  <title>Document</title>
</head>
<body>
  <h1 id="title" class="title">Fundamentos de <span>CSS</span></h1>
  <p class="text">Lorem ipsum, dolor sit amet consectetur adipisicing elit.
  Quisquam cumque ipsa quis <a class="enlace" href="#">Enlace ejemplo</a> est
  inventore! Cum similique aliquid reiciendis sit, veritatis pariatur vitae
  laboriosam possimus nulla quam commodi omnis recusandae at.</p>
```

#YOAPRENDOCONDORIAN

```

<ul class="list">
  <li class="list-item">Item 1</li>
  <li class="list-item">Item 2</li>
  <li class="list-item">Item 3</li>
  <li class="list-item-extra">Item 4</li>
</ul>
</body>
</html>

```

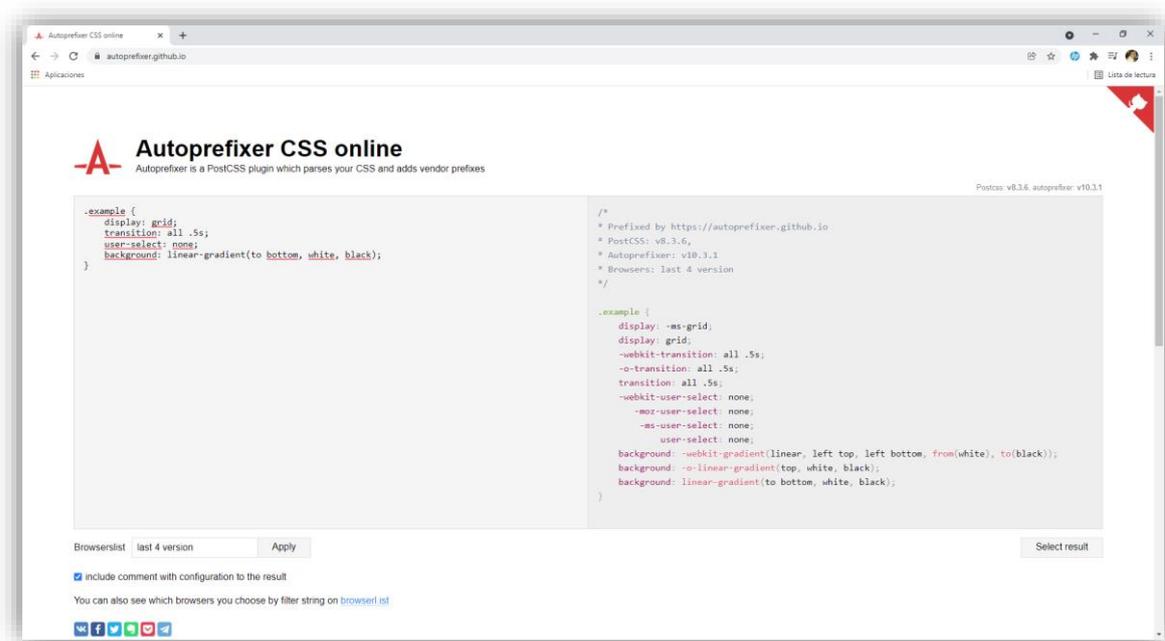
La vinculamos antes de vincular nuestra hoja de estilos.

Prefijos Proprietarios

Cuando una propiedad css se convierte en un estándar o se empieza a hacer pruebas con ella los navegadores necesitan implementarla. Si esta propiedad se ha reconvertido en recomendación y a entrado en los que pertenece a css los prefijos ya no hace falta pero hasta propiedad llega a convertirse en una propiedad como tal necesitamos colocar ciertos prefijos para que los navegadores sean capaces de interpretarla.

Vamos a la siguiente página web:

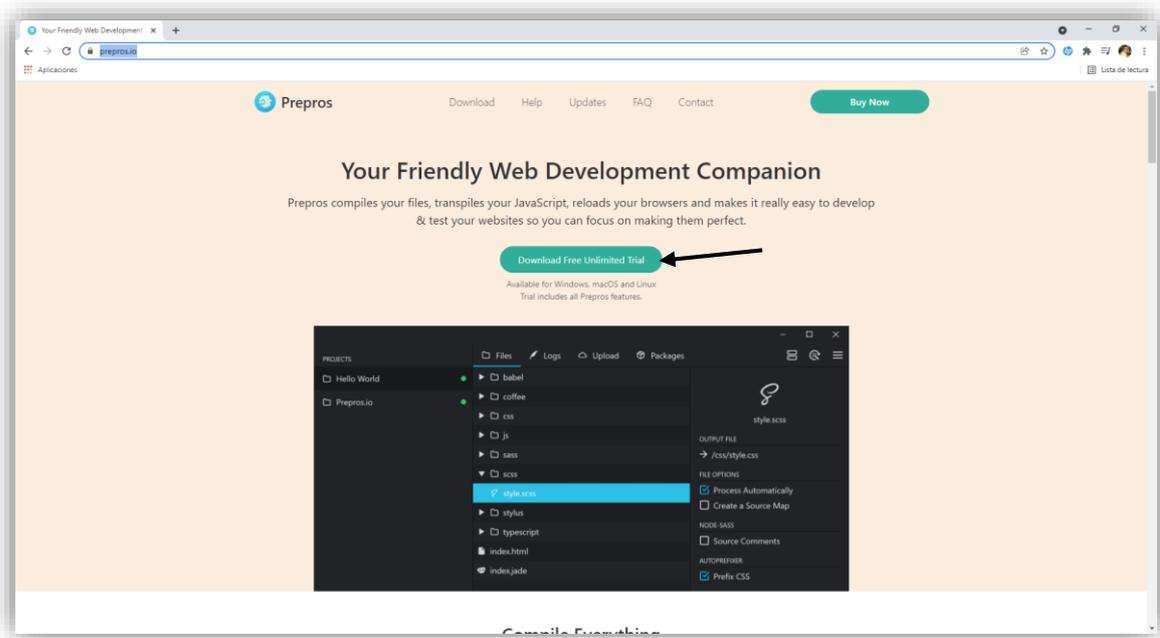
<https://autoprefixer.github.io/>



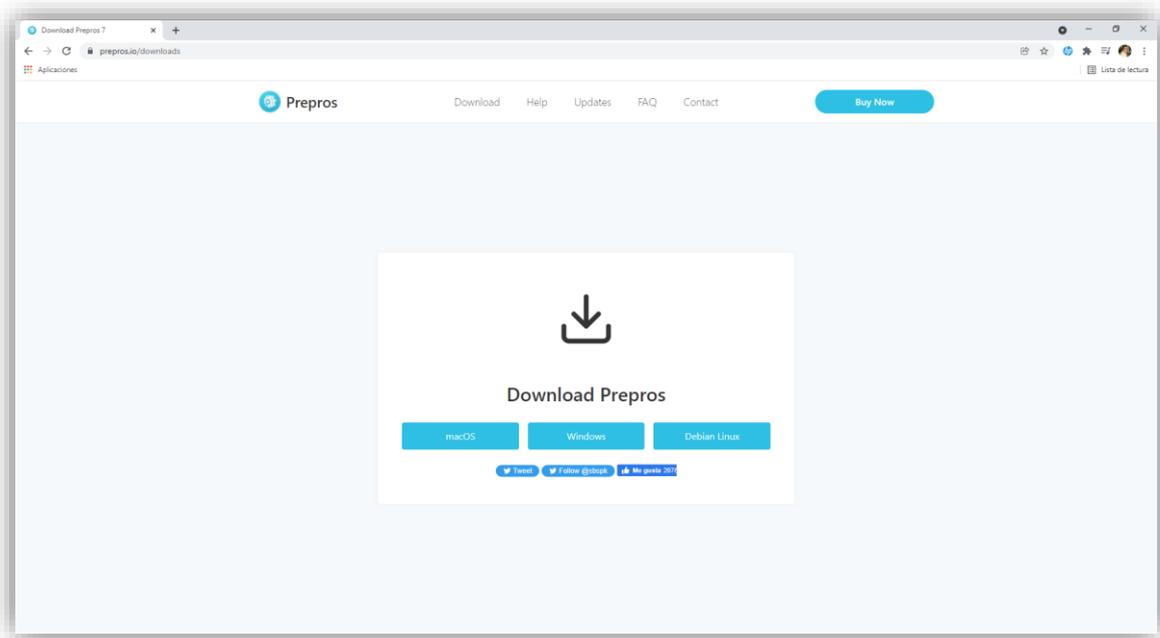
En la parte derecha vemos 4 propiedades que no está implementada en todos los navegadores.

Para esto necesitaremos un programa.

<https://prepros.io/>



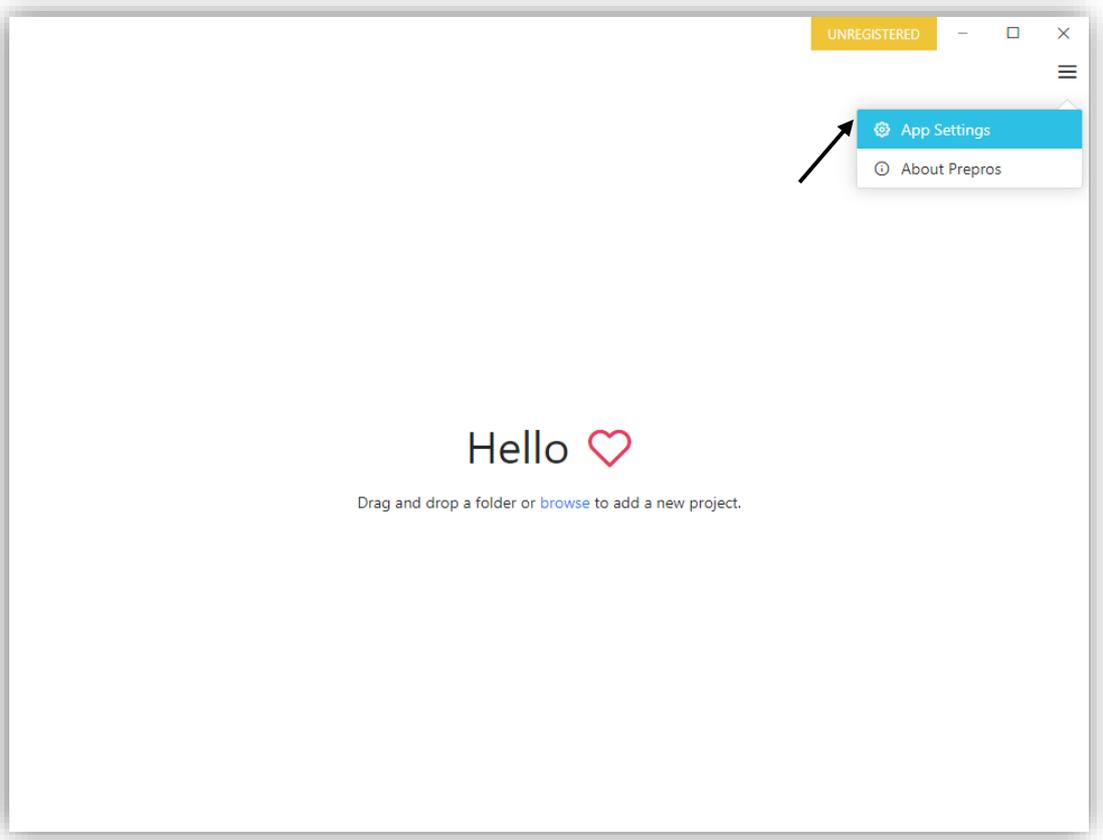
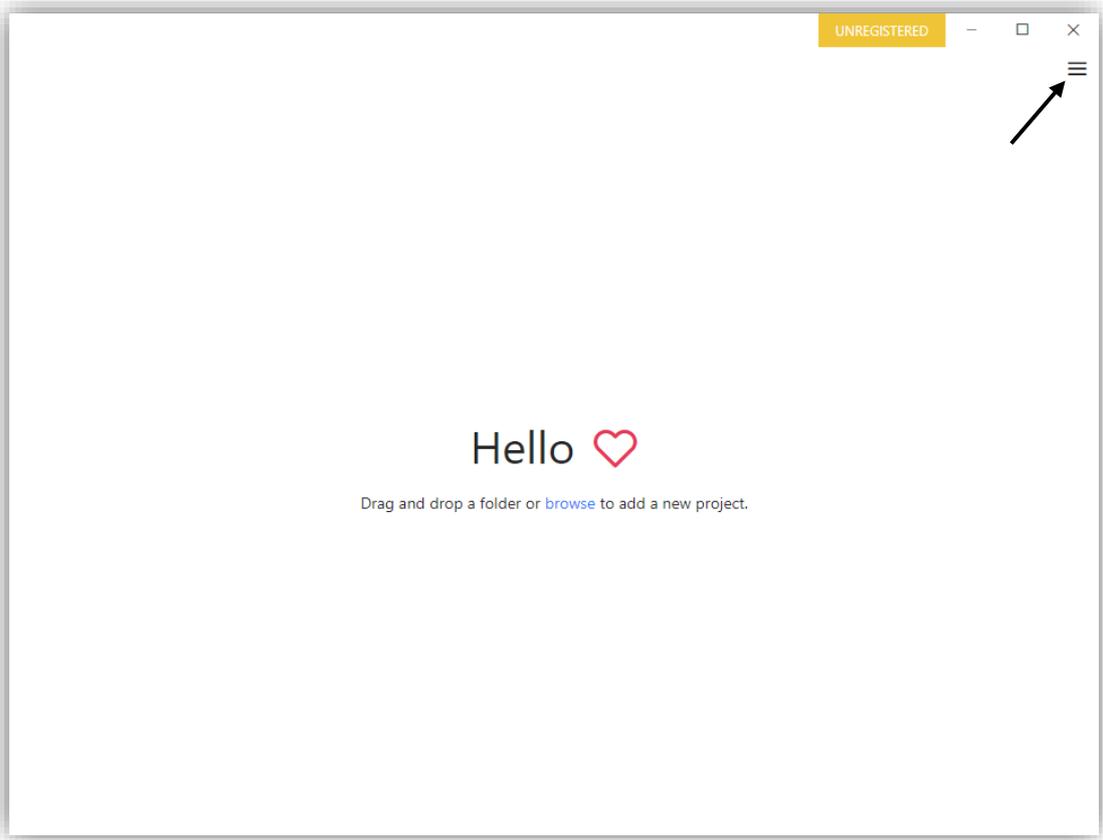
Lo descargamos.

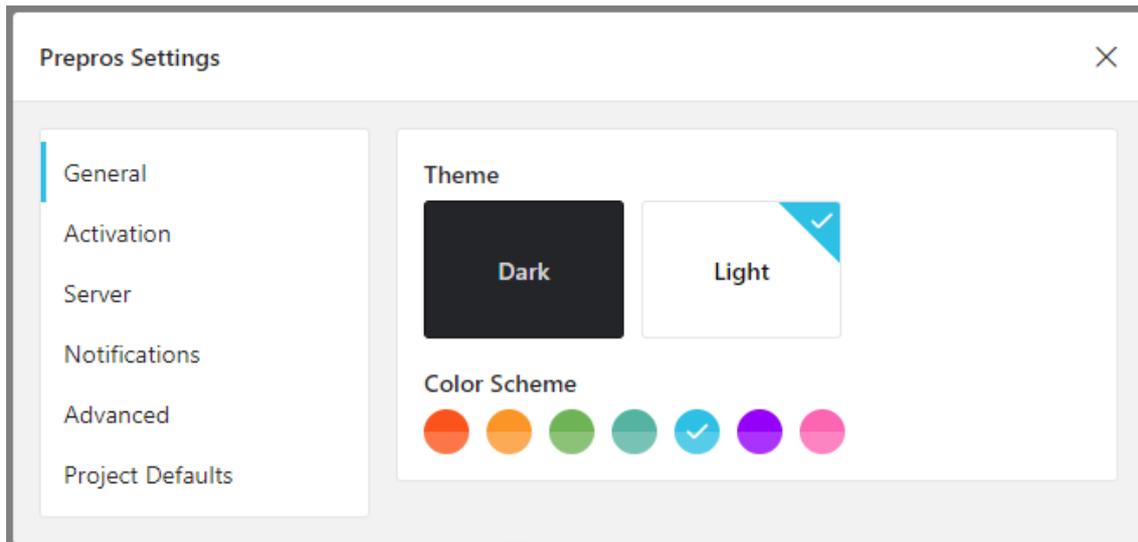


Elegimos nuestro sistema operativo.

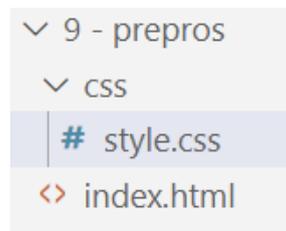


Y lo instalamos.





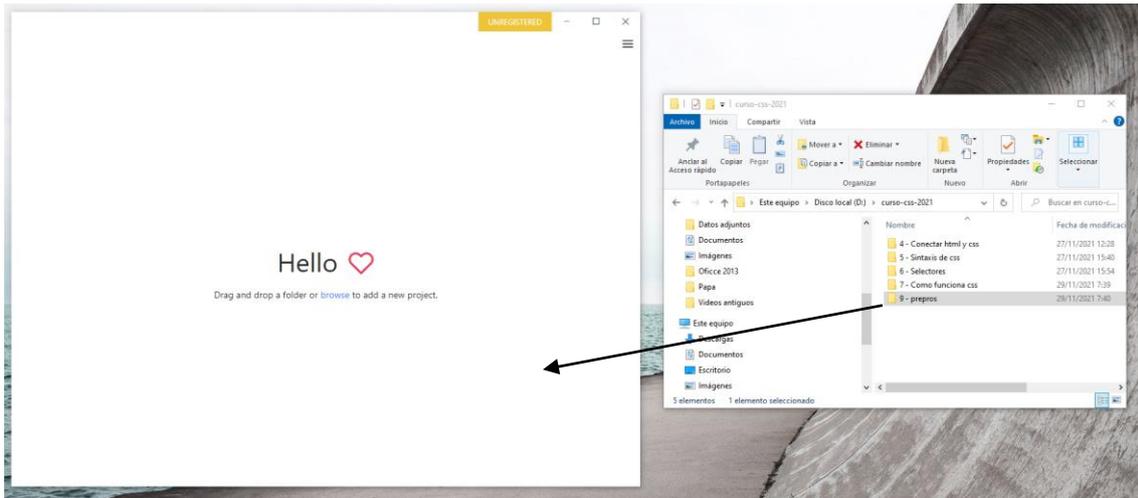
Yo voy a seleccionar el tema claro.
Vamos a crear un nuevo proyecto en Visual Studio Code.



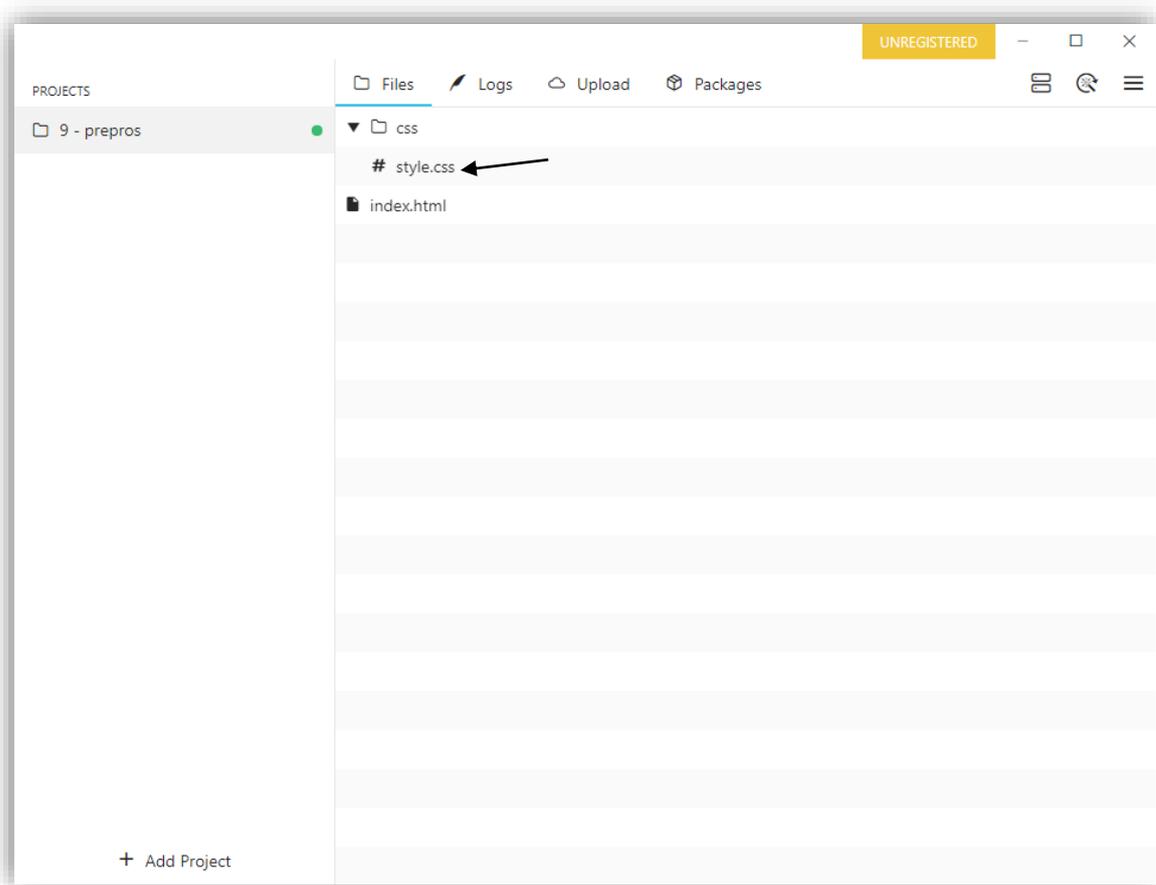
Rellenamos en index nuestra plantilla y lincamos stiles.css.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/style.css">
  <title>Document</title>
</head>
<body>

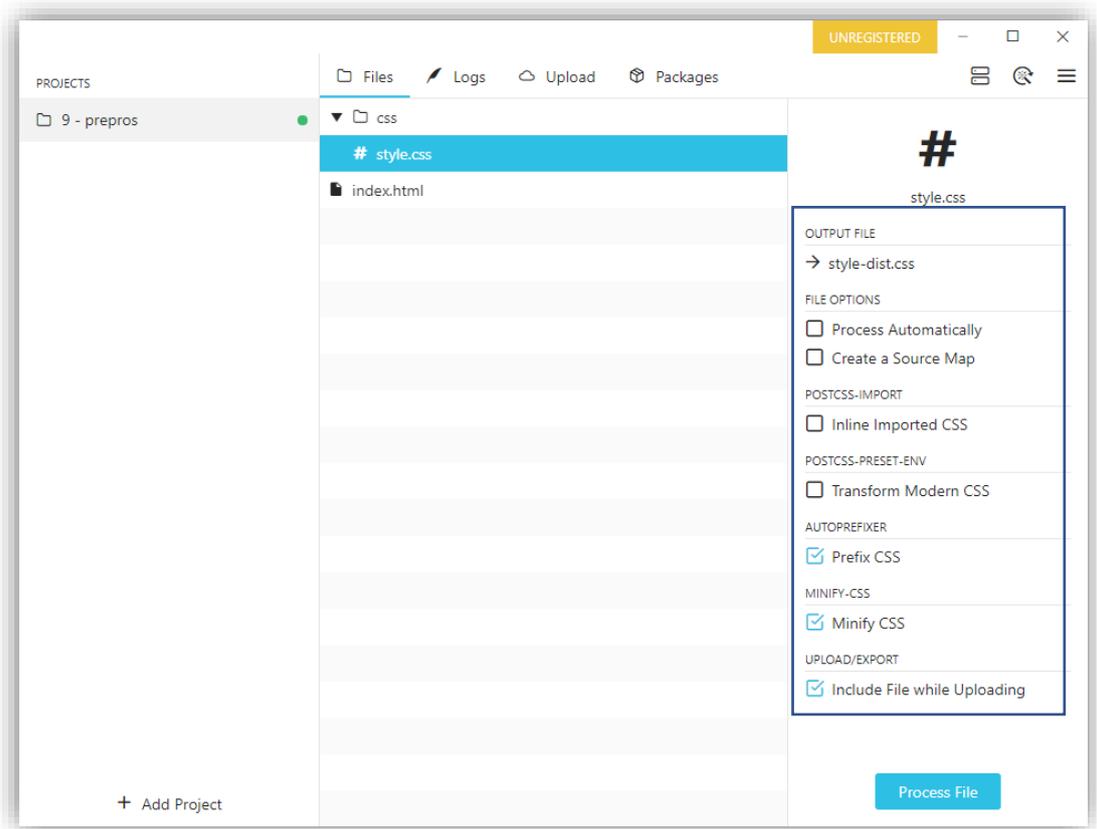
</body>
</html>
```



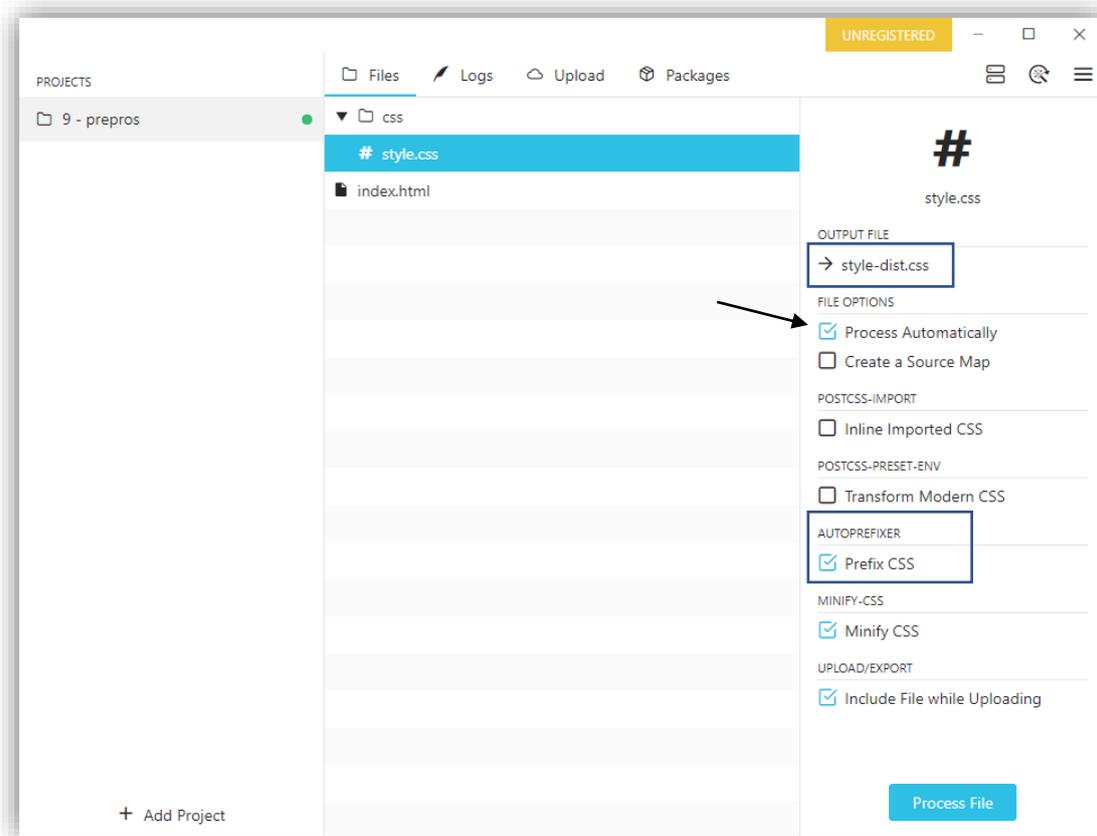
Arrastramos nuestra carpeta.



Al hacer clic en el archivo style.css.



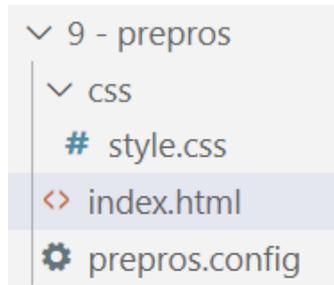
Muestra una serie de opciones.



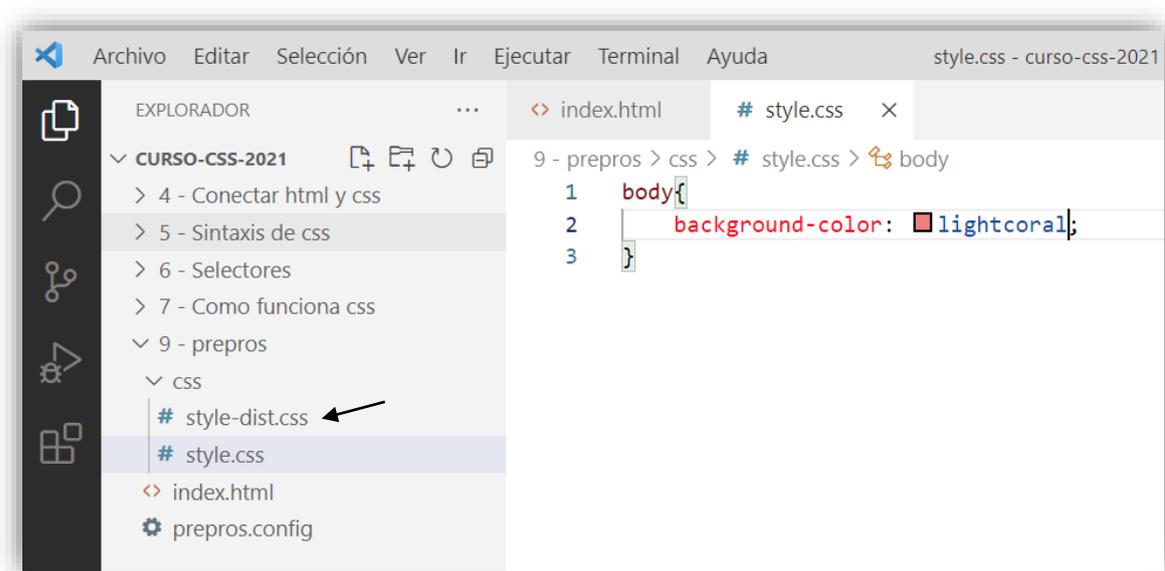
#YOAPRENDOCONDORIAN

Nos fijamos que en OUTPUT FILE pone style-dist.css.

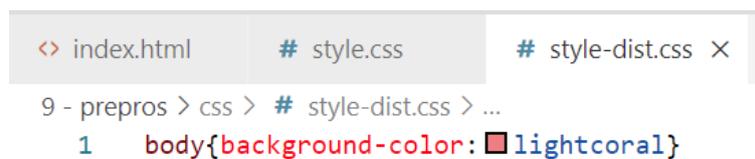
Además activamos la opción "Process Automáticamente además fijamos que la opción AUTOPREFIXER está marcada.



Se nos ha creado un archivo prepros.config que no tenemos que tocar.



En el momento que aplicamos un estilo se crea el nuevo estilo llamado style-dist.css.



Este archivo lo que hace es interpretarlo.

#YOAPRENDOCONDORIAN

```

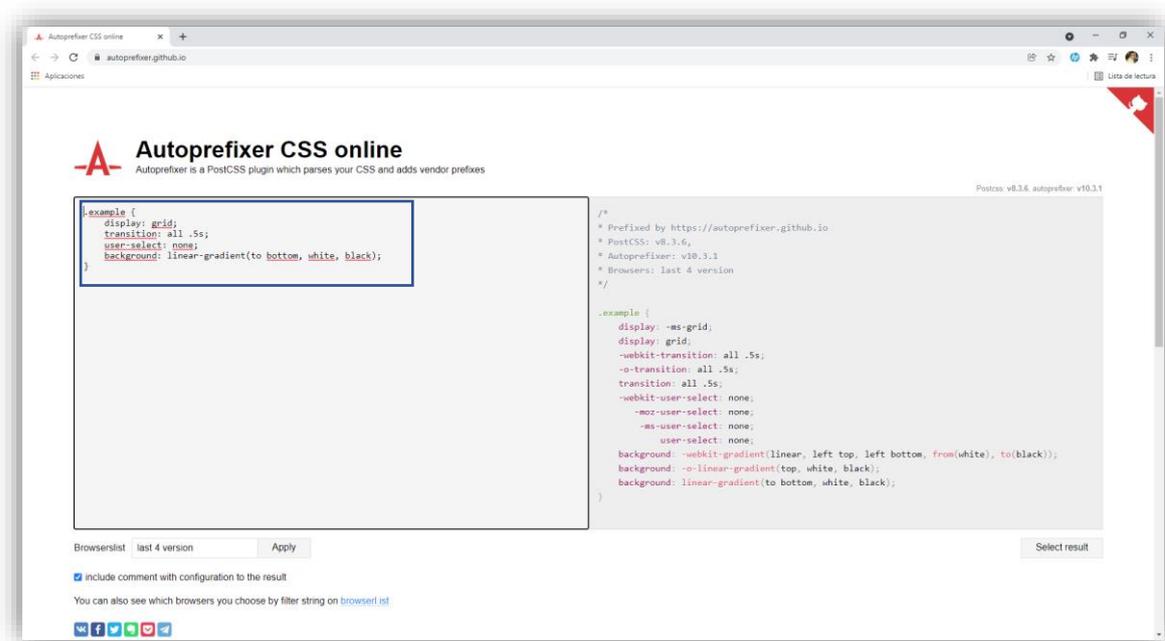
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <link rel="stylesheet" href="css/style-dist.css">
8      <title>Document</title>
9  </head>
10 <body>
11
12 </body>
13 </html>

```

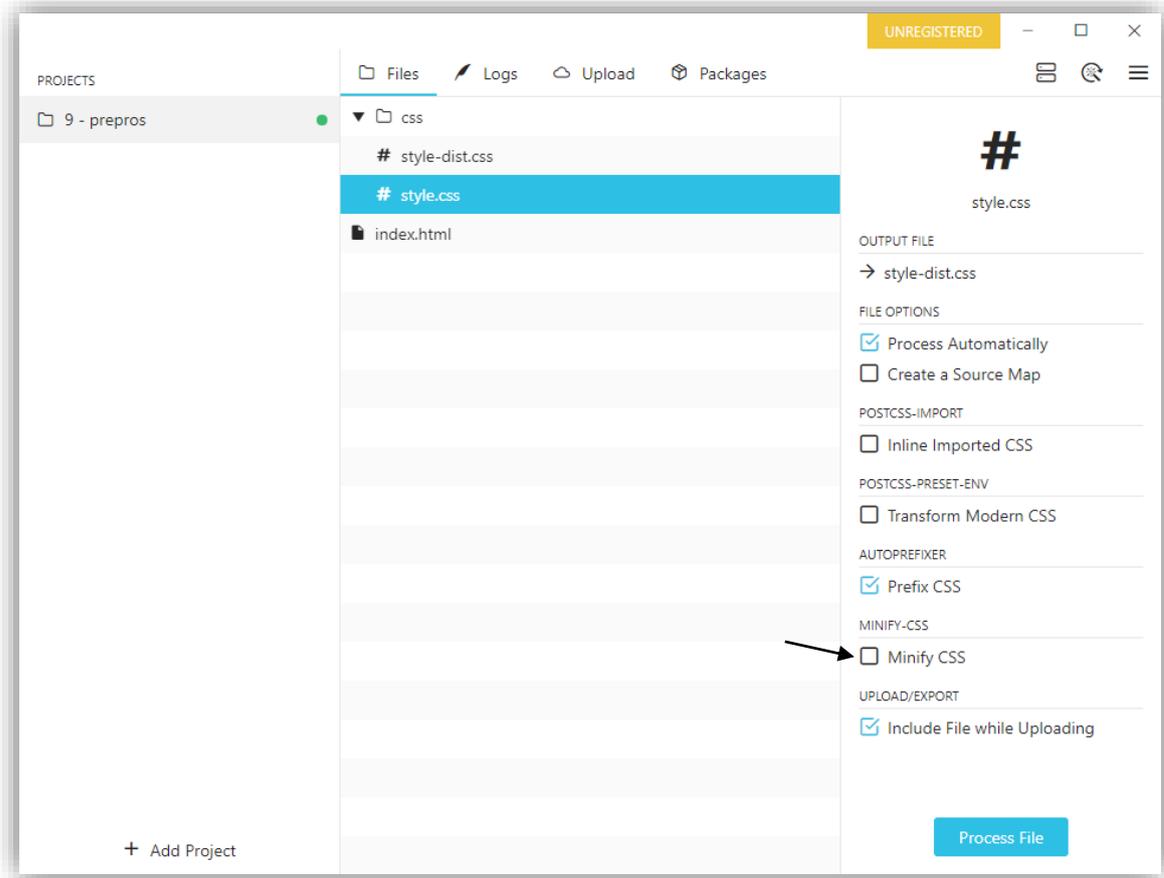
Cambiamos al nuevo documento css.

El archivo style-dist.css es únicamente para que lo interprete el navegador.

Vamos a volar a Autoprefixer css online.



Copiamos el código de la izquierda.



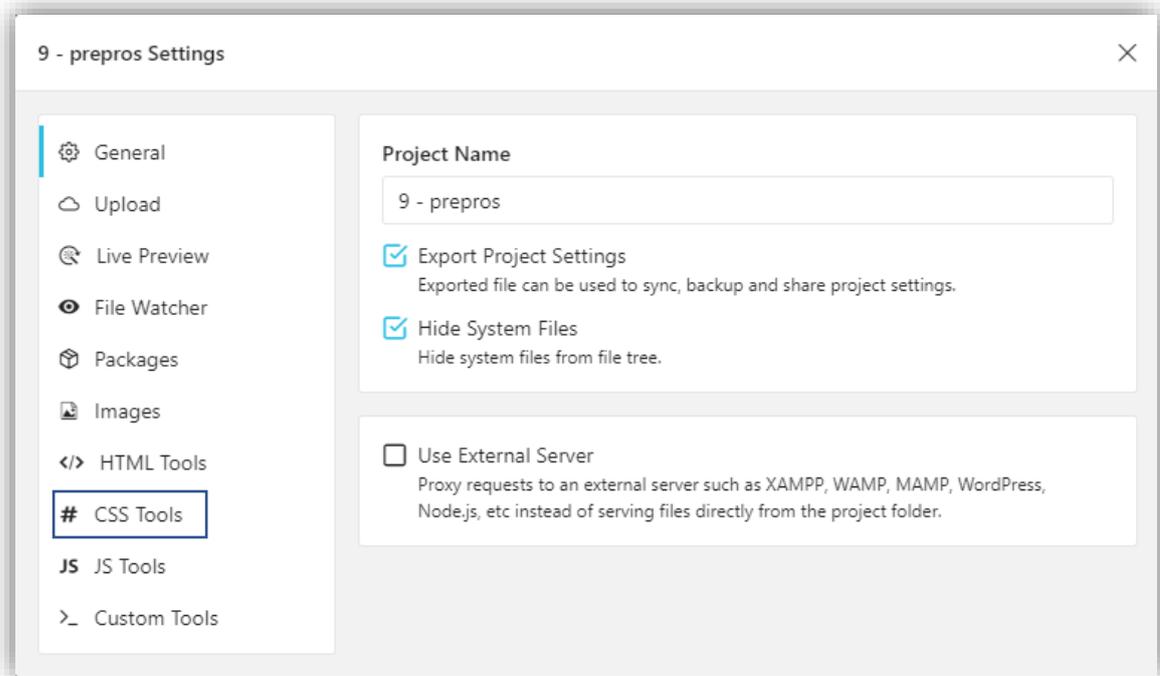
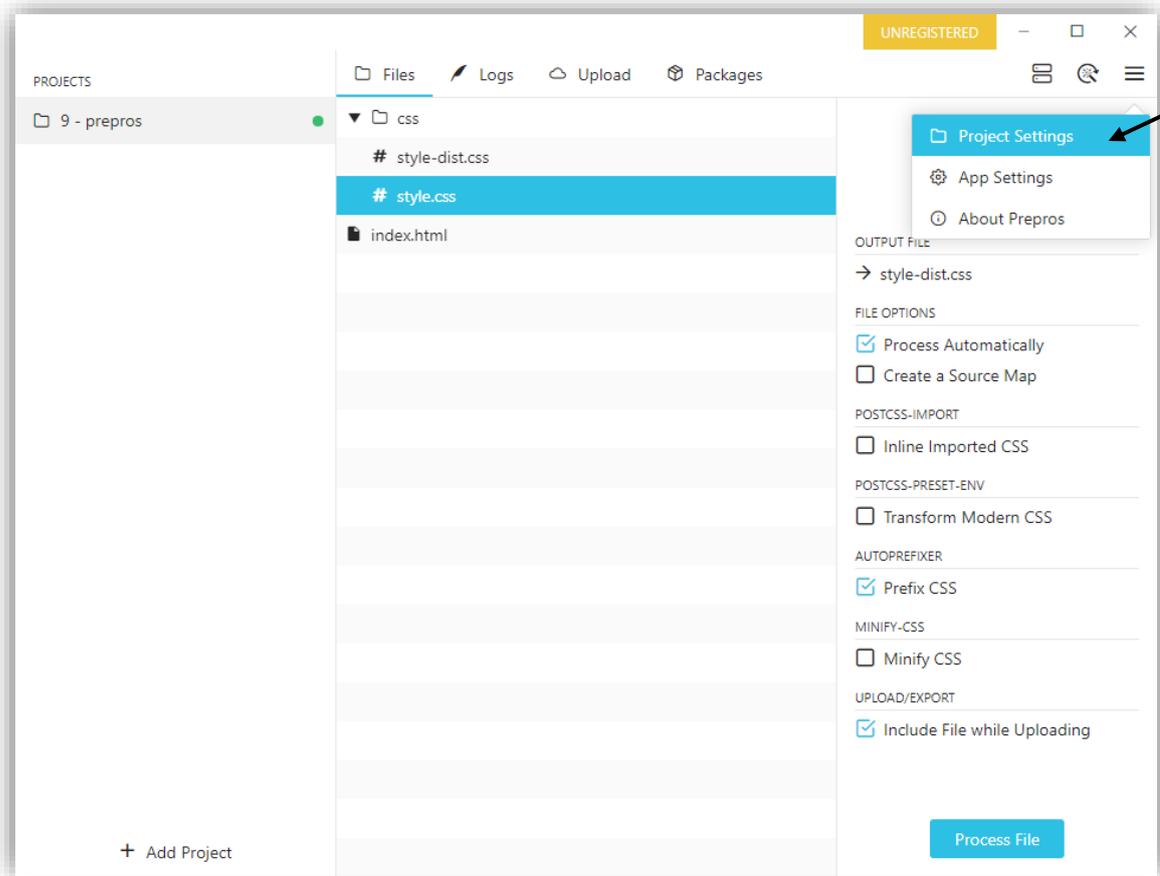
Desactivamos la opción Minify CSS.

```
<> index.html # style.css × # style-dist.css
9 - prepros > css > # style.css > ...
1  .example {
2      display: grid;
3      transition: all .5s;
4      user-select: none;
5      background: linear-gradient(to bottom, white, black);
6  }
```

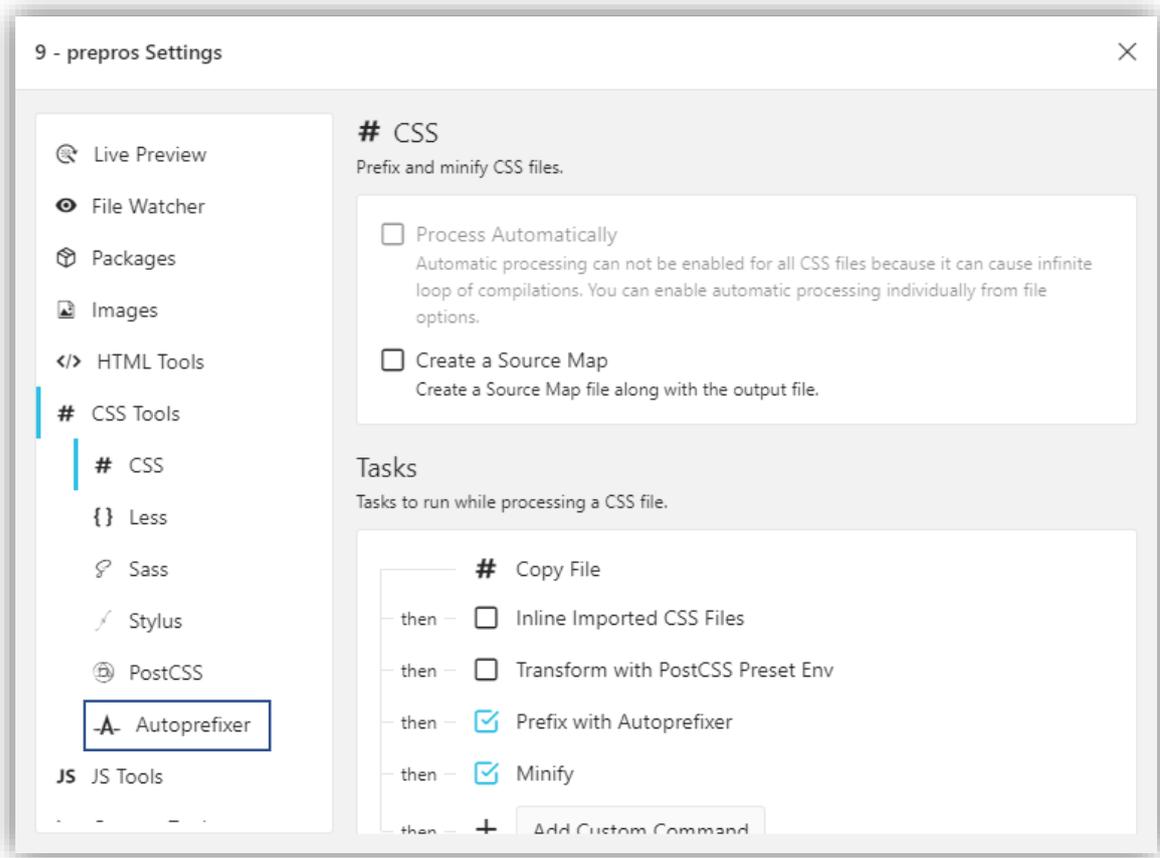
Lo pegamos en nuestro código. Ahora queremos ver el resultado en style-dist.css.

```
<> index.html # style.css # style-dist.css ×
9 - prepros > css > # style-dist.css > ...
1  .example {
2      display: -ms-grid;
3      display: grid;
4      -webkit-transition: all .5s;
5      transition: all .5s;
6      -webkit-user-select: none;
7      -moz-user-select: none;
8      -ms-user-select: none;
9      user-select: none;
10     background: -webkit-gradient(linear, left top, left bottom, from(white), to
11     (black));
12     background: linear-gradient(to bottom, white, black);
}
```

Si observamos que no sale igual que en el ejemplo de la página.

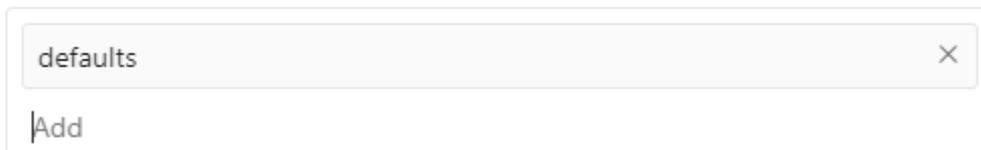


Seleccionamos CCS Tools

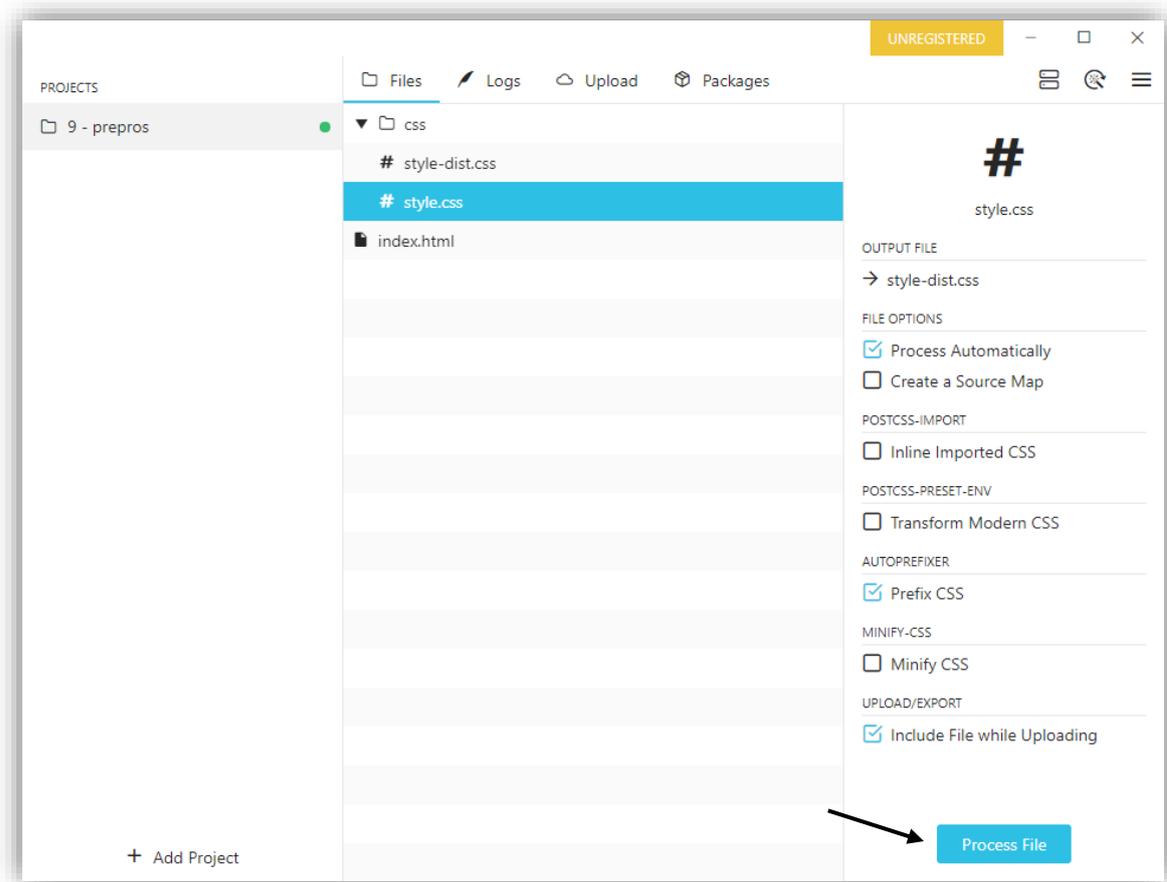


Seleccionamos Autoprefixer.

Target Browsers



En el apartado Target Browsers solo ponemos defaults.
Ya lo podemos cerrar.



Procesamos de nuevo.

Recuerda que nosotros vamos a escribir el código donde siempre pero el que está enlazado con html es el style-dist.css.

Box Model

Al final la web se compone de cajas.

Vamos a ver estas propiedades ya que es el principio básico que tenemos que saber para crear nuestra página web.



Si queremos que el borde se separe del contenido tenemos una propiedad que se denomina Padding (relleno).



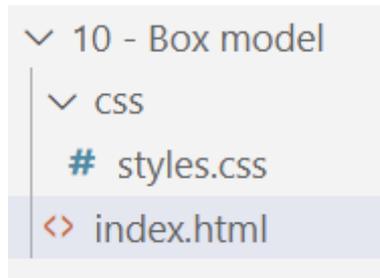
Para poder separar unas cajas de otras Margin.



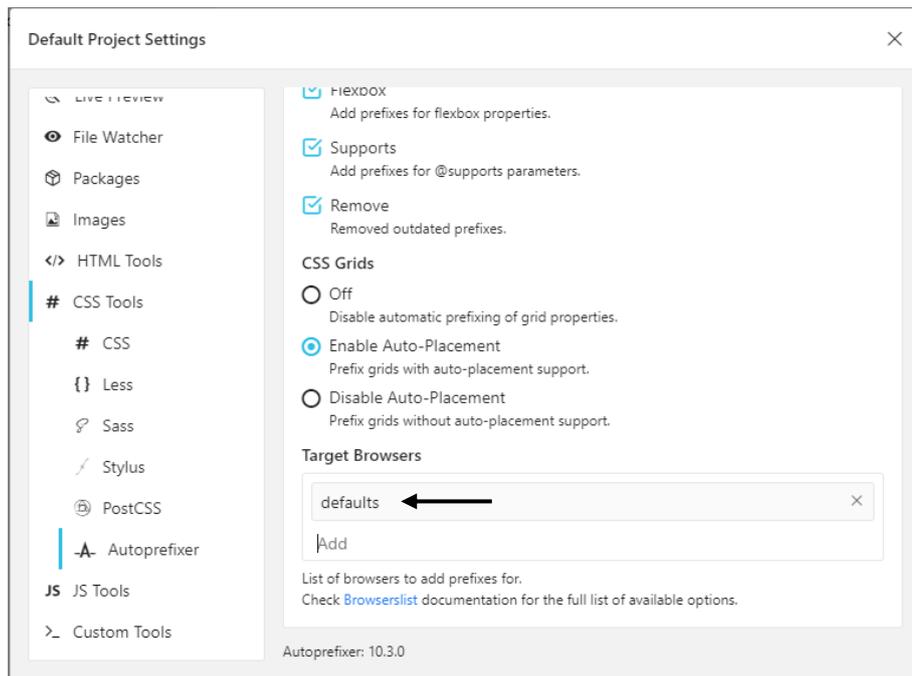
Ademas del ancho y alto de nuestra caja.

Width y Height

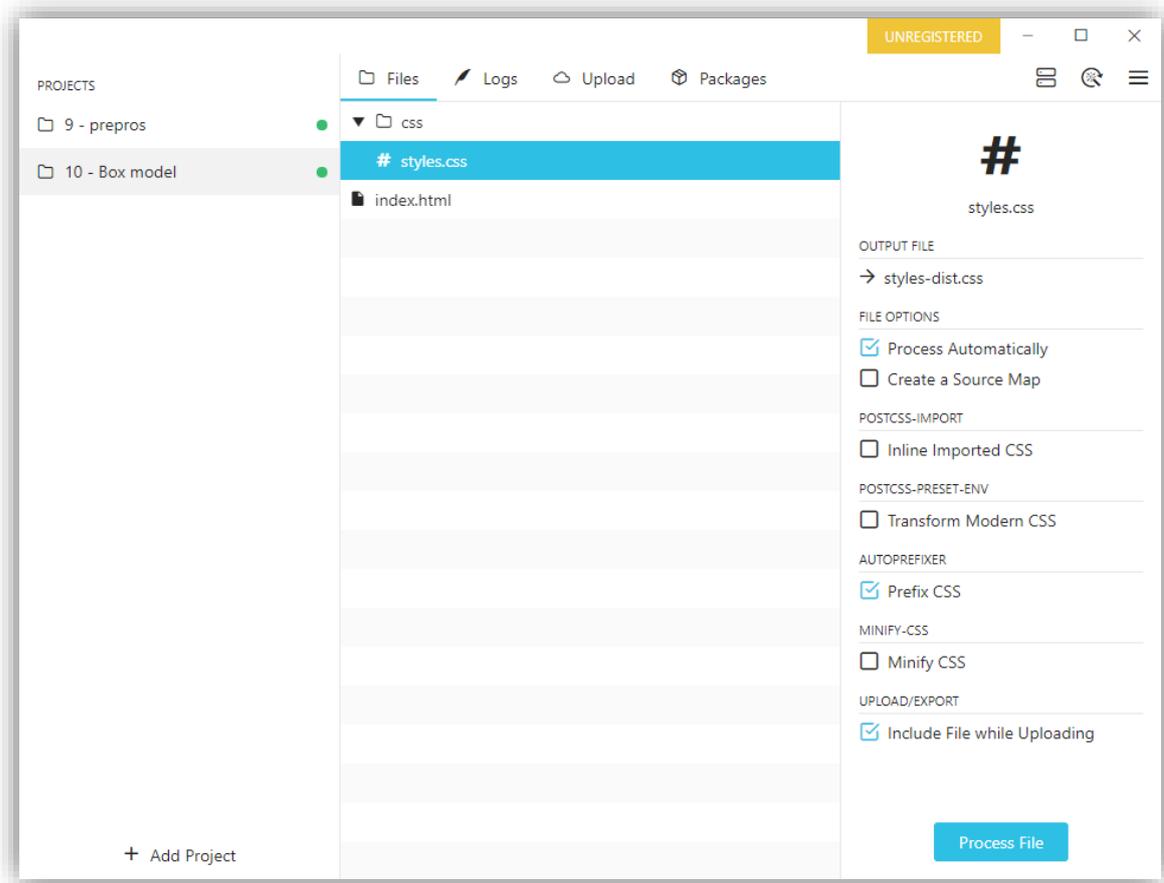
Para este proyecto vamos a realizar la siguiente estructura:



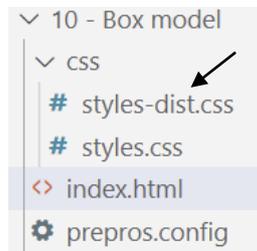
Configuramos Prepros:



En el capítulo anterior se explica paso a paso. (Página 173)
Arrastramos el nuevo proyecto.



Seleccionamos style.css seguido del botón procesar.



Ya tenemos nuestra nueva hoja de estilos.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles-dist.css">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

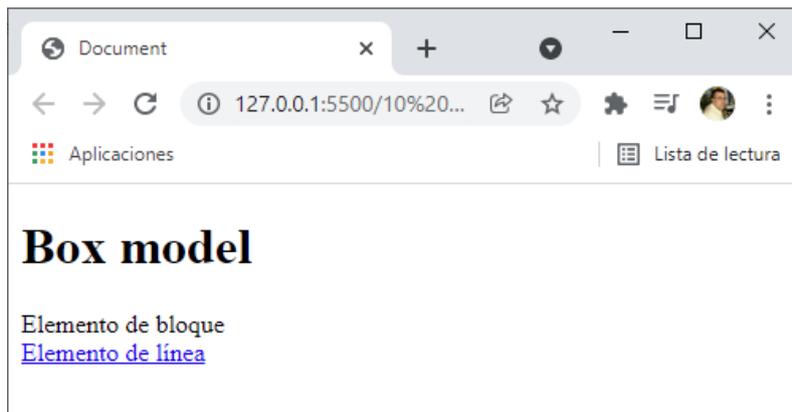
#YOAPRENDOCONDORIAN

La vinculamos no la nueva hoja de estilos.

Escribimos en siguiente código en HTML:

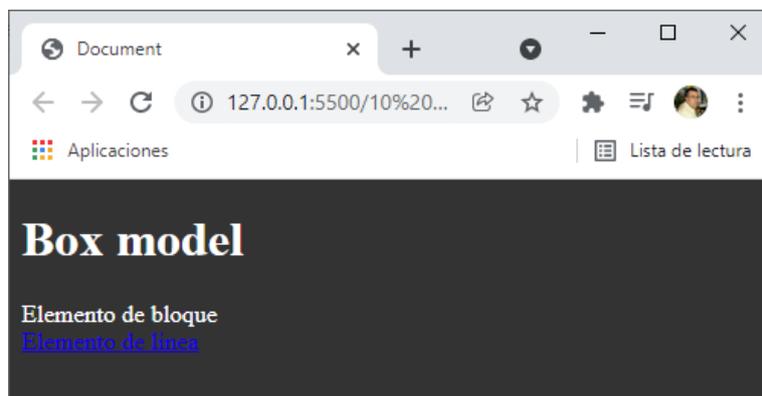
```
<body>
  <h1>Box model</h1>
  <div class="block">Elemento de bloque</div>
  <a href="#" class="inline">Elemento de línea</a>
</body>
```

Este será el resultado:



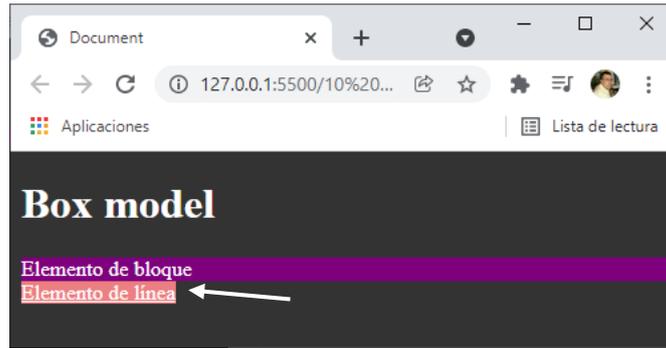
Nos vamos a style.css.

```
body{
  background-color: #333;
  color: #fff;
}
```



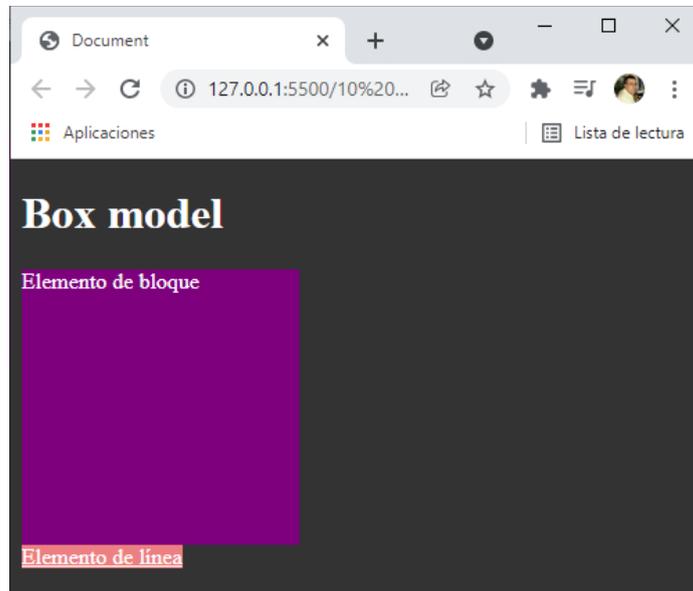
```
.block{
  background-color: purple;
}

.inline{
  background-color: lightcoral;
  color: inherit;
}
```



Los elementos de línea no tienen medida si los elementos de bloque.

```
.block{  
  background-color: purple;  
  width: 200px;  
  height: 200px;  
}
```



Margin

Es la propiedad que nos permite generar espacio entre elementos.

Es un shorthand (propiedad abreviada) que controla los 4 lados posibles a los que dar márgenes.

margin-top: Margen superior.

margin-right: Marge derecho.

margin-bottom: Margen inferior.

margin-left: Margen izquierdo.

Admiten hasta 4 valores que van en el orden de las agujas del reloj.

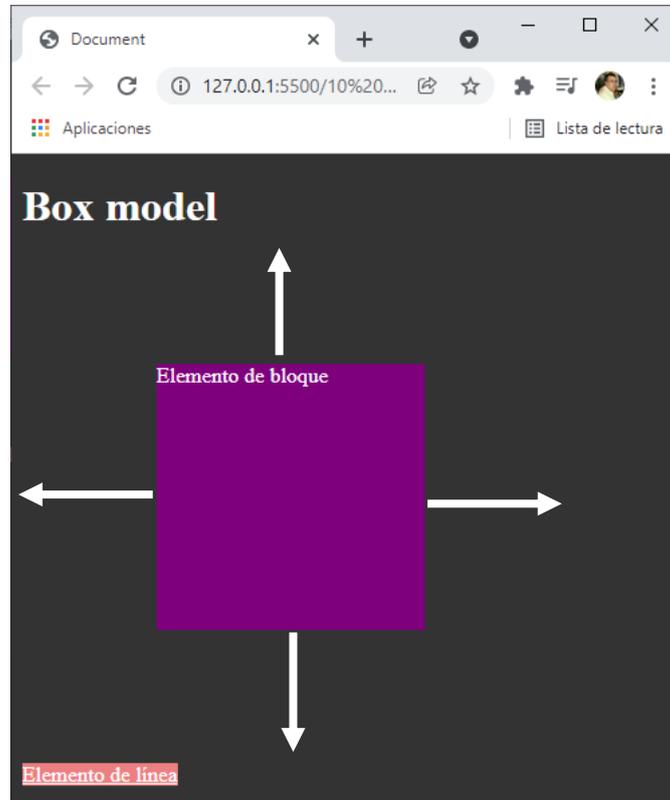
4 valores -> margin: top right bottom left;

3 valores -> margin: top left/right bottom;

2 valores -> margin: top/bottom left/right;

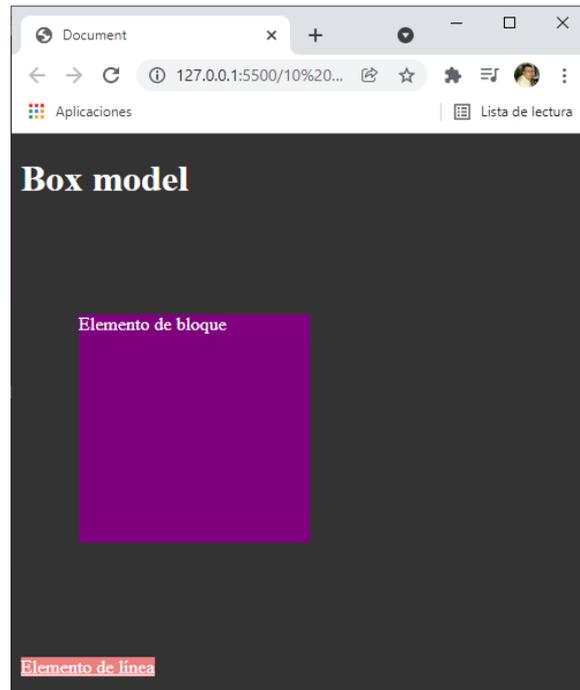
1 valor -> margin: top/right/bottom/left;

```
.block{  
  background-color: purple;  
  width: 200px;  
  height: 200px;  
  margin: 100px;  
}
```



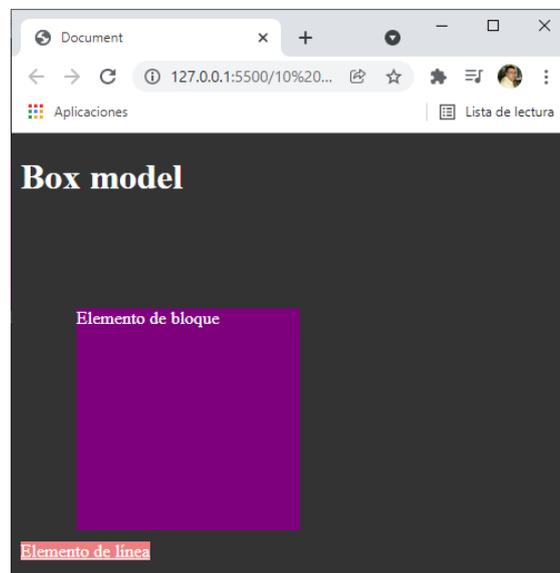
100 pixeles por cada lado.

```
.block{  
  background-color: purple;  
  width: 200px;  
  height: 200px;  
  margin: 100px 50px;  
}
```



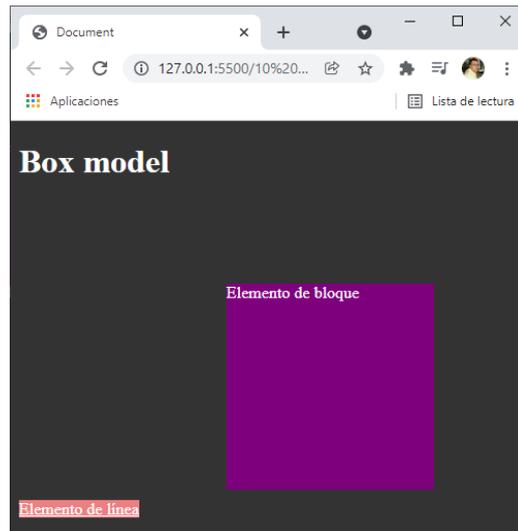
100 pixeles arriba y abajo y 50 pixeles derecha e izquierda.

```
.block{  
  background-color: purple;  
  width: 200px;  
  height: 200px;  
  margin: 100px 50px 100px;  
}
```



100 pixeles arriba, 50 pixeles a derecha e izquierda y 10 pixeles abajo.

```
.block{  
  background-color: purple;  
  width: 200px;  
  height: 200px;  
  margin: 100px 50px 10px 200px;  
}
```



100 pixeles arriba, 50 pixeles derecha, 10 pixeles abajo y 200 pixeles izquierda.

```
<body>  
  <h1>Box model</h1>  
  → <a href="#" class="inline">Elemento de línea 1</a>  
  → <a href="#" class="inline">Elemento de línea 2</a>  
  <div class="block">Elemento de bloque</div>  
</body>
```

Hemos introducido dos elementos en línea.



```
.inline{  
  background-color: lightcoral;  
  color: inherit;  
  margin-right: 100px;  
}
```



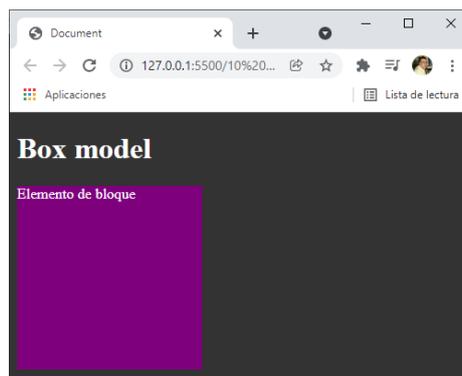
También funciona con margin-left: 100px;



Los elementos de línea solo tienen márgenes horizontales en cambio los elementos de bloque tienen márgenes verticales y horizontales.

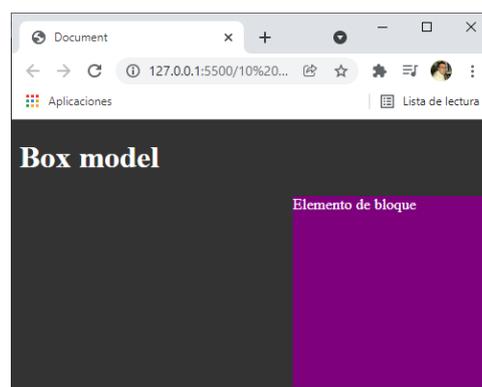
Algunos trucos:

```
.block{
  background-color: ■purple;
  width: 200px;
  height: 200px;
  margin-right: auto;
}
```



Pegamos el elemento a la izquierda.

```
.block{
  background-color: ■purple;
  width: 200px;
  height: 200px;
  margin-left: auto;
}
```



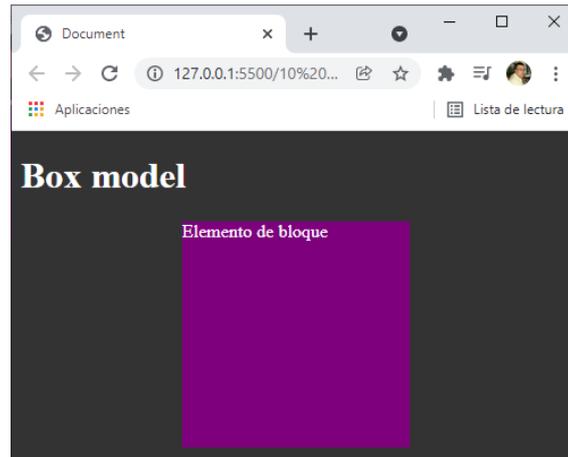
#YOAPRENDOCONDORIAN

Pegamos el elemento a la derecha.

Estas propiedades funcionan si tenemos el ancho declarado.

```
.block{
  background-color: purple;
  width: 200px;
  height: 200px;
  margin-left: auto;
  margin-right: auto;
}
```

Si ponemos margin-left y margin-right en auto, lo que conseguimos es centrarlo.



Errores comunes con Margin

```
*{
  margin: 0;
  padding: 0;
}
```

Quitar todos los márgenes y rellenos en todo un documento se considera una práctica poco aconsejable.

```
.block{
  Background-color: purple;
  Width: 200px;
  Height: 200px;
  margin-top: 100px;
  margin: 0 auto;
}
```

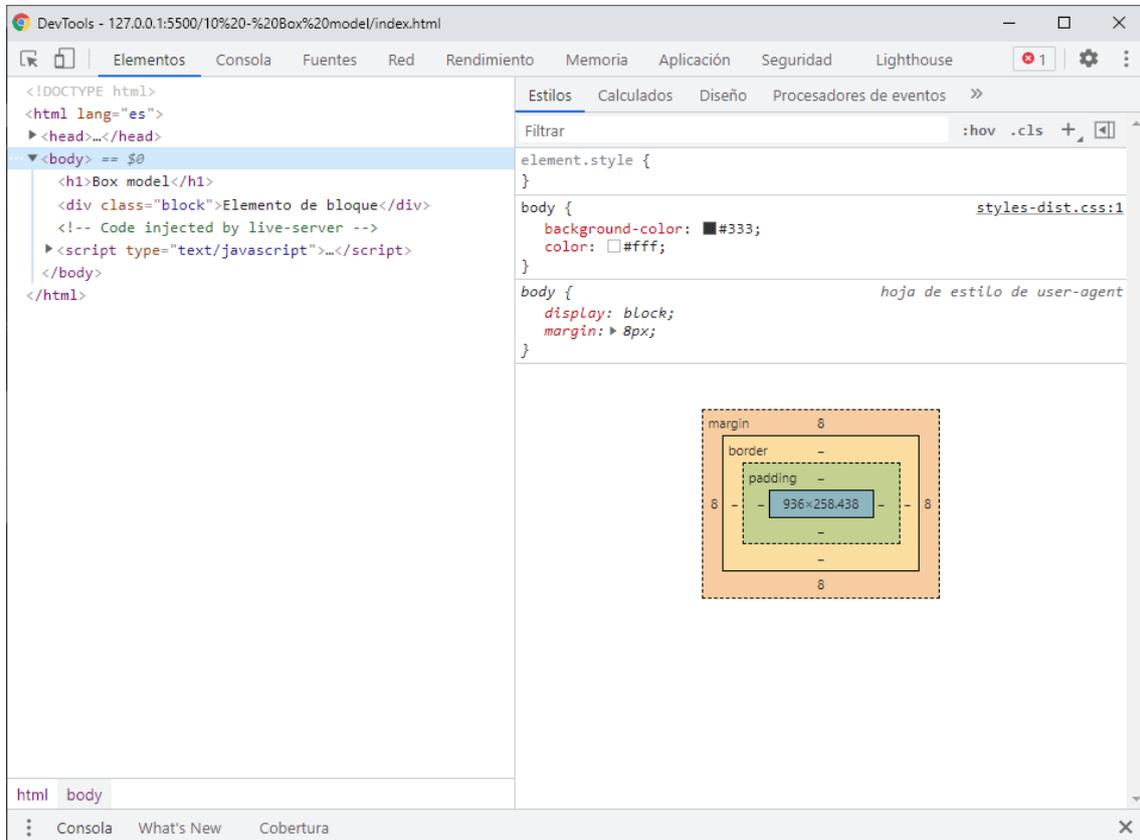
La instrucción margin-top: 100px; será sobrescrita por margin: 0 auto; por el sistema de cascada.

Lo correcto sería:

```
.block{
  Background-color: purple;
  Width: 200px;
  Height: 200px;
```

```
margin-left: auto;
margin-right: auto;
}
```

Poner un padding y un border a 0 en el body no tiene sentido ya que no lo tiene.



Padding

Es la propiedad que nos permite generar espacio interno entre el borde y la caja.

Es un shorthand (propiedad abreviada) que controla los 4 lados posibles a los que dar padding.

padding-top: padding superior.

padding-right: padding derecho.

padding-bottom: padding inferior.

padding-left: padding izquierdo.

Admite hasta 4 valores que van en el orden de las agujas del reloj.

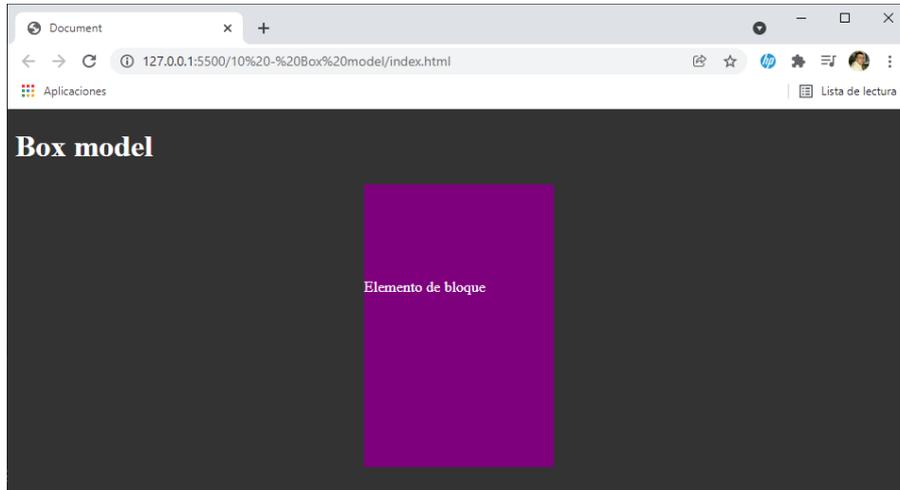
4 valores -> padding: top right bottom left;

3 valores -> padding: top left/right bottom;

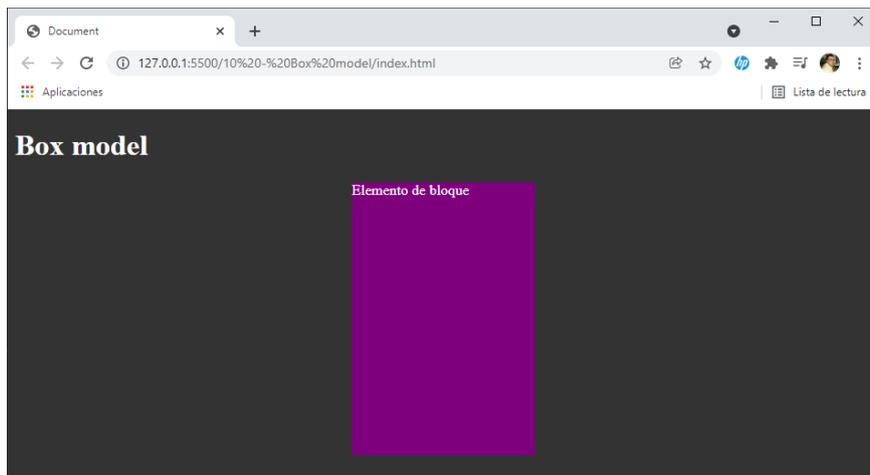
2 valores -> padding: top/bottom left/right;

1 valor -> top/right/bottom/left;

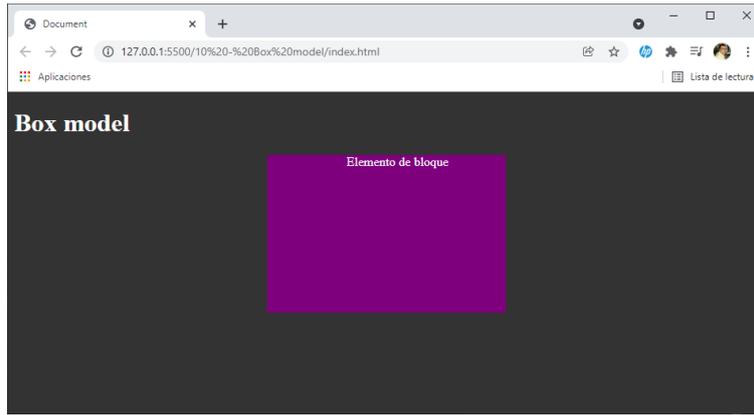
```
.block{  
  background-color: ■purple;  
  width: 200px;  
  height: 200px;  
  margin-left: auto;  
  margin-right: auto;  
  padding-top: 100px;  
}
```



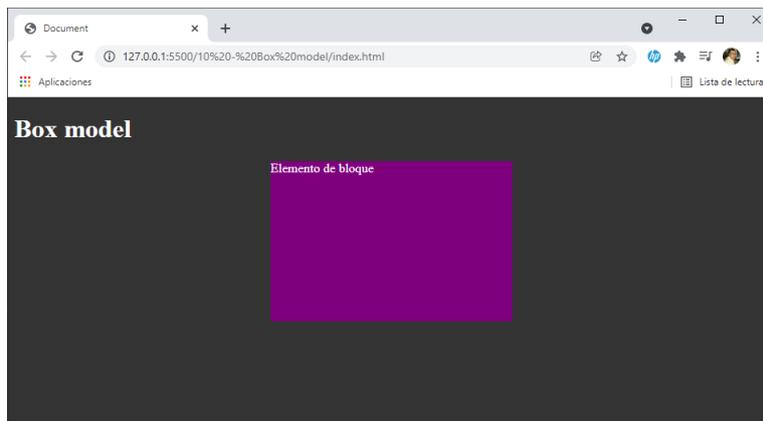
```
.block{  
  background-color: ■purple;  
  width: 200px;  
  height: 200px;  
  margin-left: auto;  
  margin-right: auto;  
  padding-bottom: 100px;  
}
```



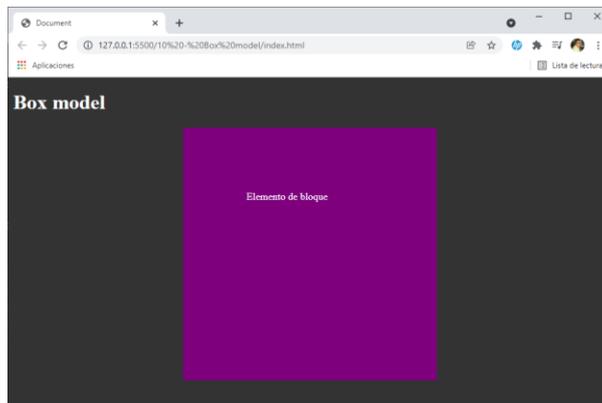
```
.block{  
  background-color: ■purple;  
  width: 200px;  
  height: 200px;  
  margin-left: auto;  
  margin-right: auto;  
  padding-left: 100px;  
}
```



```
.block{  
  background-color: purple;  
  width: 200px;  
  height: 200px;  
  margin-left: auto;  
  margin-right: auto;  
  padding-right: 100px;  
}
```



```
.block{  
  background-color: purple;  
  width: 200px;  
  height: 200px;  
  margin-left: auto;  
  margin-right: auto;  
  padding: 100px;  
}
```



Podemos poner 1 valor, 2 valores, 3 valores o 4 valores igual que margin.

Con los elementos de línea funciona exactamente igual.

Resumiendo si el objetivo es separar vuestra caja de otra utilizaremos margin.

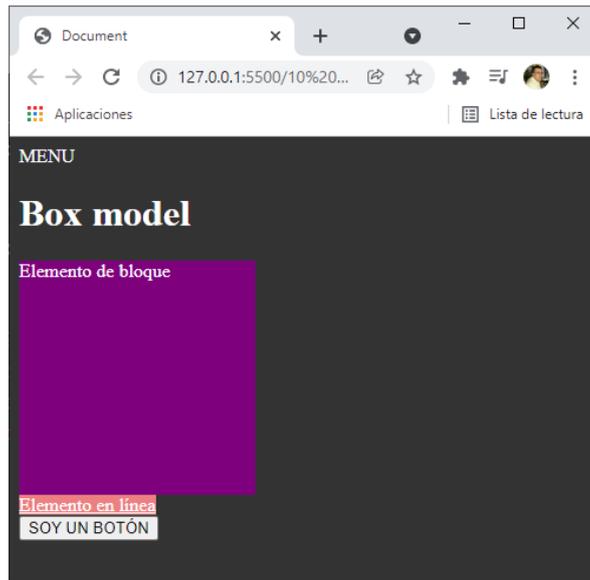
Si queremos aumentar el tamaño de la caja utilizaremos padding.

Vamos a realizar una práctica.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles-dist.css">
  <title>Document</title>
</head>
<body>
  <header class="header">
    <nav class="menu">MENU</nav>
  </header>
  <h1 class="title">Box model</h1>
  <div class="block">Elemento de bloque</div>
  <a href="#" class="inline">Elemento en línea</a>
  <div class="buttons">
    <button class="buttons">SOY UN BOTÓN</button>
  </div>
</body>
</html>
```

```
1  body{
2    background-color: #333;
3    color: #fff;
4  }
5
6  .block{
7    background-color: purple;
8    width: 200px;
9    height: 200px;
10 }
11
12 .inline{
13   background-color: lightcoral;
14   color: inherit;
15 }
```

Este será el resultado:

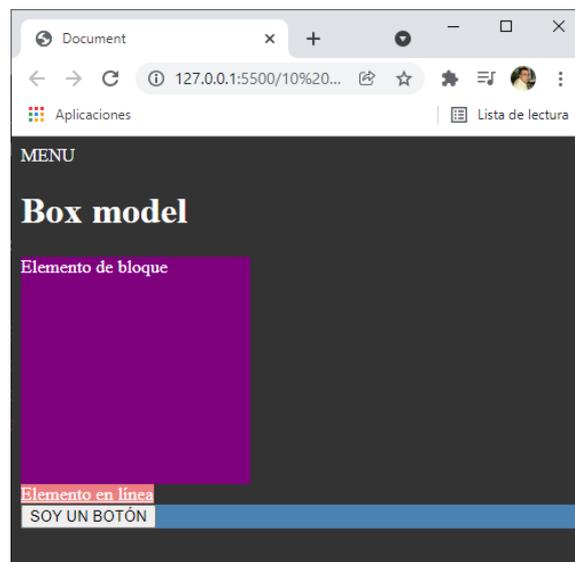


```
<body>
  <header class="header">
    <nav class="menu">MENU</nav>
  </header>
  <h1 class="title">Box model</h1>
  <div class="block">Elemento de bloque</div>
  <a href="#" class="inline">Elemento en línea</a>
  <div class="buttons">
    <button class="button">SOY UN BOTÓN</button>
  </div>
</body>
```

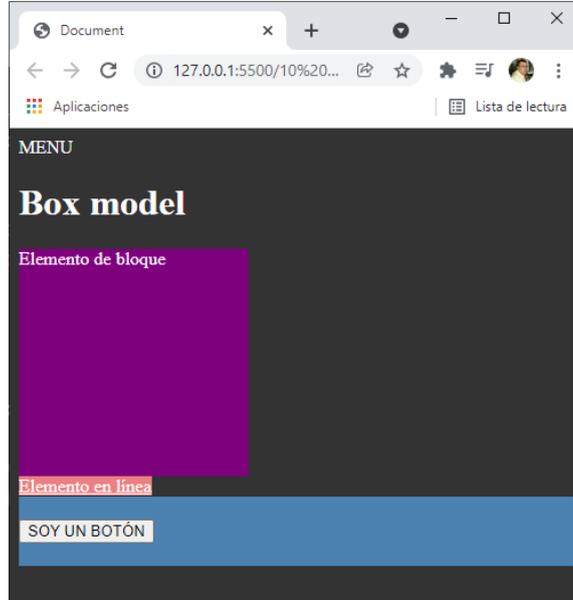
Este código es en html.

```
.buttons{
  background-color: steelblue;
}
```

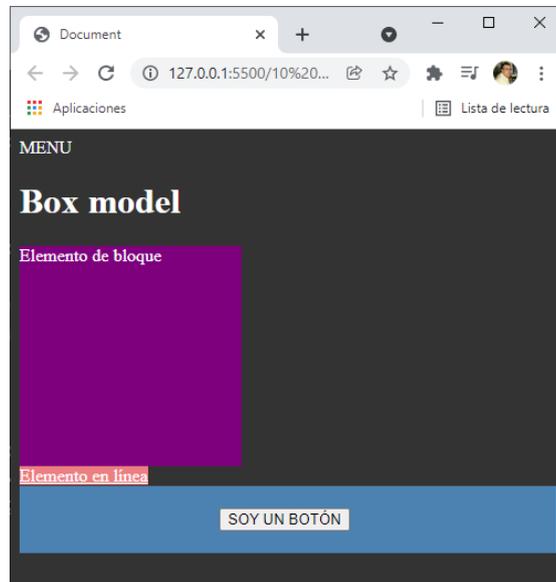
Este será el resultado:



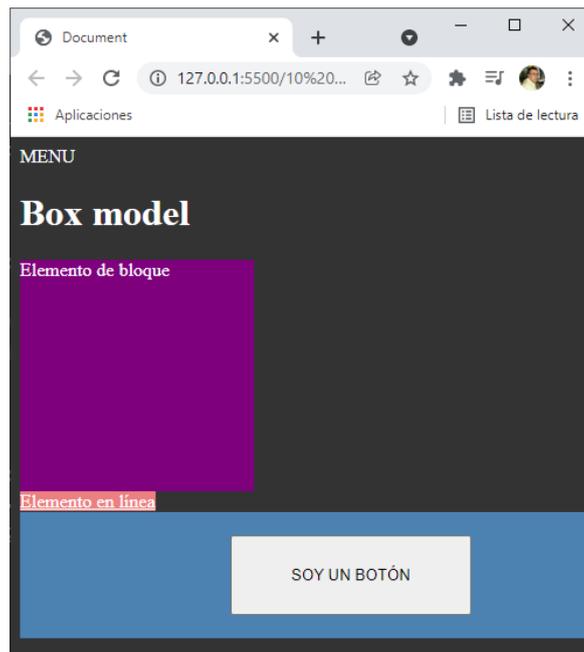
```
.buttons{  
  background-color: steelblue;  
  padding-top: 20px;  
  padding-bottom: 20px;  
}
```



```
.button{  
  margin-left: 180px;  
}
```



```
.button{  
  margin-left: 180px;  
  padding: 25px 50px;  
}
```



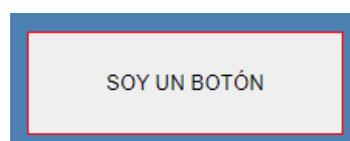
Border

Es la propiedad que nos permite modificar el borde de la caja.

Es un shorthand (propiedad abreviada) que agrupa 3 propiedades:

- border-width: ancho del borde.
 - border-top-width
 - border-right-width
 - border-bottom-width
 - border-left-width
- border-style: estilo de borde.
 - border-top-style
 - border-right-style
 - border-bottom-style
 - border-left-style
- border-color: color del borde.
 - border-top-color
 - border-right-color
 - border-bottom-color
 - border-left-color

```
.button{  
  margin-left: 180px;  
  padding: 25px 50px;  
  border: 1px solid red;  
}
```



#YOAPRENDOCONDORIAN

```
.button{
  margin-left: 180px;
  padding: 25px 50px;
  border:20px solid red;
}
```



El estilo border: 20px solid red;

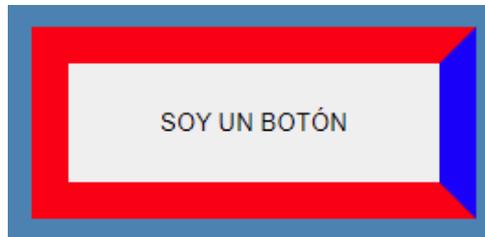
Engloba 3 estilos:

border-width: 20px;

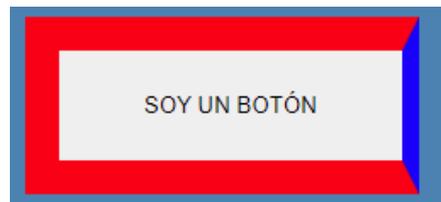
border-style: solid;

border-color: red;

```
.button{
  margin-left: 180px;
  padding: 25px 50px;
  border:20px solid red;
  border-right-color: blue;
}
```



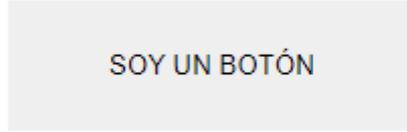
```
.button{
  margin-left: 180px;
  padding: 25px 50px;
  border:20px solid red;
  border-right-color: blue;
  border-right-width: 10px;
}
```



#YOAPRENDOCONDORIAN

Listado de valores style:

- none



- hidden



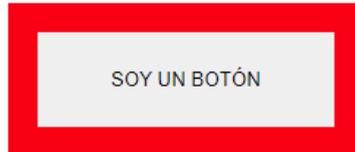
- dotted



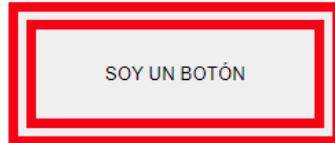
- dashed



- solid



- double



- groove



- ridge

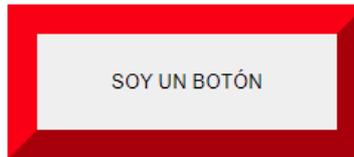


#YOAPRENDOCONDORIAN

- inset



- outset



Box sizing

Es la propiedad que nos permite controlar el cálculo que hace el navegador a la hora de modificar las propiedades content, padding y border.

Los dos valores que podemos darle son:

- content-box -> Valor por defecto
- border-box -> Cálculo de tamaño del elemento incluyendo el padding y el border.

DevTools - 127.0.0.1:5500/10%20-%20Box%20model/index.html

Elementos Consola Fuentes Red Rendimiento Memoria Aplicación Seguridad Lighthouse

Estilos Calculados Diseño Procesadores de eventos

```

<!DOCTYPE html>
<html lang="es">
  <head>...</head>
  <body>
    <header class="header">...</header>
    <h1 class="title">Box model</h1>
    <div class="block">Elemento de bloque</div> == $0
    <!--<a href="#" class="inline">Elemento en línea</a>
    <div class="buttons">
      <button class="button">SOY UN BOTÓN</button>
    </div-->
    <!-- Code injected by live-server -->
    <script type="text/javascript">...</script>
  </body>
</html>

```

margin -

border -

padding -

200x200

Filtrar Mostrar todo Agrupar

- ▶ background-color ■ rgb(128, 0, 128)
- ▶ color □ rgb(255, 255, 255)
- ▶ display block
- ▶ height 200px
- ▶ width 200px

Fuentes renderizadas

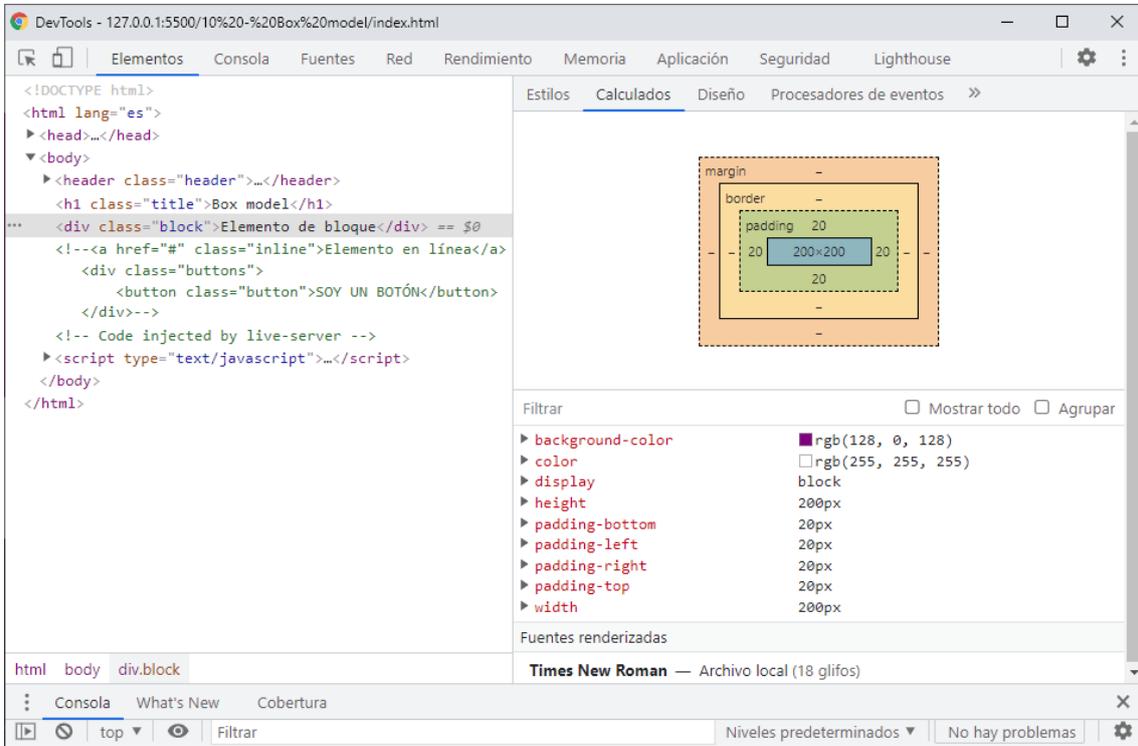
Times New Roman — Archivo local (18 glifos)

html body div.block

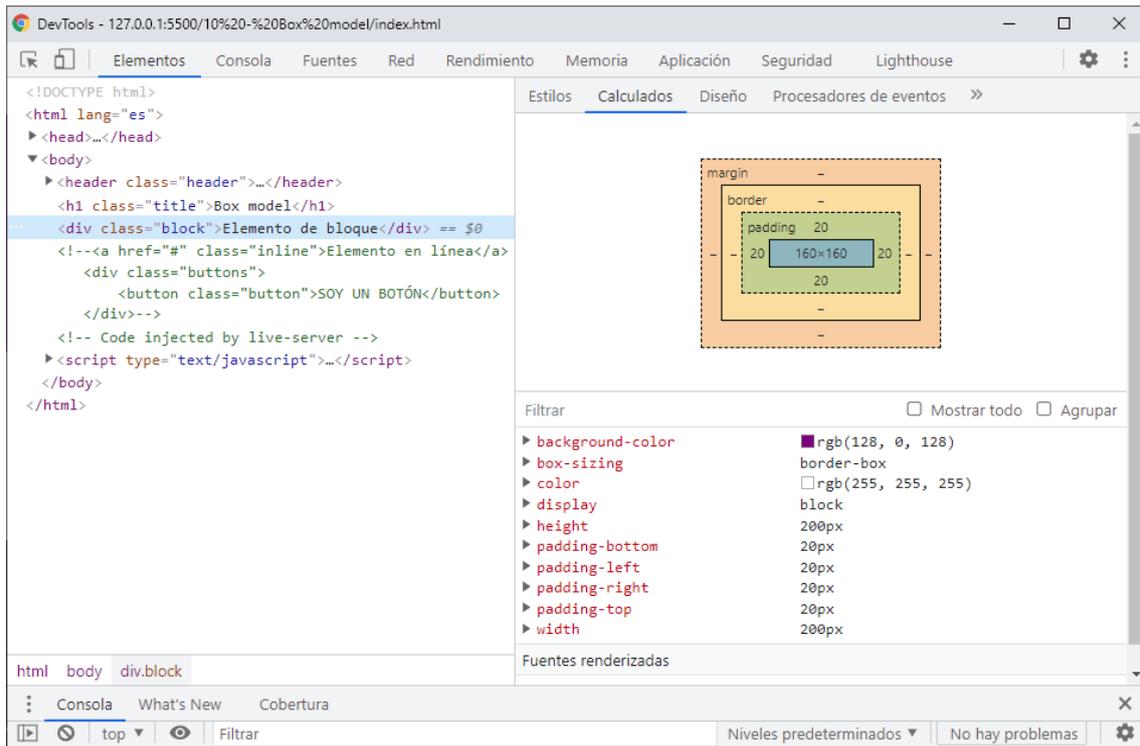
Consola What's New Cobertura

Filtrar Niveles predeterminados No hay problemas

```
.block{  
  background-color: ■ purple  
  width: 200px;  
  height: 200px;  
  padding: 20px;  
}
```



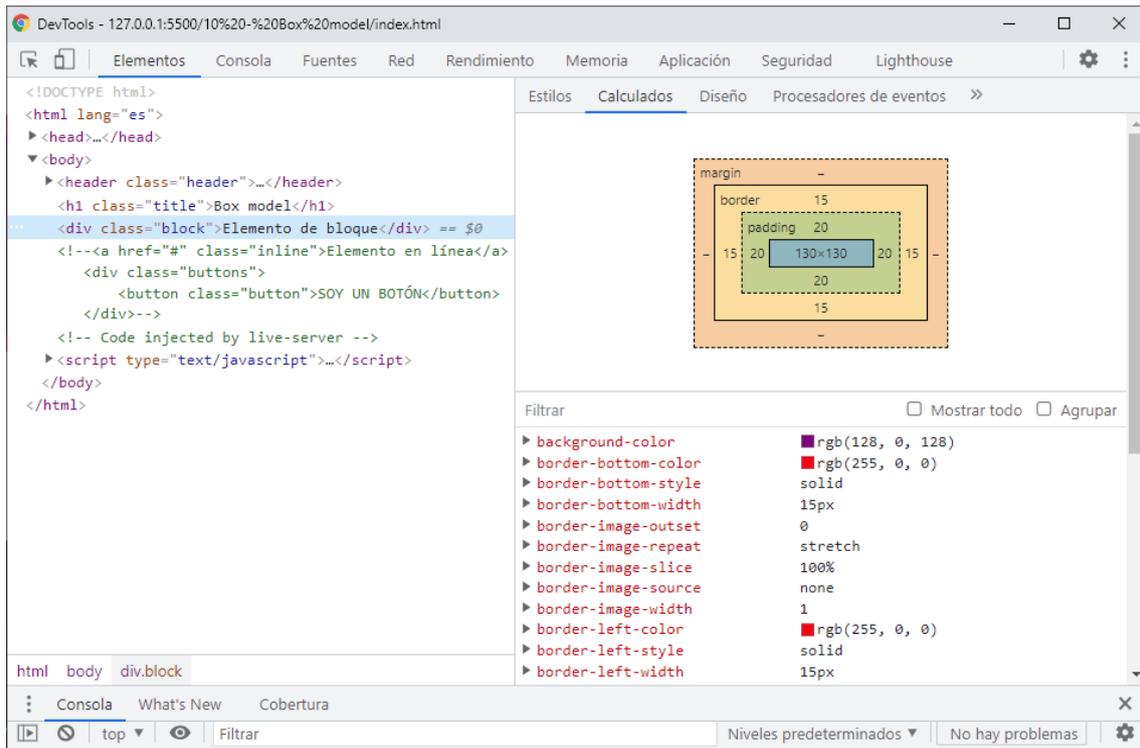
```
.block{  
  background-color: ■ purple;  
  width: 200px;  
  height: 200px;  
  padding: 20px;  
  box-sizing: border-box;  
}
```



Recalcula con el padding la caja tenga unas dimensiones de 200px x 200px.

```
.block{
  background-color: purple;
  width: 200px;
  height: 200px;
  padding: 20px;
  border: 15px solid red;
  box-sizing: border-box;
}
```

Si agregamos un border.



Si queremos que todas las cajas van a respetar el tamaño pondremos:

```
*{
  box-sizing: border-box;
}
```

Bordes radius (circular)

Border-radius: es la propiedad que nos permite redondear vértices de forma independiente.

Es un shorthand que engloba 4 propiedades:

- border-top-left-radius -> Radio del borde superior izquierdo.
- border-top-right-radius -> Radio del borde superior derecho.
- border-bottom-right-radius -> Radio del borde inferior derecho.
- border-bottom-left-radius -> Radio del borde inferior izquierdo.

Si solo recibe un valor dibujará un círculo en cada vértice del radio que especifiquemos.

border-radius: 50px; -> Círculo con 50px de radio.

Si utilizamos el border-radius como shorthands hay diferencias entre si queremos dar un valor a cada lado (círculo) o dos (elipses).

Con un solo valor:

border-radius: 50px; -> Círculo de 50px a cada vértice.

border-radius: 50px 100px; -> Círculo de 50px en el vértice superior izquierdo y el inferior derecho y de 100px en el superior derechos y el inferior izquierdo.

border-radius: 50px 100px 200px; -> Círculo de 50px en el vértice superior izquierdo, de 100px en el superior derecho y el inferior izquierdo y de 200px en el inferior derecho.

#YOAPRENDOCONDORIAN

border-radius: 50px 100px 200px 300px; -> Circulo de 50px en el vertice superior izquierdo, de 100px en el superior derecho, de 200px en el inferior derecho y de 300px en el inferior izquierdo.

Para este proyecto vamos a realizar la siguiente estructura:

```

  v border-radius
    v CSS
      # styles.css
    <> index.html

```

El código html

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Border radius</title>
</head>
<body>
  <div class="box"></div>
</body>
</html>

```

Código css.

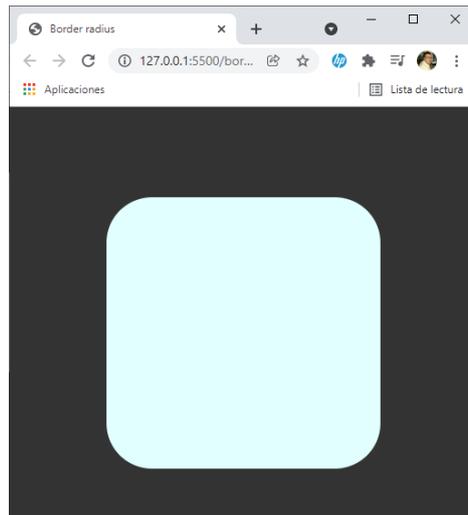
```

body{
  background-color: #333;
  color: #fff;
  margin: 0;
}
.box{
  margin-left: auto;
  margin-right: auto;

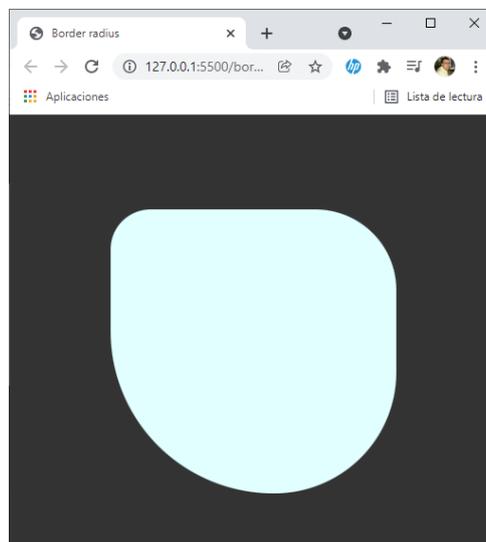
  margin-top: 100px;
  background-color: lightcyan;
  width: 300px;
  height: 300px;
  border-radius: 50px;
}

```

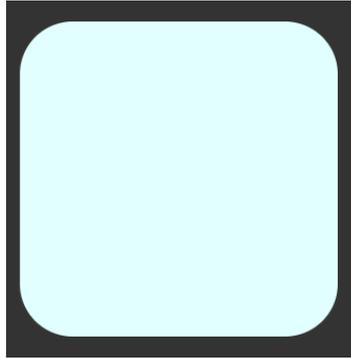
Este será el resultado:



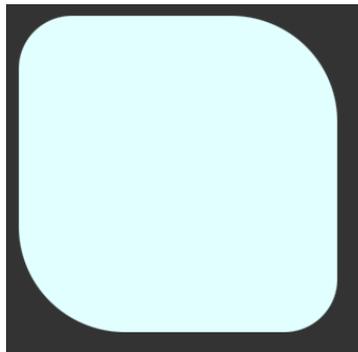
```
.box{  
  margin-left: auto;  
  margin-right: auto;  
  margin-top: 100px;  
  background-color: lightcyan;  
  width: 300px;  
  height: 300px;  
  border-top-left-radius: 50px;  
  border-top-right-radius: 100px;  
  border-bottom-right-radius: 150px;  
  border-bottom-left-radius: 200px;  
}
```



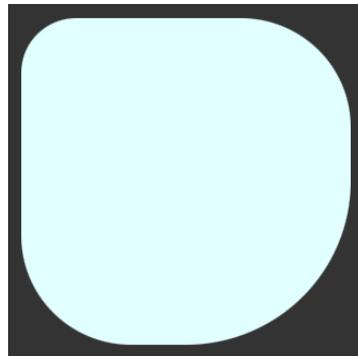
`border-radius: 50px;`



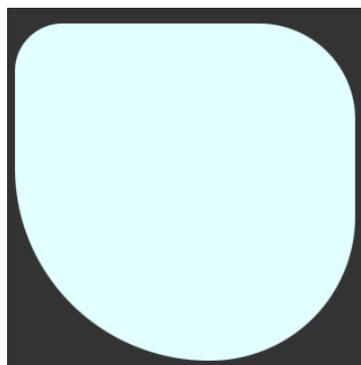
```
border-radius: 50px 100px;
```



```
border-radius: 50px 100px 150px;
```

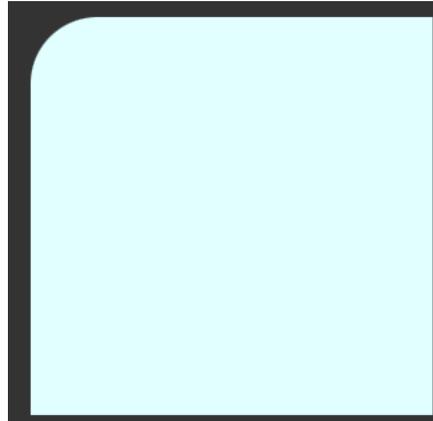


```
border-radius: 50px 100px 150px 200px;
```



Border radius (Elíptico)

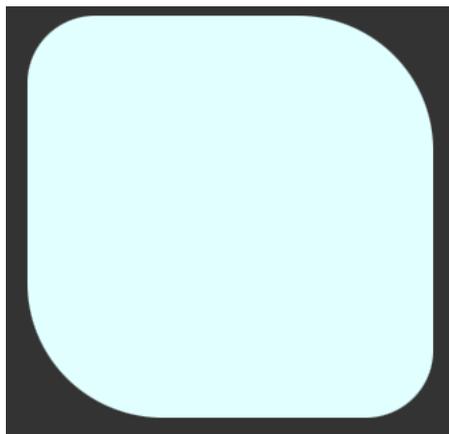
```
border-top-left-radius: 50px;
```



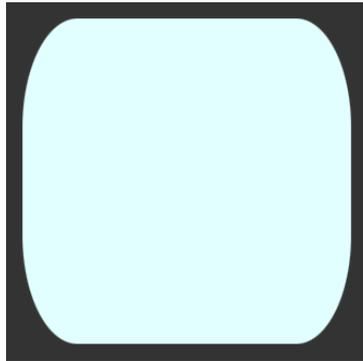
```
border-top-left-radius: 50px 100px;
```



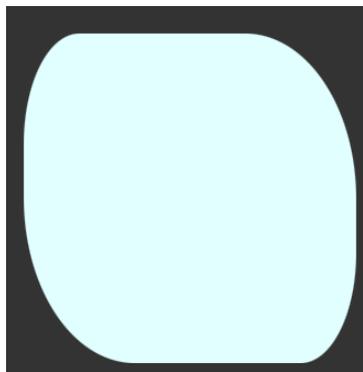
```
border-radius: 50px 100px;
```



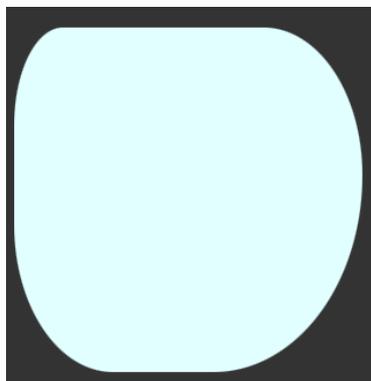
```
border-radius: 50px / 100px;
```



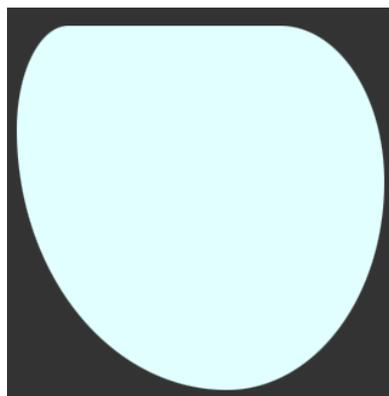
```
border-radius: 50px 100px / 100px 150px;
```



```
border-radius: 50px 100px 150px / 100px 150px 200px;
```



```
border-radius: 50px 100px 150px 200px / 100px 150px 200px 250px;
```



Overflow

Decimos que un contenido se desborda cuando la caja es más pequeña que el contenedor.

Overflow: Es la propiedad que controla que debe hacer la caja cuando su contenido se desborda del contenedor.

Es un shorthand que engloba overflow-x y overflow-y .

Tiene cuatro posibles valores:

- visible -> Es el valor por defecto.
- hidden -> El contenido que se desborda no se verá. Si aplicamos esta propiedad en un solo eje, el otro se pondrá automáticamente en el valor scroll.
- scroll -> Aparecerá barras de scroll en el eje asignado (x, y o ambos).
- auto -> Aparecerán barras de scroll en el caso que se hicieran falta.

Código html:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Border radius</title>
</head>
<body>
  <div class="box">
    <p class="text">
      Lorem ipsum, dolor sit amet consectetur adipisicing elit. Accusamus, sapiente.
      Minus corrupti debitis, magnam nisi similique laboriosam? Voluptate velit
      reprehenderit at sed ab nam, laboriosam eligendi similique inventore perferendis
      soluta!
      At iure, voluptatibus error, similique et ipsum nam, neque dolorem aliquid sunt
      libero rem repellendus cum voluptate odio voluptatum incidunt saepe rerum!
      Obcaecati consequuntur eos consectetur ex, porro neque magni.
      Reiciendis mollitia a exercitationem expedita et adipisci quo magnam, deleniti
      iste voluptas, voluptatibus iusto similique sunt quod minus minima excepturi
      numquam facere odio. Et deleniti quos distinctio dolorem voluptas quisquam.
      Inventore adipisci soluta aspernatur deleniti a quidem animi asperiores id nisi
      atque, veniam non suscipit illo amet porro? Officiis aut amet rerum eum ad
      corporis nostrum et nisi autem! Sint?
    </p>
  </div>
</body>
</html>
```

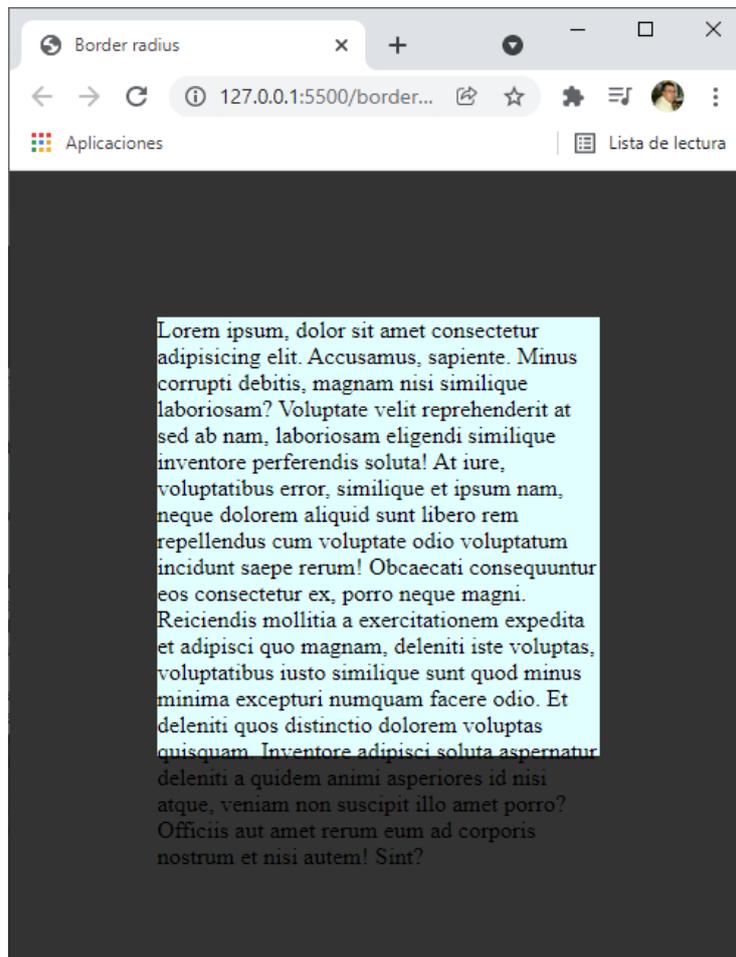
Código css:

```
body{
  background-color: #333;
  color: #000;
  margin: 0;
}
```

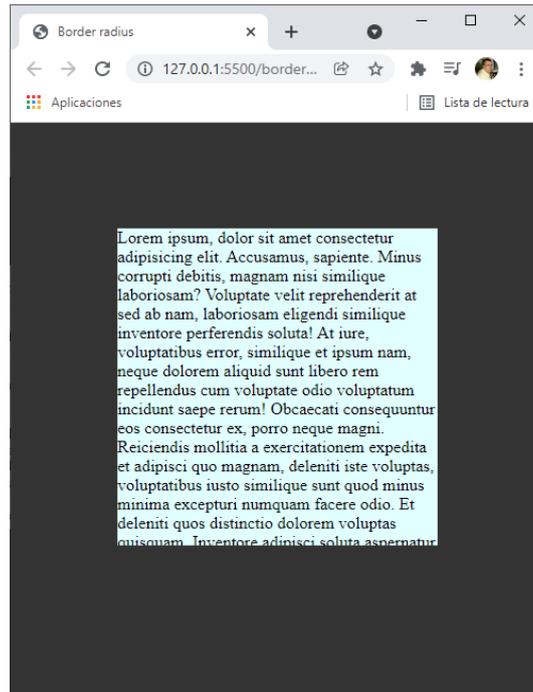
#YOAPRENDOCONDORIAN

```
.box{
  margin-left: auto;
  margin-right: auto;
  margin-top: 100px;
  background-color: lightcyan;
  width: 300px;
  height: 300px;
  color: black;
  overflow-y: hidden;
}
```

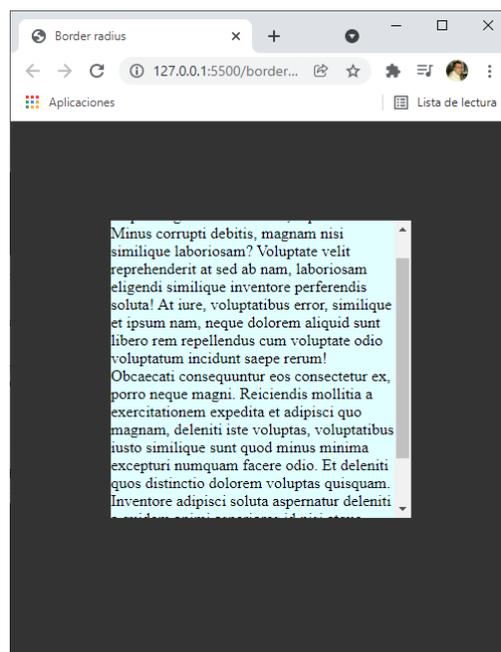
Este será el resultado:



```
.box{
  margin-left: auto;
  margin-right: auto;
  margin-top: 100px;
  background-color: lightcyan;
  width: 300px;
  height: 300px;
  color: black;
  overflow-y: hidden; ←
}
```



```
.box{  
  margin-left: auto;  
  margin-right: auto;  
  margin-top: 100px;  
  background-color: lightcyan;  
  width: 300px;  
  height: 300px;  
  color: #000;  
  overflow-y: scroll; ←  
}
```

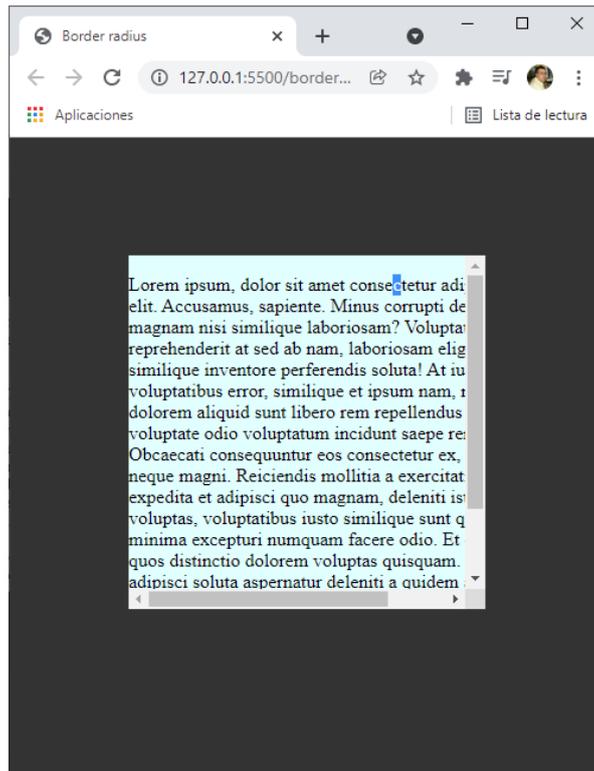


```
.text{
  margin-left: auto;
  margin-right: auto;
  margin-top: 100px;
  background-color: lightcyan;
  width: 300px;
  height: 300px;
  overflow-y: auto;
}
```

Muestra la barra scroll si fuere necesario.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Border radius</title>
</head>
<body>
  <div class="box">
    <p class="text">
      Lorem ipsum, dolor sit amet consectetur adipisicing elit. Accusamus, sapiente.
      Minus corrupti debitis, magnam nisi similique laboriosam? Voluptate velit
      reprehenderit at sed ab nam, laboriosam eligendi similique inventore perferendis
      soluta!
      At iure, voluptatibus error, similique et ipsum nam, neque dolorem aliquid sunt
      libero rem repellendus cum voluptate odio voluptatum incidunt saepe rerum!
      Obcaecati consequuntur eos consectetur ex, porro neque magni.
      Reiciendis mollitia a exercitationem expedita et adipisci quo magnam, deleniti
      iste voluptas, voluptatibus iusto similique sunt quod minus minima excepturi
      numquam facere odio. Et deleniti quos distinctio dolorem voluptas quisquam.
      Inventore adipisci soluta aspernatur deleniti a quidem animi asperiores id nisi
      atque, veniam non suscipit illo amet porro? Officiis aut amet rerum eum ad
      corporis nostrum et nisi autem! Sint?
    </p>
  </div>
</body>
</html>
```

```
body{
  background-color: #333;
  color: #000;
  margin: 0;
}
.box{
  margin-left: auto;
  margin-right: auto;
  margin-top: 100px;
  background-color: lightcyan;
  width: 300px;
  height: 300px;
  color: #000;
  overflow-y: scroll;
}
.text{
  width: 350px;
}
```



Vamos a realizar el siguiente ejercicio.

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Border radius</title>
</head>
<body>
  <div class="box">
    <div class="box-1">
      <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing elit. Sequi neque
        adipisci unde repellat! Reiciendis, accusamus consequuntur quaerat
        voluptatum architecto voluptatem, quisquam aliquid ratione ipsam provident
        quod ad totam ipsum adipisci.
      </p>
    </div>
    <div class="box-2">
      <p>
        Lorem ipsum, dolor sit amet consectetur adipisicing elit. Temporibus
        exercitationem veritatis in commodi voluptatum reprehenderit magnam
        officia recusandae necessitatibus quasi facere sed, asperiores quod
        corporis nihil illum quaerat omnis! Ducimus?
      </p>
    </div>
  </div>
</body>
</html>

```

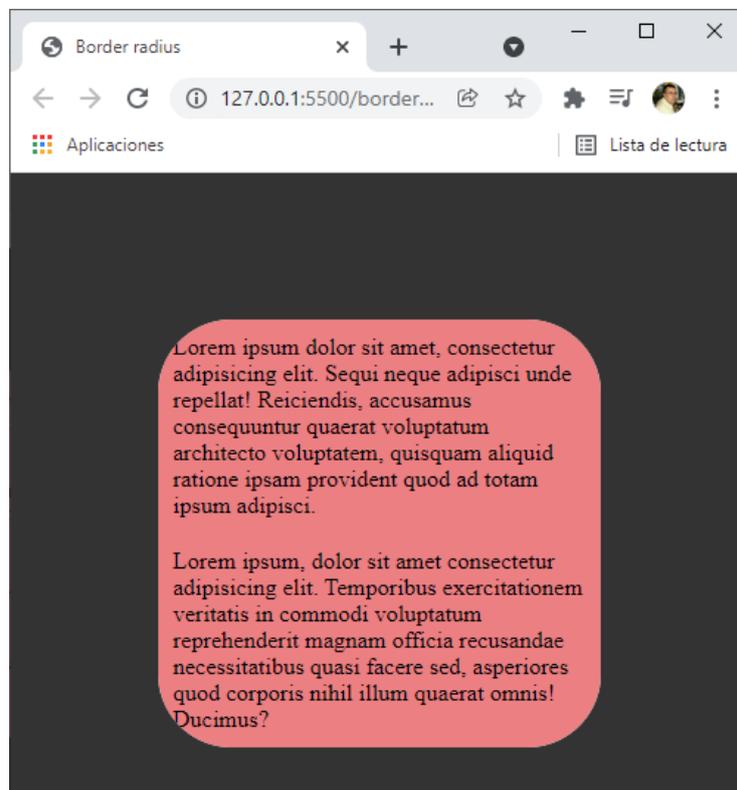
#YOAPRENDOCONDORIAN

Código css

```
body{
  background-color: #333;
  color: #000;
  margin: 0;
}
.box{
  margin-left: auto;
  margin-right: auto;
  margin-top: 100px;
  background-color: lightcyan;
  width: 300px;
  height: 300px;
  color: #000;
  border-radius: 50px;
  overflow: hidden;
}

.box-1, .box-2{
  padding: 10px;
  background-color: lightcoral;
}
.text{
  margin: 0;
}
```

Este será el resultado:



Colapsado de márgenes

Código html:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Colapso de márgenes</title>
</head>
<body>
  <header class="header"></header>
  <div class="box-1"></div>
  <div class="box-2"></div>
</body>
</html>
```

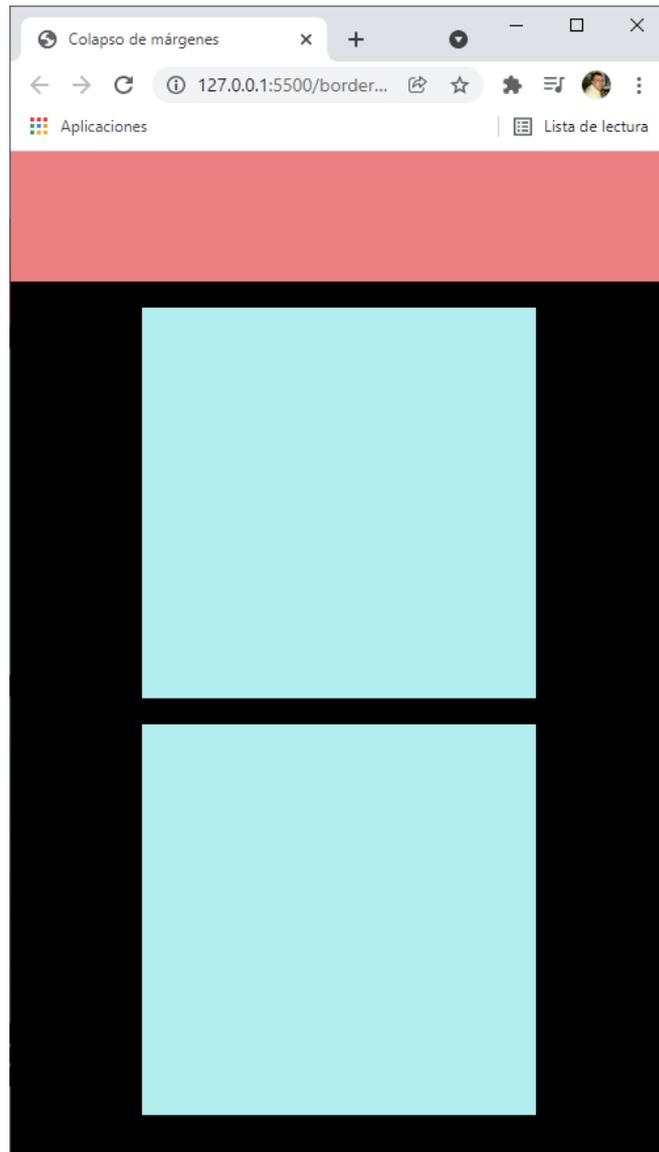
Código css:

```
body{
  background-color: ■ black;
  margin: 0;
}

.header{
  background-color: ■ lightcoral;
  height: 100px;
  margin-bottom: 20px;
}

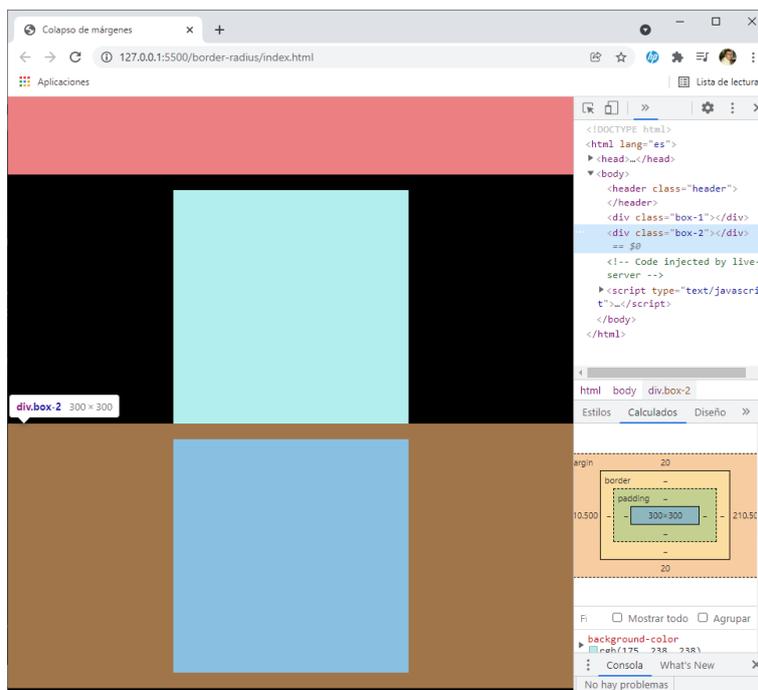
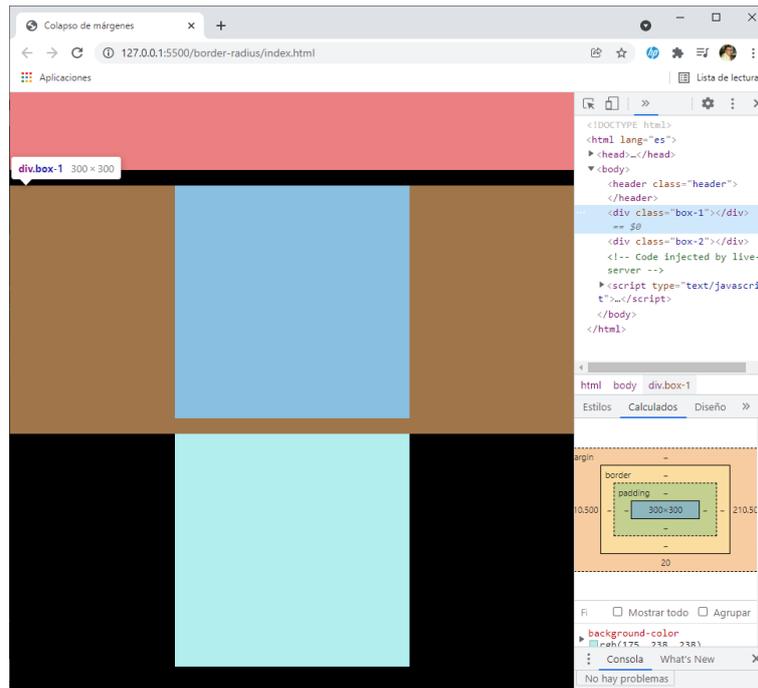
.box-1,
.box-2{
  width: 300px;
  height: 300px;
  background-color: ■ paleturquoise;
  margin-bottom: 20px;
  margin-left: auto;
  margin-right: auto;
}
```

Este será el resultado:



```
.box-2{  
  margin-top: 20px;  
}
```

Por lógica las dos cajas azules tendrían que estar más separadas, pero no es así.



Podemos observar que los márgenes verticales se colapsan. No jugar con margin-bottom y margin-top.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Colapso de márgenes</title>
```

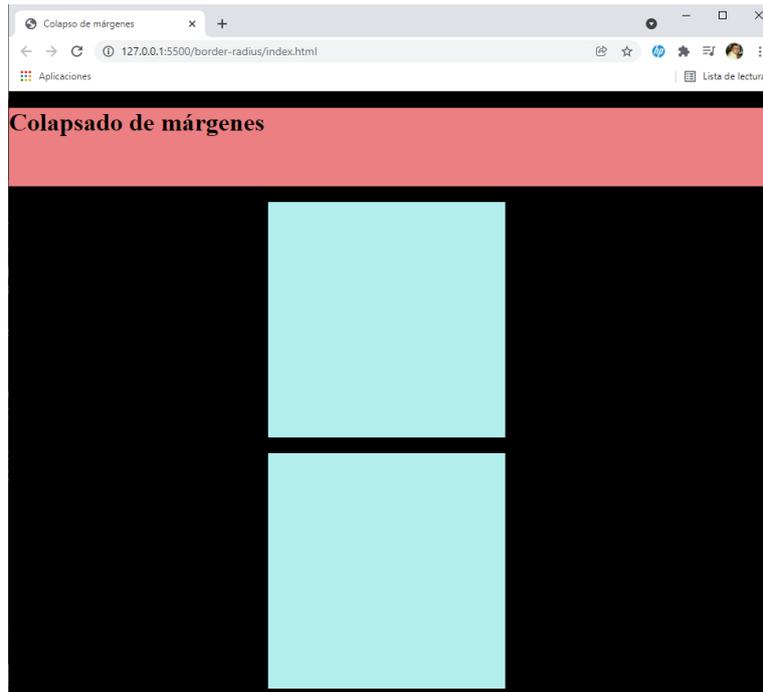
#YOAPRENDOCONDORIAN

```

</head>
<body>
  <header class="header">
    <h1 class="title">Colapsado de márgenes</h1>
  </header>
  <div class="box-1"></div>
  <div class="box-2"></div>
</body>
</html>

```

Que observamos:



Observamos en la parte superior un margen no nosotros no hemos decidido, pero se genera al insertar la etiqueta h1.

```

.title{
  margin: 0;
}

```

Si agregamos este estilo:



Ya hemos solucionado el margen superior.

#YOAPRENDOCONDORIAN

```
.title{
  margin: 0;
  margin-top: 50px;
}
```

Si queremos bajar las letras.



Pero lo que baja es el contenedor header.

```
.header{
  background-color: lightcoral;
  height: 100px;
  margin-bottom: 20px;
  overflow: hidden;
}
```

Agregamos la última línea.



Ya hemos conseguido bajar el texto y hemos evitado el colapsado entre padres e hijos.

```
.header{
  background-color: lightcoral;
  height: 100px;
  margin-bottom: 20px;
  padding-top: 0.1px;
}
```

También se puede solucionar de esta forma.

```
.header{
  background-color: lightcoral;
  height: 100px;
  margin-bottom: 20px;
  border-top: 0.1px solid lightcoral;
}
```

Display

Código html:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Colapso de márgenes</title>
</head>
<body>
  <p class="text">Lorem ipsum dolor sit amet</p>
  <p class="text">Lorem ipsum dolor sit amet consectetur, adipisicing elit.</p>
  <a href="#" class="link">Enlace</a>
  <a href="#" class="link">Enlace</a>
</body>
</html>
```

Código css:

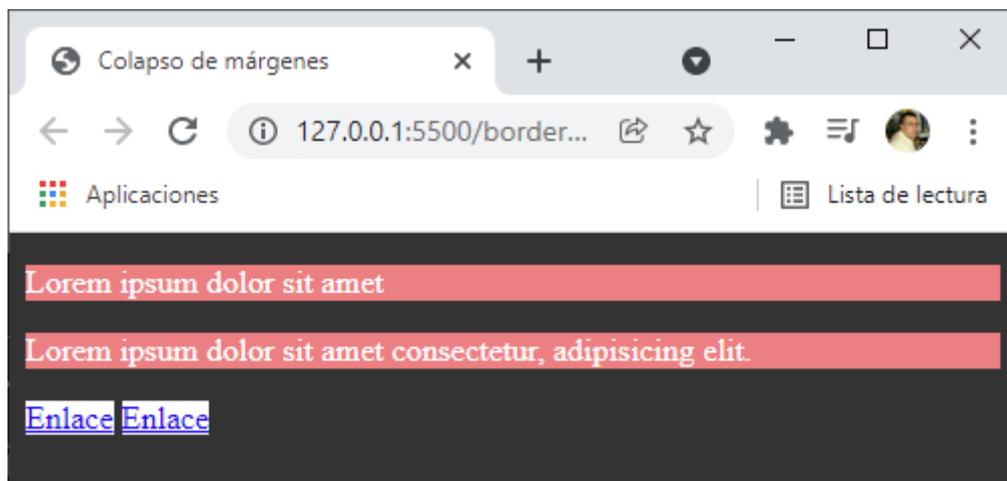
```
*{
  box-sizing: border-box;
}

body{
  background-color: #333;
  color: #fff;
}

.text{
  background-color: lightcoral;
}

.link{
  background-color: #fff;
}
```

Este será el resultado:



#YOAPRENDOCONDORIAN

La propiedad `display` nos sirve para cambiar el contexto de los elementos dentro del navegador.

Admite varios valores:

`none` -> Hace que el elemento no se muestre, pero sigue cargándose.

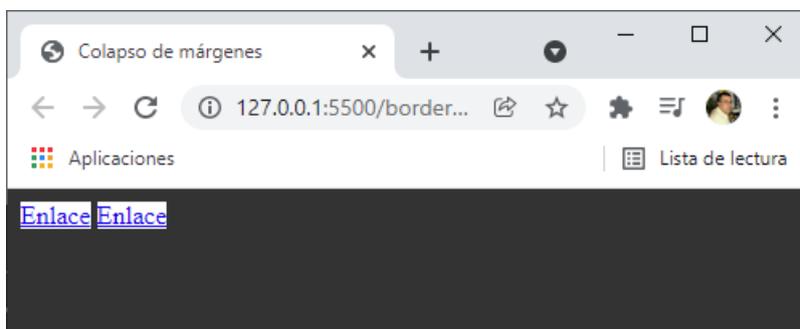
`block` -> Hace que el elemento sea de bloque.

`inline` -> Hace que el elemento sea de línea.

`inline-block` -> Hace que el elemento sea de línea pero admite medidas y márgenes verticales.

```
.text{
  background-color: lightcoral;
  display: none;
}
```

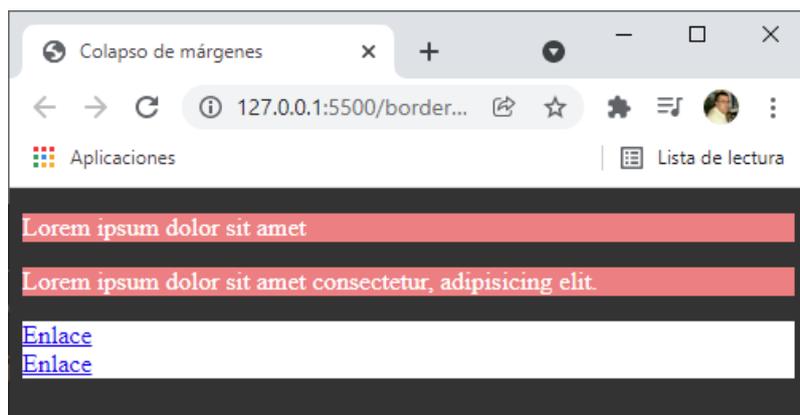
Si agregamos el `display: none`;



No se muestran.

```
.link{
  background-color: #fff;
  display: block;
}
```

Si al enlace le hacemos `display: block`;

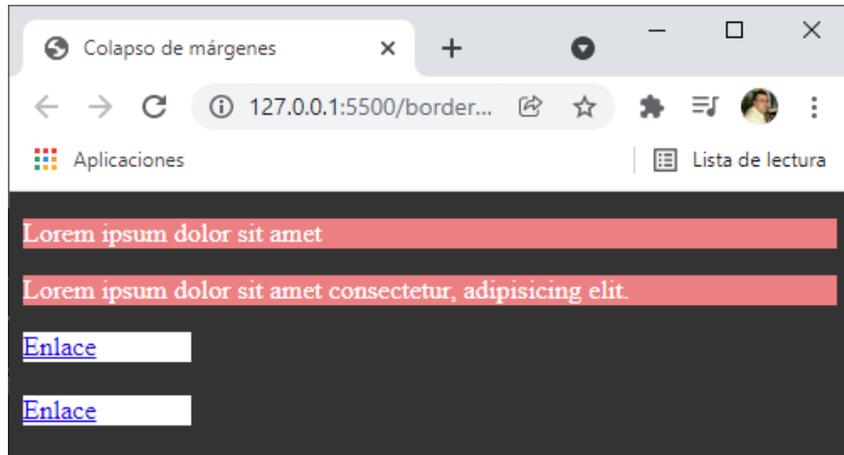


Elementos que son de línea se comportan como elementos de bloque.

#YOAPRENDOCONDORIAN

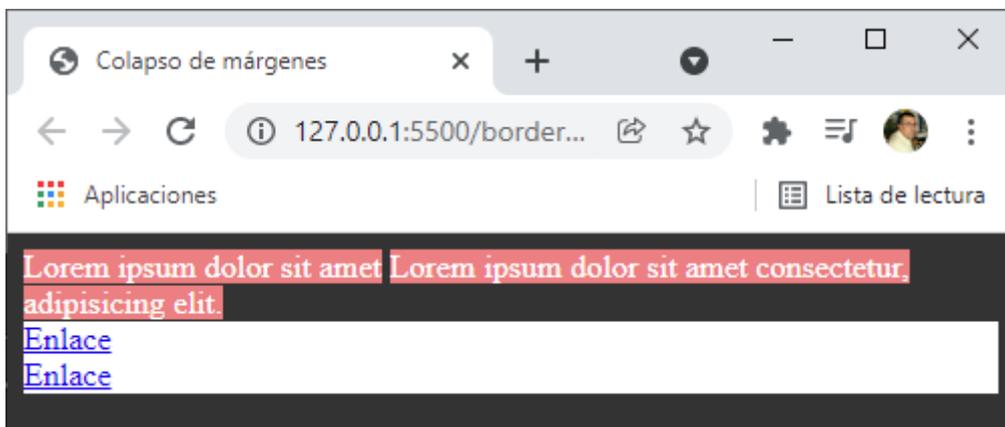
```
.link{
  background-color: #fff;
  display: block;
  width: 100px;
  margin-bottom: 20px;
}
```

Ahora le podemos agregar las propiedades width, margin-bottom, etc.



```
.text{
  background-color: lightcoral;
  display: inline;
}
```

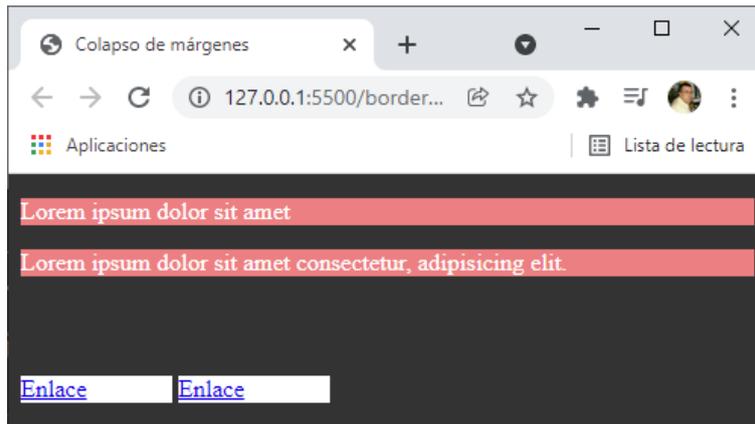
Queremos elementos de bloque que se comporte como elementos de línea.



Ahora las propiedades width y margin-bottom no las reconoce.

```
.link{
  background-color: #fff;
  display: inline-block;
  width: 150px;
  margin-top: 50px;
}
```

Si tenemos un elemento de línea y queremos que siga siendo de línea pero que admita las propiedades de bloque.



Outline

Outline es la propiedad que nos permite dibujar un borde por fuera del modelo de caja.

Es un shorthand que engloba:

- `outline-width`: Controla en ancho del outline.
- `outline-style`: Controla el estilo del outline.
- `Outline-color`: Controla el color del outline.

Tiene las mismas propiedades y sintaxis que `border`, pero con algunas diferencias.

No ocupa sitio, ya que no forma parte del `box-model`.

No se puede redondear.

No se pueden controlar los lados independientemente.

También cuenta con la propiedad `outline-offset`, que nos permite aumentar o disminuir la distancia del outline respecto a la caja a la que pertenece.

Código html:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Outline</title>
</head>
<body>
  <div class="border">Border</div>
  <div class="outline">Outline</div>
</body>
</html>
```

#YOAPRENDOCONDORIAN

Código css:

```
*{
  box-sizing: border-box;
}

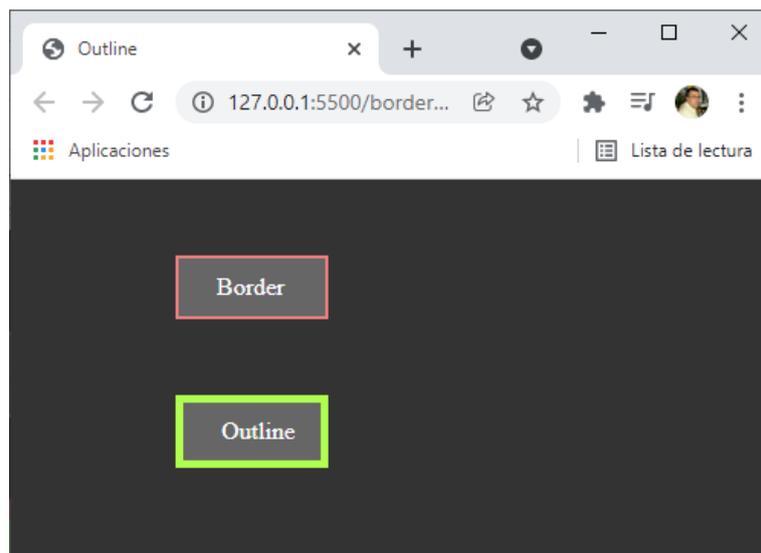
body{
  background-color: #333;
  color: #fff;
}

.border,
.outline{
  background-color: #666;
  width: 100px;
  padding: 10px 25px;
  margin: 50px 100px;
}

.border{
  border: 2px solid lightcoral;
}

.outline{
  border: 10px solid lightcoral;
  border: 5px solid greenyellow
}
```

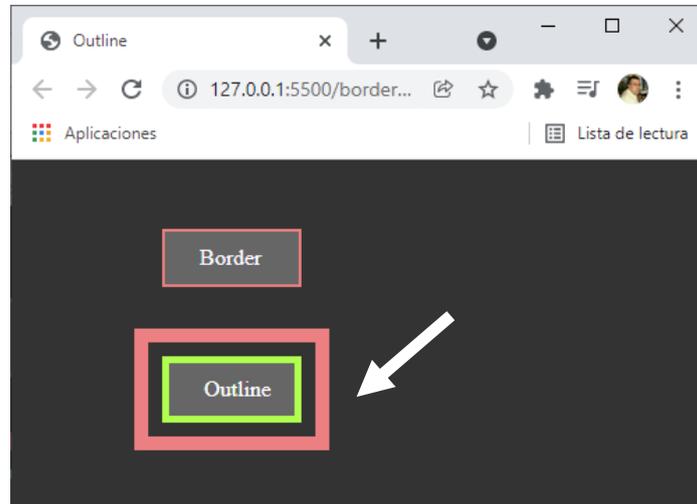
Este es el resultado:



#YOAPRENDOCONDORIAN

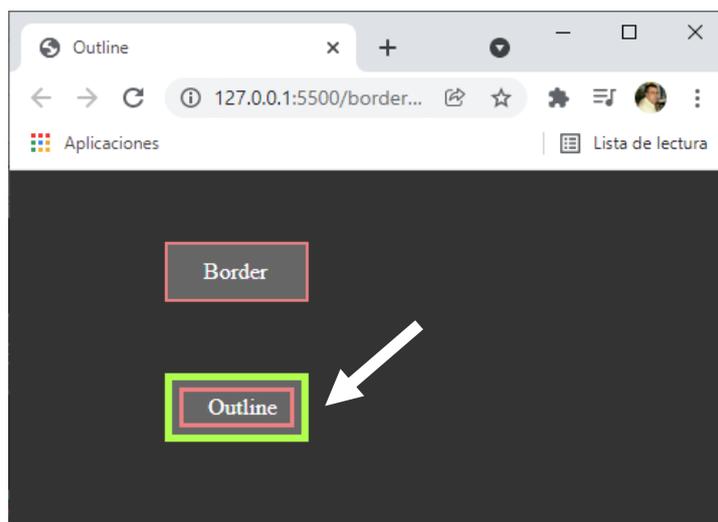
```
.outline{
  outline: 10px solid lightcoral;
  outline-offset: 10px;
  border: 5px solid greenyellow
}
```

Este será el resultado:



```
.outline{
  outline: 3px solid lightcoral;
  outline-offset: -13px;
  border: 5px solid greenyellow
}
```

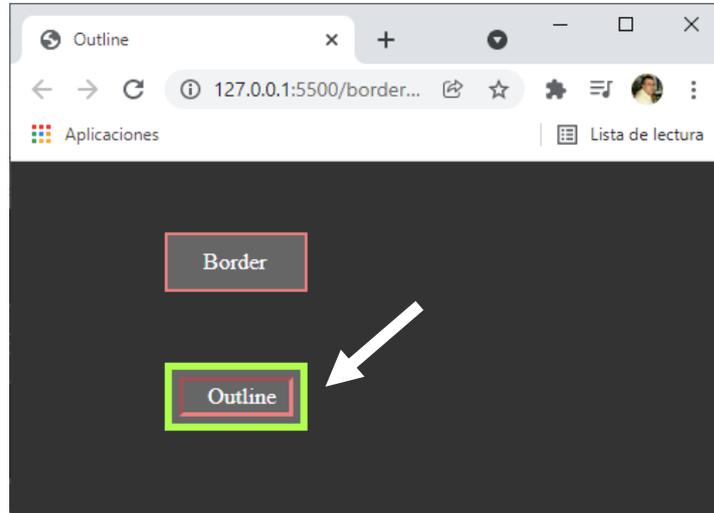
Si outline-offset lo ponemos en negativo:



Que dentro de la caja.

#YOAPRENDOCONDORIAN

```
.outline{
  outline: 3px inset lightcoral;
  outline-offset: -13px;
  border: 5px solid greenyellow
}
```



Text align

Text-align es la propiedad que nos permite alinear horizontalmente el contenido de un elemento de bloque siempre que el contenido no tenga ancho declarado.

Acepta 4 posibles valores:

- text-align: left; -> Alinea el contenido a la izquierda
- text-align: right; -> Alinea el contenido a la derecha
- text-align: center -> Alinea el contenido al centro
- text-align: justify -> Alinea el contenido de forma justificada. No es recomendable usarlo.

Código html:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Outline</title>
</head>
<body>
  <p class="text">Lorem ipsum dolor sit amet, consectetur adipisicing elit. Optio, accusantium.
  Tempore quidem dolor sit porro voluptatibus perferendis quam est nulla reprehenderit, ea officia
  natus quis nesciunt libero ex delectus commodi!
  Laboriosam officiis illo maxime quis tenetur distinctio, tempora nobis culpa molestiae, libero
  blanditiis corporis et corrupti ratione repellat labore exercitationem aut qui, quaerat incidunt
  perferendis. Molestias optio autem sit omnis.
  Voluptates repellat saepe, beatuae facilis maxime nemo aut debitis accusantium pariatur
  voluptatem, eaque laudantium! Quo nemo beatuae illo vel a asperiores veniam maxime! Laudantium
  dolor, et totam temporibus eveniet expedita.
```

#YOAPRENDOCONDORIAN

```

Excepturi ab adipisci dolore repudiandae vitae, ullam enim assumenda facilis dolores iure
corporis, pariatur ducimus, at non consectetur. Ipsa earum magni iste reprehenderit? Quos
eveniet accusantium autem tenetur optio necessitatibus.
Adipisci, aperiam. Laudantium ullam illum sapiente, consectetur aliquid asperiores accusantium
expedita consequatur libero excepturi enim modi quam eligendi veniam provident necessitatibus
eius iure. Ut consectetur reiciendis assumenda necessitatibus soluta at?
</p>
<div class="box">Border</div>
<a href="#" class="link">Soy un enlace</a>

</body>
</html>

```

Código css:

```

*{
|   box-sizing: border-box;
}

body{
|   background-color: ■ #333;
|   color: □ #fff;
}

.box{
|   background-color: ■ #666;
|   width: 100px;
|   padding: 10px 10px;
|   margin: 10px 10px;
|   border: 1px solid ■ red;
|   margin-left: auto;
|   margin-right: auto;
|   text-align: center;
}

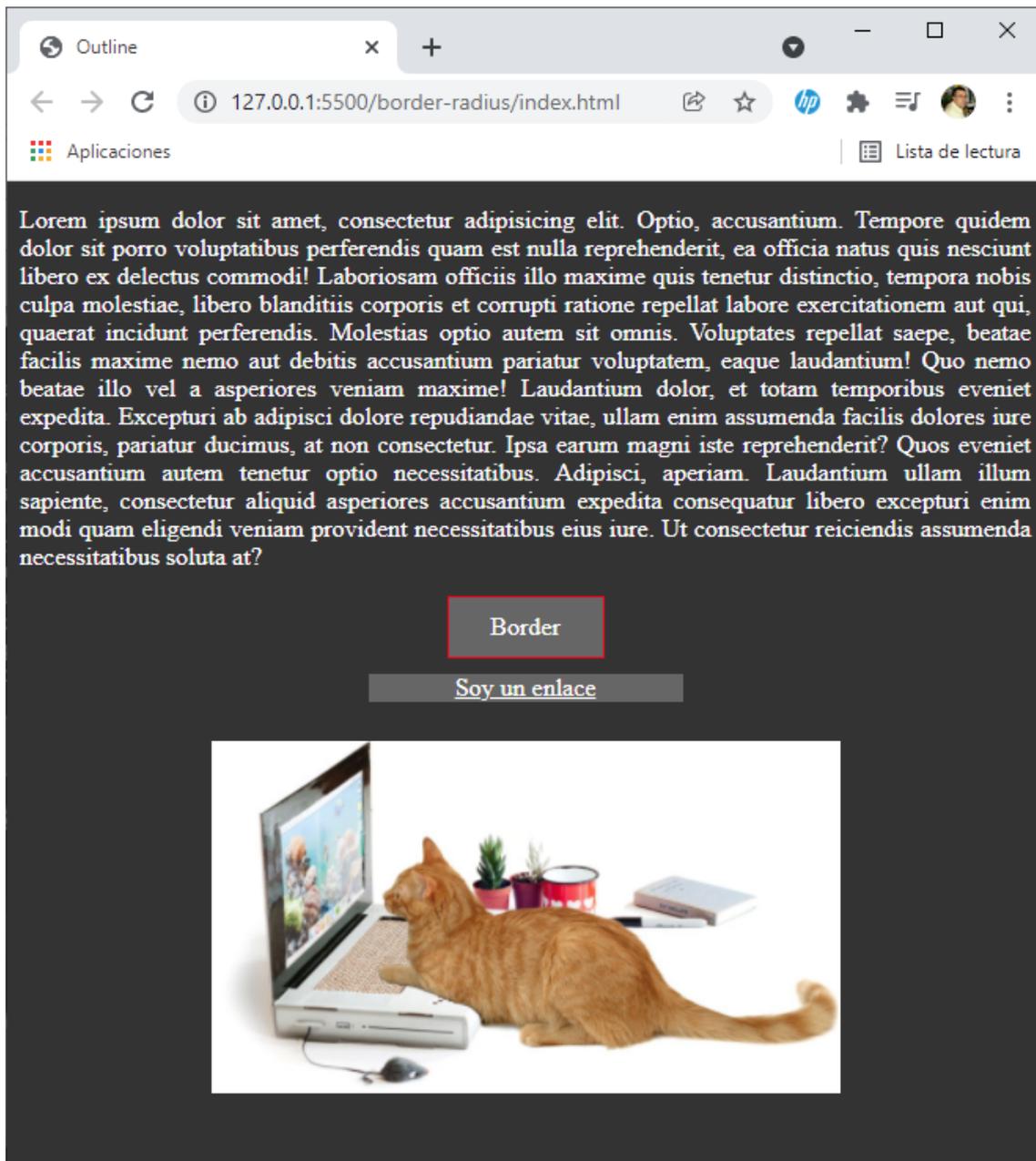
.img{
|   display: block;
|   width: 400px;
|   margin-left: auto;
|   margin-right: auto;
}

.text{
|   text-align: justify;
}

.link{
|   width: 200px;
|   margin-left: auto;
|   margin-right: auto;
|   display: block;
|   color: □ white;
|   background-color: ■ #666;
|   text-align: center;
|   margin-bottom: 25px;
}

```

Este será el resultado:



Otra forma de centrar la imagen.

```
<div class="img-container">
  
</div>
```

Poner la imagen en un contenedor.

```
.img-container{
  text-align: center;
}
```

Ahora le podemos decir que centre lo del contenedor, pero para que funcione tenemos que dar la siguiente propiedad a la imagen.

```
.img{
  display: inline;
  width: 400px;
}
```

Box shadow

La propiedad box-shadow se creó para añadir efectos de sombra a nuestra caja.

En esencia lo que hace es crear un clon de la caja respetando la forma de su box-model (ancho, alto, redondez).

La sintaxis de box-shadow se puede escribir de distintas formas según lo que queramos conseguir.

Los valores que le podemos poner son:

- offset-x -> Desplazamiento en x (obligatorio).
- offset-y -> Desplazamiento en y (obligatorio).
- blur-radius -> Desenfoque de la sombra.
- spread-radius -> Expansión de la sombra.
- color -> El color de la sombra, si no lo especificamos lol heredará del elemento al que pertenece.
- inset -> Determina si la sombra será interior o exterior.

Código html:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Outline</title>
</head>
<body>
  <div class="caja-contenedor">
    <div class="caja">
      <h1 class="text">BOX</h1>
    </div>
  </div>
</body>
</html>
```

#YOAPRENDOCONDORIAN

Código css:

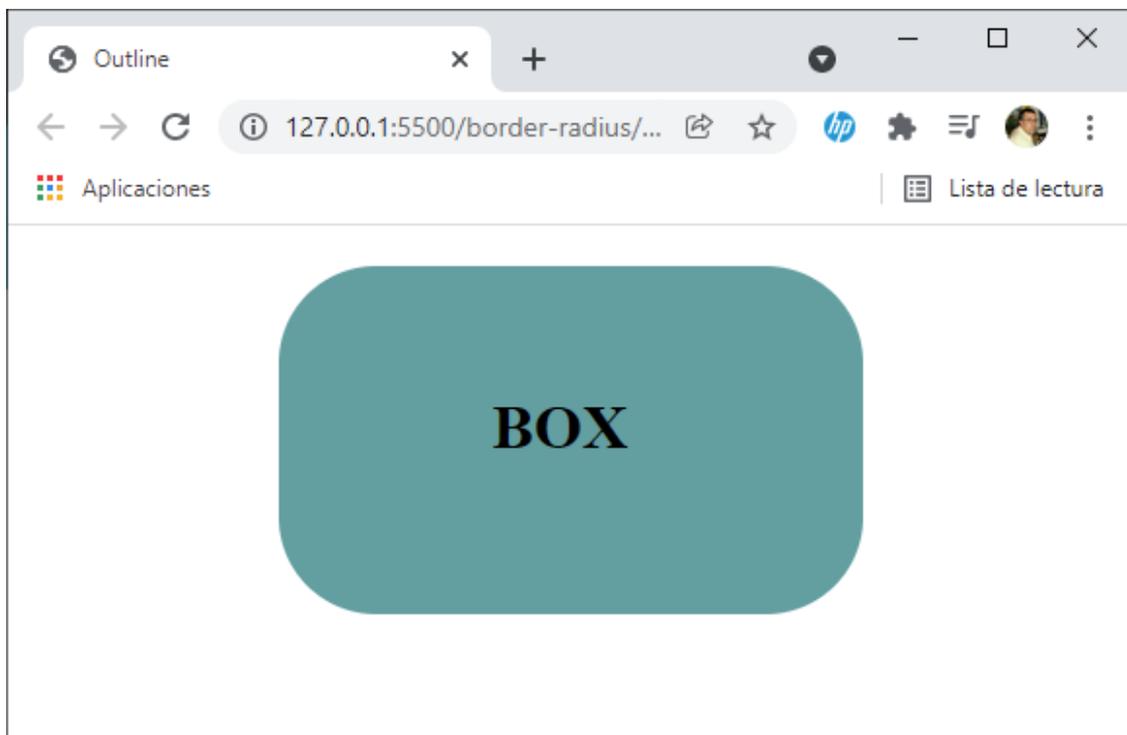
```
*{
  box-sizing: border-box;
}

.caja-contenedor{
  display: block;
}

.caja{
  width: 300px;
  height: 180px;
  border-radius: 50px;
  background-color: #4682B4;
  margin-left: auto;
  margin-right: auto;
}

.text{
  padding: 65px 0 0 110px;
}
```

Este será el resultado:

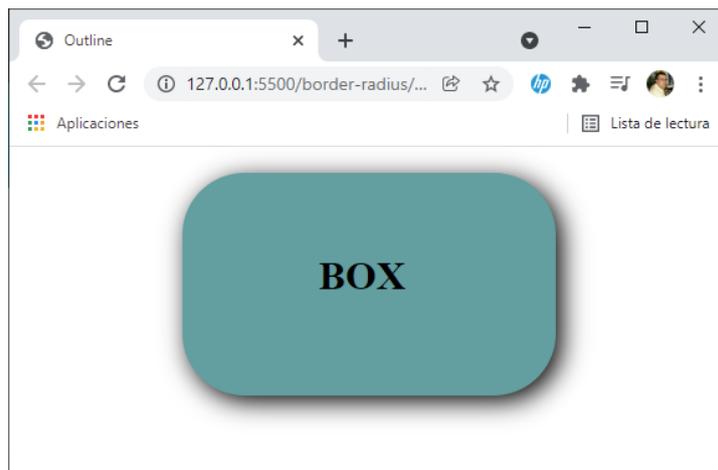


#YOAPRENDOCONDORIAN

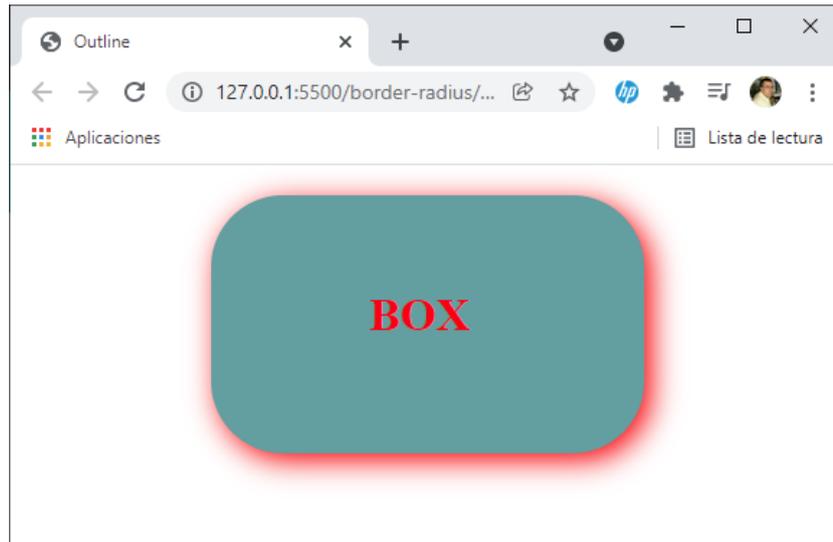
```
.caja{  
  width: 300px;  
  height: 180px;  
  border-radius: 50px;  
  background-color: cadetblue;  
  margin-left: auto;  
  margin-right: auto;  
  box-shadow: 5px 5px; ←  
}
```



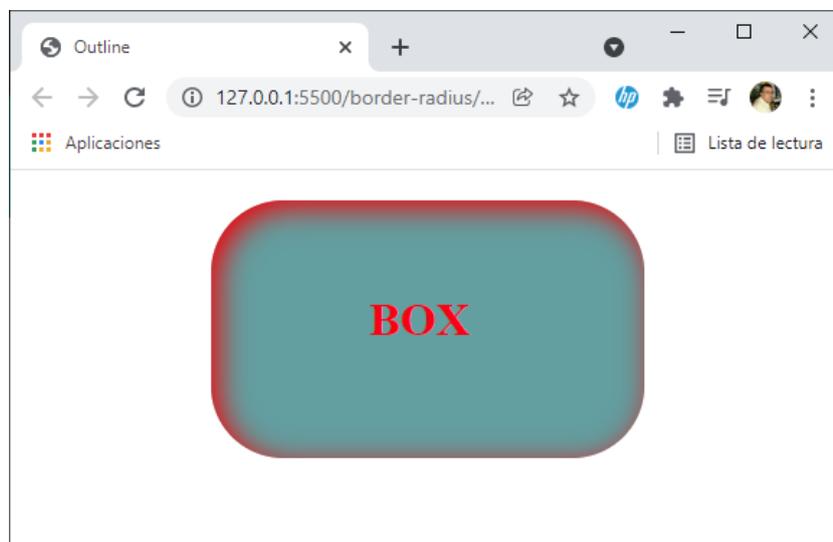
```
.caja{  
  width: 300px;  
  height: 180px;  
  border-radius: 50px;  
  background-color: cadetblue;  
  margin-left: auto;  
  margin-right: auto;  
  box-shadow: 5px 5px 20px; ←  
}
```



```
.caja{  
  width: 300px;  
  height: 180px;  
  border-radius: 50px;  
  background-color: cadetblue;  
  margin-left: auto;  
  margin-right: auto;  
  color: red; ←  
  box-shadow: 5px 5px 20px ;  
}
```

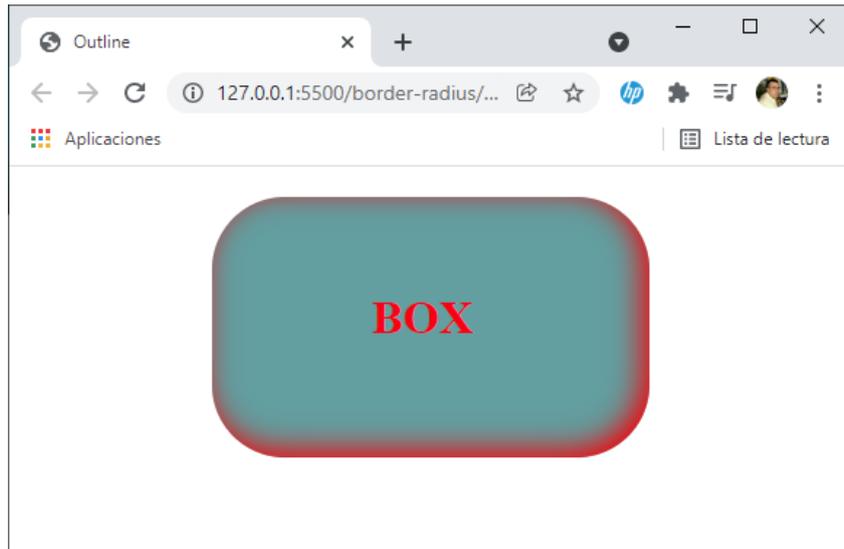


```
.caja{  
  width: 300px;  
  height: 180px;  
  border-radius: 50px;  
  background-color: cadetblue;  
  margin-left: auto;  
  margin-right: auto;  
  color: red;  
  box-shadow: inset 5px 5px 20px ;  
}
```

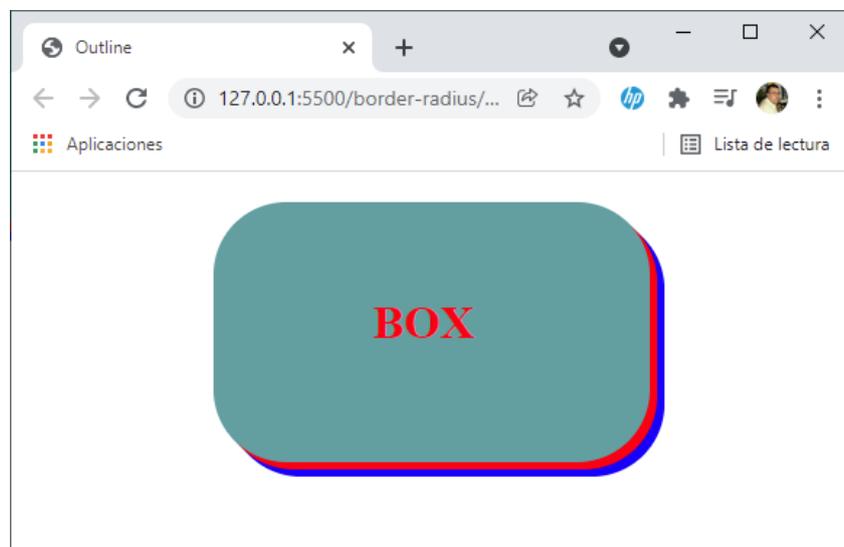


#YOAPRENDOCONDORIAN

```
.caja{
  width: 300px;
  height: 180px;
  border-radius: 50px;
  background-color: cadetblue;
  margin-left: auto;
  margin-right: auto;
  color: red;
  box-shadow: inset -5px -5px 20px;
}
```



```
.caja{
  width: 300px;
  height: 180px;
  border-radius: 50px;
  background-color: cadetblue;
  margin-left: auto;
  margin-right: auto;
  color: red;
  box-shadow: 5px 5px red, 10px 10px blue;
}
```



¿Qué es POSITION?

La propiedad position nos permite posicionar los elementos. Hay algunos conceptos que debéis conocer para entender position.

Flujo de renderizado -> Por norma general los elementos se dibujan de izquierda a derecha y de arriba abajo. El punto 0,0 de los elementos, por norma general, es la esquina superior izquierda.

Espacio reservado -> Es el espacio que tiene un elemento asignado en el navegador.

Elemento posicionado -> Esto significa que el elemento tiene la propiedad position con un valor distinto de "static", que es el valor que tiene esta propiedad por defecto.

Stackin context -> Contexto de apilamiento. Es el orden en el que se apilan las cajas que se superponen, dentro del mismo contenedor.

Al posicionar un elemento se habilitan 5 propiedades que podemos utilizar para mover los elementos en los 3 ejes.

top -> El elemento se moverá desde la parte superior la distancia que le hayamos indicado.

right -> El elemento se moverá desde la parte derecha la distancia que le hayamos indicado.

bottom -> El elemento se moverá desde la parte inferior la distancia que le hayamos indicado.

left -> El elemento se moverá desde la parte izquierda la distancia que le hayamos indicado.

z-index -> Nos permite mover el elemento en el contexto de apilamiento (eje z)

NOTA: Si a un elemento le declaramos la propiedad top y/o left, las propiedades bottom y/o right no funcionarán.

Los posibles valores que le podemos dar a position son:

- Static -> Es el valor que tiene por defecto un elemento, con este valor el elemento NO ESTA POSICIONADO y por lo cual no podremos moverlo.
- Relative -> El elemento mantendrá su posición y medidas en el flujo de renderizado y mantendrá su espacio reservado. Si lo movemos lo hará usando su posición en el html como punto de referencia.
- Absolate -> El elemento perderá sus medidas y su espacio reservado. Si lo movemos usará el elemento padre posicionando como referencia. Si no tiene ninguno, usará el elemento html de referencia.
- Fixed -> El elemento perderá sus medidas y su espacio reservado. Si lo movemos usará el elemento html de referencia, y además se quedará fijo en esa posición aunque hagamos scroll.
- Stichy -> Es una mezcla de position relative y fixed. Con este tipo de posicionamiento los valores top, left, bottom y right no sirven para mover el elemento, si no para indicarle en que punto pasará a tener un comportamiento de posicionamiento fixed, hasta llegar a ese punto se comportará como si tuviera relative.

¿Qué es POSITION RELATIVE?

Código html:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Outline</title>
</head>
<body>
  <div class="box box-1"></div>
  <div class="box box-2"></div>
</body>
</html>
```

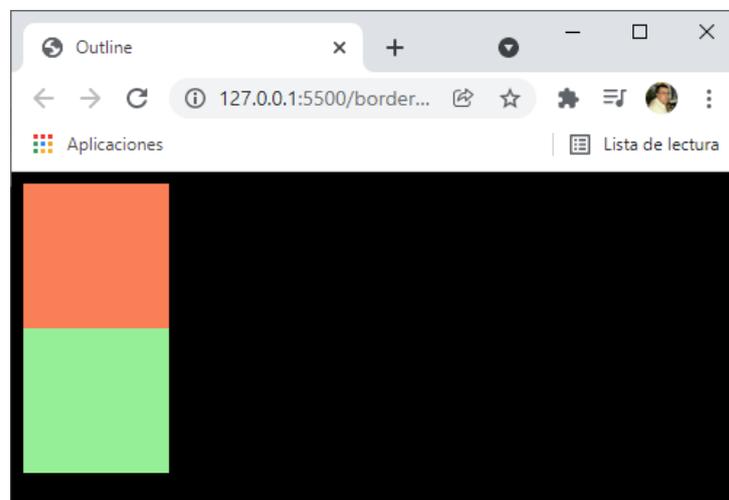
Código css:

```
body{
  background-color: black;
}
.box{
  width: 100px;
  height: 100px;
}

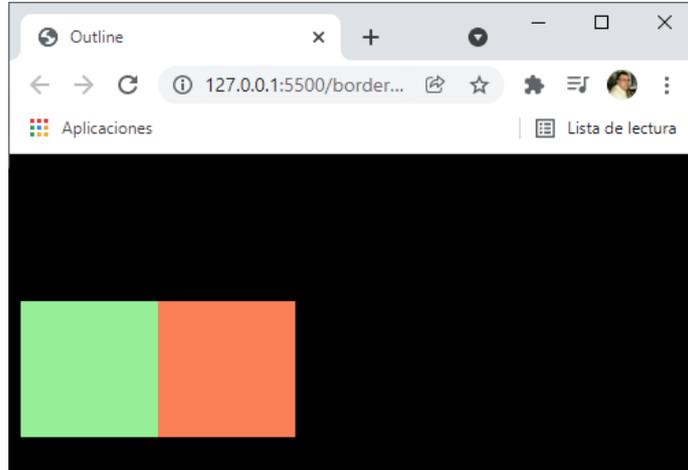
.box-1{
  background-color: coral;
  position: relative;
}

.box-2{
  background-color: lightgreen;
}
```

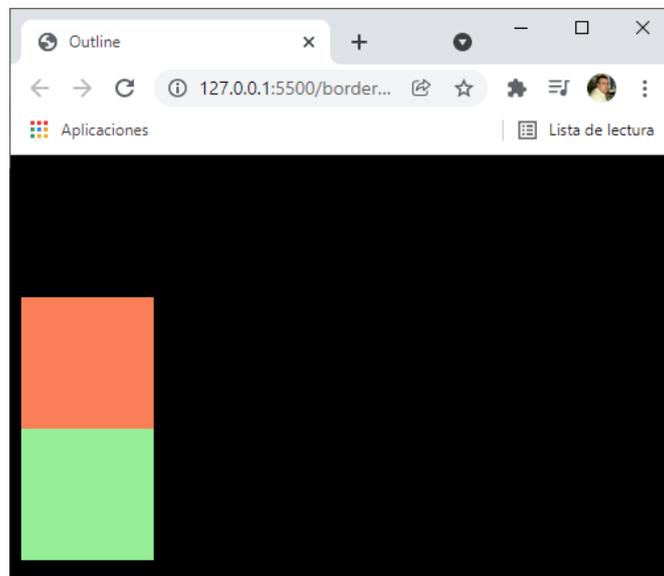
Este será el resultado:



```
.box-1{  
  background-color: coral;  
  position: relative;  
  left: 100px;  
  top: 100px;  
}
```



```
.box{  
  width: 100px;  
  height: 100px;  
  position: relative;  
  top: 100px;  
}
```



¿Qué es POSITION ABSOLUTE?

Código HTML:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Outline</title>
</head>
<body>
  <div class="box box-1">Box 1</div>
  <div class="box box-2">Box 2</div>
</body>
</html>
```

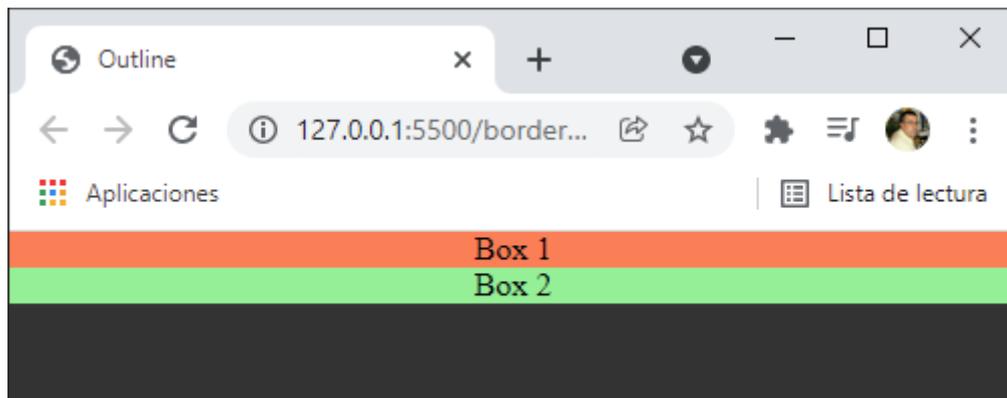
Código CSS:

```
*{
  box-sizing: border-box;
}
body{
  margin: 0;
  background-color: #333;
}
.box{
  text-align: center;
}

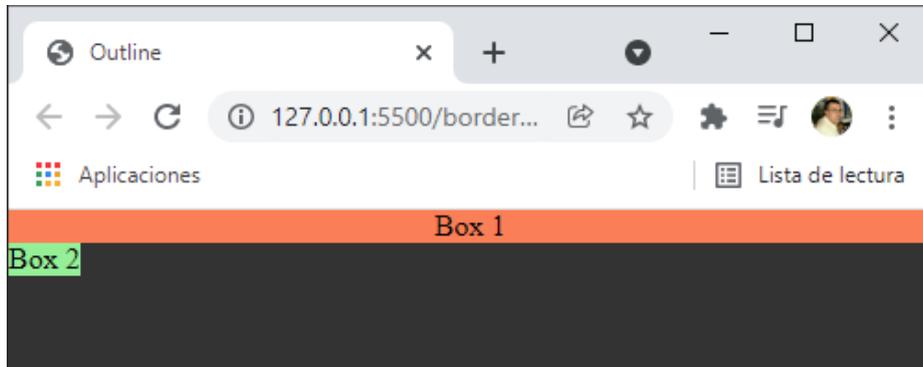
.box-1{
  background-color: coral;
}

.box-2{
  background-color: lightgreen;
}
```

Resultado:

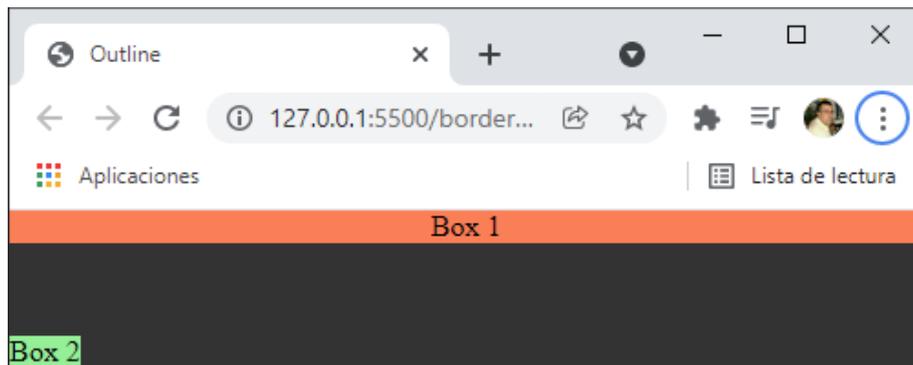


```
.box-2{  
  background-color: lightgreen;  
  position: absolute;  
}
```

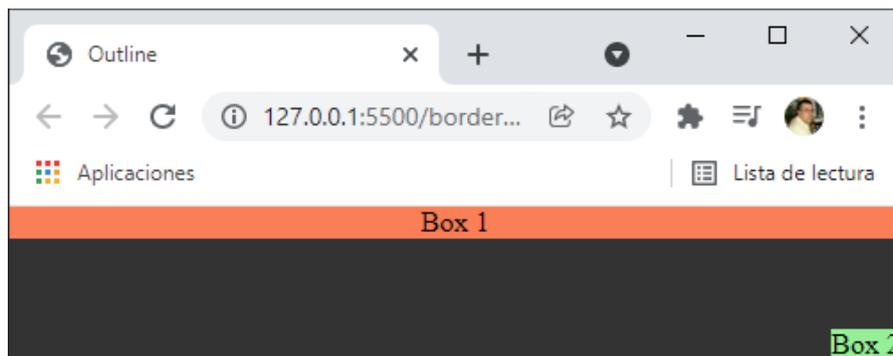


Pierde sus medidas.

```
.box-2{  
  background-color: lightgreen;  
  position: absolute;  
  bottom: 0;  
}
```



```
.box-2{  
  background-color: lightgreen;  
  position: absolute;  
  bottom: 0;  
  right: 0;  
}
```



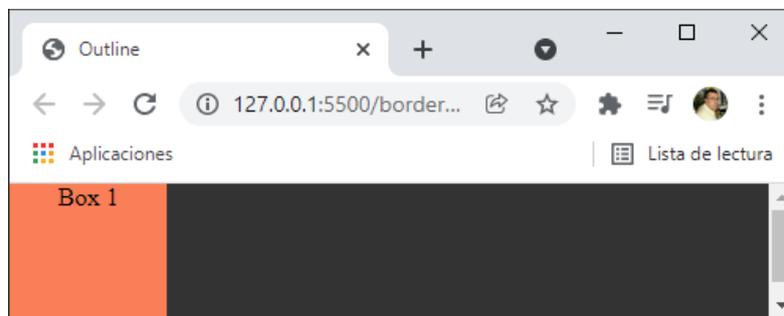
#YOAPRENDOCONDORIAN

Si el objeto no está en ningún contenedor se desplazará por todo el body.

```
.box{
  width: 100px;
  height: 100px;
  text-align: center;
}

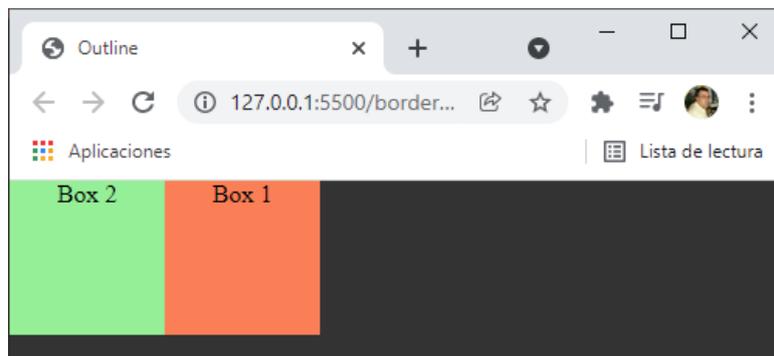
.box-1{
  background-color: coral;
  position: absolute;
}

.box-2{
  background-color: lightgreen;
}
```



El Box 2 no ha desaparecido se encuentra detrás del Box 1.

```
.box-1{
  background-color: coral;
  position: absolute;
  left: 100px;
}
```



Todo lo que debes saber sobre Position Fixed

Código html:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Position Fixed</title>
</head>
<body>
  <header class="header">
    <nav class="menu">MENU</nav>
  </header>
  <h1>Position Fixed</h1>
  <section>
    <h2>Sección 1</h2>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Minus q
      Eligendi, modi nesciunt. Ullam, praesentium saepe nisi vitae, nu
      Iusto itaque eos quia minus rerum, pariatur alias quas atque at
      Temporibus laudantium cumque at. Quis hic sit molestiae doloremq
    </p>
  </section>
  <section>
    <h2>Sección 2</h2>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Minus q
      Eligendi, modi nesciunt. Ullam, praesentium saepe nisi vitae, nu
      Iusto itaque eos quia minus rerum, pariatur alias quas atque at
      Temporibus laudantium cumque at. Quis hic sit molestiae doloremq
    </p>
  </section>
  <section>
    <h2>Sección 3</h2>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Minus q
      Eligendi, modi nesciunt. Ullam, praesentium saepe nisi vitae, nu
      Iusto itaque eos quia minus rerum, pariatur alias quas atque at
      Temporibus laudantium cumque at. Quis hic sit molestiae doloremq
    </p>
  </section>
</body>
</html>
```

#YOAPRENDOCONDORIAN

Código css:

```

*{
  box-sizing: border-box;
}
body{
  margin: 0;
  background-color: #333;
  color: white;
}

.header{
  background-color: turquoise;
  font-size: 20px;
  color: tomato;
  text-align: center;
  padding: 25px 0;
}

```

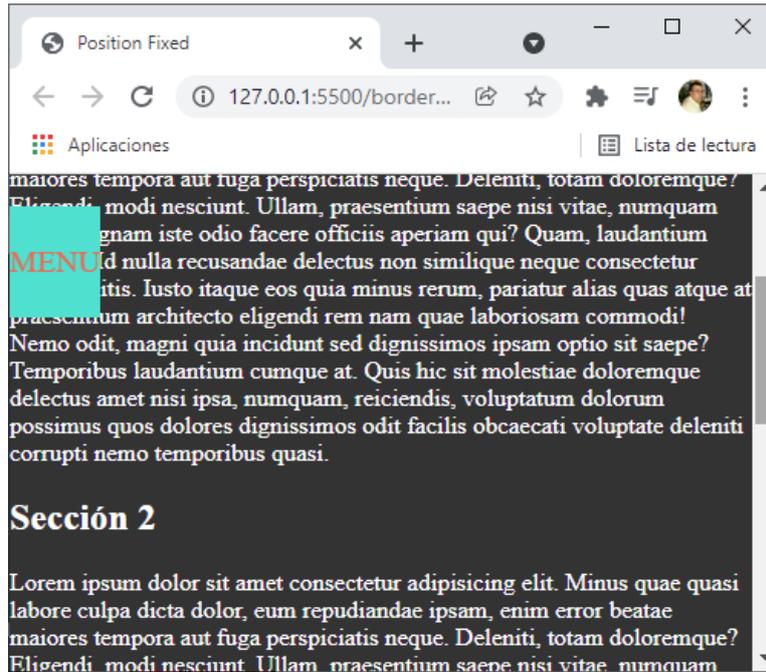
Resultado:



```

.header{
  background-color: turquoise;
  font-size: 20px;
  color: tomato;
  text-align: center;
  padding: 25px 0;
  position: fixed; ←
}

```



```
.header{  
  background-color: #40E0D0; ←  
  font-size: 20px;  
  color: #FF6347; ←  
  text-align: center;  
  padding: 25px 0;  
  position: fixed;  
  top: 0; ←  
}
```



```
.header{  
  background-color: turquoise;  
  font-size: 20px;  
  color: tomato;  
  text-align: center;  
  padding: 25px 0;  
  position: fixed;  
  top: 0;  
  width: 100%;  
}
```



El menú nos está tapando el título principal de la página.

```
body{  
  margin: 0;  
  background-color: #333;  
  color: white;  
  padding-top: 50px;  
}
```



¿Cómo se usa y por qué no funciona Position Sticky?

Código html:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Position Fixed</title>
</head>
<body>
  
  <header class="header">
    <nav class="menu">MENU</nav>
  </header>
  <h1>Position Sticky</h1>
  <section>
    <h2>Sección 1</h2>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Minus q
      Eligendi, modi nesciunt. Ullam, praesentium saepe nisi vitae, nu
      Iusto itaque eos quia minus rerum, pariatur alias quas atque at
      Temporibus laudantium cumque at. Quis hic sit molestiae doloremq
    </p>
  </section>
  <section>
    <h2>Sección 2</h2>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Minus qua
      Eligendi, modi nesciunt. Ullam, praesentium saepe nisi vitae, numq
      Iusto itaque eos quia minus rerum, pariatur alias quas atque at pr
      Temporibus laudantium cumque at. Quis hic sit molestiae doloremque
    </p>
  </section>
  <section>
    <h2>Sección 3</h2>
    <p>
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Minus qua
      Eligendi, modi nesciunt. Ullam, praesentium saepe nisi vitae, numq
      Iusto itaque eos quia minus rerum, pariatur alias quas atque at pr
      Temporibus laudantium cumque at. Quis hic sit molestiae doloremque
    </p>
  </section>
</body>
</html>
```

#YOAPRENDOCONDORIAN

Código css:

```

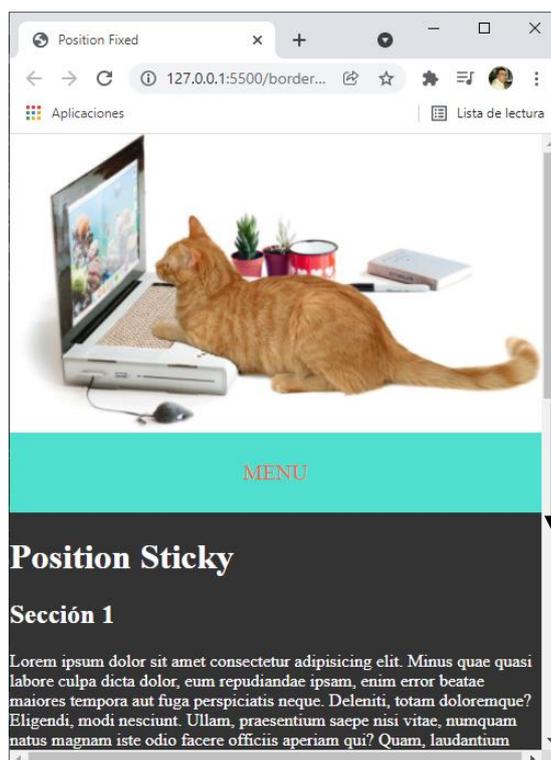
*{
  box-sizing: border-box;
}
body{
  margin: 0;
  background-color: #333;
  color: white;
  padding-top:0px;
}

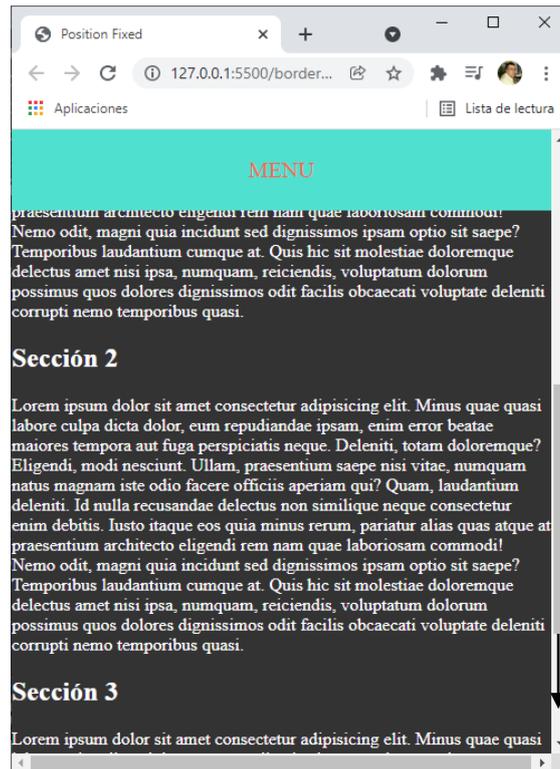
img{
  width: 484px;
  top: 0;
  display:block;
}

.header{
  background-color: turquoise;
  font-size: 20px;
  color: tomato;
  text-align: center;
  padding: 25px 0;
  position: fixed;
  top: 265px;
  width: 100%;
  position: sticky;
  top:0;
}

```

Este será el resultado:

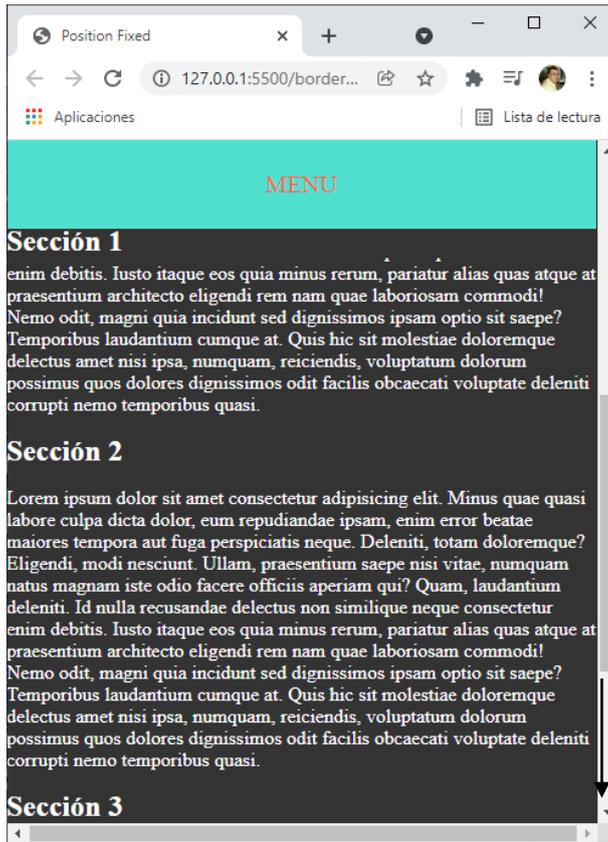




Modificación código css:

```
.header{
  background-color: turquoise;
  font-size: 20px;
  color: tomato;
  text-align: center;
  padding: 25px 0;
  position: fixed;
  top: 0 px;
  width: 100%;
  position: sticky;
  top:0;
  z-index: 1;
}
.article{
  position: sticky;
  top: 70px;
  background-color: #333;
  margin: 0;
}
```

Este será el resultado:



¿Qué es y como funciona la propiedad z-index?

Código html:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Position z-index</title>
</head>
<body>
  <div class="caja">
    <div class="box box-1">
      <h2>Box 1</h2>
    </div>
  </div>
  <div class="box box-2">
    <h2>Box 2</h2>
  </div>
  <div>
    <h2 class="box box-3" >Box 3</h2>
  </div>
</body>
</html>
```

Código css:

```
*{
  box-sizing: border-box;
  margin: 0;
}
body{
  margin: 0;
  background-color: #333;
  color: black;
  padding-top: 0px;
}
.caja{
  width: 200px;
  height: 200px;
  background-color: cadetblue;
}

.box{
  width: 100px;
  height: 100px;
  padding: 35px 20px;
}
```

#YOAPRENDOCONDORIAN

```

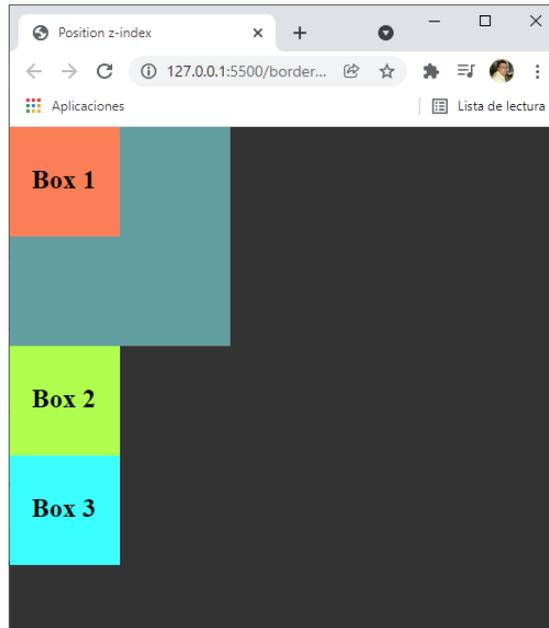
.box-1{
  background-color: coral;
}

.box-2{
  background-color: greenyellow;
}

.box-3{
  background-color: cyan;
}

```

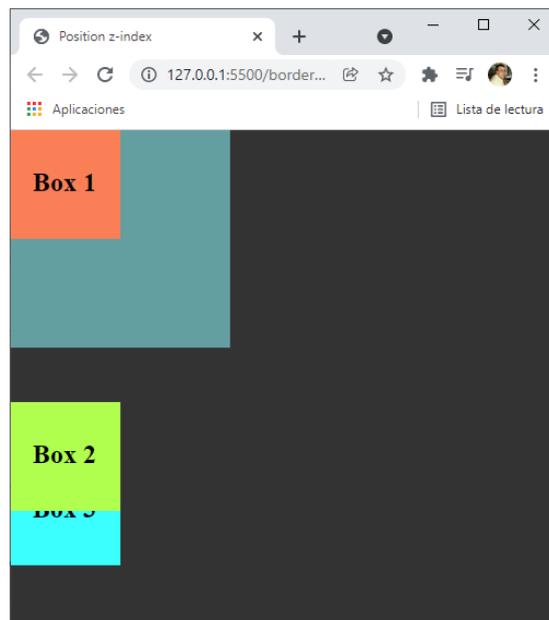
Este será el resultado:



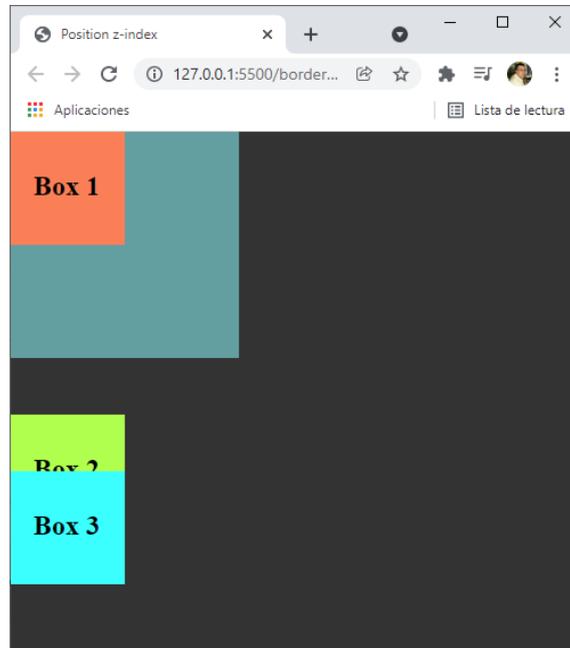
```

.box-2{
  background-color: greenyellow;
  position: relative;
  top: 50px;
}

```



```
.box-3{  
  background-color: cyan;  
  position: relative;  
  z-index: 1;  
}
```



En z-index el número mayor es el que prevalece.

No utilizar números consecutivos, ya que las aplicaciones pueden crecer y nos pueden faltar números intermedios.

Si queremos colorar un padre por delante del hijo esto no va a ser posible, pero si podemos poner al hijo detrás del padre.

```
.box-1{  
  background-color: coral;  
  position: relative;  
  left: 150px;  
  z-index: -1;  
}
```

Este será el resultado:



#YOAPRENDOCONDORIAN

Al padre no se le puede declarar la propiedad z-index solo la que tiene por defecto:

z-index: auto;

Te propongo que realices por tu cuenta la siguiente practica:



Con los conocimientos adquiridos hasta la fecha lo podrás realizar.

En la siguiente página podrás ver la solución.

Código html:

```
<body>
  <div class="caja">
    <div class="box box-1">
      <h2>Box 1</h2>
    </div>
    <div class="box box-1-1">
      <h2>Box 1-1</h2>
    </div>
  </div>
  <div class="box box-2">
    <h2>Box 2</h2>
  </div>
  <div>
    <h2 class="box box-3" >Box 3</h2>
  </div>
</body>
```

Código Css:

```
.box-1-1{
  background-color: red;
  position: relative;
  z-index: -10;
  left: 200px;
  top: -35px;
}
```

¿Qué es stacking context o Concepto de apilamiento?

Código html:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Position z-index</title>
</head>
<body>
  <div class="box box-1">
    <h2>Box 1</h2>
  </div>
  <div class="caja">
    <h1>Apilamiento</h1>
    <p class="parrafo">
      Lorem, ipsum dolor sit amet consectetur adipisicing elit. Tempora praesentium minus
      doloremque expedita repudiandae, earum, soluta, dolore aspernatur quibusdam minima
      maiores odio aliquam vel eum possimus vero commodi accusantium quos.
    </p>
  </div>
```

#YOAPRENDOCONDORIAN

```

<div class="box box-2">
  <h2>Box 2</h2>
</div>
<div>
  <h2 class="box box-3" >Box 3</h2>
</div>
</body>
</html>

```

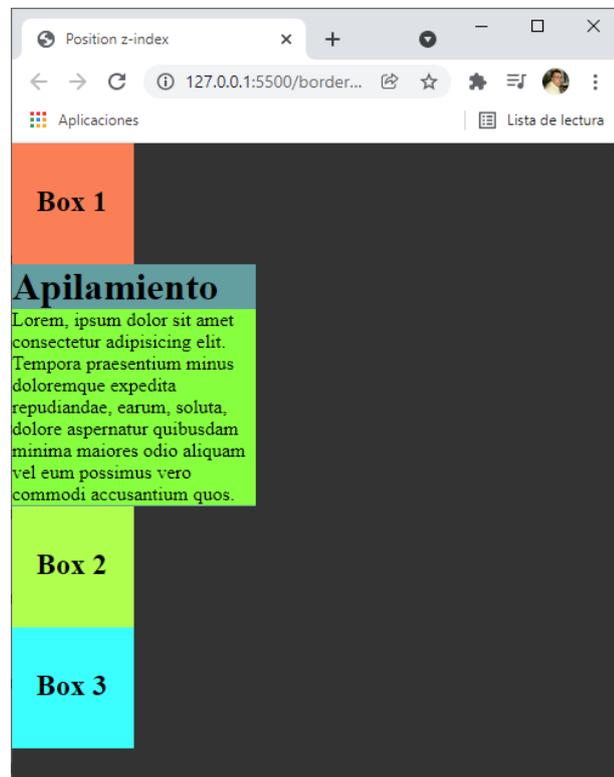
Código css:

```

*{
  box-sizing: border-box;
  margin: 0;
}
body{
  margin: 0;
  background-color: #333;
  color: black;
  padding-top: 0px;
}
.caja{
  width: 200px;
  height: 200px;
  background-color: cadetblue;
}
.box{
  width: 100px;
  height: 100px;
  padding: 35px 20px;
}
.box-1{
  background-color: coral;
}
.box-2{
  background-color: greenyellow;
}
.box-3{
  background-color: cyan;
}
.parrafo{
  background-color: chartreuse;
}

```

Este será el resultado:



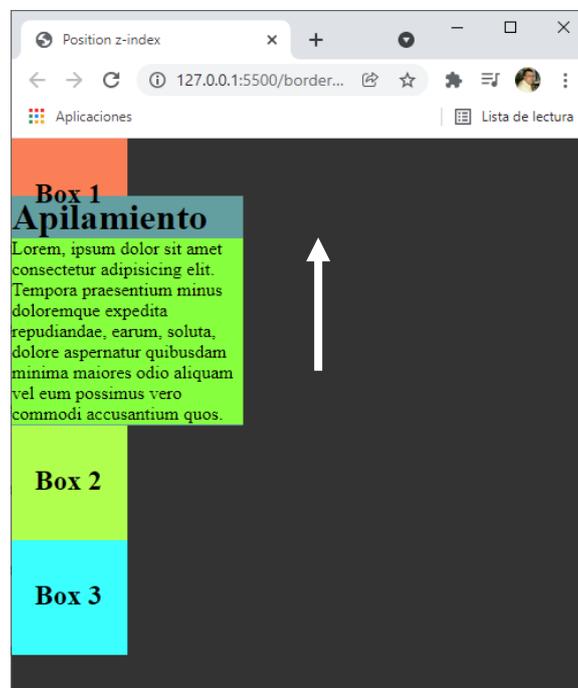
#YOAPRENDOCONDORIAN

El stacking context o el contexto de apilamiento es el espacio donde nuestros elementos se van a ir apilando para que unos queden detrás y otros por delante.

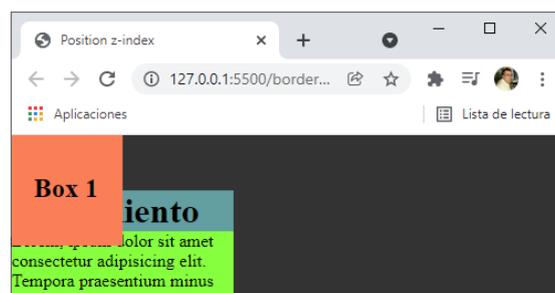
El orden del stacking context es (de delante a atrás):

- Elementos posicionados con un z-index de 1 ó más.
- Elementos posicionados sin z-index declarado (o z-index: auto).
- Elementos no posicionados.
- Elementos con z-index negativo.

```
.caja{
  width: 200px;
  height: 200px;
  background-color: cadetblue;
  margin-top: -50px; ←
}
```

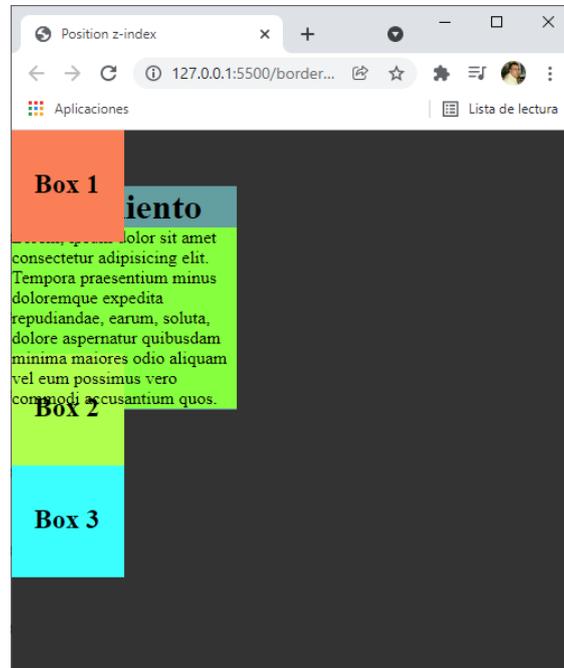


```
.box-1{
  background-color: coral;
  position: relative; ←
}
```

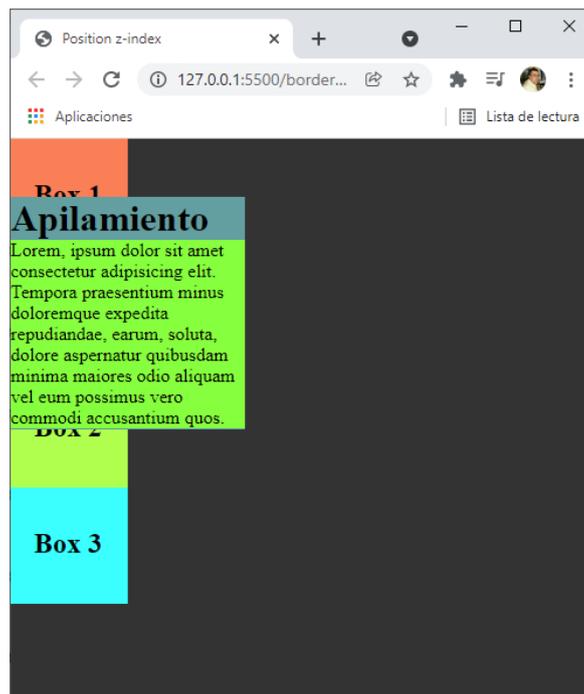


#YOAPRENDOCONDORIAN

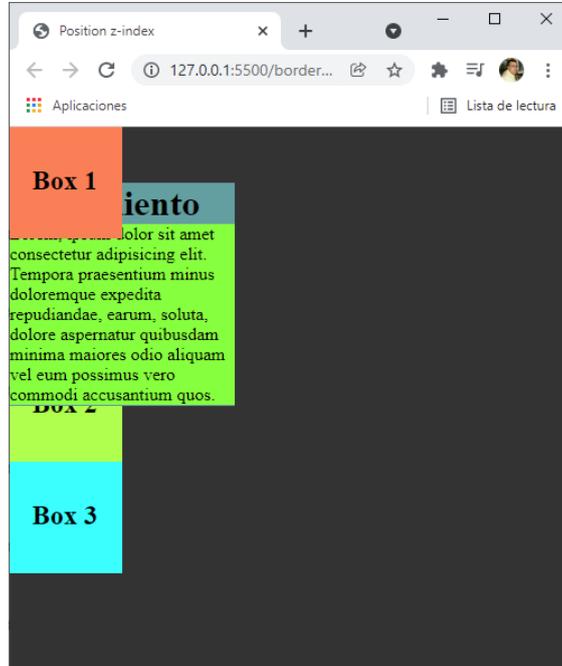
```
.box-2{
  background-color: greenyellow;
  margin-top: -50px; ←
}
```



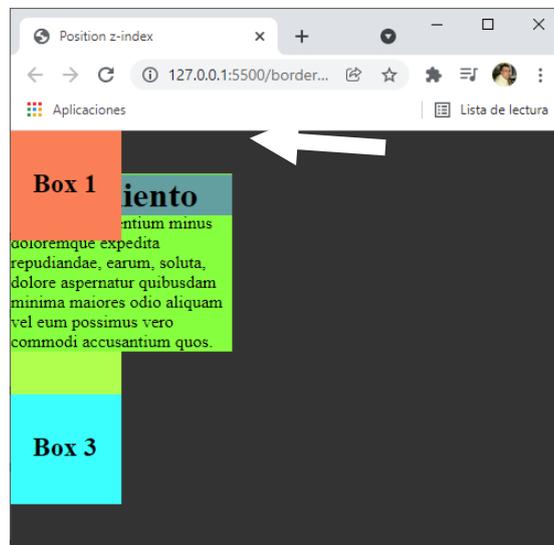
```
.caja{
  width: 200px;
  height: 200px;
  background-color: cadetblue;
  margin-top: -50px;
  position: relative; ←
}
```



```
.box-1{  
  background-color: coral;  
  position: relative;  
  z-index: 1; ←  
}
```



Realiza los pasos necesarios:



En la página siguiente tienes la solución:

Código html:

```
<body>
  <div class="box box-1">
    <h2>Box 1</h2>
  </div>
  <div class="caja">
    <h1>Apilamiento</h1>
    <div>
      <p class="parrafo">
        Lorem, ipsum dolor sit amet consectetur ad
        doloremque expedita repudiandae, earum, so
        maiores odio aliquam vel eum possimus vero
      </p>
    </div>
  </div>
  <div class="box box-2">
    <h2>Box 2</h2>
  </div>
  <div>
    <h2 class="box box-3" >Box 3</h2>
  </div>
</body>
```

Código css:

```
*{
  box-sizing: border-box;
  margin: 0;
}
body{
  margin: 0;
  background-color:#333;
  color: black;
  padding-top:0px;
}
.caja{
  width: 200px;
  height: 200px;
  margin-top: -60px;
}
h1{
  background-color: cadetblue;
  position: relative;
}
.box{
  width: 100px;
  height: 100px;
  padding: 35px 20px;
```

#YOAPRENDOCONDORIAN

```
}  
.box-1{  
  background-color: coral;  
  position: relative;  
  z-index: 1;  
}  
.box-2{  
  background-color: greenyellow;  
  margin-top: -100px;  
  position: relative;  
  z-index: -10;  
}  
  
.box-3{  
  background-color: cyan;  
}  
  
.parrafo{  
  background-color: chartreuse;  
  margin: 0;  
  margin-top: -38px;  
}
```

¿Cómo ordenar propiedades?

Ordenar las propiedades es importante, no existe ninguna norma ni especificación sobre como hacerlo, pero la mayoría de expertos coinciden en los mismos puntos.

1. Propiedades de posicionamiento.
2. Propiedades de box model.
3. Propiedades de texto.
4. Propiedades visuales (colores, bordes, background...)
5. El resto.

```
body{  
  /* Position */  
  Position: relative;  
  Top: 0;  
  Left: 0;  
  
  /* box model */  
  display: block;  
  width: 300px;  
  height: 600px;  
  padding: 10px;  
  margin: 10px;  
  overflow: hidden;
```

```
/* Text */
font-size: 16px;
text-align: left;

/* Visual */
color: blue;
border: 2px solid red;
border-radius: 10px;

/* Varios */
opacity: 1;
}
```

¿Que son las medidas absolutas y medidas relativas?

Las medidas de CSS se pueden agrupar en 2 grupos, absolutas y relativas.

Las medidas absolutas NO CAMBIAN y siempre mantendrán su tamaño independientemente del dispositivo o la interacción del usuario.

La más usada es px (píxeles) cuya medida es 1/96 de 1in, pero existen otras como:

- cm (centímetros) -> 1cm = 96px/2,54
- mm (milímetros) -> 1mm = 1/10 de 1 cm
- Q (cuarto de milímetro) -> 1Q = 1/40 de 1cm
- ln(inches, pulgadas) -> 1in = 2,54cm = 96px
- pc (picas) -> 1pc = 1/16 de 1in
- pt (puntos) -> 1pt = 1/72 de 11in

Las medidas relativas SIEMPRE dependen de un contexto son las que se recomiendan utilizar en la gran mayoría de ocasiones porque nos van a permitir que todos los elementos escalen proporcionalmente.

Las más usadas son em, rem, %, vw, vh, vmin y vmax aunque tenemos otras como:

- ex -> Altura x de la fuente del elemento.
- ch -> La medida del ancho del grifo "0" de la letra del elemento.
- lh -> Altura de la línea del elemento.

Unidades de medida EM y REM

rem -> Corresponde a la media de la "m" de la raíz del documento (html).

Ejemplo: 16px = 1rem.

em -> Corresponde a la media de la "m" del contexto donde nos encontremos.

El estándar del tamaño de fuente por defecto en la raíz del documento es 16px, y no es recomendable modificarlo.

NOTA: Tanto "em" como "rem" se calcula en base a la propiedad Font-size (tamaño de fuente).

Código html:

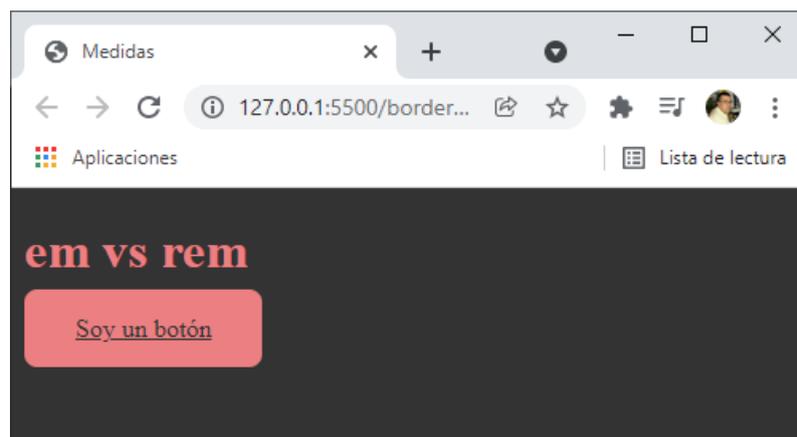
```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Medidas</title>
</head>
<body>
  <h1 class="title">em vs rem</h1>
  <a href="#" class="link">Soy un botón</a>
</body>
</html>
```

Código css:

```
body{
  background-color: #333;
  color: lightcoral;
}

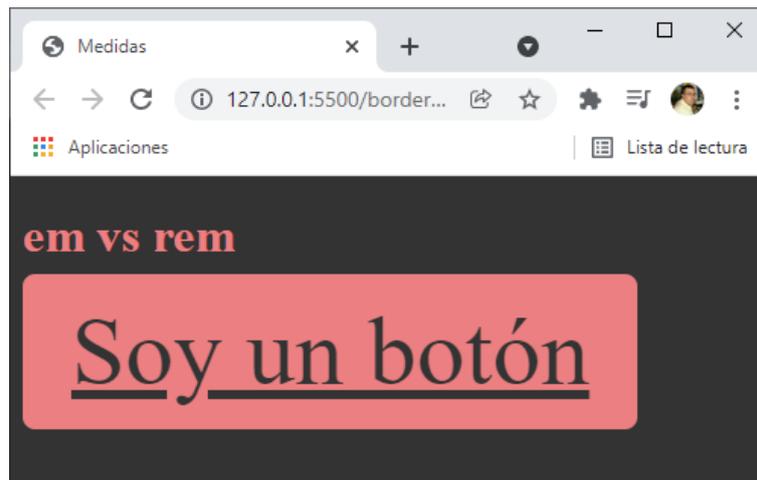
.link{
  background-color: lightcoral;
  color: #333;
  padding: 16px 32px;
  border-radius: 8px;
}
```

Resultado:

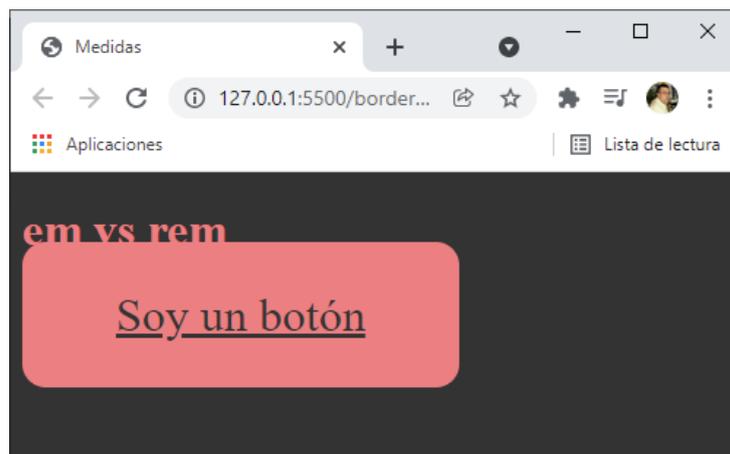


```
.link{
  background-color: lightcoral;
  color: #333;
  padding: 16px 32px;
  border-radius: 8px;
  font-size: 4rem; ←
}
```

Este será el resultado:



```
.link{  
  background-color: lightcoral;  
  color: #333;  
  padding: 1em 2em;  
  border-radius: 0.5em;  
  font-size: 2rem;  
}
```



Mantiene mejor las proporciones.

¿Qué es y cómo se usa width en % y width auto?

% -> Cuando usamos porcentaje usamos de referencia el tamaño del contenedor y el navegador calcula ese porcentaje. Esto no ocurre cuando usamos transform.

auto -> Cuando usamos auto le pedimos al navegador que calcule en función al espacio disponible.

Código html:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Medidas</title>
</head>
<body>
  <h1>% vs auto</h1>
  <div class="contenedor">
    <div class="contenedor-hijo"></div>
  </div>
</body>
</html>
```

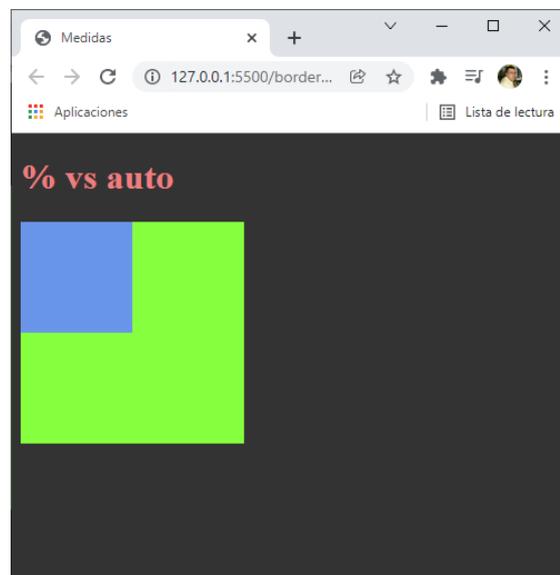
Código css:

```
body{
  background-color: #333;
  color: lightcoral;
}

.contenedor{
  width: 200px;
  height: 200px;
  background-color: chartreuse;
}

.contenedor-hijo{
  width: 50%;
  height: 50%;
  background-color: cornflowerblue;
}
```

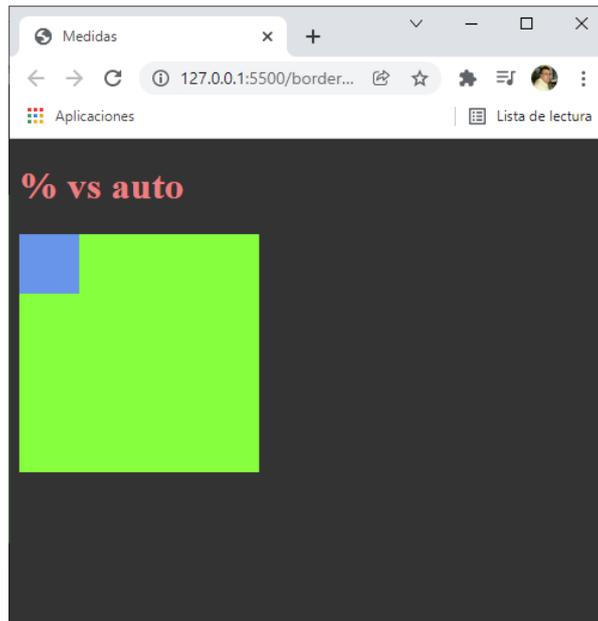
Este será el resultado:



Ahora vamos a cambiar el porcentaje del contenedor-hijo.

```
.contenedor-hijo{  
  width: 25%;  
  height: 25%;  
  background-color: #cornflowerblue;  
}
```

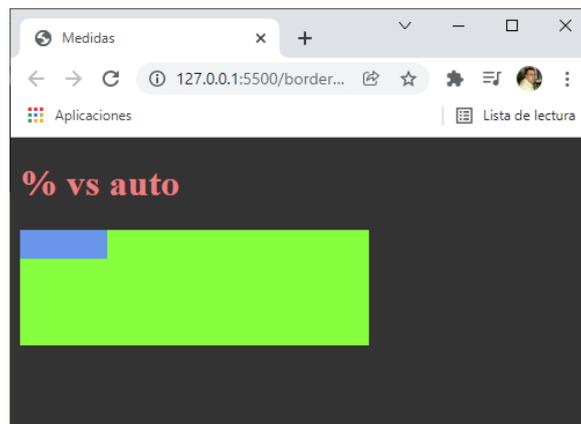
Este será el resultado:



Ahora vamos a cambiar las dimensiones del contenedor padre.

```
.contenedor{  
  width: 300px;  
  height: 100px;  
  background-color: #chartreuse;  
}
```

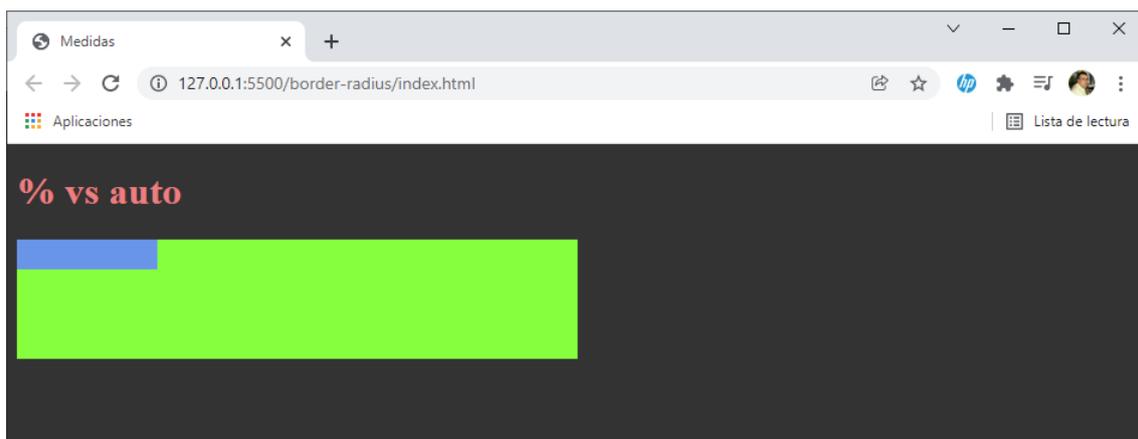
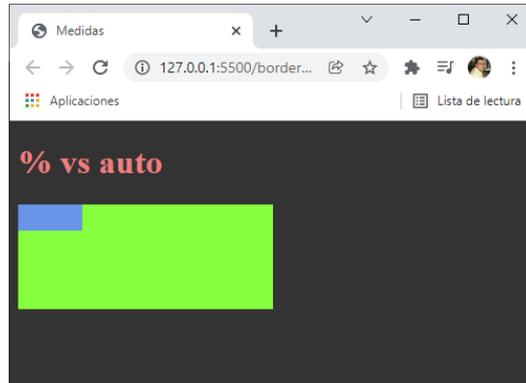
Este será el resultado:



El contenedor-hijo se adapta a sus porcentajes.

```
.contenedor{
  width: 50%;
  height: 100px;
  background-color: ■ chartreuse;
}
```

Si al contenedor padre en lo ancho le ponemos 50%, este lo calculará con respecto al body, esto significa que si el tamaño del navegador se modifica es también se modificará proporcionalmente.



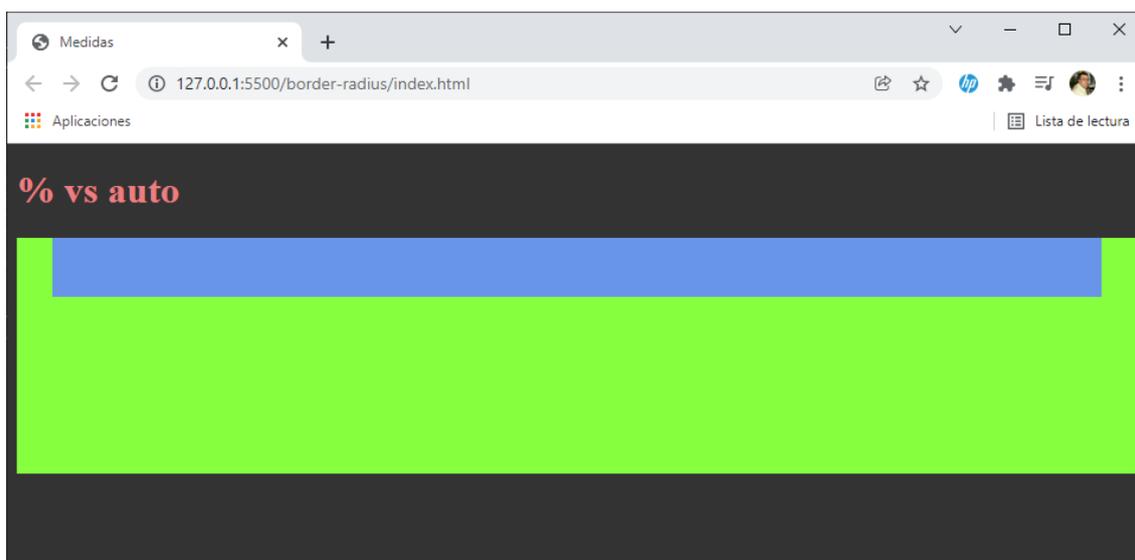
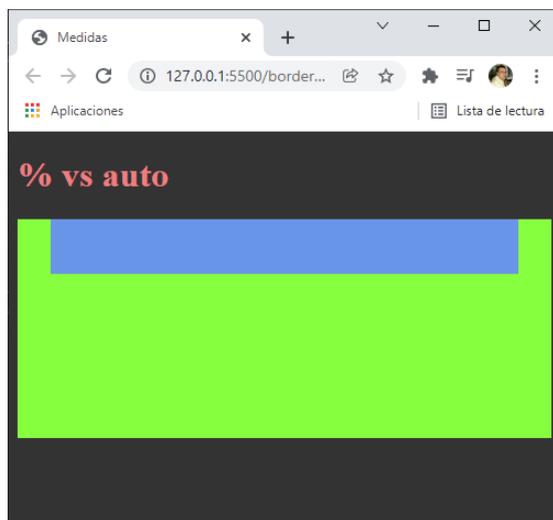
```
.contenedor{
  width: 100%;
  height: 200px;
  background-color: ■ chartreuse;
}

.contenedor-hijo{
  width: auto; ←
  height: 25%;
  margin: 0 30px;
  background-color: ■ cornflowerblue;
}
```

El contenedor padre tiene un width 100%, esto significa que utiliza el 100% del tamaño del body, es decir el ancho de nuestro navegador.

#YOAPRENDOCONDORIAN

Si en el contenedor-hijo queremos aplicar un margen de 30px tanto a la derecha como a la izquierda y el resto sea para el ancho del contenedor hijo, pondremos auto, de este modo se irá adaptando al ancho del navegador.



¿Cómo se utilizan y funcionan HEIGHT % y HEIGHT AUTO?

% -> Cuando estamos dentro de un contenedor, este tiene que tener un alto declarado, si no podrá calcular el porcentaje. Si se le aplica una medida con porcentaje se sustituirá automáticamente por auto.

auto -> Cuando utilizamos auto el alto lo calcula el navegador en base al contenido del elemento.

NOTA: Height es una propiedad que tenéis que usar con mucho cuidado. Si no es necesario establecer un alto, dejad que el contenido sea el que decida el alto del elemento.

#YOAPRENDOCONDORIAN

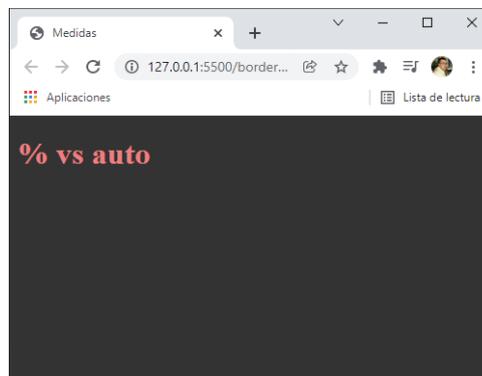
```

.contenedor{
  width: 300px;
  /*height: 300px;*/
  background-color: #008080;
}

.contenedor-hijo{
  width: 50%;
  height: 50%;
  background-color: #6495ED;
}

```

Al contenedor padre hemos comentado su dimensión en altura, si al contenedor-hijo le estamos diciendo que tenga un height del 50%, como no tiene referencia del contenedor-padre, este será el resultado:



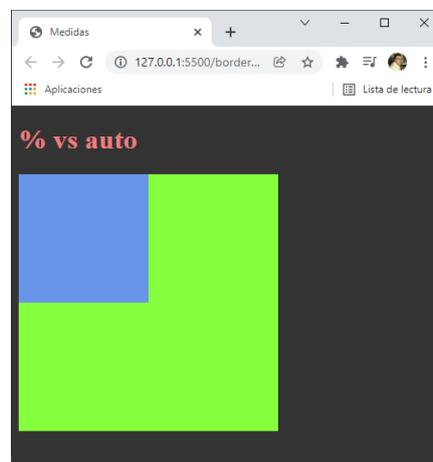
```

.contenedor{
  width: 300px;
  height: 300px; ←
  background-color: #008080;
}

.contenedor-hijo{
  width: 50%;
  height: 50%;
  background-color: #6495ED;
}

```

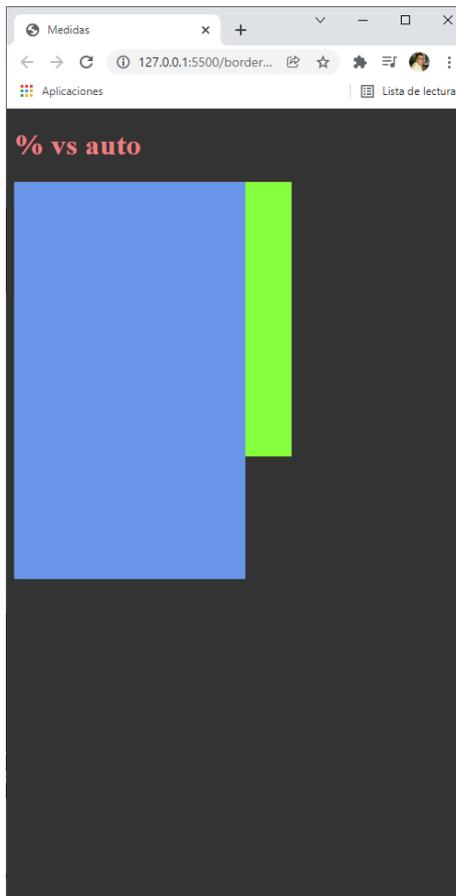
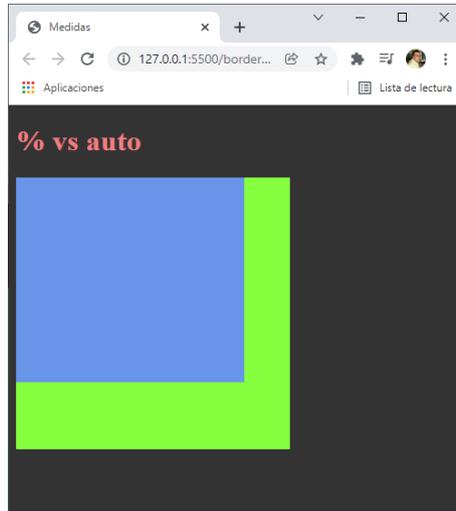
Hemos quitado el modo comentario, este será el resultado:



#YOAPRENDOCONDORIAN

```
.contenedor-hijo{  
  position: absolute; ←  
  width: 50%;  
  height: 50%;  
  background-color: cornflowerblue;  
}
```

Al contenedor-hijo le hemos agregado la propiedad `position: absolute;`; cual será el resultado:



El porcentaje lo calcula según el alto de la página web.

Si le ponemos position (relative o sticky) el problema se resuelve.

¿Cómo funcionan las medidas relativas al Viewport responsive?

VIEWPORT

El viewport es el área útil donde se mostrará la página web. **IMPORTANTE**, tened en cuenta que es el área visible sin hacer scroll.

Podemos usar viewPort como medida de varias formas:

vw -> Viewport Width, hace referencia al ancho del viewport.

vmax -> Utiliza el valor más grande entre el ancho y alto del viewport.

vmin -> Utiliza el valor más pequeño entre el ancho y el alto del viewport.

1vm = 1% de ancho del viewport.

1vh = 1% del alto del viewport.

1vmin = 1% del valor que sea más pequeño del viewport.

1vmax = 1% del valor que sea más grande del viewport.

Código html:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Medidas</title>
</head>
<body>
  <div class="header">
    Header
  </div>
  <div>
    <h1>Medidas en viewport</h1>
  </div>
  <div class="contenedor-padre">
    <div class="contenedor-hijo">
    </div>
  </div>
  <div class="text">
    <p>
      Lorem ipsum dolor, sit amet consectetur adipisicing elit. Cumque
      quo cum, explicabo consequatur natus, dolorem, deleniti molestia
      numquam architecto! Earum, omnis? Nam necessitatibus architecto
    </p>
  </div>
</body>
</html>
```

#YOAPRENDOCONDORIAN

```

|   </div>
| </body>
| </html>

```

Código css:

```

body{
|   background-color: #333;
|   color: lightcoral;
| }
.header{
|   text-align: center;
|   background-color: cyan;
|   width: 300px;
| }
.contenedor-padre{
|   width: 300px;
|   height: 300px;
|   background-color: darkmagenta;
| }

.contenedor-hijo{
|   width: 50%;
|   height: 100%;
|   background-color: yellow;
| }

.text{
|   width: 300px;
|   text-align: justify;
| }

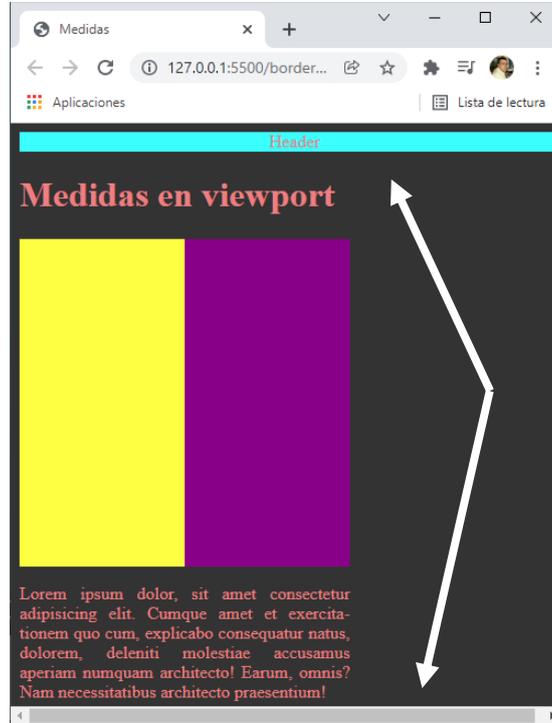
```

Este será el resultado:

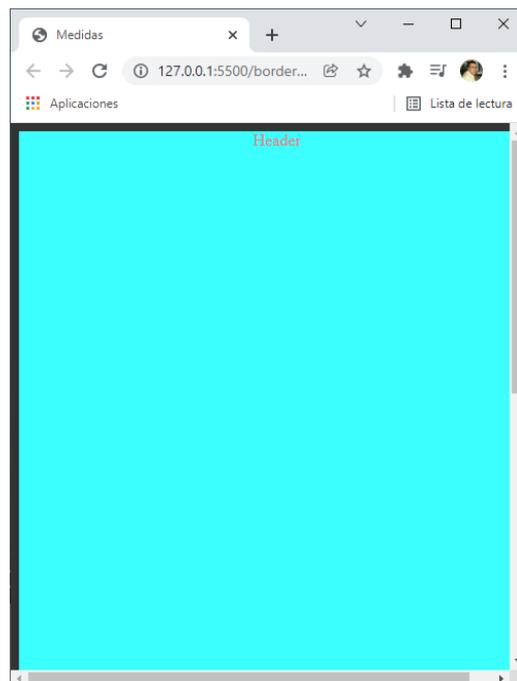


#YOAPRENDOCONDORIAN

```
.header{  
  text-align: center;  
  background-color: cyan;  
  width: 100vw;  
}
```



```
.header{  
  text-align: center;  
  background-color: cyan;  
  width: 100vw;  
  height: 100vh;  
}
```



#YOAPRENDOCONDORIAN

Vamos a modificar los valores de css.

```
body{
  background-color: #333;
  color: lightcoral;
}
.header{
  text-align: center;
  background-color: cyan;
}

.contenedor-padre{
  width: 300px;
  height: 300px;
  background-color: darkmagenta;
}

.contenedor-hijo{
  width: 100%;
  height: 50%;
  background-color: yellow;
}

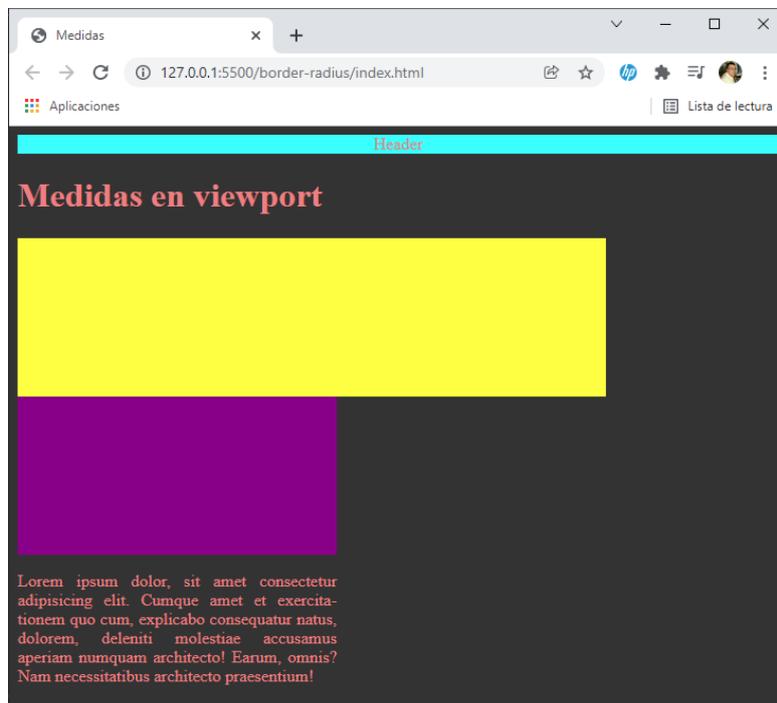
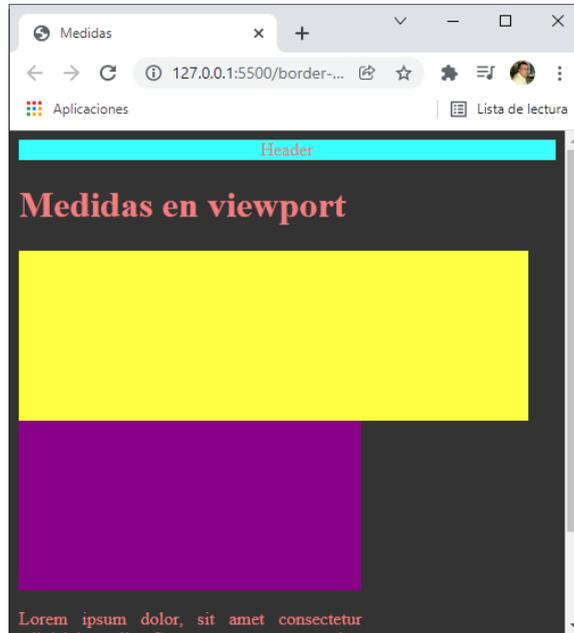
.text{
  width: 300px;
  text-align: justify;
}
```



#YOAPRENDOCONDORIAN

```
.contenedor-hijo{  
  width: 100vmin;  
  height: 50%;  
  background-color: yellow;  
}
```

Mientras el ancho sea más pequeño que el alto.



#YOAPRENDOCONDORIAN

```
.contenedor-hijo{
  position: sticky;
  top: 0;
  width:30vmax;
  height: 150px;
  background-color: yellow;
}
```



Para que puedas ver el resultado.

Medidas relativas Min-Width, Min-Height, Max-Width y Max-height

Limitaciones al ancho y alto.

Cuando establecemos un width y/o height con una medida que es relativa, es posible que no queramos que se respeten esa medida en todos los casos.

Para ello tenemos propiedades que limitan esas medidas:

- min-width: Indica el ancho mínimo que puede alcanzar el elemento.
- max-width: Indica el ancho máximo que puede alcanzar el elemento.
- min-height: Indica el alto mínimo que puede alcanzar el elemento.
- max-height: Indica el alto máximo que puede alcanzar el elemento.

Código html:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Medidas</title>
</head>
<body>
  <header>Header</header>
</body>
<h1>Min and max width & height</h1>
<div class="elemento">

</div>
</html>
```

Código css:

```
html{
  height: 300px;
}

body{
  background-color: #333;
  color: lightcoral;
}

.elemento{
  width: 100%;
  max-width: 600px; ←
  height: 200px;
  background-color: greenyellow;
}
```

Este será el resultado:



#YOAPRENDOCONDORIAN

```
.elemento{
  width: 100%;
  max-width: 600px;
  min-width: 300PX; ←
  height: 200px;
  background-color: greenyellow;
}
```

Cuando vayamos encogiendo el navegador horizontalmente llegará un momento que se activará la barra de scroll horizontal porque nuestro elemento deja de estrecharse.

```
.elemento{
  width: 100%;
  max-width: 600px;
  min-width: 300PX;
  height: 40vw;
  max-height: 300px;
  min-height: 100px;
  background-color: greenyellow;
}
```

Con la medida 40vw le decimos que el alto se tiene que ir ajustando proporcionalmente según el ancho de la ventana de nuestro navegador.

Con las propiedades max-height y min-height limitamos el alto máximo y el alto mínimo.

Si queremos que un elemento nos quede siempre al final del último elemento, y si este crece que no lo pise.

Código html:

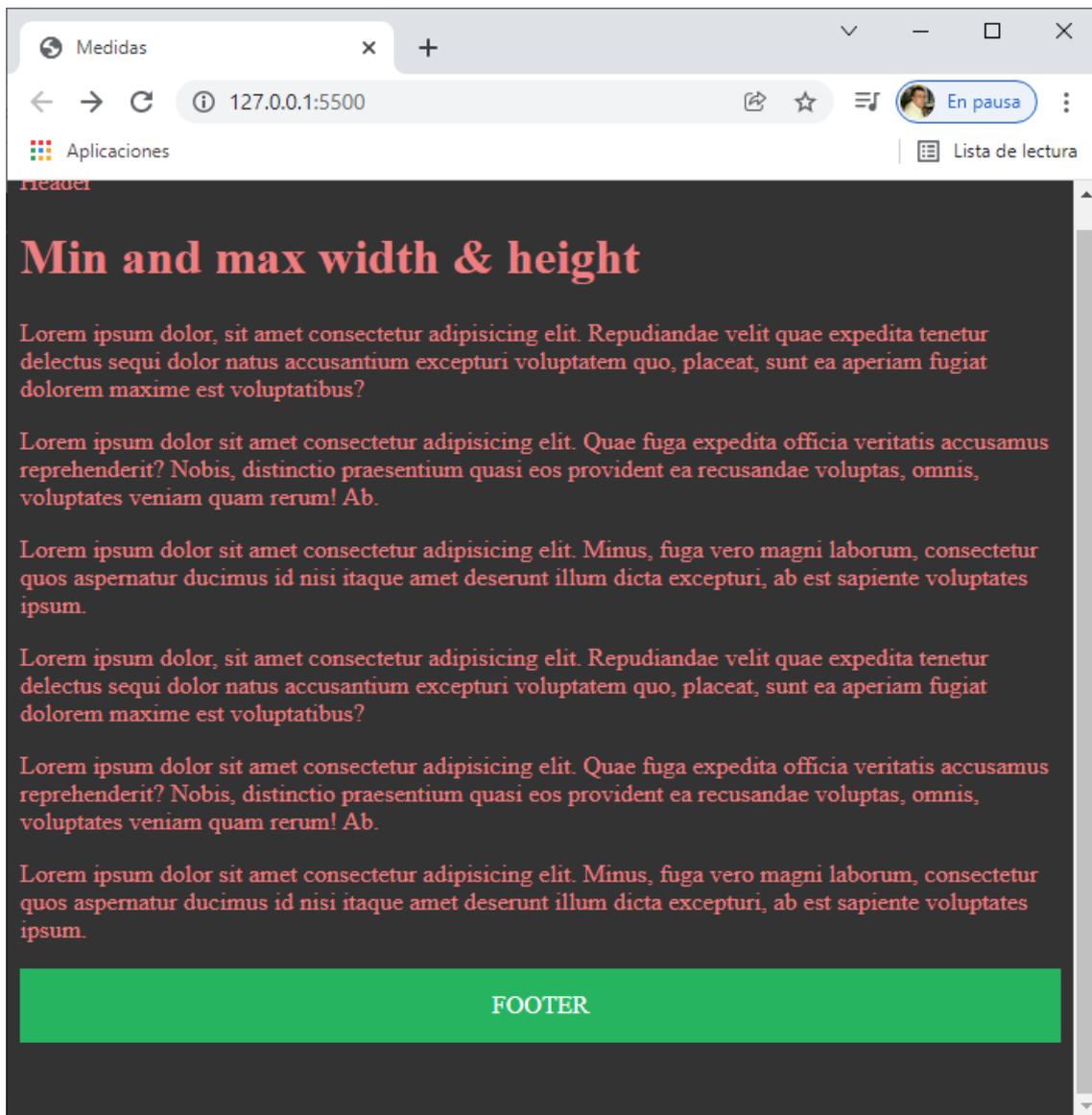
```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <link rel="stylesheet" href="css/styles.css">
8   <title>Medidas</title>
9 </head>
10 <body>
11   <header>Header</header>
12   <h1>Min and max width & height</h1>
13   <div class="elemento">
14     <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Repudiandae ve
15     <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Quae fuga expe
16     <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Minus, fuga ve
17     <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Repudiandae ve
18     <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Quae fuga expe
19     <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Minus, fuga ve
20   </div>
21   <footer>FOOTER</footer>
22 </body>
23 </html>
```

Código css:

```
body{
  background-color: #333;
  color: lightcoral;
  position: relative;
  min-height: 100vh;
}

footer{
  background-color: rgb(12, 180, 91);
  text-align: center;
  color: white;
  padding: 15px 0;
}
```

Este será el resultado:



#YOAPRENDOCONDORIAN

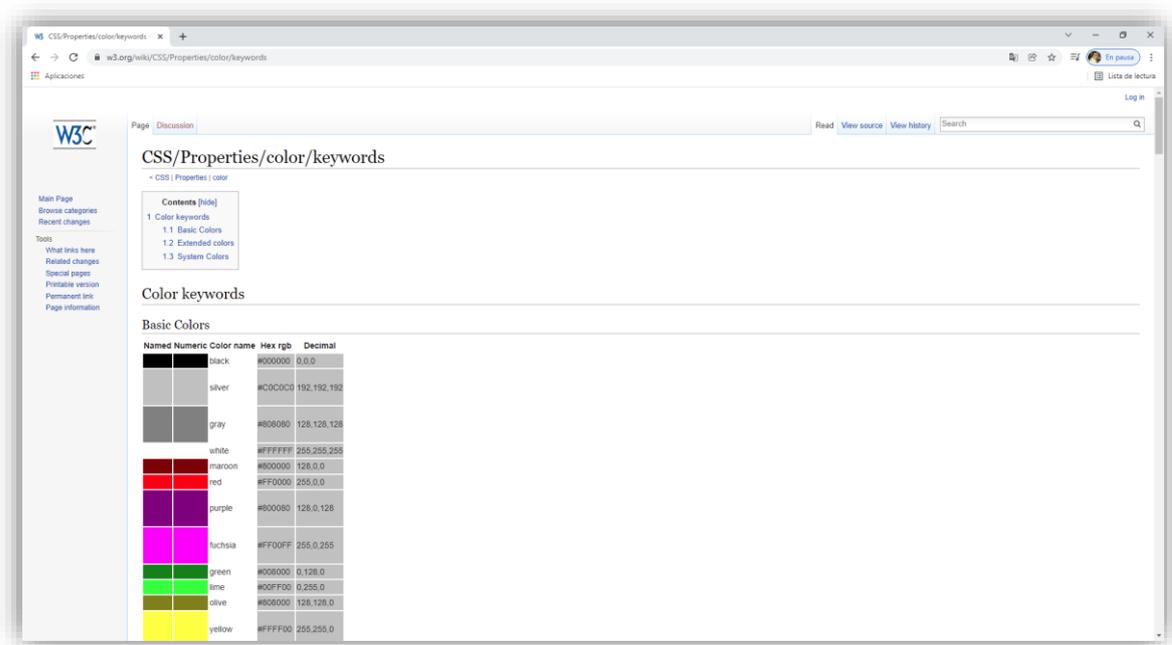
Todo sobre códigos de colores en css [palabras clave]

La mayoría de las pantallas funcionan con un modo de color aditivo. La suma de RGB da como resultado un blanco.

Tenemos varias formas de dar color en CSS.

keywords -> Palabra clave que representan colores

<https://www.w3.org/wiki/CSS/Properties/color/keywords>



IMPORTANTE: Recordar que existe el color "transparent".

Código html:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Medidas</title>
</head>
<body>
<div class="box">

</div>

</body>
</html>
```

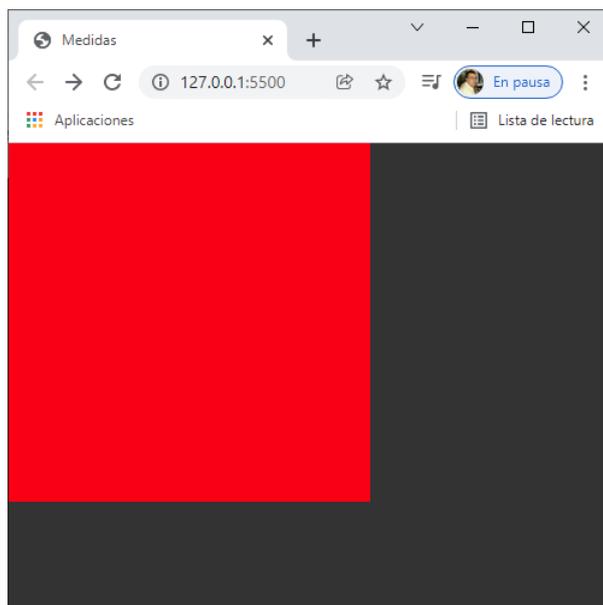
#YOAPRENDOCONDORIAN

Código css:

```
body{
  margin: 0;
  background-color: #333;
}

.box{
  width: 300px;
  height: 300px;
  background-color: red;
}
```

Resultado:



Código html.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Medidas</title>
</head>
<body>
<div class="box">

</div>
</body>
</html>
```

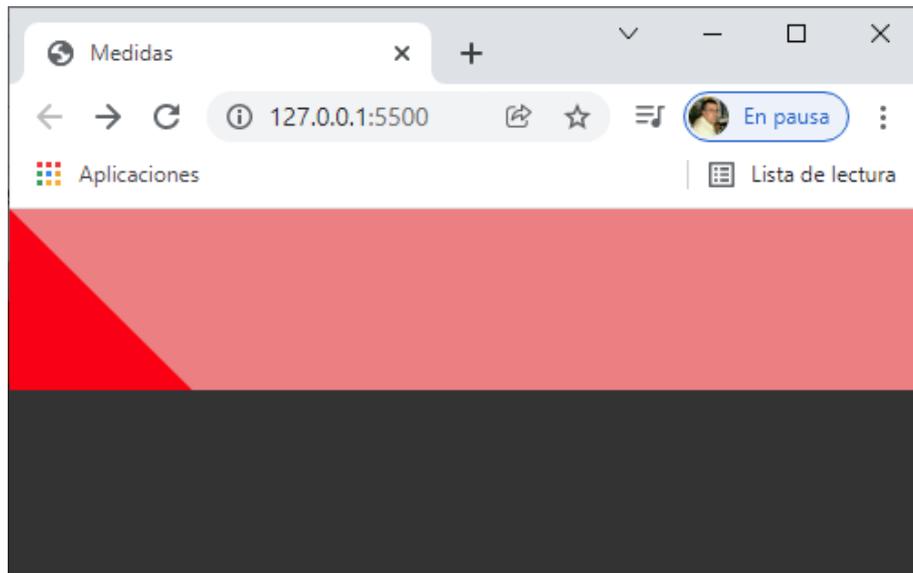
#YOAPRENDOCONDORIAN

Código css:

```
body{
  margin: 0;
  background-color: #333;
}

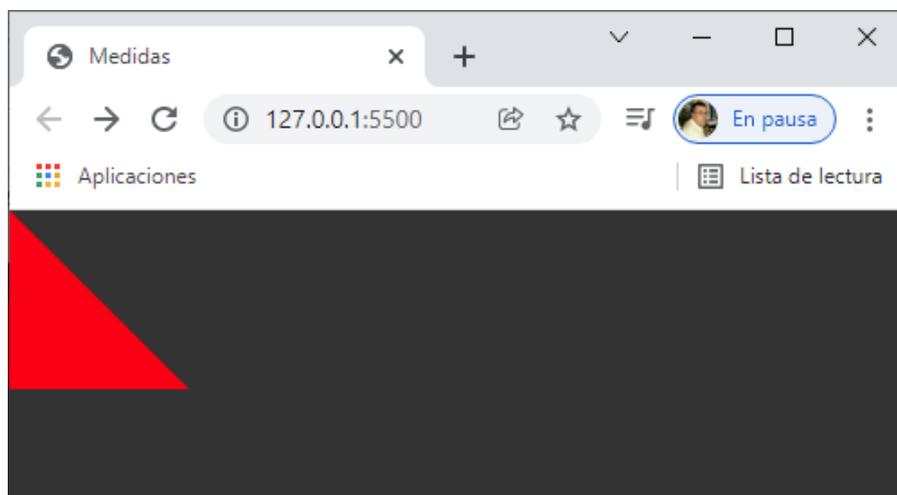
.box{
  border-left: 100px solid red;
  border-top: 100px solid lightcoral
}
```

Este será el resultado:



```
.box{
  border-left: 100px solid red;
  border-top: 100px solid transparent
}
```

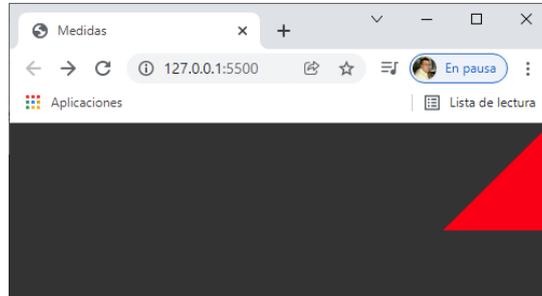
Lo cambiamos a transparent.



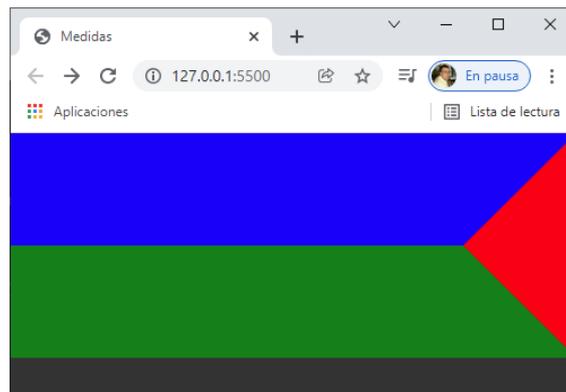
#YOAPRENDOCONDORIAN

```
.box{
  border-right: 100px solid red;
  border-top: 100px solid transparent;
}
```

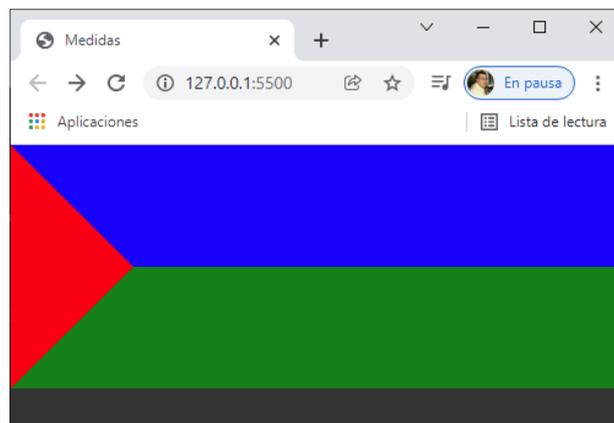
Cambiamos left por right.



```
.box{
  border-right: 100px solid red;
  border-top: 100px solid blue;
  border-bottom: 100px solid green;
}
```

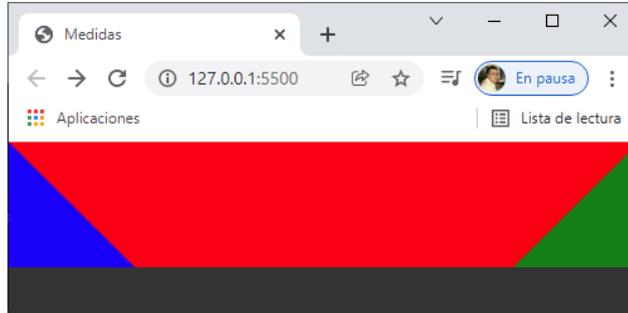


```
.box{
  border-left: 100px solid red;
  border-top: 100px solid blue;
  border-bottom: 100px solid green;
}
```

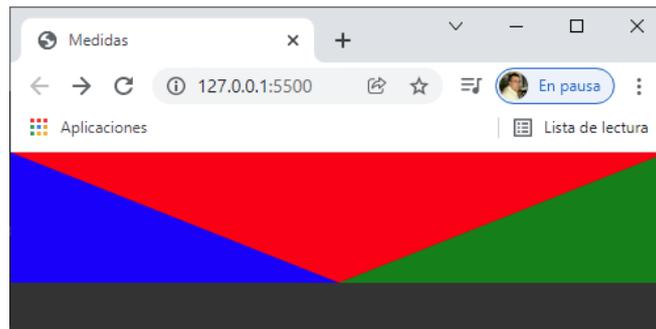


#YOAPRENDOCONDORIAN

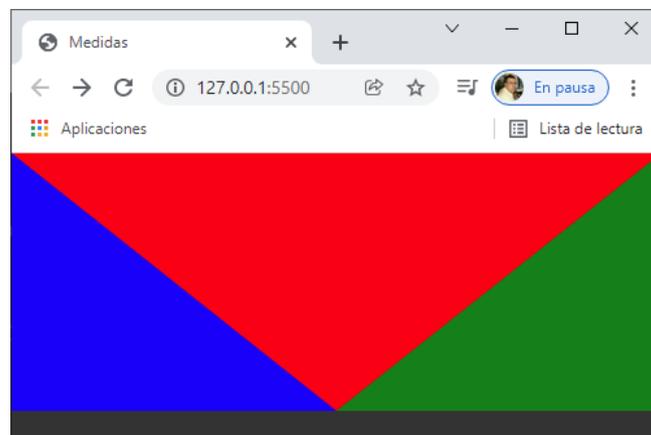
```
.box{
  border-top: 100px solid red;
  border-left: 100px solid blue;
  border-right: 100px solid green;
}
```



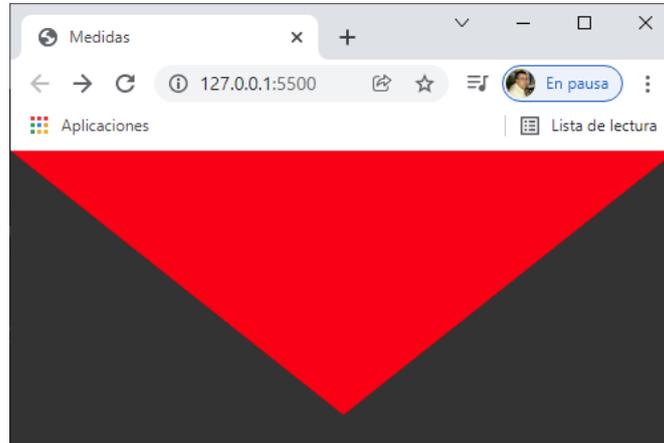
```
.box{
  border-top: 100px solid red;
  border-left: 250px solid blue;
  border-right: 250px solid green;
}
```



```
.box{
  border-top: 200px solid red;
  border-left: 250px solid blue;
  border-right: 250px solid green;
}
```



```
.box{  
  border-top: 200px solid red;  
  border-left:250px solid transparent;  
  border-right: 250px solid transparent;  
}
```



Currentcolor ¿Qué es y cómo se usa?

CurrentColor: Es una palabra clave que se puede usar en lugar de cualquier valor de color y usará el valor ya sea establecido o heredado en la propiedad color.

Código html:

```
<!DOCTYPE html>  
<html lang="es">  
<head>  
  <meta charset="UTF-8">  
  <meta http-equiv="X-UA-Compatible" content="IE=edge">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <link rel="stylesheet" href="css/styles.css">  
  <title>Medidas</title>  
</head>  
<body>  
  <div class="main">  
    <div class="box">  
    </div>  
  </div>  
</body>  
</html>
```

Código CSS:

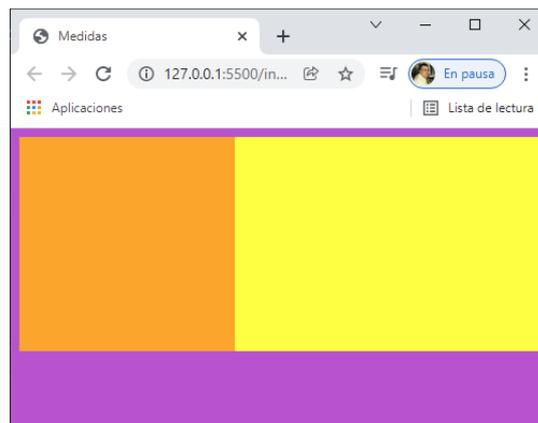
```
body{
  background-color: #mediumorchid;
  color: #333;
}

.main{
  color: yellow;
  background-color: currentColor;
}

.box{
  width: 200px;
  height: 200px;
  background-color: orange;
}
```

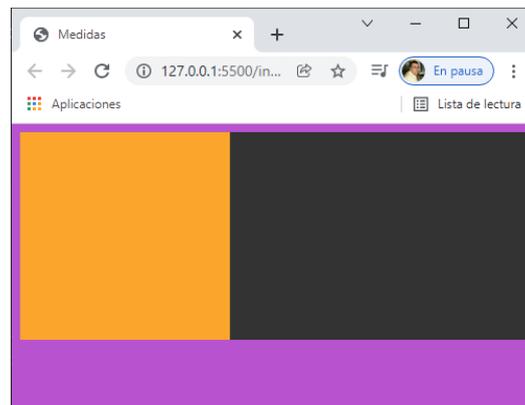
Currentcolor asume el color definido con anterioridad más cercano, en este ejemplo el la clase main tenemos seleccionado el color yellow.

Por este motivo el elemento main como color de fondo será amarillo.



```
.main{
  background-color: currentColor;
}
```

Eliminamos el color amarillo del elemento la clase main, ahora el color definido que tiene está en el elemento body que es un #333, este será el resultado:

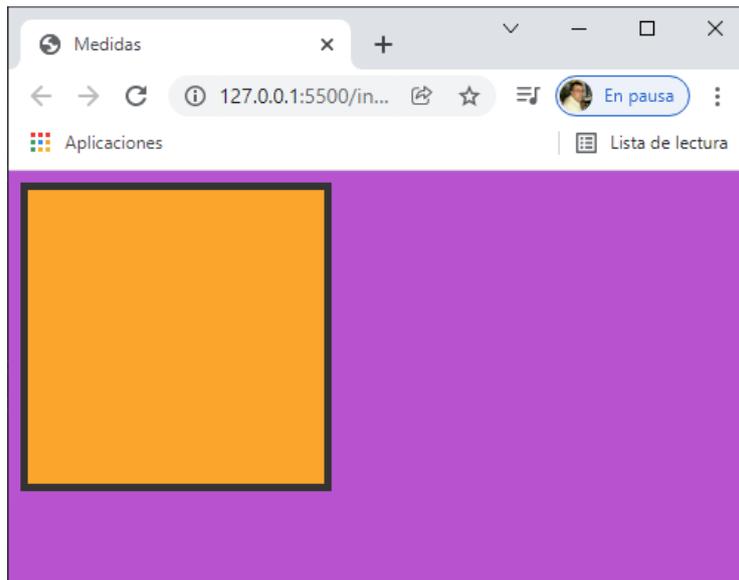


#YOAPRENDOCONDORIAN

Si no hubiera declarado ningún color asumirá por defecto el color negro.

```
body{
  background-color: #mediumorchid;
  color: #333;
}

.box{
  width: 200px;
  height: 200px;
  background-color: orange;
  border: solid 5px currentColor;
}
```



La propiedad borde también heredera los colores.

Modo de color RGB [Diferencia entre colores RGBA]

RGB/RGBA

En el modo RGB tenemos tres canales, (Red, Green, Blue) y tenemos 8 bits de color por canal, (cada bit tiene dos posibles valores 0 ó 1) lo que significa que tenemos 2^8 posibles valores, que van desde 0 hasta 255.

Esto nos da un total de $255 * 255 * 255 = 16.581.375$ colores.

Su sintaxis es `rgb(R,G,B)`

Si ponemos todos los canales a 0 tendremos un negro puro y si los ponemos a 255 tendremos un blanco.

Para movernos por la gama de grises debemos poner los 3 canales con el mismo valor, si ponemos 3 canales a 128 tendremos un gris puro.

El modo RGB nos da la opción de utilizar un cuarto canal que corresponde al canal Alpha, es decir, la transparencia, y su valor va desde 0 a 1. Tenemos dos sintaxis para utilizar el canal Alpha.

#YOAPRENDOCONDORIAN

rgba(R,G,B,A)

rgb(R G B / A)

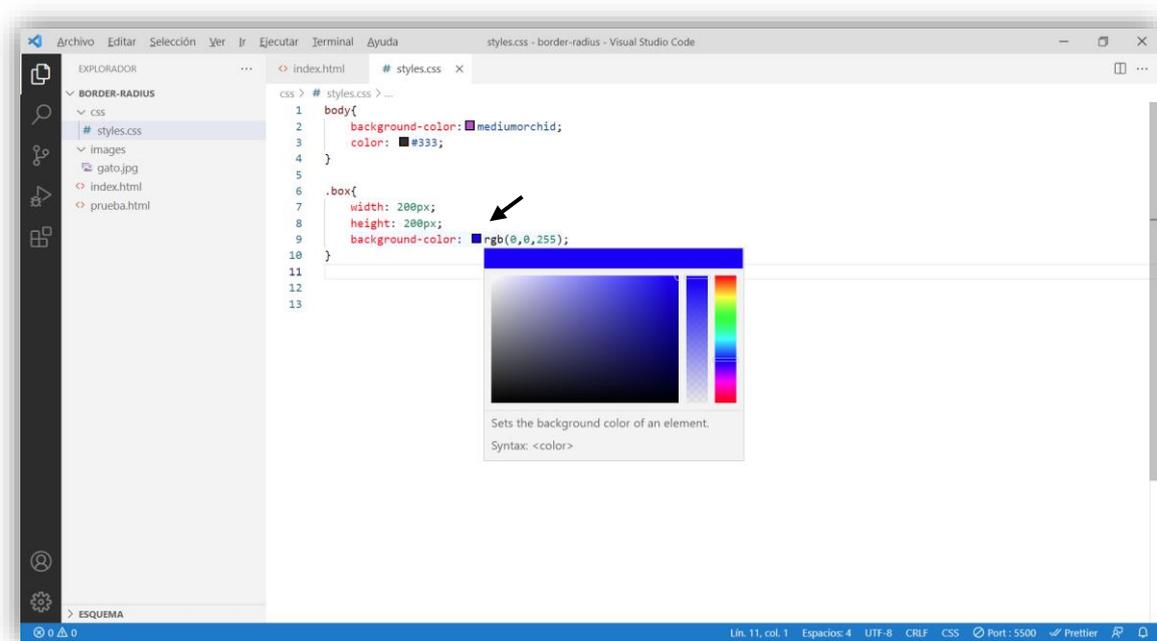
```
.box{
  width: 200px;
  height: 200px;
  background-color: ■rgb(0,0,0);
}
```

background-color: □rgb(255,255,255);

background-color: ■rgb(255,0,0);

background-color: ■rgb(0,255,0);

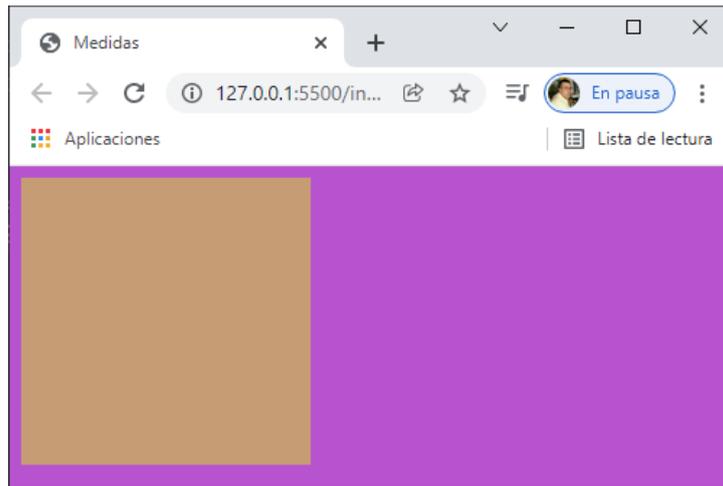
background-color: ■rgb(0,0,255);



Si nos colocamos en el color con el panel de colores podremos seleccionar, no hace falta aprenderse la combinación de colores.

background-color: ■rgba(214, 228, 17, 0.5);

En este ejemplo agregamos un cuarto valor que irá comprendido entre 0 y 1 que será el porcentaje de transparencia.



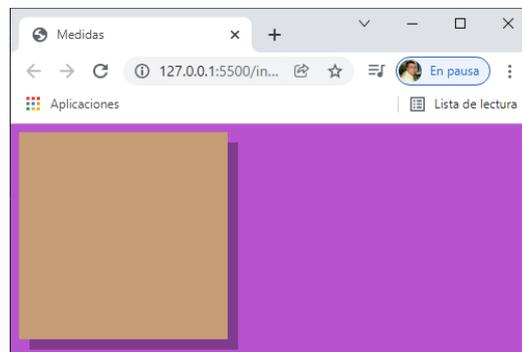
Ha aparecido una nueva sintaxis.

```
background-color: rgba(214 228 17 / 0.5);
```

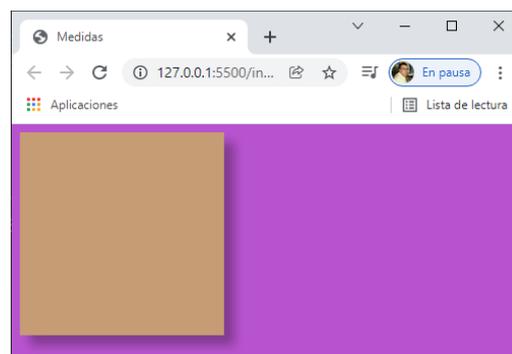
En este caso obtendríamos el mismo resultado.

```
.box{  
  width: 200px;  
  height: 200px;  
  background-color: rgba(214 228 17 / 0.5);  
  box-shadow: 10px 10px rgb(0 0 0 / 0.3);  
}
```

Este será el resultado:



```
box-shadow: 10px 10px 10px rgb(0 0 0 / 0.3);
```



¿Qué es el Modo de Color Hexadecimal?

Hexadecimal

La notación hexadecimal es la más común en desarrollo web, se basa en los mismos principios que el RGB pero escrito en notación hexadecimal.

El sistema hexadecimal es un sistema basado en 16 valores del 0 al 15.

Utiliza los dígitos del 0 al 9 y las letras de la A a la F.

Se compone de 16 caracteres en parejas ya que $16 * 16 = 256$

0 1 2 3 4 5 6 7 8 9 A B C D E F

0 1 2 3 4 5 6 7 8 9 A B C D E F

Para indicar que vamos a usar notación hexadecimal debemos usar el símbolo de hash/almojadilla # y a continuación usaremos esta notación de parejas.

#RRGGBB

Cuando una pareja usa el mismo valor se puede omitir el segundo valor, siempre que se hagan en las tres parejas.

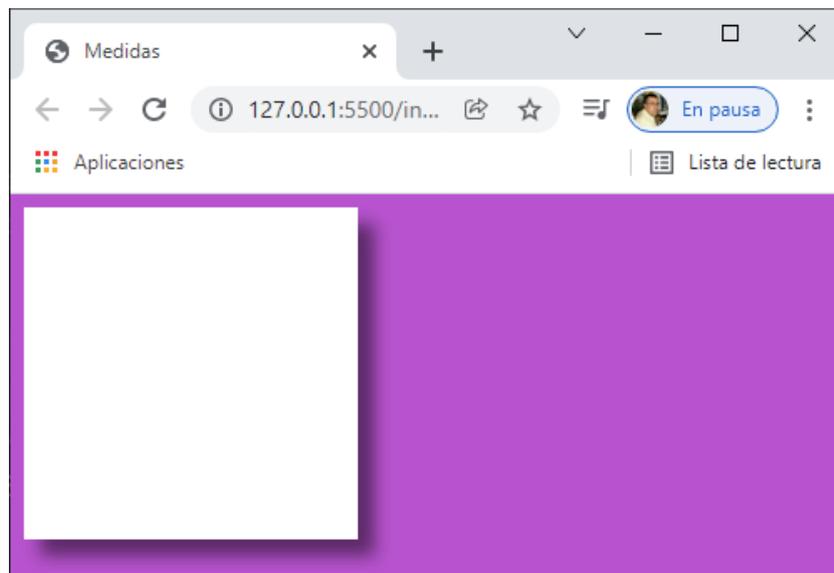
rgb(255,255,255) -> #FFFFFF -> #FFF

rgb(0,0,0) -> #000000 -> #000

También tenemos la opción de usar transparencias en hexadecimal añadiendo un cuarto canal que sigue las mismas normas.

rgb(0 0 0 / .5) -> #00000080 ~> #0007.

```
.box{
  width: 200px;
  height: 200px;
  background-color: #FFFFFF;
  box-shadow: 10px 10px 10px #00000080;
}
```



`background-color: #FF0000;`

`background-color: #00FF00;`

`background-color: #0000FF;`

```
.box{
  width: 200px;
  height: 200px;
  background-color: #FFF;
  box-shadow: 10px 10px 10px #0007;
}
```

Simplificado por parejas, el resultado será el mismo.

`background-color: #000;`

`background-color: #F00;`

`background-color: #0F0;`

`background-color: #00F;`

`background-color: #FFF;`

```
.box{
  width: 200px;
  height: 200px;
  background-color: #80808080;
  box-shadow: 10px 10px 10px #0007;
}
```

En este ejemplo agregamos un color gris con transparencia al 50% y una sombra negra con una transparencia al 50% como son tres valores emparejados los podemos simplificar.

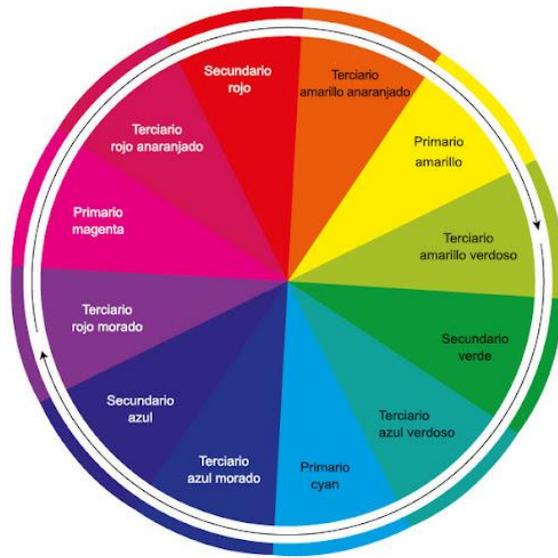
¿Qué es el modo de color HSL y para qué funciona?

HSL / HSLA

El modo de color SHL es el más intuitivo para los humanos. Su nombre viene de las siglas Hue (tono) Saturation (saturación) y Lightness (luminancia).

El primer valor es el ángulo en el círculo cromático donde 0 y 360 será el color rojo. Hay ciertos valores que os pueden servir como referencia para relacionar mejor los colores con sus ángulos.

0 -> rojo
 60 -> amarillo
 120 -> verde
 180 -> cyan
 240 -> azul
 300 -> magenta



El segundo valor es la saturación o intensidad de color.

0% -> gris
100% -> color puro.



El tercer valor es la luminosidad del color

0% -> negro, nada de luz.
100% -> blanco. Luz máxima.

Nota: Es importante que aunque el valor de saturación o luminancia sea 0 hay que poner 0% si no, no funcional

El modo HSL también admite transparencias y su sintaxis es la misma que la del modo RGB. Tenemos dos sintaxis disponibles:

hsla(h,s,l,a)
hsl(h s l / a)

#YOAPRENDOCONDORIAN

En ambos casos el valor alpha va de 0 a 1.

Código html:

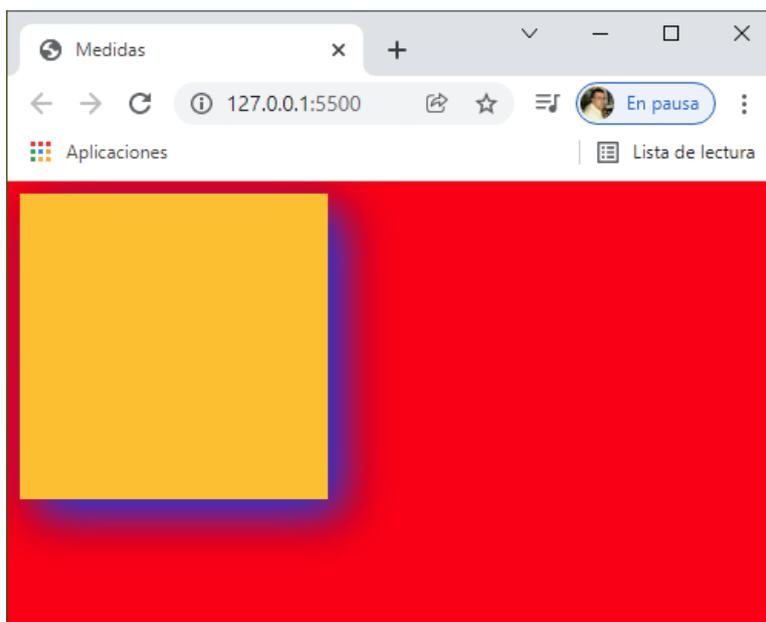
```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Medidas</title>
</head>
<body>
<div class="main">
  <div class="box">
  </div>
</div>
</body>
</html>
```

Código css:

```
body{
  background-color: hsl(0, 100%, 50%);
  color: #333;
}

.box{
  width: 200px;
  height: 200px;
  background-color: hsl(45,100%,50%);
  box-shadow: 10px 10px 30px hsl(225, 100%,50%);
}
```

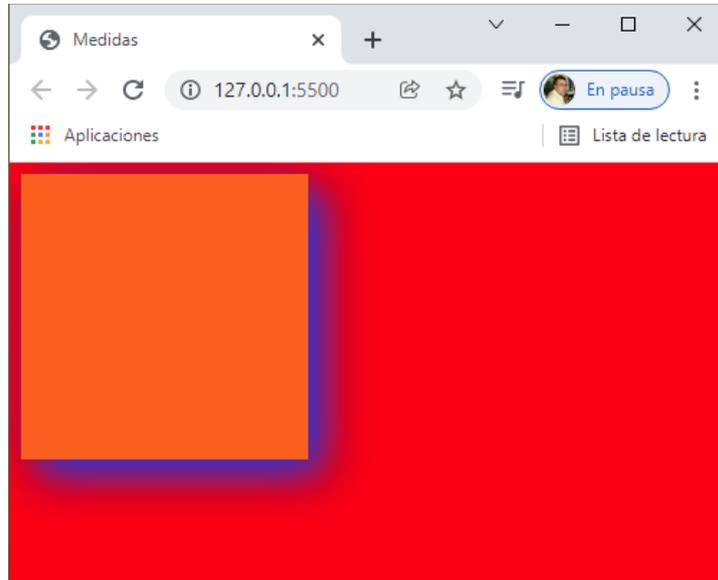
Este será el resultado:



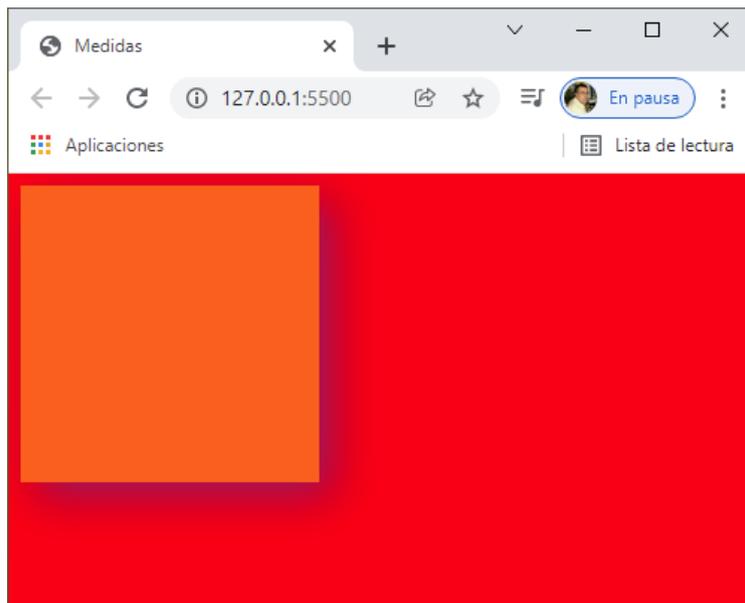
#YOAPRENDOCONDORIAN

```
.box{  
  width: 200px;  
  height: 200px;  
  background-color: hsl(45,100%,50%, .5);  
  box-shadow: 10px 10px 30px hsl(225, 100%,50%);  
}
```

Transparencia.



```
.box{  
  width: 200px;  
  height: 200px;  
  background-color: hsl(45,100%,50%, .5);  
  box-shadow: 10px 10px 30px hsl(225 100% 50% / .4);  
}
```



#YOAPRENDOCONDORIAN

Para consultar colores:

<https://brandcolors.net/>

Nos dicen la paleta de color que utilizan las empresas.

<https://htmlcolorcodes.com/es/tabla-de-colores/>

Nos ayuda a compaginar colores.

<https://colorhunt.co/>

Nos muestra paletas completas que podemos elegir.

<https://color.adobe.com/es/create/color-wheel>

Uno de lo más completo para crear nuestra paleta de colores.

Existen otros cuatro modos de color en los que está trabajando que serían CMYK (Cyan, Magenta, Yellow y Black), HWB (Hue, Whiteness, Blackness), lab (Lightness, distance a, distance b) y lch (Lightness, chroma, hue), todos están en el borrador del módulo de color del nivel 4 pero aún no es estándar.

¿Qué es la tipografía, ¿Cómo poner fuentes en CSS?

La tipografía es la ciencia que estudia el diseño y uso de los tipos de letra.

Las fuentes son tipos específicos de letra.

Los glifos son cada uno de los dibujos que representan cada símbolo de la fuente, incluidos los espacios.

Los estilos de fuentes son las modificaciones que se hacen a las fuentes, como el grosor o la inclinación.

Familia tipográfica, son las diferentes fuentes que pertenecen a una misma fuente base.

Cuando usamos una propiedad text-* estamos afectando a un bloque completo.

Cuando utilizamos una propiedad Font-* afectamos a la fuente del texto.

Fuentes predeterminadas, son las fuentes que se consideran seguras ya que son genéricas y no establecen una familia tipográfica, si no un estilo de fuente.

Serif: Fuentes que tienen serifas (adornos en los caracteres)

Sans-serif: Fuentes sin serifas.

Monospace: Fuentes que tienen el mismo espaciado entre todos sus caracteres.

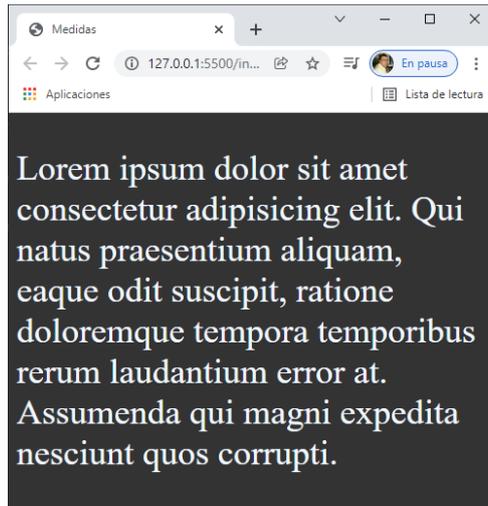
Cursive: Fuentes que pretenden emular escritura a mano, con ligaduras y trazos fluidos.

Fantasy: Fuentes que suelen ser decorativas.

```
body{
  background-color: ■ #333;
  color: □ aliceblue;
  font-size: 36px;
  font-family: serif;
}
```

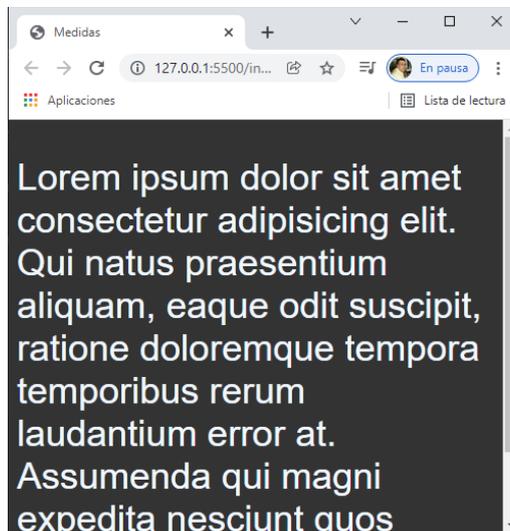
#YOAPRENDOCONDORIAN

La fuente serif tienen adornos.



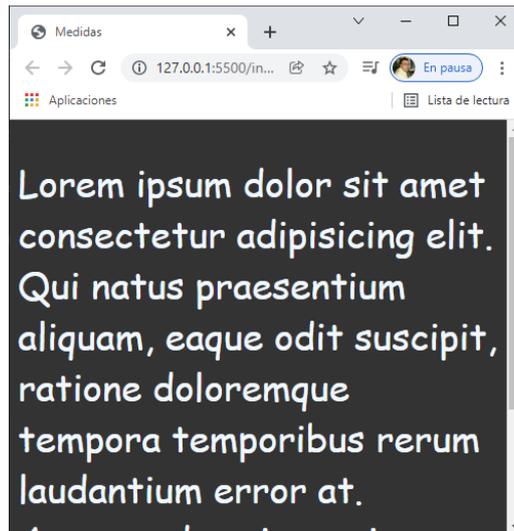
```
body{
  background-color: #333;
  color: aliceblue;
  font-size: 36px;
  font-family: serif;
}
```

La fuente sans-serif no tiene adornos.



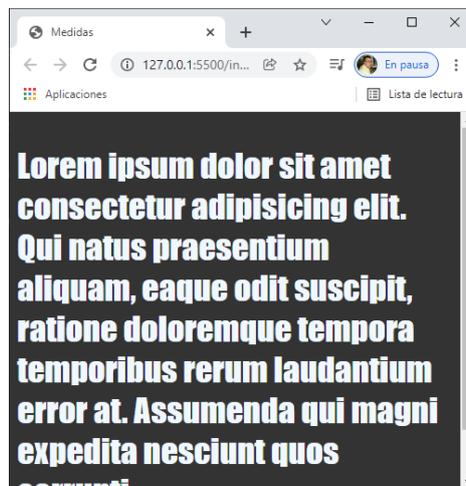
No abusar con las fuentes con serifas, cuestan más de leer.

```
body{
  background-color: #333;
  color: aliceblue;
  font-size: 36px;
  font-family: cursive;
}
```



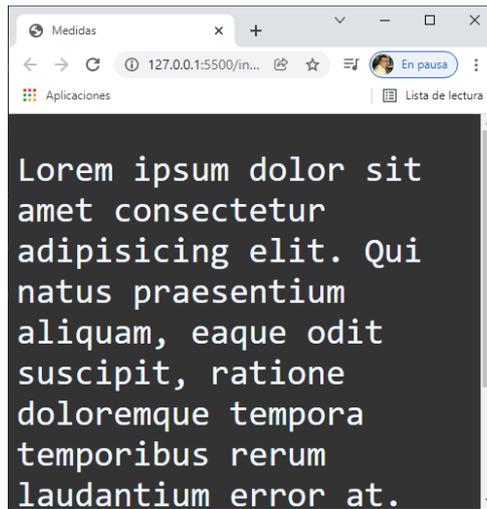
Consiste en emular la escritura a mano.

```
body{  
  background-color: #333;  
  color: aliceblue;  
  font-size: 36px;  
  font-family: fantasy;  
}
```



Fuentes decorativas.

```
body{  
  background-color: #333;  
  color: aliceblue;  
  font-size: 36px;  
  font-family: monospace;  
}
```



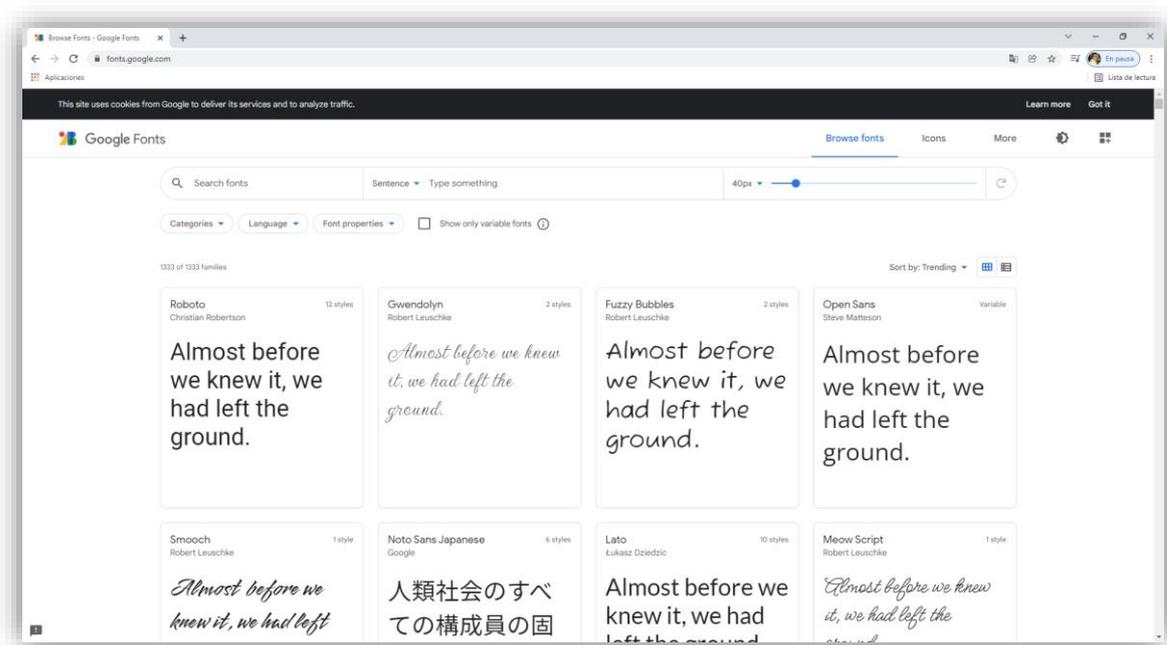
Aprender a cómo usar Google Fonts con CSS

Fuentes desde un servidor remoto, con fuentes que se encuentran en un servidor y que nosotros descargamos para su uso, la más famosa es Google Fonts.

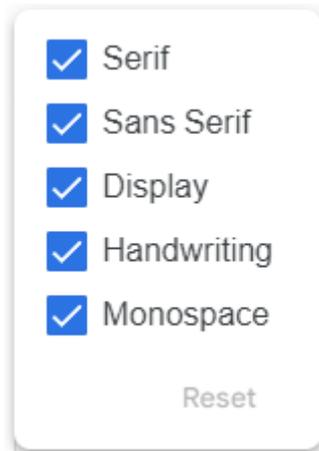
Siempre que utilicemos fuentes no genéricas, debemos poner fuentes genéricas de fallback (respaldo).

Vamos a acceder al siguiente enlace:

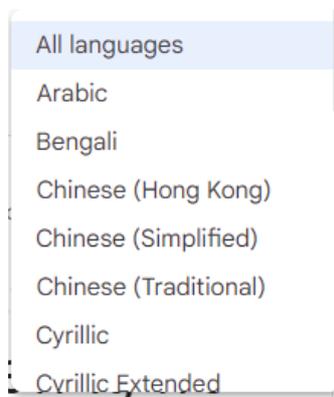
<https://fonts.google.com/>



Lo podemos consultar por Categorías.



Lenguaje:



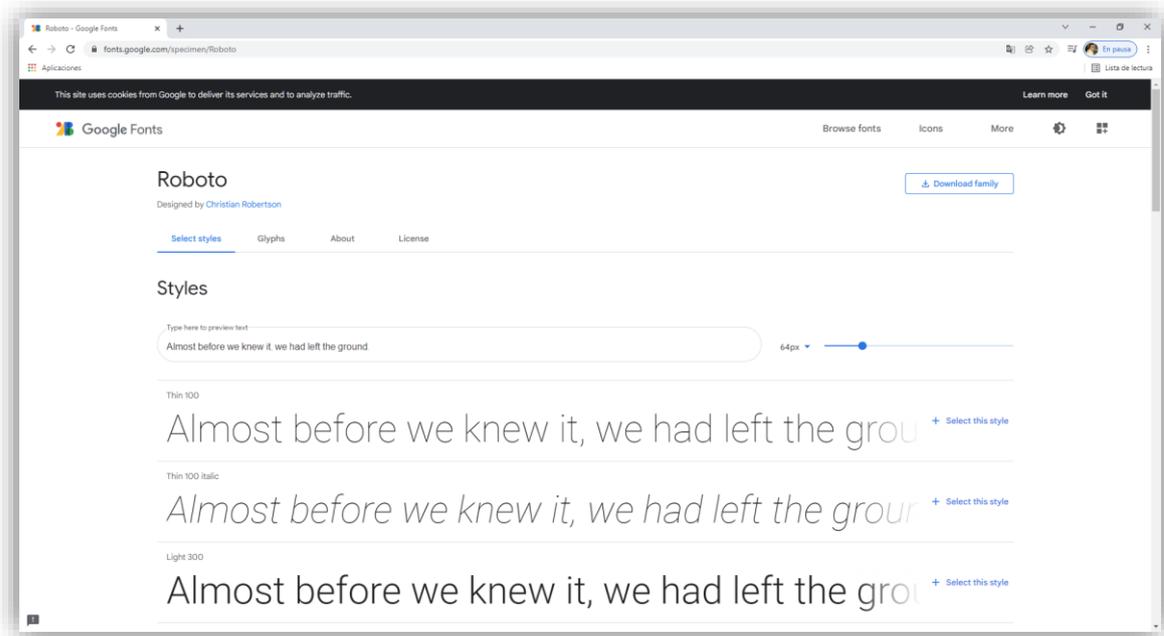
Propiedades de la fuente:



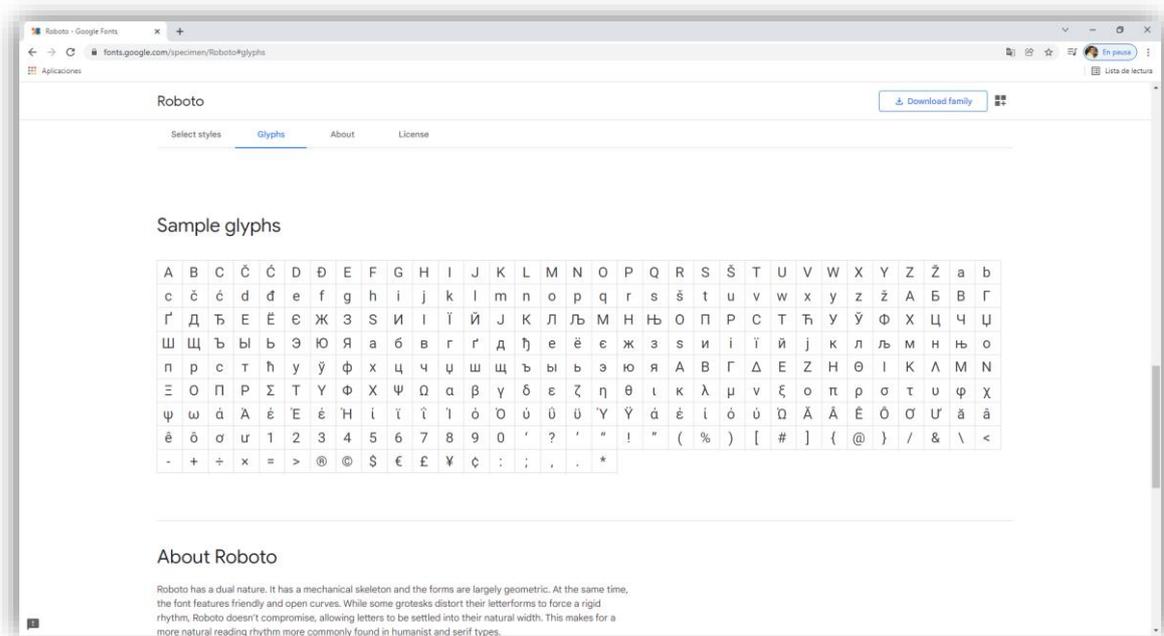
Nosotros vamos a elegir:

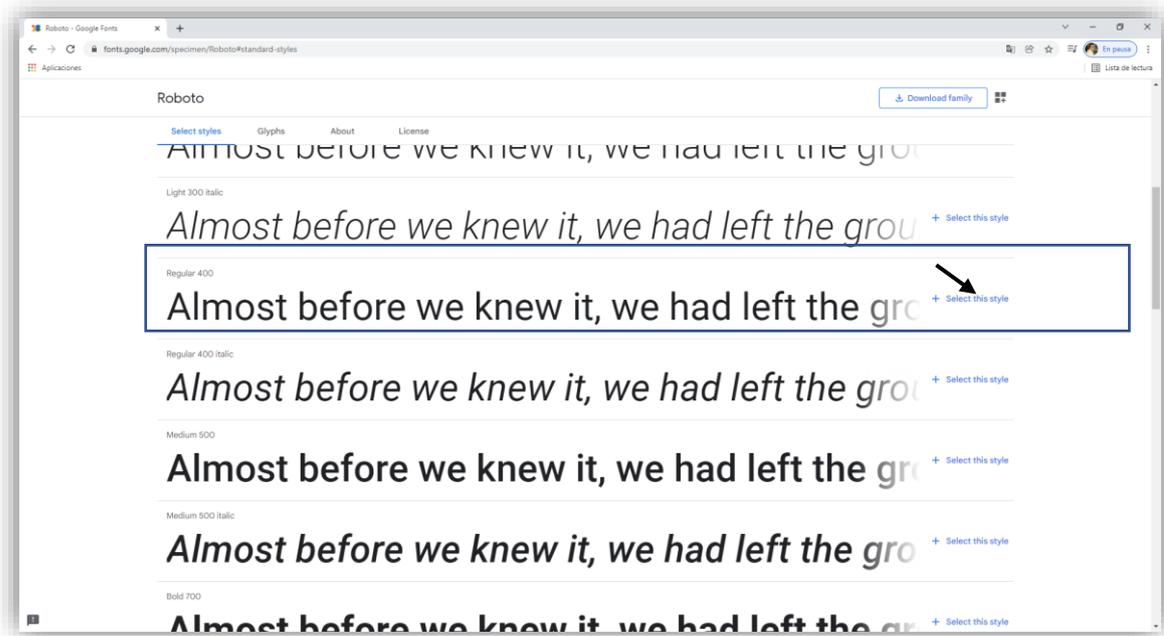
Roboto
Christian Robertson 12 styles

Almost before
we knew it, we
had left the
ground.

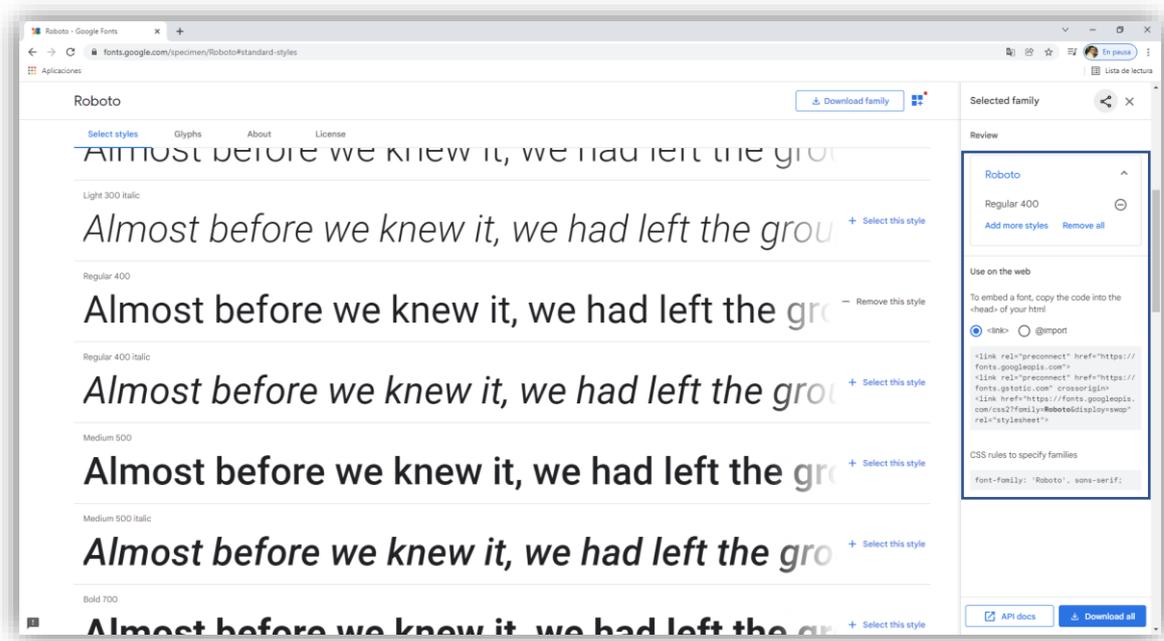


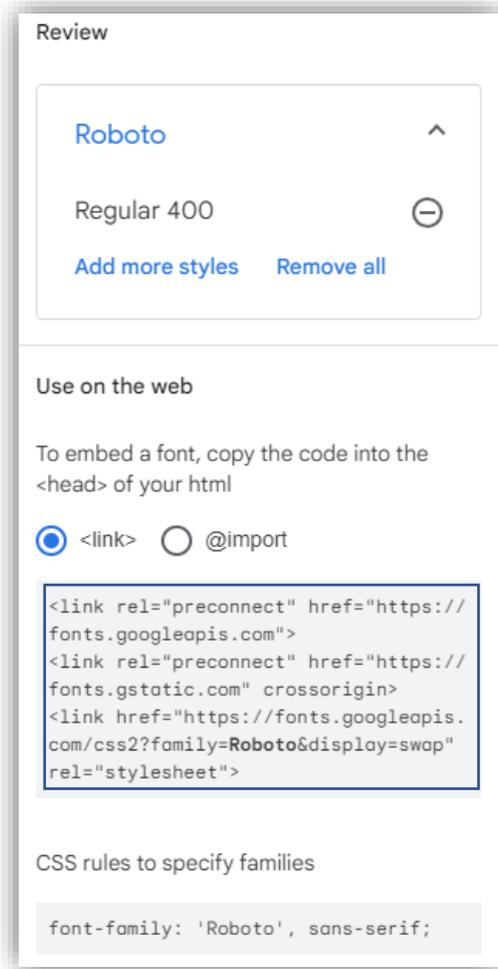
En la parte inferior veremos los Glyphs.





Vamos a elegir la regular 400, hacemos clic donde pone seleccionar estilos.



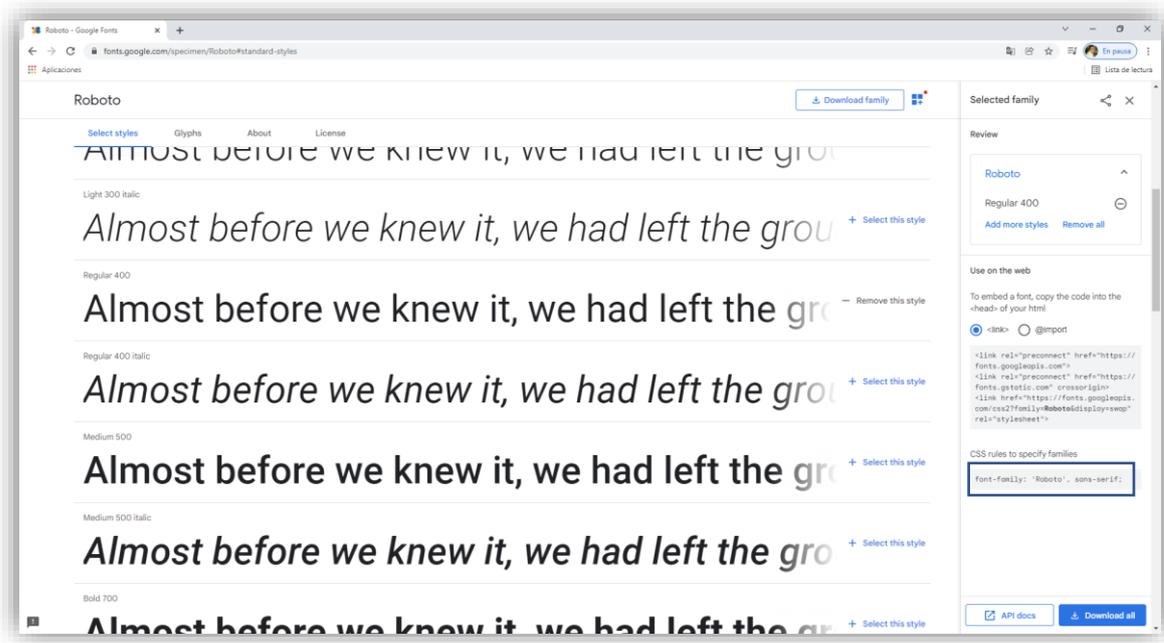


Copiamos el texto que está en el recuadro.

En nuestro documento html.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap" rel="stylesheet">
  <link rel="stylesheet" href="css/styles.css">
  <title>Medidas</title>
</head>
<body>
<p>
  Lorem ipsum dolor sit amet consectetur adipisicing elit. Qui natus praesentium aliquam, e
  doloremque tempora temporibus rerum laudantium error at. Assumenda qui magni expedita nes
</p>
</body>
</html>
```

Lo pegamos por encima de nuestros estilos.



CSS rules to specify families

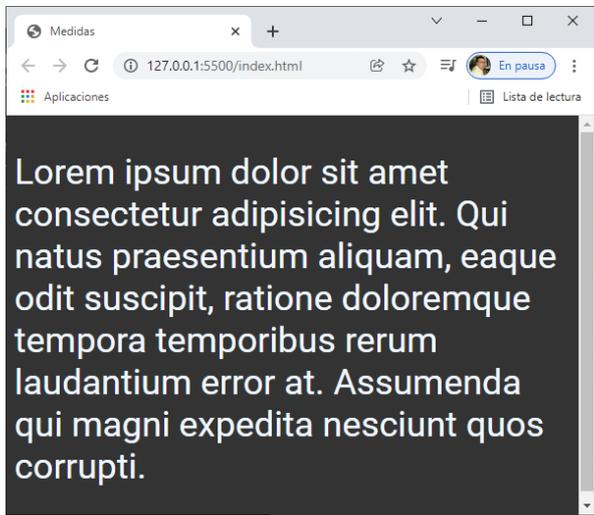
```
font-family: 'Roboto', sans-serif;
```

Lo seleccionamos y lo copiamos.

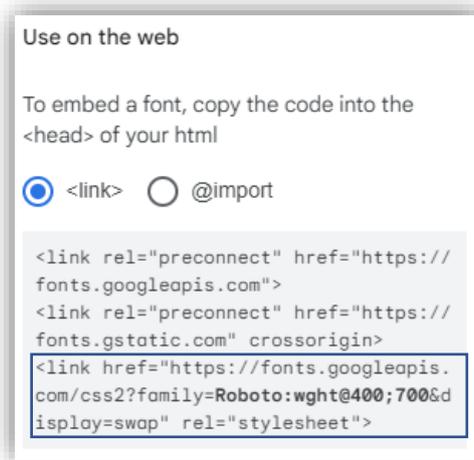
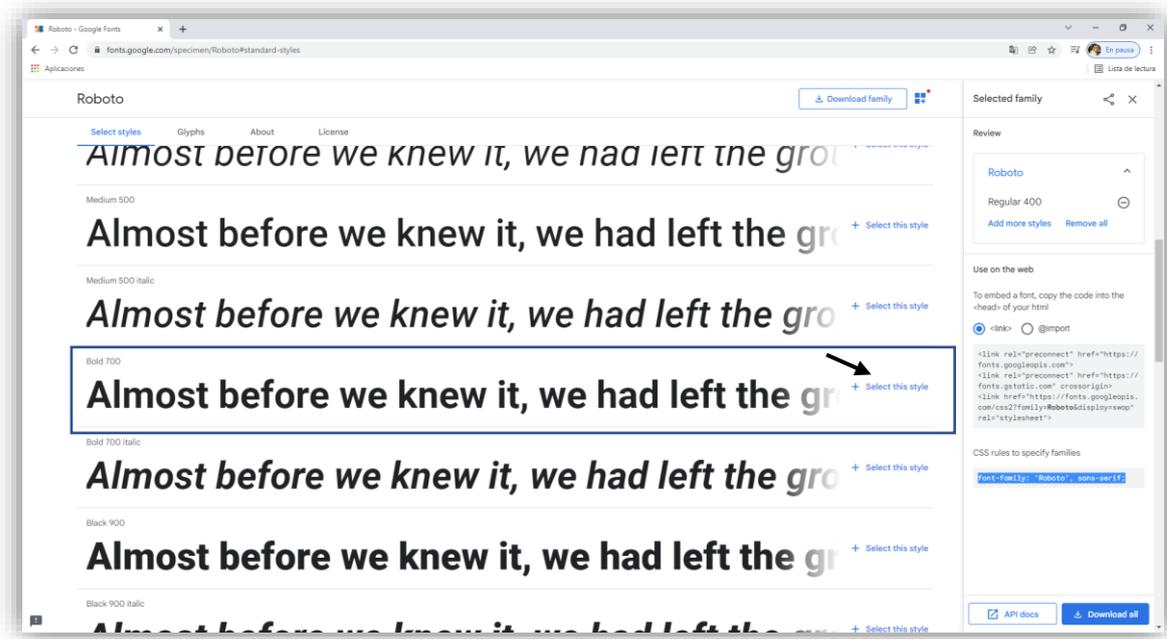
En el código css:

```
body{  
  background-color: ■ #333;  
  color: □ aliceblue;  
  font-size: 36px;  
  font-family: 'Roboto', sans-serif; ←  
}
```

Lo agregamos, este será el resultado:



Después de Roboto encontramos sans-serif es una fuente de respaldo.
Si agregamos otro estilo.



Ya tiene las dos.

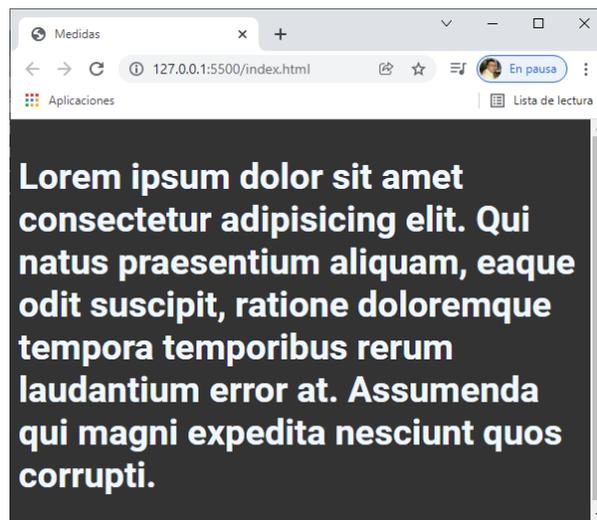
```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap" rel="stylesheet">
  <link rel="stylesheet" href="css/styles.css">
  <title>Medidas</title>
</head>
<body>
<p>
  Lorem ipsum dolor sit amet consectetur adipisicing elit. Qui natus praesentium aliquam, eaque odit susci-
  doloremque tempora temporibus rerum laudantium error at. Assumenda qui magni expedita nesciunt quos cori
</p>
</body>
</html>
```

#YOAPRENDOCONDORIAN

Cambiamos el último link de fonts.foofleapis.com.

```
body{
  background-color: #333;
  color: aliceblue;
  font-size: 36px;
  font-family: 'Roboto', sans-serif;
  font-weight: 700; ←
}
```

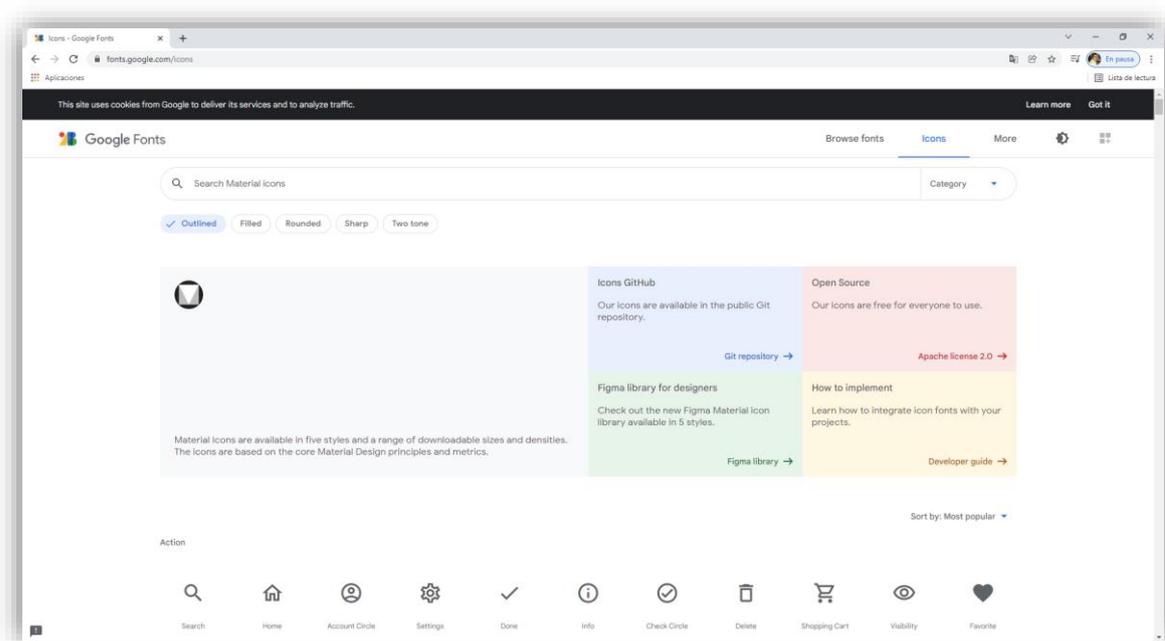
Como ahora tenemos el grosor de 400 y el de 700 es la forma de especificar.



Google Icons ¿Cómo usar librerías de ICONOS?

Desde este enlace:

<https://fonts.google.com/icons>



#YOAPRENDOCONDORIAN

Material Icons:

<https://fonts.googleapis.com/css?family=Material+Icons>

Material Icons Outlined:

<https://fonts.googleapis.com/css?family=Material+Icons+Outlined>

Material Icons Two Tone:

<https://fonts.googleapis.com/css?family=Material+Icons+Two+Tone>

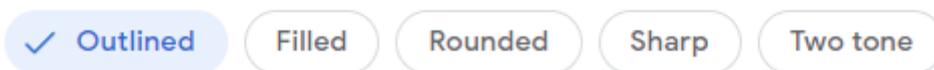
Material Icons Rounded:

<https://fonts.googleapis.com/css?family=Material+Icons+Round>

Material Icons Sharp

<https://fonts.googleapis.com/css?family=Material+Icons+Sharp>

Como vamos a insertar un icono del grupo Outlined.

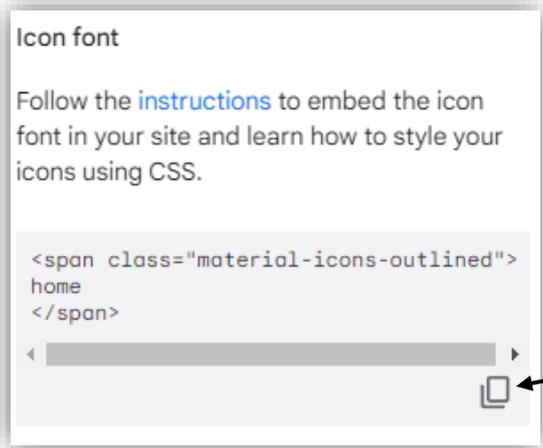
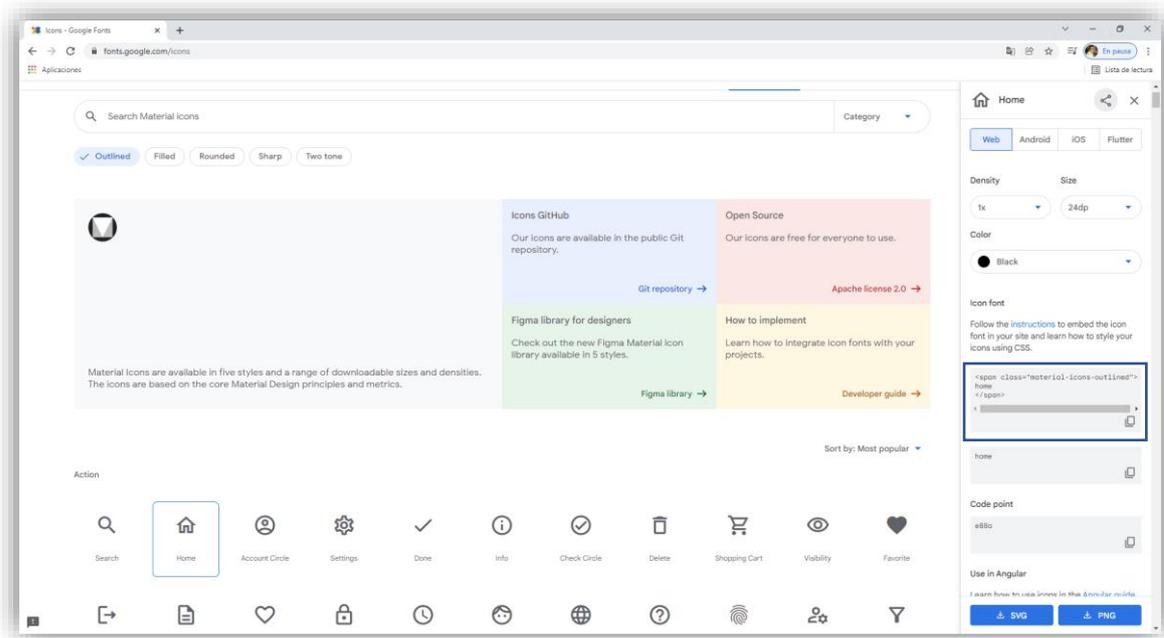


Copiaremos el correspondiente url.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Material+Icons+Outlined">
  <link rel="stylesheet" href="css/styles.css">
  <title>Medidas</title>
</head>
<body>
  <span class="material-icons-outlined">
    home
  </span>
<p>
  Lorem ipsum dolor sit amet consectetur adipiscing elit. Qui natus praesentium aliquam, eaque od.
</p>
</body>
</html>
```

En nuestro documento html agregamos un link una línea antes de los estilos css y en href pegamos la correspondiente url.

Ahora seleccionaremos el icono que queremos agregar.



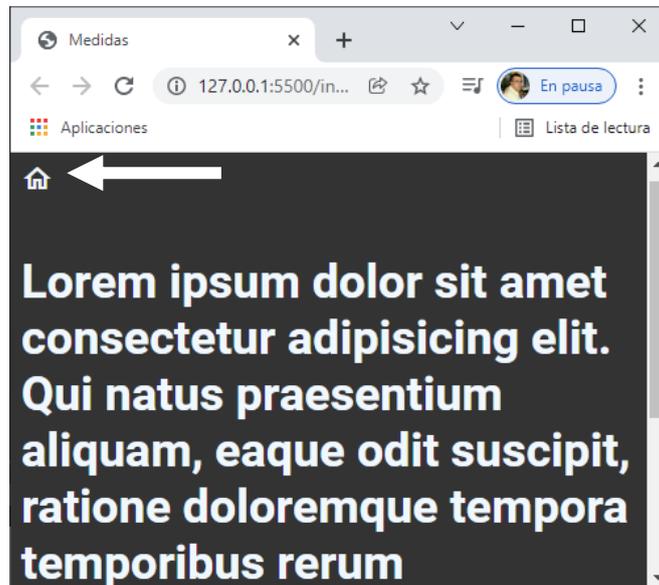
Copiamos el código:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Material+Icons+Outlined">
  <link rel="stylesheet" href="css/styles.css">
  <title>Medidas</title>
</head>
<body>
  <span class="material-icons-outlined">
    home
  </span>
<p>
  Lorem ipsum dolor sit amet consectetur adipisicing elit. Qui natus praesentium aliquam, eaque o
</p>
</body>
</html>
```

#YOAPRENDOCONDORIAN

Y pegamos el código.

Este será el resultado:

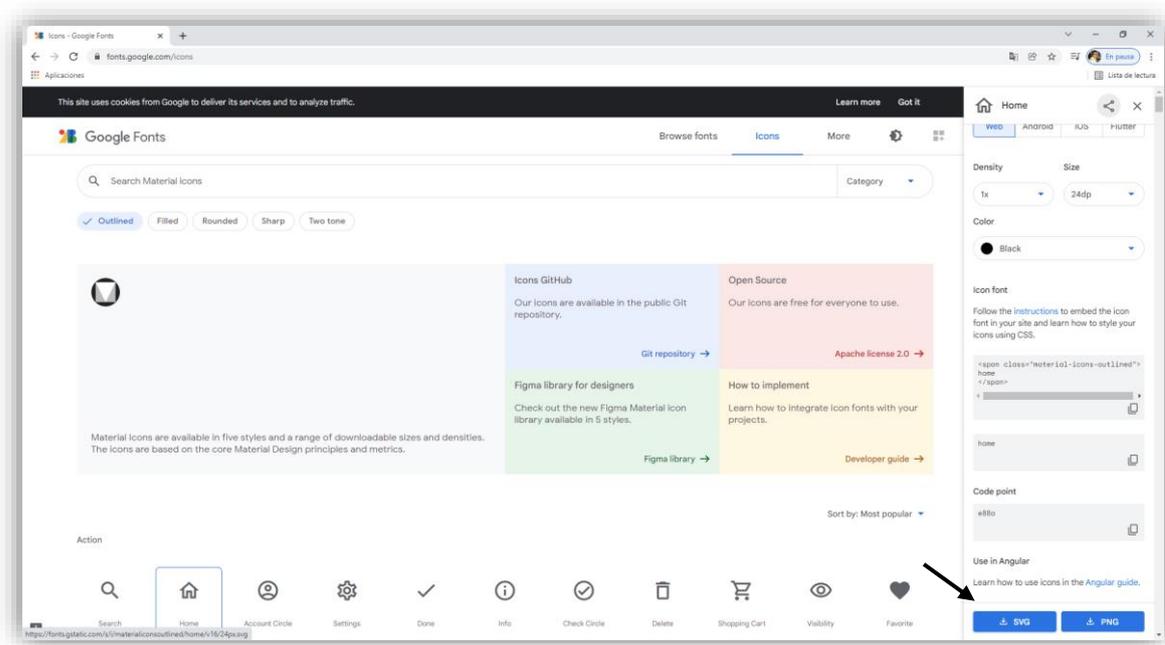


De este modo podemos ir agregando iconos en nuestra página web.

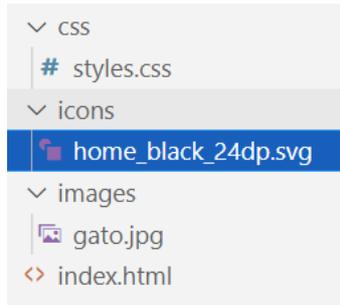
Nota:

Si vas a usar una gran cantidad de iconos en tu web compensa usar la librería remota.

Si sólo usarás unos pocos es mejor descargarlos en formato svg y utilizarlos como imágenes.



Y la guardamos en su correspondiente carpeta.



```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/styles.css">
  <title>Medidas</title>
</head>
<body>
  
  <p>
    Lorem ipsum dolor sit amet consectetur adipisicing elit. Qui natus prae
    doloremque tempora temporibus rerum laudantium error at. Assumenda qui
  </p>
</body>
</html>
```

Vamos a agregar una clase.

Abrimos la imagen vga

```
<svg xmlns="http://www.w3.org/2000/svg" height="96px" viewBox="0 0
24 24" width="96px" fill="#FFFFFF"><path d="M0 0h24v24H0V0z"
fill="none"/><path d="M12 5.6915 4.5V18h-2v-6H9v6H7v-7.8115-4.5M12
3L2 12h3v8h6v-6h2v6h6v-8h3L12 3z"/></svg>
```

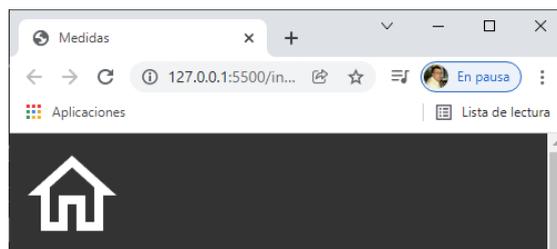
Cambiar los valores que están enmarcados:

height="96px"

width="96px"

fill="#FFFFFF"

Este será el resultado:



¿Cómo usar @FONT-FACE en CSS?

Formatos para fuentes

True Type Font (TTF) -> Fue el primer formato estándar.

Open True Type Font (OTF) -> Facilita o implementa la creación de las curvas.

Web Open Font Format (WOFF) (WOFF2) -> Es un formato mucho más ligero y mejora el rendimiento. Funciona a través de metadatos. WOFF tiene soporte total y WOFF2 no funciona en Internet Explorer.

Scalable Vector Graphics (SVG) -> Para efectos y animaciones sobre el texto, no es recomendable usarlo en el texto genérico de la web, tiene muchos problemas.

Embedded Open Type (EOT) -> Primer formato digital propuesto para sitios WEB y es comprimido. Se quedó como una propuesta pero no pasó a ser un estándar.

Regla @font-face

Esta es la forma de utilizar fuentes que tenemos descargadas y que no traemos de un servidor remoto, cuando usamos Google fonts, esto es lo que está pasando por detrás.

La sintaxis es:

```
@font-face {
```

Font-family: Nombre de la familia, se lo asignamos nosotros;

src: aquí pondremos dos tipos de valores.

Local() -> pondremos el nombre de la fuente para buscarla en el ordenador del usuario, de esa forma podremos evitar descargarla.

url() -> Si no encontramos esa fuente procederemos a descargarla.

Existe otro valor, menos usado que es format.

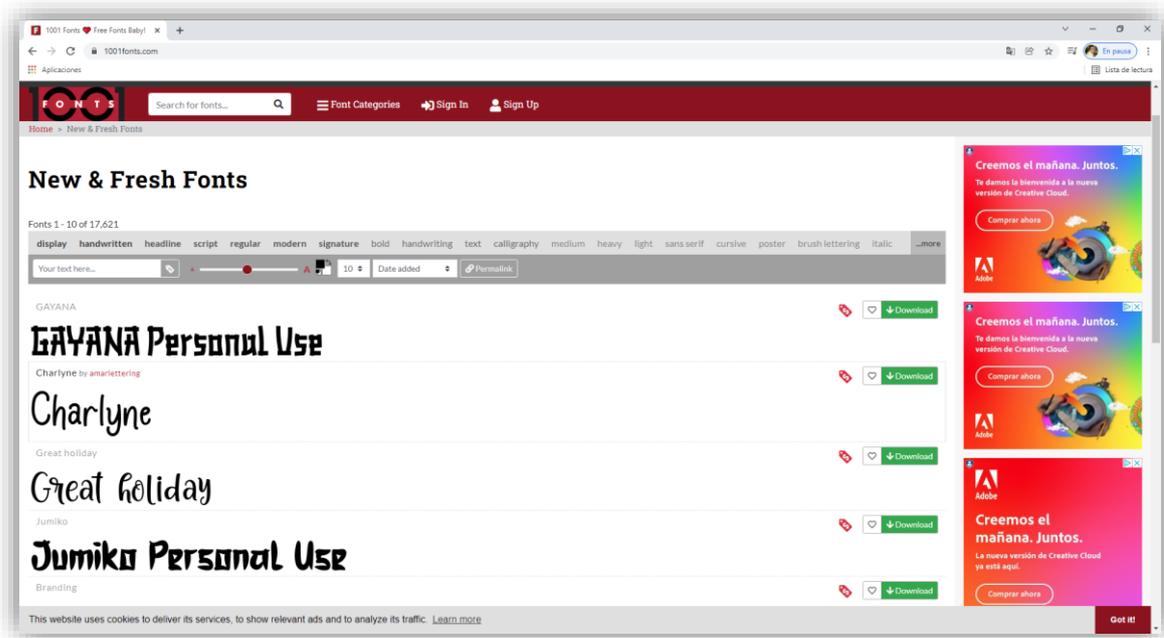
Format() -> Este valor lo pasaremos de forma opcional si queremos ser más específicos con la fuente que vamos a buscar o descargar, si no lo ponemos cogerá el primer formato soportado, si lo ponemos sólo usará el que hemos establecido, y si no lo encuentra no hará nada.

```
}
```

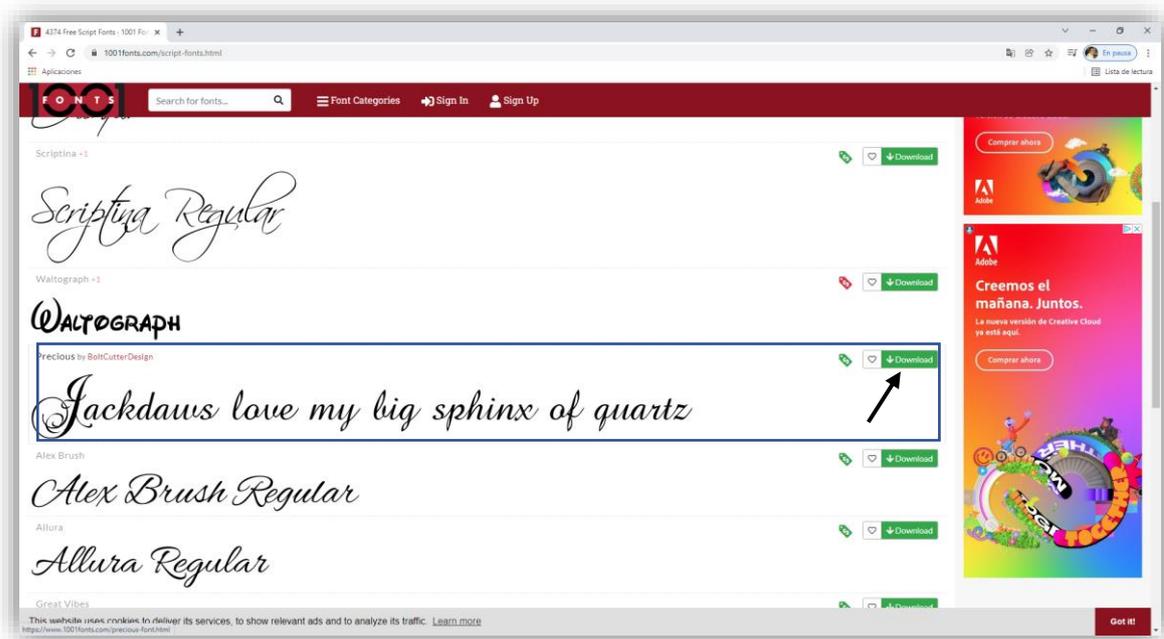
Podemos añadirle más valores como font-style, font-weight y font-display (experimental), pero esos son opcionales.

Páginas que utilizo para descargar fuentes:

<https://www.1001fonts.com/>



Al lado de cada descarga encontramos una etiqueta que puede ser roja, naranja o verde, las de color verde las puedes descargar y utilizarlas en tus proyectos que después podrás publicar.



Vamos a descargar esta fuente.

```
✓ CSS
# styles.css
✓ font
A Precious.ttf
```

#YOAPRENDOCONDORIAN

En nuestro proyecto creamos una carpeta llamada font y dentro la fuente que hemos descargado.

Código en css:

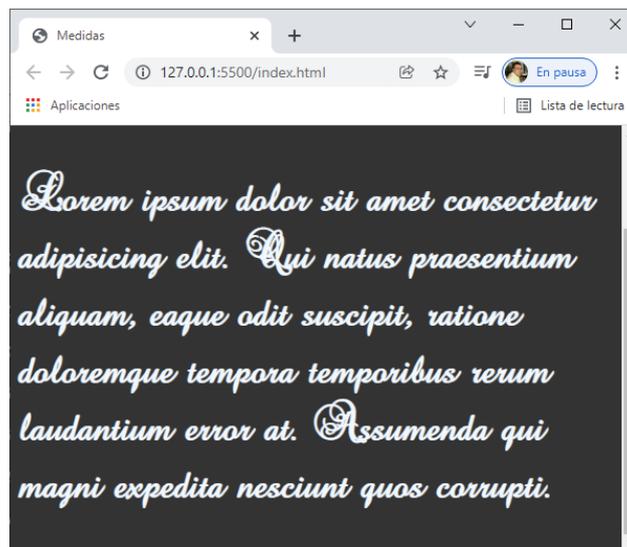
```
@font-face{
  font-family: 'precious' ;
  src: url(/font/Precious.ttf);
}

body{
  background-color: #333;
  color: aliceblue;
  font-size: 36px;
  font-family: 'Roboto', sans-serif;
  font-weight: 700;
  font-family: 'precious';
}
```

Puede ir cualquier nombre.

Con el mismo nombre lo llamaremos en font-family.

Este será el resultado:

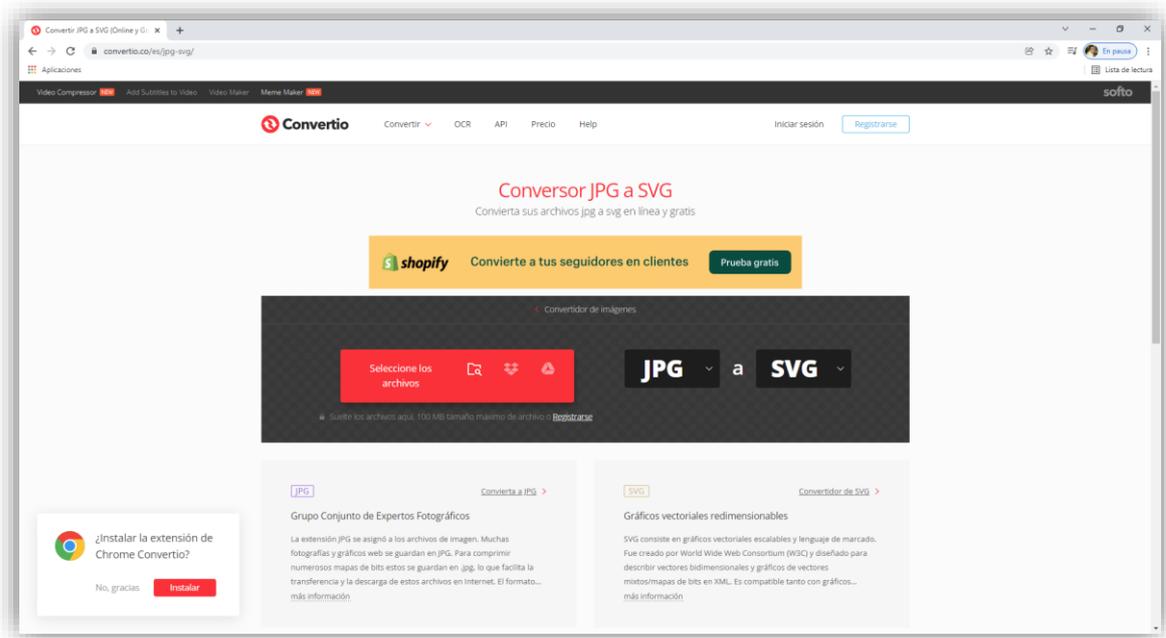


```
@font-face{
  font-family: 'precious' ;
  src: local("Great Vibes"), url(/font/Precious.ttf);
}
```

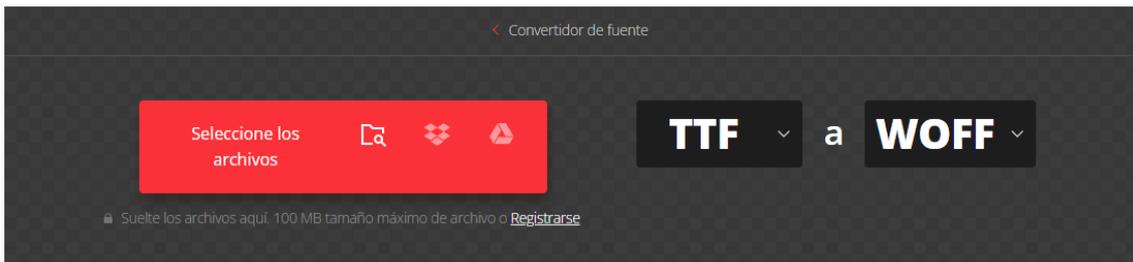
Si el usuario que visite la página tiene dicha fuente en su ordenador la usará de lo contrario descargará la fuente.

Otro enlace:

<https://convertio.co/es/jpg-svg/>



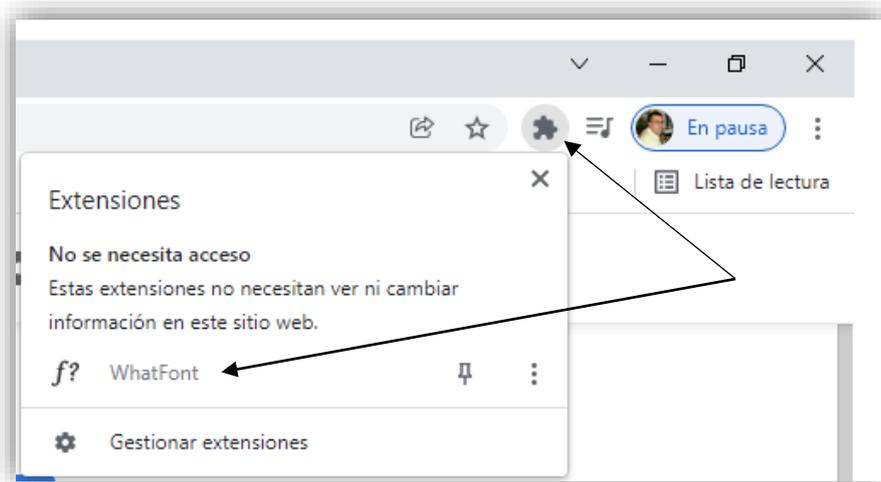
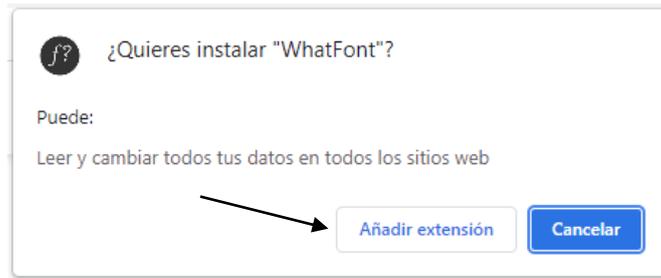
Puedes convertir fuentes a formato WOFF.



Además existe una extensión WhatFont para detectar los tipos de letras que se utilizan en la páginas web.

<https://chrome.google.com/webstore/detail/whatfont/jabopobgcpjmedljbcaablplmfcoqm?hl=ES>





Colocando encima de cualquier fuente te dirá su nombre.

¿Cómo decorar textos en CSS?

Transformaciones de texto:

Text-transform: Esta propiedad nos permite hacer transformaciones estéticas en nuestro texto.

Admite los siguientes valores:

- none: No se aplica ninguna transformación.
- uppercase: El texto se pondrá en mayúsculas.
- lowercase: El texto se pondrá en minúsculas.
- capitalize: Transforma la primera letra de cada palabra en mayúsculas.

Adornos en el texto:

Text-decoration: Esta propiedad nos permite controlar la línea que podemos poner en los textos.

Admite los siguientes valores:

- none: Elimina cualquier línea.
- underline: Coloca una línea bajo el texto.
- overline: Coloca una línea sobre el texto.
- line-through: Coloca una línea tachando el texto.

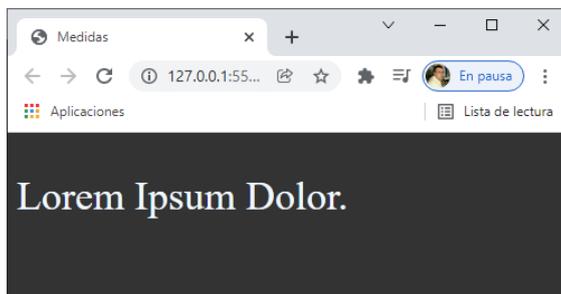
#YOAPRENDOCONDORIAN

Text-shadow: funciona exactamente igual que box-shadow, con la única diferencia de que no admite valor inset ni la expansión de la sombra.

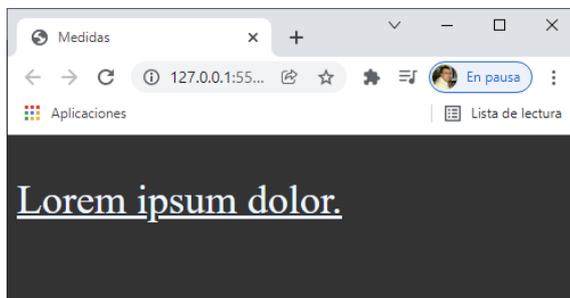
Los valores son:

- offsetX: desplazamiento horizontal.
- offsetY: desplazamiento vertical.
- blur: desenfoque.
- color: color, por defecto coge el del texto.

```
.texto{
  text-transform: capitalize;
}
```



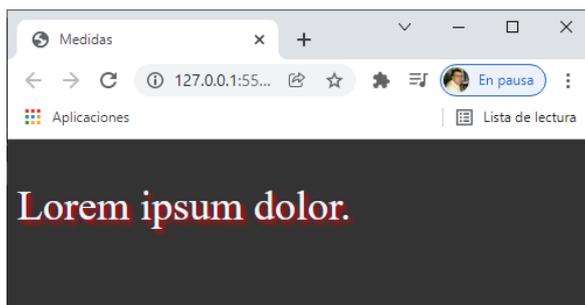
```
.texto{
  text-decoration: underline;
}
```



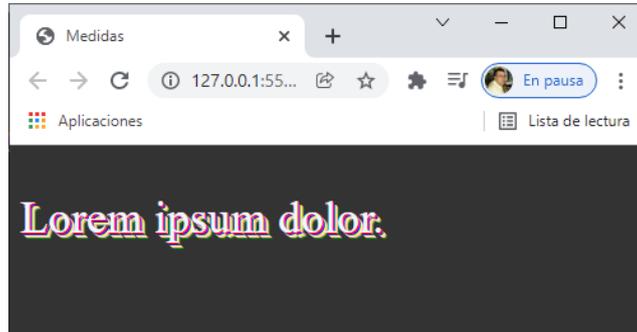
A la hora de recuperar el texto es el que está en el documento html.

```
.texto{
  Text-shadow: 3px 3px 5px red;
}
```

La sombra se desplaza 3px derecha 3px abajo desenfoque 5px color rojo.



```
.texto{  
  Text-shadow: 1px 1px red, 2px 2px blue, 3px 3px yellow, 4px 4px pink, 5px 5px green;  
}
```



¿Cómo dar espacio entre las letras en CSS?

Espaciado en fuentes y textos.

letter-spacing: Establece la separación entre caracteres, se puede establecer en cualquier medida válida para CSS.

word-spacing: Establece la separación entre palabras, se puede establecer en cualquier medida válida para CSS.

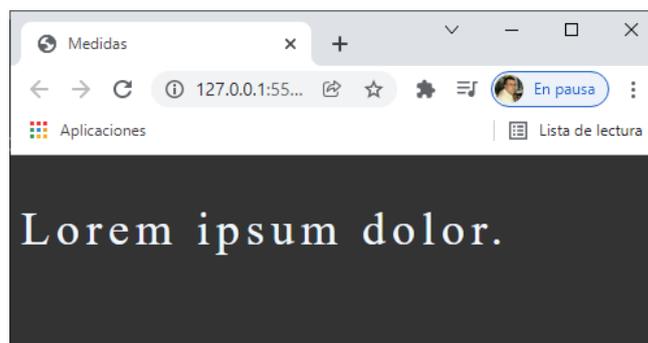
line-height: Establece el alto de cada línea, se puede dar una medida concreta o se puede poner un número que se multiplicará por el font size del texto al que se lo estamos aplicando.

Orientación del texto.

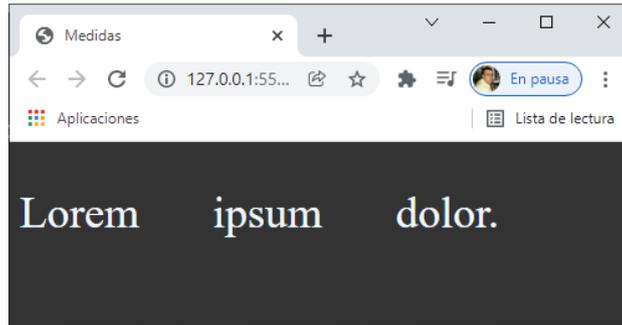
writing-mode: Establece la dirección del texto.

```
.texto{  
  letter-spacing: 5px;  
}
```

Con valores negativos se juntan los caracteres.

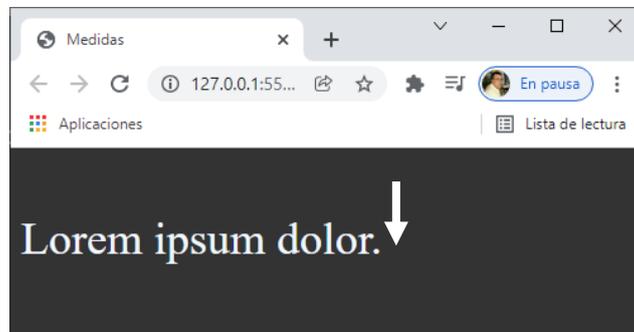


```
.texto{  
  word-spacing: 50px;  
}
```



```
.texto{  
  line-height: 2;  
}
```

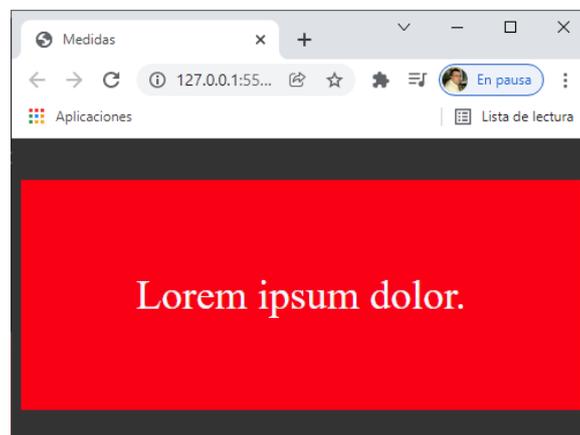
Al introducir un número sin px hacemos que este valor se multiplique por el tamaño de la fuente, admite otras unidades de medidas px, rem, etc.



Un truco:

```
.contenedor{  
  background-color: red;  
  height:200px;  
  text-align: center;  
}  
  
.texto{  
  line-height: 200px;  
}
```

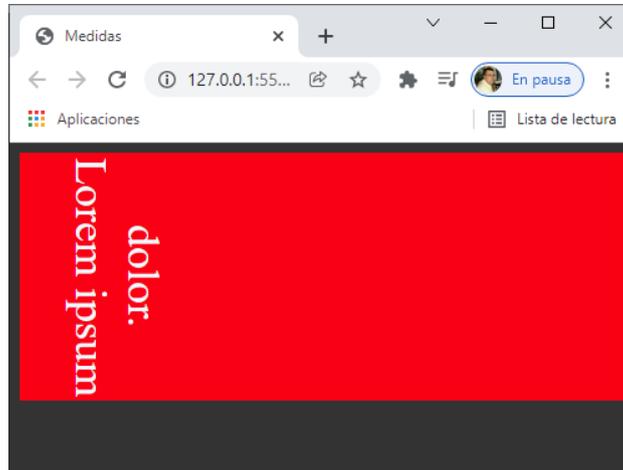
Podemos centrar un texto verticalmente.



#YOAPRENDOCONDORIAN

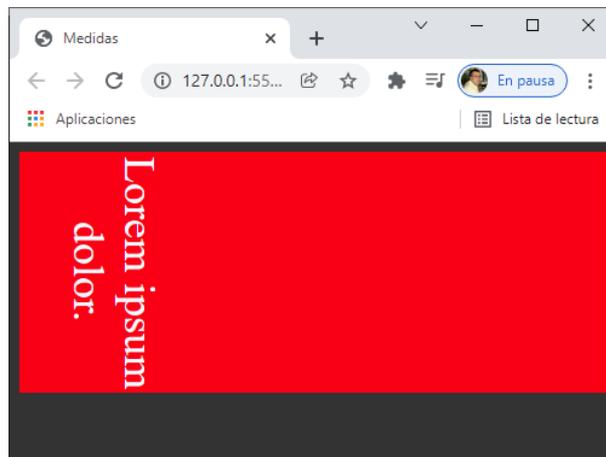
```
.texto{
  writing-mode:vertical-lr
}
```

De left a right.



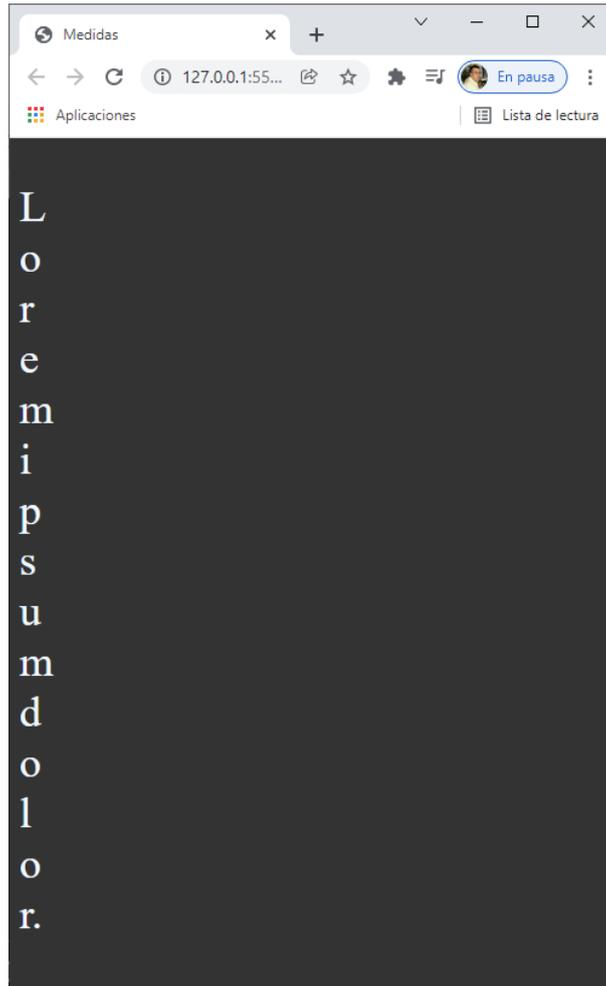
```
.texto{
  writing-mode:vertical-rl
}
```

De right a left



```
.texto{
  word-break: break-all;
  width: 20px;
}
```

Para que pueda romper las palabras.



Contenido

Temario	1
Preguntas	2
Instalación de Visual Studio Code	3
Lo podrás descargar desde: https://code.visualstudio.com/	3
Herramientas para Visual Studio Code	5
Vocabulario web	7
¿Qué es HTML?	8
Historia de HTML.....	8
Sintaxis de HTML.....	9
Estructura de un sitio web	13
Creando un servidor de desarrollo.....	15
Títulos y párrafos.....	18
Header, Main y Footer	21
Section vs article I.....	23
Section vs article II.....	23
Section vs article III.....	26
Aside.....	27
Elementos de bloque y elementos de línea	27
Práctica de sección de contenido.....	28
Elementos de línea	34
Elementos de línea II	38
Introducción a los atributos	41
Atributos ya utilizados.....	41
Atributos globales	41
Introducción a enlaces	46
Rutas absolutas y relativas.....	48
Atributos de los enlaces	50
Navegación con anclas	52
Introducción a las listas.....	58
Listas desordenadas	58
Listas ordenadas.....	59
Listas de definición.....	60
Listas anidadas y atributos	61
Introducción a las tablas	63
Estructura básica de una tabla	63

Estructura completa de una tabla	65
Atributos de las tablas.....	68
Seleccionar columnas.....	71
Cómo (NO) dar estilos a una tabla	75
Más etiquetas importantes de bloque.....	77
Más etiquetas de línea	80
Introducción a Formularios	82
Estructura básica de un formulario.....	82
Asociar input y label.....	82
button vs type button	83
Input para fechas.....	84
Inputs para móviles	86
Inputs extras.....	88
Input radio.....	89
Input checkbox	91
Elemento select básico.....	92
Elementos select avanzado	93
Datalist	94
Más elementos para formularios	95
Atributos para formularios.....	98
Envío GET vs POST	100
¿Qué es el contenido embebido?	103
Imágenes de bits vs imágenes vectoriales	103
Imágenes de bits WEBP.....	104
Insertar imágenes en HTML	106
Device Pixel Ratio	111
Atributo srcset.....	113
Etiqueta Picture.....	115
Etiqueta Audio.....	116
Etiqueta video	117
Iframes	120
Etiqueta Figure	122
Etiquetas meta	123
Favicon – Creación y uso	123
Fontawesome.....	125
Atributos de accesibilidad	130

Open Graph Protocol	132
Introducción CSS 2021	134
Requisitos para el curso	135
¿Qué es CSS? Estado actual e historia.....	135
Instalaciones necesarias.....	135
Unir HTML y CSS	135
Sintaxis de CSS.....	139
Tipos de selectores en CSS	140
Selectores elementales	140
Selectores de id y de clase	142
Selectores de atributos	146
Selectores agrupados.....	149
Selectores Descendente.....	152
Selector de hermano.....	154
Selector de hijo directo	157
Especificidad y Cascada.....	158
Herencias en CSS	161
Estilos computados	164
Normalizar estilos del navegador.....	167
Prefijos Propietarios.....	169
Box Model	181
Width y Height	182
Margin	185
Errores comunes con Margin	190
Padding.....	191
Border.....	197
Box sizing.....	200
Bordes radius (circular)	203
Border radius (Elíptico)	207
Overflow	209
Colapsado de márgenes	215
Display	220
Outline.....	223
Text align	226
Box shadow	229
¿Qué es POSITION?	234

¿Qué es POSITION RELATIVE?	235
¿Qué es POSITION ABSOLUTE?	237
Todo lo que debes saber sobre Position Fixed.....	240
¿Cómo se usa y por qué no funciona Position Sticky?	244
¿Qué es y como funciona la propiedad z-index?	248
¿Qué es stacking context o Concepto de apilamiento?	252
¿Cómo ordenar propiedades?.....	258
¿Que son las medidas absolutas y medidas relativas?	259
Unidades de medida EM y REM	259
¿Qué es y cómo se usa width en % y width auto?	261
¿Cómo se utilizan y funcionan HEIGHT % y HEIGHT AUTO?	265
¿Cómo funcionan las medidas relativas al Viewport responsive?	268
Medidas relativas Min-Width, Min-Height, Max-Width y Max-height	273
Todo sobre códigos de colores en css [palabras clave].....	277
Currentcolor ¿Qué es y cómo se usa?.....	282
Modo de color RGB [Diferencia entre colores RGBA]	284
¿Qué es el Modo de Color Hexadecimal?	287
¿Qué es el modo de color HSL y para qué funciona?	288
¿Qué es la tipografía, ¿Cómo poner fuentes en CSS?	292
Aprender a cómo usar Google Fonts con CSS	295
Google Icons ¿Cómo usar librerías de ICONOS?	302
¿Cómo usar @FONT-FACE en CSS?	307
¿Cómo decorar textos en CSS?	311
¿Cómo dar espacio entre las letras en CSS?	313