



HTML5

HTML 5 - <!DOCTYPE>

- Esta etiqueta indica el tipo de documento que estamos creando.
- Para indicar que es un documento de HTML5 deberemos poner:

`<!DOCTYPE html>`

- **IMPORTANTE:** Esta línea debe ser la primera línea del archivo, sin espacios o líneas que la precedan. De esta forma, el modo estándar del navegador es activado y las incorporaciones de HTML5 son interpretadas siempre que sea posible, o ignoradas en caso contrario.

HTML 5 - `<html>`

- Luego de declarar el tipo de documento, debemos comenzar a construir la estructura HTML.
- Como siempre, la estructura tipo árbol de este lenguaje tiene su raíz en el elemento `<html>`.
- Este elemento envolverá al resto del código.

```
<!DOCTYPE html>
```

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

```
</html>
```

- El atributo **lang** en la etiqueta de apertura `<html>` es el único atributo que necesitamos especificar en HTML5. Este atributo define el idioma humano del contenido del documento que estamos creando, en este caso “es” por español.

HTML 5 - <header>

- Proveer información introductoria (títulos, subtítulos, logos), pero Mientras que el elemento **<head>** tiene el propósito de proveer información acerca de todo el documento, **<header>** es usado solo para el cuerpo o secciones específicas dentro del cuerpo .

```
<!DOCTYPE html>
<html lang="es">
<body>
<header>
<h1>Este es el título principal del sitio web</h1>
</header>
</body>
</html>
```

HTML 5 - <nav>

- Hace referencia a la **barra de navegación**. Esta barra es generada en HTML5 con el elemento **<nav>**:.

```
<html>
  <body>
    <nav>
      <ul>
        <li>principal</li>
        <li>fotos</li>
        <li>videos</li>
        <li>contacto</li>
      </ul>
    </nav>
  </body>
</html>
```

HTML 5 - <section>

- La etiqueta **<section>** y su correspondiente etiqueta de cierre **</section>** se utilizan para encerrar el código correspondiente a una sección genérica dentro de un documento o aplicación:

```
<html>
  <body>
    <section>
      <!--contenido-->
    </section>
  </body>
</html>
```

HTML5 - <aside>

- Es usado para referenciar una sección que normalmente contiene datos relacionados con la información principal pero que no son relevantes o igual de importantes, suele contener la barra lateral y por consiguiente, ubicarse en una columna lateral.

<aside>

`<blockquote>Mensaje número uno</blockquote>`

`<blockquote>Mensaje número dos</blockquote>`

</aside>

HTML5 - <footer>

- Este elemento es usado para finalizar la construcción de la plantilla, introduciendo un final al cuerpo del documento a pie de pagina.

<footer>

Derechos Reservados © 2010-2011

</footer>

(Ver ejemplo 1)

HTML 5 - <article>

- Del mismo modo que los blogs están divididos en entradas, sitios web normalmente presentan información relevante dividida en partes que comparten similares características. El elemento **<article>** nos permite identificar cada una de estas partes.

```
<body>
```

```
<section>
```

```
  <article>
```

```
    Este es el texto de mi primer mensaje
```

```
  </article>
```

```
...
```

HTML 5 - <article>

- Como puede observarse en el código las etiquetas **<article>** se encuentran ubicadas dentro del elemento **<section>**.

...

```
<article>
```

```
    Este es el texto de mi segundo mensaje
```

```
</article>
```

```
</section>
```

```
</body>
```

HTML 5 - <hgroup>

- El elemento **<hgroup>** es necesario cuando tenemos un título y subtítulo o más etiquetas H juntas en la misma cabecera. Este elemento puede contener solo etiquetas H.

```
<header>
  <hgroup>
    <h1>Título del mensaje uno</h1>
    <h2>Subtítulo del mensaje uno</h2>
  </hgroup>
  <p>publicado 10-12-2011</p>
</header>
```

HTML 5 - <figure>

- La etiqueta **<figure>** fue creada para ayudarnos a ser aún más específicos a la hora de declarar el contenido del documento.
- Las etiquetas **<figcaption>** encierran el texto relacionado con **<figure>** y establecen una relación entre ambos elementos y su contenido.

```
<figure>
```

```

```

```
  <figcaption>
```

```
    Esta es la imagen del primer mensaje
```

```
  </figcaption>
```

```
</figure>
```

```
<footer>
```

(Ver ejemplo 2)

HTML 5 - <form>

- Los formularios en HTML no han cambiado mucho. La estructura sigue siendo la misma, pero HTML5 ha agregado nuevos elementos, tipos de campo y atributos para expandirlos tanto como sea necesario y proveer así las funciones actualmente implementadas en aplicaciones web.
- Atributos nuevos I: autocomplete, novalidate, placeholder, required.
- Atributos nuevos II: multiple, autofocus, pattern y form.

HTML 5 - <form>

autocomplete :

- Es un viejo atributo que se ha vuelto estándar en esta especificación. Puede tomar dos valores: **on** y **off**. El valor por defecto es **on**. Cuando es configurado como **off** los elementos **<input>** pertenecientes a ese formulario tendrán la función de autocompletar desactivada, sin mostrar entradas previas como posibles valores.
- Puede ser implementado en el elemento **<form>** o en cualquier elemento **<input>** independientemente.

HTML 5 - <form>

- **Ejemplo autocomplete**

```
<form>
```

```
  Dato1: <input type="text" name="name1" autocomplete="off">
```

```
  Dato2: <input type="text" name="name2" autocomplete>
```

```
  Dato3: <input type="text" name="name3">
```

```
  <input type="submit" value="enviar">
```

```
</form>
```

HTML 5 - <form>

novalidate:

- Una de las características de formularios en HTML5 es la capacidad propia de validación.
- Los formularios son automáticamente validados. Para evitar este comportamiento, podemos usar el atributo **novalidate**.
- Para lograr lo mismo para elementos **<input>** específicos, existe otro atributo llamado **formnovalidate**.
- Ambos atributos son booleanos, ningún valor tiene que ser especificado (su presencia es suficiente para activar su función).

HTML 5 - <form>

- **Ejemplo novalidate**

```
<form novalidate>
```

```
  <p>Area invalidada</p>
```

```
  <p><label>Nombre: <input type="text" required /></label></p>
```

```
  <p><label>Calle: <input type="text" required /></label></p>
```

```
  <input type="submit" name=enviar value="Guardar" />
```

```
</form>
```

HTML 5 - <form>

placeholder:

- Mediante esta propiedad del Html5 podemos definir el texto que queremos que aparezca dentro del campo del formulario a modo de ayuda para las visitas.
- Por ejemplo, se suele colocar un texto explicativo acerca de la información que se debe rellenar en ese campo.

HTML 5 - <form>

- **Ejemplo placeholder**

```
<form>
```

```
  <input type="text" id="ejemplo" name="ejemplo"  
placeholder="bla bla bla">
```

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

HTML 5 - <form>

required:

- Este atributo booleano no dejará que el formulario sea enviado si el campo se encuentra vacío.
- Por ejemplo, cuando usamos el tipo **email** para recibir una dirección de email, el navegador comprueba si la entrada es un email válido o no, pero validará la entrada si el campo está vacío.
- Cuando el atributo **required** es incluido, la entrada será válida sólo si se cumplen las dos condiciones: que el campo no esté vacío y que el valor ingresado esté de acuerdo con los requisitos del tipo de campo.

HTML 5 - <form>

- **Ejemplo required**

```
<form>
```

```
  <p><label>*Nombre: <input type="text" required  
/></label></p>
```

```
  <p><label>*Calle: <input type="text" required /></label></p>
```

```
  <input type="submit" name=enviar value="Guardar" />
```

```
  <input type="submit" formnovalidate name="guardar"  
value="Guardar despu&eacute;s" />
```

```
  <p>* Campo requerido</p>
```

```
</form>
```

HTML 5 - <form>

multiple:

- El atributo **multiple** es otro atributo booleano que puede ser usado en algunos tipos de campo (por ejemplo, **email** o **file**) para permitir el ingreso de entradas múltiples en el mismo campo.
- Los valores insertados deben estar separados por coma para ser válidos

HTML 5 - <form>

- **Ejemplo multiple**

<form>

Seleccione varias imagenes: <input type="file" name="img" multiple="multiple" />

Entre varios Email:<input type="email" multiple="multiple">

<input type="submit" value="guardar" name="guardar"/>

</form>



HTML 5 - <form>

autofocus:

- El atributo **autofocus** enfocará la página web sobre el elemento seleccionado pero considerando la situación actual.
- No moverá el foco cuando ya haya sido establecido por el usuario sobre otro elemento.

HTML 5 - <form>

- **Ejemplo autofocus**

```
<form>
```

```
Nombre:<input type="text" name="fname" /><br />
```

```
Apellido: <input type="text" name="lname"  
autofocus="autofocus"/><br />
```

```
E-mail: <input type="email" name="email" /><br />
```

```
<input type="submit" />
```

```
</form>
```

HTML 5 - <form>

pattern:

- El atributo **pattern** es para propósitos de validación. Usa expresiones regulares para personalizar reglas de validación.
- Algunos de los tipos de campo ya estudiados validan cadenas de texto específicas, pero no permiten hacer validaciones personalizadas, como por ejemplo un código postal que consiste en 5 números.
- No existe ningún tipo de campo predeterminado para esta clase de entrada.
- El atributo **pattern** nos permite crear nuestro propio tipo de campo para controlar esta clase de valores no ordinarios.
- Puede incluso incluir un atributo **title** para personalizar mensajes de error.

HTML 5 - <form>

- **Ejemplo pattern**

<form>

```
<input pattern="[0-9]{5}" name="codigopostal" id="codigopostal"
title="inserte los 5 números de su c&acute;digo postal"/>
```

```
<input type="submit" />
```

</form>

HTML 5 - <form>

form:

- El atributo **form** es una adición útil que nos permite declarar elementos para un formulario fuera del ámbito de las etiquetas **<form>**.
- Hasta ahora, para construir un formulario teníamos que escribir las etiquetas **<form>** de apertura y cierre y luego declarar cada elemento del formulario entre ellas.
- En HTML5 podemos insertar los elementos en cualquier parte del código y luego hacer referencia al formulario que pertenecen usando su nombre y el atributo **form**

HTML 5 - <form>

- **Ejemplo form**

```
<nav>
```

```
  <input type="search" name="busqueda" id="busqueda"  
    form="formulario">
```

```
</nav>
```

```
<section>
```

```
  <form name="formulario" id="formulario" method="get">
```

```
    <input type="text" name="nombre" id="nombre">
```

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

```
  </form>
```

```
</section>
```

HTML 5 - <input>

- Los nuevos tipos de propiedades que ofrece HTML 5 con respecto a la etiqueta <input> son las siguientes:

- Tipo mail. Es la forma que implementa HTML 5 para validar un campo dedicado a almacenar email, evitando así la necesidad de implementar Javascript. Además, según el navegador, en caso de que no se introduzca un valor correcto, se rodeara con un cuadrado de un color predeterminado.

Ej:

```
<input type="email" name="miemail"
id="miemail">
```

- Tipo search. Esta propiedad unicamente es una indicación para los navegadores.

Ej:

```
<input type="search" name="busqueda"
id="busqueda">
```

HTML 5 - <input>

- Tipo url. Este tipo de campo trabaja exactamente igual que el tipo email pero es específico para direcciones web. Está destinado a recibir solo URLs absolutas y retornará un error si el valor es inválido.

Ej:

```
<input type="url" name="miurl" id="miurl">
```

- Tipo Tel. Este tipo de campo es para números telefónicos. No requiere ninguna Es solo una indicación para el navegador en caso de que necesite hacer ajustes de acuerdo al dispositivo en el que la aplicación es ejecutada.sintaxis en particular.

Ej:

```
<input type="tel" name="telefono"  
id="telefono">
```

HTML 5 - <input>

- Tipo Number. Es sólo válido cuando recibe una entrada numérica. Existen algunos atributos nuevos que pueden ser útiles para este campo:
 - **min:** El valor de este atributo determina el mínimo valor aceptado para el campo.
 - **max:** El valor de este atributo determina el máximo valor aceptado para el campo.
 - **step:** El valor de este atributo determina el tamaño en el que el valor será incrementado o disminuido en cada paso. Por ejemplo, si declara un valor de 5 para step en un campo que tiene un valor mínimo de 0 y máximo de 10, el navegador no le permitirá especificar valores entre 0 y 5 o entre 5 y 10.

Ej:

```
<input type="number" name="numero"
id="numero" min="0" max="10" step="5">
```


HTML 5 - <input>

- Tipo Range. Es una nueva clase de control que no existía previamente. Este nuevo control le permite al usuario seleccionar un valor a partir de una serie de valores o rango. El tipo range usa los atributos min y max estudiados previamente para configurar los límites del rango. También puede utilizar el atributo step para establecer el tamaño en el cual el valor del campo será incrementado o disminuido en cada paso.

Ej:

```
<input type="range" name="numero"
id="numero" min="0" max="10" step="5">
```

(Ver ejemplo 4)

HTML 5 - <input type="date">

- Tipo de campo que genera una nueva clase de control .
- En este caso fue incluido para ofrecer una mejor forma de ingresar una fecha. Algunos navegadores muestran en pantalla un calendario que aparece cada vez que el usuario hace clic sobre el campo.
- El calendario le permite al usuario seleccionar un día que será ingresado en el campo junto con el resto de la fecha.
- Compatible: Chrome, Opera, Safari

<input type="date" name="fecha" id="fecha">

HTML 5 - <input type="week">

- Este tipo de campo es similar al date, permitiendo seleccionar una semana completa.
- El valor esperado tiene la sintaxis 2011-w50 donde 2011 es el año y 50 el numero de semana.
- Compatible:Opera,Safari.

<input type="week" name="sem" id="sem">

HTML 5 - <input type="month">

- Similar al tipo de campo previo, éste es específico para seleccionar meses.
- Normalmente el valor esperado tiene la sintaxis año-mes
- Compatible:Opera,Safari

<input type="month"name="mes" id="mes">

HTML 5 - <input type="time">

- El tipo de campo time es similar a date, pero solo para la hora. Toma el formato de horas y minutos, pero su comportamiento depende de cada navegador en este momento.
- Normalmente el valor esperado tiene la sintaxis hora:minutos:segundos, pero también puede ser solo hora:minutos
- Compatible:Opera,Safari

<input type="time" name="hora" id="hora">

HTML 5 - <input type="datetime">

- El tipo de campo datetime es para ingresar fecha y hora completa, incluyendo la zona horaria
- Compatible: Opera, Safari
- **<input type="datetime" name="fhora" id="fhora">**



HTML 5 - `<input type="datetime-local">`

- El tipo de campo `datetime-local` es como el tipo `datetime` sin la zona horaria.

`<input type="datetime-local" name="hlocal" id="hlocal">`

HTML 5 - <input type="color">

- Además de los tipos de campo para fecha y hora existe otro tipo que provee una interface predefinida similar para seleccionar colores.
- Normalmente el valor esperado para este campo es un número hexadecimal , como #00FF00
- Compatible:Opera,Chrome.

<input type="color" name="color1" id="color1">

(Ver ejemplo 5)

HTML 5 - Datalist



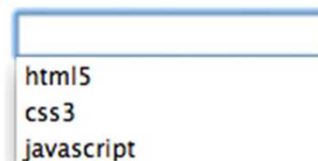
- La etiqueta `<datalist>` nos permite incluir una lista desplegable de valores en un `<input type="text">`
- Ej:
- Primero introducimos el input
`<input type="text" list="tecnologias" />`
- Despues introducimos el datalist, este debe tener el atributo id idéntico al atributo list del input.
`<input type="text" list="tecnologias" />`
`<datalist id="tecnologias"></datalist>`

HTML 5 - Datalist

- Por ultimo agregamos las opciones dentro del datalist como si de un select se tratara.

```
<input type="text" list="tecnologias" />  
<datalist id="tecnologias">  
    <option value="html5" />  
    <option value="css3" />  
    <option value="javascript" />  
</datalist>
```

Resultado:



HTML 5 - Progress



- Esta etiqueta sirve para mostrar una barra de carga o progreso para aplicaciones, imágenes, etc. Se implementa con JS para hacerlo dinámico.

- Ej:

```
<progress value="50" max="100">
```

```
  <p>Su navegador no soporta la tecnologia html5</p>
```

```
</progress>
```

- El atributo value indica hasta que porcentaje ha cargado la tarea.
- El atributo max asigna el valor maximo de la barra de progreso
- Resultado:



HTML 5 - Meter



- Tiene un uso parecido a la etiqueta progress solo que este se emplearía a la hora de representar un rango o ranking. También se usa en conjunto con JS.
- Al igual que con la etiqueta anterior se inserta un mensaje dentro por si el navegador no puede leer la etiqueta.
- Ej:

```
<meter min="0" max="100" value="70">
```

```
  <p>Su navegador no soporta la tecnologia  
html5</p>
```

```
</meter>
```



HTML 5 - Output



- Representa el resultado de un calculo realizado por un script.
- Ej:

```
<form onsubmit="return false"
oninput="res.value=parseInt(op1.value) *
parseInt(op2.value)">
  <input name="op1" type="number" step="any"> x
  <input name="op2" type="number" step="any"> =
  <output name="res" for "op1 op2"></output>
</form>
```

x = 30



HTML 5 - Output

- Los atributos específicos de esta etiqueta son:
- *Name*: asigna un nombre al elemento output.
- *Form*: asocia este elemento con el formulario en el cual está incluido.
- *For*: establece una relación entre el resultado del cálculo y los elementos del mismo.

HTML 5 - Mark



- Se emplea para resaltar partes del texto
- Ej:

`<p>ejemplo de`

`<mark>html5</mark> mark </p>`

ejemplo de **html5** mark

HTML 5 - Small



- Define un texto de fuente mas pequeña
- Ej:

```
<p>Parrafo de ejemplo</p>
```

```
<p><small>texto pequeño</small></p>
```

Parrafo de ejemplo

texto pequeño

HTML 5 - Cite



- Se utiliza para definir el nombre de una obra (libro, cuadro, serie, película, canción, etc..)
- Ej:
 - `<p>No está muerto lo que puede yacer eternamente; y con el paso de los extraños eones por venir, incluso la Muerte puede morir.</p>`
 - `<p><cite>H.P. Lovecraft</cite></p>`

No está muerto lo que puede yacer eternamente; y con el paso de los extraños eones por venir, incluso la Muerte puede morir.

H.P. Lovecraft

HTML 5 - Address



- Se usa para definir la información de un contacto, autor o propietario, etc..
- Debe estar contenido dentro de un elemento `<article>` dentro del `<body>`

- Ej:

```
<address>
```

```
Isidro Rebollo<br>
```

```
Monfragüe ,2<br>
```

```
41704, Sevilla<br>
```

```
</address>
```

Isidro Rebollo

Monfragüe ,2

41704, Sevilla

HTML 5 - Time



- Define una hora o fecha según el calendario gregoriano (aaaa-mm-dd).
- Se emplea el atributo datetime para asignar la fecha u hora al contenido del elemento si este no lo especifica.
- Esta etiqueta no transforma el contenido.
- Ej:
 - `<p>Son las <time>10:00</time> PM </p>`
 - `<p>Ya mismo es <time datetime="2012-12-25">Nochebuena</time>.</p>`

Son las 10:00 PM

Ya mismo es Nochebuena.

HTML 5 - Wbr



- Especifica en un texto o palabra el lugar correcto para un salto de linea.
- En textos muy largos el navegador a veces rompe la frase por el lugar equivocado, con esta etiqueta se corrige eso.
- Ej:

```
<p>supercalifragilisticoespialidoso</p>
```

```
<p>super<wbr>cali<wbr>fragi<wbr>listico<wbr>espia<wbr>lidoso</p>
```

*el uso es mayormente para dispositivos móviles y pantallas o ventanas reducidas

supercalifragilistic

supercalifragi
listicoespia
lidoso

(Ver ejemplo 6)