

## 1. Introducción a HTML

Las siglas HTML significan *HyperText Markup Language*; es decir, lenguaje de marcas de hipertexto. HTML es el lenguaje que se utiliza para la creación de páginas web estáticas. Las páginas web estáticas son documentos HTML que se conectan unos con otros a través de enlaces.

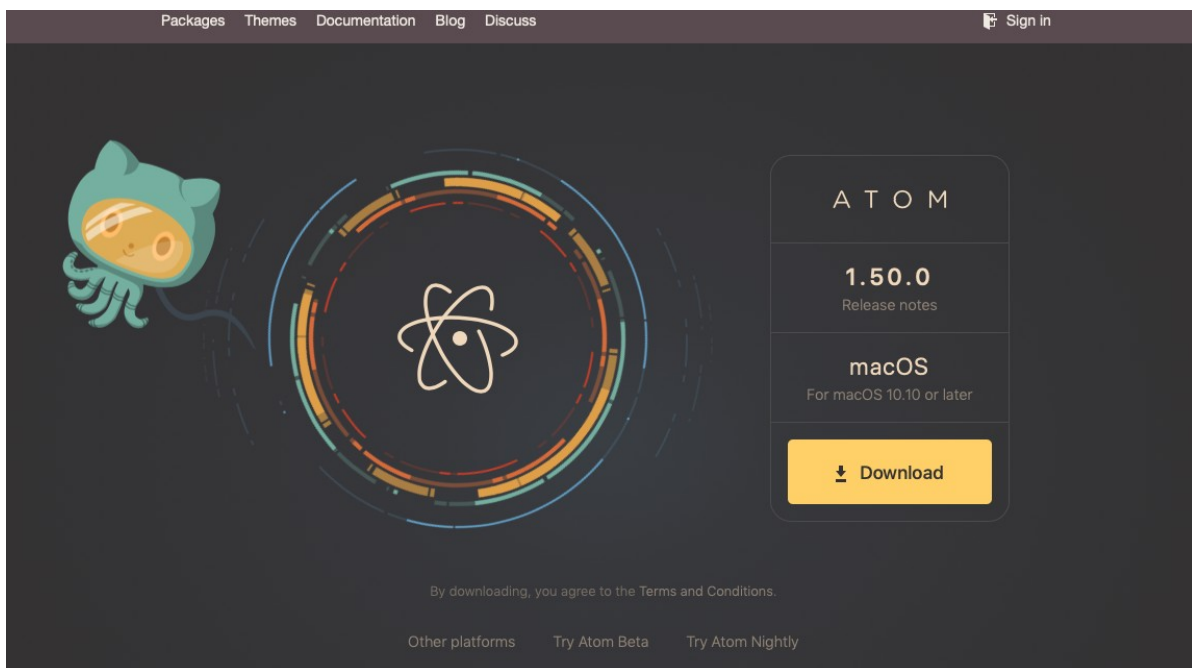


Figura 1. Imagen de ejemplo de un editor para poder crear archivos html

Con el lenguaje HTML vamos a poder incluir los elementos y definir la estructura que forma una página web.

### 1.1. Versiones



HTML 2.0 -> Cuando se produjo la explosión de Internet, el estándar de HTML que circulaba era el 2.0 (establecido en noviembre del 95).

HTML 3.X -> Aunque la versión 2.0 cumplía bien el objetivo para el que se creó, carecía de herramientas para tener un control en las páginas. No se consideró necesario que lo tuviera, ya que por aquel entonces Internet era un fenómeno enfocado a la actividad académica o la empresarial y el contenido se basaba en el diseño. Debido a ello, Netscape (líder del mercado de navegadores por aquel entonces) empezó a incluir etiquetas nuevas no incluidas en ningún estándar.

Esto creó problemas, por lo que el IETF, (el comité que suele decidir todos los estándares dentro de Internet) comenzó a elaborar el borrador del HTML 3.0, que resultó ser demasiado grande para la época, lo que atrasó su aceptación en el mercado.

Al final, una serie de compañías (entre ellas Netscape, Microsoft, IBM, Sun, etc.) crearon un nuevo comité llamado **W3C** <http://www.w3.org> que es el que actualmente elabora las nuevas versiones del HTML. Su primer trabajo consistió en crear el borrador del HTML 3.2, que incluía muchas de las mejoras que los principales navegadores (Netscape y Explorer) habían introducido en Internet, como son las tablas, los *applets*, etc.

Este borrador fue aprobado en enero de 1997 e inmediatamente el W3C se puso a trabajar en la elaboración del siguiente estándar: el 4.0.

HTML 4.X -> En julio de 1997 se presenta el borrador de este estándar. Por fin se estandarizan los marcos (*frames*), las hojas de estilo y los *scripts* (entre otras cosas). El 17 de diciembre de 1997 dicho borrador fue finalmente aprobado.

HTML 5.X -> **HTML5** es la última versión de HTML. El término representa dos conceptos diferentes: Se trata de una nueva versión de HTML, con nuevos elementos, atributos y comportamientos. Contiene un conjunto más amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser más diversas y de gran alcance. A este conjunto se le llama *HTML5* y *amigos*, a menudo reducido a *HTML5*.

Recientemente, El Consorcio World Wide Web (W3C) y el Grupo de Trabajo de Tecnología de Aplicación de Hipertexto Web (WHATWG) han firmado un acuerdo que establece que los dos grupos colaborarán para desarrollar una versión única de las especificaciones HTML y DOM. ¿Qué nos espera?

## 2. Estructura del documento

La estructura de un documento HTML se puede resumir así:

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

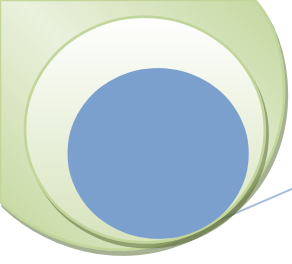
  <head>
    <title>titulo de la página</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>

  <body>

  </body>
</html>
```

⚡ VSCode tiene integrada una extensión llamada Emmet que nos va a ahorrar tiempo ya que tecleando pequeñas abreviaturas (shortcuts) va a generar estructuras o cantidades de código. En el caso de este editor, una vez creado un documento html, tecleando ! O html5 nos creará la estructura básica del documento.

Un documento HTML tiene una estructura que lo separa en dos partes: cuerpo y encabezado. En la primera parte, el cuerpo, encontraremos todo el marcado con contenido visible para el usuario, mientras que en la segunda incluiremos información que no se



suele ser visible pero que puede llegar a ser muy importante.

Comencemos a ver un poco más en detalle las etiquetas de la estructura básica. En un documento HTML5, al no estar basado en SGML tendremos que poner la declaración de la siguiente manera:

### **<!DOCTYPE HTML>**

Lo siguiente que debemos indicar, es que el texto que estamos creando es un documento HTML, para eso encerramos todo el documento entre las etiquetas:

### **<html lang="es"> ... </html>**

El atributo lang no es de carácter obligatorio pero es interesante ponerlo por temas de accesibilidad, traducción...

Seguidamente tendremos la cabecera:

### **<head>...</head>**

- Es el lugar donde se colocan los elementos de la página que no alteran el contenido de la misma, aunque si la forma de presentar la página y su comportamiento en internet. Es por eso, que es un lugar muchas veces recomendable para colocar los scripts y algunos estilos, como veremos en los capítulos correspondientes.

Además del título de la página, que es de carácter obligatorio, uno de los elementos que se pueden incluir aquí son los META. Entre otras cosas, sirven para indicar propiedades de la página como pueda ser el nombre de su autor, palabras clave, etc. Algunos de ellos pueden ser:

#### **Meta Keywords**

***<meta name="keywords" content="frutas, manzana, naranjas, pera">***

Los "keywords" son palabras claves que describen el contenido de tu página web.

#### **Meta Description**

***<meta name="description" content="ventas de frutas frescas a través de internet.">***

La descripción de tu página es muy importante. Debes saber que la mayoría de los motores

de búsqueda solo toman los primeros 150 caracteres de la descripción. Así que debes colocar una descripción bastante concisa y efectiva.

#### **Meta Redirect.**

**`<meta http-equiv="refresh" content="10; url=http://www.iesdoctorbalmis.com">`**

Con este meta podemos redireccionar al usuario de una página a otra. Por ejemplo: En esta META de arriba, al cabo de 10 segundos el usuario será trasladado a la página del instituto.

#### **Meta Refresh**

**`<meta http-equiv="refresh" content="600">`**

Recarga la página periódicamente. El número 600 indica los segundos que se tardará en recargar la página.

#### **Meta Author**

**`<meta name="author" content="el autor">`**

#### **Meta Generator**

**`<meta name="generator" content="program name and version" >`**

Es añadido automáticamente por la mayoría de los programas para crear páginas web y nos indicaría la herramienta con que hemos creado la página.

#### **Meta Juego de caracteres**

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

Se añade para que los navegadores interpreten las tildes y caracteres como la ñ.

#### **Meta ViewPort**

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

Nos va a ayudar a configurar la página web para adaptarse a múltiples dispositivos

Por último tenemos la etiqueta body:

**`<body>...</body>`**

En este lugar se incluirá todo lo que mostraremos en la página, ahora no vamos a explicar todas las etiquetas que pueden entrar dentro del cuerpo (se hará posteriormente)

### 3. Características

#### 3.1. ¡Hola mundo!

Un ejemplo, de lo que sería un primer documento en HTML sería el 'Hola mundo'.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Hola mundo</title>
  </head>
  <body>
    <p>HOLA MUNDO!</p>
  </body>
</html>
```

#### 3.2. Características generales

El lenguaje HTML se basa en la sintaxis SGML (Standard Generalized Markup Language). Esto quiere decir que cualquier información que adjuntemos en HTML estará encerrada entre dos etiquetas de esta manera:

**<etiqueta atributos> ... </etiqueta>**

Un atributo no es más que un parámetro incluido en el interior de la etiqueta de inicio que ayuda a definir la estructura del elemento de una forma más específica.

Los atributos tienen sus valores indicados entre comillas ("), pero si no los indicamos entre comillas también funcionará en la mayoría de los casos. Sin embargo, es aconsejable que siempre las pongamos: para acostumbrarnos a utilizarlas, para dar homogeneidad a nuestros códigos y para evitar errores futuros en sistemas más inflexibles.

Con HTML vamos a poder definir la estructura del documento y agregar los elementos visibles dentro de una página web, usando esas etiquetas y atributos. Es un lenguaje que tiene una sintaxis poco estricta, por ejemplo, no diferencia entre mayúsculas y

minúsculas (nosotros pondremos las etiquetas en minúsculas), sin embargo deberemos seguir unas guías de estilo por lo ya comentado. Se considera además fácil de aprender.

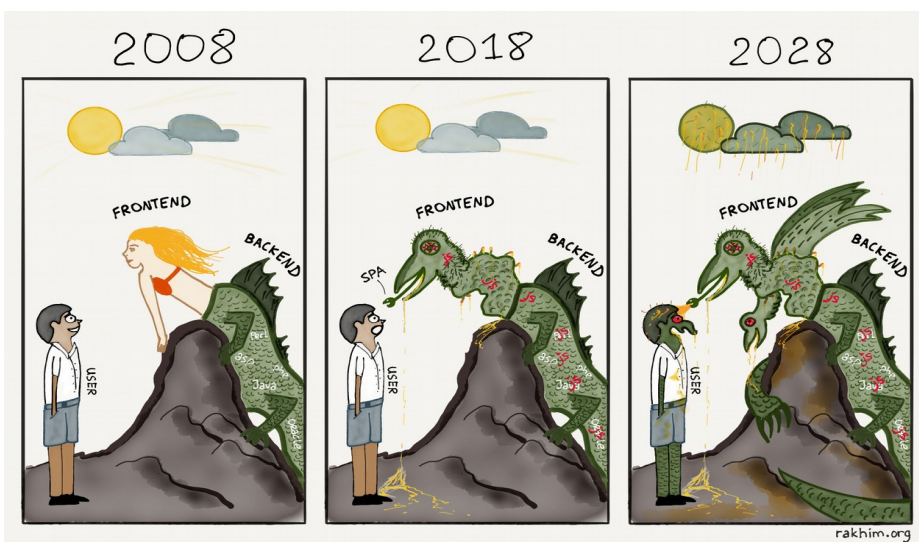
El código dentro de los navegadores es visible. La mayoría de los navegadores tienen una opción en el menú ver con la opción “Ver código fuente” o incorporan una herramienta de inspeccionar código fuente.

The screenshot shows the website of I.E.S. Doctor Balmis. The header includes the school's name, a search bar, and social media links. The main content area features a banner for the bicentennial (1819-2019) and a news item about a blood donation campaign. A table titled 'INFORME DE RESULTADOS DE LA CAMPAÑA DE DONACIÓN DE SANGRE' is displayed, showing data for a collection on 17/03/2022. The Chrome DevTools 'Elements' panel is open on the right, showing the HTML structure of the page, including the head with meta tags and the body content.

Fecha	Colecta	Personas Presentadas	Donaciones realizadas	Donantes de primera vez
17/03/2022	Hall planta baja	50	44	19

El código se puede copiar, pero este código sólo es la estructura e igualmente se podrá copiar el estilo, o los pequeños Scripts. Esto es posible debido a que este código se está ejecutando en el lado del cliente, no en el lado del servidor y disponemos de él.

Relacionado con esto tenemos los conceptos de [front-end y back-end](#).



#### 4. Etiquetas

Las etiquetas son marcas que vamos a usar para estructurar la información. Es importante tener presente que el objetivo de HTML es definir la estructura del documento, el esqueleto de la página web, no el aspecto, pues eso se considera estilo y la tecnología a usar sería CSS.



Con Emmet, podremos insertar las etiquetas en el editor sin teclear los símbolos de `<` o `>` con lo cual ganaremos mucha rapidez. Pondremos sólo el nombre de la etiqueta y ya nos saldrán las sugerencias, que seleccionaremos y aparecerán en nuestro documento con sus atributos obligatorios o usuales.

##### 4.1 Texto

A continuación se explican algunas de las etiquetas destinadas a estructurar el contenido textual. Recuerda que hay muchas más (consulta la CheatSheet propuesta), además de poder informarte en la [documentación oficial](#).

Etiqueta	Utilidad
<code>&lt;em&gt;</code>	El objetivo es dar énfasis a la palabra No implica un cambio



	de voz, para eso se utilizaría <i>
<strong>	Destacar un texto importante, con importancia semántica
<mark>	Subrayar texto
<small>	Texto más pequeño para licencias, etc.
 	Dar un salto de línea dentro de un párrafo, por ejemplo, cuando estás escribiendo un poema o una dirección postal.
<span>	Para formatear una palabra dentro de un párrafo
<b>	Énfasis. Palabra clave.
<i>	Cambiar tono de voz para el contenido
<u>	Para destacar anotaciones.
<sub>,<sup>	Subíndice y superíndice
<code>	Fragmento de código
<var>	Variable de código
<time>	Introducir una fecha

→ texto.html

## 4.2 Listas

Existen varios tipos de listas en HTML. En todas ellas tiene que haber una etiqueta padre que contenga la lista e indique el tipo y luego una etiqueta por cada ítem de la lista.

### Listas desordenadas

La etiqueta <ul> nos permite presentar listas de elementos sin orden alguno. Cada

elemento de la lista irá normalmente precedido por un círculo, pero veremos que esto se puede cambiar. Por ejemplo:

```
<ul>
  <li>Primer elemento</li>
  <li>Segundo elemento</li>
</ul>
```

- Primer elemento
- Segundo elemento

### Listas ordenadas

La etiqueta <ol> nos permite presentar listas de elementos ordenados de menor a mayor. Normalmente cada elemento de la lista irá precedido por su número en el orden o una letra. Por ejemplo:

```
<ol>
  <li>primer elemento</li>
  <li>segundo elemento</li>
</ol>
```

1. Primer elemento
2. Segundo elemento

La etiqueta <ol> admite una serie de atributos:

Atributo	Valores
Start	Especifica el valor de inicio de la lista
Type	Especifica el tipo de lista
Reversed	Indica que el orden debe ser descendente

Listas anidadas: Hay a veces que queremos meter una lista dentro de otra. Para eso podemos utilizar la etiqueta <ol> o la etiqueta <ul> dependiendo del tipo de lista que necesitemos, pero siempre tendremos que meter la lista dentro de una etiqueta li.

```
<ol>
  <li>Conceptos básicos
    <ol>
      <li>Sintaxis</li>
      <li>Estructura</li>
    </ol>
  </li>
</ol>
```

1. Conceptos básicos
  1. Sintaxis
  2. Estructura
2. Texto

```

</li>
<li>Texto</li>
</ol>

```



### 4.3 Enlaces

Un enlace es una zona de texto o gráficos que si son seleccionados nos trasladan a otra página o a otra posición dentro del documento actual. Son los enlaces lo que hacen de la telaraña de la *World Wide Web*.

Los Links <a> Para incorporar un enlace hay que utilizar esta etiqueta. Todo lo que encerremos entre <a> y </a>, ya sea texto o imágenes, será considerado como enlace y sufrirá dos modificaciones:

1. Se visualizará de manera distinta en el navegador. El texto aparecerá subrayado y de un color distinto al habitual, y las imágenes estarán rodeadas por un borde del mismo color que el del texto del enlace.
2. Al pulsar sobre el enlace, seremos enviados al recurso que apuntaba el enlace.

Para que el enlace sirva para algo debemos especificarle una dirección o referencia. Lo haremos de la siguiente manera:

```
<a href="dirección">Púlsame</a>
```

La dirección estará en formato URL (*Uniform Resource Locator*). Una URL nos indica tanto una dirección de Internet como el servicio que esperamos nos ofrezca el servidor al que corresponde la dirección. Tiene el siguiente formato:

```
servicio://servidor:puerto/ruta/pagina.html
```

Donde el servicio podrá ser uno de los siguientes:

**http**

*Es el servicio para transmitir páginas web y el que usaremos normalmente en los enlaces.*

*Es una innovación sobre el anterior, que nos permite acceder a servidores (generalmente comerciales) que nos ofrecen el uso de técnicas de encriptación para proteger los datos que intercambiamos con terceras personas.*

**ftp** *Permite transmitir files desde una computadora a otra. En general el navegador se encargará de mostrarnos el contenido del mismo para que podamos transferir las páginas en internet cómodamente.*

**mailto** *Para poder mandar un correo electrónico.*

*Existe otro formato de URL. Cuando queremos acceder a una página en nuestra computadora:*

***C:\My Documents\html\index.html***

Las rutas pueden [ser relativas o absolutas](#). Debemos usar rutas relativas.

No dejes de consultar los atributos de la etiqueta <a> en documentación oficial:

[https://www.w3schools.com/tags/tag\\_a.asp](https://www.w3schools.com/tags/tag_a.asp)

→ [enlaces.html](#)

## 4.4 Imágenes

Para incluir gráficos e imágenes en nuestras páginas utilizaremos la etiqueta <img>

de esta manera:

```

```

El parámetro src especifica el nombre de referencia que contiene el gráfico, la dirección. En este sentido habrá que tener en cuenta dónde está guardada la imagen.

El parámetro alt especifica el texto que se mostrará en lugar del gráfico en aquellos navegadores que no sean capaces de mostrarlos, cuando el usuario los haya desactivado o cuando la red es lenta y tardan en cargarse. Este atributo es de carácter obligatorio además muy interesante para aumentar la [accesibilidad](#) de nuestra web.

Algunos navegadores muestran la descripción cuando pasamos el mouse por encima de la imagen, para eso usaremos el atributo title.

Existe un atributo que conviene indicar: la anchura del gráfico en píxeles. De este modo, el navegador puede mostrar dibujar la imagen correctamente mientras la va leyendo de la red y así poder mostrar el resto de la página. Destacar que no hay que poner unidades de medida en este atributo.

```

```

Se ve así:



Imágenes y enlaces Suele ser común incluir enlaces dentro de un gráfico. Esta es la forma:

```
<a href="https://www.rtve.es/noticias/20190808/the-beatles-abbey-road-imagen-inmortalizo-paso-cebra-mas-famoso-mundo/1976826.shtml">
  
</a>
```

→ [enlaces.html \(último ejemplo\)](#)

## 4.5 Formularios

Los formularios nos permiten recoger información introducida por el usuario de nuestra web.

Todos los formularios están encerrados por `<form>` y `</form>`. Dentro de esta etiqueta, se definen los campos del formulario. Cada vez que queramos crear un formulario, para nosotros, será obligatorio el uso de esta etiqueta sin embargo existe la posibilidad de poner etiquetas de control fuera del formulario. Los campos que estén fuera de un contenedor form pueden vincularse a un formulario con el atributo form, pero esto no es recomendable. Está prohibido anidar etiquetas form (introducir una etiqueta form dentro de otra) puesto que los comportamientos pueden ser impredecibles.

Vamos a ver un ejemplo de etiqueta `<form>`, suponiendo que va a ser un formulario que se enviará por correo electrónico.

Dentro de esta etiqueta podemos encontrar dos atributos. Los datos al servidor se pueden enviar mediante JavaScript, forma más habitual hoy en día, y estos atributos no harían falta. Pero si el envío de información se hace mediante otra forma deberemos incluir:

- **action="ejemplo.php"** recoge qué se va a hacer una vez que se pulse el botón de enviar (Submit). En ese ejemplo, el formulario se envía al servidor al script ejemplo.php que procesará los datos recibidos.

- **method="post"** especificamos cómo va a ser enviada la información. Los dos valores posibles que puede tomar este atributo son:
  - GET: Envía los datos encadenados al URL especificado en action, para ello utiliza como separador de las variables el símbolo "?".
  - POST: El método POST pasa los datos codificados en el flujo de datos HTTP y los datos no son visibles directamente para el usuario (aunque sí se podrían capturar con un sniffer)

### Campos del formulario

La mayoría de los campos de un formulario se crea con una sola etiqueta, `<input>`, y mediante su parámetro `type` especificamos el tipo de campo que queremos. Un campo de texto básico quedaría así:

**`<input type="text" name="nombre" size="20" placeholder="Introduzca aquí su nombre" />`**

Algunos ejemplos de los valores de input:

Código HTML	Resultado
<code>&lt;input type="text" name="nombre" placeholder="Federico Fdez."&gt;</code>	<input type="text" value="Federico Fdez."/>
<code>&lt;input type="submit" value="Enviar E-mail"&gt;</code>	<input type="submit" value="Enviar E-mail"/>
<code>&lt;input type="Password"&gt;</code>	<input type="password" value="....."/>
<code>&lt;input type="radio" id="male" name="gender" value="male"&gt;</code>	<input type="radio"/> Male
<code>&lt;input type="checkbox" id="vehicle1" name="vehicle1" value="Bike"&gt;&lt;label for="vehicle1"&gt; I have a car&lt;/label&gt;&lt;br&gt;</code>	<input type="checkbox"/> I have a car
<code>&lt;input type="color" id="favcolor" name="favcolor" value="#ff0000"&gt;</code>	<input type="color" value="#ff0000"/>

Dentro de los campos input podemos encontrar una multitud de atributos (los puedes consultar en la documentación oficial o en la chuleta proporcionada) que permiten dotar a

los formularios de ciertas restricciones y ayudas para que el usuario los rellene correctamente.

### Etiquetas para campos

Para asociar un texto a su campo de texto correspondiente, debemos hacerlo de la forma correcta que será utilizando la etiqueta **label**, un ejemplo:

```
<form>
  <label for="nombre ">Nombre :</ label >
  <input type ="text " name =" nombre " id=" nombre " />
</form>
```

**for** es un atributo que se usa para almacenar el id de la etiqueta a la cual hacemos referencia

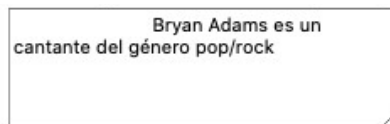
### Áreas de texto

Con las áreas de texto damos a nuestros visitantes la posibilidad de introducir texto con varias líneas. La etiqueta a usar es `<textarea>`, y su funcionamiento es un poco diferente al de `<input>`. La etiqueta `<textarea>` dispone de los atributos `id`, `name` y `title`, que funcionan como en el resto de campos de formulario.

Además, disponemos de otros dos, para indicar las dimensiones del área de texto: **cols** se encarga de establecer el ancho (medido en caracteres) y **rows** el alto, medido en líneas. Como ejemplo:

```
<form>
  <label for="comentario">¿Algún comentario ?</label>
  <textarea name ="comentario" id="comentari " cols ="30" rows ="5">
    Bryan Adams es un cantante del género pop/rock </textarea>
</form>
```

¿Algún comentario ?





El texto que hay entre la etiqueta de apertura y la de cierre es el valor por defecto que contendrá el campo; en este caso “Bryan Adams es un cantante del género pop/rock”.

### Elementos fieldset y legend

El elemento fieldset lo usaremos para crear grupos de controles que comparten el mismo propósito con fines semánticos. Visualmente, veremos una caja que agrupa los controles, pero recordamos que el objetivo de HTML no es el estilo si no la estructura del sitio web, por lo que no nos guiaremos por el aspecto gráfico para unir los controles si no por su significado. Podemos etiquetar el elemento fieldset usando un campo legend justo debajo de la etiqueta . Legend debe describir formalmente el propósito del elemento fieldset.

```
<form><fieldset name="Person">
```

```
  <legend>Employee Details</legend>
```

```
  <label>Employee Id</label><input type="text" name="EmpId">
```

```
  <label>Employee FirstName</label><input type="text" name="Emp  
  FName">
```

```
  <label>Employee LastName</label><input type="text" name="EN">
```

```
  <label>Employee Address</label><input type="text" name="EmpAdd">
```

```
</fieldset>
```

```
<fieldset>
```

```
  <legend>Programming Skills</legend>
```

```
  <label>Programming Language</label><input type="text" name="progl">
```

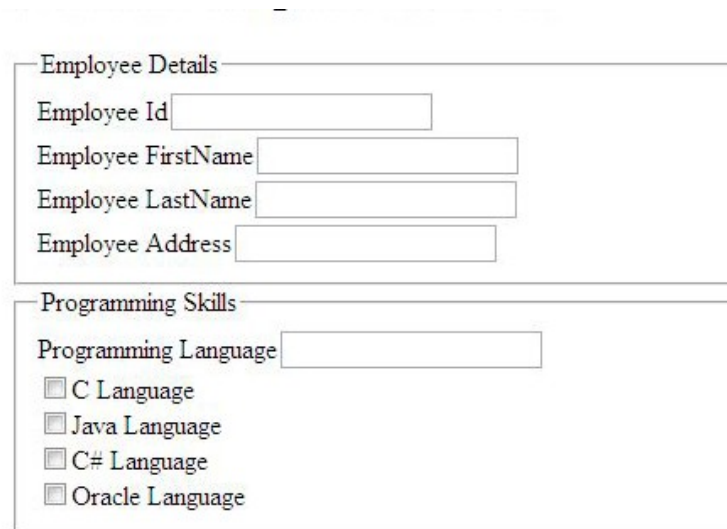
```
  <input type="checkbox" name="c"><label>C Language</label>
```

```
  <input type="checkbox" name="Java"><label>Java Language</label>
```

```
  <input type="checkbox" name="C#"><label>C# Language</label>
```

```
  <input type="checkbox" name="Oracle"><label>Oracle Language</label>
```

```
</fieldset></form>
```



### Listas de selección

Las listas de selección tienen una función parecida a los radio buttons, ya que nos presentan múltiples opciones agrupadas en las que escogemos una. La diferencia es que la lista aparece replegada y no ocupa apenas espacio en la web, así que son muy útiles cuando tenemos muchas opciones a elegir.

La etiqueta que las crea es **<select>** y tiene etiqueta de cierre. Entre ellas, insertamos las opciones que tenemos con la etiqueta **<option>**.

```
<form>
<label for=" peli ">Grupo preferido:</ label >
<select name =" grupo " id=" grupo ">
  <option value ="lzal">lzal </ option >
  <option value ="oreja">La Oreja de Van Gogh </ option >
  <option value ="weeknd" selected =" selected "> The Weeknd</option>
</ select >
</form>
```

Grupo preferido:



Con `selected="selected"` indicamos cuál es la opción por defecto.

→ Formularios.html

## 4.6 Tablas

Las tablas las utilizaremos para presentar datos estadísticos o información que deseemos ver en este formato. No las debemos utilizar para maquetar, pues para eso hoy en día se usa la opción `display` que aplicaremos a contenedores que nos ayudará a adaptar el diseño responsive.

Las tablas se definen de la siguiente manera. Primero, las características de la tabla, luego las de cada fila y dentro de ésta, cada celda. Así pues, una tabla con 2 filas y 3 columnas se declarará así:

1, 1	1, 2	1, 3
2, 1	2, 2	2, 3

⚡ Aunque en la imagen se pueden ver los bordes, realmente no se visualizarían. Existe un atributo, `border`, que actualmente no forma parte del standard aunque puede usarse ya que en versiones anteriores era normativo.

Dentro de la etiqueta `td` podemos indicar dos atributos:

- colspan** Especifica el número de celdas de la fila situadas a la derecha de la actual que se unen a ésta (incluyendo la celda en que se declara este parámetro).
- rowspan** Especifica el número de celdas de la columna situadas debajo de la actual que se unen a ésta.

En muchas ocasiones podemos necesitar estructurar los datos con mayor complejidad, diferenciando secciones como encabezados, pies... Es por ello que aparecen otras etiquetas interesantes relacionadas con las tablas:

- `<caption>`: sirve para definir un título descriptivo a las tablas.
- `<colgroup>`: se usa para aplicar estilos a columnas
- `<col>`: Establece cada estilo dentro del colgroup
- `<thead>`: Agrupa las columnas que forman parte de la cabecera de la tabla.
- `<tbody>`: Agrupa las columnas que forman parte del cuerpo de la tabla
- `<tfoot>`: Agrupa las columnas que forman parte del pie de la tabla.
- `<th>`: Primera fila, donde dentro de cada celda especificamos el título del contenido de esa columna. Para eso, en esas celdas, usaremos la etiqueta `<th>` en vez `<td>`.

```
<table border="1">
  <thead><tr>
    <th>A</th>
    <th>B</th>
  </tr></thead>
  <tr>
    <td>C</td>
    <td>D</td>
  </tr>
  <tr>
    <td>E</td>
    <td>F</td>
  </tr>
</table>
```

A	B
C	D
E	F



## 4.8 Multimedia

Con la última versión de HTML5 podemos incorporar algunos elementos de carácter multimedia a nuestra página web, que cada vez cobran más protagonismo.

- **AUDIO:** La etiqueta para incorporar un audio es la etiqueta `audio`. Con el atributo `src` podemos decirle el archivo a reproducir. Igualmente dentro de esta etiqueta podemos incorporar la etiqueta `source` para incorporar archivos alternativos (normalmente el mismo audio en el mismo formato porque si el navegador no lo soporta). Es importante también incorporar el atributo `controls` para que desde el navegador esta audio sea visible.

```
<audio src="a.mp3" controls>Texto para los navegadores que no soportan audio</audio>
```

Fíjate en que no todos los formatos son compatibles en todos los navegadores, por eso es importante conocer el uso de la etiqueta `source`:

Browser	MP3	WAV	OGG
Edge/IE	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	YES	NO
Opera	YES	YES	YES

- **VIDEO:** La etiqueta para incorporar un video es la etiqueta `video`. Con el

atributo src podemos decirle el archivo a reproducir. Igualmente dentro de esta etiqueta podemos incorporar la etiqueta source para incorporar archivos alternativos (normalmente el mismo audio en el mismo formato porque si el navegador no lo soporta). Es importante también incorporar el atributo controls para que desde el navegador este video sea visible.

```
<video width="" src="a.mp4" controls></video>
```

Fíjate en que no todos los formatos son compatibles en todos los navegadores:

Browser	MP4	WebM	Ogg
Edge	YES	YES	YES
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	YES	NO
Opera	YES	YES	YES

- Imágenes SVG: Al igual que puedo agregar imágenes jpg, png... Puedo cargar imágenes vectoriales SVG. La ventaja es que puedo obtener su código. Con programas de edición gráfica podemos guardarlas en formato código. Nos incorporará la etiqueta SVG que nos permitirá como en un lienzo, etiquetar las características que al final forma la imagen. Piensa en la ventaja de esto, ya que la imagen en si no viaja por la red. Puedes utilizar tus imágenes SVG que veas por la red o que tengas diseñadas por ti y arrastrándolas al editor obtendrás su código para utilizarla.

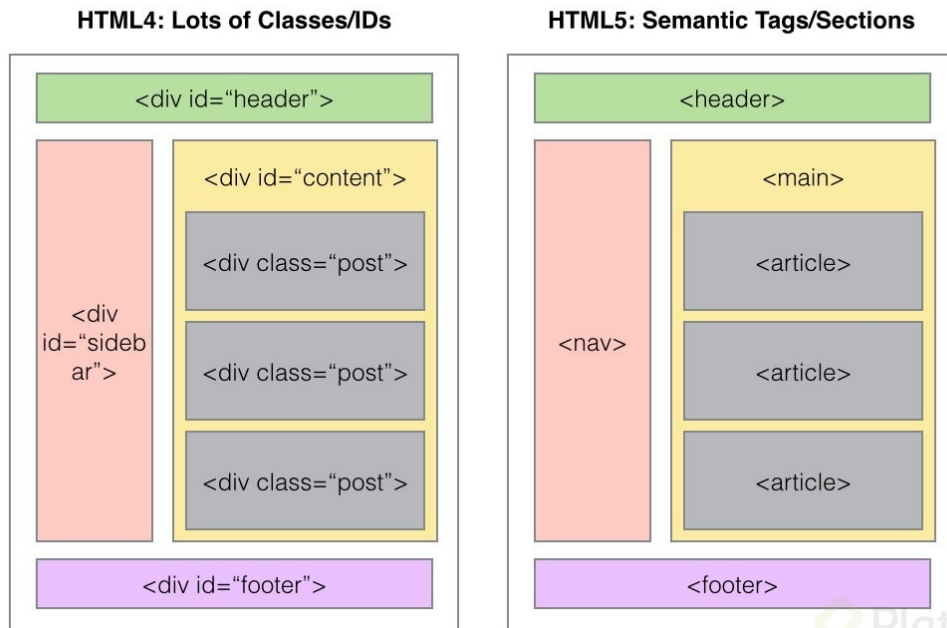


#### 4. Maquetación web

Con HTML5 aparecen una serie de nuevas etiquetas que hacen la función de etiquetas de maquetado, es decir, etiquetas contenedoras que van a incluir todos los elementos HTML vistos hasta ahora. A esto se le conoce con el nombre de maquetación semántica. Es importante hacer uso de etiquetas contenedoras para organizar la información ya que después con CSS podremos organizarla visualmente.

Mientras que con HTML4.0 todo se contenía en capas DIV en HTML5 aparecen distintos tipos de contenedores. Estos nos van a ayudar a estructurar la página según el significado (la semántica) de las partes, proporcionarán información extra a los motores de búsqueda y evitarán problemas de [divitis](#) que aportan dificultad de mantenimiento y de comprensión del marcado.

## HTML4 vs HTML5 Page Structure on a Blog



A continuación se explican algunas de las etiquetas de maquetación semántica más utilizadas:

- **Header:** define el bloque de la cabecera de la página web
- **Nav:** define un bloque que representa un menú, por lo tanto los buscadores al encontrar esta etiqueta saben que es la estructura de navegación.
- **Hgroup:** define y agrupa títulos y así darle distintas importancia a cada título.
- **Section y article:** Los anteriores elementos que definen capas <div> y se cambian por las etiquetas section y article, que tienen un sentido más semántico al definir secciones y artículos dentro de las secciones.
- **Aside :** define y agrupa elementos de tipo tangencial de la pagina web. Es de contenido que no es el principal como por ejemplo las columnas laterales-
- **Footer:** define y agrupa elementos en el pie de pagina de la web.



Puede llevar a confusión la diferencia entre div, article y section. Vamos a aclarar unas cuantas consideraciones:

- Div: contenedor general. En él pondremos todo aquello que queramos poner de manera general.
- Section: Sección hace referencia a las partes de el contenedor que estemos trabajando.
- Article: Contenido textual de la página, artículos.

→ [maquetación.html](#)

## 5. Otras etiquetas

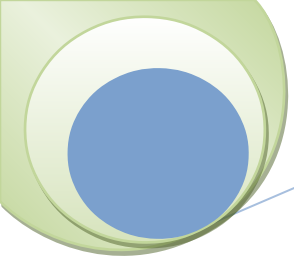
Si consultas la documentación, verás que hay bastantes etiquetas más que no hemos aprendido. No pasa nada, puesto que seguiremos usando etiquetas en los próximos temas. Pero entre las que quedan destacar la etiqueta `<iframe>`. La etiqueta `<iframe>` sustituye a los antiguos frames. Mientras que antiguamente, se utilizaban para maquetar, hoy en día no debes usarlos, están altamente no recomendados. Esta etiqueta, que se usa para adjuntar otras páginas web a nuestra página, la veremos usada para adjuntar vídeos de youtube, vimeo... publicidad, etc.

Para insertar un vídeo:

1. Visitar página del vídeo
2. Escoger opción compartir
3. Copiar código. Observar que es la etiqueta `<iframe>`...

## 6. Bibliografía

<https://www.w3schools.com>



HTML 5

<https://www.w3.org/TR/html52/>