

# Tema 2: Desarrollo web en cliente: HTML.



Bloque 1: Desarrollo web en cliente: HTML y CSS.

Módulo: Lenguajes de marcas y sistemas de gestión de la información.

IES San Vicente  
Curso 2021-2022

# Índice de contenidos

1	Estructura de un documento HTML/xHTML.....	4
1.1	Introducción a HTML.....	4
1.2	Historia.....	4
2	Elementos HTML: etiquetas y atributos.....	5
2.1	Anidar elementos.....	6
2.2	Elementos vacíos.....	6
3	Estructura básica de un documento HTML.....	6
3.1	DOCTYPE.....	7
3.2	Elementos de la cabecera.....	8
3.2.1	Las metaetiquetas.....	8
3.2.1.1	charset.....	8
3.2.1.2	description.....	9
3.2.1.3	keywords.....	9
3.2.1.4	author.....	9
3.2.1.5	viewport.....	9
3.2.2	Elemento title.....	9
4	Ejemplo de página web.....	9
5	Comentarios en HTML.....	10
6	xHTML.....	10
7	Principales etiquetas HTML.....	12
7.1	Encabezados y párrafos.....	12
7.1.1	Resaltado de texto: Negrita y cursiva.....	13
7.1.1.1	Otras etiquetas.....	14
7.1.2	La etiqueta span.....	15
7.1.3	Inclusión de caracteres especiales en HTML.....	15
7.2	Elementos contenedor.....	16
7.2.1	Elemento div.....	16
7.3	Listas.....	16
7.3.1	Listas no ordenadas.....	16
7.3.2	Listas ordenadas.....	17
7.3.3	Listas anidadas.....	18
7.3.4	Listas de descripción: etiquetas dl, dt y dd.....	18
7.4	Enlaces.....	19
7.4.1	Ruta absoluta y ruta relativa.....	20
7.5	Imágenes.....	20
7.5.1	Imágenes y enlaces.....	21
7.5.2	Favicon (icono de pestaña).....	21
7.5.3	Elemento figure.....	22
7.6	Tablas.....	22
7.7	Otras etiquetas.....	25
7.7.1	Citas: etiquetas cite, blockquote y q.....	25
7.7.2	Texto preformateado: etiquetas pre y code.....	26
7.7.3	Marcado de texto: etiqueta mark.....	27
7.7.4	Borrados e inserciones: etiquetas del e ins.....	27
7.7.5	Etiqueta delimitadora: br y hr.....	28
7.7.6	El elemento time.....	28

7.7.7 El elemento small.....	28
7.7.8 El elemento address.....	29
7.8 Etiquetas de secciones.....	29
7.8.1 Elementos de diseño HTML en detalle.....	31
7.8.1.1 Encabezado <header>.....	31
7.8.1.2 Menú <nav>.....	32
7.8.1.3 Contenido principal <main>.....	32
7.8.1.4 Elemento <article>.....	33
7.8.1.5 Elemento <section>.....	33
7.8.1.6 Barra lateral <aside>.....	34
7.8.1.7 Pie de página <footer>.....	35
8 Validación de nuestra web.....	36
8.1 Como validar utilizando VSCode.....	37
9 Bibliografía.....	38

# 1 Estructura de un documento HTML/xHTML

## 1.1 Introducción a HTML

Una página web es un archivo de texto con una serie de etiquetas de marcas que contienen la información a mostrar en el navegador.

El lenguaje HTML (Hypertext markup language, en castellano Lenguaje de marcas de hipertexto) consiste en un conjunto de etiquetas predefinidas que el navegador web interpreta a la hora de construir una página para su visualización. Es el lenguaje utilizado para estructurar una página web.

Aunque nos refiramos a él como un lenguaje, no lo es como tal, ya que carece de ciertas características que definen un lenguaje, como las estructuras de control y la toma de decisiones. Así, se puede definir el HTML como una serie de instrucciones o etiquetas que indican al navegador cómo debe estructurar el contenido. Es un lenguaje estándar, reconocido en todo el mundo, y definido por el [W3C](#) (World Wide Web Consortium), una entidad sin ánimo de lucro que desarrolla estándares que aseguran el crecimiento de la Web a largo plazo.

## 1.2 Historia

Tim Berners-Lee, un científico inglés que trabajaba en el CERN, la Organización Europea para la Investigación Nuclear, inventó un protocolo de comunicación basado en hipertexto (enlaces dentro de palabras), que permitía crear páginas con texto e imágenes que se enlazaban entre sí. Lo llamó World Wide Web (WWW). Esta es la [primera página web](#) de la historia:

### World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#), [Policy](#), November's [W3 news](#), [Frequently Asked Questions](#).

[What's out there?](#)

Pointers to the world's online information, [subjects](#), [W3 servers](#), etc.

[Help](#)

on the browser you are using

[Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#), [X11 Viola](#), [NeXTStep](#), [Servers](#), [Tools](#), [Mail robot](#), [Library](#).)

[Technical](#)

Details of protocols, formats, program internals etc

[Bibliography](#)

Paper documentation on W3 and references.

[People](#)

A list of some people involved in the project.

[History](#)

A summary of the history of the project.

[How can I help?](#)

If you would like to support the web..

[Getting code](#)

Getting the code by [anonymous FTP](#), etc.

La primera web se creó el 6 de agosto de 1991, pero no se abrió a todo el mundo hasta el 23 de agosto de 1991. En 1995 se publicó el primer estándar oficial de HTML, pero no fue hasta 1996 cuando pasó las especificaciones del HTML a manos de W3C con la versión HTML3. HTML3 nunca se llegó a estandarizar, por lo que la siguiente versión que se publicó fue HTML3.2 en 1997, donde se incorporaban tecnologías recientes para la época como los applets de Java.

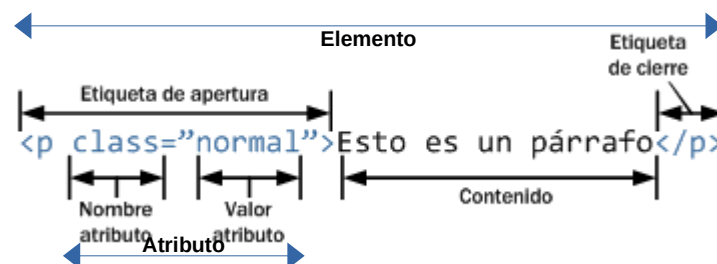
La siguiente versión fue lanzada en 1998, HTML4.0 que incluyó una serie de novedades como la posibilidad de incluir scripts en las páginas web, añadir hojas de estilos CSS, tablas complejas, formularios y la accesibilidad de las páginas. Un año más

tarde se publicó HTML4.0.1, que no es más que una revisión y actualización del lenguaje HTML 4.0

Tras esta publicación, la actividad en torno a HTML se ralentiza ya que se interesan más en el desarrollo del XML dejando parado el desarrollo de HTML. En 2004, Apple, Mozilla y Opera formaron una asociación denominada WHATWG (Web HyperText Application Technology Working Group) y desarrollan HTML5. En 2007, W3C se une a este grupo y publican HTML5. El objetivo de esta versión era dar la posibilidad de crear aplicaciones multiplataforma con HTML5, CSS3 y JS.

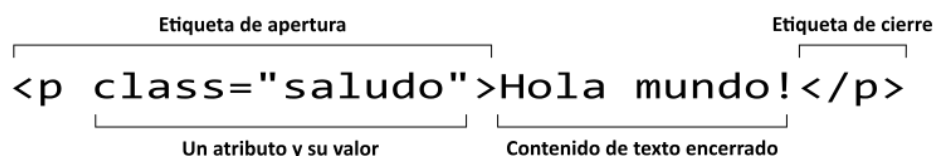
## 2 Elementos HTML: etiquetas y atributos

Cuando trabajamos con XHTML/HTML lo haremos con elementos, etiquetas y atributos. Ejemplo:



Un **elemento** es un componente del documento que tiene un significado semántico (un párrafo, una cabecera, un listado, etc.). Las partes principales por las que están formados los elementos son:

- **La etiqueta de apertura:** consiste en el nombre del elemento (en este caso es la `p` de párrafo), encerrado por paréntesis angulares (`<` `>`) de apertura y cierre. Esta etiqueta indica dónde comienza el elemento, en este caso el párrafo.
- **La etiqueta de cierre:** es igual que la etiqueta de apertura, excepto que incluye una barra de cierre (`/`) antes del nombre de la etiqueta. Establece dónde termina el elemento, en este caso el párrafo.
- **El contenido:** este es el contenido del elemento, que en este caso es sólo texto ("Esto es un párrafo").
- **El elemento:** la etiqueta de apertura, más la etiqueta de cierre, más el contenido equivalente al elemento.



Los atributos contienen información adicional acerca del elemento, en el ejemplo anterior *class* es el nombre del atributo y *saludo* el valor del atributo.

Un atributo siempre debe tener:

1. Un espacio entre este y el nombre del elemento (o del atributo previo si hay más).
2. El nombre del atributo seguido del signo =
3. Comillas de apertura y de cierre, encerrando el valor del atributo.

Los atributos siempre se encuentran en la etiqueta de apertura, **nunca** en la de cierre.

## 2.1 Anidar elementos

Podemos también tener elementos dentro de otros elementos (anidamiento). En el siguiente ejemplo vamos a poner en negrita las palabras *se llama* del párrafo.

```
<p>Mi perra <strong>se llama</strong> Lola</p>
```

Debemos asegurarnos que los elementos están correctamente anidados, es decir, el último elemento que abrimos deberá ser el primero en cerrar, y así sucesivamente. En el siguiente ejemplo donde la negrita se le pone a la palabra *Lola* **es incorrecto** al haber cerrado primero el párrafo y luego la negrita.

```
<p>Mi perra se llama <strong>Lola</p></strong>
```

Los elementos deben abrirse y cerrarse ordenadamente. Si estos se encuentran solapados el navegador web intentará adivinar lo que intentas hacer y puede que el resultado no sea el deseado.

## 2.2 Elementos vacíos

Algunos elementos no poseen contenidos, y son llamados elementos vacíos. Por ejemplo, el elemento imagen.

```

```

El ejemplo imagen anterior tiene dos atributos, pero no hay etiqueta de cierre `</img>` ni contenido encerrado. Esto es porque es un elemento de imagen no encierra contenido.

## 3 Estructura básica de un documento HTML

En general, cualquier página web en HTML debe tener la extensión `.html` o `.htm`. El contenido de una página por un navegador, encargado de mostrar el contenido de la misma.

Un documento HTML tiene una estructura básica similar a la siguiente:

```
<!DOCTYPE ...>
<html ...>
  <head>
    ...
  </head>
  <body>
    ...
  </body>
</html>
```

En esta estructura nos encontramos:

- **<!DOCTYPE html>**: Nos indica el tipo de documento que es, es obligatorio.
- **<html>...</html>**: Este elemento encierra todo el contenido de la página entera, se le conoce como el elemento raíz. Es recomendable incluir dentro de la etiqueta html el atributo *lang* seguido del idioma de la página. En el siguiente ejemplo veremos cómo indicar que el documento está en español (*lang="es"*), pero este atributo no es obligatorio, hoy en día desde la versión HTML5 este atributo se puede prescindir y el documento validaría y funcionaría correctamente.

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

- **<head>...</head>**: Actúa como un contenedor de todo aquello que queremos incluir en la web pero que no es contenido visible (metadatos). Dentro de este elemento se incluyen las *keywords*, las hojas de estilo CSS, la declaración de caracteres, etc.
- **<body>...</body>**: Encierra todo el contenido que desees mostrar, ya sea texto, imágenes, etc.

### 3.1 DOCTYPE

Como hemos visto en el punto anterior es lo primero que aparece en un documento html y sirve para indicar qué versión del lenguaje se está utilizando.

Dependiendo de la versión que estemos utilizando las declaraciones de las distintas versiones de HTML o XHTML son:

#### Para HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

#### Para XHTML 1.1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

## Para HTML5

```
<!DOCTYPE html>
```

Como vemos, desde la versión HTML 5, la declaración es muy sencilla.

### 3.2 Elementos de la cabecera

La siguiente parte del documento es la sección `<head>`, donde deberemos incluir:

- `<title>`: El título de la página web
- `<meta>`: Las metaetiquetas que describen el documento, la codificación y la información a la que pueden acceder los motores de búsqueda (keywords)
- Las referencias a archivos externos que necesita la web como hojas de estilos(`<link>`) o scripts(`<script>`), estos elementos los veremos en las siguientes unidades.

#### 3.2.1 Las metaetiquetas

Las metaetiquetas se utilizan para indicar el juego de caracteres, la descripción de la página, las palabras clave, el autor del documento y la configuración del área visible e una página (`viewport`), que será diferente dependiendo del dispositivo.

Las metaetiquetas utilizan dos atributos: `name` para indicar el nombre de la metaetiqueta y `content`, para indicar su contenido, a excepción de la metaetiqueta de la codificación de caracteres que sólo utiliza el atributo `charset`.

```
<meta charset="UTF-8">
<meta name="description" content="Mi primera página web en html5">
<meta name="keywords" content="web html5">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

##### 3.2.1.1 *charset*

Define la codificación de caracteres que se utiliza en el documento. Este es otro elemento que se ha simplificado, antiguamente hacíamos esto:

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

En HTML5, se reduce a:

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

La codificación predeterminada en HTML5 es UTF-8, que cubre casi todos los caracteres, signos de puntuación y símbolos del mundo.

Es importante que la declaración de caracteres se encuentre dentro de los 512 primeros caracteres del documento. También es importante que esté antes de cualquier elemento de contenido (como el `<title>`). Una buena práctica sería poner la codificación como la primera etiqueta dentro del bloque `<head>`.



### 3.2.1.2 *description*

Se utiliza para describir brevemente el contenido de una página web. La descripción aparece debajo del título y URL de la página en los resultados del motor de búsqueda. Para permanecer visible dentro de Google debe tener entre 140-160 caracteres. Se recomienda que para tener un buen posicionamiento tenga una descripción de forma significativa y que incluya alguna de sus palabras claves.

```
<meta name="description" content="Mi primera página web en html5">
```

### 3.2.1.3 *keywords*

Define una serie de palabras relacionadas con el contenido de la página web que pueden resultar útiles para los motores de búsqueda, aunque su uso cada vez tiene menos influencia en el posicionamiento web, por ejemplo, Google lo ignora.

```
<meta name="keywords" content="web html5">
```

### 3.2.1.4 *author*

Indica el nombre del autor de la página web.

```
<meta name="author" content="Rosa Medina">
```

### 3.2.1.5 *viewport*

Configura el área visible de la página web para que se visualice bien en todos los dispositivos.

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

Con el anterior código estamos indicándole al navegador:

- `width=device-width`: Ajusta el ancho de la página al ancho de la pantalla del dispositivo.
- `initial-scale=1.0`: Establece el nivel de zoom inicial cuando el navegador carga la web por primera vez. Al poner 1 no se aplica zoom, si por ejemplo no queremos que se pueda hacer zoom se pondría: `maximum-scale=1.0`

## 3.2.2 Elemento title

El elemento `<title>` es el título de la página web, se escribe entre las etiquetas `<title>` y `</title>` y se muestra en la barra de título del navegador.

```
<title>Mi primera página web</title>
```

## 4 Ejemplo de página web

Un ejemplo de una página web sencilla sería:

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="description" content="Mi primera página web en html5">
  <meta name="keywords" content="web html5">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Mi primera página web</title>
</head>
<body>
  <p>Mi perra se llama <strong>Lola</strong></p>
</body>
</html>

```

## 5 Comentarios en HTML

HTML permite insertar comentarios en el código, estos comentarios no se visualizarán en el navegador. Podemos utilizar los comentarios para explicar nuestro código, o para delimitar/marcar una franja de nuestro código.

Un comentario en HTML comienza con el símbolo `<!--` y termina con el símbolo `-->`. Lo que hay en medio de los dos símbolos será el comentario que tendremos visible sólo en el editor de texto que usemos para crear/modificar la web. Por ejemplo:

```

<!DOCTYPE html>
<!-- El idioma de la web es opcional, en este caso se incluye e indicamos que está en español -->
<html lang="es">
<head>
  <!-- Primero ponemos el juego de caracteres utilizado -->
  <meta charset="UTF-8">
  <!-- Insertamos el resto de metadatos -->
  <meta name="description" content="Mi primera página web en html5">
  <meta name="keywords" content="web html5">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Ponemos el título de la web -->
  <title>Mi primera página web</title>
</head>
<body>
  <!-- En el cuerpo de nuestra web, y por tanto lo que nos mostrará el editor
  ponemos un párrafo -->
  <p>Mi perra se llama <strong>Lola</strong></p>
</body>
</html>

```

## 6 xHTML

xHTML es una variante de los documentos HTML asociada a XML, es decir, son documentos HTML expresados como XML válido. Para escribir documentos xHTML tendríamos que modificar la estructura base vista anteriormente.

El primer cambio sería que el documento debería comenzar con:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Con esto, estamos indicando la versión de XML y la codificación del documento. Tras esta línea pondríamos el DOCTYPE que en este caso es diferente a HTML5, y tenemos que especificarle el documento DTD que se utilizará para verificar si es válido el documento.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Y la última diferencia es la etiqueta html, que debe tener información sobre el espacio de nombres a utilizar, es decir, el conjunto de etiquetas válidas que podemos emplear en el documento. Además, indicamos el idioma de la página en el atributo *lang* y también en el atributo *xml:lang*.

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
```

El resto del documento es prácticamente igual, sólo debemos tener en cuenta las etiquetas de XHTML que no coinciden con las que HTML5 permite.

### Ejercicio 1:

Crea en VSCode un documento HTML con la estructura básica para que tu web contenga:

- Un título que sea: "Mi primera página web HTML"
- Una codificación UTF-8
- Información acerca del autor
- Una breve descripción
- Que la configuración de la web al cargarse no sea con zoom
- Un párrafo con tu nombre
- Un comentario al final del todo que indique si has podido visualizar el documento correctamente en el navegador o no

### Ejercicio 2:

En el siguiente documento HTML hay errores, modifícalo para que sea correcta:

```
<html>
<head lang="es">
<meta name="charset" content="utf-8">
<meta content="author" name="Rosa Medina">
</head>
<title>Ejercicio 2 - Tema 2 - LM</title>
<body><p>¡Hola mundo!</body></p>
</html>
```

## 7 Principales etiquetas HTML

### 7.1 Encabezados y párrafos

En HTML, cada párrafo tiene que estar delimitado por un elemento `<p>`, por ejemplo:

```
<p>Había una vez, una perrica llamada Lola</p>
```

Dentro de un párrafo podemos generar un salto de línea con la etiqueta `<br>`. Sólo se usará para hacer un salto de línea simple, nunca para separar párrafos. Esta etiqueta es necesario ponerla ya que HTML interpreta los saltos de línea en el código como espacios en blanco

```
<p>Verso 1, <br> Verso 2, <br> Verso 3.</p>
```

Cada sección tiene que estar delimitada por un elemento encabezado:

```
<h1>La historia de mi perra Lola</h1>
```

Hay seis elementos de encabezado: `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` y `<h6>`. Cada elemento representa un nivel de contenido diferente en el documento, `<h1>` representa el título principal, `<h2>` el subtítulo, `<h3>` representa el subtítulo del subtítulo y así sucesivamente. Es decir, `<h1>` es el encabezado de mayor importancia y `<h6>` es el de menor.

Por ejemplo, en esta historia `<h1>` representa el título de la historia principal, `<h2>` representa el título de cada capítulo, y `<h3>` representa las diferentes secciones del capítulo y así sucesivamente.

```
<h1>La historia de mi perra Lola</h1>
<p>Por Rosa Medina</p>
<h2>Capítulo 1: Buscamos un perro</h2>
<p>Era una noche oscura, se había ido la luz tras la
fuerte tormenta...</p>
<h2>Capítulo 2: Contactamos con los dueños</h2>
<p>Paqui tenía puesto en FB una manada de perros.</p>
<h3>La hija de Paqui</h3>
<p>La hija de Paqui me facilitó el teléfono....</p>
```

## La historia de mi perra Lola

Por Rosa Medina

### Capítulo 1: Buscamos un perro

Era una noche oscura, se había ido la luz tras la fuerte tormenta...

### Capítulo 2: Contactamos con los dueños

Paqui tenía puesto en FB una manada de perros...

#### La hija de Paqui

La hija de Paqui me facilitó el teléfono....

Como podemos observar el `<h1>` es el más grande y por tanto más importante y a medida que hemos incrementado, los títulos han sido más pequeños.

### Ejercicio 3:

Crea una página web utilizando los 6 encabezados y la etiqueta párrafo vistos anteriormente. El resultado a obtener es el siguiente:

# Encabezado 1

## Encabezado 2

### Encabezado 3

#### Encabezado 4

##### Encabezado 5

###### Encabezado 6

Soy un párrafo!

#### 7.1.1 Resaltado de texto: Negrita y cursiva

En HTML existen dos tipos principales de resaltado o enfatizado de texto: negrita y cursiva.

Hace unos años, se utilizaba la etiqueta `<b>` para poner el texto en negrita, sin embargo esta etiqueta no aporta ningún significado especial por lo que su uso está desaconsejado. En su lugar, se recomienda utilizar la etiqueta `<strong>`, que también muestra el texto en negrita pero que además nos indica que ese texto es muy importante. Al haber elegido la etiqueta `<strong>` estamos haciendo que los navegadores de voz lean ese texto dándole el énfasis correspondiente.

```
<!DOCTYPE html>
<html lang="es">
<head>...</head>
<body>
  <p>Cuento contigo. <strong>¡No llegues tarde!</strong></p>
</body>
</html>
```

Del mismo modo, elemento `<i>` pone el texto en cursiva, pero no aporta ningún significado especial esta etiqueta y se desaconseja. En su lugar, se utiliza la etiqueta `<em>` que indica a los lectores de pantalla que el texto debe tener un mayor énfasis. Por ejemplo:

```
<!DOCTYPE html>
<html lang="es">
<head>...</head>
<body>
  <p>Me alegro de que no llegues tarde</p>
  <p>Me <em>alegro</em> de que no llegues <em>tarde</em></p>
</body>
</html>
```

La primera frase suena aliviada porque la persona no llega tarde. Por lo contrario, la segunda suena sarcástica y un tanto pasivo-agresiva, expresa molestia porque la persona ha llegado algo tarde.

El navegador, de manera predeterminada, aplica el estilo de letra itálica, pero no debes utilizar esta etiqueta solamente para establecer el estilo de letra itálica. Para usar ese estilo, debes utilizar únicamente la etiqueta del elemento `<span>` y algo de CSS.

#### Ejercicio 4:

Crea una página web llamada deportes que tenga lo siguiente:

- El título: "Los deportes de *tu nombre*" (en la etiqueta title)
- Un encabezado principal (h1) con el texto: "Mis deportes favoritos"
- Un encabezado secundario (h2) con el texto: "Tenis", donde deberás poner en uno o dos párrafos información sobre este tipo de deporte: con qué se juega, los tipos de pistas y los deportistas más conocidos (Si alguna información no conoces puedes inventártela)
- Otro encabezado secundario (h2) con el texto de: "Fútbol", que tendrá dos subapartados (h3) uno llamado "La liga" donde pondremos información sobre algunos de los equipos de primera división, sus jugadores y ligas ganadas. Y otro párrafo llamado "Mundiales" con dos párrafos con algún equipo que haya participado en el último mundial, jugadores y en qué posición quedó (Toda esta información te la puedes inventar)
- Debes utilizar las etiquetas propias para el resaltado del texto en aquella información que consideres como palabra(s) clave(s).

##### 7.1.1.1 Otras etiquetas

Hay muchas otras etiquetas que se utilizan para formatear el código como: `<u>` que se utiliza para subrayar el texto, `<sup>` para crear un superíndice, y `<sub>` un subíndice. Más info [aquí](#) y a en el punto de [otras etiquetas de texto](#).

### 7.1.2 La etiqueta span

El elemento `<span>` es un elemento que no tiene semántica, se utiliza para delimitar contenido cuando se le quiere aplicar CSS (o tratarlo con JS) sin proporcionarle ningún significado extra. Por ejemplo:

```
<!DOCTYPE html>
<html lang="es">
<head>...</head>
<body>
  <p>El boli <span style="color: red;">rojo</span> no funciona</p>
</body>
</html>
```

**Nota:** En la siguiente unidad se estudiará CSS y por tanto será ahí cuando estudiemos qué es el atributo `style` que le hemos puesto a la etiqueta `<span>`.

### 7.1.3 Inclusión de caracteres especiales en HTML

EN HTML, los caracteres `<`, `>`, `"`, `'` y `&` son caracteres especiales. Forman parte de la sintaxis HTML. Entonces, ¿cómo podemos incluir estos caracteres en nuestro texto?

Para incluir estos símbolos especiales, debemos utilizar una serie de códigos que representan caracteres. Cada referencia de carácter comienza con un signo de ampersand (`&`) y finaliza con un punto y coma (`;`)

Carácter literal	Equivalente de referencia de caracteres
<code>&lt;</code>	<code>&amp;lt;</code>
<code>&gt;</code>	<code>&amp;gt;</code>
<code>"</code>	<code>&amp;quot;</code>
<code>'</code>	<code>&amp;apos;</code>
<code>&amp;</code>	<code>&amp;amp;</code>

Por ejemplo, los siguientes párrafos:

```
<p>En HTML, defines un párrafo con el elemento<p>.</p>
<p>En HTML, defines un párrafo con el elemento &lt;p&gt;.</p>
```

En el navegador se visualizaría la siguiente imagen donde podemos ver que el navegador interpreta la segunda instancia de `<p>` como el inicio de nuevo párrafo. El segundo se ve bien porque hemos remplazado `<` y `>` por sus referencias correspondientes.

En HTML, defines un párrafo con el elemento

.

En HTML, defines un párrafo con el elemento `<p>`.

Para obtener más información sobre la referencia de entidad ver [aquí](#). Si nos fijamos **no necesitamos** ninguna referencia más para ningún otro carácter porque los navegadores modernos gestionan estos símbolos si hemos establecido nuestro HTML con la codificación UTF-8.

## 7.2 Elementos contenedor

En HTML disponemos de dos tipos de elementos: elementos en bloque y elementos en línea. Un elemento en bloque es aquel que incorpora un salto de línea antes y después del mismo, mientras que un elemento en línea se colocará a continuación del elemento anterior, sin introducir ningún salto de línea.

Ejemplo de elementos en línea: `<span>`, `<a>`, `<img>`, `<code>`, etc. Por otro lado, algunos de los elementos en bloque son: `<div>`, `<p>`, `<h1>`, `<li>`, etc.

### 7.2.1 Elemento div

El elemento `<div>` se puede utilizar como un contenedor para agrupar otros elementos HTML. Este elemento no tienen ningún significado especial, salvo que, por tratarse de un elemento a nivel de bloque, el navegador mostrará un salto de línea antes y después de él, pero semánticamente no aporta nada al documento.

Cuando se utiliza con CSS, el elemento `<div>` se puede utilizar para establecer atributos de estilo para grandes bloques de contenido. Otro uso común del elemento `<div>` es para el diseño del documento. Sustituyendo al anterior método, que utilizaban tablas para definir el diseño de la página.

```
<div class="container">
  <h1>Título de mi web</h1>
  <p>Elementos de una página web</p>
  <p>...</p>
</div>
```

El elemento `<span>` puede utilizarse como un contenedor para texto. Al igual que `<div>`, no tiene ningún significado especial. Cuando se utiliza junto con CSS, se puede utilizar para establecer los atributos de estilo a partes del texto.

## 7.3 Listas

Las listas están en todos los aspectos de nuestra vida, desde lista de la compra, lista de tareas, etc. Debemos distinguir entre tres tipos de listas:

### 7.3.1 Listas no ordenadas

Las listas no ordenadas se usan para marcar listas de artículos cuyo orden no es importante. Por ejemplo, la lista de una compra. Cada lista desordenada comienza con un elemento `<ul>` (unordered list) que delimita todos los elementos de la lista, y cada uno de los artículos de la lista está delimitado con un elemento `<li>` (list item).



```
<ul>
  <li>leche</li>
  <li>huevos</li>
  <li>pan</li>
  <li>chocolate</li>
</ul>
```

## Lista no ordenada

- leche
- huevos
- pan
- chocolate

### 7.3.2 Listas ordenadas

Las listas ordenadas son aquellas en las que el orden de los elementos sí importa. Por ejemplo, las instrucciones para seguir la ruta para llegar al instituto.

La estructura del marcado es la misma que para las listas ordenadas, excepto que debes delimitar los elementos de la lista con una etiqueta `<ol>` (ordered list), en lugar de `<ul>`.

## Lista ordenada

```
<ol>
  <li>Conduce hasta el final de la calle</li>
  <li>Gira a la derecha</li>
  <li>Sigue derecho por las dos primeras gloriets</li>
  <li>Gira a la izquierda en la tercer glorieta</li>
  <li>El instituto está a tu derecha, 300 metros más adelante</li>
</ol>
```

1. Conduce hasta el final de la calle
2. Gira a la derecha
3. Sigue derecho por las dos primeras gloriets
4. Gira a la izquierda en la tercer glorieta
5. El instituto está a tu derecha, 300 metros más adelante

Podemos personalizar la lista con los siguientes atributos:

- **type**: indica el tipo de viñeta o marcador que se utiliza en la lista: **1** (enteros), **A** (letras mayúsculas), **a** (letras minúsculas), **I** (números romanos en mayúsculas), **i** (números romanos en minúscula)
- **start**: se utiliza para determinar el valor inicial, debe ser un número entero válido.
- **reversed**: es un atributo booleano (verdadero o falso). Indica que la lista es descendente (... , c, b, a) mientras que si se omite indica que la lista es ascendente (a, b, c, ...)

Si no indicamos ninguno de estos tres atributos se crea una lista ordenada de números enteros empezando desde el 1.

```
<ol reversed start="6" type="A">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

## Lista ordenada invertida

- F. Coffee
- E. Tea
- D. Milk

Además de estos atributos que le podemos dar a la lista, podemos modificar el valor de un determinado ítem de la lista (el valor del número de la lista), para ello utilizamos el atributo **value**.

### 7.3.3 Listas anidadas

Las listas se pueden anidar, es decir, podemos incluir una lista dentro de otra y cada una tendrá su propia numeración o viñeta. Para anidarlas, crearemos otra lista (con `<ol>` o `<ul>`) dentro de una etiqueta `<li>`, y cerraremos `</li>` después de cerrar previamente la definición de la sublista. Ejemplo:

```
<ol>
  <li> Opción 1 </li>
  <li> Opción 2 <!-- La etiqueta de cierre no va aquí -->
    <ul>
      <li> Opción 2A </li>
      <li> Opción 2B </li>
      <li> Opción 2C </li>
    </ul>
  </li> <!-- Aquí está la etiqueta de cierre -->
  <li> Opción 3 </li>
  <li> Opción 4 </li>
</ol>
```

#### Listas anidadas

1. Opción 1
2. Opción 2
  - Opción 2A
  - Opción 2B
  - Opción 2C
3. Opción 3
4. Opción 4

### 7.3.4 Listas de descripción: etiquetas dl, dt y dd

Son listas donde se asocian términos con descripciones o valores, por ejemplo, glosarios, listas de vocabulario, listas de metadatos o listas de preguntas y respuestas, no es adecuado para representar diálogos.

Para representar o definir estas listas utilizamos el elemento `<dl>`, donde cada término está envuelto en un elemento `<dt>` (término de descripción) y cada descripción está envuelta en un elemento `<dd>` (definición de descripción). Se puede asociar un término con múltiples descripciones y viceversa. El navegador muestra el contenido del texto `<dd>` con una sangría.

```
<dl>
  <dt>Coffee</dt>
  <dd>Black hot drink</dd>
  <dt>Milk</dt>
  <dd>White cold drink</dd>
</dl>
```

#### Lista de descripción

Coffee  
     Black hot drink  
 Milk  
     White cold drink

#### Ejercicio 5:

Crea la siguiente lista:

#### Ejercicio 5:

- II. Desayunar
- III. Lavarme los dientes
- IV. vestirme
- V. ...
- VIII. Coger la mochila

**Nota:** Fíjate en el valor que tiene el último ítem de la lista

**Ejercicio 6:**

Crea la siguiente lista:

## Ejercicio 6

- **Herramientas impresas**
  - *Libros*
  - Revistas
- Herramientas electrónicas
  - CD-ROM
  - Wikipedia
- Herramientas on-line
  - Páginas de recursos
    - De organismos públicos
    - De universidades
    - WTF

**Nota:** Fíjate en la negrita, cursiva y subrayado que hay

**Ejercicio 7:**

Crea la siguiente lista:

## Estudios ofertados:

1. ESO
2. Bachiller
3. Formación profesional
  - Grado superior
  - Grado medio

## 7.4 Enlaces

Una parte fundamental de la web es que los documentos estén enlazados entre sí, es decir, que podamos ir de una página a otra con un simple clic. La etiqueta utilizada para crear enlaces usaremos la etiqueta `<a>`. El texto o imagen introducido dentro de esta etiqueta, reaccionará ante un clic de ratón llevándonos al destino especificado. Esta etiqueta utiliza el atributo `href` para indicar la url de destino del enlace.

```
<a href="https://iessanvicente.com/">Ir a la web del IES</a>
```

Otro de los atributos que admite la etiqueta `<a>` es `title`, este atributo nos permite especificar información extra sobre el elemento. Esta información se muestra cuando ponemos el ratón sobre este elemento. En el ejemplo anterior, podría ser algo así: `title="URL del instituto"`.

Si queremos crear un enlace que en lugar de ir a url determinada vaya a un punto determinado de nuestro documento actual, haremos:

1. Marcar el elemento a dónde queremos ir, para ello usaremos el atributo `id` y le asignamos un valor sin espacios.
2. En el enlace ponemos el símbolo `#` seguido del `id` al que queremos que vaya.

```
<h1 id="sobreNosotros">Sobre nosotros</h1>
<p>Somos una empresa...</p>
<p>...</p>
<a href="#sobreNosotros">Ir a Sobre nosotros</a>
```

**Nota:** El id debe ser un valor único en la página, es un identificador. Si queremos que vaya a una sección de otra página sería: `href="pagina2.html#sobreNosotros"`

Tenemos la posibilidad de abrir un enlace en una ventana o pestaña nueva, para ello utilizaremos el atributo `target` dándole el valor de `_blank`. Más opciones [aquí](#).

Para enlazar una dirección de correo electrónico, se indica con `mailto`: seguido de la dirección de correo.

```
<p><a href="mailto:rmedina@iessanvicente.com" title="Correo profe">Envía un correo</a></p>
```

### 7.4.1 Ruta absoluta y ruta relativa

Si estamos desarrollando nuestra aplicación en un directorio llamado lenguajes que tiene el siguiente contenido:

```
lenguajes/
├── contacto.html
├── css
│   └── estilos.css
└── index.html
```

Dentro del atributo `href` debemos indicar la ruta a la web que queremos acceder al hacer clic. Esta ruta la podemos indicar de dos modos:

1. Como una ruta absoluta a la raíz de la aplicación. Como `contacto.html` está en la carpeta raíz de la aplicación `lenguajes`, el enlace quedaría así:

```
<a href="/contacto.html" title="Página de contacto">Contacta con nosotros</a>
```

2. Como una ruta relativa desde la carpeta en donde nos situamos. Por ejemplo si nos encontramos en la misma carpeta el enlace sería:

```
<a href="contacto.html" title="Página de contacto">Contacta con nosotros</a>
```

- a) En el caso de estar dentro de una subcarpeta y debemos subir de nivel (acceder a la carpeta padre) debemos utilizar `..` pudiendo enlazarlos para subir de nivel en varias carpetas, por ejemplo `../../` para salir de tres carpetas.

```
<a href="../contacto.html" title="Página de contacto">Contacta con nosotros</a>
```

## 7.5 Imágenes

Para insertar imágenes en nuestra web utilizamos la etiqueta `<img>` con su atributo `src` para indicar en dónde se encuentra la imagen. La etiqueta `<img>` es un elemento vacío, por lo que no es necesario que introduzcamos la etiqueta de cierre. Podemos utilizar 3 formatos de imágenes para nuestra página: JPEG, GIF y PNG

- Con **JPEG** conseguimos el menor tamaño, aunque no la mejor calidad. Está recomendado para imágenes medianas o grandes con muchos colores y variedad. No soporta transparencias.
- El formato **GIF** soporta transparencias, pero está muy limitado en colores (256). También nos permite animaciones sencillas.
- El formato **PNG** es el que mejor calidad nos da, además de soportar transparencias. Todo esto a un coste un poco mayor en cuanto tamaño.
- Formato **SVG** es el formato vectorial, este formato es el ideal para imágenes escalables

Se debe poner el atributo **alt** a todas las imágenes, establece un texto alternativo que describa la imagen a mostrar.

```

```

Además de estos dos atributos, la etiqueta **<img>** admite los siguientes atributos:

- **width**: ancho de la imagen (en píxeles o porcentaje). Si no indicamos altura, mantiene proporciones de la imagen. Normalmente se establece por CSS
- **height**: alto de la imagen.

### 7.5.1 Imágenes y enlaces

Para que una imagen nos sirva como enlace, simplemente hay que meterla entre la etiqueta **<a>** como si fuera texto (de hecho, en HTML una imagen es tratada a nivel de texto en muchos aspectos, no es un bloque independiente).

```
<a href="https://iessanvicente.com/">
  
</a>
```

Si cargamos una imagen grande (Ej: 200KB) y hacemos que se vea mucho más pequeña, no conseguimos nada a nivel de ahorro en ancho de banda. El navegador cargará la imagen entera (los 200KB) y luego la escalará.

Para los llamados thumbnails, o previzualizaciones pequeñas, lo mejor es crearse una imagen pequeña con algún programa de edición, a partir de la original que ocupe mucho menos espacio.

### 7.5.2 Favicon (icono de pestaña)

Cuando tenemos varias pestañas en nuestro navegador, para una búsqueda más fácil e intuitiva entre las pestañas que tenemos abiertas, el navegador suele mostrarnos unos iconos a la izquierda del título de cada pestaña, estos iconos se conocen como favicons (iconos de favoritos).

La etiqueta que se utiliza para poner este icono es la etiqueta **<link>** la cual va dentro de la etiqueta **<head>** de nuestro HTML (**no en el body**). El problema que presenta favicon es que cada navegador (y casi cada sistema operativo) la implementa como quiere. Hay navegadores que sólo soportan favicon en formato png, otros que solo soportan resoluciones específicas, etc.

```
<link rel="shortcut icon" href="/favicon.ico" /> <!-- HTML4 -->
<link rel="icon" sizes="64x64" href="/favicon.png" /> <!-- HTML5 -->
<link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon-180x180.png" /> <!--
iPhone/iPad -->
<meta name="msapplication-TileImage" content="/mstile-144x144.png" /> <!-- Windows Phone -->
```

Si queremos crearnos un favicon de forma cómoda podemos utilizar la herramienta [Real Favicon Generator](#), donde le pasamos un logo a alta resolución y nos genera un .zip con el código y diferentes versiones y resoluciones de los iconos.

### 7.5.3 Elemento figure

En ocasiones puede que nos interese añadir un pie de foto a una imagen. Para poderlo tratar como un único elemento (imagen + pie) podemos utilizar la etiqueta `<figure>`, así como la etiqueta `<figcaption>` para definir dicho pie de foto.

```
<figure>
  
  <figcaption>Imagen reciente de la Sagrada Familia</figcaption>
</figure>
```

## 7.6 Tablas

Las tablas se definen en HTML con la etiqueta `<table>`. Una tabla está dividida en filas con la etiqueta `<tr>` y una fila se compone de celdas de datos, mediante la etiqueta `<td>`, aunque también se puede dividir en celdas de cabecera con la etiqueta `<th>`.

Los elementos `<td>` son los contenedores de los datos en la tabla y pueden contener todo tipo de elementos HTML como texto, imágenes, listas, otras tablas, etc.

```
<table>
  <tr>
    <th>Nombre</th>
    <th>Apellidos</th>
    <th>Correo</th>
  </tr>
  <tr>
    <td>Rosa</td>
    <td>Medina</td>
    <td>rmedina@iessanvicente.com</td>
  </tr>
  <tr>
    <td>Carlos</td>
    <td>Bernal</td>
    <td>carlos@iessanvicente.com</td>
  </tr>
</table>
```

Nombre	Apellidos	Correo
Rosa	Medina	rmedina@iessanvicente.com
Carlos	Bernal	carlos@iessanvicente.com

**Nota:** Con solo el código anterior sólo conseguimos la estructura de la tabla, si queremos que nuestra tabla tenga el mismo aspecto que en la imagen tenemos que aplicar estilos CSS.

Los algunos de los atributos que pueden aparecer en la etiqueta `<table>` son:

- **summary:** es parecido al atributo `alt` de las imágenes. Proporciona na descripción de la tabla para navegadores en modo texto, o invidentes.
- **border:** el valor es numérico y establece el grosor del borde de la tabla (0 para quitarlo)
- **cellpadding:** espacio que debe quedar entre las propias celdas (entre los bordes)
- **width:** ancho de la tabla (suele usarse con CSS)
- Otros atributos como `align`, para alinear la tabla o `bgcolor` para establecer el color de fondo de la tabla están desaconsejados, mejor establecerlos con CSS.

Por otro lado, los atributos que admiten los elementos `<td>` (aunque es mejor aplicarlos con CSS) son:

- **align:** alinea el texto de la celda en horizontal. Puede tener los valores `left`, `right` o `center`.
- **valign:** alinea el texto de la celda en vertical. Puede tener los valores `top`, `bottom` o `middle`.
- **rowspan:** expande la celda x filas hacia abajo.
- **colspan:** expande la celda x filas a la derecha
- **width:** indica la anchura específica de esa columna, su valor viene dado en px ó %
- **height:** altura de la celda, en px ó %
- **bgcolor:** color de fondo de la celda (o de la tabla si se aplica en `<table>`)

```
<table border="1">
  <tr>
    <td rowspan="2" valign="middle" align="right" width="50%">Lenguajes de marcas</td>
    <td>Entornos</td>
    <td>Programación</td>
    <td>FOL</td>
  </tr>
  <tr>
    <td>lunes</td>
    <td colspan="2" align="right">martes</td>
  </tr>
  <tr>
    <td colspan="4" align="center">Primera evaluación</td>
  </tr>
</table>
```

Lenguajes de marcas	Entornos	Programación	FOL
	lunes	martes	
Primera evaluación			

Como podemos observar con el atributo **colspan** podemos hacer que una celda ocupe el espacio equivalente a 2 o más celdas. El valor será un número con las columnas que queramos que ocupe. Ejemplo:

```
<table summary="Precio de venta" border="1">
  <tr>
    <td colspan="2">Ventas mayo 2021</td>
  </tr>
  <tr>
    <td>Fruta</td>
    <td>€/Kg</td>
  </tr>
</table>
```

Ventas mayo 2021	
Fruta	€/Kg

También podemos hacer que una celda ocupe varias filas (en ese caso también tendremos que poner una celda menos en la fila ocupada) para ello usaremos el atributo **rowspan**

```
<table summary="Precio de venta" border="1">
  <tr>
    <td colspan="3">Ventas del 2021</td>
  </tr>
  <tr>
    <td rowspan="2">Mayo</td>
    <td>Manzanas</td>
    <td>280€</td>
  </tr>
  <tr>
    <td>Peras</td>
    <td>385€</td>
  </tr>
</table>
```

Ventas del 2021		
Mayo	Manzanas	280€
	Peras	385€

Además de estas etiquetas vistas para tablas, tenemos:

<caption>	Permite asignar una breve descripción a la tabla. Es un atributo de <b>&lt;table&gt;</b>
<thead>	Define el encabezado de una tabla
<tbody>	Define el cuerpo de una tabla
<tfoot>	Define el pie de la tabla



```

<table border="1">
  <caption>Ahorros del 2021</caption>
  <thead>
    <tr>
      <th>Month</th>
      <th>Savings</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>January</td>
      <td>$100</td>
    </tr>
    <tr>
      <td>February</td>
      <td>$80</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td align="right">Sum</td>
      <td>$180</td>
    </tr>
  </tfoot>
</table>

```

Ahorros del 2021

Month	Savings
January	\$100
February	\$80
Sum	\$180

## 7.7 Otras etiquetas

Además de las etiquetas vistas anteriormente para introducir texto, tenemos otras como son las siguientes:

### 7.7.1 Citas: etiquetas cite, blockquote y q

La etiqueta `<cite>` sirve para marcar una referencia a una fuente o el autor de un texto citado.

```

<p> <cite>Don Quijote</cite> dijo: "En un lugar de la Mancha..." </p>

```

En navegadores basados en Mozilla (y en otros) el contenido de cite es mostrado con un estilo de texto en cursiva. Es importante saber que este elemento no se debe utilizar para marcar el texto de las citas, es para marcar el autor.

El elemento HTML `<q>` indica que el texto adjunto es una cita corta en línea. La mayoría de los navegadores muestran el texto entre comillas. Este elemento está destinado a citas breves que no requieren saltos de párrafo, para citas de bloque independiente, se debe utilizar el elemento `<blockquote>`

```
<p>Conforme al sitio web de Mozilla,
  <q cite="https://www.mozilla.org/en-US/about/history/details/">Firefox 1.0
fue lanzado en 2004 y se convirtió en un gran éxito.</q></p>
```

El elemento `<blockquote>` se utiliza para crear citas en bloque, marca las citas a otros autores o documentos.

```
<blockquote cite='http://html.conclase.net/w3c/html401...def-BLOCKQUOTE'>
  <p>
    <strong>Nota.</strong> Recomendamos que las implementaciones de hojas de estilo
    porporcionen un mecanismo para insertar signos de puntuación de citas antes y después de una cita
    delimitada por un BLOCKQUOTE de un modo apropiado según el contexto
    del idioma actual y el grado de anidamiento de las citas.
  </p>
</blockquote>
```

Tanto para el elemento `<q>` como para `<blockquote>`, se utiliza el atributo `cite` para proporcionar un enlace al documento original o fuente.

### 7.7.2 Texto preformateado: etiquetas `pre` y `code`

La etiqueta `<pre>` se utiliza para mostrar texto respetando el formato con el que está escrito, es decir, respetando los espacios en blanco, saltos de línea, etc. Esta etiqueta es muy habitual utilizarla junto con la etiqueta `<code>`.

```
<!-- Un poco de código CSS -->
<pre> body{
  color:  red;}
a  {
  color:green;
}
</pre>
```

La etiqueta `<code>` es apropiada para marcar el código de un programa. Por ejemplo:

```
<code>
public class HolaMundo {
  public static void main (String[] args) {
    System.out.println("Hola mundo");
  }
}
</code>
```

```
public class HolaMundo { public static void main (String[] args) { System.out.println("Hola mundo"); } }
```

Si el ejemplo anterior lo hacemos utilizando las dos etiquetas (`<pre>` y `<code>`) el resultado a mostrar en el navegador sería muy diferente:

```
<pre>
  <code>
    public class HolaMundo {
      public static void main (String[] args) {
        System.out.println("Hola mundo");
      }
    }
  </code>
</pre>
```

```
public class HolaMundo {
    public static void main (String[] args) {
        System.out.println("Hola mundo");
    }
}
```

### 7.7.3 Marcado de texto: etiqueta mark

El elemento `<mark>` representa un texto marcado o resaltado como referencia o anotación, debido a su relevancia o importancia en un contexto particular. Es un fragmento de texto que parece estar subrayado con un rotulador amarillo. Ejemplo:

```
<blockquote>It is a period of civil war. Rebel spaceships, striking from a hidden base, have won their first victory against the evil Galactic Empire. During the battle, <mark>Rebel spies managed to steal secret plans</mark> to the Empire's ultimate weapon, the DEATH STAR, an armored space station with enough power to destroy an entire planet.
</blockquote>
```

It is a period of civil war. Rebel spaceships, striking from a hidden base, have won their first victory against the evil Galactic Empire. During the battle, Rebel spies managed to steal secret plans to the Empire's ultimate weapon, the DEATH STAR, an armored space station with enough power to destroy an entire planet.

### 7.7.4 Borrados e inserciones: etiquetas del e ins

El elemento `<del>` marca las partes de un texto o documento que han sido suprimidas o sustituidas. Nos muestra esa parte como tachada con una línea. Por otro lado, el elemento `<ins>` marca las partes de un texto que han sido añadidas a un documento. Veamos un ejemplo:

```
<p>El agua es insípida <del>y húmeda.</del> <ins>inodora e incolora.</ins></p>
```

El agua es insípida ~~y húmeda.~~ inodora e incolora.

### 7.7.5 Etiqueta delimitadora: br y hr

El elemento `<br>` produce un salto de línea en el texto (retorno de carro). Es útil para escribir un poema o dirección donde la división de las líneas es significativa.

```
<p>IES San Vicente<br> Calle Lillo Juan<br> San Vicente del Raspeig</p>
```

IES San Vicente  
Calle Lillo Juan  
San Vicente del Raspeig

Por otro lado, la etiqueta `<hr>` representa un cambio de tema entre párrafos. En versiones previas de HTML se representaba una línea horizontal, todavía puede representarse como una línea en los navegadores visuales, pero ahora es definida en términos semánticos y no tanto en términos representativos, por tanto para representar una línea horizontal deberíamos utilizar CSS.

```
<p>Este es un párrafo únicamente visual para ver el uso de la etiqueta hr</p>
<hr>
<p>Este es el segundo párrafo, separado del primero por la etiqueta hr, que puede representarse
mediante una línea horizontal.</p>
```

Este es un párrafo únicamente visual para ver el uso de la etiqueta hr

---

Este es el segundo párrafo, separado del primero por la etiqueta hr, que puede representarse mediante una línea horizontal.

### 7.7.6 El elemento time

Las fechas y horarios son partes fundamentales de los contenidos de las páginas web. Los motores de búsqueda son capaces de filtrar los resultados basándose en el tiempo y, en algunos casos, un resultado de búsqueda específico puede recibir más o menos peso en función de cuándo fue publicado por primera vez. El elemento `<time>` representa un período específico en el tiempo. Puede incluir el atributo `datetime` para convertir las fechas en un formato interno legible por un ordenador.

```
<p>El concierto fue el <time datetime="2001-05-15T19:00">15 de Mayo</time>.</p>
```

### 7.7.7 El elemento small

El elemento `<small>` hace el tamaño del texto una “talla” más pequeña que el tamaño mínimo de fuente del navegador. En HTML5, este elemento es reutilizado para representar comentarios laterales y letra pequeña incluyendo derechos de autor y texto legal, independientemente de su estilo de presentación.

```
<p>MDN Web Docs is a learning platform for Web technologies and the software that powers the
Web.</p>
<hr>
```

```
<p><small>The content is licensed under a Creative Commons Attribution-ShareAlike 2.5 Generic License.</small></p>
```

MDN Web Docs is a learning platform for Web technologies and the software that powers the Web.

The content is licensed under a Creative Commons Attribution-ShareAlike 2.5 Generic License.

### 7.7.8 El elemento address

El elemento `<address>` aporta información de contacto para el elemento `<article>` (que veremos más adelante) más cercano o ancestro `<body>`; en el último caso lo aplica a todo el documento.

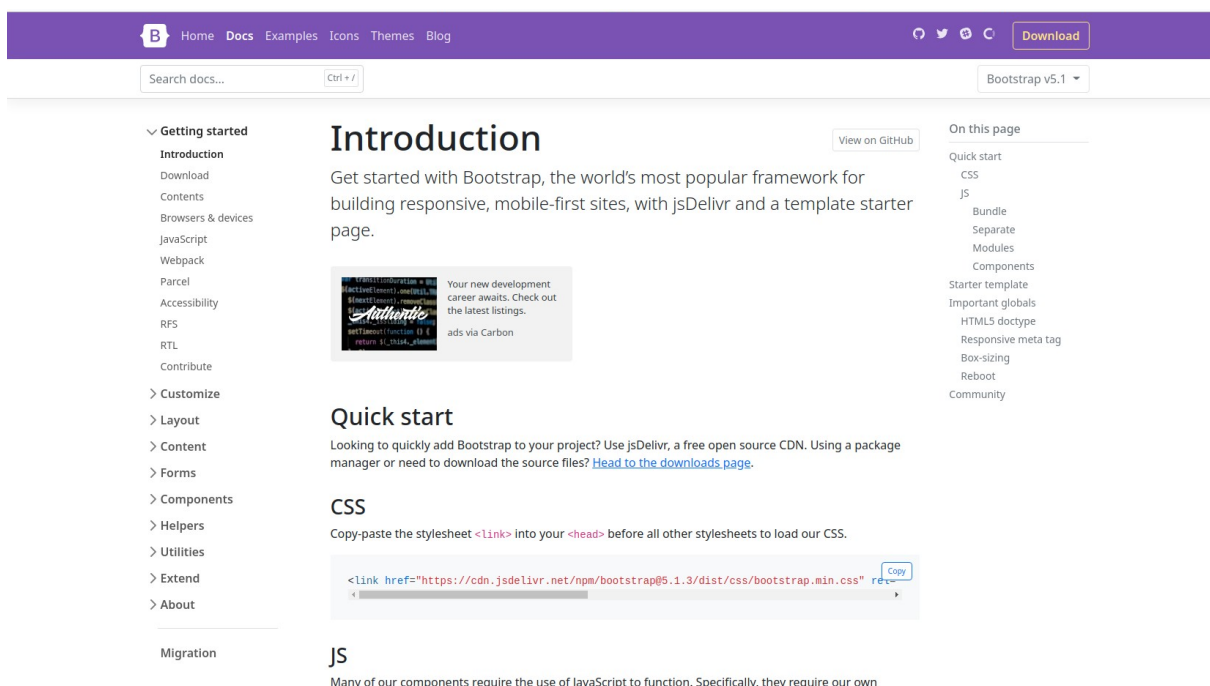
```
<address>Puedes contactar con el autor <a href="https://iessanvicente.com/secretaria.php">https://iessanvicente.com/</a>.<br>O enviando un email a <a href="mailto:rmedina@iessanvicente.com">Correo de contacto</a>.<br>Nos encontramos en:<br>IES San Vicente<br>Calle Lillo Juan<br>San Vicente del Raspeig<br>03690, Alicante<br></address>
```

## 7.8 Etiquetas de secciones

Una página web puede y se deben diferenciar unas de otras, pero todas tienden a tener elementos comunes, una página web normalmente se suele componer de:

- **Encabezado:** Normalmente está formado por una franja en la parte superior de la página donde está el título, y el logo de nuestra web. Esta parte está fija mientras hacemos scroll en la web o mientras navegamos entre las páginas del mismo sitio.
- **Barra de navegación:** Son los enlaces a las distintas secciones de nuestra web. Normalmente está formada por un menú con enlaces. Del mismo modo que el encabezado, esta franja permanece fija en todas las páginas de nuestro sitio web.
- **Contenido principales:** Es la gran parte central de la página y por tanto contiene la mayor parte del contenido de una web. Esta es la parte de la web que realmente varía mucho del resto de páginas de nuestro sitio web.
- **Barra lateral:** Suele incluir algún tipo de información adicional, enlaces, citas, publicidad, etc. Normalmente está relacionada con el contenido de la página, pero a veces puede contener elementos recurrentes como un menú de navegación secundario.
- **Pie de página:** Es la parte inferior de la página, que generalmente contiene la letra pequeña, el copyright o la información de contacto. En esta franja como en el encabezado se coloca la información común, pero en esta parte la información no es tan importante. El pie de página también se suele usar para propósitos SEO e incluye enlaces de acceso rápido al contenido más popular.

Una página web típica podría ser la de [Bootstrap](#):



En versiones anteriores a HTML5, cuando creábamos una página normalmente utilizábamos la etiqueta `<div>` para ir agrupando/separando secciones de una página o incluso se desarrollaban las páginas utilizando `<table>`. Cada una de estas secciones (`<div>`) tenía un identificador (id) para ir identificando cada una de las secciones, lo que hacía que nuestras páginas estuvieran llenas de estos elementos. Ejemplo:

```
<html>
  <head>... </head>
  <body>
    <div id="cabecera">...</div>
    <div id="menu">...</div>
    <div id="contenido">
      <div class="apartado">...</div>
      <div class="apartado">...</div>
    </div>
    <div id="pie">...</div>
  </body>
</html>
```

En HTML5, se introducen una serie de etiquetas de agrupación que funcionan exactamente igual que un `<div>`, pero que además tienen un significado semántico porque indican el contenido que agrupan. Por tanto, en nuestro código HTML debemos crear secciones de contenido basadas en su funcionalidad, podemos utilizar elementos que representen sin ambigüedad las diferentes secciones de contenido descritas, de forma que las tecnologías de accesibilidad y los lectores de pantalla puedan reconocer esos elementos y asistir en tareas como “encontrar el menú de navegación” o “encontrar el contenido principal”.

HTML5 dispone de etiquetas adecuadas que puedes usar para establecer estas secciones semánticas, por ejemplo:

- **encabezado:** `<header>`
- **menú de navegación:** `<nav>`
- **contenido principal:** `<main>`, con varias subsecciones (además de la barra lateral) representadas por los elementos `<article>`, `<section>`, y `<div>`
- **barra lateral:** `<aside>`; a menudo colocada dentro de `<main>`.
- **pie de página:** `<footer>`

### 7.8.1 Elementos de diseño HTML en detalle

El típico diseño utilizando elementos HTML5 de una página web es el siguiente:

<code>&lt;header&gt;&lt;/header&gt;</code>	
<code>&lt;nav&gt;&lt;/nav&gt;</code>	
<code>&lt;main&gt;</code>	<code>&lt;aside&gt;</code>
<code>&lt;article&gt;&lt;/article&gt;</code>	
<code>&lt;article&gt;&lt;/article&gt;</code>	
<code>&lt;article&gt;&lt;/article&gt;</code>	
<code>&lt;article&gt;&lt;/article&gt;</code>	
<code>&lt;/main&gt;</code>	<code>&lt;/aside&gt;</code>
<code>&lt;footer&gt;&lt;/footer&gt;</code>	

Esta tabla muestra el típico ejemplo de una página con los correspondientes elementos HTML5 incluyendo sus etiquetas de apertura y cierre. Si dentro del contenido principal tuviéramos artículos agrupados por diferente temática, cada grupo estaría dentro de un elemento llamado `<section>`

#### 7.8.1.1 Encabezado `<header>`

Esta sección representa un grupo de contenido introductorio. Si este es “hijo” directo del `<body>` se convertirá en el encabezado principal de nuestra web, pero si es hijo de un elemento `<article>` o de un elemento `<section>`, entonces simplemente será el encabezado particular de cada sección. Es importante que no lo confundamos con títulos o encabezados y mucho menos con el `<head>`. Ejemplo:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Página con secciones html5</title>
```

```

</head>
<body>
  <header>
    
    <h1>Titulo principal de nuestra web</h1>
  </header>
  ...
</body>
</html>

```

### 7.8.1.2 Menú <nav>

El elemento `<nav>` contiene la funcionalidad de navegación principal de la página. Los enlaces secundarios, etc. no entrarán en la navegación.

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Página con secciones html5</title>
</head>
<body>
  <header>
    
    <h1>Titulo principal de nuestra web</h1>
  </header>
  <nav>
    <a href="#">Home</a>
    <a href="ventas.html">Ventas</a>
    <a href="sobrenosotros.html">About us</a>
  </nav>
  ...
</body>
</html>

```

### 7.8.1.3 Contenido principal <main>

Este elemento encierra el contenido particular a esta página. Utilizaremos una sola vez la etiqueta `<main>` para cada página, y lo pondremos directamente dentro del `<body>`. Es mejor que no lo anidemos en otros elementos.

Este elemento es muy importante, ya que en el caso de utilizar navegadores de voz, el navegador puede ir directamente al contenido principal, saltando toda la información previa.

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Página con secciones html5</title>
</head>

```



```

<body>
  <header>
    
    <h1>Titulo principal de nuestra web</h1>
  </header>
  <nav>
    <a href="#">Home</a>
    <a href="ventas.html">Ventas</a>
    <a href="sobrenosotros.html">About us</a>
  </nav>
  <main>
    ...
  </main>
  ...
</body>
</html>

```

#### 7.8.1.4 Elemento <article>

Este elemento encuadra un bloque de contenido que tiene sentido por sí mismo aparte del resto de la página (por ejemplo la entrada en un blog)

#### 7.8.1.5 Elemento <section>

Es parecido al elemento <article> pero se usa más para agrupar cada parte de la página que por su funcionalidad constituye una sección en sí misma (Por ejemplo, un conjunto de titulares y resúmenes). Se considera una buena práctica comenzar cada una de estas secciones con un título de encabezado (<h1>, <h2>, ...), pudiendo dividir los artículos (<article>) en distintas secciones (<section>) o también secciones en distintos artículos dependiendo del contexto.

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Página con secciones html5</title>
</head>
<body>
  <header>
    
    <h1>Titulo principal de nuestra web</h1>
  </header>
  <nav>
    <a href="#">Home</a>
    <a href="ventas.html">Ventas</a>
    <a href="sobrenosotros.html">About us</a>
  </nav>
  <main>
    <section>

```

```

    <h2>Venta de bicis</h2>
    <article>
        <!-- Ventas de bicis -->
    </article>
</section>
<section>
    <h2>Venta de componentes</h2>
    <article>
        <!-- Ventas de componentes -->
    </article>
</section>
</main>
...
</body>
</html>

```

### 7.8.1.6 Barra lateral <aside>

El elemento `<aside>` incluye contenido que no está directamente relacionado con el contenido principal, pero que puede aportar información adicional relacionada indirectamente con él (resúmenes, biografías del autor, enlaces relacionados, etc.)

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Página con secciones html5</title>
</head>
<body>
    <header>
        
        <h1>Titulo principal de nuestra web</h1>
    </header>
    <nav>
        <a href="#">Home</a>
        <a href="ventas.html">Ventas</a>
        <a href="sobrenosotros.html">About us</a>
    </nav>
    <main>
        <section>
            <h2>Venta de bicis</h2>
            <article>
                <!-- Ventas de bicis -->
            </article>
        </section>
        <section>
            <h2>Venta de componentes</h2>
            <article>
                <!-- Ventas de componentes -->
            </article>

```

```

    </section>
</main>
<aside>

    <!-- Anuncios a patrocinadores -->

</aside>

...
</body>
</html>

```

### 7.8.1.7 Pie de página <footer>

Este elemento representa un grupo de contenido al final de una página, como el autor, dirección de contacto, licencia, condiciones de uso, etc. Este elemento puede estar incluido también en `<article>`, `<section>`, `<nav>` y `<aside>` entonces se entiende que es el pie de página del elemento que lo contiene.

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Página con secciones html5</title>
</head>
<body>
    <header>
        
        <h1>Titulo principal de nuestra web</h1>
    </header>
    <nav>
        <a href="#">Home</a>
        <a href="ventas.html">Ventas</a>
        <a href="sobrenosotros.html">About us</a>
    </nav>
    <main>
        <section>
            <h2>Venta de bicis</h2>
            <article>
                <!-- Ventas de bicis -->
            </article>
        </section>
        <section>
            <h2>Venta de componentes</h2>
            <article>
                <!-- Ventas de componentes -->
            </article>
        </section>
    </main>
    <aside>

        <!-- Anuncios a patrocinadores -->
    </aside>

```

```

</aside>
<footer>
    <p>Creado por: <a href="mailto:rmedina@iessanvicente.com">Rosa Medina</a></p>
</footer>
</body>
</html>

```

## 8 Validación de nuestra web

Antes de continuar, vamos a ver cómo podemos comprobar si el código web que estamos haciendo se ajusta a los estándares establecidos. Por ejemplo, se nos olvida cerrar una etiqueta o estamos insertando elementos donde no es posible hacerlo.

Para evitar errores y asegurarnos que nuestro código está correctamente escrito podemos utilizar <https://validator.w3.org/>. Es un sistema que analiza nuestro código y nos dice el número de errores encontrados así como una breve descripción de cada uno.

Este validador de HTML5 todavía está en fase experimental aunque puede utilizarse a día de hoy sin problemas.

Tenemos tres posibilidades a la hora de validar nuestro HTML en NuHTML Checker

- **Address:** indicamos la dirección URL de la página que se quiere validar. Obviamente, es necesario que la página esté colgada en Internet.
- **File Upload:** Seleccionando el archivo HTML de nuestro PC y subirlo al validador para analizarlo.
- **Text Input:** copiando el código de nuestro HTML directamente en el validador.

Vamos a ver un ejemplo de cómo funciona. Si accedemos a la web del [instituto](#):

**Nu Html Checker**

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <https://iessanvicente.com/>

Checker Input

Show ☐ source ☐ outline ☐ image report

Check by

<https://iessanvicente.com/>

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

- Error** An `img` element must have an `alt` attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#).  
From line 34, column 1; to line 34, column 72  
`<p> &n`
- Error** The `align` attribute on the `img` element is obsolete. [Use CSS instead](#).  
From line 108, column 5; to line 108, column 119  
`</a>`

Document checking completed.

Used the HTML parser. Externally specified character encoding was UTF-8.  
Total execution time 463 milliseconds.

Podemos observar que nos está marcando dos errores, y ambos relacionados con el elemento `<img>`, el primero de ellos nos indica que le falta el atributo `alt` y el segundo de ellos que el atributo `align` está obsoleto y debemos utilizar CSS en su lugar.

Con el objetivo de poder crear código HTML de calidad y sin errores debemos utilizar el validador y que sea más probable que se vea bien en cualquier dispositivo.

Si nos fijamos, podemos observar que en la herramienta de validación en la parte superior, hay 3 casillas disponibles para marcar con las que podemos agregar nuevas características que se mostrarán tras la validación. Dichas características son:

- **Source:** Nos muestra el código HTML numerado.
- **Outline:** Nos muestra una especie de árbol o esquema de la página
- **Image report:** Genera un informe de las imágenes de nuestra web con una miniatura y sus características.

## 8.1 Como validar utilizando VSCode

En VSCode podemos encontrar una [extensión](#) que nos permite analizar los documentos directamente desde el editor sin necesidad de acceder a la web anterior. Una vez que tenemos instalado el paquete, a la izquierda en la parte de abajo nos aparecerá un icono nuevo llamado W3C validation.

```

ejemplohtml5.html - Tema 2 - Visual Studio Code
Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
ejemploCitas.html EjemploTablas.html ejemploOtrasEtiquetas.html ejemplohtml4.html ejemplohtml5.html x
ejemplohtml5.html > html > body > footer
1 <!DOCTYPE html>
2 <html lang="es">
3
4 <head>
5   <meta charset="UTF-8">
6   <title>Página con secciones html5</title>
7 </head>
8
9 <body>
10   <header>
11     
12     <h1>Titulo principal de nuestra web</h1>
13   </header>
14   <nav>
15     <a href="#">Home</a>
16     <a href="ventas.html">Ventas</a>
17     <a href="sobrenosotros.html">About us</a>
18   </nav>
19   <main>
20     <section>
21       <h2>Venta de bicis</h2>
22       <article>

```

Terminal: rosa@rosa-desktop: ~\$

Status bar: W3C validation icon (red arrow), Lin. 36, Col. 5 (108 seleccionada), Espacios: 4, UTF-8, LF, HTML, Prettier

Para validar un documento bastará con pulsar sobre el icono y comenzará la validación. En el caso de detectarse errores, aparecerán directamente en el documento. Todos los errores y advertencias vienen con una descripción que nos ayuda a corregirlos.

## 9 Bibliografía

- <https://www.w3schools.com/html/>
- [https://developer.mozilla.org/es/docs/Learn/Getting\\_started\\_with\\_the\\_web/HTML\\_basics](https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/HTML_basics)
- <https://lenguajehtml.com/html/>