

## EXERCISES QUE TRABAJAREMOS EN LA CUE

- EXERCISE 1: ENTORNO DE DESARROLLO.
- EXERCISE 2: CONOCIENDO HTML.
- EXERCISE 3: ETIQUETAS HTML.
- EXERCISE 4: ASSETS E IMAGENES.
- EXERCISE 5: ENLACES E HIPERVINCULOS

### EXERCISE 1: ENTORNO DE DESARROLLO

Para comenzar a trabajar con **HTML**, lo primero que necesitaremos será configurar nuestro entorno de desarrollo.

Existen muchos editores de texto que podríamos utilizar, tales como [Sublime Text](#), [Atom](#), [Notepad](#), entre otros pero, para conceptos de nuestro curso, utilizaremos **Visual Studio Code**: editor de código fuente que permite trabajar con diversos lenguajes de programación, además de ser gratuito y de código abierto. Sumaremos a esos atributos una interfaz gráfica amigable con el usuario.

#### DESCARGA DE VISUAL STUDIO CODE

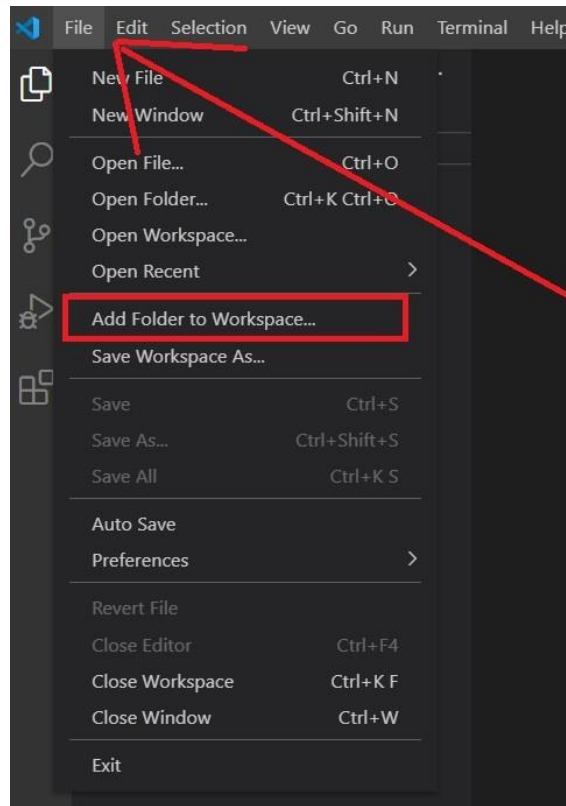
En su sitio oficial <https://code.visualstudio.com/> nos dirigiremos a la zona de descarga y seleccionaremos la versión para nuestro Sistema Operativo. **Visual Studio Code** cuenta con versión para **MacOS, Windows y Linux**.

Posteriormente seguiremos los pasos de instalación e incorporaremos un acceso directo en nuestro escritorio. Cuando ya se encuentre instalado, procedemos a conocerlo.

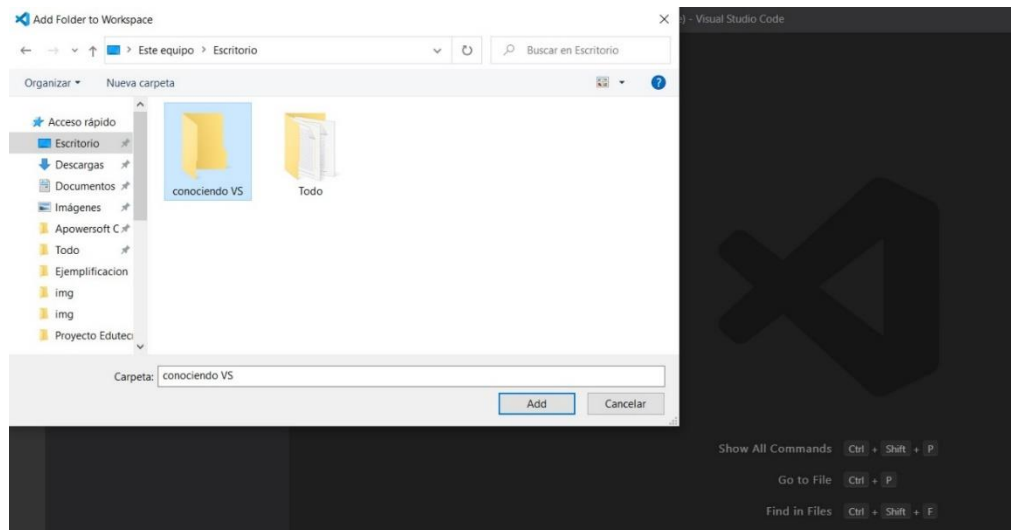
#### CONOCIENDO VISUAL STUDIO CODE

Antes de comenzar a utilizar nuestro editor, debemos conocer algunas funciones muy básicas de él.

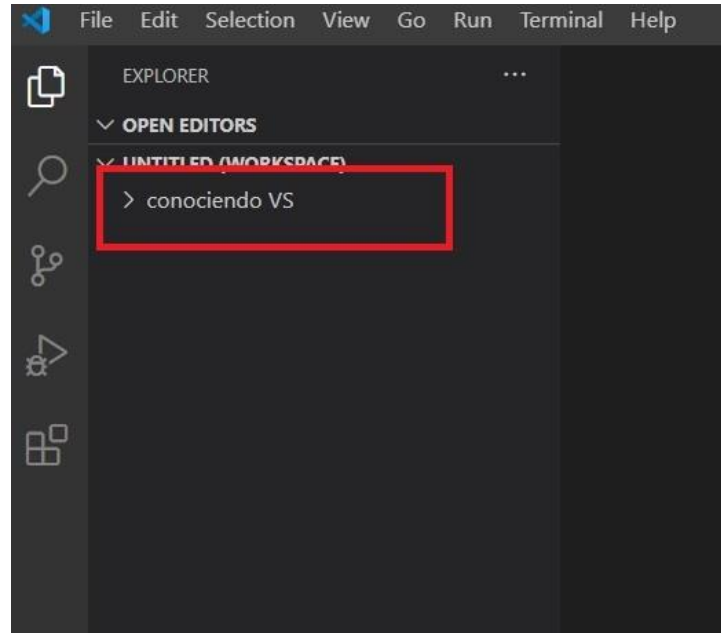
Comenzaremos creando una carpeta en nuestro escritorio a la cual le daremos por nombre "conociendo VS". Luego de eso, en nuestro editor **Visual Studio Code** iremos a **file - > Add Folder to Workspace**.



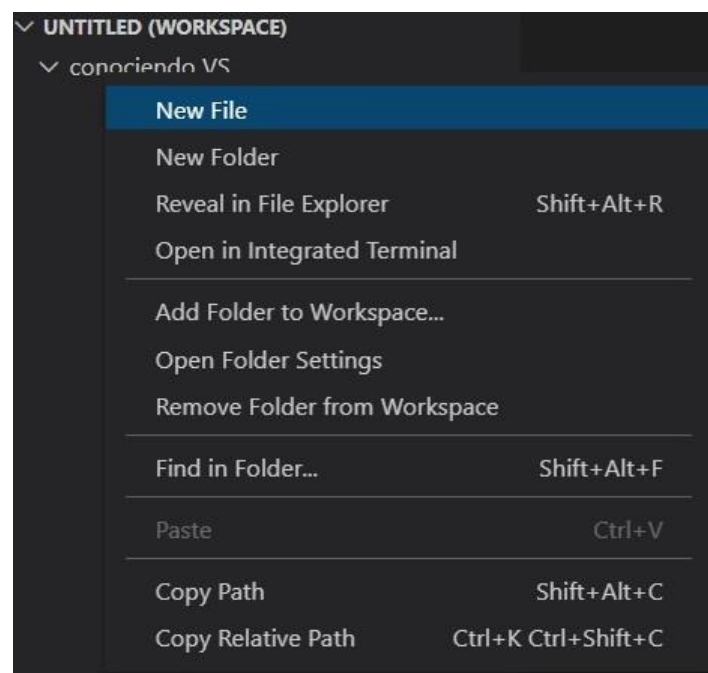
Seleccionamos la carpeta que deseamos y posteriormente seleccionamos **add**.



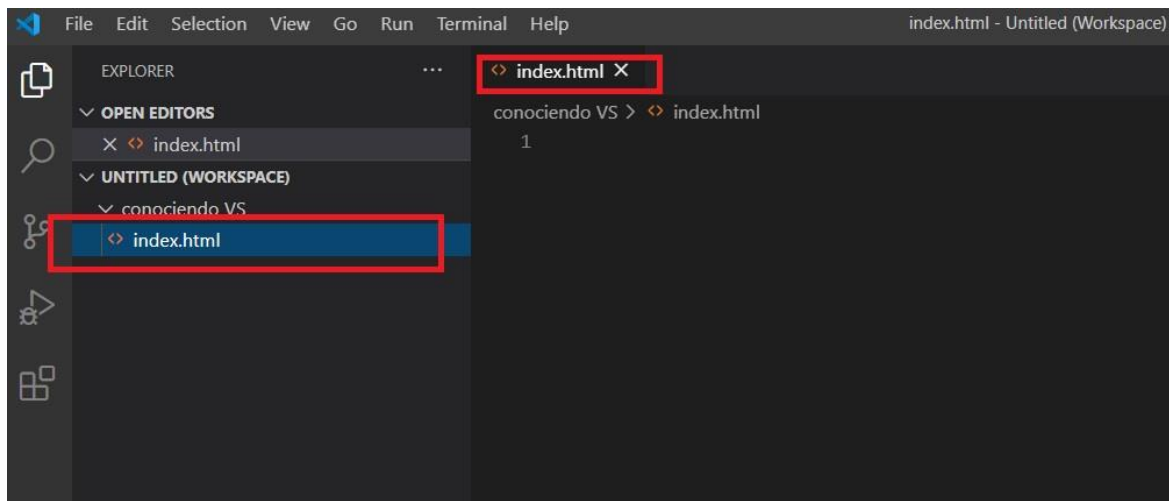
En nuestro editor se encuentra ahora la carpeta que añadimos para trabajar con ella.



Seleccionamos **New File** haciendo clic derecho sobre la carpeta. De esta forma podremos crear los archivos que queramos. Podemos ver que el menú nos permite también crear una nueva carpeta, añadir otra carpeta, abrir el terminal, copiar, entre otros.



Para crear un archivo de tipo **HTML** basta con colocar la extensión: para el ejemplo **index.html** y el archivo se crea de inmediato. Puedes reconocerlo también por sus corchetes angulares que representan una etiqueta **< >**.



## EXERCISE 2: CONOCIENDO HTML.

Lenguaje de Marcado de Hipertexto o **HyperText Markup Language** en inglés, es el significado de las siglas **HTML**, un lenguaje de etiquetas utilizado para estructurar y presentar el contenido para la web.

**HTML** ha contado con distintas versiones, encontrándose actualmente en la versión **HTML5**. En la siguiente tabla se presenta el resumen de la evolución de **HTML**:

Versión	Historia
<b>HTML 2.0</b>	En 1995 se publica el estándar HTML 2.0. A pesar de su nombre, HTML 2.0 es el primer estándar oficial de HTML, es decir, el HTML 1.0 no existió como estándar. HTML 2.0 no soportaba tablas. Se simplificaba al máximo la estructura del documento para agilizar su edición, donde la declaración explícita de los elementos body, html y head es opcional.
<b>HTML 3.2</b>	La versión HTML 3.2 se publicó en 1997 y es la primera recomendación de HTML publicada por el W3C (Consortio internacional). Esta revisión incorporó los últimos avances de las

	páginas web desarrolladas hasta 1996, como applets de Java y texto que fluye alrededor de las imágenes.
<b>HTML 4.01</b>	La última especificación oficial de HTML se publicó en 1998 y se denomina HTML 4.01. Desde la publicación de HTML 4.01, el W3C se centró en el desarrollo del estándar XHTML. Por este motivo, en el año 2004, las empresas Apple, Mozilla y Opera mostraron su preocupación por la falta de interés del W3C en HTML y decidieron organizarse en una nueva asociación llamada WHATWG (Web Hypertext Application Technology Working Group) que comenzó el desarrollo del HTML 5, cuyo primer borrador oficial se publicó en enero de 2008. Debido a la fuerza de las empresas que forman el grupo WHATWG y a la publicación de los borradores de HTML 5.0, en marzo de 2007 el W3C decidió retomar la actividad estandarizadora de HTML, dentro del cual decidió integrar el XHTML.
<b>HTML 5</b>	El consorcio internacional W3C, después de una evolución de varios años, liberó el HTML 5 como estándar oficial a finales de octubre de 2014. HTML 5 incorpora nuevos elementos no contemplados en HTML 4.01. Hay diversos cambios respecto a HTML 4.01. Hay nuevas etiquetas, se introduce la posibilidad de introducir audio y video de forma directa en la web sin necesidad de plugins o complementos en los navegadores, y otras novedades. El W3C irá lanzando progresivamente nuevas evoluciones del HTML 5.

HTML no es un lenguaje de programación, ya que no posee funciones aritméticas, estructuras de control o las variables por las que se caracterizan dichos lenguajes. HTML solo produce páginas web estáticas, pero que pueden ser combinadas con distintos lenguajes de programación para elaborar páginas web dinámicas.

## ¿PARA QUÉ SIRVE HTML?

**HTML** se utiliza para describir la estructura básica de una página. Podríamos decir que para crear algo así como el esqueleto de la página.

Organiza la forma en que se muestra el contenido de la página web, pudiendo ser este contenido texto, imágenes, videos, enlaces a otras páginas, formularios, entre otras cosas.

Esto quiere decir, en otras palabras, que **HTML** será la tecnología encargada de dar toda la estructura a la página web, mientras que no se preocupará del diseño ni la funcionalidad, siendo estas últimas dos cosas incorporadas con los elementos restantes de la triada de la que hablamos previamente y que serán vistos en profundidad más adelante.

## PRINCIPALES ETIQUETAS DE HTML

Este lenguaje se basa en etiquetas, cuyo nombre se encuentra delimitado por corchetes angulares `< >` y que se utilizan para la descripción de lo que se quiere representar en una página web.

La etiqueta de apertura siempre se representa como `<nombre etiqueta>` y la etiqueta de cierre siempre se representa como `</nombre etiqueta>` encontrándose entre medio de ambas el contenido que será graficado en el navegador al momento de ver nuestra página web.



*Ejemplo de etiqueta P con contenido TEXTO que será visualizado en el navegador*

## ESTRUCTURA BÁSICA DE UN DOCUMENTO HTML

Las etiquetas principales del documento **HTML** que conoceremos son las que componen la estructura básica del mismo (trabajando con **HTML 5**).

```
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-
7 scale=1.0">
8     <title>Document</title>
9 </head>
10 <body>
11     <!-- Contenido que será visualizado por el usuario -->
12 </body>
13 </html>
```

En un documento de extensión **HTML** (**index.html** para este ejemplo) escribiremos **HTML 5** en la primera línea y seleccionaremos la opción **HTML: 5** que **Visual Studio Code** nos entrega. De manera automática, creará la estructura que vemos en el código expuesto previamente. En caso de que no lo realice de manera automática, se puede realizar manualmente.

Lo importante es conocer que significa cada uno de los elementos que nos aparecen en el documento.

**1.- <!doctype html>:** o “Declaración del tipo de documento” es una instrucción especial que va al inicio de nuestro **documento HTML** y que permite al navegador entender qué versión de HTML estamos utilizando. Esta información determinará la manera en la que el navegador procesará el documento, un **DOCTYPE** distinto podría implicar hasta una visualización diferente del sitio web dentro del mismo navegador.

**2.- <html>:** Esta etiqueta indica la apertura del documento de tipo **HTML**. Es la última en cerrarse (podemos ver la etiqueta de cierre **</html>** en la línea 13 del código de ejemplo) ya que dentro de ella va todo el contenido del documento.

**3.- <head>:** Esta etiqueta contiene la información para el navegador. Se denominan Metadatos y pueden ser el título, autor, enlaces a hojas de estilo, entre otros. El cierre de la sección del **<head>** se realiza cuando encontramos la etiqueta de cierre **</head>** (en la línea 9 de código de ejemplo), justo antes del comienzo de la etiqueta **<body>**.

**4.- <meta>:** Esta etiqueta contiene los metadatos del **<head>**. Llevan uno o más atributos (**charset** por ejemplo) que es la indicación del metadato que contiene la etiqueta. Los atributos son pares nombre – valores separados por el signo igual (=). El valor va entre comillas dobles o simples. Existen distintos atributos que pueden incluirse en una etiqueta, que veremos con detalle más adelante.

Para el ejemplo de esta etiqueta **<meta>**, el **charset** es el atributo que indica los caracteres permitidos o utilizados en este documento. **UTF-8**, que es el valor que se le asignó al atributo, indica que se permite caracteres latinos.

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

**5.- <title>:** Esta etiqueta contiene el título del documento.

**6.- <body>:** Esta etiqueta indica el inicio del cuerpo del documento. En este lugar va el contenido general que se ha escrito para ser presentado en la página web. Será lo que el usuario verá. La etiqueta de cierre de cuerpo del documento ( **</body>** ) indica hasta donde está el contenido que será mostrado al usuario.

Se puede observar que, a diferencia de lo descrito unas líneas antes en esta lectura, algunas etiquetas no cuentan con etiqueta de cierre. Efectivamente es así, no es que se encuentren mal escritas. Estas etiquetas llevan la información como atributos.

## ATRIBUTOS DE LAS ETIQUETAS BÁSICAS

La etiqueta `<html>` tiene el atributo `Lang = "en"`: se utiliza para indicar el idioma del documento. Por defecto esta como "en" (inglés). Para indicar que estará en español debemos cambiarlo a "es".

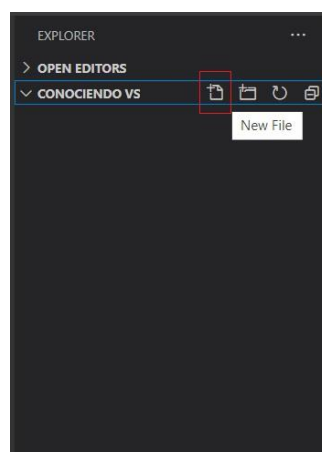
Explicamos previamente el atributo `charset`, pero tenemos otra etiqueta `<meta>` con el atributo `http-equiv`, cuyo valor tiene relación con la compatibilidad del navegador donde se está consultando la página web. Además, cuenta con el atributo `content`, donde indicamos el modo de compatibilidad, para el caso, el más alto disponible.

Seguimos con la etiqueta `<meta>` y el atributo `name = "viewport"`: este atributo nos permite configurar como se debe interpretar una página web en diferentes dispositivos.

Contiene también el atributo `content = "width = device - width, initial-scale=1.0"` para definir la forma de interpretar la página web (va en conjunto con `viewport`). `Width` permite definir la anchura de la pantalla. `Device-width`, que es el valor por defecto que aparece, hace referencia a la anchura de la pantalla del dispositivo que está desplegando la página y no a una anchura fija (este es el valor habitualmente definido). `Initial-scale` se utiliza para definir la escala inicial del documento, para evitar transformaciones.

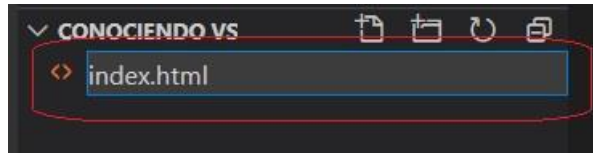
## MI PRIMER HOLA MUNDO CON HTML

Ya contamos con el archivo "Conociendo VS" incorporado en el editor **Visual Studio Code**. Para generar el documento **HTML** seleccionamos la opción **new file** del archivo **Conociendo VS**:

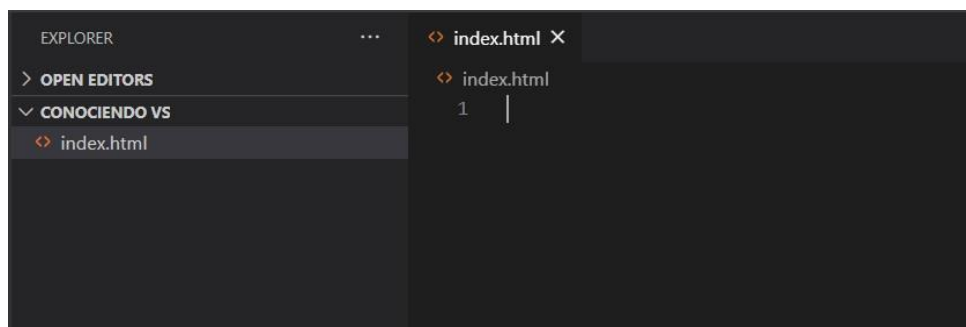




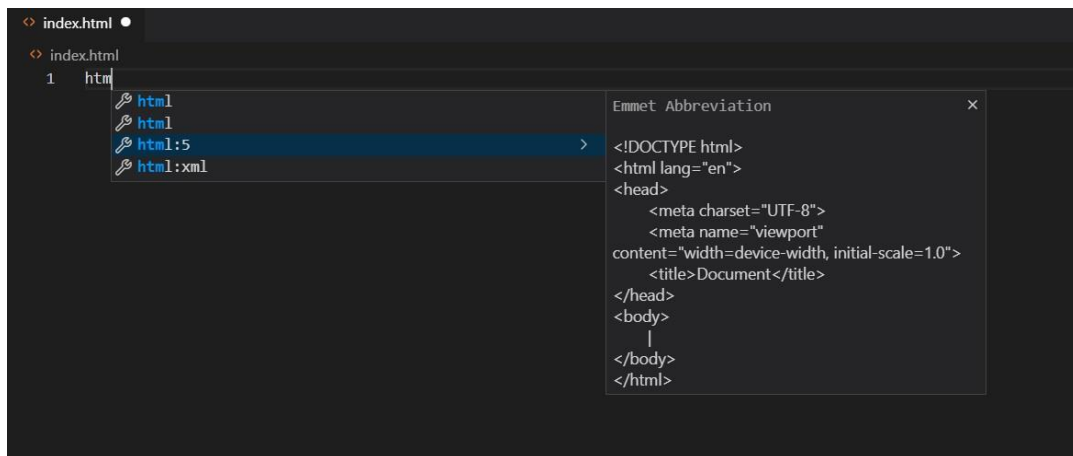
En la ventana desplegada colocaremos el nombre del archivo y la extensión de este: En el ejemplo anterior “Descarga e instalación de VS Code” también se ejemplifico la creación del documento **HTML**.



Después de realizar esto se desplegará el documento aún vacío:



En la línea 1 escribimos **HTML** y seleccionamos la opción **HTML:5**



De esa forma se generará la estructura inicial básica de **HTML** de manera automática: también podemos escribirlo manualmente si así lo deseamos.

```
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
```

```
7   <meta name="viewport" content="width=device-width, initial-  
8 scale=1.0">  
9   <title>Document</title>  
10 </head>  
11 <body>  
12  
13 </body>  
   </html>
```

Entre la etiqueta **<title>** borraremos “Document” y escribiremos como título del documento “Mi primer documento HTML”. Esta es información para el navegador y no se verá reflejada gráficamente en el cuerpo de la página.

Dentro de las etiquetas **<body>** donde se encuentra el cuerpo del documento o, en otras palabras, todo lo que será visto gráficamente en el navegador colocaremos la etiqueta **<h1>** y dentro de ella la frase ¡HOLA MUNDO!

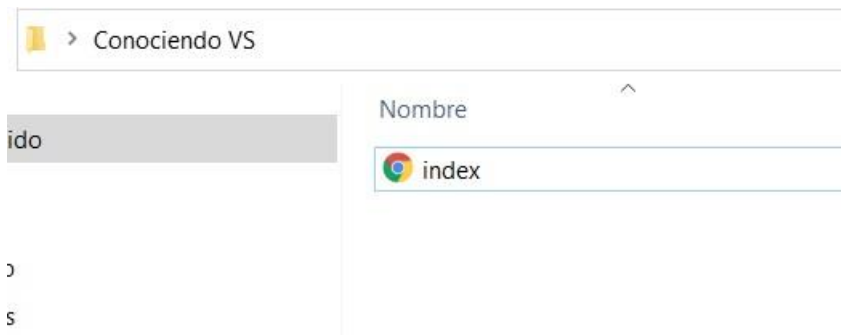
```
1 <!DOCTYPE html>  
2 <html lang="es">  
3 <head>  
4   <meta charset="UTF-8">  
5   <meta name="viewport" content="width=device-width, initial-  
6 scale=1.0">  
7   <title>Mi primer documento HTML</title>  
8 </head>  
9 <body>  
10  <h1>¡HOLA MUNDO!</h1>  
11 </body>  
12 </html>
```

Se colocó una marca en la parte de arriba del documento: **index.html** y al lado hay un punto blanco. Este punto se encarga de recordarnos que los cambios realizados en el documento aún no han sido guardados (por lo que no se verían reflejados al momento de ver el documento en un navegador).

Para guardar los cambios, podemos hacer clic sobre **File** y seleccionar **save** o utilizar el atajo de teclado **ctrl + S**. Al guardar los cambios desaparecerá el círculo y será reemplazo por la **x** característica para cerrar el documento.

```
<> index.html •
<> index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Mi primer documentos HTML</title>
7  </head>
8  <body>
9      <h1>¡HOLA MUNDO!</h1>
10 </body>
11 </html>
```

En la carpeta **Conociendo VS** haremos doble clic sobre el archivo (o con **botón secundario -> abrir**) y se levantará nuestra página web en nuestro navegador predeterminado o en el que seleccionemos para que ocurra.



Una vez abierto el archivo ya podemos ver nuestra primera codificación en **HTML** en un navegador.



## ¡HOLA MUNDO!

¡Felicidades! ¡Ya escribimos código **HTML** por primera vez!

## EXERCISE 3: ETIQUETAS HTML

Sabemos que **HTML** es el lenguaje con el que se escriben páginas web a través de etiquetas, por lo que es necesario conocer algunas de las etiquetas que podremos utilizar dentro del documento para dar forma a la página.

Conocemos ya las etiquetas de la estructura básica, que detallamos en el ejemplo “Conociendo HTML”. Conozcamos algunas nuevas etiquetas básicas para poder desarrollar con **HTML**:

### 1.- Etiquetas de Párrafos:

Estas etiquetas nos permitirán escribir párrafos dentro del documento y existen varias distintas:

- **Encabezado:**

```
1 <h1>Soy un encabezado h1</h1>
```

Los encabezados tienen 6 niveles, siendo **<h1>** el más alto y **<h6>** el más bajo. El encabezado **<h1>** es generalmente utilizado para ser el encabezado principal de la página. Los demás van disminuyendo su tamaño.

- **Párrafo normal:**

```
1 <p>Soy un párrafo </p>
```

Para la escritura de un párrafo.

- **Salto y líneas:**

```
1 <p>Lorem ipsum dolor sit, amet consectetur<br> adipisicing elit. Sed  
2 deleniti molestiae delectus  
3     recusandae quos facilis placeat aperiam quo rerum <hr>  
4 reprehenderit atque, dolorum eius voluptatibus non,  
5     reiciendis ipsa laboriosam nam.</p>
```

La etiqueta **<br>** nos permite colocar un salto de línea sin espacio en el párrafo.

La etiqueta **<hr>** nos coloca una línea entre el párrafo.

**Lorem ipsum:** Texto prediseñado que se utiliza para colocar párrafo de prueba cuando no tenemos el texto final que llevará una página.

- **Sangría:**

```
1  <blockquote>
2  <p>
3  Lorem ipsum dolor sit, amet consectetur adipisicing elit. Debitis, ea
4  in! Veniam atque tenetur illum suscipit
5  nostrum quia sed commodi,
6  doloribus architecto cupiditate optio fugiat ullam modi dolorum harum
7  ducimus?
8  </p>
9
10 </blockquote>
```

La página web se verá así:

## Soy un encabezado h1

### Encabezado h2

#### Encabezado h3

#### Encabezado h4

#### Encabezado h5

#### Encabezado h6

Soy un párrafo

Lorem ipsum dolor sit, amet consectetur  
adipiscing elit. Sed deleniti molestiae delectus recusandae quos facilis placeat aperiam quo rerum

---

reprehenderit atque, dolorum eius voluptatibus non, reiciendis ipsa laboriosam nam.

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Debitis, ea in! Veniam atque tenetur illum suscipit nostrum quia sed commodi, doloribus architecto cupiditate optio fugiat ullam modi dolorum harum ducimus?

## 2.- Etiquetas de Listas:

Estas etiquetas nos permitirán definir listas dentro del documento. Son bastante utilizadas para realizar, por ejemplo, los vínculos que podemos encontrar en los inicios de las páginas, en conjunto con etiquetas de hipervínculos que veremos más adelante:

```
1  <li>Elemento 1</li>
2  <li>Elemento 2</li>
3  <li>Elemento N</li>
```

Para mostrar los elementos de la lista. Esta etiqueta se utiliza dentro de las siguientes que veremos.

- **Lista ordenada:**

```
1  <ol>
2    <li>Elemento 1</li>
3    <li>Elemento 2</li>
4    <li>Elemento N</li>
5  </ol>
```

Nos ordena la lista de forma numérica.

- **Lista desordenada:**

```
1 <ul>
2   <li>Elemento 1</li>
3   <li>Elemento 2</li>
4   <li>Elemento N</li>
5 </ul>
```

Sin orden numérico.

- **Lista de definición:**

```
1 <dl>
2   <dt>Término1</dt>
3   <dd>Definición 1</dd>
4   <dt>Término 2</dt>
5   <dd>Definición 2</dd>
6   <dt>Término N</dt>
7   <dd>Definición N</dd>
8 </dl>
```

Permite colocar una definición debajo de cada termino de manera visualmente entendible.

La página web se verá así:

---

- Elemento 1
- Elemento 2
- Elemento N

1. Elemento 1
2. Elemento 2
3. Elemento N

- Elemento 1
- Elemento 2
- Elemento N

Término1  
Definición 1

Término 2  
Definición 2

...

Término N  
Definición N

### 3.- Etiquetas de tabla:

Nos permiten mostrar información en una tabla.

- **Definir una tabla**

```
1 <table>
2
3 </table>
```

Dentro de la etiqueta `<table>` se añadirán los elementos para darle forma. Estas etiquetas son fundamentales para indicar que lo que se realizará es una tabla y por eso contienen al resto de etiquetas que veremos a continuación.

- **Cabecera y cuerpo de la tabla:**

```
1 <table>
2   <thead>
3
4   </thead>
5
6   <tbody>
7
8   </tbody>
9 </table>
```

La etiqueta `<thead>` contiene las cabeceras de la tabla y la `<tbody>` contiene el cuerpo de la tabla, ósea, la información que se desplegará en ella.

- **Filas:**

```
1 <table>
2   <thead>
3     <tr>
4
5     </tr>
6
7   </thead>
8
9   <tbody>
10    <tr>
11
12    </tr>
13  </tbody>
14 </table>
```

Las etiquetas `<tr>` nos permiten definir las filas dentro de la tabla

- **Celdas:**

```
1 <table>
2   <thead>
3     <tr>
```

```
4      <th>NUMERO</th>
5      <th>NOMBRE</th>
6
7    </tr>
8
9  </thead>
10
11 <tbody>
12   <tr>
13     <td>1</td>
14     <td>PEPITO</td>
15   </tr>
16   <tr>
17     <td>2</td>
18     <td>JUANITO</td>
19   </tr>
20 </tbody>
21</table>
```

La etiqueta **<th>** en el head de la tabla nos permite definir celdas de título.

La etiqueta **<td>** en el body nos permite definir celdas de datos.

La página web se verá así:

NUMERO	NOMBRE
1	PEPITO
2	JUANITO

## ETIQUETAS SEMÁNTICAS Y DIV

Una etiqueta semántica es aquella que ayuda a definir la estructura del documento y nos informa sobre que trata su contenido.

Se incorporan en **HTML5** para dar mayor facilidad tanto a los desarrolladores como a los buscadores en el desarrollo y análisis de las páginas, respectivamente.

Vienen a cambiar el uso de la etiqueta **<div>** como único medio de dividir la página, pero antes que todo ¿qué es la etiqueta **<div>**?

El nombre de esta etiqueta proviene de división y su función es dividir y contener bloques de instrucciones, es decir, puede contener uno o varios elementos **HTML** dentro de ella. Es una etiqueta



muy utilizada incluso después de la incorporación de las etiquetas semánticas, pero solamente divide o agrupa los elementos, sin entregar información sobre su cometido.

El uso de las etiquetas semánticas busca dividir en orden estos contenidos y entregar información sobre su función dentro del código fuente.

### Las nuevas etiquetas Semánticas:

- **Main:** esta etiqueta contiene el cuerpo de la aplicación o del documento. El contenido principal consiste en aquel que está directamente relacionado con el tema central del documento o con la funcionalidad principal de la aplicación.
- **header:** el elemento header debe utilizarse para marcar la cabecera de una página (contiene el logotipo del sitio, una imagen, un cuadro de búsqueda etc.). Puede estar anidado en otras secciones de la página (es decir que no sólo se utiliza para la cabecera de la página).
- **footer:** se utiliza para indicar el pie de la página o de una sección. Un pie de página contiene información general acerca de su sección el autor, enlaces a documentos relacionados, datos de derechos de autor etc.
- **nav:** representa una parte de una página que enlaza a otras páginas o partes dentro de la página. Es una sección con enlaces de navegación.
- **section:** el elemento section representa un documento genérico o de la sección de aplicación. Una sección, en este contexto, es una agrupación temática de los contenidos. Puede ser un capítulo, una sección de un capítulo o básicamente cualquier cosa que incluya su propio encabezado. Una página de inicio de un sitio Web puede ser dividida en secciones: introducción, noticias, información de contacto etc.
- **aside:** el elemento aside representa una nota, consejo, una explicación, etc. Este elemento puede ser utilizado para efectos de atracción, como las comillas tipográficas o barras laterales, para la publicidad, por grupos de elementos de navegación, y por otro contenido que se considera por separado del contenido principal de la página.
- **article:** el elemento article representa una entrada independiente en un blog, revista, periódico etc.
- **figure:** representa una unidad de contenido que es autosuficiente y que puede ser movido fuera del flujo principal del documento sin que esto afecte a su significado. Por lo general contiene una figura o imagen.

## EXERCISE 4: ASSETS E IMAGENES

Los **assets** (activos, en inglés) son aquellos recursos referenciados desde el código **HTML** de la página. Pueden ser archivos **CSS**, **JAVASCRIPT**, imágenes, videos, entre otros.

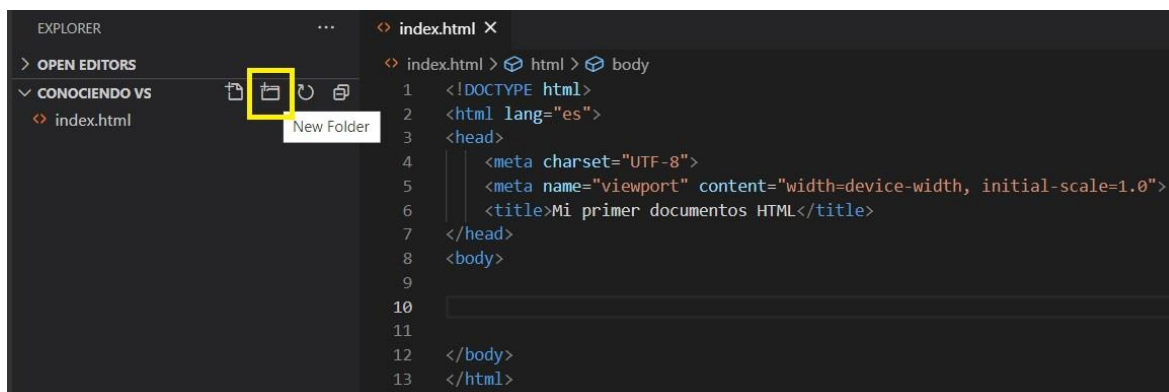
Se considera una buena práctica almacenar todos los **assets** en el directorio web del proyecto, ya que dispersarlos en diferentes directorios vuelve más difícil la gestión de los mismos.

El directorio web es público, por lo que cualquier usuario puede acceder a todos los contenidos directamente desde su navegador.

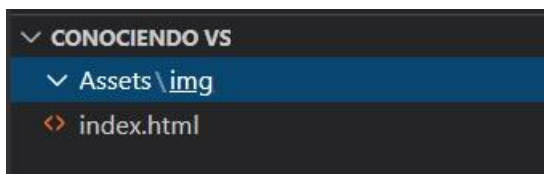
En esta oportunidad, incorporaremos dentro del proyecto una imagen.

Las imágenes nos permiten dar mayor atractivo al contenido web y mejorar la forma de presentarlo.

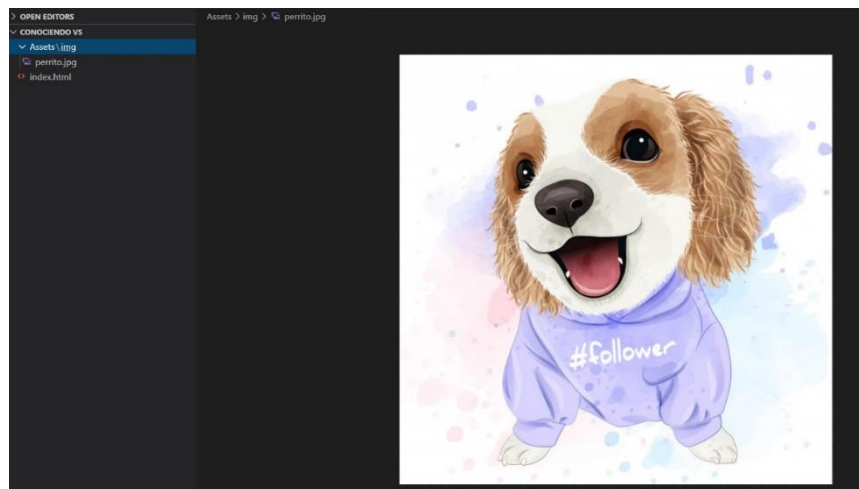
Vamos a crear una nueva carpeta haciendo clic a **new folder** y le colocaremos por nombre **Assets**.



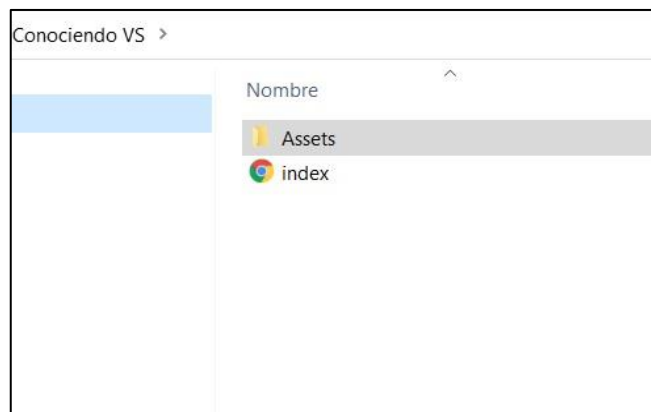
Dentro de ella crearemos otra de nombre **img**.



Tomaremos una imagen desde la galería de imágenes de nuestro pc (no importa que imagen sea) y la arrastraremos hasta la carpeta **img** que se encuentra dentro de la carpeta **Assets**.



Ahora que tenemos incorporada la imagen dentro del proyecto podemos corroborarlo dirigiéndonos al proyecto e ingresando a las carpetas **Assets** e **img**, dentro de la cual encontraremos la imagen:



En el archivo HTML, dentro de las etiquetas `<body>` incorporaremos la etiqueta `<img>`.

La etiqueta `<img>` es una etiqueta que no cuenta con etiqueta de cierre y que inserta una imagen dentro del documento valiéndose de atributos.

## PERO AQUÍ NOS ASALTA LA DUDA, ¿QUÉ ES UN ATRIBUTO?

Previamente mencionamos a grandes rasgos lo que son los atributos, pero ahora llegó el momento de profundizar un poco más en ellos.

Los atributos son pares de **nombre – valor**, separados por “=” y escritos en la etiqueta inicial de un elemento, después del nombre del elemento. El valor puede estar encerrado entre comillas dobles o simples y su función es modificar el comportamiento de las etiquetas.

La estructura de una etiqueta con atributos incorporados es la siguiente:

```
1 <tag attribute1="value1" attribute2="value2">content</tag>
```

## TIPOS DE ATRIBUTOS:

Algunas etiquetas tienen sus atributos propios, sin embargo, hay atributos comunes a las etiquetas que se dividen en: básicos, eventos, foco, internacionalización.

- **Atributos básicos:**
  - **Id** : Establece un indicador único a cada elemento.
  - **Class** : establece la clase **CSS** que se aplica a los estilos del elemento.
  - **Style** : aplica estilos **CSS** de forma directa en el documento. Se considera buena practicar trabajar el estilo en un documento CSS separado.
  - **Title** : establece el título del elemento.

- **Atributos de eventos:**

Se utilizan en páginas web dinámicas que incluyen **JavaScript**.

- **OnClick** : ejecuta una acción cuando se realiza clic sobre el elemento.
- **Ondblclick** : ejecuta una acción cuando se realiza doble clic sobre el elemento.
- **Onmousedown** : ejecuta una acción cuando se detecta el botón pulsado del ratón.
- **Onmouseup** : ejecuta la acción cuando se detecta que se ha soltado el botón del ratón.
- **Onmouseover** : ejecuta la acción cuando se detecta que el ratón se sitúa sobre el elemento.
- **Onload** : ejecuta la acción cuando se carga el documento.
- **Onkeydown** : ejecuta la acción cuando se detecta que la tecla esta pulsada.

- **Onsubmit** : ejecuta la acción cuando el formulario es enviado. Se utiliza con formularios `<form>`.

- **Atributos de foco:**

Se le denomina foco o *focus*, cuando un control o elemento del documento ha sido seleccionado.

- **Accesskey** : establece una tecla de acceso rápido a un elemento **HTML**.
- **Onfocus,onblur** : controla los eventos **JavaScript** que se ejecutan cuando el elemento obtiene o pierde el foco.

- **Atributos de internacionalización:**

Estos atributos se utilizan en aquellas páginas que muestran sus contenidos en varios idiomas y las que quieran indicar de forma explícita el idioma de sus contenidos:

- **Lang** : indica el idioma del elemento.
- **Dir** : indica la dirección del texto.

## RETOMANDO EL EJEMPLO

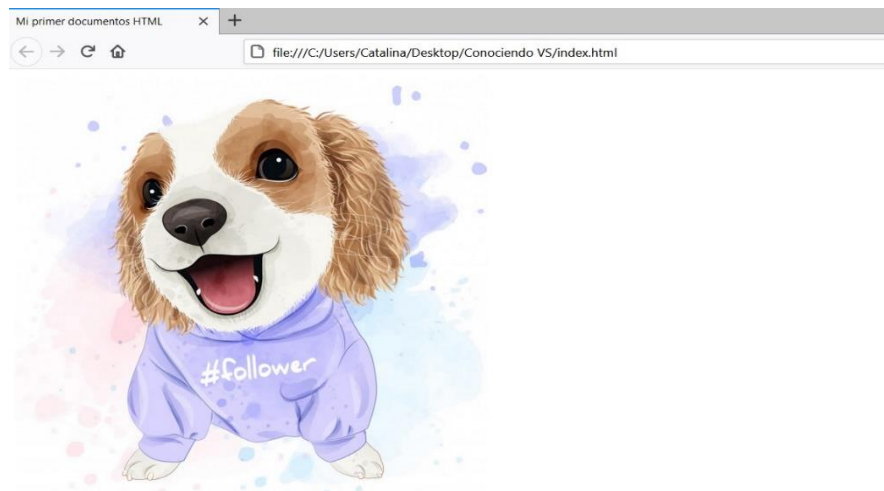
Ahora que tenemos claridad sobre que es un atributo y que existen distintos tipos de ellos, retomaremos el ejemplo desde la etiqueta `<img>` que dejamos inconclusa previamente, donde incorporaremos los siguientes atributos:

- **src**: **URL** o ruta donde se encuentra la imagen.
- **alt**: texto alternativo. Debe ser un texto descriptivo de la imagen. Se utiliza en lectores para invidentes, además de mostrarse en caso de que la imagen no sea encontrada.
- **height**: altura de la imagen en pixeles.
- **width**: anchura de la imagen en pixeles

Se verá de la siguiente manera:

```
1 
```

Guardamos los cambios realizados y actualizamos en el navegador la página web.



Finalmente hemos incorporado al archivo una imagen de nuestra preferencia a la cual le dimos un tamaño predeterminado con los atributos **width** y **height**.

## EXERCISE 5: ENLACES E HIPERVINCULOS

Enlaces o hipervínculos, conocidos también como hipertextos, son los textos o los objetos sobre los que podemos hacer clic para que nos lleven a otra parte del documento, a otra página web o a otra página de nuestro mismo proyecto.

Esto quiere decir que, cuando hablamos de los enlaces e hipervínculos hablamos de la acción de establecer una referencia a una dirección o **URL** donde se encuentra alojado otro documento de destino que, como previamente indicamos, puede ser un sector dentro de la misma página, de otra página, un archivo pdf, una imagen u otro tipo de documento.

Para poder realizar los enlaces, usaremos la etiqueta **<a>** que debe tener, como mínimo, el atributo **href**, atributo en el cual se especifica la dirección **URL** al documento que se quiere enlazar.

```
1 <a href=""></a>
```

Para ejemplificarlo mejor, vamos a realizar un enlace directamente a la página de **mozilla**, específicamente al apartado de hipervínculos.



Esta página fue traducida del inglés por la comunidad, pero no se mantiene activamente, por lo que puede estar desactualizada. Si desea ayudar a mantenerlo, descubra cómo activar las configuraciones regionales inactivas. [↗](#)

## Table of contents

[¿Qué es un hipervínculo?](#)[Anatomía de un enlace](#)[Primer acercamiento a URLs y rutas](#)[Buenas prácticas en el uso de los enlaces](#)[Aprendizaje activo: crear un menú de navegación](#)[Enlace a correo electrónico](#)[¡Pon a prueba tus habilidades!](#)[Resumen](#)[En este módulo](#)

## Crear hipervínculos

[Anterior](#)[Overview: Introduction to HTML](#)[Siguiente](#)

Los hipervínculos (o enlaces) son elementos verdaderamente importantes — son los que hacen que la web sea **web**. Este artículo expone la sintaxis necesaria para crear un enlace, además contiene un catálogo de buenas prácticas para crearlos.

**Prerrequisitos:**

Estar familiarizado con HTML, cubierto en [Empezar con HTML](#). Aplicación de formato a texto en documentos HTML, según lo expuesto en la sección [fundamentos de texto HTML](#).

**Objetivo:**

Aprender a implementar un *hipervínculo* de forma efectiva y enlazar múltiples archivos.

Para eso, en el atributo **href**, simplemente vamos a copiar el enlace de la página a la que deseamos hacer referencia y, para que el usuario tenga acceso a ver el enlace, vamos a escribir, entre las etiquetas **<a>** una frase que será visible para el usuario que vea nuestra página web.

```
<a  
1 href="https://developer.mozilla.org/es/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks">Enlace a página externa</a>
```

Vamos a guardar los cambios y abrir (o actualizar) nuestra página web en nuestro navegador de preferencia, obteniendo como resultado:

[Enlace a página externa](https://developer.mozilla.org/es/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks)

Al hacer clic en el enlace, de manera automática, somos redirigidos a la página web que habíamos referenciado.

Es importante destacar que **href** es el único atributo obligatorio de la etiqueta **<a>** pero no el único permitido. Existen varios atributos utilizables con distinto valor que veremos en la tabla a continuación.

ATRIBUTO	VALOR	DESCRIPCIÓN
<b>href</b>	URL	Enlace al documento que se quiere cargar. Este tributo es obligatorio.
<b>download</b>	Nombre.extensión	Descarga el enlace (href) en lugar de abrirlo. Si se indica valor, se renombra.
<b>target</b>	_blank	Abre el enlace en una nueva pestaña
	_self	Abre el enlace en la pestaña actual.
	_parent	Abre el enlace en el documento padre.
<b>rel</b>	alternate	Índice que el enlace es una versión alternativa.
	tag	Asocia una etiqueta al enlace especificado, en relación con el documento actual.
	search	Indica que el enlace es una página para realizar búsquedas.

Además, podemos utilizar el atributo **title** que no es exclusivo para esta etiqueta y podemos utilizar en cualquier otra etiqueta. Este atributo nos entrega un texto que será visible cuando el usuario se pose sobre el elemento.

```
<a  
1 href="https://developer.mozilla.org/es/docs/Learn/HTML/Introduction_to  
_HTML/Creating_hyperlinks" title="leeme">Enlace a página externa</a>
```

Dando como resultado:

[Enlace a página externa](https://developer.mozilla.org/es/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks)

leeme

De esta forma, hemos logrado realizar enlaces dentro de nuestra página web.